

# Data Transmission Opportunities for Collaborative Cloud-Based Building Information Modeling

**Kereshmeh Afsari**

Georgia Institute of Technology, USA  
keresh@gatech.edu

**Charles M. Eastman**

Georgia Institute of Technology, USA  
charles.eastman@coa.gatech.edu

**Dennis R. Shelden**

Georgia Institute of technology  
dennis.shelden@coa.gatech.edu

## Abstract

Collaboration within Building Information Modeling process is mainly based on file transfer while BIM data being exchanged in either vendor specific file formats or neutral format using Industry Foundation Classes (IFC). However, since the Web enables Cloud-based BIM services, it provides an opportunity to exchange data via Web transfer services. Therefore, the main objective of this paper is to investigate what features of Cloud interoperability can assist a network-based BIM data transmission for a collaborative work flow in the Architecture, Construction, and Engineering (AEC) industry. This study indicates that Cloud-BIM interoperability needs to deploy major components such as APIs, data transfer protocols, data formats, and standardization to redefine BIM data flow in the Cloud and to reshape the collaboration process.

**Keywords:** BIM; Cloud Computing; Data Transmission; Interoperability; IFC

## Introduction

Collaboration in the architecture, engineering and construction (AEC) industry mainly relies on file transfer while BIM data are being stored and exchanged in the form of files with several formats (Zhang et al., 2014; Shafiq et al., 2013). The formats of these files could be either vendor specific or neutral format using Industry Foundation Classes (IFC) (Zhang et al., 2014) as an open BIM standard.

On the other hand, Cloud-BIM is considered to be the second generation of BIM development and studies suggest that it will produce a major change across the industry although the technology is still relatively new (Wong et al., 2014). Cloud-BIM has caused a new direction in BIM implementation and development to support BIM data generation and consumption among members of the project (Wu & Issa, 2012). By applying Cloud Computing in BIM services, Building Information Modeling can achieve a higher performance with a relatively low cost (Zhang et al., 2014). Therefore, Cloud-BIM technology is believed to lead to higher levels of information interaction and provide an effective cross-disciplinary collaboration (Wong et al., 2014; Juan & Zheng, 2014). While interoperability is the key to the success of Cloud implementation (Wong et al., 2014) and data exchange is critical for successful implementation (Juan & Zheng, 2014), the challenge is that making multiple Cloud-based BIM applications interoperate would be very difficult when they are developed by different vendors (Yang, et al., 2011). Therefore, Challenges of BIM data integration and interoperability are the motivation for this study.

On the other hand, since the Web typically enables Cloud-based services, it performs a critical role than just a Web-based user interface for these services. Most importantly, the

Web can facilitate data sharing and interoperability (Curry, et al., 2013). In fact, Cloud offers an opportunity that software packages can be connected via Web, therefore these services can manage information communication in a different way (Yang, et al., 2011) than the conventional file-based system. This study highlights that currently there is a gap in research with regard to the identification of alternative technologies that can assist Cloud-BIM interoperability solutions. Therefore, the main objective of this paper is to investigate what approaches and techniques of Web technologies exist that can assist with improving BIM data transmission to address a network-based data exchange for a collaborative workflow. Hence, this research underlines recent developments and opportunities to address fundamental challenges of BIM data exchange.

This study first outlines the features and issues of current Cloud-BIM solutions especially with regard to data transmission approaches. Then, IFC data model and data representations as the effort for building information standardization will be discussed. In addition, the study will point out Cloud interoperability components that can be deployed to address current Cloud-BIM data transmission challenges and it eventually specifies a potential direction that future research will most likely undertake.

## Cloud-BIM Development and Issues

Cloud computing as defined by the U.S. National Institute of Standards and Technology (NIST), "is a model for enabling ubiquitous, convenient, on-demand access to a shared pool of configurable computing resources" (NIST, 2014). Cloud computing relocates the computing process and data from desktop to large data centers (Jaeda & Modi, 2012). Thus Cloud computing aims at making a better use of distributed

resources and combining them to achieve higher performance so that large scale computation can be managed (Jaeda & Modi, 2012; Rimal et al. 2009). Because of the potential advantages of Cloud computing, BIM applications are also steadily moving to the Cloud and gradually BIM web services and Cloud-based apps are gaining increased popularity (Wu & Issa, 2012; Berlo, 2016; Shafiq et al., 2013). Some examples of the Cloud-based BIM solutions are GRAPHISOFT BIM Explorer (BIMx), Autodesk BIM360, BIMServer.org developed by TNO and the University of Eindhoven, ONUMA System (Beetz et al., 2011; Wu & Issa, 2012; Wong et al., 2014) and Trimble Connect. Each of these Cloud-based solutions are being developed mostly in isolation and by different vendors and therefore, there is a need to reconsider the approach to interoperability of new Cloud-BIM services otherwise they will suffer from the issues similar to interoperability challenges in conventional desktop-based applications (Curry, et al., 2013).

Previous work (Afsari et al., 2016) identified current Cloud-BIM interoperability approaches in three categories of data flow architecture. First, manual file transfer that is currently a common way of exchanging BIM data across applications. Project data can be exported and shared in the form of vendor specific formats or neutral format using IFC standard (Zhang et al., 2014). Second, BIM server technologies in which server-based BIM solutions known as model collaboration systems (Shafiq et al., 2013) have provided a central BIM service with a single-sourced data server accessible for project partners (Zhang et al., 2014; Beetz et al., 2011). These model server technologies utilize information directly from the models and are intended to improve multidisciplinary collaboration (Shafiq et al., 2013). The issue with BIM server technologies is that their implementations are still limited (Shafiq et al., 2013) thus have faced scalability issues (Wu & Issa, 2012). Third, Data Interchange Hub such as Flux project (Flux, 2016) that can automate data flow between certain applications. This type of solution currently has a very limited implementation and supports very few design applications to be connected via the hub. Also, in this approach the inter-connection of applications relies on the hub solution and its capabilities although supported software packages can exchange data directly. The study (Afsari et al., 2016) indicates that existing Cloud-BIM interoperability solutions face and mix different challenges and have not fully utilized the potential of Cloud computing by implementing a standardized network-based data transmission process.

## BIM Data Exchange Standardization

Information sharing is the starting point of collaboration and it requires applications to be able to exchange data regardless of vendors and data formats. To achieve this, in the AEC industry, building data is described in IFC specification to support a neutral data format and to facilitate cross-platform BIM interoperability (Eastman et al., 2010; buildingSMART, 2016). IFC data model which describes building data, provides a means to define building components and processes in a publicly available data schema as an open standard (Eastman et al., 2010) to address cross-platform BIM data exchange. IFC data schema is represented as an

EXPRESS schema specification (ISO, 2015). EXPRESS is an information model specification language based on ISO standard and specified as part of the STEP standard for product model data exchange (Schenck and Wilson, 1994) and uses the STEP physical file structure. IFC data model is alternatively represented in XSD schema specification using the XML document structure (i.e. ifcXML). The specification of ifcXML ensures to handle the same data as represented in EXPRESS specification of IFC data model (buildingSMART, 2016). The ifcXML file structure with ".ifcXML" or ".ifx" or ".xml" extension, is the XML document structure and can be used as a data format in Web-based systems.

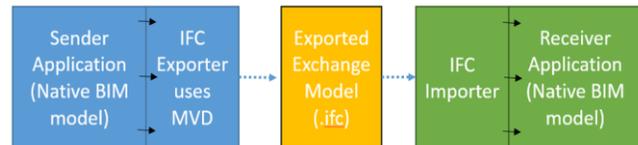


Figure 1: Current process of BIM data exchange using IFC MVD

On the other hand, Model View Definitions (MVDs) proposed by the US National BIM Standard (NBIMS) is supposed to assist BIM data exchange (NBS, 2015) in cross-platform data transfer. An MVD or specifically an IFC view definition, specifies a subset of the IFC schema that is needed to satisfy one or many exchange requirements of the AEC industry (buildingSMART, 2012). MVD consists of one or multiple exchange requirements which must be provided by the sender of data to support work in the receiving application. Sender, as illustrated in Figure 1, uses an MVD to generate the exported IFC model in a BIM application and then passes the file to the receiver of data. Upon receiving the exported IFC model, the receiver of data imports the model in the receiving application which uses an IFC importer module to translate the IFC file to native binding.

## Cloud Interoperability Features

There are several categories of Cloud services such as infrastructure, platform, application, etc. Based on the services that the Cloud solution provides, there are three major types of Cloud computing models as IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service) (Lewis, 2013; Jaeda & Modi, 2012; Rimal et al. 2009; Gong et al., 2010).

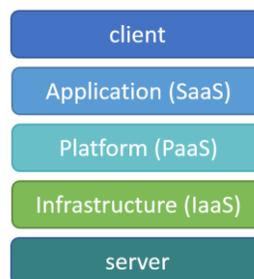


Figure 2: Layers of Cloud Architecture

Figure 2 illustrates the different layers of Cloud architecture: a client includes computer software and hardware for application delivery, a Cloud application delivers SaaS, platform services or PaaS provide a computing platform using the cloud infrastructure provided by IaaS, and a server contains computer software and hardware for the delivery of Cloud services (Jaeda & Modi, 2012).

In Cloud computing the term “interoperability” might sometimes refer to “portability” which is the ability to move a system from one cloud platform to another (Lewis, 2013; Rimal et al. 2009). Interoperability which is discussed in this research, is considered as the ability of data exchange and integration. Here, interoperability deals with what is known as “enabling products/software components to work with or integrate with each other seamlessly, in order to achieve a desired result” (Antonopoulos & Gillam, 2010). The issue is each Cloud provider incorporate Cloud computing with a self-contained set of conventions, data formats, and application programming interfaces (APIs) and in order to allow Cloud services interoperate identification of major Cloud interoperability components is critical (Bernstein et al., 2009). Therefore, this paper highlights four key features of Cloud interoperability as API, Data Format, Data Transfer Protocol, and Standards.

### Cloud APIs

Capabilities of Cloud services are represented through interfaces which can be accessed through APIs. Therefore, if in Cloud applications a common set of APIs with agreed terminologies are used, these applications can interoperate (Loutas et al., 2011). “Cloud APIs specify how software applications interact with a Cloud-based platform where these applications can be deployed” (Petcu et al., 2011). These APIs define how applications can request information from the platforms and how to use their facilities. Cloud APIs can be in the form of Web services e.g. based on Representational State Transfer (REST) or Simple Object Access Protocol (SOAP), application dependent protocols, high level programming languages, or remote calls (e.g. Java RMI, AMF) (Petcu et al., 2011; Endo et al., 2010; Dillon et al., 2010; Lewis, 2013). However, Cloud services that are developed separately by different vendors have different APIs and this makes the interoperability difficult (Dillon et al., 2010). Such differences require human intervention to connect and interact with Cloud services.

### Data Transfer Protocols

Cloud services provide architectural principles and software specifications to connect computers using standardized internet protocols (Dillon et al., 2010). Cloud computing is based on the Transmission Control Protocol/Internet Protocol (TCP/IP). In fact, standard protocols have made Cloud computing reality (Gong et al., 2010).

Common TCP/IP protocols are HTTP (Hyper Text Transfer Protocol) as a protocol designed to allow the transfer of Hypertext Markup Language (HTML) documents, FTP (File Transfer Protocol) for high-speed disk-to-disk file transfers, and Simple Mail Transfer Protocol (SMTP) as an internet mailing system (IBM, 2006). FTP deals with the transmission

of files between computers but HTTP is based on request-response activity. World Wide Web traffic mostly uses the HTTP protocol because HTTP uses the most bandwidth across the Internet (IBM, 2006). As illustrated in Figure 3, HTTP takes care of the communication between a web server and a web browser. It first establishes a secure connection between a server and a browser. Then, a client that is running an application on the web browser can send requests to the connected server and upon request, the server sends data that reside on a repository or a datastore to the client (Gong et al., 2010).

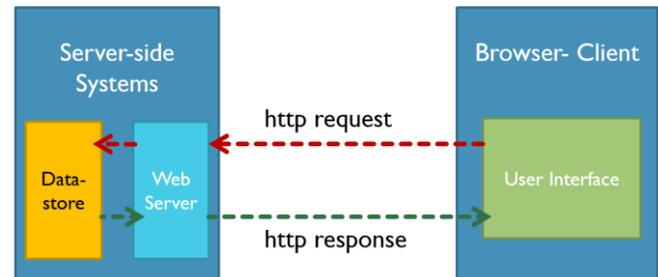


Figure 3: HTTP request/response model

### Data Formats

Access through the Cloud APIs is supported by several data formats. Choosing the proper data serialization format is critical due to the increase in data exchange over the internet. Data format is even more significant in mobile devices because these devices use limited resources and are bandwidth limited (Sumaray, and Makki, 2012). Two most common text-based approaches of data serialization are eXtensible Markup Language (XML) and JavaScript Object Notation (JSON) and have been used widely in Web applications for data transmission (Wang, 2011; Sumaray, and Makki, 2012). XML has a prescriptive grammar and stores all data in the closed tag with the data indexed by labels. JSON on the other hand is a lightweight data exchange format uses a text format independent of the language and has higher parsing efficiency than XML (Wang, 2011). More recently, binary data serialization formats are being used such as Google’s Protocol Buffers (ProtoBuf) and Thrift developed by Facebook (Sumaray, and Makki, 2012) as well as Apache Avro developed as a Hadoop subproject (Maeda, 2012) which are extremely lightweight, and fast to serialize and deserialize (Sumaray, and Makki, 2012; Maeda, 2012). The studies show that the size of binary-based serialized data is better than XML or JSON-based serialization (Maeda, 2012).

### Standards

Standardization is an effective solution to address the interoperability issue of the Cloud services (Dillon et al., 2010; Petcu et al., 2011). There are many Cloud standardization projects such as Open Cloud Computing Interface (OCCI) and Open Cloud Manifesto (Endo et al., 2010; Petco et. al, 2011). Some of these efforts explain standardizing parts of a Cloud computing solution such as data access and others deal with standardizing how parts of

a solution should work together (Lewis, 2013). Although existing standards can support Cloud interoperability, there are different levels of system interoperability that should be carefully considered such as technical interoperability (i.e. deals with exchanging data), semantic interoperability (i.e. deals with exchanging meaningful data), and organizational interoperability (i.e. deals with participating in multi-organizational business processes). Existing standards can address technical interoperability but might not be able to guarantee semantic or organizational interoperability (Lewis, 2013). To address semantic and organizational interoperability, domain knowledge based on established schema and use cases are required. In this regard, the need for a standardized API is the main semantic interoperability requirement since a standardized Cloud API created and supported by the Cloud vendors can resolve compatibility conflicts between Cloud services and can allow one Cloud system cooperate with another Cloud system (Loutas et al., 2011; Dillon et al., 2010; Lewis, 2013). Thus the main step of managing the interactions of Cloud resources and services is establishing a standardized API. In addition, a common Cloud data model should be used as the basis to describe Cloud systems components such as resources, services and APIs (Loutas et al., 2011; Endo et al., 2010).

## Alternative Approach for Cloud-BIM Interoperability

Identification of major components of Cloud interoperability highlight the potential aspects that should be considered in implementing Cloud-based BIM applications to exploit the potential of the Cloud for addressing cross-platform network-based BIM data transmission.

As discussed, current file-based BIM data exchange process as shown in Figure 4 is based on using the MVD in the sender application (i.e. application A) to generate an exchange model which is a file mainly in ".ifc" format. This file is then passed to the receiving application (i.e. application B) which uses a translation module as the IFC importer to interpret and visualize the imported model in a native environment.

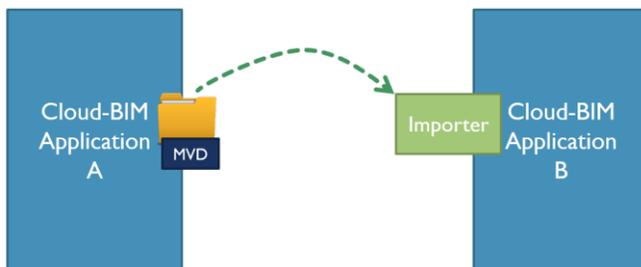


Figure 4: Manual file-based BIM data exchange process

In this process, the request for data happens outside of the BIM model, usually through emails or other methods of correspondence in which the receiver of data (e.g. architectural design team) requests for a specific model (e.g. structural design model) within an MVD from the sender of

the data (e.g. structural design team). In fact, the receiver of data needs to wait until the sender of data exports the model as a file and passes it, meaning that the receiver party or application does not deal with getting the data until the file is exported and is ready to be imported.

The shortcomings of existing methods of cross-platform BIM data transfer are identified in previous study (Afsari et al., 2016). To address these challenges, an alternative approach should utilize four major features of Cloud interoperability identified earlier by implementing a standardized API based on proper profile of data format and protocols. However, Cloud APIs provided by Cloud-BIM vendors have not been standardized yet. BIMSie project (Berlo, 2016) has worked towards introducing a standardized Service Interface (i.e. API) for Cloud-BIM solutions. This standard Service Interface is supposed to automate interaction between Cloud-BIM applications by introducing an API for Cloud-BIM applications with standardized methods (Berlo, 2016). However, in the BIMSie initiative when services are connected, the data exchange is file-based using industry standards as IFC files. This project is an important step towards facilitating Cloud-BIM interoperability by allowing APIs to get connected. But the issue of file-based data transfer has not been addressed yet.

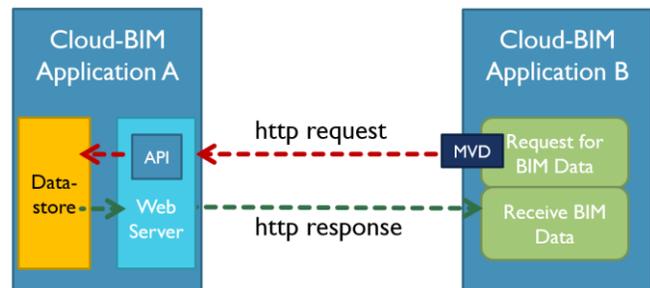


Figure 5: Web-based data exchange between Cloud-BIM applications

Therefore, to take advantage of the opportunities exist in the Cloud interoperability, an alternative architecture should utilize the aforementioned features. An example architecture is shown in Figure 5.

In this architecture, data stored in a datastore within application A can be accessed through an API with a standard set of methods and terminologies that can be interpreted by application B. The API in fact, provides an on-demand access to the pool of data on application A. Accordingly, data request happens only in application B that becomes the client for application A and uses a data exchange requirements (i.e. MVD) to request for data. The data exchange requirements follow the rules defined in the MVD specification. Major features of this data flow are as follows:

- **The API:** This architecture suggests that each application API in Cloud service provider should implement a set of standardized resources within datastore with an appropriate data serialization

format that are retrievable over HTTP. This API needs to follow a common data model based on IFC specification as industry-wide BIM standard to address semantic interoperability.

- **Data Transfer Protocol:** This architecture defines a data flow for exchanging model-based data between Cloud-BIM applications that is different from the conventional methods such as manual file transfer. Since this alternative approach uses HTTP calls for cross-platform communication, it is based on a simple request/response mechanism. As a result, in this architecture the receiver of data directly deals with getting the data instead of waiting for the sender to export data in the forms of files. Also, this architecture will not be tying an application to an integration system like a BIM server technology or an interchange hub while it also provides opportunities for real-time data exchange.
- **Data Format:** In this architecture, the datastore performs as a repository to persist BIM data. The data should be serialized in an efficient compatible format based on common agreements and terminologies. Since IFC is the common standard for open BIM, IFC specification can provide the capabilities to capture domain knowledge and understanding with a common data schema. Currently ifcXML is the only IFC data representation that is compatible with Web technologies. However, limitations of XML-based documents should be taken into account. Web applications are typically represented in the form of Asynchronous JavaScript and XML (AJAX) and Web Services. Unlike Web Services, AJAX applications are aimed at enhancing the user experience while data transmission speed is very important (Wang, 2011). When AJAX was first introduced, XML was used widely but XML showed inadequacies because of its data structure with redundant tags, larger size and the low efficiency of its analysis in serializing and deserializing (Wang, 2011; Maeda, 2012). Therefore, data serialization approaches such as JSON and binary as discussed earlier, should be considered as the alternatives to achieve higher efficiency especially in the applications of AJAX technology.

## Discussions

As discussed, current BIM collaboration and data exchange process are mainly based on manual file transfer while data request by the receiver of data (i.e. Application B illustrated in Figure 4) happens outside of an automated process with no direct control on interacting with required data. In addition, data exchange might be based on BIM integration solutions which are addressing model federation in a centralized platform or interconnecting a limited number of design applications on premise through an interchange hub with the help of plug-ins. Server-based and interchange hub solutions are mainly based on limited data formats and allow limited number of applications to interoperate. Therefore,

collaboration with other applications that are not supported in the cycle of these systems will again rely on exporting/importing BIM models in the form of files. The Web-based data exchange methodology specified in this study (Figure 5) suggests to redefine the information flow while BIM data exchange is powered by Web-based data transfer protocols and formats.

In the conventional BIM data flow, if an architectural designer who is using application B, needs to request a specific information from another discipline such as structural design team that are using application A, the architect should ask the structural designer to export a model containing required data from application A and pass the file to him/her and then the architect imports the file in application B to get access to the required data. By using the methodology specified in this paper, while all project parties are using Cloud-based BIM applications, the architect using application B only needs to connect to the API of the application A that the structural designer is using and then all the architect needs to do is to directly request for BIM data within the BIM application and he/she will immediately receive data represented in Web compatible formats. This way the receiver of the data interacts with the actual BIM model in the datastore of sending application for retrieving the required data. This approach will reshape BIM collaboration process by introducing a new data flow.

Most importantly, there is a major issue in the conventional file export/import process because current process of data exchange as described in Figure 4, implements the MVD in the sender application (i.e. application A) in order to export the model. While there are methods to validate the model (i.e. after being exported) against the initial exchange requirements of the MVD, there is no methodology or tool to validate the model after being imported. In other words, importing the model in the receiving application (i.e. application B) will end up with vague and incomplete data. In the methodology proposed here as Figure 5 illustrates, the MVD implementation will become an effort in the receiver side (i.e. application B) and not in the sender side (i.e. application A). This means that while data is available on the sender side (i.e. application A), data request is managed on the receiver side (i.e. application B) and thus will be directed based on the exchange requirements when the receiver of data requests for it. This approach provides the receiving application and parties with more control over BIM data exchange. In addition, by considering the MVD on the receiver side, data validity focus will be in the receiving application (i.e. application B). This capability can only be achieved through using network-based BIM data transmission.

## Conclusion and Recommendations

This research specified a data flow for Cloud-BIM that utilizes true potentials of the Web technologies and can perform as an enabler for a collaborative process allowing Cloud-BIM services communicate through their standardized APIs. This data flow provides a fundamental basis for a common framework for Cloud-BIM interoperability. It is anticipated that

future study will be towards the implementation and evaluation of the methodology described in this paper.

On the other hand, Cloud-BIM interoperability research and its standardization effort is still in its infancy, and the body of knowledge in the area has not been well defined yet. Therefore, this study highlights the need for standardization of Cloud-BIM APIs using IFC data model as industry established schema. IFC specification can assist to ensure semantic interoperability of building data in the Cloud and future research should investigate how IFC data model can be integrated in the standardization of Cloud-BIM APIs. If Cloud-BIM services and their APIs are designed and implemented with no standardization in mind they are subject to data interoperability problems similar to current challenges with desktop-based BIM applications. In addition, compatible and efficient BIM data representation in the Cloud is another critical aspect in Cloud-BIM interoperability. Future study should investigate how different data serialization methods used in the data flow specified in this paper can affect the efficiency of network-based BIM data exchange in the Cloud. Also, the study of encoding IFC specification in alternative data serialization techniques suitable for Web-based data interchange is a future step.

## References

- Afsari K., Eastman C. M. and Dennis R. S. (2016). Cloud-based BIM Data Transmission: Current Status and Challenges. In Proceedings of the 33rd International Symposium on Automation and Robotics in Construction (ISARC). Auburn, Alabama.
- Antonopoulos, N., & Gillam, L. (2010). Cloud Computing: Principles, Systems and Applications. Springer.
- Bernstein D, Ludvigson E, Sankar K, Diamond S, Morrow M. (2009) Blueprint for the intercloud – protocols and formats for cloud computing interoperability. In Proceedings of the 4th International Conference on Internet and Web Applications and Services (ICIW 2009). IEEE: Venice/Mestre, Italy, 2009; 328–336.
- Beetz, J., Berlo, L., Laat, R., & Bonsma, P. (2011). Advances in the development and application of an open source model server for building information. CIB W078 – Information Technology for Construction. Sophia Antipolis, France.
- Berlo L. (2016). BIM Service interface exchange (BIMSie). Retrieved May 2016 from [https://www.nibs.org/?page=bsa\\_bimsie](https://www.nibs.org/?page=bsa_bimsie)
- buildingSMART (2013) Industry Foundation Classes Release 4 (IFC4), retrieved July 2016 from: <http://www.buildingsmart-tech.org/ifc/IFC4/final/html/>
- buildingSMART. (2012). An Integrated Process for Delivering IFC Based Data Exchange. buildingSMART International User Group.
- buildingSMART. (2016). IFC Overview summary. Retrieved from buildingSMART: <http://www.buildingsmart-tech.org/specifications/ifc-overview>
- Curry, E., O'Donnell, J., Corry, E., Hasan, S., Keane, M., & O'Riain, S. (2013). Linking building data in the cloud: Integrating cross-domain building data using linked data. Advanced Engineering Informatics. 206–219.
- Dillon, T., Wu, C., and Chang, E. (2010) Cloud Computing: Issues and Challenges. IEEE International Conference on Advanced Information Networking and Applications. 27–33.
- Eastman, C., Jeong, Y., Sacks, R., and Kaner, I. (2010). Exchange Model and Exchange Object Concepts for Implementation of National BIM Standards. Journal of Computing in Civil Engineering. 25-34.
- Endo, P. T., Goncalves G. E., Kelner J., and Sadok D. (2010). A Survey on Open-source Cloud Computing Solutions. VIII Workshop em Clouds, Grids e Aplicacoes. 3-16.
- Flux (2016). Flux Overview, Retrieved July 2016 from: <https://community.flux.io/content/kbentry/1258/flux-overview.html>.
- Gong, C., Liu, J., Zhang, Q., Chen, H., & Gong, Z. (2010). The Characteristics of Cloud Computing. 39th International Conference on Parallel Processing Workshops. San Diego, CA: IEEE. 275 – 279.
- IBM. (2006). TCP/IP Tutorial and Technical Overview. International Technical Support Organization. Retrieved July 2016 from <https://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf>
- ISO (2015), ISO 16739:2013- Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries.
- Jadeja Y. & Modi K. (2012). Cloud Computing - Concepts, Architecture and Challenges, in the proceedings of International Conference on Computing Electronics and Electrical Technologies. Nagercoil, India. 877-880.
- Juan, D., & Zheng, Q. (2014). Cloud and Open BIM-Based Building Information Interoperability Research. Journal of Service Science and Management. Vol.7. 47-56.
- Loutas, N., Kamateri, E., and Konstantinos, T. (2011) Cloud computing interoperability: the state of play, in: Proceedings, CloudCom 2011, IEEE Computer Society. 752-757.
- Lewis, G. A. (2013). The Role of Standards in Cloud Computing Interoperability. Software Engineering Institute, Carnegie Mellon.
- Maeda, K. (2012). Performance evaluation of object serialization libraries in XML, JSON and binary formats. 2nd Conference on Digital Information and Communication Technology and its Applications (DICTAP). 177-182.
- NBS. (2015). National BIM Standard- United States. Version 3. National Institute of Building Sciences. buildingSMART alliance.
- NIST. (2014). US Government Cloud Computing Technology Roadmap, Volume I and II, Special Publication 500-293. National Institute of Standards and Technology.
- Petcu D., Craciun C., Rak M. (2011). Towards a cross platform Cloud API—components for Cloud federation, in: Proceedings, CLOSER 2011, SciTe Press. 166–169.
- Rimal BP, Choi E, Lumb I (2009). A taxonomy and survey of cloud computing systems. In: The fifth international joint conference on INC, IMS and IDC. 44–51.
- Shafiq MT, Matthews J, Lockley SR. (2013). A Study of BIM Collaboration Requirements and Available Features in Existing Model Collaboration Systems. Journal of Information Technology in Construction. Vol. 18. 148-16.
- Schenck D. and Wilson P. (1994), Information Modeling: The EXPRESS Way, Oxford University Press.
- Sumaray, A., Makki, S. (2012). A comparison of data serialization formats for optimal efficiency on a mobile platform. In: Proc. 6th International Conf. on Ubiquitous Information Management and Communication. Kuala Lumpur. Malaysia. 48:1–48:6.
- Wang, G. (2011). Improving Data Transmission in Web Applications via the Translation between XML and JSON. Third International Conference on Communications and Mobile Computing. Qingdao: IEEE. 182 – 185.
- Wong, J., Wang, X., Li, H., Chan, G., & Li, H. (2014). A Review of Cloud-Based BIM Technology in the Construction Sector. Journal of Information Technology in Construction. Vol. 19, 282.

- Wu, W., & Issa, R. (2012). Leveraging Cloud-BIM for LEED Automation. *Journal of Information Technology in Construction*. Vol. 17. 367-384.
- Yang, J., Anand, R., Hobson, S., Lee, J., Wang, Y., & Xu, J. (2011). Data Service Portal for Application Integration in Cloud Computing. *Proceedings of 8th International Conference & Expo on Emerging Technologies for a Smarter World (CEWIT)*. New York, NY: IEEE. 1-3.
- Zhang, J., Liu, Q., Yu, F., Hu, Z., and Zhao, W. (2014). A Framework of Cloud-computing-based BIM Service for Building Lifecycle. *Computing in Civil and Building Engineering*. 1514-1521.