

Analyse und Integration von BIM Technologien für smino

Studienarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Herbstsemester 2018

Autor(en):	Patrick Faic, Benjamin Moosmann
Betreuer:	Prof. Stefan Keller
Projektpartner:	smino Rapperswil
Experte:	-
Gegenleser:	-

Abstract

Die digitale Transformation des Baumarktes ist in vollem Gange. Dabei ist ein Akronym in aller Munde: BIM - Building Information Modelling. Mit Hilfe von BIM wollen die beteiligten Akteure eine Effizienz- und Qualitätssteigerung erzielen und die interdisziplinäre Zusammenarbeit an Bauprojekten vereinfachen. Auch das Startup BBC Systems AG beteiligt sich an dieser Transformation und möchte ihre Online-Plattform smino mit BIM-bezogenen Funktionen erweitern.

In einem ersten Schritt wurde in dieser Arbeit einen Überblick von BIM verschafft. Über relevante Standards und über Workflows in der Baubranche. Quellen für diese Arbeit waren Interviews mit Schweizer Baufirmen und mit Personen, die BIM bereits einsetzen, sowie Internetrecherchen über Firmen, die Produkte für den BIM-Markt anbieten.

Als Datenaustauschstandard für 3D-Modelle wurden die „Industry Foundation Classes“ (IFC) identifiziert und für die Kollaboration das „BIM Collaboration Format“ (BCF). Es wurden Cloud Services, Software-Frameworks, Closed- und Open-Source Software evaluiert. Ein wichtiges Kriterium für smino war dabei die Integration eines performanten 3D-Model-Viewers in ihre gleichnamige Webapplikation. Die Wahl fiel dabei auf Forge, den Cloud Service und das Software-Framework der Firma Autodesk.

Mit dem Wissen aus der Recherche wurde ein Softwareprototyp mit den JavaScript-Technologien React und Node.js entwickelt. Der Prototyp simuliert das Zusammenspiel zwischen IFC und BCF. Implementiert wurden das Hochladen und Betrachten von Modellen im Forge-Viewer. Zu jedem Modell können Tickets erfasst werden, welche unter anderem Kameraposition und selektierte Objekte abspeichert. Beim Laden eines Tickets wird der Viewer wieder in den gespeicherten Zustand versetzt. Der Prototyp dient smino als Vorlage für die Erweiterung ihres Produktes.



Management Summary

Analyse und Integration von BIM Technologien für smino

Diplomanden	Patrick Faic, Benjamin Moosmann
Examinator	Prof. Stefan Keller
Experte	-
Themengebiet	Software
Industriepartner	smino

Ausgangslage

Die digitale Transformation des Bauplatzes ist in vollem Gange. Dabei ist ein Akronym in aller Munde: BIM - Building Information Modelling. Mit Hilfe von BIM wollen die beteiligten Akteure eine Effizienz- und Qualitätssteigerung erzielen und die interdisziplinäre Zusammenarbeit an Bauprojekten vereinfachen. Zurzeit befinden sich die meisten Schweizer Bauunternehmen am Anfang des BIM-Stufenplans. Erst wenige Firmen fordern den Status Quo heraus und arbeiten aktiv mit BIM auf den Stufen 1 und 2. Auch das Startup BBC Systems AG beteiligt sich an dieser Transformation und möchte ihre Online-Plattform smino mit BIM-bezogenen Funktionen erweitern.

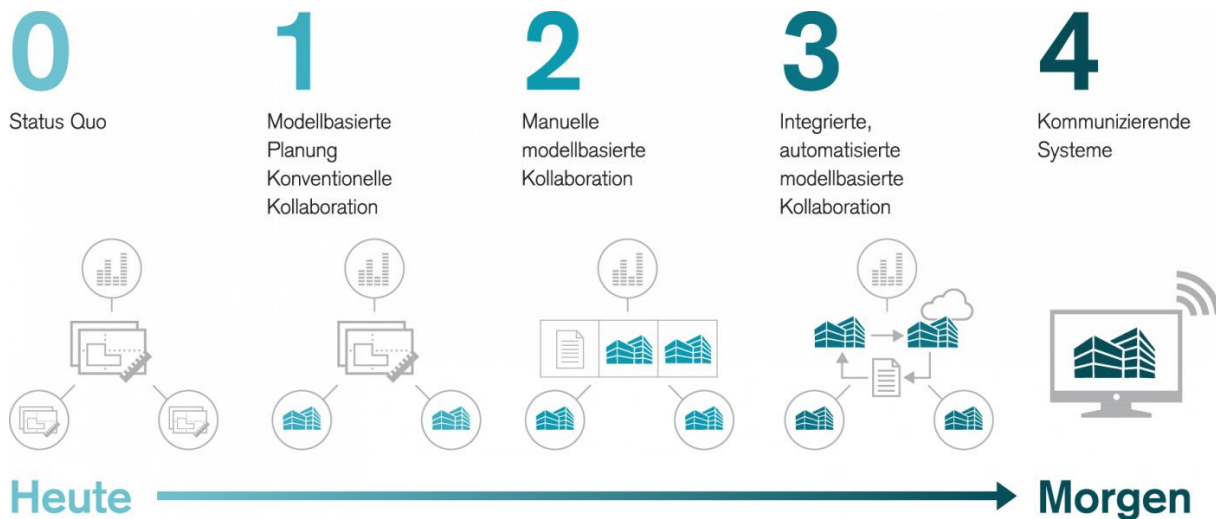


Abbildung 1: BIM-Stufenplan von Bauen Digital Schweiz; diese Arbeit befasst sich mit Stufe 1 und 2.

Vorgehen

Um einen Überblick über BIM zu verschaffen, wurden in einem ersten Teil Standards und Workflows in der Baubranche analysiert. Quellen waren Interviews mit Schweizer Bauunternehmen und mit Personen, die BIM bereits einsetzen, sowie Internetrecherchen über Firmen, die Produkte für den BIM-Markt anbieten.

Als Datenaustauschstandard für 3D-Modelle wurden die „Industry Foundation Classes“ (IFC) und für die Kollaboration das „BIM Collaboration Format“ (BCF) identifiziert.

Der BIM Workflow zeigt auf, wie Firmen mit BIM arbeiten. Für die Punkte 4 und 5 kommt ein BIM-Koordinator zum Einsatz. Seine Aufgabe besteht darin den BIM Workflow zu unterstützen in dem Modelle in den Model Checker importiert und daraus Aufgaben für die entsprechenden Planer erstellt werden.

1. Die Planer der verschiedenen Branchen erstellen ihre 3D-Modelle auf ihren lokalen Maschinen.
2. Aus den einzelnen Modellen werden IFCs auf einen File Hoster (wird von der Bauleitung bestimmt) geladen.
3. Die Modelle aller Branchen werden bei einem File Hoster gesammelt.
4. 1-N IFCs werden in einen Model-Checker geladen, wo die Modelle auf Grund eines Regel-Sets auf Clashes überprüft werden.
5. Aus den Clashes werden BCF-Issues generiert und auf eine Kollaborationsplattform geladen.
6. Der Planer, an den das Issue adressiert ist, lädt es sich von der Kollaborationsplattform herunter und ...
 1. ... bearbeitet sein Modell auf Grund der Beschreibung im BCF.
 2. Das bearbeitete Modell wird wieder auf die IFC-Plattform geladen.
7. Das Issue wird als «gelöst» auf die Kollaborationsplattform zurück geladen und der Workflow wird bei 3 fortgesetzt bis (fast) alle Clashes in den Modellen behoben wurde.

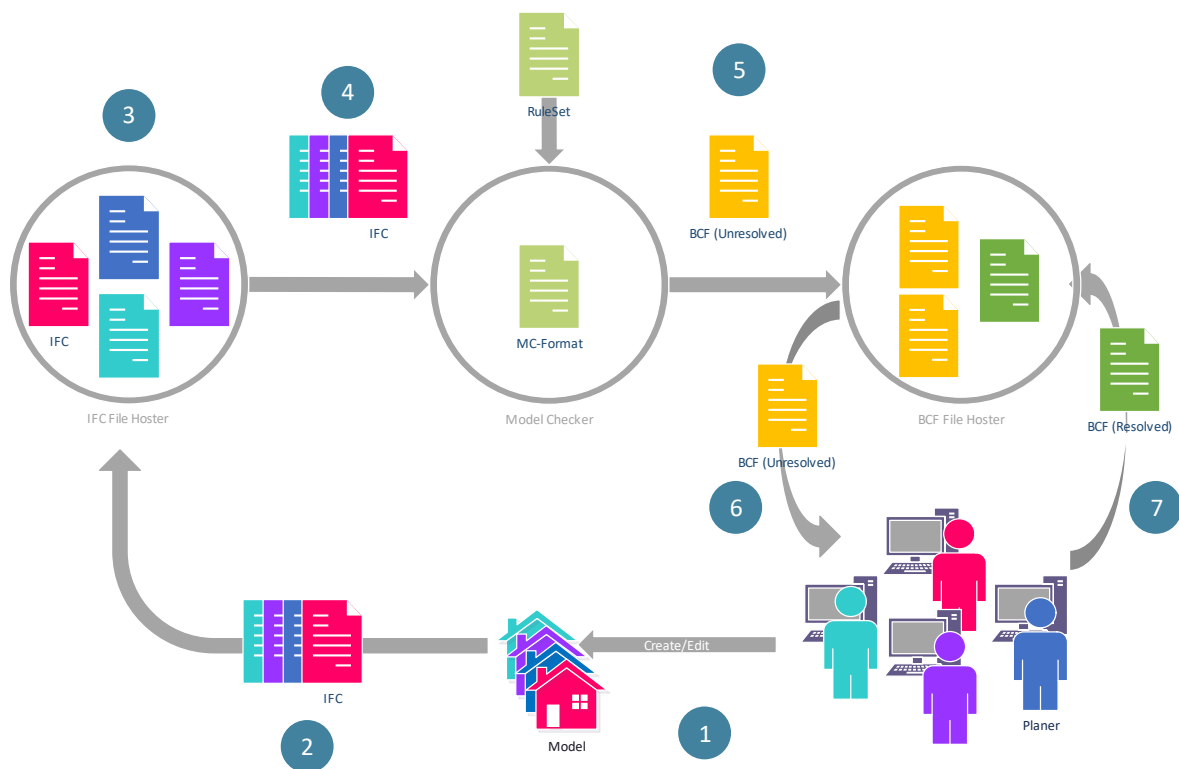


Abbildung 2: BIM-Workflow mit IFC- und BCF-Dateien und Plattformen (Schritte 1-7)

smino möchte diesen Prozess vereinfachen, indem Sie Daten- und Kommunikationsplattform vereint. Einzig die Modellüberprüfung, wird vorläufig auf einer externen Lösung durchgeführt. (Siehe Abbildung 3)

Dafür wurden Cloud Services, Frameworks, Closed- und Open-Source Software evaluiert. Ein wichtiges Kriterium für smino bildet die Integration eines performanten 3D-Model-Viewers für ihre Webapplikation. Die Wahl fiel dabei auf Forge, den Cloud Service und das Framework der Firma Autodesk.

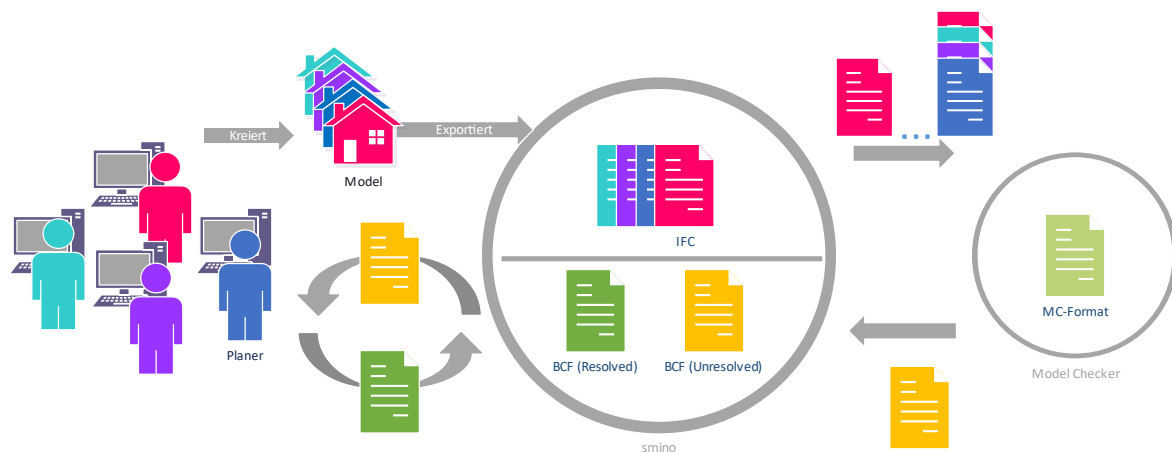


Abbildung 3: Darstellung des möglichen smino-Workflows

Ergebnisse




Mit den Erkenntnissen aus dem ersten Teil der Arbeit wurde in einem zweiten Teil ein Softwareprototyp mit den JavaScript-Technologien React und Node.js entwickelt.

Der Prototyp integriert den 3D-Model-Viewer und simuliert das Zusammenspiel zwischen IFC und BCF. Implementiert wurden das Hochladen und Betrachten von Modellen im Forge-Viewer. Zu jedem Modell können Tickets erfasst werden, wobei beispielsweise die Kameraposition und selektierte Objekte mitverwaltet werden. Beim Laden eines Tickets wird der Viewer wieder in dessen ursprünglichen Zustand gesetzt. So können in der Realität Fehler am Modell einfach beschrieben und wieder dargestellt werden. Der Prototyp dient smino als Vorlage für die Erweiterung ihres Produktes.

BIM - smino

My Models

Upload Model: Keine Datei ausgewählt.

Thumbnail	Name	Size	Status
	klein.ifc	3.76mb	Show
	gross.ifc	67.07mb	Show
	mittel.ifc	13.49mb	Show

My Tickets

Thumbnail	Title	Date	Show
	Wände falsch	17.12.18 15:06:42	Show
	Schnitt Gebäude	17.12.18 15:09:28	Show

Forge Viewer



Abbildung 4: Webapplikations-Prototyp: Screenshot mit Modellen und Tickets (links) sowie mit 3D-Modell im Forge-Viewer (rechts)

Ausblick

Mit dem Prototyp als Hilfestellung wird smino nun ihre bestehende Lösung erweitern. Die Verfasser dieser Arbeit werden auch über die Projektdauer hinaus für Fragen zur Verfügung stehen.

Wenn sich Open-Source-Technologien weiterentwickelt haben, kann es für smino interessant sein ihre Lösung auf dem aufzubauen.

Inhalt

I.	Projektübersicht	9
1.	Aufgabenstellung	10
2.	Aufbau der Arbeit	13
3.	Projektorganisation	14
4.	Projektbudget	15
5.	Phasen / Meilensteine	16
6.	Infrastruktur	17
7.	Projektmanagement	18
II.	Technischer Bericht	26
8.	BIM – Building Information Modeling	27
9.	BIM Standards	32
10.	Interviews	41
11.	Chancen für smino	42
12.	BIM-Plattformevaluation	46
13.	Schlussfolgerung	59
14.	Ausblick	60
III.	Software-Prototyp	61
15.	Zweck der Applikation	62
16.	Anforderungsspezifikation	63
17.	Analyse	65
18.	Architektur	73
19.	Implementation	79
IV.	Anhang	81
20.	Glossar	82
21.	Verzeichnisse	83
22.	Persönliche Berichte	85
23.	Danksagung	87
24.	Aufgabenstellung	88

I. Projektübersicht

1. Aufgabenstellung

Der folgende Abschnitt ist aus der Aufgabenstellung abgeleitet. Die offizielle Aufgabenstellung befindet sich im Anhang.

1.1. Vision

In der Baubranche hat die Digitalisierung noch viel Entwicklungspotential. Um diese voranzutreiben, ist bei Baufirmen ein Akronym in aller Munde: BIM (Building Information Modelling). Mit der Hilfe von BIM erhoffen sich die Betriebe eine Effizienzsteigerung zu erzielen und die Zusammenarbeit an Projekten erheblich zu vereinfachen.

Die Entwicklung des vielversprechenden Marktes für BIM basierte Lösungen ist in vollem Gange. Deswegen möchte das Startup BBC Systems AG evaluieren, mit was für einem Produkt sie sich am besten im Markt positionieren können. Aktuell betreiben und entwickeln sie die Webapplikation smino. Die Plattform erleichtert die Kommunikation bei der Realisierungsphase von Bauvorhaben. Es ist wichtig, dass smino eine attraktive Lösung für die Planungsphase von Gebäuden anbietet. Dies gelingt am besten durch die Integration von BIM-Standards.

1.2. Ziele

Ziel dieser Arbeit ist es, BIM und dessen Workflows zu analysieren, die dafür verwendeten Tools testen und einen Mehrwert für smino zu kreieren. Anhand eines lauffähigen Prototyps wird dabei das Konzept von BIM aufgezeigt.

Um eine geeignete Lösung finden zu können, will diese Arbeit smino mit folgenden Zielen unterstützen:

Teilziel 1: Recherche:

Ziel	Erledigt	Kapitel
Was ist BIM genau? Wie sehen die Standards aus? Was ist die Technologie dahinter?	✓	8
Wie sieht die Arbeitsmethodik mit BIM aus? Wie lässt sich diese auf smino übertragen?	✓	8
Welche Komponenten/Libraries gibt es, um BIM Modelle darzustellen, manipulieren, auszulesen die smino verwenden kann?	✓	12
Könnte smino als BIM Hub funktionieren? Das Modell würde hier in der Cloud gehostet. Macht das für smino Sinn? Gibt es Verlierer dabei? Wer könnten allenfalls Partner sein?	✓	11
Wie sieht der Aufbau eines BCFs aus und wie kann dieses auf ein smino-Issue abgebildet werden?	✓	9.2

Teilziel 2: Feasibility-Prototyp und Evaluation

Framework-Evaluation:

Ziel	Erledigt	Kapitel
Implementierbarer 3D-Viewer finden und dessen Performance für PC, Tablet und Mobile testen.	✓	12.7
Preis/Leistung des Frameworks evaluieren.	✓	12
Entscheidungshilfe für smino präsentieren.	✓	12

Feasibility-Prototyp Funktionalitäten:

Ziel	Erledigt
Import eines IFCs und Darstellung in einer Liste.	✓
IFCs können versioniert und verwaltet werden. (optional)	✗
IFCs können ausgewählt und in einem 3D-Viewer dargestellt werden.	✓
Es können Kamera-Positionen smino-extern über API angesteuert werden.	✓
Es können Objekte smino-extern über API selektiert und fokussiert werden.	✓
Es können Kamera-Positionen ausgelesen werden (inkl. Thumbnail).	✓
Es können selektierte Objekte als Referenzen ausgelesen werden (inkl. Thumbnail).	✓

Zusätzliche Informationen an Koordinaten/Objekte anzeigen (z.B. Marker oder Text). Als Erkenntnis aus dem Prototyp geht ein Vorschlag hervor zur Integration dieser BIM-Komponente in die smino Webapplikation.

1.3. Lieferobjekte

1. Dokumentation, inkl. Textabstract (deutsch), Management Summary (deutsch), technischer Bericht und Software Engineering-Projekt (deutsch); Anhänge (Literaturverzeichnis, Inhalt).
2. Software (englisch).
3. Die vom Studiengang geforderten bzw. empfohlenen Lieferobjekte: Poster (nur digital), Broschüren-Abstract, kein Kurzvideo.

1.4. Vorgaben/Rahmenbedingungen

- Infrastruktur: Windows Azure, .Net, JavaScript, React, BIM-Tools.
- Wo sinnvoll moderne SW-Entwicklung (u.a. mit Unit Testing, Repositories/Versioning, kontinuierliche Integration, Docker).
- Nichtfunktionale Anforderungen: keine besonderen.
- Die Projektdauer ist auf ein Semester beschränkt und endet am Freitag 21.12.2018.

Abgrenzung: Die Arbeit untersucht CAD-Modelle und hat nicht im Fokus, andere BIM-Aspekte zu untersuchen wie beispielsweise den Austausch von Devis, Arbeitsplanungen und auch keine Integration von Geodaten (Terrain, Energie, Ver- und Entsorgung).

2. Aufbau der Arbeit

Die Studienarbeit setzt ihren Schwerpunkt auf die Analyse von BIM und deren Technologien sowie auf die Machbarkeit einer Lösung für den externen Partner smino.

Die Arbeit wird in folgende drei Teilbereiche unterteilt:

Technischer Bericht

Der Technische Bericht verschafft einen Überblick von BIM und wie es in der Baubranche verwendet wird. Es werden Prozessschritte und die dazu verwendeten Tools analysiert. Des Weiteren werden die wichtigsten BIM Standards analysiert. Die Resultate der Analysen dienen smino als Entscheidungsgrundlage für die Implementation der Standards und der dazu verwendeten Tools.

Software-Prototyp

Aus den Erkenntnissen des technischen Berichts und den Entscheidungen von smino wird ein Prototyp erstellt. Damit soll bewiesen werden, dass es machbar ist aus dem ausgewählten Framework einen lauffähigen Prototyp zu erstellen.

Der Prototyp ermöglicht das Darstellen eines 3D-Gebäudemodells in einer Webapplikation. Aus dem hochgeladenen Modell können nun einzelne Objekte ausgewählt und zusammen mit der Kameraposition als sog. Tickets exportiert werden.

3. Projektorganisation

Das Projekt wird in einer flachen Organisationsstruktur durchgeführt. Alle Mitglieder sind gleichberechtigt und an allen wichtigen Entscheidungen beteiligt.

Mit dem Auftraggeber smino findet ein reger Austausch statt. Zusammen mit dem Projektbetreuer legen sie den Umfang des Prototyps fest.

3.1. Organisationsstruktur

Mitglied	Aufgabe
Patrick Faic	Engineering
Benjamin Moosmann	Engineering

3.2. Externe Schnittstellen

Projektbetreuer	Prof. Stefan Keller	stefan.keller@hsr.ch
CEO smino	Silvio Büsser	silvio.buesser@bbc-systems.ch
CTO smino	Mischa Trecco	mischa.trecco@bbc-systems.ch

4. Projektbudget

4.1. Kostenvoranschlag

Das Modul Engineering Projekt gibt 8 ECTS Punkte. Dies entspricht einem Zeitaufwand von 240 Stunden pro Person. Auf die Dauer des Projekts fallen dementsprechend ca. 17 Arbeitsstunden pro Student pro Woche an. Wir wollen den Stundenaufwand kontinuierlich halten.

Der Zeitaufwand wird mit Hilfe von Clockify erfasst.

Projektstart	19.09.2018
Projektende	21.12.2018
Projektdauer	14 Wochen
Anzahl Studenten	2
Arbeitsstunden pro Woche pro Student	17.1
Totale Arbeitsstunden	480

5. Phasen / Meilensteine

Wir halten jede Woche ein Meeting mit unserem Projektbetreuer und dem CEO und CTO von smino.

Phase	Besprechung	Meilenstein	Beschreibung
Inception	W1	- 9.09.2018	0 Kickoff
	W2		1 Einarbeit BIM
	W3		Workflows verstanden
			Projektplan erstellen
		- 28.09.2018	
Inception	W4		2 Interviews
			Business Case
		- 12.10.2018	
Elaboration			3 Standards analysiert
			Vorhandene Tools analysieren
			Aufgabenstellung für Prototyp
	W5		
	W6		
	W7		
	W8		
	W9		
	W10		
		- 16.11.2018	
Construction	W11		4 Architektur
			Deployment
	W12	- 7.12.2018	5 Modelle im Viewer darstellen
Construction	W13		6 Tickets aus Modell erstellen
		- 14.12.2018	
Transition	W14	- 21.12.2018	7 Abgabe




Abbildung 5: Meilensteine des Projekts

6. Infrastruktur

Server	Azure	
Geräte	Laptop	Persönliche Laptops
Dokumentation	OneDrive	https://1drv.ms/f/s!AjhpTKSaHefgmygS9Ke2TFRPcBmw
Code	GitHub	https://github.com/JonBonSnovi/SA_HS18_BIMsmino
Deployment	Azure	https://bimsmino.azurewebsites.net/
Storyboard	Trello	https://trello.com/b/n2QZGali/sa-hs18
Zeiterfassung	Clockify	https://clockify.me/
Diagramme	Visio	
UML Diagramme	StarUML	http://staruml.io/
	plantuml	http://plantuml.com/starting

7. Projektmanagement

7.1. Vorgehen

Es wird nach Scrum vorgegangen. Sprints dauern eine Woche. Tasks werden mit Hilfe von Trello erstellt. Für die Zeiterfassung wurde Clockify an Trello Tasks angebunden.

Am Ende jeder Arbeitswoche wird ein Meeting mit dem externen Partner und unserem Betreuer gehalten.

Wird an einem Task gearbeitet wird dafür ein neuer Branch erstellt. Ist die Implementierung fertig und kompiliert, wird ein Pull-Request erstellt. Dieser wird von dem anderen Teammitglied begutachtet bevor ein Merge in den Master-Branch stattfindet. Nach dem Merge auf den Master-Branche wird automatisch auf Azure deployed.

7.2. Storyboard

Das Storyboard wurde in Trello erstellt. Am Ende jeder Woche wird ein Screenshot des Boards erstellt und in der Dokumentation festgehalten.

Definition of Done

- Code Style Guidelines eingehalten.
- Code kompiliert (0 Warnings, 0 Errors)
- Code review/Pair-Programming gemacht
- Source Code committed
- Build fehlerfrei
- Dokumentation erstellt/erweitert

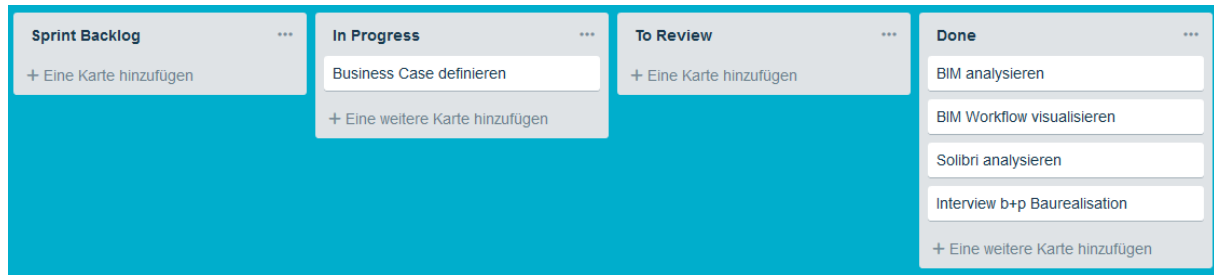
Woche 1

Sprint Backlog	In Progress	To Review	Done
+ Eine Karte hinzufügen	Analyse BIM + Eine weitere Karte hinzufügen	+ Eine Karte hinzufügen	Kickoff Meeting Aufsetzen Projektmanagement Tools + Eine weitere Karte hinzufügen

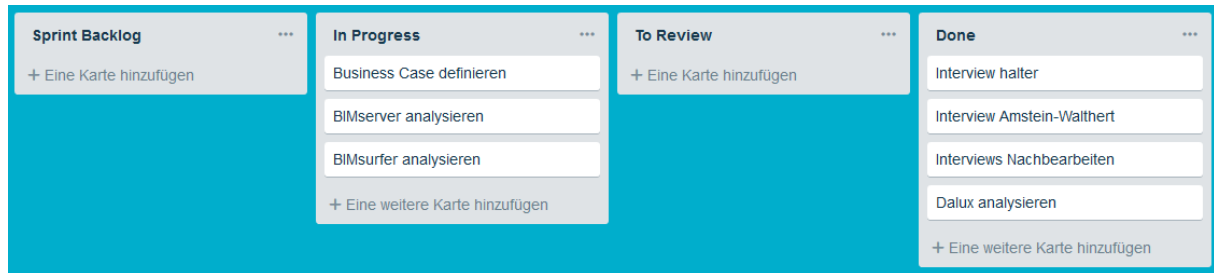
Woche 2

Sprint Backlog	In Progress	To Review	Done
+ Eine Karte hinzufügen	BIM analysieren + Eine weitere Karte hinzufügen	+ Eine Karte hinzufügen	smino analysieren Interviewpartner kontaktieren Interview mit Christian Graf + Eine weitere Karte hinzufügen

Woche 3



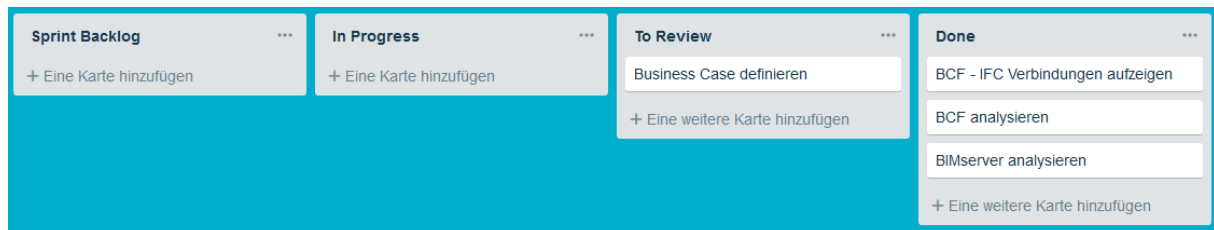
Woche 4



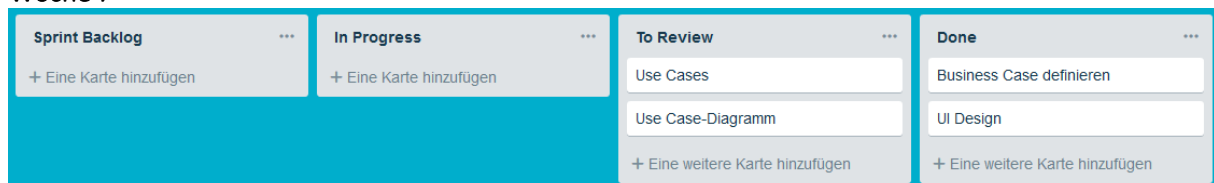
Woche 5



Woche 6



Woche 7



Woche 8

Sprint Backlog	In Progress	To Review	Done
+ Eine Karte hinzufügen	+ Eine Karte hinzufügen	BCF-Forum Plugin auf BIMserver installieren IFC File auf Forge deployen Analyse Forge API Analyse Forge Viewer + Eine weitere Karte hinzufügen	Azure Service bestellen Azure VM aufsetzen BIMserver installieren BCF Datenstruktur beschreiben BCF - sminoissue mapping IFC auf BIMserver laden IFC im Viewer darstellen + Eine weitere Karte hinzufügen

Woche 9

Sprint Backlog	In Progress	To Review	Done
+ Eine Karte hinzufügen	+ Eine Karte hinzufügen	+ Eine Karte hinzufügen	BCF-Forum Plugin auf BIMserver installieren Analyse Forge Viewer Analyse Forge API IFC File auf Forge deployen + Eine weitere Karte hinzufügen

Woche 10

Sprint Backlog	In Progress	To Review	Done
+ Eine Karte hinzufügen	+ Eine Karte hinzufügen	+ Eine Karte hinzufügen	Analysieren bimsync Vergleich BIMserver, Autodesk Forge und bimsync aufstellen BIMserver analysieren BIMsurfer, Autodesk Viewer, bimsync Viewer vergleichen. BIMsurfer analysieren Aufgabenstellung + Eine weitere Karte hinzufügen

Woche 11

Sprint Backlog	In Progress	To Review	Done
+ Eine Karte hinzufügen	+ Eine Karte hinzufügen	+ Eine Karte hinzufügen	Webkonferenz mit bimsync Analyse Hxgn Analyse Allplan Analyse Trimble Connect Viewer Analyse xBIM + Eine weitere Karte hinzufügen

Woche 12

Sprint Backlog	In Progress	To Review	Done
+ Eine Karte hinzufügen	Deployment auf Azure + Eine weitere Karte hinzufügen	+ Eine Karte hinzufügen	US 010: Projekt erstellen US 001: IFC hochladen US 003: Modelle anzeigen + Eine weitere Karte hinzufügen

Woche 13

Sprint Backlog	In Progress	To Review	Done
+ Eine Karte hinzufügen	+ Eine Karte hinzufügen	+ Eine Karte hinzufügen	Prototype: Upload and translate model in forge Prototype: Load models form all buckets or create new bucket if there is none Prototype: Show upload state in viewer Prototype: Show thumbnail of model in viewer Prototype: Get camera position and selected objects (ViewerState) from viewer Prototype: Set camera position and selected objects (ViewerState) to viewer Prototype: Save/Load (ViewerState) as ticket in local db on node and show in GUI Prototype: Create screenshot for tickets and use as thumbnail Deployment auf Azure + Eine weitere Karte hinzufügen

Woche 14

Sprint Backlog	In Progress	To Review	Done
+ Eine Karte hinzufügen	+ Eine Karte hinzufügen	Abschluss Dokumentation + Eine weitere Karte hinzufügen	Dokumente unterzeichnen Persönliche Berichte Verzeichnisse Management Summary Zeiterfassung einbinden Doku: Viewer Analyse Diagramme Plakat Doku: Use Cases Abstract auf HSR Doku: BIM Kongress + Eine weitere Karte hinzufügen

7.3. Zeitplan

Die Zeiterfassung wurde mit der Hilfe von Clockify durchgeführt. Daraus wurden die Daten exportiert und für jedes Teammitglied nach Tasks, welche in Clockify definiert wurden, ausgewertet.

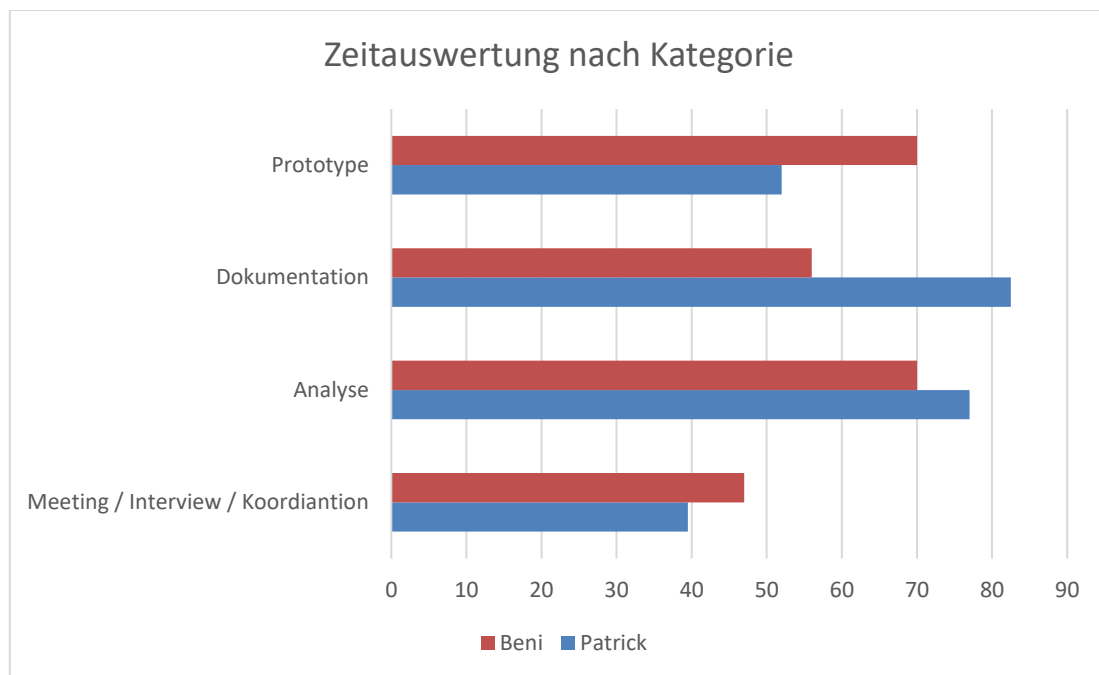


Abbildung 6: Zeitauswertung der Arbeitsaufteilung

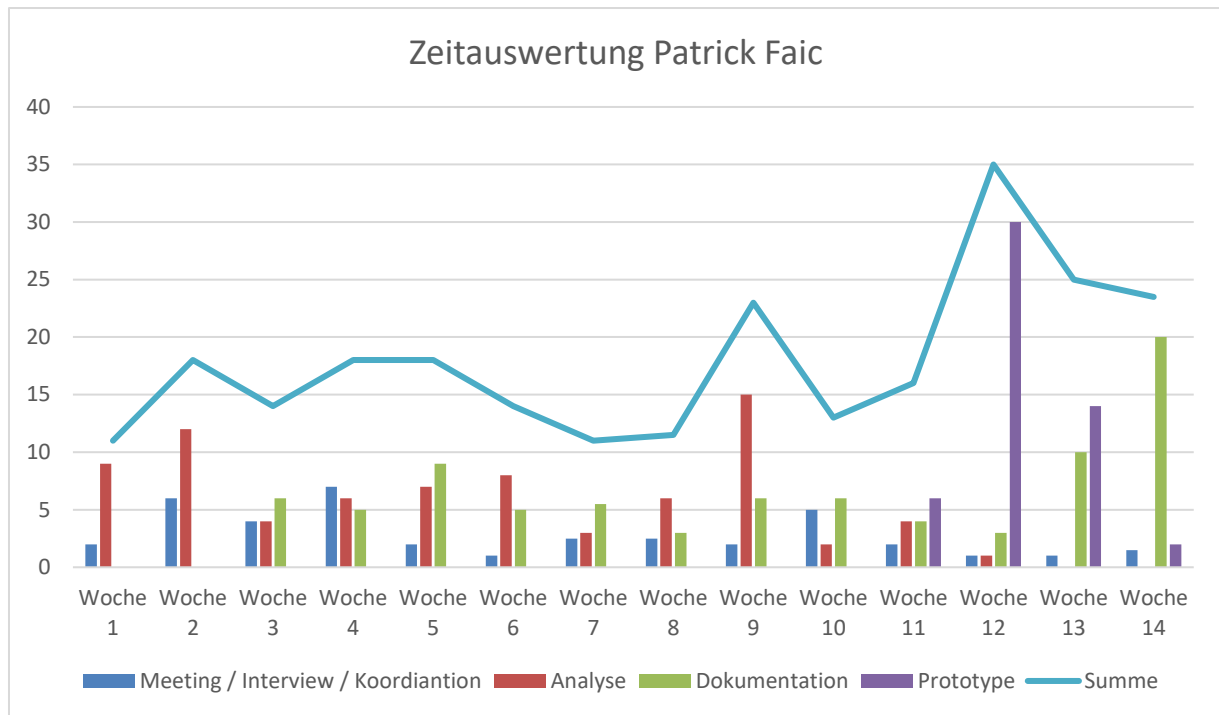


Abbildung 7: Zeitauswertung von Patrick Faic

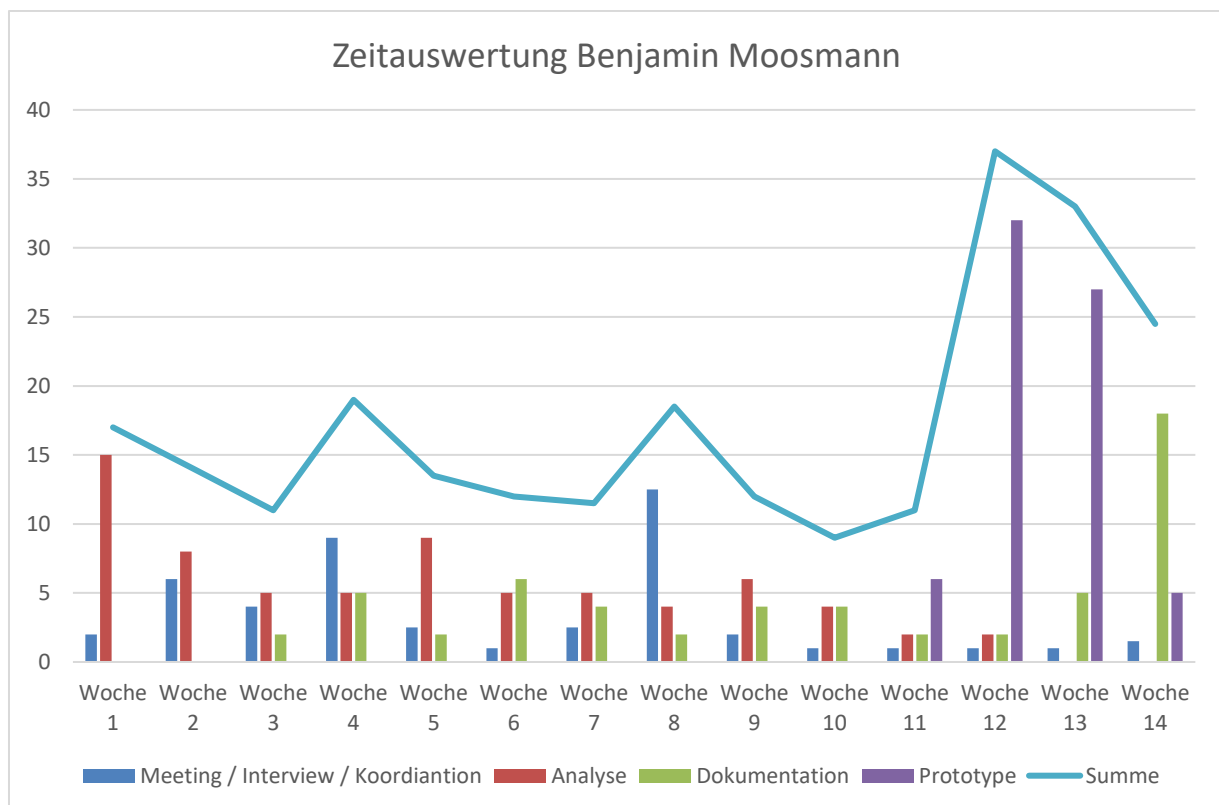


Abbildung 8: Zeitauswertung von Benjamin Moosmann

7.4. Risiken

ID	Risiko	Schaden [h]	Eintrittswahr.	Gewichtung
1	Unklare Anforderungen	10h	30%	8
2	Mangelhafte Schnittstellen zu Umsystemen (Libraries, Frameworks, Services)	20h	20%	7
3	Kein umsetzbarer Prototyp (zu grosser Aufwand, zu lange Einarbeitungszeit, Technologie nicht vorhanden.	50h	20%	10
4	Ausfall Projektmitarbeiter	15h	40%	4
5	Mangelhafte Kommunikation zwischen den Beteiligten	5h	10%	5

Tabelle 1: Risikotabelle

Umgang mit Risiken

Mögliche Risiken wurden im Voraus festgehalten, mit dem Ziel frühzeitig Vorkehrungen treffen zu können, damit der Arbeitsprozess optimal voranschreiten kann

Im Fall von auftretenden Problemen wird ein Zeitpuffer eingebettet. Dieser ermöglicht es Anpassungen vorzunehmen, um für die Behebung Auftretender Probleme Zeit zu schaffen. Der Fokus wird dann auf die hoch priorisierten Grundelemente der Applikation gelegt.

7.5. Qualitätsmassnahmen

Jeder ist für die Qualität seiner Arbeit und die der anderen Team-Mitgliedern verantwortlich.

Massnahme	Zeitraum	Ziel
Team-Meetings	wöchentlich	Übersicht zum aktuellen Projektstatus gewinnen und Planung dementsprechend aktualisieren. Auf tretende Probleme frühzeitig erkennen und abwenden.
Meilenstein-Reviews	vordefinierte Termine	Projektstatus präsentieren und Experten-Input erhalten.
Testing (manuell)	Kontinuierlich	Fehler frühzeitig erkennen und beheben.
Code-Reviews	Bei Master-Pull-Requests	Hohe Code-Qualität erreichen.
Pair Programming	Wöchentlich	Soviel wie möglich voneinander lernen.

Code Reviews

Auf Wunsch oder vor jedem Merge in den Master-Branch findet ein Review durch ein anderes Teammitglied statt.

Code Style Guidelines

Es wird der Default Code Style der Webstorm IDE verwendet.

Testing

Da nur ein Prototyp gebaut wird, wurde auf das Schreiben von Unit-Tests verzichtet. Aus zeitlichen Gründen wurden die höher priorisierten Aspekte des Prototyps berücksichtigt. Der Prototyp wurde jedoch manuell getestet, indem die verschiedenen Use Cases (siehe III 16.1) durchgespielt wurden.

II. Technischer Bericht

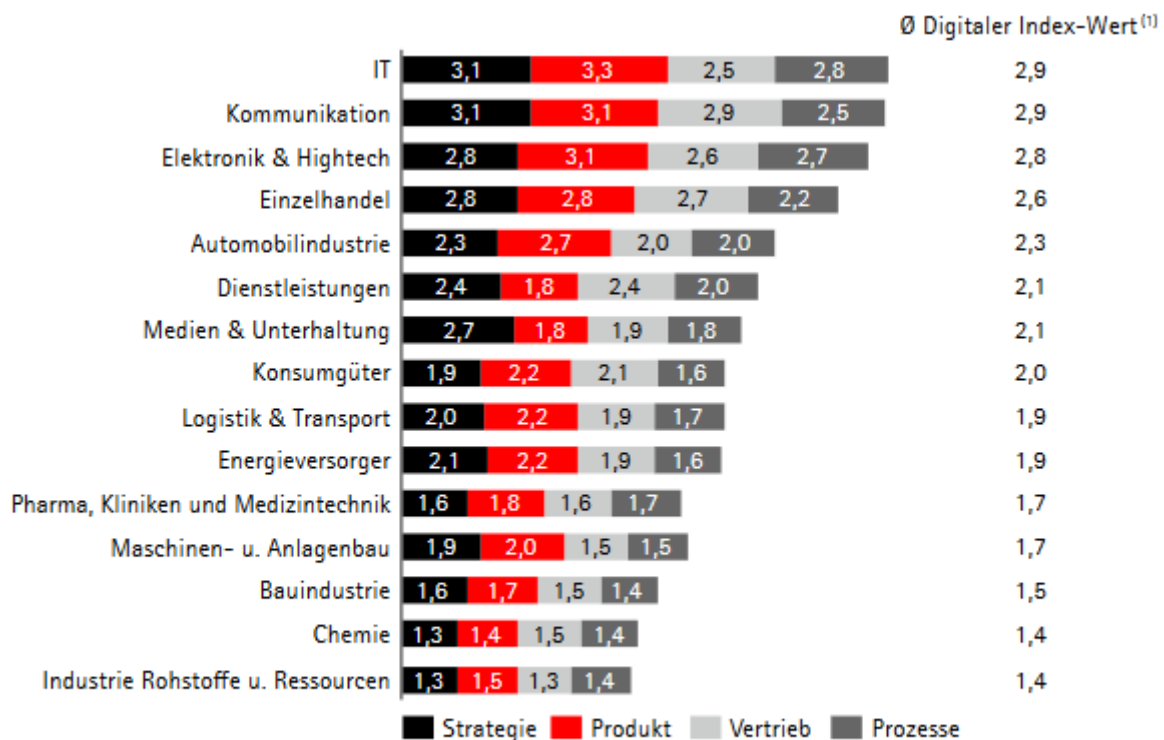
8. BIM – Building Information Modeling

8.1. Was ist BIM?

Building Information Modelling (BIM) ist ein Prozess der Bauindustrie. Er unterstützt die Planung, den Bau und den Unterhalt eines Gebäudes. Bestehende Prozess werden mit Hilfe von verschiedener Software digitalisiert mit dem Ziel die Kollaboration an Bauprojekten zu verbessern.

8.2. Warum BIM?

Im Rahmen der Digitalisierung darf die Baubranche den Anschluss nicht verlieren. Wie man in der Abbildung sehen kann besteht grosses Verbesserungspotential. Dies soll mit der Hilfe von BIM erreicht werden.



(1) Ungewichteter Durchschnitt der vier Dimensionen (Prozesse, Vertrieb, Produkt und Strategie).
Erweiterte Methodik gegenüber Digitalisierungsindex 2015 und 2014

Abbildung 9¹: Digitalisierungsgrad nach Industrie-Cluster (Accenture, 2016)

BIM soll eine Effizienzsteigerung in der Planungsphase von Bauprojekten bringen. Der Kommunikationsfluss soll gezielter und einheitlicher organisiert werden. Durch frühzeitiges und kollaboratives Planen können Fehler in der Konstruktionsphase vermieden und Kosten gespart werden.

Auch am diesjährigen BIM Kongress in Basel war BIM ein grosses Thema. Allerdings nicht auf der Technologieebene, sondern auf der Prozessebene. Es wurde viel über agile Methoden wie z.B. Scrum gesprochen. Ein Umdenken wird gefordert und ist auch dringend nötig, um in der digitalen

¹ https://www.accenture.com/_acnmedia/PDF-5/Accenture-Top500-DE-Executive-Summary-Final-Web.pdf

Transformation voranzukommen. In diesem Bereich konnte der BIM Kongress sicher einen Akzent setzen.

8.3. BIM Integration

BIM soll in mehreren Schritten und über mehrere Jahre hinweg integriert werden. Diese Schritte hat Bauen Digital Schweiz in folgendem Stufenplan dargestellt. «Bauen Digital Schweiz ist die führende Plattform für die digitale Transformation der Schweizer Bau- und Immobilienwirtschaft [...]»² (Bauen Digital Schweiz, kein Datum)

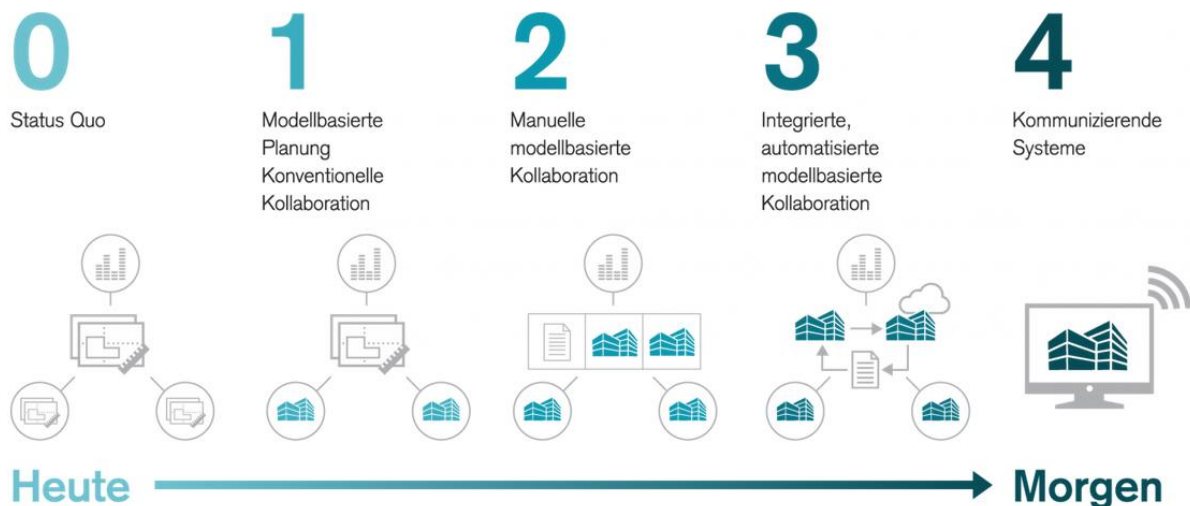


Abbildung 10³: Stufenplan von Bauen Digital Schweiz

Stufe 0

BIM wird nicht verwendet. 2D-Gebäudepläne werden per E-Mail als PDF versendet oder sogar auf Papier ausgedruckt. Laut Prof. Christian Graf befinden sich momentan ca. 90% der Schweizer Bauindustrie auf der Stufe 0. (Bauen Digital Schweiz, 2018)

Stufe 1 - 2

Vereinzelte Firmen in der Schweiz befinden sich zwischen Stufe 1 und 2. Sie setzen sich aktiv mit dem Thema BIM auseinander und verwenden in ihren Arbeitsprozessen bereits BIM Standards und diverse BIM-unterstützende Tools. Die Kollaboration wird immer noch manuell ausgeführt. (Bauen Digital Schweiz, 2018)

Stufe 3

Die Stufe 3 wird von vielen als heiliger Gral bezeichnet. Auf dieser Stufe sind die Prozessschritte automatisiert «und schafft durch „echte“ digitale Interoperabilität die Grundlage für neue, disruptive Geschäftsmodelle» (Bauen Digital Schweiz, 2018)

können alle simultan am gleichen Modell arbeiten. Laut Prof. Graf wird es noch einige Jahre dauern bis die Industrie auf diesem Stand ist.

² <https://bauen-digital.ch/de/ueber-uns/>

³ <https://bauen-digital.ch/de/produkte/stufenplan/>

Stufe 4

Die letzte Stufe geht in den Bereich des Internet of Things (IoT). Intelligente Gebäude können über Sensoren Informationen zur Verfügung stellen und so mit anderen Systemen kommunizieren. (Bauen Digital Schweiz, 2018)

8.4. BIM Workflow

Der BIM Workflow zeigt auf, wie Firmen mit BIM arbeiten. Für die Punkte 4 und 5 kommt ein BIM-Koordinator zum Einsatz. Seine Aufgabe besteht darin den BIM Workflow zu unterstützen, indem er Modelle in den Model Checker importiert und daraus Aufgaben für die entsprechenden Planer erstellt.

1. Die Planer der verschiedenen Branchen erstellen ihre 3D-Modelle auf ihren lokalen Maschinen.
2. Aus den einzelnen Modellen werden IFCs auf einen File Hoster (wird von der Bauleitung bestimmt) geladen.
3. Die Modelle aller Branchen werden bei einem File Hoster gesammelt.
4. 1-N IFCs werden in einen Model-Checker geladen, wo die Modelle auf Grund eines Regel-Sets auf Clashes überprüft werden.
5. Aus den Clashes werden BCF-Issues generiert und auf eine Kollaborationsplattform geladen.
6. Der Planer, an den das Issue adressiert ist, lädt es sich von der Kollaborationsplattform herunter und ...
 1. ...bearbeitet sein Modell auf Grund der Beschreibung im BCF.
 2. Das bearbeitete Modell wird wieder auf die IFC-Plattform geladen.
7. Das Issue wird als «gelöst» auf die Kollaborationsplattform zurück geladen und der Workflow wird bei 3 fortgesetzt bis (fast) alle Clashes in den Modellen behoben wurde.

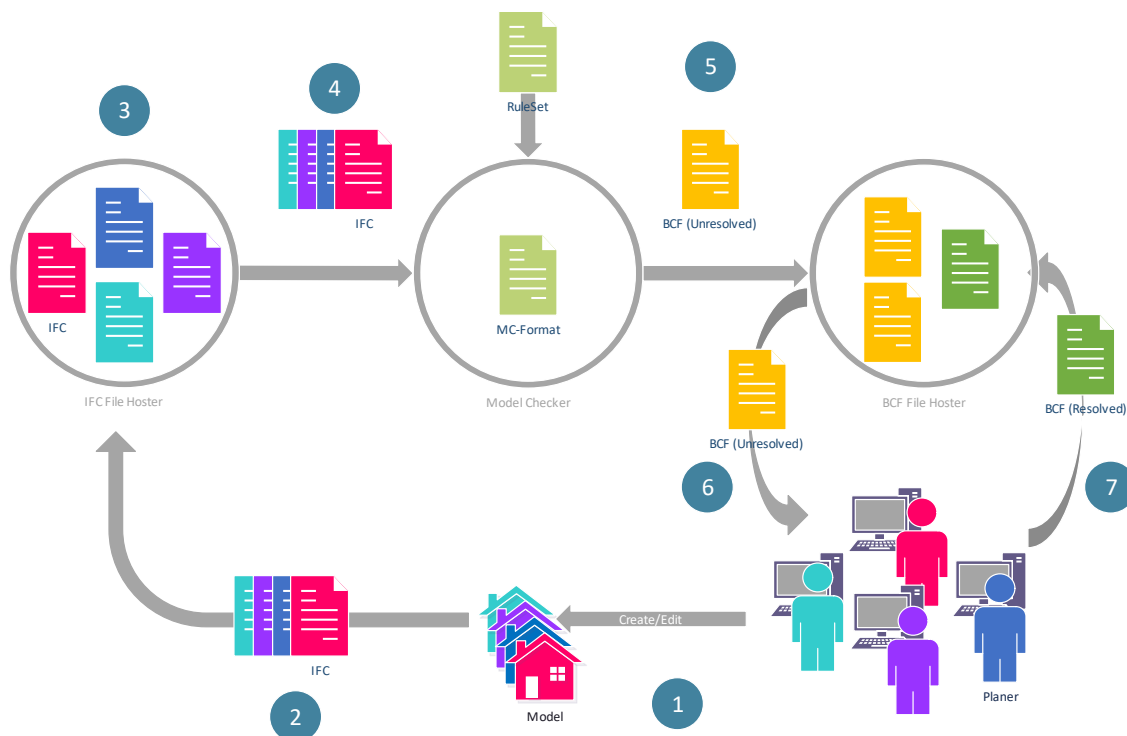


Abbildung 11: Darstellung des BIM Workflows

8.5. BIM Software

Zurzeit gibt es eine Unmenge an BIM Software auf dem Markt, die sich ihren Platz in der BIM-Welt verdienen wollen. Diese Fluktuation wurde auch in den Interviews erwähnt. In der jetzigen Situation sei es üblich, dass etwa alle 6 Monate eine komplett neue Werkzeugkiste verwenden, meinte Amstein-Waltherth.

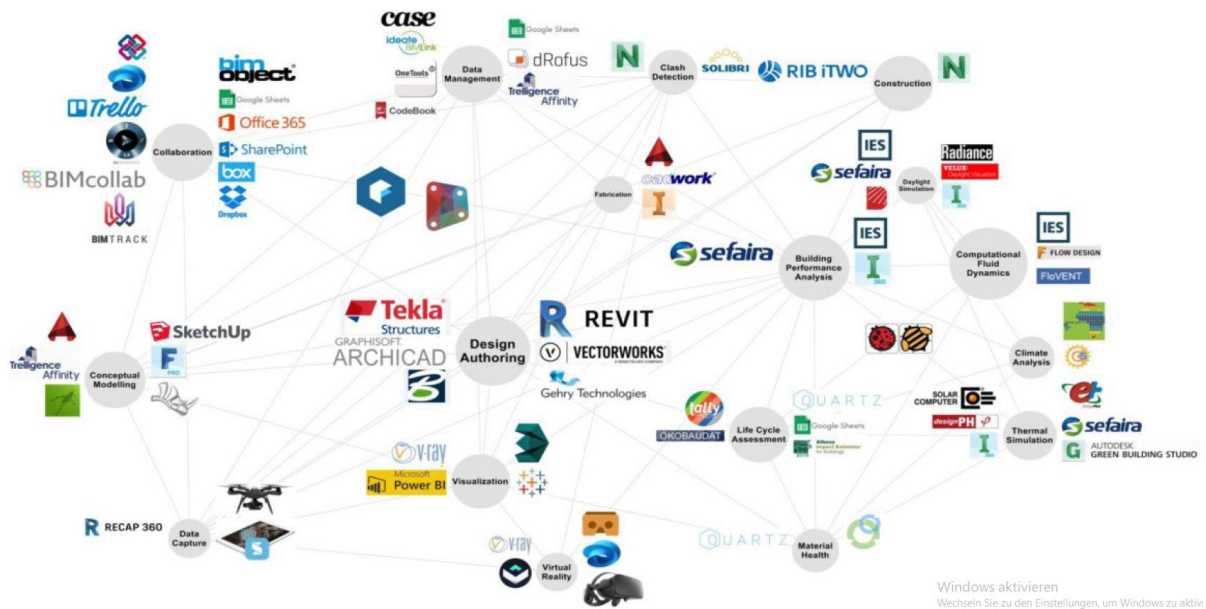


Abbildung 12: Eingesetzte Software von Baufirmen

Die drei wichtigsten Punkte im heutigen BIM Workflow wurden genauer analysiert: Die Modellierung, das Model-Checking und die Kollaborationsplattform mit Issue-Management.

Modellierung

Für das Erstellen von 3D-Modellen ist Revit⁴ ein beliebtes Programm. Das Tool der amerikanischen Firma Autodesk, in Kombination mit Autodesk 360⁵ wäre bereits eine gute Lösung. Mit dieser Kombination wird bereits ein grosser Teil des Workflows nativ abdeckt. Jedoch wäre dies nicht openBIM konform. Ausserdem ist es nicht möglich die gesamte Industrie zu einem Tool zu zwingen, was uns auch die Interviewpartner bestätigt haben. Darum ist IFC ein wichtiger Standard für 3D-Modelle um mit den vielen anderen Tools auf dem Markt wie z.B. Tekla, Vectorworks oder Archicad zu kommunizieren.

⁴ <https://knowledge.autodesk.com/support/revit-products>

⁵ <https://info.bim360.autodesk.com/bim360-api-build-integration>

Model-Checking

Ein Model-Checker ist ein Tool, welches Clashes in einem oder mehreren 3D-Modellen erkennt. Analog zur Rechtschreibprüfung in einem Texteditor. Clashes werden vom Model Checker anhand eines definierten Regelsets erkannt.

Aus den Interviews hat sich gezeigt, dass der Solibri Modell-Checker der de facto Standard für das Überprüfen von Modellen ist.

Der Model Checker verbindet mehrere IFCs zu einer solibri-eigenen SMC-Datei. Das SMC wird auf Kollisionen überprüft, die in Solibri angezeigt werden. Zum Erkennen dieser Kollisionen kann ein selbstdefinierter Regelsatz verwendet werden. Aus den generierten Kollisionen können Issues erstellt und als BCF an die gewünschten Personen zur Bearbeitung geschickt werden.

Kollaborationsplattform

Die Kollaborationsplattform dient zur vereinfachten und zentralisierten Kommunikation der verschiedenen Akteure an einem Bauprojekt. Es ist der Ort, wo Dokumente und Modelle gehostet werden, wo Termine festgelegt und Aufgaben zugewiesen werden.

Einer dieser Kollaborationsplattformen ist BIMcollab. Sie ist darauf spezialisiert BCFs zu verwalten. Mit dem BCF-Manager können Issues in einem beliebigen Modellierungstool importiert und exportiert werden. Dazu muss man nur das Plugin installieren und sich mit seinem BIMcollab-Account anmelden. Es gibt auch einen Modell-Viewer BIMcollab ZOOM, in welchem IFCs visualisiert und validiert werden können.

Eine weitere Kollaborationsplattform ist Trimble Connect. Sie erlaubt es unter anderem 2D Zeichnungen und 3D Modelle auf die Plattform zu laden und mit anderen Projektmitgliedern zu teilen.

9. BIM Standards

Die BIM Standards, sind öffentliche Standards, die von buildingSMART⁶ vorgegeben werden. buildingSMART ist eine not-for-profit Organisation und die treibende Kraft hinter openBIM. openBIM bedeutet, die Benutzung von Standards. (buildingSMART, kein Datum)

Die wichtigsten buildingSMART-Standards für diese Arbeit sind folgende:

- IFC Base Standards ISO 16739:2013⁷ (International Organization for Standardization, kein Datum)
- IFC2x3⁸
- IFC4.1⁹ Infrastructure alignment
- BCF-XML S1005¹⁰
- BCF-API S1006¹¹

9.1. Industry Foundation Classes (IFC)

IFC ist ein allgemeines Datenmodell, welches aus einem nativen Datenmodell generiert werden kann. Dies ermöglicht den Austausch von Daten zwischen verschiedenen proprietären Software-Anwendungen. Das IFC beinhaltet Informationen aller am Bauprojekt mitwirkender Disziplinen über dessen gesamten Lebenszyklus (Planung, Bau, Wartung, Abriss).¹² (buildingSMART, kein Datum)

Jeder Ersteller eines IFCs bleibt alleiniger Besitzer. Änderungen am IFC sollten nie direkt am IFC vorgenommen werden. Dies gehört nicht zum IFC-Workflow. Vielmehr wird nur das Original-Modell geändert aus welchem wieder ein IFC exportiert wird. So kann sichergestellt werden, dass niemand ausser dem qualifizierten Planer, dass IFC bearbeitet und dadurch keine ungewollten Änderungen gemacht werden.

Spezifikation – IFC4 Add2: <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/>

Datenaustausch

Die Lagerung und Teilung der Daten ist durch EXPRESS(ISO 10303-11) und XSD (XML Schema Definition). Das XSD wird aus EXPRESS generiert.

Das Austauschformat von Daten bezüglich der konzeptuellen Schema sind clear text encoding(ISO 10303-21) und XML.

⁶ <https://www.buildingsmart.org/standards>

⁷ <https://www.iso.org/standard/51622.html>

⁸ <http://www.buildingsmart-tech.org/specifications/ifc-releases/ifc2x3-tc1-release/summary>

⁹ <http://www.buildingsmart-tech.org/ifc/IFC4x1/final/html/>

¹⁰ https://github.com/buildingSMART/BCF-XML/tree/release_2_1/Documentation

¹¹ <https://github.com/BuildingSMART/BCF-API>

¹² <https://www.buildingsmart.de/bim-knowhow/standards>

IFC-Workflow

Nachfolgend wird der IFC-Workflow erläutert.

Kurz: Alle IFCs werden unter allen Planer ausgetauscht.

Im Detail:

- Planer 1 (z.B. Architekt) und Planer N kreieren je ihr eigenes lokales Datenmodell in Ihrem lokalen Modellierungswerkzeug.
- Das lokale Modell wird als IFC-Modell exportiert.
- Die IFC-Modelle können nun mit allen beteiligten ausgetauscht werden.

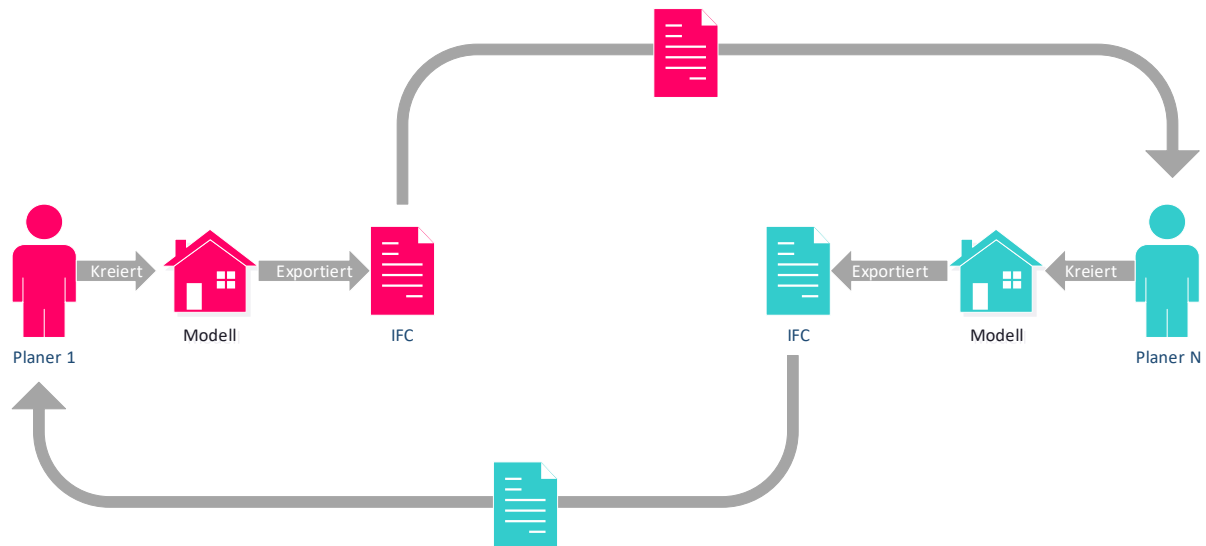


Abbildung 13: Schematischer BIM Workflow, bei dem alle IFC-Modell ausgetauscht werden

9.2. BIM Collaboration Format (BCF)

BCF ist ein Datenformat zum vereinfachten Austausch von Gebäudemodellinformationen zwischen verschiedenen Anwendern. Es basiert auf dem IFC-Austauschformat.

Es ermöglicht eine modellbasierte Kommunikation zwischen verschiedenen Anwendern und informiert über Status, Ort, Blickrichtung, Bauteil, Bemerkung, Anwender und Zeitpunkt im IFC Datenmodell.

Hauptvorteil von BCF ist der beigefügte Betrachter-Standort mit Blickrichtung (engl. viewpoint). Dieser Status wird in einem sog. Topic abgespeichert. (siehe Abbildung) Mit Hilfe dieser Funktion kann ein Fehler im Modell sofort lokalisiert werden und muss nicht, wie bis anhin, von Hand gesucht werden. Dies bedeutet eine enorme Zeitersparnis.

Beispiel ohne BCF: Ein Planer erhält einen 2D-Plan eines Gebäudes als PDF. Folgende Notiz liegt dem Plan bei: «3. Steckdose, im 2. Stockwerk, im 5. Zimmer links ist zu hoch! Bitte verbessern».

Problem: Der Planer navigiert von Hand durch den 2D-Plan zu der angegebenen Position. Nach einiger Zeit findet er die Stelle in seinem Modell und behebt das Problem.

Lösung: Mit Hilfe des BIM Standards BCF kann dieses Problem behoben werden. Das BCF beinhaltet die View desjenigen, welcher den Fehler im Modell erkannt hat. Dadurch wird derjenige, der den Fehler beheben muss, mit Hilfe der View direkt an den Punkt im Modell geleitet wird. Kein aufwendiges Suchen mehr nach der «Nadel im Heuhaufen» bzw. der «Steckdose im Hochhaus»

Beispiel eines Topics

Nachfolgend ist ein Beispiel eines BCF-Topics zusehen.

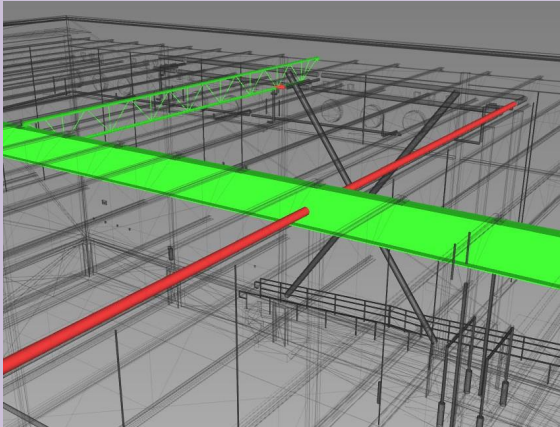
Titel	Träger-Kabel-Clash	Viewpoint
Author	BIM Koordinator	 <p>Abbildung 14¹³: Ansicht eines Clashes (The Webber/Smith Group, kein Datum)</p>
Zugewiesen an	Herr Kabel	
Datum	01.01.2018	
Kommentar	Das Kabel muss um 20cm gesenkt werden.	
Status	Offen	

Abbildung 15: Beispiel eines BCF-Topics

¹³ http://www.webbersmith.com/wp-content/uploads/2015/10/Clash_2-845x650.jpg

BCF-Workflow

- Ein Planer exportiert ein oder mehrere IFCs in einen Modell-Checker.
- Der Modell-Checker überprüft die Modelle nach einem definierten Regel-Set.
- Clashes werden vom Modell-Checker angezeigt und es können daraus BCFs generiert werden.
- Die BCFs werden an die verantwortliche Person gesendet.
- Der Planer passt auf Grund des BCFs sein Original-Modell an und (re-)exportiert das IFC-Modell daraus; dann kann der Ablauf von vorne beginnen.

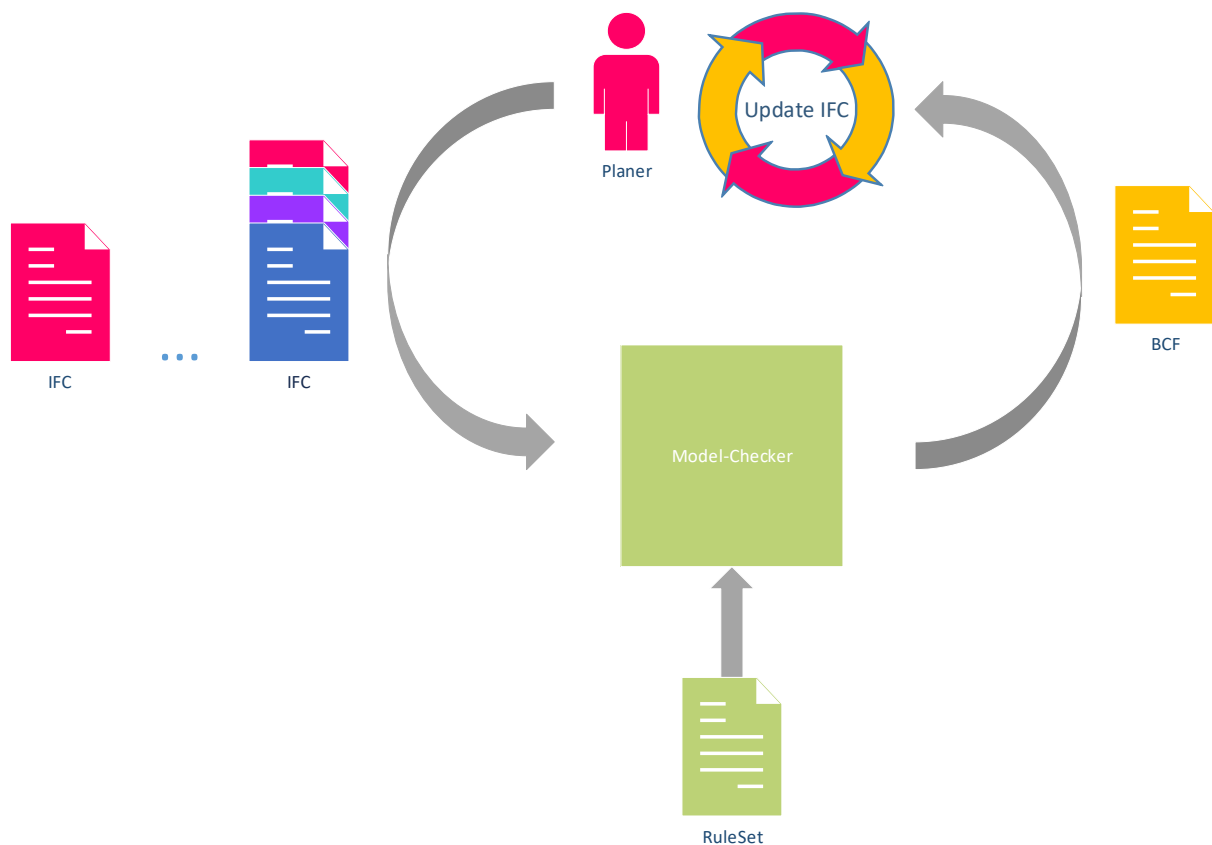


Abbildung 16: Schematischer BIM-Workflow, bei dem IFC-Modelle anhand von BCFs aktualisiert werden.

Datenaustausch

Das BCF-XSD¹⁴ definiert das bcfXML¹⁵ für den Austausch von BCFs zur Workflowkommunikation im BIM-Prozess.

Die BCF-API¹⁶ unterstützt den Austausch von BCF-Topics zwischen Applikationen über eine RESTful API. Die Daten werden mittels URL-Queryparameter und JSON über http gesendet. (buildingSMART, 2018)

Datenstruktur¹⁷

Die Datenstruktur eines BCF (zip):

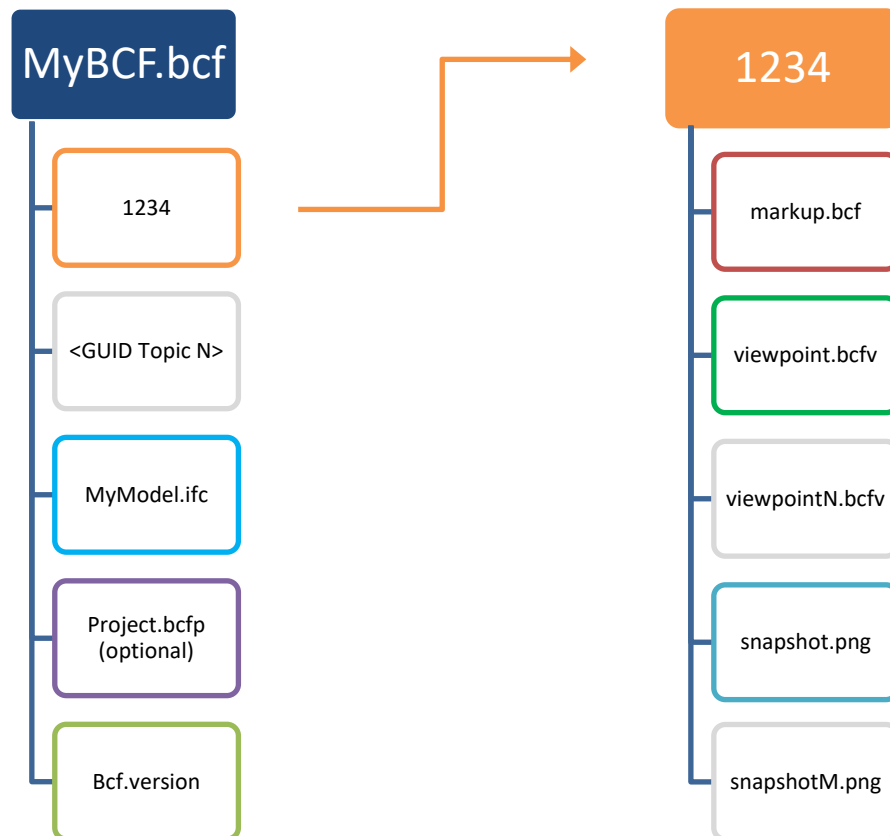


Abbildung 17: Datenstruktur eines BCF

Nachfolgend werden die Details und der Zusammenhang zwischen den einzelnen Dateien aufgezeigt. Alle **roten** Tags in den XMLs sind zwingend zu verwenden.

¹⁴ <http://www.buildingsmart-tech.org/specifications/bcf-releases/bcfxml-v1/markup.xsd/view>

¹⁵ https://github.com/buildingSMART/BCF-XML/tree/release_2_1/Documentation

¹⁶ <https://github.com/BuildingSMART/BCF-API>

¹⁷ https://github.com/buildingSMART/BCF-XML/tree/release_2_1/Documentation

MyBCF.bcf
<p>Beschreibung: Ab BCF v2.1 wurde das BCFzip Format in BCF umbenannt. Dies ist aus unserer Sicht eine ungeschickte Namenswahl, weil es dadurch die gleiche Endung wie das Markup hat. Einziger Unterschied ist das eine kann mit einem Archivierungstool entpackt werden, das andere nicht.</p> <p>Schema: bcf.xsd</p>
Project.bcfp
<p>Beschreibung: Beinhaltet Projektinformationen (ID, Name), zu welchem das Topic gehört.</p> <p>Schema: project.xsd</p>
<pre><?xml version="1.0" encoding="utf-8"?> <Version xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" VersionId="2.1"> <DetailedVersion>2.1</DetailedVersion> </Version></pre>
Bcf.version
<p>Beschreibung: Version des BCF</p> <p>Schema: version.xsd</p>
<pre><?xml version="1.0" encoding="utf-8"?> <Version xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" VersionId="2.1"> <DetailedVersion>2.1</DetailedVersion> </Version></pre>

Markup.bcf
<p>Beschreibung: XML mit BCF Informationen. Braucht im Minimum ein Topic</p> <p>Schema: markup.xsd</p> <p>Referenzen:</p> <ul style="list-style-type: none"> <code>../MyModel.ifc</code> -> referenziert auf IFC innerhalb der Ordnerstruktur <code>Topic Guid</code> -> Ordner Name <code>Viewpoint GUID</code> -> <i>viewpoint.bcfv</i>
<pre> <?xml version="1.0" encoding="utf-8"?> <Markup xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <Header> <File IfcProject="3LIQL2UvjC6xkGKOQxhhVW" isExternal="false"> <Filename>MyModel.ifc</Filename> <Date>2014-02-25T11:50:32</Date> <Reference>../MyModel.ifc</Reference> </File> </Header> <Topic Guid="1234"> <Title>Example BCF</Title> <CreationDate>2015-10-16T12:13:15Z</CreationDate> <CreationAuthor>pfaic@hsr.ch</CreationAuthor> <Index>0</Index> <TopicStatus>Open</TopicStatus> <Description> This topic has thre selected components (see snapshot). All other components should be displayed with their default settings. </Description> <AssignedTo>bmoosman@hsr.ch</AssignedTo> <DueDate>2016-10-16T12:13:15Z</DueDate> <Priority>1</Priority> <DocumentReference> <ReferencedDocument>../MyPDF.pdf</ReferencedDocument> <Description>This is my PDF</Description> </DocumentReference> </Topic> <Viewpoints Guid="5678"> <Viewpoint>Viewpoint_5678.bcfv</Viewpoint> <Snapshot>Snapshot_5678.png</Snapshot> </Viewpoints> <Comment Guid="9999"> <Date>2015-08-31T12:40:17Z</Date> <Author>pfaic@hsr</Author> <Comment> This is a Comment. </Comment> <Viewpoint Guid="5678" /> </Comment> </Markup> </pre>
<p>Alternativ: Verweis auf externes IFC</p> <ul style="list-style-type: none"> <code>isExternal="true"</code> <code><Reference>https://.../MyExternalModel.ifc </Reference></code>

Viewpoint.bcfv

Beschreibung: Beinhaltet eine oder mehrere Kameraperspektiven und Topic bezogene Komponenteninformationen.

Schema: visinfo.xsd

Referenzen: **Component** -> Hervorhebung im [snapshot.png](#).

```
<?xml version="1.0" encoding="utf-8"?>
<VisualizationInfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  Guid="5678">
  <Components>
    <ViewSetupHints SpacesVisible="false"
      SpaceBoundariesVisible="false"
      OpeningsVisible="true" />
    <Selection>
      <Component IfcGuid="1GU8BMEqHBQxVAbwRD$4Jj" />
      <Component IfcGuid="0AQJSsoeDDvwVqSNcwjy55" />
      <Component IfcGuid="3DOu_tSXP6evQgY8Ml4CtC" />
    </Selection>
    <Visibility DefaultVisibility="true" />
  </Components>

  <PerspectiveCamera>
    <CameraViewPoint> <!-- Kameraposition im Raum -->
      <X>-16.4296623429377</X>
      <Y>-0.160221005721745</Y>
      <Z>4.97078624471391</Z>
    </CameraViewPoint>
    <CameraDirection> <!-- Ausrichtung der Kamera -->
      <X>0.803157911518554</X>
      <Y>-0.590456679305644</Y>
      <Z>-0.0793614726687355</Z>
    </CameraDirection>
    <CameraUpVector> <!-- Bestimmt wo die Oberseite der Kamera ist -->
      <X>-0.379182758930675</X>
      <Y>0.278763630961051</Y>
      <Z>0.882332858610135</Z>
    </CameraUpVector>
    <FieldOfView>60</FieldOfView>
  </PerspectiveCamera>

  <ClippingPlane> <!-- Querschnitt des Modells -->
    <Location>
      <X>0</X>
      <Y>1.234</Y>
      <Z>2.468</Z>
    </Location>
    <Direction>
      <X>0</X>
      <Y>0</Y>
      <Z>0</Z>
    </Direction>
  </ClippingPlane>
</VisualizationInfo>
```

Snapshot.png

Beschreibung: Darstellung der 3 Ausgewählten **Komponenten** (Bild, hellblau), auf die im Viewpoint.bcfv referenziert wird.

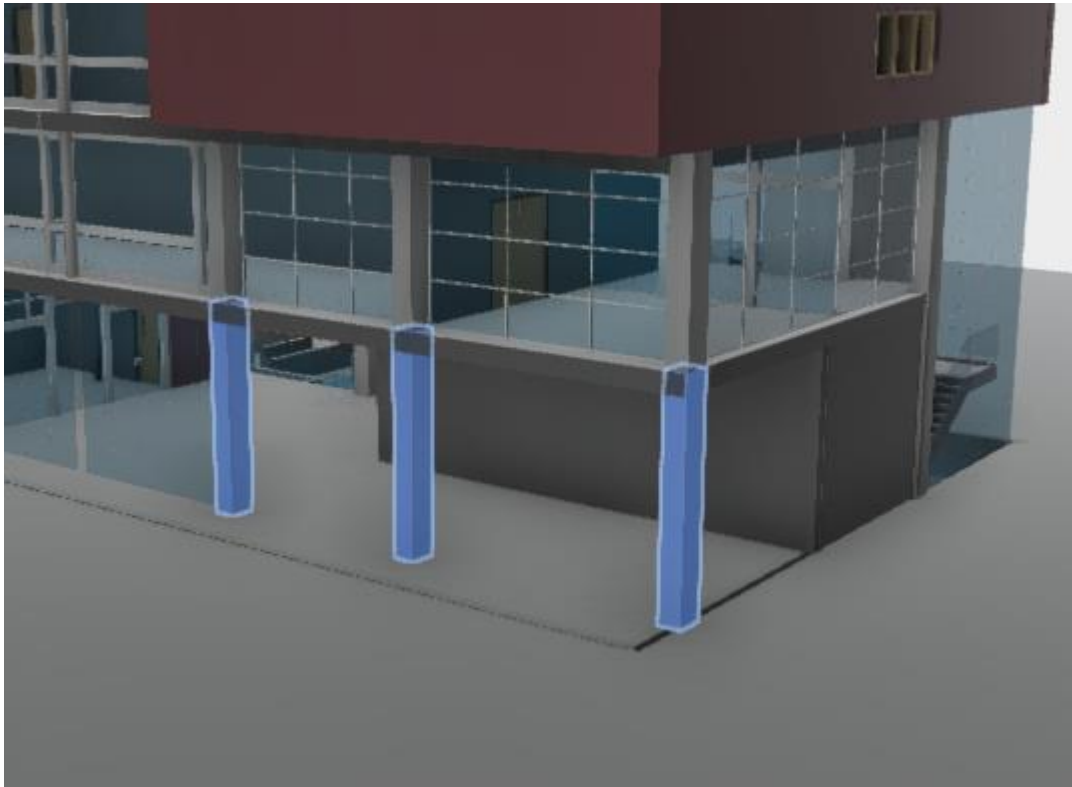


Abbildung 18: Ansicht eines Modells, in welchem 3 Objekte ausgewählt wurden.

10. Interviews

Die Interviews wurden mit folgenden Firmen geführt:

- Dozent für Projektierung an der HSR
- b+p Baurealisation
- Leiter Digital Planen und Bauen
- BIM Consultant Amstein-Walthert

Das Hauptziel der geführten Interviews ist es den BIM Workflow zu verstehen und wie dieser bei den Firmen durchgeführt wird.

Die durchgeführten Interviews haben gezeigt, dass das Anbieten eines Model-Viewer für smino Pflicht ist. Die IFC Modelle, die in diesem Viewer angezeigt werden sollen, könnten auch direkt von smino gehostet werden. b+p Baurealisation sind nicht zufrieden mit den vorhandenen 3D-Model-Viewern und wünschen sich einen performanten Viewer, welcher auch mit grösseren Files problemlos umgehen kann.

Bei Amstein-Walthert sei es in der momentanen Situation üblich, dass ca. alle 6 Monate eine komplett neue Werkzeugkiste verwendet wird. Dies ist eine gute Möglichkeit für smino ihr Produkt im BIM Markt platzieren zu können.

Der Solibri Modell Checker ist das einzige etablierte Tool in der Branche. Er ist weit verbreitet und funktioniert sehr gut. Eine weitere Möglichkeit, welche von Amstein-Walthert erwähnt wurde, ist die Verwaltung und das Definieren von Regelsets, mit welchen der Modell Checker arbeitet.

11. Chancen für smino

smino kann den bestehenden BIM-Workflow (siehe Punkt 8.4) vereinfachen in dem sie IFC und BCF auf derselben Plattform verwalten. (Siehe Abbildung 18)

Dadurch sieht der smino-Workflow folgendermassen aus:

- Die Planer (links) laden ihre kreierten Modelle auf smino
- Von dort aus werden sie in einem Model Checker (rechts) überprüft.
- Die daraus entstandenen Topics werden wieder auf smino geladen.
- Nun können die Planer wieder auf smino ihre Tickets bearbeiten.

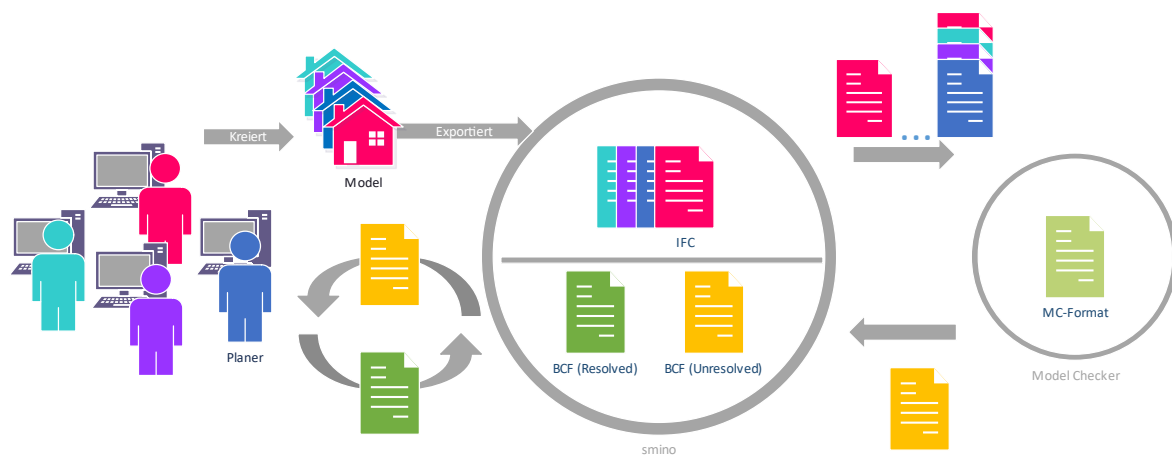


Abbildung 19: Darstellung des möglichen smino-Workflows

Die folgenden Chancen für smino ergaben sich aus der Recherche und den geführten Interviews:

IFC Verwaltung

Auf der smino Plattform soll der Up- und Download von IFCs ermöglicht werden. Des Weiteren sind folgende Funktionalitäten für die IFCs denkbar:

- Versionierung
- Change History
- Notifikation für Änderungen abonnieren

BIM Viewer

Smino könnte ihre bereits bestehende Plattform um einen 3D Viewer erweitern. Der Viewer soll folgende Funktionen können:

- Integration auf Plattform
- Visualisierung mehrerer IFC
- Caching von 3D Modell für offline Benützung
- Schnittstelle von 3D Modell zu 2D Plänen (z.B. PDF)

Vergleiche der verschiedenen Viewer sind unter Punkt 14.7 aufgeführt.

BCF Verwaltung

Die bereits bestehenden smino Issues können um BCFs erweitert werden. Auf der Plattform können BCFs importiert oder erstellt werden. Dafür müssten BCFs auf smino Issues abgebildet werden.

Mapping BCF Topics - Smino Issues

Die Tabelle 1 zeigt auf wie ein BCF auf ein smino Issue abgebildet werden kann und wo Schwierigkeiten entstehen können.

Die Liste besteht aus Typen aus den Schemata Markup¹⁸ und Extension¹⁹

¹⁸ https://github.com/buildingSMART/BCF-XML/blob/release_2_1/Schemas/markup.xsd

¹⁹ https://github.com/buildingSMART/BCF-XML/blob/release_2_1/Extension%20Schemas/extensions.xsd

BCF Topic			Smino Issue		
Name	Typ	Muss	Name	Typ	Muss
GUID	String	X			
IfcProject	String				
RelatedTopic (1..n)	String				
Title	String	X	Title	nvarchar	
CreationDate	DateTime	X	CreatedDate	DateTime	X
CreationAuthor	String	X	OwnerCollaboratorId	Collaborator	X
AssignedTo	String		ResponsibleCollaboratorId	Collaborator	
Description	String		Description	nvarchar	
DueDate	DateTime		DueDate	DateTime	
TopicType (Comment, Issue, Request, Solution)	String		Type	int	
TopicStatus (Open, In Prog., Closed, Reopen.)	String		Status	int	X
DocumentReference	String		File		
			BlueprintMarkerId		
			Image		
Comment (1..n)	String		Konversation	Konversation	
			Id		X
			ProjectId		X
			CreationType (BCF hinzufügen)		X
			ProjectIssueSeqNumber	int	X
			ParentIssued	Id	
			Location	nvarchar	
			PublishedDate	DateTime	
			ModifiedDate	DateTime	

Tabelle 2: Abbildung von smino Issues auf BCF Topics

Legende

BCF Tag ist nicht Teil des smino Issue Schemas

Logischer Typ vorhanden aber falscher effektiver Typ

Smino Tag ist nicht Teil des BCF Schemas

Probleme

- Allgemein
 - CreationAuthor und AssignedTo müssen auf die entsprechenden Namen des Collaborators gemappt werden.
 - TopicType und TopicStatus müssen auf die entsprechenden Integer-Werte der smino Typen und Status gemappt werden und umgekehrt.
 - CreationType muss um den Typ «BCF» erweitert werden.
 - Eine Konversation kann weitere Konversationen beinhalten, während ein Topic 1 oder mehrere Comments angehängt werden können.
- BCF to smino Issue
 - Id, ProjectId, CreationType, ProjectIssueSegNumber müssen erstellt werden.
- Smino Issue to BCF
 - GUID muss generiert werden.

Metadaten

- Metadaten aus IFC extrahieren.
- Metadaten mit Kommunikationsdaten (Protokolle, etc.) als Tags verknüpfen.
- Filtern und Sortieren nach Tags (z.B. Raum / Material)

Validierungsmöglichkeiten

- XSD Validierung
- Validierung neuer IFC Datei gegen Regeln (z.B. gesetzliche Vorgaben)
- Validierung neuer IFC Datei gegen Referenzmodell (z.B. für Clash-Report)
- Erzeugen von BCF Topics, wenn in anderem Modell Änderungen anfallen

Nicht interessant für smino

Das Feature Field-To-BIM ist nicht der Fokus dieser Arbeit, kann jedoch für zukünftige Erweiterungen interessant sein. Field-To-BIM ist der Vorgang des Scannens einer Baustelle, um daraus ein 3D-Modell zu generieren.

12. BIM-Plattformevaluation

Die Plattformevaluation dient dazu einen Überblick der Softwarelösungen auf dem BIM-Markt zu verschaffen. Ziel ist es für smino eine Entscheidungsgrundlage zu bieten und eine geeignete Software zur Entwicklung eines Prototypens auszuwählen.

12.1. Kriterien

Es wurden mehrere Produkte getestet mit dem Ziel ein BIM Setup für smino zu finden, mit welchem sie Ihre bestehende Lösung erweitern können.

Auswahlkriterien

Die getesteten Frameworks **mussten** folgende Kriterien erfüllen, um für smino relevant zu sein:

- API für Viewer-Integration anbieten
- Modell muss rendern
- Modellrendering ≤ 10 Sekunden
- Markierungen an einem Modell im Viewer anbringen.

Folgende Frameworks wurden analysiert und auf Grund eines oder mehrerer der oben genannten Kriterien **nicht** weiter evaluiert:

- BIMer (Einfacher Online Viewer)
- Apstex IFC Framework (Viewer hat zu wenig Funktionalität)
- Allplan (Eigenständige Lösung)
- Hxgnsmartbild (Eigenständige Lösung)
- ACCA Software (Eigenständige Lösung)

Bewertungskriterien

Folgende Bewertungskriterien wurden berücksichtigt:

- Viewer-Performance
- Viewer-Funktionalität (Einige Funktionen wie Explosionsansicht oder Querschnitt sind vom Modell abhängig.)
- Preis
- Open Source
- Dokumentation
- Look and Feel

12.2. Übersicht

Eine Übersicht der nähererevaluierten Tools:

BIMserver	xBIM	Autodesk Forge	bimsync
<ul style="list-style-type: none"> + Open source + Plugin-Basiert + Gratis + Guter Viewer 	<ul style="list-style-type: none"> + Open Source 	<ul style="list-style-type: none"> + Gut dokumentiert + Grosse Firma + Unterstützt viele Datenformate + Viewer braucht weniger RAM als bimsync 	<ul style="list-style-type: none"> + 100% open BIM Standards + 2D und 3D verbunden im Viewer + Sehr Guter Viewer + Gut dokumentiert
<ul style="list-style-type: none"> - Wenig Viewer-Funktionen - Wenig dokumentiert 	<ul style="list-style-type: none"> - Viewer master-branch failed 	<ul style="list-style-type: none"> - Nicht open source - Flimmerndes Rendering des Viewers 	<ul style="list-style-type: none"> - 2 GB RAM reichen für Viewer nicht aus.

Tabelle 3: Übersicht der getesteten Plattformen

BIMserver

Der BIMserver hat einen performanten Viewer, der jedoch fast keine Funktionalität besitzt.

xBIM

Der Master-Branche des xBIM hat den Status «failed» und wurde aus diesem Grund nicht weiter evaluiert.

Autodesk Forge

Für Autodesk Forge spricht der performantere Viewer. Er läuft mit weniger RAM flüssiger als der bimsync Viewer. Das Negativste an Forge ist der proprietäre Aspekt.

bimsync

bimsync hatte den benutzerfreundlichsten Viewer. Das bimsync zu 100% auf BIM Standards setzt, ist ein weiterer Grund, der für bimsync spricht. Das mit Abstand beste bimsync Feature ist jedoch, dass 2D Pläne direkt aus den 3D Modellen generiert und es kann auf ihnen Navigiert (Stockwerk, Raum) werden.

(Die einzelnen Plattformen werden anschliessend im Detail aufgezeigt)

12.3. BIMserver

BIMserver ist ein open source Framework, das zur Lagerung und Verwaltung von IFC-Modellen dient. Der Server ist Plugin orientiert und soll als Grundbaustein für Entwickler dienen.²⁰

Das BIMserver Framework startete als holländisches Geek-Projekt, welches anschliessend von Partnern (Netherlands Organisation for applied scientific research TNO und der Eindhoven University of Technology) unterstützt wurde²¹.

Systemanforderungen

Systemanforderungen für BIMserver 1.5:

- Memory: ca. 15-mal die Grösse des grössten IFC-Files (ungezippt) mal Anzahl Benutzer.
- JRE, JDK 8
- Tomcat 8 oder höher / Jetty 8 oder höher
- Ist von Maven abhängig.

APIs und Services

Produkt	Beschreibung
Revisioning ²²	
IFC Versionisierung ²³	Daten sind in einer Key (PID + OID + RID) -Value Tabelle hinterlegt. Modelle werden in einem Projekt gespeichert.
IFC Hosting	Kein Datenverlust durch das Mapping
Service Interface ²⁴	

Viewer

BIMsurfer ist der Viewer des BIMserver. Hierbei handelt es sich um ein Plugin, welches bereits bei der Installation des BIMservers dabei ist.

Für mehr Informationen zum Viewer siehe Punkt 14.7 Viewer-Analyse.

Lizenz

Der BIMserver ist unter einer Kombination der Affero GPL²⁵, GPLv3²⁶ und LGPL. Er kann in kommerziellen und closed Produkten verwendet werden. Auch die verwendeten Plugins werden von Affero GPL abgeleitet. Dies gilt jedoch nicht für Snippets, GUIs und remote Services.²⁷

Daten

Die Philosophie von BIMserver besagt, dass die Daten einem selbst gehören. Es soll kein Vendor Lock-In geben. Geschlossene Datenformate werden vermieden.²⁸

²⁰ <http://bimserver.org/use-cases/>

²¹ <http://bimserver.org/about/who-is-doing-this/>

²² <https://github.com/opensourceBIM/BIMserver/wiki/SRevision>

²³ <https://github.com/opensourceBIM/BIMserver/wiki/Database---Versioning>

²⁴

<https://github.com/opensourceBIM/BIMserver/blob/master/PluginBase/src/org/bimserver/shared/interfaces/ServiceInterface.java>

²⁵ <http://www.gnu.org/licenses/agpl.html>

²⁶ <http://www.gnu.org/licenses/gpl.html>

²⁷ <http://bimserver.org/license/>

²⁸ <http://bimserver.org/about/philosophy/>

Preis

BIMserver ist gratis.

Es gibt jedoch eine Patreon Seite (<https://www.patreon.com/opensourcebim>) mit folgenden Optionen:

Produkt	Beschreibung
Early Access für Testing und Feedback	10\$ / Monat
Beinflusse die Entwicklungsrichtung	100\$ / Monat
Support innerhalb weniger Stunde und Hilfe beim Setup, Hosting und der Entwicklung des eigenen Plugins	800\$ / Monat

12.4. xBIM

xBIM ist ein .NET Toolkit zur Entwicklung von open-source BIM Software. Es unterstützt den buildingSMART Standard IFC und besteht aus zwei Core Libraries: xBIM Essentials und xBIM Geometry.

xBIM Essentials erlaubt den Entwicklern CRUD Operationen auf IFCs auszuführen. Auf dem Git-Repository wird ständig committed.

Viewer

Modelle werden mit Hilfe der xBIM Geometry Library²⁹ dargestellt. Leider ist der Build des Master-Banches nicht erfolgreich. Dies hat uns verunsichert und haben die Analyse von xBIM vernachlässigt.

Relates to Nuget Package Xbim.Geometry

Build Status (master branch): **TC build failed**

Build Status (develop branch): **TC build success**

Abbildung 20³⁰: Ausschnitt aus dem Git-Repository von xBIM

Lizenzen

Die xBIM Library ist unter der CDLL Open Source Library verfügbar.

Des Weiteren werden folgende Third Party Lizenzen verwendet:

- 'OpenCASCADE' Geometry Engine: <http://www.opencascade.org/> - OPEN CASCADE Public License
- 'Gardens Point Parser Generator' <http://gppg.codeplex.com/> - New BSD Licence
- Elements of '3D Tools' WPF library <http://3dtools.codeplex.com/> - MS Permissive Licence
- Log4net: <http://logging.apache.org/log4net/> - Apache 2.0 Licence
- NPOI: http://npoi.codeplex.com - Apache 2.0 Licence
- NewtonSoft JSON: <http://json.codeplex.com/> - MIT Licence

Preis

Gratis

²⁹ <https://github.com/xBimTeam/XbimEssentials/blob/master/LICENCE.md>

³⁰ <https://github.com/xBimTeam/XbimGeometry>

12.5. Autodesk Forge

Forge ist ein PaaS (Platform as a Service) der amerikanischen Firma Autodesk. Die cloudbasierte Plattform unterstützt bei der Entwicklung von BIM Lösungen.

APIs und Services

Forge bietet ein Set von RESTful APIs an. API calls können nur noch mit TLS 1.2 gemacht werden.

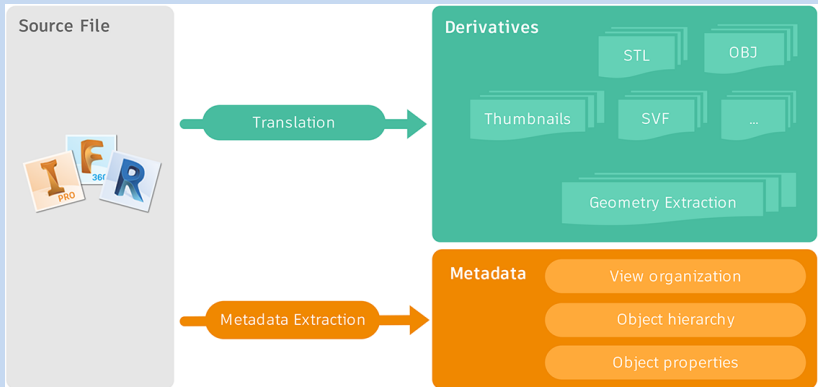
Produkt	Beschreibung
BIM 360 API ³¹	Projekt- und Issue-Management und Revisioning. Upload von Source Files.
Data Management API ³²	Zugriff von 1...N Forge Apps auf dasselbe Backend (Project-, Data-, Schema-, Object Storage Service)
Model Derivative API ³³	<p>Um Modelle im Forge Viewer rendern zu können, wird eine SVF-Datei (Serial Vector Format, spezifiziert gemäss Three.js) benötigt. Das SVF wird gemäss einer Translationstabelle³⁴ aus einem Source File generiert. Weiter wird ein URN und ein Thumbnail generiert.</p> <p>Mit der Model Derivative API werden auch Metadaten aus den Modellen extrahiert.</p> 
Design Automation API ³⁶	Repetitive Tasks mit Hilfe von Skripts auf DWG-Dateien (2D/3D Design) Ausführen. (erstellen, abfragen, oder in andere Formate konvertieren)
Reality Capture API ³⁷	Mittels Scans oder Drohnenshots Modelle oder Texturen herstellen.
Authentication ³⁸	Token-Basierte Authorization und Authentication basierend auf OAuth2

Abbildung 21³⁵: Funktionsweise der Model Derivative API

³¹ <https://forge.autodesk.com/en/docs/bim360/v1/overview/introduction/>

³² https://forge.autodesk.com/en/docs/data/v2/developers_guide/overview/

³³ https://forge.autodesk.com/en/docs/model-derivative/v2/developers_guide/overview/

³⁴ https://forge.autodesk.com/en/docs/model-derivative/v2/developers_guide/supported-translations/

³⁵ https://forge.autodesk.com/en/docs/model-derivative/v2/developers_guide/basics/

³⁶ https://forge.autodesk.com/en/docs/design-automation/v2/developers_guide/overview/

³⁷ https://forge.autodesk.com/en/docs/reality-capture/v1/developers_guide/overview/

³⁸ https://forge.autodesk.com/en/docs/oauth/v2/developers_guide/overview/

Viewer

Siehe 14.4 Viewer-Analyse.

Daten

Forge betreibt Servers in den vereinigten Staaten und Europa (EMEA: Europa, mittlerer Osten, Afrika). Am BIM-Kongress in Basel 2018 wurde in einem Gespräch mit einem Autodesk Mitarbeiter

Bei Daten, die auf den Forge Object Storage Service (OSS) geladen werden, kann die Region (USA / EMEA) ausgewählt werden.³⁹

Preis

Produkt	Preis
Allgemein	Zugriff auf alle Forge APIs 5GB Storage 100 Cloud Credits (CC) = \$100
Viewer	Gratis
Data Management API	Gratis
	1.5 CC für komplexe Jobs (Konvertierung einer Revit- oder Navisworksdatei in ein unterstütztes Format ⁴⁰)
Model Derivative API	0.2 CC für einfache Jobs (Konvertierung einer Nicht-Revit- oder Navisworksdatei in ein unterstütztes Format ⁴¹)
Design Automation API	4.0 CC / Prozessierungsstunde
Reality Capture API	3.5 CC / Gigapixel verarbeitet

³⁹ <https://forge.autodesk.com/blog/where-data-stored-currently>

⁴⁰ https://forge.autodesk.com/en/docs/model-derivative/v2/developers_guide/supported-translations/

⁴¹ https://forge.autodesk.com/en/docs/model-derivative/v2/developers_guide/supported-translations/

12.6. bimsync

Bimsync ist eine Plattform der norwegischen Firma Catenda. Die Firmenmehrheit liegt bei den Angestellten.

Bimsync basiert zu 100% auf open BIM Standards.

Services

Es werden zwei Produkte angeboten:

Produkt	Beschreibung
Bimsync Arena	<ul style="list-style-type: none"> Fertige Lösung, welche mit Bimsync Boost erstellt wurde (inkl. Viewer) Dashboard mit Statistiken Issue Management
Bimsync Boost	<ul style="list-style-type: none"> Zugriff auf bimsync APIs IFC Hosting (inklusive Revisionen und Changehistory) Hilfe von bimsync Entwicklern. White-Label-Product (Benutzer wissen nicht, dass Daten bei bimsync gehosted werden.)

APIs

API	Beschreibung
bimsync REST API ⁴²	<ul style="list-style-type: none"> Projektmanagement Revisionen Objektinformationen und Modellstrukturen Wird laufend weiterentwickelt.
2D Viewer API ⁴³	<ul style="list-style-type: none"> Ermöglicht das verwenden eines 2D Viewers
3D Viewer API ⁴⁴	<ul style="list-style-type: none"> Ermöglicht das verwenden eines 3D Viewers
BCF REST API ⁴⁵	<ul style="list-style-type: none"> Basiert auf der buildingSMART BCF-API⁴⁶ Unterstützt alle BCF Versionen. Import und Export auf dem eingebauten BCF Server mittels HTTPS und JSON. (App muss beim bimsync registriert werden.)
Viewer Widget API ⁴⁷	<ul style="list-style-type: none"> 2D und 3D synchronisieren Tatenkürzel ...

Viewer

Der 2D und 3D Viewer benötigt jQuery 1.x.

Für den Viewer muss zuerst ein Access-Token über die bimsyncAPI generiert werden.

Für weitere Informationen siehe Punkt 14.7 Viewer-Analyse.

⁴² <https://bimsync.com/developers/reference/api/v2#introduction>

⁴³ <https://bimsync.com/developers/reference/viewer-2d/beta>

⁴⁴ <https://bimsync.com/developers/reference/viewer-3d/1.0>

⁴⁵ <https://bimsync.com/developers/reference/bcf/beta#create-issue-board>

⁴⁶ https://github.com/BuildingSMART/BCF-API/tree/release_1_0

⁴⁷ <https://bimsync.com/developers/reference/viewer-widget/1.0>

Daten

bimsync verspricht, dass die eigenen Daten nur einem selbst gehören. Es gibt kein Vendor lock-in.⁴⁸
Die Daten werden alle in Europa (Irland) gespeichert.

Preis

Produkt	Preis
Arena	Prozentualer Anteil an Bauprojekten
Viewer	Grundgebühr + Gebrauchsgebühren

⁴⁸ <https://home.bimsync.com/>

12.7. Viewer-Analyse

Die Viewer-Performance-Tests wurden auf folgenden Systemen getestet:

Testsysteme:

PC (P)	
Hersteller	Microsoft Surface Book
Betriebssystem	Windows 10
CPU	i5 2.5GHz
RAM	8 GB
Auflösung	3000 x 2000
Browser	Firefox 63.0.1 (64-Bit)

Tablet (T)	
Hersteller	Huawei MediaPad T3 10
Betriebssystem	Android 7.0.0
CPU	Snapdragon 425
RAM	2 GB
Auflösung	1280 x 800
Browser	Chrome 70.0.3538.80

Mobile (M)	
Hersteller	Redmi 4X
Betriebssystem	Android 7.1.2
CPU	1.4GHz
RAM	3 GB
Auflösung	720 x 1.280
Browser	Chrome 70.0.3538.80

BIMserver	
Umgebung	Azure VM
Betriebssystem	Windows Server 2016 Datacenter
CPU	E5-2673 v4 2.3GHz
RAM	8 GB
Speicher	126 GB

Viewer-Performance-Test

Die Upload- und Processing-Geschwindigkeit ist der Mittelwert aus 3 Uploads.

Auf dem Tablet und Mobiltelefon wurden keine FPS gemessen, sondern Hands-On Tests durchgeführt.

Dafür wurde folgende Bewertung verwendet:

G = Gut, Viewer läuft flüssig.

M = Medium, Viewer läuft langsam aber er kann noch produktiv verwendet werden.

S = Schlecht, Viewer ist sehr langsam, produktiv nicht zu gebrauchen.

	BIMserver	Forge	Bimsync
Upload & Process Time (in seconds)			
3.76 MB	50	48	13
13.49 MB	165	60	16
34.82 MB	345	115	90
67.07 MB	672	136	96
Load Viewer on PC (in seconds)			
3.76 MB	3	7	3
13.49 MB	5	8	3
34.82 MB	18	8	5
67.07 MB	22	8	5
Load Viewer on Tablet (in seconds)			
3.76 MB	x	6	4
13.49 MB	x	6	6
34.82 MB	x	8	22
67.07 MB	x	10	26
Load Viewer on Mobile (in seconds)			
3.76 MB	x	6	10
13.49 MB	x	7	11
34.82 MB	x	8	20
67.07 MB	x	9	25
FPS while navigating on PC			
3.76 MB	20	37	55
13.49 MB	18	35	53
34.82 MB	12	32	49
67.07 MB	10	25	45
Response feeling while navigating on Tablet			
3.76 MB	x	G	G
13.49 MB	x	G	G
34.82 MB	x	G	M
67.07 MB	x	M	S
Response feeling while navigating on Mobile			
3.76 MB	x	G	G
13.49 MB	x	G	G
34.82 MB	x	G	M
67.07 MB	x	G	M

Tabelle 4: Viewer-Performance-Vergleich

Nutzwertanalyse

Kriterium	Gewicht	BIMserver		Forge		Bimsync	
		Note	Punkte	Note	Punkte	Note	Punkte
Viewer-Performance	0.4	1.67	0.67	5.33	2.13	3.67	1.47
PC		3		5		6	
Tablet		1		5		3	
Mobile		1		6		2	
Viewer-Funktionalität*	0.3	2.43	0.73	4.81	1.44	4.33	1.3
2D					1		1
3D			1		1		1
Querschnitt					1		
Aus-/Einblenden			1		1		1
Explosionsansicht**					1		
1st Person					1		1
Messen					1		1
Markieren							1
Mehrere Modelle			1				1
IFC2x3			1		1		1
IFC4			1		1		
Native Modelle					1		
Navigationsgeschwindigkeit einstellen							1
Revisionen			1		1		1
Offline-Fähigkeit					1		
Stand-Alone							
Kameraposition über API steuern					1		1
Kameraposition auslesen					1		1
Objekte über API selektieren und fokussieren					1		1
Selektierte Objekte als Referenz auslesen					1		1
Preis	0.05	6	0.3	4	0.2	3	0.15
Open Source	0.1	6	0.6	1	0.1	1	0.1
Dokumentation	0.05	2	0.1	5	0.25	4	0.2
Look and Feel	0.1	4	0.4	6	0.6	5	0.5
	1		2.80		4.73		3.72

Tabelle 5: Nutzwertanalyse

* Die Note für die Viewer-Funktionalität lässt sich mit Hilfe folgender Formel bestimmen wobei die max. mögl. Punktezahl 21 beträgt:

$$Note = \frac{erzielte\ Punktezahl * 5}{max.\ mögl.\ Punktezahl} + 1$$

Formel 1: Formel zur Notenberechnung

** Funktioniert nicht auf dem Tablet und Mobile.

Viewer-Analyse-Fazit

Aus der Nutzwertanalyse geht deutlich hervor, dass Forge eine gute Wahl für ein Framework ist, welches smino für den Aufbau ihrer Lösung verwenden können.

Nichtsdestotrotz könnte auch das zweitplatzierte bimsync Frameworks für smino interessant sein. Hier werden die beiden Plattformen nochmal direkt, anhand der für smino wichtigsten Punkte, miteinander verglichen:

Punkt 1 - Daten

Wenn es für smino wichtig ist, dass Ihre Daten ausschliesslich in Europa gehalten werden, dann könnte bimsync von Interesse sein. Bimsync ist eine europäische Firma, welche die Daten in Irland lagert. Autodesk Forge ist eine amerikanische Firma mit Servern in den USA und Europa. Es kann jedoch ausgewählt werden, auf welchem Server die Daten gelagert werden sollen.

Punkt 2 – Mobile Fähigkeit

Sollen IFCs mobil dargestellt werden können, dann ist Autodesk Forge mit einer durchschnittlichen Renderszeit von ca. 8 Sekunden auf einem Tablet und Mobiltelefon die bessere Wahl. Da bei bimsync diese Zahlen für grosse Modelle bei 26 bzw. 25 Sekunden liegen, können sie produktiv nur sehr schlecht verwendet werden.

Das verwendete Tablet muss mindestens 3GB RAM haben, um grössere IFCs flüssig rendern zu können.

Punkt 3 – Pricing

Ein weiterer wichtiger Punkt ist das Pricing, welches smino direkt mit bimsync verhandeln muss. Bei Autodesk Forge werden die sog. Cloud Credits Angeboten, welche für die Verschiedenen APIs verwendet werden können.

Punkt 4 – Open Source

Weder Autodesk Forge noch bimsync sind open source. Falls in der Zukunft auf open source gewechselt werden soll, stehen der BIMserver und xBIM als mögliche Frameworks zur Verfügung. Beide haben den Vorteil, nicht von einer Firma abhängig zu sein.

Leider stehen für die zwei Open-Source-Lösungen noch zu wenig Funktionalität bzw. Plugins zur Verfügung, als dass sie produktiv verwendet werden könnten. Eine Lösung mit dieser Grundlage wäre mit einem Mehraufwand verbunden, was für smino aktuell wenig attraktiv ist.

Smino's Entscheidung

smino hat sich für Autodesk Forge entschieden, um den Prototypen zu erstellen. Da sich die Frameworks Forge und bimsync nicht gross voneinander unterscheiden, ist ein Wechsel nicht ausgeschlossen.

In Zukunft ist auch eine open source Lösung mit dem BIMserver oder xBIM denkbar, wenn die Angebote in Sachen Funktionalität erweitert werden.

13. Schlussfolgerung




Mit den Erkenntnissen aus dem ersten Teil der Arbeit wurde in einem zweiten Teil ein Softwareprototyp mit den JavaScript-Technologien React und Node.js entwickelt.

Der Prototyp integriert den 3D-Model-Viewer und simuliert das Zusammenspiel zwischen IFC und BCF. Implementiert wurden das Hochladen und Betrachten von Modellen im Forge-Viewer. Zu jedem Modell können Tickets erfasst werden, wobei beispielsweise die Kameraposition und selektierte Objekte mitverwaltet werden. Beim Laden eines Tickets wird der Viewer wieder in dessen ursprünglichen Zustand gesetzt. So können in der Realität Fehler am Modell einfach beschrieben und wieder dargestellt werden. Der Prototyp dient smino als Vorlage für die Erweiterung ihres Produktes.

BIM - smino

My Models

Upload Model: Keine Datei ausgewählt.

Thumbnail	Name	Size	Status
	klein.ifc	3.76mb	<button>Show</button>
	gross.ifc	67.07mb	<button>Show</button>
	mittel.ifc	13.49mb	<button>Show</button>

My Tickets

Thumbnail	Title	Date	Show
	Wände falsch	17.12.18 15:06:42	<button>Show</button>
	Schnitt Gebäude	17.12.18 15:09:28	<button>Show</button>

Forge Viewer



Abbildung 22: Screenshot des Prototypens

14. Ausblick

Der BIM Markt ist noch jung und entwickelt sich täglich weiter. Diese Veränderungen müssen beobachtet werden, um die neuen Technologien und Features erkennen, analysieren, testen und implementieren zu können.

Mit dem Prototyp als Hilfestellung wird smino nun erstmal ihre bestehende Lösung erweitern. Das Team wird auch über die Projektdauer hinaus für Fragen zur Verfügung stehen.

Mögliche Weiterentwicklungen

Für den Prototyp wurden keine BCFs gemäss Building Smart Standard verwendet. Dies ist eine wichtige Implementierung für smino, welche dann auf smino-Issues abgebildet werden kann. (siehe Punkt 11)

Wenn sich Open-Source-Technologien weiterentwickelt haben, kann es für smino interessant sein ihre Lösung auf dem aufzubauen.

Die Validierung von Modellen gegen Regeln oder Referenzmodellen könnte für die Zukunft interessant sein.

III. Software-Prototyp

15. Zweck der Applikation

Der Prototyp soll smino als Inspiration für ihre Integration von BIM Komponenten in ihr Produkt dienen. Mit der Umsetzung des Prototyps soll aufgezeigt werden, wie der von Forge zur Verfügung gestellter 3-D-Viewer auf einer Webseite eingebunden wird. Es soll möglich sein, Modelle hochzuladen und anzuschauen. Zu den Modellen können Tickets zusammen mit Ansichtsinformationen gespeichert und später wieder geladen werden. Ebenfalls soll es möglich sein BCF Dateien zu importieren, welche zu Tickets konvertiert werden.

Im Rahmen der begrenzten Zeit für die Implementation, werden nur die wichtigsten Features beachtet, die für die Interaktion mit dem Viewer benötigt werden.

Herkömmliche Funktionalität die mit dem Verwalten von Dateien und Aufgaben anfällt, wie Editieren, Löschen, Versionieren sowie Exportieren werden bewusst weggelassen. Ebenfalls vernachlässigt wird das unterschiedliche Benutzer Zugriff auf andere Modelle haben müssten. Stattdessen simuliert der Prototyp nur eine Projektumgebung, in welcher alle Benutzer gleichberechtigt sind.

Vielmehr soll der Prototyp ein Gefühl für die Verwendung des Viewers geben. Des Weiteren sollen die Möglichkeiten des Speicherns und Ladens von Ansichten im Viewer aufgezeigt werden. Abschliessend soll die Kompatibilität von BCF Dateien mit dem Forge Viewer aufgezeigt werden.

Mit dem Prototyp wird erhofft, dass smino der Einstieg in die Programmierung und Nutzung der Forge Schnittstellen leichter fällt.

15.1. Features

Folgende Features sollen mit absteigender Priorität umgesetzt werden:

1. IFCs und Viewer:

- 1.1: IFCs auf Webseite hochladen.
- 1.2: IFCs im Viewer darstellen.

2. Prototyp Issue erstellen

- Issue im Viewer erstellen.
- Issue wird im Browser angezeigt.
- Beim Click auf das Issue, wird dies im Viewer im Modell angezeigt.

3. Prototyp BCF

- BCFs hochladen, konvertiert/verarbeitet und in Datenbank speichern.

Abgrenzung BCF

Im Verlauf der Planung des Prototyps hat sich gezeigt, dass die Konvertierung von BCF Dateien sich zeitlich nicht umsetzen lässt. Deshalb wird dieses Feature in den kommenden Kapiteln nicht mehr behandelt.

16. Anforderungsspezifikation

16.1. Use Cases

UC 1	
Name	Modell hochladen
Primary Actor	User
Preconditions	<ul style="list-style-type: none"> Modell ist erstellt worden und Lokal vorhanden
Postconditions	<ul style="list-style-type: none"> Das Modell wird in der Liste «My Models» dargestellt
Main Success Scenario	<ol style="list-style-type: none"> Bei «Upload Model» nach einem Modell suchen Modell auswählen und hochladen
Frequency of Occurrence	Mehrmals pro Session

UC 2	
Name	Modell im Viewer darstellen
Primary Actor	User
Preconditions	<ul style="list-style-type: none"> Viewer muss ein gültiges Token von Forge haben Modell muss in SVF konvertiert worden sein Ein Modell ist in der «My Models»-Liste vorhanden
Postconditions	<ul style="list-style-type: none"> Das Modell wird im Viewer angezeigt
Main Success Scenario	<ol style="list-style-type: none"> Ein Modell aus der Liste auswählen und «Show»-Knopf drücken
Frequency of Occurrence	Mehrmals pro Session

UC 3	
Name	Ticket erstellen
Primary Actor	User
Preconditions	<ul style="list-style-type: none"> UC 2
Postconditions	<ul style="list-style-type: none"> Ticket wird abgespeichert und in der Liste «My Tickets» angezeigt
Main Success Scenario	<ol style="list-style-type: none"> Ein oder mehrere Objekte im Viewer auswählen Name des Tickets eingeben und «Create Ticket»-Knopf drücken.
Extensions	<ol style="list-style-type: none"> A) Zusätzlich können auch eine Schnittanalyse oder Explosionsgrafik ausgewählt und gespeichert werden.
Frequency of Occurrence	Mehrmals pro Session

UC 4	
Name	Ticket im Viewer darstellen
Primary Actor	User
Preconditions	<ul style="list-style-type: none"> Ein Ticket ist in der «My Tickets»-Liste
Postconditions	<ul style="list-style-type: none"> Das Modell wird mit den gespeicherten Eigenschaften im Ticket geladen.
Main Success Scenario	<ol style="list-style-type: none"> Ticket in der «My Tickets»-Liste auswählen und «Show»-Knopf drücken.
Frequency of Occurrence	Oft

16.2. Use Case-Diagramm

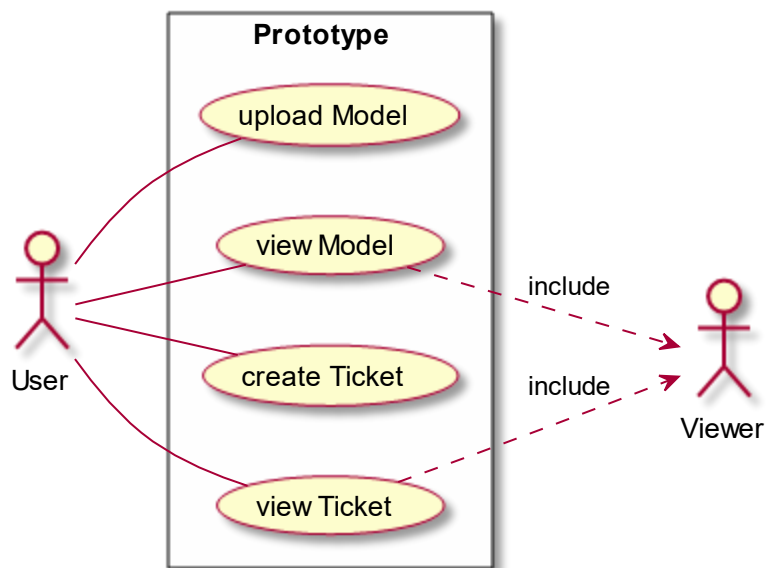


Abbildung 23⁴⁹: Use Case-Diagramm des Prototyps

16.3. Non Functional Requirements

Wie in den Rahmenbedingungen festgehalten gibt es keine besondere NFRs.

⁴⁹ http://www.plantuml.com/plantuml/svg/POz12i8m44NtSufPwg8zW0lf1GGNwZuatngCcJATAYBUtIWjLBhB-_d_mJlzERNXbYZXR6p26jkBSOWARr4oaVCcoiSUEZvF4Oy2z6sut2RmNiN4dXrUnBycLnMlXYw92xoJWBIUVzoBO8hdKDVRKRw9sQsXWEOpht06FaH_XIs5K_v7IsNPqPiQv52k_W00

17. Analyse

17.1. Domainmodell

Das Domainmodell zeigt die wichtigsten Komponenten des Prototyps. Im Viewer kann ein Modell dargestellt werden. Das Modell ist nur visualisierbar, wenn das Feld «progress» den Zustand «complete» angenommen hat. Das bedeutet, dass das Modell erfolgreich transformiert wurde. Zu jedem Modell gibt es eine beliebige Anzahl an Tickets. Jedes dieser Tickets speichert einen früheren Zustand des Modells im Viewer. Durch das Auswählen eines Tickets wird dieser Zustand, welcher unter anderem die Kameraposition und selektierte Elemente enthält, wiederhergestellt. Im Viewer können nur Tickets dargestellt werden, die zum aktuell geladenen Modell gehören.

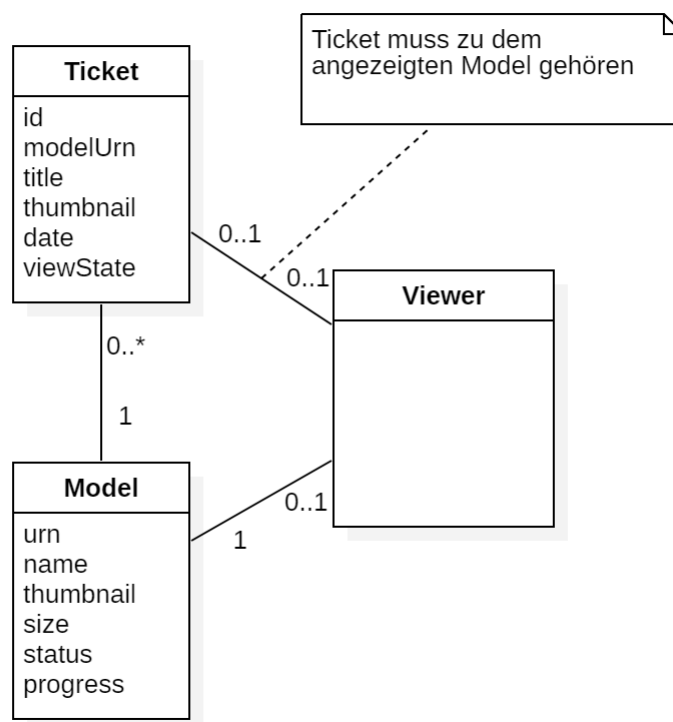


Abbildung 24: Domainmodell

17.2. Datenmodell

Das Daten Datenmodell besteht aus zwei Bereichen. Einerseits werden Modelle, Buckets und App von Forge vorgegeben und persistiert. Andererseits werden Tickets lokal von dem Prototyp verwaltet. Ein Ticket referenziert immer auf ein Model, wobei das Modell nichts von seinen Tickets weiss.

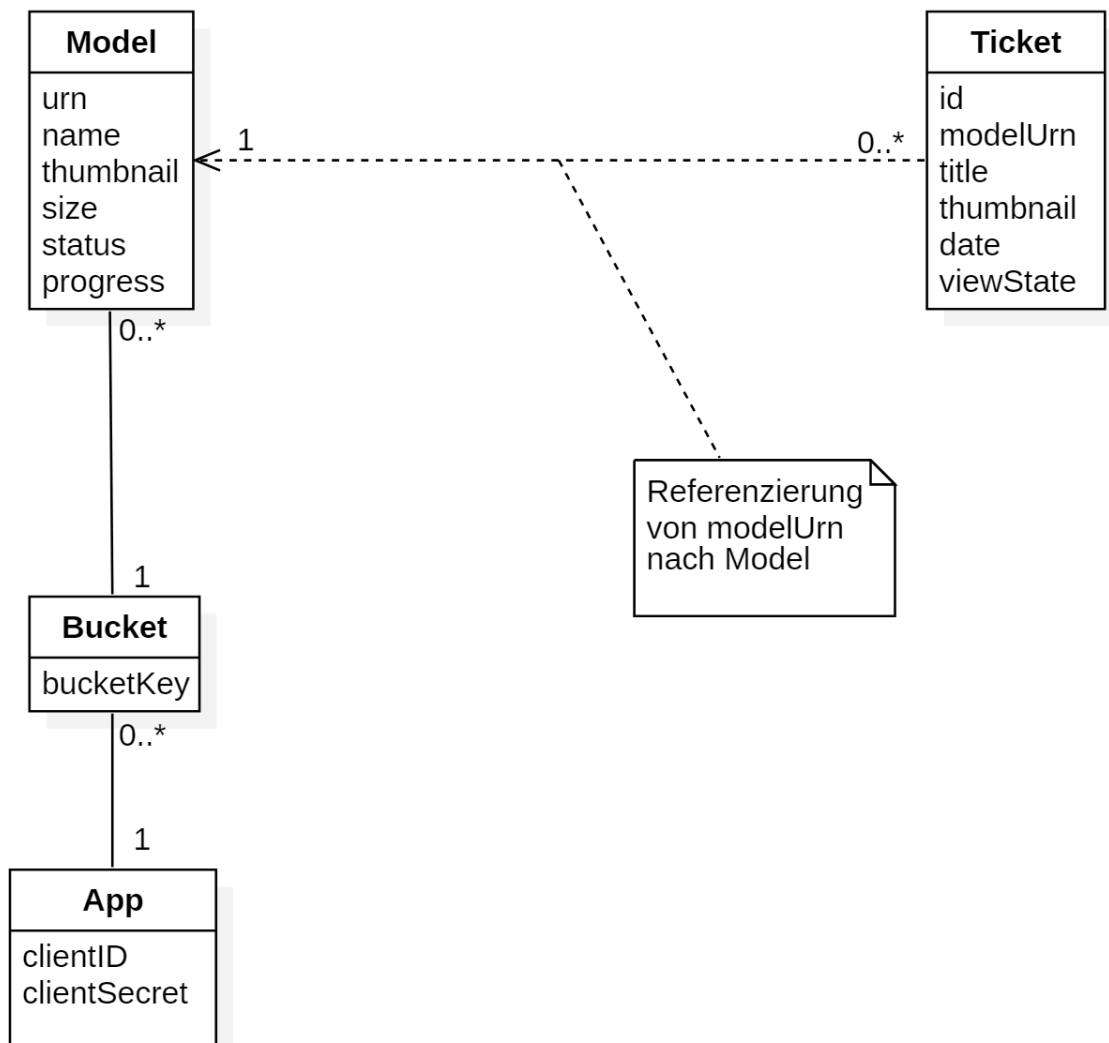


Abbildung 25: Datenmodell

17.3. Sequenzdiagramme

Folgende Abläufe wurden in einem Sequenzdiagramm dargestellt:

- Start der Applikation
- Hochladen eines Modells
- Erstellen eines Tickets

Diese wurden ausgewählt, weil sie kompliziertere Zusammenhänge mit mehreren Komponenten verdeutlichen.

Start der Applikation

Beim Start der Applikation wird davon ausgegangen, dass noch kein Bucket existiert.

Bei (2) wird nach einem Bucket gefragt. Da noch keiner vorhanden ist, wird ein neuer Bucket angelegt.

Danach werden alle Modelle geladen (10). Um zu überprüfen ob ein Modell komplett transformiert

ist, muss für jedes das Manifest (11) geladen werden. Das Abfragen des Thumbnail Links (11) muss

ebenfalls für jedes Modell separat gemacht werden. Wenn die Modelle geladen sind, fragt die React

Komponente nach einem Public Token (18), um Modelle im Viewer darstellen zu können.

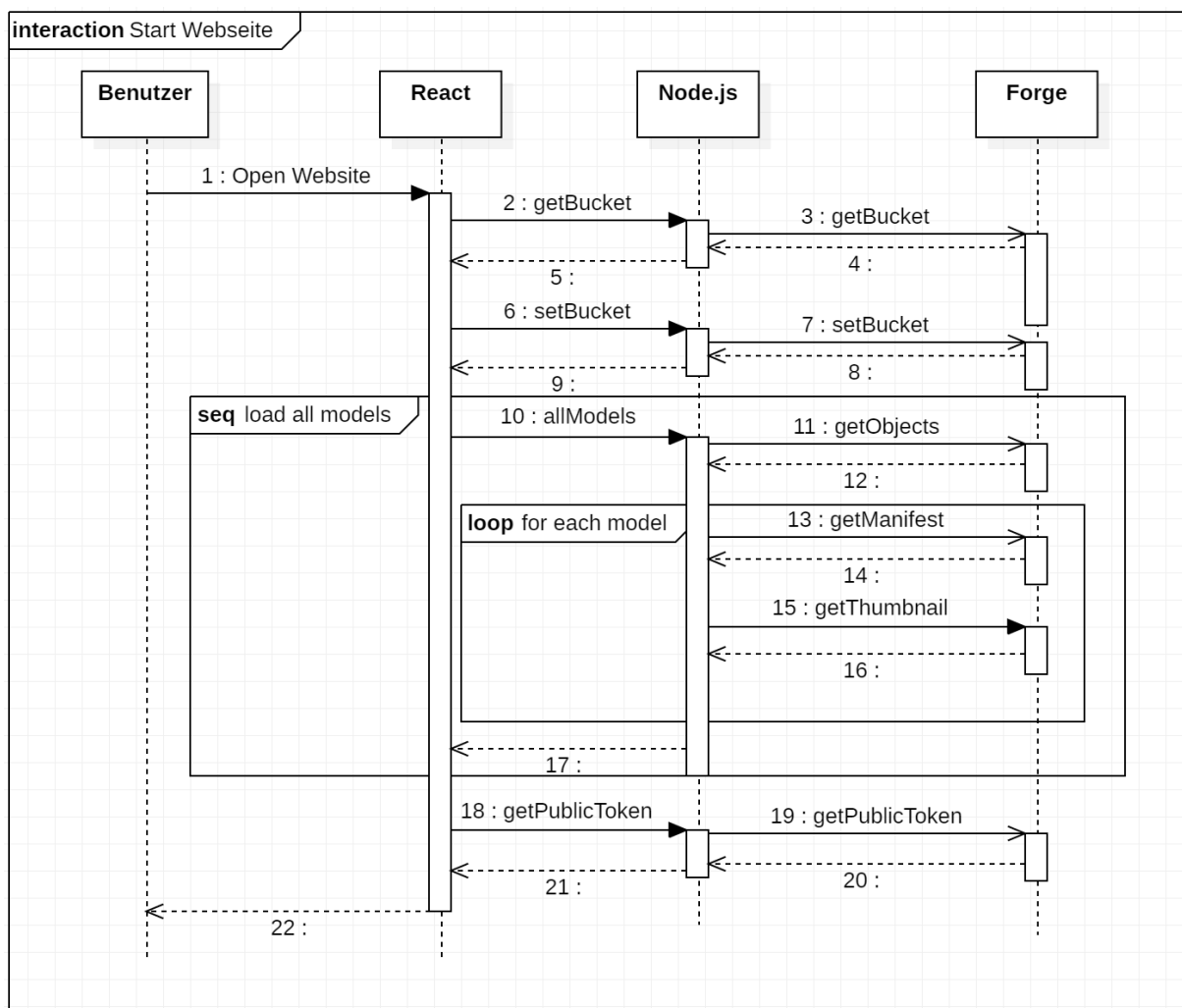


Abbildung 26: Sequenzdiagramm für den Start der Applikation

Hochladen eines Modelles

Um ein Modell hochladen zu können, wird die Datei, welche der Benutzer ausgewählt hat zuerst auf die Node.js Komponente geladen (2). Im Anschluss wird diese an Forge übermittelt (3) und danach der Übersetzungsprozess (5) gestartet.

Schlussendlich werden alle Modelle neu geladen, um den Benutzer eine aktuelle Ansicht anbieten zu können.

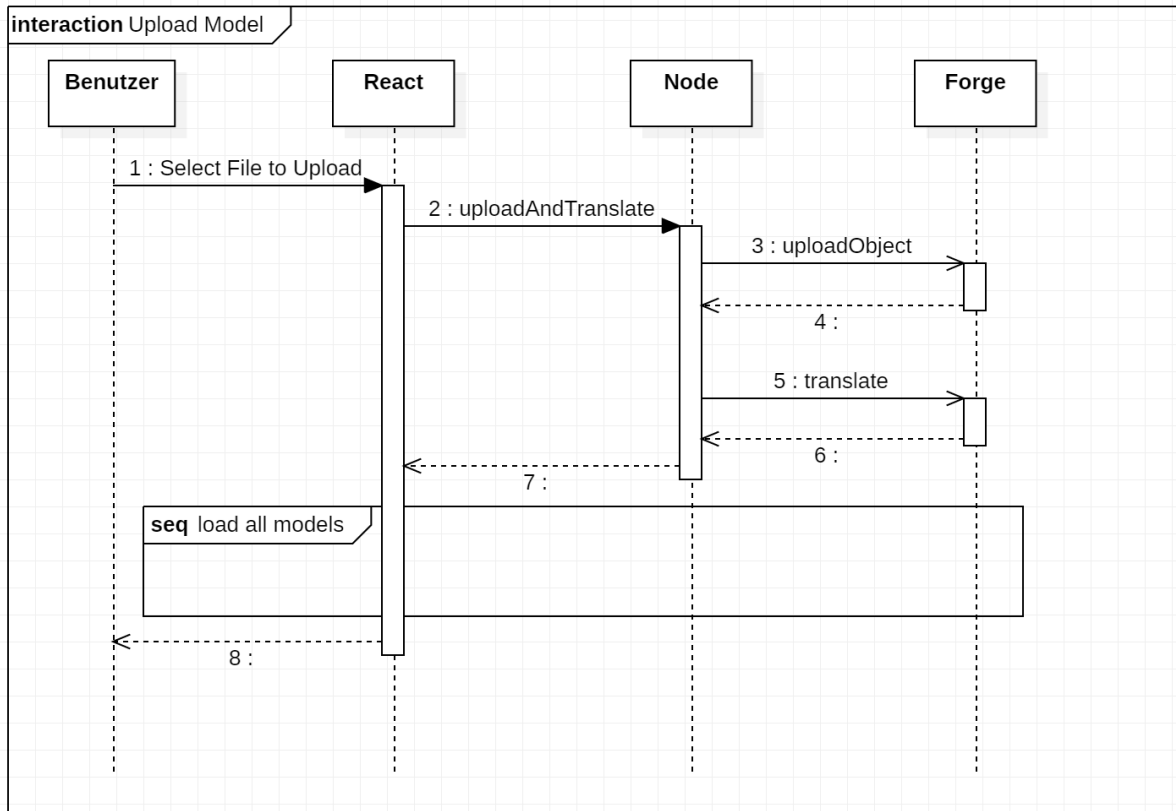


Abbildung 27: Sequenzdiagramm für das Hochladen eines Modells

Erstellen eins Tickets

Um ein neues Ticket zu erstellen, muss zuerst ein Modell ausgewählt werden. Die Darstellung des Modells im Viewer läuft direkt über die Kommunikation mit Forge (2). Die zum Modell gehörenden Tickets werden aus einer Datenbank bei der Node.js Komponente geladen (4).

Wenn der Benutzer ein neues Ticket erstellt, wird der aktuelle Viewer Zustand von Forge ausgelesen (8) und in einem Ticket in der Datenbank gespeichert (10). Danach wird die Ticket Ansicht für den Benutzer aktualisiert (12).

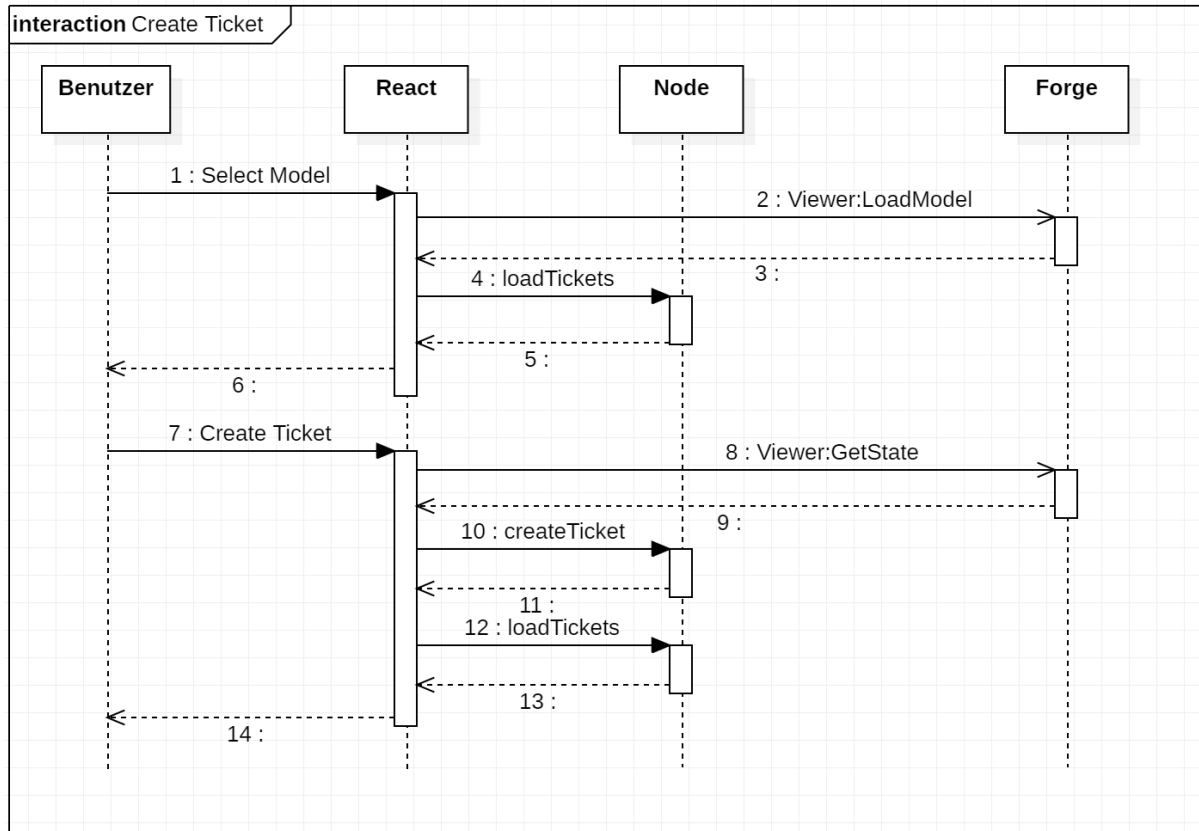


Abbildung 28: Sequenzdiagramm für das Erstellen eines Tickets

17.4. UI Design

Dies ist ein erster Entwurf für die BIM-Implementation in smino. Das Mockup wurde mit Hilfe des Mockup-Tools balsamiq von smino selbst erstellt.

Dashboard

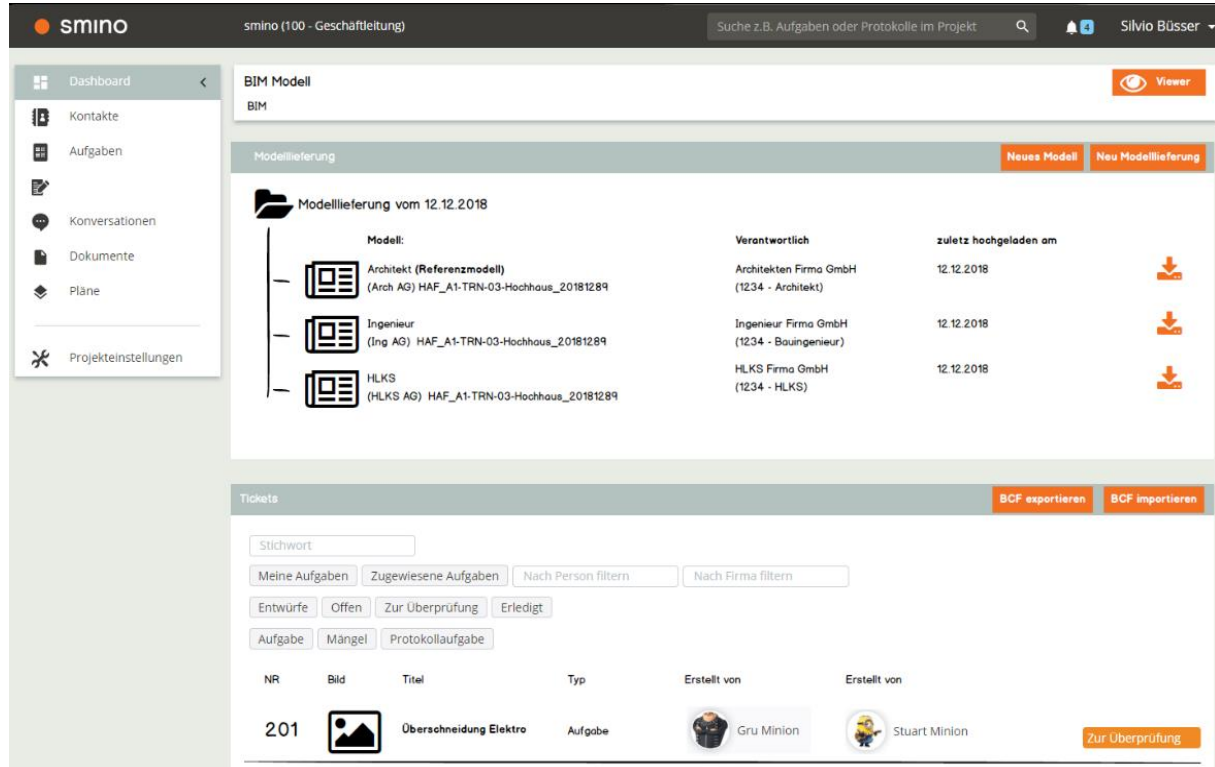


Abbildung 29: Dashboard mit einer Auflistung der Modelle und Tickets

Das Dashboard besteht aus zwei Komponenten: Modelle und Tickets.

Modelle (IFCs)



Abbildung 30: Vergrößerte Ansicht der Modellkomponente

Tickets

Tickets

BCF exportieren

BCF importieren

Meine Aufgaben

Zugewiesene Aufgaben

Nach Person filtern

Nach Firma filtern

Entwürfe

Offen

Zur Überprüfung

Erledigt

Aufgabe

Mängel

Protokollaufgabe

NR	Bild	Titel	Typ	Erstellt von	Erstellt von	
201		Überschneidung Elektro	Aufgabe	Gru Minion	Stuart Minion	<div>Zur Überprüfung</div>
202		Stütze verschieben	Aufgabe	Gru Minion	Stuart Minion	<div>Zur Überprüfung</div>
202		Stütze verschieben	Aufgabe	Bob Minion	Stuart Minion	<div>Zur Überprüfung</div>

Abbildung 31: Vergrösserte Ansicht der Ticketkomponente

Viewer

Der Viewer stellt ein oder mehrere IFCs, inkl. erstellte Tickets und Konversationen dar. Diese Ansicht wird in einem separaten Fenster gezeigt.

Modelllieferung vom 12.12.2018

- ☒ Architekt (Referenzmodell)
- ☒ Ingenieur
- ☒ HLK
- ☐ Strom

Tickets

Bildschirm	Status	Datum
Überschneidung Elektro	Offen	
Stütze verschoben	Offen	
Stütze verschoben	Offen	

Auswahl (2)

Element: 324/jz/34234
Name: TG 123
MetaInfoX: Inhalt MetaInfoX
MetaInfoX: Inhalt MetaInfoX
MetaInfoX: Inhalt MetaInfoX

Element: 324/jz/34234
Name: TG 123
MetaInfoX: Inhalt MetaInfoX
MetaInfoX: Inhalt MetaInfoX
MetaInfoX: Inhalt MetaInfoX

Überschneidung Elektro
Offen

R die tralloff R die tralloff R die tralloff R die tralloff R die tralloff

Konversation

Siehe unten
Mitteilungen

Siehe Br-Global Konversations?

In ALU 10/11/14

Br-Global
Konversation

Kommunikation zwischen:
Frankfurt am Main, Hamburg,
Berlin, Frankfurt am Main,
Hamburg, Berlin, Frankfurt am Main,
Hamburg, Berlin, Frankfurt am Main,
Hamburg, Berlin, Frankfurt am Main,

ALU test ng product management
validation. Particular growth
handling handling user
experience from research contains
from the right full group new case
interface interface experience
handling handling interface
Twitter android ios status series

Abbildung 32: Viewer-Fenster

Prototyp

Für unser Prototyp hat das Design keine Priorität und wird deshalb rudimentär gehalten. Das Design beläuft sich auf 3 Bereiche: Einen Bereich für die Auflistung der Modelle, einen nahezu identischen für die zu den Modellen gehörenden Tickets und die Ansicht für den Viewer mit der Möglichkeit neue Tickets zu erstellen.

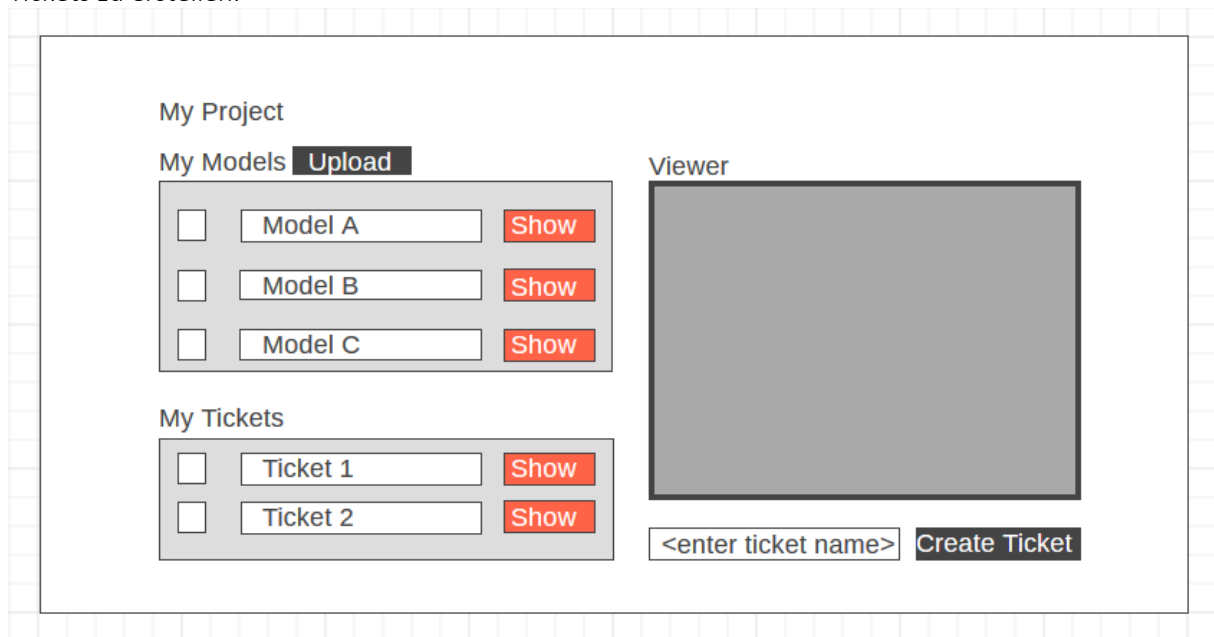


Abbildung 33⁵⁰: UI des Prototypens

⁵⁰ <https://wireframe.cc/PT9Eh9>

18. Architektur

18.1. Deployment-Diagramm

Der Benutzer greift von seinem Browser aus auf ein React Frontend zu. Im Backend ist ein Node.js Server. Dieser kapselt die benötigten API-Calls auf die Autodesk Forge API. Einzig für das Darstellen der View muss die React Komponente direkt auf die Forge Viewer API zugreifen.

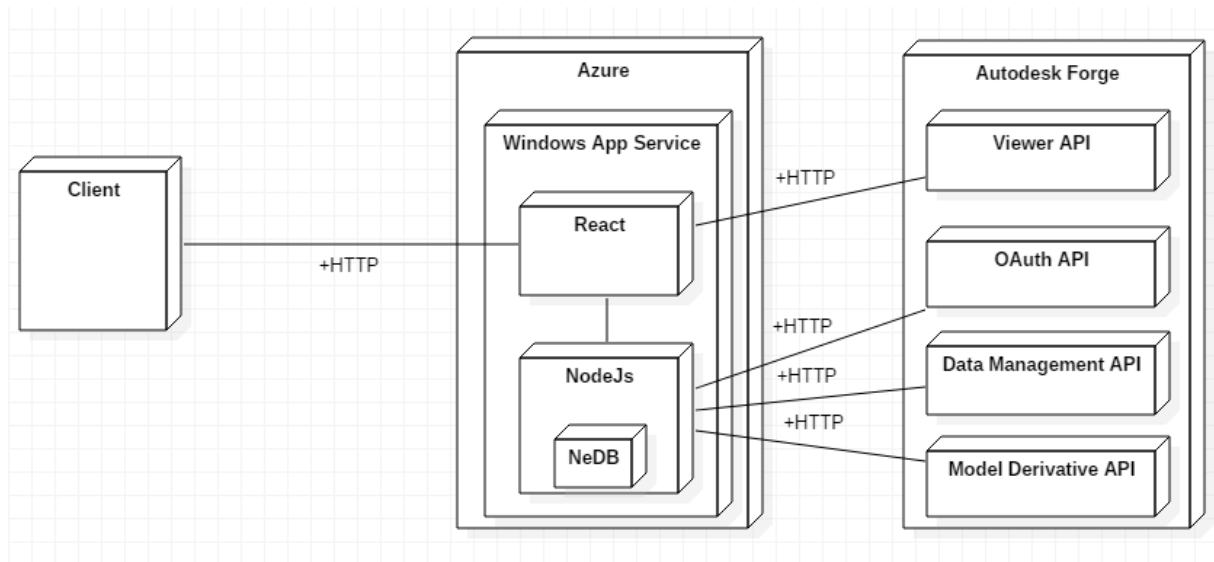


Abbildung 34: Deployment-Diagramm

18.2. Package und Klassendiagramm

Im nachfolgenden Diagramm sind die Pakete, welche im Deployment-Diagramm bereits ersichtlich sind zusammen mit ihren enthaltenen Klassen aufgezeigt. Die Klassen beinhalten ihre wichtigsten Attribute und Methoden, welche für die Kommunikation untereinander benötigt werden.

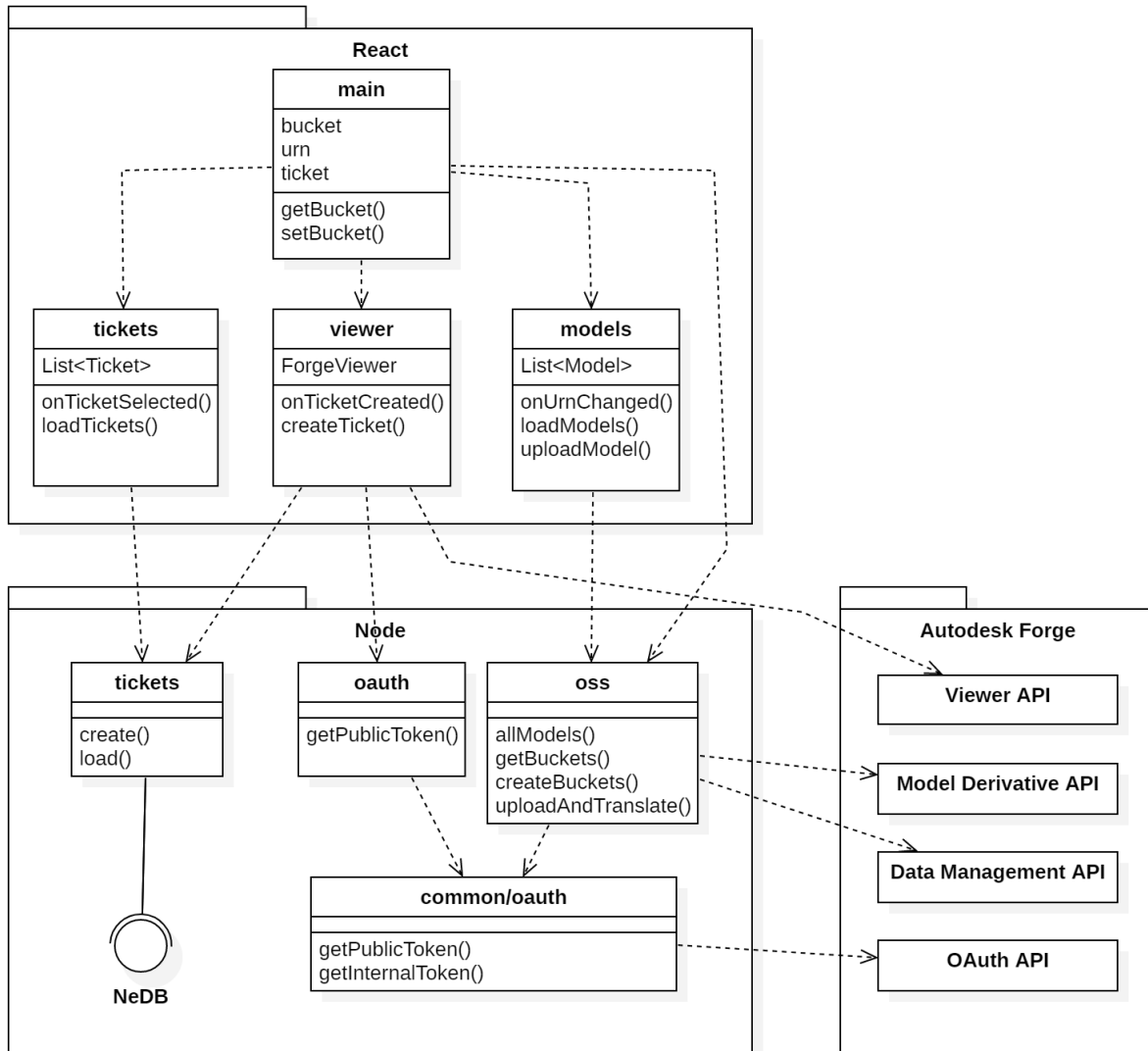


Abbildung 35: Package und Klassendiagramm

18.3. Schnittstellendokumentation

Für den Prototyp existieren 2 Arten von Schnittstellen. Einerseits die interne Schnittstelle zwischen den React und Node.js Komponenten und andererseits eine externe Schnittstelle zu den Forge API.

Interne Schnittstelle von React zu Node.js

Die internen Schnittstellen sind mittels OpenAPI 3.0 und Swagger dokumentiert. Für jeden API Aufruf sind die Request und Response Parameter zusammen mit Beispielen in einer JSON Datei und auf [swaggerhub.com](https://app.swaggerhub.com/apis/HSR-SA/bimsmino.azurewebsites/1.0) hinterlegt. Dadurch die die Anforderungen an und von der Schnittstelle klar ersichtlich. Durch dieses Setup kann auch ohne dem Code Test gegen die Schnittstelle ausgeführt werden.

Referenz Swagger: <https://app.swaggerhub.com/apis/HSR-SA/bimsmino.azurewebsites/1.0>

Oauth		▼
GET	/api/forge/oauth/token	↶
Object Storage Service		▼
GET	/api/forge/oss/allModels	↶
GET	/api/forge/oss/buckets	↶
POST	/api/forge/oss/buckets	↶
POST	/api/forge/oss/uploadAndTranslate	↶
Tickets		▼
GET	/api/tickets/load	↶
POST	/api/tickets/create	↶

Abbildung 36: Darstellung der APIs mit Hilfe von Swagger

Externe Schnittstelle zu Forge

Die Schnittstelle von Forge ist auf ihrer Dokumentationswebseite bereits ausreichend beschrieben. Da die unterschiedlichen APIs jedoch sehr viele Methoden anbieten, sind hier diejenigen herausgelöst, welche im Prototyp aufgerufen werden.

OAuth
Request
<pre>curl -v "https://developer.api.autodesk.com/authentication/v1/authenticate" -X "POST" -H "Content-Type: application/x-www-form-urlencoded" -d "client_id=<MyClientID>&client_secret=<MySecret>&grant_type=client_credentials&scope=data:read%20bucket:create%20bucket:read%20data:write"</pre>
Response
My Access (Bearer) token: eyJhbGciOiJI...

Create Bucket
Request
<pre>curl -v "https://developer.api.autodesk.com/oss/v2/buckets" -X "POST" -H "Content-Type: application/json" -H "Authorization: Bearer <MyToken>" -d "{ \"bucketKey\": \"<MyBucket>\", \"policyKey\": \"transient\" }"</pre>
Response
<pre>HTTP/1.1 200 OK HTTP/1.1 400 Bad Request HTTP/1.1 409 Conflict</pre>

Get Bucket
Request
<pre>curl -v "https://developer.api.autodesk.com/oss/v2/buckets/<MyBucket>/details" -X GET -H "Authorization: Bearer <MyToken> " -H "Content-Type: application/json"</pre>
Response
<pre>{ \"bucketKey\": \"<MyBucket>\", \"bucketOwner\": \"jtwR0A8ZQ6h6bQAhNI5BTowDwQ5PpRx9\", \"createdDate\": 1543749649619, \"permissions\": [{\"authId\": \"jtwR0A8ZQ6h6bQAhNI5BTowDwQ5PpRx9\", \"access\": \"full\"}], \"policyKey\": \"transient\" }</pre>

4 -Upload File (async)
Request
<pre>curl -v "https://developer.api.autodesk.com/oss/v2/buckets/<MyBucket>/objects/<MyModel>" -X "PUT" -H "Authorization: Bearer <MyToken>" -H "Content-Type: application/octet-stream" -H "Content-Length: 308331" -T "<MyModel>"</pre>
Response
<pre>{ "bucketKey" : "<MyBucket>", "objectId" : "urn:adsk.objects:os.object:<MyBucket>/<MyModel>", "objectKey" : "<MyModel>", "sha1" : "2bcd3ea1356cdb1163973175717dee182f25db90", "size" : 308331, "contentType": "application/octet-stream", "location": "https://developer.api.autodesk.com/oss/v2/buckets/<MyBucket>/objects/<MyModel>" }</pre>

Translate File to SVF
Request
<pre>curl -X "POST" -H "Authorization: Bearer <MyToken>" -H "Content-Type: application/json" -v "https://developer.api.autodesk.com/modelderivative/v2/designdata/job" -d "{\"input\": {\"urn\": \"<MyEncodedURN>\"}, \"output\": {\"formats\": [{\"type\": \"svf\", \"views\": [\"2d\", \"3d\"]}]}}"</pre>
Response
<pre>HTTP/1.1 200 OK { "result": "success", "urn": "<MyEncodedURN>", "acceptedJobs": { "output": { "formats": [{ "type": "svf", "views": ["2d", "3d"] }] } } }</pre>

7 – Get Manifest	
Request	
<pre>curl -X "GET" -H "Authorization: Bearer <MyToken>" -v "https://developer.api.autodesk.com/modelderivative/v2/designdata/<MyEncodedURN>/manifest"</pre>	
Response	
<pre>HTTP/1.1 200 OK ... { "type": "manifest", "hasThumbnail": "true", "status": "success", "progress": "complete", "region": "US", "urn": "<MyEncodedURN>", "derivatives": [{ ... }] }</pre>	

Autodesk Forge API Referenzen:

Authentication (OAuth): <https://forge.autodesk.com/en/docs/oauth/v2/reference/http/>

Data Management API: <https://forge.autodesk.com/en/docs/data/v2/reference/http/>

Model Derivative API: <https://forge.autodesk.com/en/docs/model-derivative/v2/reference/http/>

Viewer: <https://forge.autodesk.com/en/docs/viewer/v6/reference/javascript/>

18.4. Verwendete Dritt Komponenten

Für die vereinfachte Kommunikation mit Forge werden zwei Packages eingesetzt um die Integration in die React und Node.js Umgebung zu vereinfachen.

Aus den jeweiligen Beispielprojekten werden Code Snippets direkt oder in angepasster Form verwendet.

React:

Komponente: react-forge-viewer

Version: 0.2.2

Github: <https://github.com/outer-labs/react-forge-viewer>

npm: <https://www.npmjs.com/package/react-forge-viewer>

Lizenz: MIT

Node.js:

Komponente: forge-apis

Version: 0.4.4

Github: <https://github.com/Autodesk-Forge/forge-api-nodejs-client>

npm: <https://www.npmjs.com/package/forge-apis>

Lizenz: MIT

19. Implementation

19.1. Webapplikation




Der Prototyp ist zu testzwecken auf der folgenden Webseite abrufbar:

<https://bimsmino.azurewebsites.net/>



BIM - smino

My Models

Upload Model: Keine Datei ausgewählt.

Thumbnail	Name	Size	Status
	klein.ifc	3.76mb	<input type="button" value="Show"/>
	gross.ifc	67.07mb	<input type="button" value="Show"/>
	mittel.ifc	13.49mb	<input type="button" value="Show"/>

My Tickets

Thumbnail	Title	Date	Show
	Wände falsch	17.12.18 15:06:42	<input type="button" value="Show"/>
	Schnitt Gebäude	17.12.18 15:09:28	<input type="button" value="Show"/>

Forge Viewer



Abbildung 37 Webapplikations-Prototyp: Screenshot mit Modellen und Tickets (links) sowie mit 3D-Modell im Forge-Viewer (rechts)

19.2. Quellcodeverzeichnis

GitHub Verzeichnis: https://github.com/JonBonSnovi/SA_HS18_BIMsmino

Das Projekt besteht aus 5121 Zeilen Code.

Hier eine Übersicht der Metriken des GitHub Repositorys:

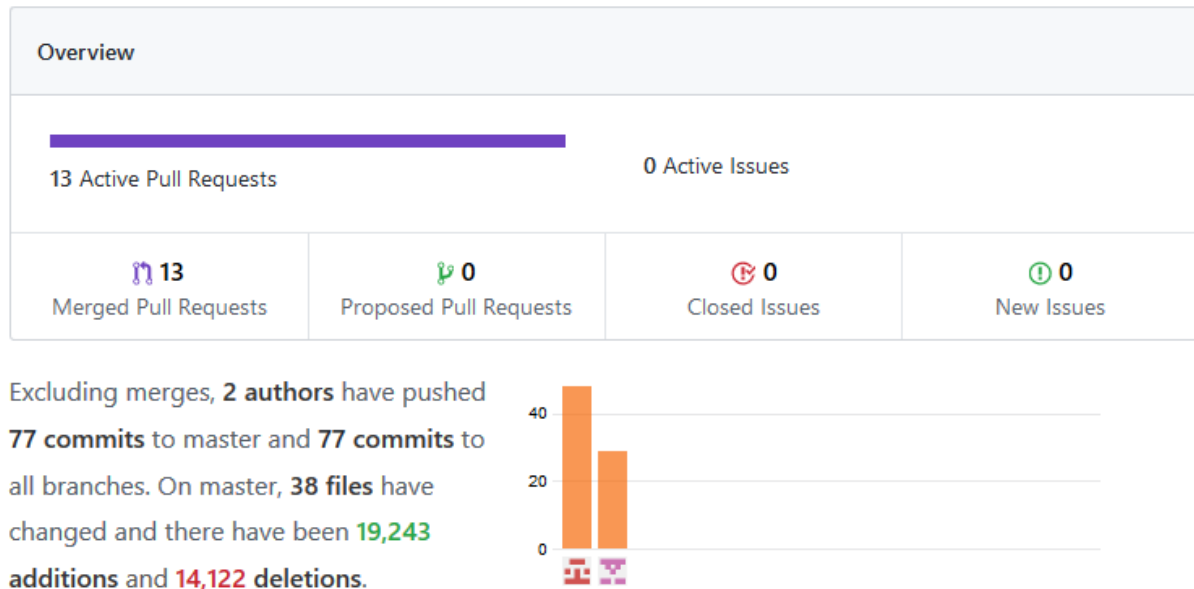


Abbildung 38: Metriken aus dem GitHub-Repository

19.3. Inbetriebnahme

Der Quellcode kann von dem entsprechenden GitHub Verzeichnis heruntergeladen werden. Danach müssen die nötigen Pakete mit dem «npm» Installer hinzugefügt werden. Optimalerweise wird dazu der Quellcode mit einer gängigen JavaScript Entwicklungsumgebung wie WebStorm oder Visual Studio Code geöffnet. Danach kann die Installation der Pakete im Terminal mit dem Befehl: «npm i» gestartet werden. Um das Projekt im Entwicklungsmodus zu starten kann im Terminal «npm run dev» eingegeben werden. Dies startet den Prototypen auf der Adresse <http://localhost:3000/> im Browser.

IV. Anhang

20. Glossar

Building Information Modeling (BIM)

Ein Prozess zur Unterstützung der verschiedenen Lebenszyklen (Planung, Bau, Unterhalt) eines Baus.

Industry Foundation Classes (IFC)

Ein allgemeines Datenmodell, das den Austausch von Daten zwischen verschiedenen proprietären Software-Anwendungen ermöglicht.

Open BIM Collaboration Format (BCF)

Eine Datenschnittstelle zum vereinfachten Austausch von Informationen während des Arbeitsprozesses zwischen verschiedenen Softwareprodukten basierend auf dem IFC-Austauschformat.

Es ermöglicht eine modellbasierte Kommunikation zwischen verschiedenen Anwendern und informiert über Status, Ort, Blickrichtung, Bauteil, Bemerkung, Anwender und Zeitpunkt im IFC Datenmodell.

Topic

Ein Topic ist die Beschreibung eines Problems in einem BCF.

Clashes

Sind Fehler in einem oder mehreren 3D-Modellen. Sie werden von einem Model Checker generiert und haben eine Gewichtung.

Beispiel:

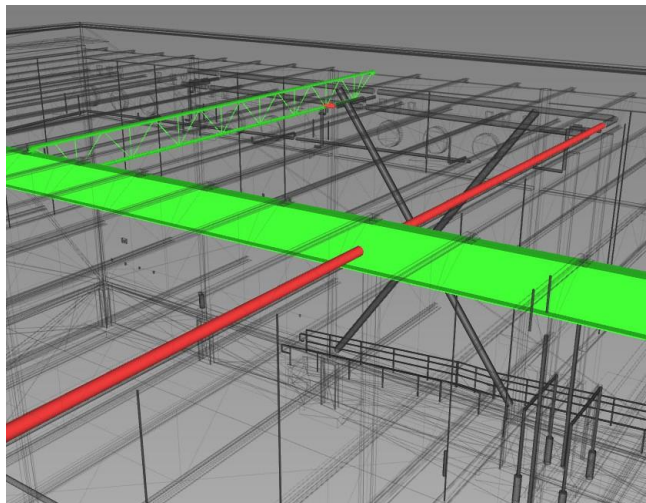


Abbildung 39⁵¹: Ansicht eines Clashes (The Webber/Smith Group, kein Datum)

Regel-Set

Ein Regel-Set definiert, nach welchen Regeln Modelle kontrolliert und Clashes angezeigt werden sollen.

⁵¹ http://www.webbersmith.com/wp-content/uploads/2015/10/Clash_2-845x650.jpg

21. Verzeichnisse

21.1. Quellenverzeichnis

Accenture. (2016). Abgerufen am 21. 12 2018 von https://www.accenture.com/_acnmedia/PDF-5/Accenture-Top500-DE-Executive-Summary-Final-Web.pdf

Autodesk. (kein Datum). Abgerufen am 21. 12 2018 von <https://knowledge.autodesk.com/support/revit-products>

Autodesk. (kein Datum). *BIM 360*. Abgerufen am 21. 12 2018 von <https://info.bim360.autodesk.com/bim360-api-build-integration>

Bauen Digital Schweiz. (kein Datum). Abgerufen am 21. 12 2018 von <https://bauen-digital.ch/de/ueber-uns/>

Bauen Digital Schweiz. (7 2018). Abgerufen am 21. 12 2018 von <https://bauen-digital.ch/de/produkte/stufenplan/>

buildingSMART. (kein Datum). Abgerufen am 21. 12 2018 von <https://www.buildingsmart.org/standards>

buildingSMART. (kein Datum). Abgerufen am 21. 12 2018 von <https://www.buildingsmart.de/bim-knowhow/standards>

buildingSMART. (10 2018). Abgerufen am 21. 12 2018

International Organization for Standardization. (kein Datum). Abgerufen am 21. 12 2018 von <https://www.iso.org/standard/51622.html>

The Webber/Smith Group. (kein Datum). Abgerufen am 21. 12 2018 von http://www.webbersmith.com/wp-content/uploads/2015/10/Clash_2-845x650.jpg

21.2. Abbildungsverzeichnis

Abbildung 1: BIM-Stufenplan von Bauen Digital Schweiz; diese Arbeit befasst sich mit Stufe 1 und 2. ... 4	4
Abbildung 2: BIM-Workflow mit IFC- und BCF-Dateien und Plattformen (Schritte 1-7)	5
Abbildung 3: Darstellung des möglichen smino-Workflows	6
Abbildung 4: Webapplikations-Prototyp: Screenshot mit Modellen und Tickets (links) sowie mit 3D-Modell im Forge-Viewer (rechts)	7
Abbildung 5: Meilensteine des Projekts.....	16
Abbildung 6: Zeitauswertung der Arbeitsaufteilung	22
Abbildung 7: Zeitauswertung von Patrick Faic	23
Abbildung 8: Zeitauswertung von Benjamin Moosmann	23
Abbildung 9: Digitalisierungsgrad nach Industrie-Cluster (Accenture, 2016)	27
Abbildung 10: Stufenplan von Bauen Digital Schweiz	28
Abbildung 11: Darstellung des BIM Workflows.....	29
Abbildung 12: Eingesetzte Software von Baufirmen.....	30
Abbildung 13: Schematischer BIM Workflow, bei dem alle IFC-Modell ausgetauscht werden	33
Abbildung 14: Ansicht eines Clashes (The Webber/Smith Group, kein Datum).....	34
Abbildung 15: Beispiel eines BCF-Topics.....	34
Abbildung 16: Schematischer BIM-Workflow, bei dem IFC-Modelle anhand von BCFs aktualisiert werden.....	35
Abbildung 17: Datenstruktur eines BCF	36
Abbildung 18: Ansicht eines Modells, in welchem 3 Objekte ausgewählt wurden.	40
Abbildung 19: Darstellung des möglichen smino-Workflows	42
Abbildung 20: Ausschnitt aus dem Git-Repository von xBIM.....	50
Abbildung 21: Funktionsweise der Model Derivative API	51

Abbildung 22: Screenshot des Prototypens.....	59
Abbildung 23: Use Case-Diagramm des Prototyps.....	64
Abbildung 24: Domainmodell	65
Abbildung 25: Datenmodell	66
Abbildung 26: Sequenzdiagramm für den Start der Applikation	67
Abbildung 27: Sequenzdiagramm für das Hochladen eines Modells	68
Abbildung 28: Sequenzdiagramm für das Erstellen eines Tickets	69
Abbildung 29: Dashboard mit einer Auflistung der Modelle und Tickets	70
Abbildung 30: Vergrößerte Ansicht der Modellkomponente	70
Abbildung 31: Vergrößerte Ansicht der Ticketkomponente	71
Abbildung 32: Viewer-Fenster	71
Abbildung 33: UI des Prototypens	72
Abbildung 34: Deployment-Diagramm	73
Abbildung 35: Package und Klassendiagramm.....	74
Abbildung 36: Darstellung der APIs mit Hilfe von Swagger	75
Abbildung 37 Webapplikations-Prototyp: Screenshot mit Modellen und Tickets (links) sowie mit 3D-Modell im Forge-Viewer (rechts)	79
Abbildung 38: Metriken aus dem GitHub-Repository	80
Abbildung 39: Ansicht eines Clashes (The Webber/Smith Group, kein Datum).....	82

21.3. Tabellenverzeichnis

Tabelle 1: Risikotabelle	24
Tabelle 2: Abbildung von smino Issues auf BCF Topics	44
Tabelle 3: Übersicht der getesteten Plattformen	47
Tabelle 4: Viewer-Performance-Vergleich	56
Tabelle 5: Nutzwertanalyse.....	57

21.4. Formelverzeichnis

Formel 1: Formel zur Notenberechnung	57
--	----

22. Persönliche Berichte

Patrick Faic

Das Arbeiten an diesem Projekt war etwas Besonderes, insofern wir einen Einblick in eine uns völlig fremde Branche erhielten. Der Einstieg wurde uns jedoch von unserem externen Partner und den Interviewpartnern sehr erleichtert und wir erhielten schnell einen kleinen Einblick in die komplexe Baubranche. Es war eine sehr interessante Anfangsphase, welcher mich lehrte, mich in kürzester Zeit in ein neues Thema einzuarbeiten.

Ein weiterer spannender Aspekt der Arbeit war, dass wir mit einem Startup zusammen Arbeiten durften, welches sich versuchte in einen relativ neuen Markt einzubringen. Wir durften die Entscheidungstreffen von CEO und CTO hautnah mitterleben und Sie mit unserer Recherche unterstützen. Dieser Prozess gab uns die Chance zu sehen, welche Faktoren für das Startup von hoher Priorität sind und welche nicht.

Nach den ersten Wochen Analyse wurde eine Phase eingeleitet, welche sich nur schwer gestalten liess. Die Rede ist von der Plattformevaluation. Tagelanges lesen von Dokumentationen und APIs gehörten leider nicht zu den Höhepunkten dieser Arbeit. Die Tatsache, dass sich diese Phase auch noch über geschlagene 4 Wochen erstreckte, dämpfte die Motivation der Teammitgliedern spürbar. Die aufgewendete Zeit hätte ohne Probleme halbiert werden können, wenn das Team die Lage besser kommuniziert hätte. Die eingesparte Zeit wäre uns in der letzten Phase sehr zugute gekommen.

Auch wenn die letzte Phase nur zwei Wochen dauerte, war sie sehr produktiv, wenn auch stressig. Wir haben in kürzester Zeit einen lauffähigen Prototyp erstellt, welcher die wichtigsten Aspekte der Arbeit aufzeigen konnte. Insbesondere diese letzte Phase hat mir gezeigt, wie reibungslos die Zusammenarbeit im Team funktioniert.

Benjamin Moosmann

Wie habe ich mir das Projekt Studienarbeit vorgestellt?

Sicher nicht so wie es Schlussendlich abgelaufen ist. Zwar hätte ich durch meine IDPA in der Lehre, mehrere Jahre Berufserfahrung, Challenge-Projekt und Engineering-Projekt an der HSR bestens darauf vorbereitet sein müssen, jedoch war es wieder eine einzigartige Erfahrung.

Der normale Arbeitstag in grossen Projektteams hat sich sehr unterschiedlich zur Studienarbeit abgespielt. Die IDPA fand in einem sehr kurzen Zeitraum von 10 Tagen statt, war stark strukturiert und ist für mich schon sehr lange her. Bei dem Challenge-Projekt musste in einem grossen Team nur ein Prototyp und eine Business Idee erstellt werden. Das Engineering-Projekt hatte noch am meisten Ähnlichkeit mit der Studienarbeit, dort wurden wir aber so stark bei der Hand genommen, dass es sich trotzdem komplett anders angefühlt hat.

Der grösste Unterschied zu den anderen bewerteten Abgaben war, dass hier ein richtiger Auftraggeber mit einer Idee aber noch nicht vollends ausgereiften Anforderungen mit an Bord war.

Das Ausbreiten der Problemstellung und Lösungsstrategien zusammen mit dem Startup smino war definitiv auch die interessanteste Komponente der Arbeit. Die Kommunikation mit und Hilfeleistungen von ihnen war immer hervorragend und hat sehr viel Spass gemacht.

Auch spannend war das Einarbeiten in die für mich völlig unbekannte aber komplexe Domäne der Baubranche, Zusammen mit der noch eher unbekannten Technologie BIM. Von Woche zu Woche wurden die Begriffe und Konzepte klarer und verständlicher.

Mit den gesammelten Erkenntnissen haben wir gemeinsam eine Strategie für smino entwickelt und daraus das Vorgehen für einen möglichen Prototypen abgeleitet.

In der nächsten Phase haben wir unterschiedliche Anbieter und deren 3D Model-Viewer evaluiert. Dies hat sich als schwieriger als gedacht herausgestellt. So sind uns nachträglich neue Anbieter aufgefallen, nachdem wir uns anfänglich nur auf Open Source Lösungen fokussiert haben. Dadurch hat sich die Phase unnötig verlängert und etwas an unserer Motivation gekratzt.

Glücklicherweise konnten wir die kleine Down-Phase überwinden und mit vollem Elan einen etwas abgespeckten aber funktionierenden Prototypen auf die Beine stellen. Dies obwohl ich nahezu keine Erfahrung in den JavaScript Technologien React und Node.js hatte.

Durch einen Endspurt ist es uns aus meiner Sicht gelungen das Projekt erfolgreich abzuschliessen und smino aufzuzeigen wie sie BIM für ihre Ziele nutzen können.

23. Danksagung

Wir möchten uns bei allen Beteiligten, die uns bei dieser Arbeit unterstützt haben, herzlichst bedanken.

Vielen Dank an die Firmen, welche sich die Zeit genommen haben, uns einen Einblick in ihre Branche und Arbeitsprozesse zu geben und für Fragen zur Verfügung gestanden sind. Dies hat uns beim Einstieg in die BIM-Welt sehr unterstützt.

Ein grosses Dankeschön an smino und unseren Betreuer Prof. Stefan Keller, dass sie uns die Zusammenarbeit ermöglicht haben und immer für Fragen zur Verfügung standen. Danke auch für die konstruktiven Meetings, für die hohe Flexibilität und schnelle Kommunikation.

Wir wünschen smino eine erfolgreiche Implementation und möchten uns für die großartige Zusammenarbeit bei ihnen bedanken.

24. Aufgabenstellung

Analyse und Integration von BIM Technologien für smino

- Studienarbeit im Herbstsemester 2018/19 Informatik
- Autoren: Patrick Faic und Benjamin Moosmann
- Betreuer: Prof. Stefan Keller, HSR Rapperswil
- Industriepartner: smino, Rapperswil (SG)

Aufgabenstellung

In der Baubranche hat die Digitalisierung noch viel Entwicklungspotential. Um diese voranzutreiben, ist bei Baufirmen ein Akronym in aller Munde: BIM (Building Information Modelling). Mit der Hilfe von BIM erhoffen sich die Betriebe eine Effizienzsteigerung zu erzielen und die Zusammenarbeit an Projekten erheblich zu vereinfachen.

Die Entwicklung des vielversprechenden Markts für BIM basierte Lösungen ist in vollem Gange. Deswegen möchte das Startup BBC Systems AG evaluieren, mit was für einem Produkt sie sich am besten im Markt positionieren können. Aktuell betreiben und entwickeln sie die Webapplikation smino. Die Plattform erleichtert die Kommunikation bei der Realisierungsphase von Bauvorhaben. Es ist wichtig, dass smino eine attraktive Lösung für die Planungsphase von Gebäuden anbietet. Dies gelingt am besten durch die Integration von BIM-Standards.

Um eine geeignete Lösung finden zu können, will diese Arbeit smino mit folgenden Zielen unterstützen:

Teilziel 1: Recherche (1-7 Wochen):

- Was ist BIM genau? Wie sehen die Standards aus? Was ist die Technologie dahinter?
- Wie sieht die Arbeitsmethodik mit BIM aus? Wie lässt sich diese auf smino übertragen?
- Welche Komponenten/Libraries gibt es, um BIM Modelle darzustellen, manipulieren, auszulesen die smino verwenden kann?
- Könnte smino als BIM Hub funktionieren? Das Modell würde hier in der Cloud gehostet. Macht das für smino Sinn? Gibt es Verlierer dabei? Wer könnten allenfalls Partner sein?
- Wie sieht der Aufbau eines BCFs aus und wie kann dieses auf ein smino-Issue abgebildet werden?

Teilziel 2: Feasibility-Prototyp und Evaluation

Framework-Evaluation:

- Implementierbarer 3D-Viewer finden und dessen Performance für PC, Tablet und Mobile testen.
- Preis/Leistung des Frameworks evaluieren.
- Entscheidungshilfe für smino präsentieren.

Der Feasibility-Prototyp beinhaltet folgende Funktionalitäten:

- Import eines IFCs und Darstellung in einer Liste.
- IFCs können versioniert und verwaltet werden. (optional)
- IFCs können ausgewählt und in einem 3D-Viewer dargestellt werden.

Der Viewer-Prototyp versucht folgende Anforderungen zu plausibilisieren:

- Es können Kamera-Positionen smino-extern über API angesteuert werden.
- Es können Objekte smino-extern über API selektiert und fokussiert werden.

- Es können Kamera-Positionen ausgelesen werden (inkl. Thumbnail).
- Es können selektierte Objekte als Referenzen ausgelesen werden (inkl. Thumbnail).
- Zusätzliche Informationen an Koordinaten/Objekte anzeigen (z.B. Marker oder Text).

Als Erkenntnis aus dem Prototyp geht ein Vorschlag hervor zur Integration dieser BIM-Komponente in die smino Webapplikation.

Abgrenzung: Die Arbeit untersucht CAD-Modelle und hat nicht im Fokus, andere BIM-Aspekte zu untersuchen wie beispielsweise den Austausch von Devis, Arbeitsplanungen und auch keine Integration von Geodaten (Terrain, Energie, Ver- und Entsorgung).

Lieferobjekte

4. Dokumentation, inkl. Textabstract (deutsch), Management Summary (deutsch), technischer Bericht und Software Engineering-Projekt (deutsch); Anhänge (Literaturverzeichnis, Inhalt).
5. Software (englisch).
6. Die vom Studiengang geforderten bzw. empfohlenen Lieferobjekte: Poster (nur digital), Broschüren-Abstract, kein Kurzvideo.

Vorgaben/Rahmenbedingungen

- Infrastruktur: Windows Azure, .Net, Java Script, React, BIM-Tools.
- Wo sinnvoll moderne SW-Entwicklung (u.a. mit Unit Testing, Repositories/Versioning, kontinuierliche Integration, Docker).
- Nichtfunktionale Anforderungen: keine besonderen.

Vorgehen und Arbeitsweise: Die Studierenden wählen nach Rücksprache ein Vorgehensmodell zur Softwareentwicklung. Es gibt wöchentliche Meetings mit vorbereiteten Unterlagen; wobei Ausnahmen vereinbart werden können.

Dokumentation

Zur Dokumentation (vgl. auch Lieferobjekte oben):

- Die Abgabe ist so zu gliedern, dass die obigen Inhalte klar erkenntlich und auffindbar sind (einheitliche Nummerierung).
- Die Zitate sind zu kennzeichnen, die Quelle ist anzugeben.
- Verwendete Dokumente und Literatur sind in einem Literaturverzeichnis aufzuführen (nicht ausschliesslich Wikipedia-Links auflisten).
- Dokumentation des Projektverlaufes, Planung etc.
- Weitere Dokumente (z.B. Kurzbeschreibung, Eigenständigkeitserklärung, Nutzungsrechte) gemäss Vorgaben des Studiengangs und Absprache mit dem Betreuer.

Form der Dokumentation zuhanden Betreuer (Studiengang siehe separate Instruktionen):

- Bericht gebunden (1 Ex.)
- Alle Dokumente und Quellen der erstellten Software auf USB-Stick.

Bewertung

Es gelten die üblichen Regelungen zum Ablauf und zur Bewertung der Studienarbeit des Studiengangs Informatik mit besonderem Gewicht auf moderne Softwareentwicklung wie folgt:

- Projektorganisation (Gewichtung ca. 1/5)

- Bericht, Gliederung, Sprache (Gewichtung ca. 1/5)
- Inhalt inkl. Code (Gewichtung ca. 2/5)
- Gesamteindruck inkl. Kommunikation mit Industriepartner (Gewichtung ca. 1/5).

Ein wichtiger Bestandteil der Arbeit ist, dass eine lauffähige, getestete Software abgeliefert wird (inkl. Softwaredokumentation).

Weitere Beteiligte

-