Transferdokumentation

Cloud-Visualisierung

Durchführung Cloud-basierter Simulationsverfahren mittels einer Simulationsplattform

CLOUDMALL
BADEN-WÜRTTEMBERG

Baden-Württemberg
MINISTERIUM FÜR WIRTSCHAFT, ARBEIT UND WOHNUNGSBAU

Cloud Mall Baden-Württemberg

TRANSFERDOKUMENTATION FÜR DEN PRAXISPILOTEN „CLOW-VIS“

Cloud-fähiger Workflow zur Visualisierung von Simulationsdaten

Öffentliche Version vom 27. Juni 2021

Beteiligte Partner

- Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO
- Falquez, Pantle und Pritz Gbr
- Fraunhofer-Institut für Graphische Datenverarbeitung IGD

Autoren

- Iris Pantle (Falquez, Pantle und Pritz GbR)
- Carlos Falquez (Falquez, Pantle und Pritz GbR)
- Christian Altenhofen (Fraunhofer IGD)
- Maxim Redkin (Fraunhofer IGD)
- Damian Kutzias (Fraunhofer IAO)
- Dimitri Evcenko (Fraunhofer IAO)

Lizenz

Das Werk „TRANSFERDOKUMENTATION FÜR DEN PRAXISPILOTEN CLOW-VIS“ steht unter der Lizenz Creative Commons Namensnennung – Nicht-kommerziell – Keine Bearbeitung 4.0 (CC BY-NC-ND 4.0). Details zur Lizenz: https://creativecommons.org/licenses/by-nc-nd/4.0/.

Cover-Foto

© campaign_creators / unsplash.com und Falquez, Pantle und Pritz GbR

Inhalt

1 Management Summary 4
2 Einführung 5
2.1 Ausgangssituation und Motivation 5
2.2 Ziele und Nutzen für die Zielgruppen 7
3 Projektrahmen 9
3.1 Konsortium und Rollen 9
3.1.1 Falquez, Pantle und Pritz GbR (FPPG) 9
3.1.2 Fraunhofer-Institut für Graphische Datenverarbeitung IGD 9
3.1.3 Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO 10
3.2 Notwendige Ressourcen sowie Kompetenzen 10
4 Lösungsbeschreibung 11
4.1 Anforderungen 11
4.1.1 IST-Zustand 11
4.1.2 SOLL-Zustand 13
4.2 Graphische Benutzeroberfläche 15
4.3 Geschäftsmodelle 18
4.4 Prototyp 18
4.5 Herausforderungen bei der Umsetzung 20
4.6 Erfahrungen von Endanwendern 21
5 Integration und Kooperation zwischen den beteiligten Unternehmen 22
5.1 Organisatorisches 22
5.2 Technisches 22
5.3 Strategisches 22
5.4 Rechtliches 22
6 Resümee 23
6.1 Rolle der Cloud 23
6.2 Lessons Learned 23
6.3 Ausblick 24
7 CMBW-Projektdarstellung 25
8 Kontakt 26

1 Management Summary

In diesem Praxispiloten wurde im Projektdurchführungszeitraum mit Start im November ein Jahr lang die Integration eines vom Fraunhofer-Institut für Graphische Datenverarbeitung IGD entwickelten Remote-Visualisierers in die von der Falquez, Pantle und Pritz GbR (FPPG) entwickelte Nuberisim-Plattform durchgeführt.

Die Nuberisim-Plattform ermöglicht es, u. a. hochauflösende Simulationen in der Cloud zu berechnen und die Simulationsergebnisse dort zu speichern. Möchte man diese Simulationen visualisieren, so müssten die Simulationsergebnisse zunächst von den üblicherweise zum Einsatz kommenden Hochleistungsrechnern bzw. Cloud-Ressourcen heruntergeladen werden. Die dabei entstehenden Datentransferkosten sind nicht zu unterschätzen. Darüber hinaus ist es vorteilhaft für die Visualisierung, die im Allgemeinen leistungsfähigeren Hochleistungsrechner oder andere Cloud-Ressourcen zu nutzen.

Vor diesem Hintergrund erscheint das Verbleiben der Ergebnisse und die damit verbundene Visualisierung derselben direkt in der Cloud attraktiv. Um dies zur ermöglichen, wird der vom IGD entwickelte Remote-Visualisierer in den Visualisierungsworkflow und damit in die Nuberisim-Plattform eingebunden. Simulationsergebnisse können so direkt in der Cloud visualisiert werden ohne die Ergebnisdaten herunterladen zu müssen.

Die Integration im Projekt hat die gemeinsame Nutzung beider Systeme ermöglicht und somit den Bedarf an umfangreichen Datentransfer reduziert. Somit wurde die Grundlage für einfachere Simulation und Visualisierung geschaffen. Aktuell wird das Potential einer gemeinsamen Vermarktung geprüft.

2 Einführung

Im nachfolgenden Kapitel wird eine Einführung in den Kontext des Praxispiloten gegeben. Dabei werden die Ausgangssituation und Motivation, die Ziele und der Nutzen für die Zielgruppen sowie der Lösungsansatz näher betrachten.

2.1 Ausgangssituation und Motivation

Die Visualisierung von Ergebnisdaten technischer Simulationen ist essenziell für die Analyse und Entwicklung technischer Geräte und Bauteile. Die dafür benötigten hochauflösenden Simulationen werden auf speziellen Hochleistungsrechnern durchgeführt. Unternehmen, die keinen eigenen Zugang zu solchen Hochleistungsrechnern besitzen, können externe Cloud-Ressourcen buchen.

Für zeitlich und räumlich hochauflösende Simulationen bieten Cloud-Ressourcen dem Anwender spezifische Vorteile:

- Kosteneinsparungen durch bedarfsgenaue Nutzung
- praktisch uneingeschränkte Skalierbarkeit
- parallele und daher zeitlich effiziente Abarbeitung

Nichtsdestotrotz ist mit diesen Vorteilen, die sich grundsätzlich aus der Nutzung von Cloud-Ressourcen ergeben, ein in diesem Fall nicht zu unterschätzender Aufwand verbunden.

Um hochauflösende Simulationen visualisieren zu können, müssen die Simulationsergebnisse typischerweise von den gemieteten Cloud-Ressourcen auf den eigenen Computer geladen werden. Da die Ergebnisse solcher Simulationen u. U. mehrere hundert oder auch tausend GB umfassen können, ist mit langen Downloadzeiten und damit auch mit hohen Datentransferkosten zu rechnen.

So müssen bei 10 Simulationsingenieuren, die je 4, 1 TB-große Simulationen im Monat bearbeiten, 40 TB Daten aus der Cloud auf ihren lokalen Computer geladen werden. Bei Amazon Web Services (AWS) würde dieser Datentransfer etwa 3500$ im Monat kosten[1], was in etwa 2.998 € entspricht.[2]

Zusätzliche Kosten entstehen durch die Wartezeit. Ein Beispiel:

In einem Unternehmen arbeiten 10 Simulationsingenieure, von denen jeder 4 transiente Simulationen mit je 1 TB an Datengröße im Monat bearbeitet. Das Unternehmen besitzt einen Glasfaseranschluss mit einer Bandbreite von 1 Gbit/s. Dadurch können 0,125 GB/s heruntergeladen werden. Um die insgesamt 40 TB Daten im Monat aus der Cloud zu laden, werden etwa 90 h benötigt. Zusätzliche Kosten können durch die Wartezeit entstehen. Ein Beispiel: Bei einem Stundenlohn von 50 € betragen die Kosten für die Download-Wartezeit 4.500 € pro Monat. Die Vorteile Cloud-basierter Simulationsprozesse werden durch die hohen Kosten relativiert.

1 AWS - Ausgehende Datenübertragung aus Amazon EC2 in das Internet, URL: https://aws.amazon.com/de/ec2/pricing/on-demand/ (Zugriff am 28.09.2020)
2 Wechselkurs vom 28.09.2020

Vor diesem Hintergrund ist der Verbleib der Simulationsergebnisse in der Cloud und die dortige Weiterverarbeitung in Form einer Visualisierung attraktiv, zumal auch die im Allgemeinen gegenüber dem PC leistungsfähigeren Ressourcen dann für die Visualisierung genutzt werden können. Cloud-Ressourcen wie z. B. Speicherplatz sind vergleichsweise günstig und die Bedeutung eines effizienten Renderings über das Internet steigt.

Bei dieser Form der Datenverarbeitung ist ein zeitlich effizienter Zugriff auf die Daten gewährleistet, da die Visualisierung in relativer Nähe zu den erzeugten Datenmengen stattfindet und der Download der Simulationsergebnisse obsolet wird. Dieses Vorgehen bezeichnen wir mit dem Begriff der Remote-Visualisierung.

Die Falquez, Pantle und Pritz GbR entwickelt und vermarktet die Nuberisim-Plattform. Diese Simulationsplattform bietet einen Cloud-Zugang für technische Simulationsverfahren. Derzeit liegt der Schwerpunkt auf Strömungssimulationen (Computational Fluid Dynamics [CFD]) sowie strömungsinduzierten Lärmprognosen (Computational Aeroacoustic [CAA]) für eine Vielzahl technischer Geräte.

Die Plattform ermöglicht es, hochauflösende Simulationen in der Cloud zu berechnen und die Simulationsergebnisse im Anschluss auch dort zu speichern. Darüber hinaus kann die Software über eine Browser-API auf die individuellen Simulationsanforderungen angepasst werden. Durch die Steuerung der Simulation sowie die tatsächliche Datenverarbeitung und -speicherung in der Cloud unterscheidet sich der Simulations-Workflow elementar von dem, der bei einem regulären Hochleistungsrechner zum Einsatz kommt.

So werden beispielsweise keine direkten Abfragen von Daten auf einem Dateisystem durchgeführt. Stattdessen erfolgt die Abfrage einer Datenbank nach der Verlinkung zum Speicherort. Information und Anwendung sind voneinander entkoppelt, wodurch der Zugriff kanalisiert und z. B. verschlüsselt werden kann. Die dadurch entstehende modulare Architektur ermöglicht die Einbindung vieler unterschiedlicher verteilter Systeme und damit auch vieler unterschiedlicher Dateisysteme, wodurch sich jedoch auch die Komplexität der Datenabfrage im Simulationsworkflow erhöht. Dieses Vorgehen verursacht daher einen Overhead, der z. B. im Vergleich zu klassischem Hochleistungsrechner den Datenzugriff verlangsamt. Der wesentliche Vorteil ist jedoch, dass sich dieses Vorgehen flexibel in ein mandantenfähiges Cloud-Gesamtsystem integrieren lässt.

Typische Anwendungen für z. B. die CFD-Visualisierung (z. B. mit Tecplot oder Paraview) verfügen zwar über sehr ausgereifte und vielfältige Funktionen. Diese können jedoch nur dann sinnvoll angewandt werden, wenn sich die zu visualisierenden Simulationsergebnisse im selben Netzwerk befinden und darauf über eine entsprechend schnelle Datenverbindung zugegriffen werden kann. Sobald das jedoch nicht mehr der Fall ist und der Datenzugriff über das Internet aus der Ferne erfolgen muss, werden die von den typischen Anwendungen angebotenen Lösungen unbefriedigend oder nicht nutzbar: die Benutzung wird träge, die Visualisierungsmöglichkeiten stark eingeschränkt oder eine Remote-Visualisierung ist schlichtweg nicht möglich. Die Gründe für dieses Verhalten sind oftmals technischer Natur oder stehen im Zusammenhang mit dem verfügbaren Lizenz-Modell, das eine solche Nutzung nicht ermöglicht.

Der Bedarf nach einer effizienten Remote-Visualisierung mit vielfältigen Visualisierungsmöglichkeiten ist somit vorhanden. Das Fraunhofer-Institut für Graphische Datenverarbeitung IGD forscht seit einiger

Zeit im Bereich der Remote-Visualisierung und entwickelt einen effizienten Remote-Visualisierer für Simulationsergebnisse. Nicht nur kann damit der Download der Simulationsergebnisse zu Visualisierungszwecken obsolet gemacht und Kostensenkungen durch einen effizienten Einsatz von Cloud-Ressourcen erzielt werden, auch ergibt sich durch die Remote-Visualisierung und des damit verbundenen Ablaufs des Simulations-Workflows im Browser eine Geräteunabhängigkeit, die die momentan am Markt verfügbaren Lösungen nicht bieten. Die besondere Architektur der Visualisierer ermöglicht außerdem volle Interaktivität auch bei niedrigen Bandbreiten.

2.2 Ziele und Nutzen für die Zielgruppen

Das wesentliche Ziel dieses Praxispiloten besteht darin, eine Lösung aufzuzeigen, mit deren Hilfe die Visualisierung von Simulationsergebnissen in Form eines Cloud-basierten Workflows im Browser durchgeführt werden kann. Kurzfristig zielt die Lösung vor allem auf Entwicklerinnen und Entwickler strömungsführender Geräte und Anlagen (z. B. Lüfter, Pumpen, Turbinen, etc.) in der Maschinenbaubranche. Langfristig ist die Ausweitung auf Entwickelnde mit einem allgemeinen Bedarf an Simulationsservices in weiteren Branchen wie dem Bauingenieurwesen oder der Medizintechnik denkbar, da dort auch Simulationen für die Entwicklung technischer Geräte verwendet werden.

Unabhängig von der konkreten Branche oder Art der Simulation besteht der Nutzen der in diesem Praxispiloten entwickelten Lösung neben der Einsparung von Datentransfer- und -haltungskosten (s. Abschnitt 2.1) vor allem in der Möglichkeit der Nutzung von Hochleistungsrechner-Architekturen für anspruchsvolle Visualisierungen.

Darüber hinaus ermöglicht die Remote-Visualisierung ein verbessertes Vorgehen, wenn es um die Qualitätssicherung von Simulationen geht. Im Normalfall müssen Simulationsergebnisse zunächst visualisiert werden, ehe Fehler (z. B. unangemessene Randbedingung) erkannt werden können. Dieses sequenzielle Vorgehen verlangsamt den Simulationsprozess und macht ihn zudem schwerfällig, wenn es um das Auffinden und Korrigieren von Fehlern geht. Im Gegensatz dazu, erlaubt die Remote-Visualisierung eine vollumfängliche Live-Analyse der Simulationsergebnisse, wodurch Fehler in der Simulation frühzeitig erkannt und behoben werden können. Da die Visualisierung im Browser abläuft ergibt sich eine Geräteunabhängigkeit, die den Online Zugriff z. B. im Rahmen einer Kundenpräsentation von Simulationsergebnissen ohne lizenzrechtliche Beschränkungen auf ein bestimmtes Netzwerk erlaubt. Als langfristiger Nutzen, ist der Ausbau des Remote-Visualisierers zu einem intuitiven und themenangepassten Tool mit geringer Latenz zu sehen.

Das Fraunhofer-Institut für Graphische Datenverarbeitung IGD verfügt über eine browser-basierte Visualisierungslösung, die bereits auf einem klassischen Hochleistungsrechner erfolgreich implementiert wurde. Die Lösung ist so gestaltet, dass sie prinzipiell auch auf einer Cloud-Plattform als Remote-Visualisierung eingesetzt werden kann. Sie ist außerdem so angelegt, dass sie für typische Visualisierungsanforderungen von CFD-artigen Simulationen erweiterbar ist. Der Lösungsansatz basiert auf einer zweiteiligen Implementierung des Visualisierers in Form einer Client-Server-Architektur.

Der Lösungsansatz sieht eine Integration der Browser-basierten Visualisierungslösung (Fraunhofer IGD) mit der Nuberisim-Plattform (Falquez, Pantle und Pritz GbR) vor. Dadurch soll eine Remote-Visualisierung ermöglicht werden, mit deren Hilfe CFD- und CAA-artige Simulationen im Browser und unter der Nutzung von Cloud-Ressourcen visualisiert und analysiert werden sollen.

Dadurch, dass anstelle der unverarbeiteten (100 – X000 GB) nur noch bereits in der Cloud visualisierte bzw. gerenderte und komprimierte Bilder von Simulationsergebnissen auf den Client übertragen werden müssen, deren Größe sich im MB-Bereich befindet, werden erhebliche Downloadressourcen (z. B. Zeit) eingespart. Abbildung 1 stellt den Unterschied zwischen dem bisherigen Simulationsworkflow (oben) und dem Workflow der Remote-Visualisierung (unten) gegenüber.

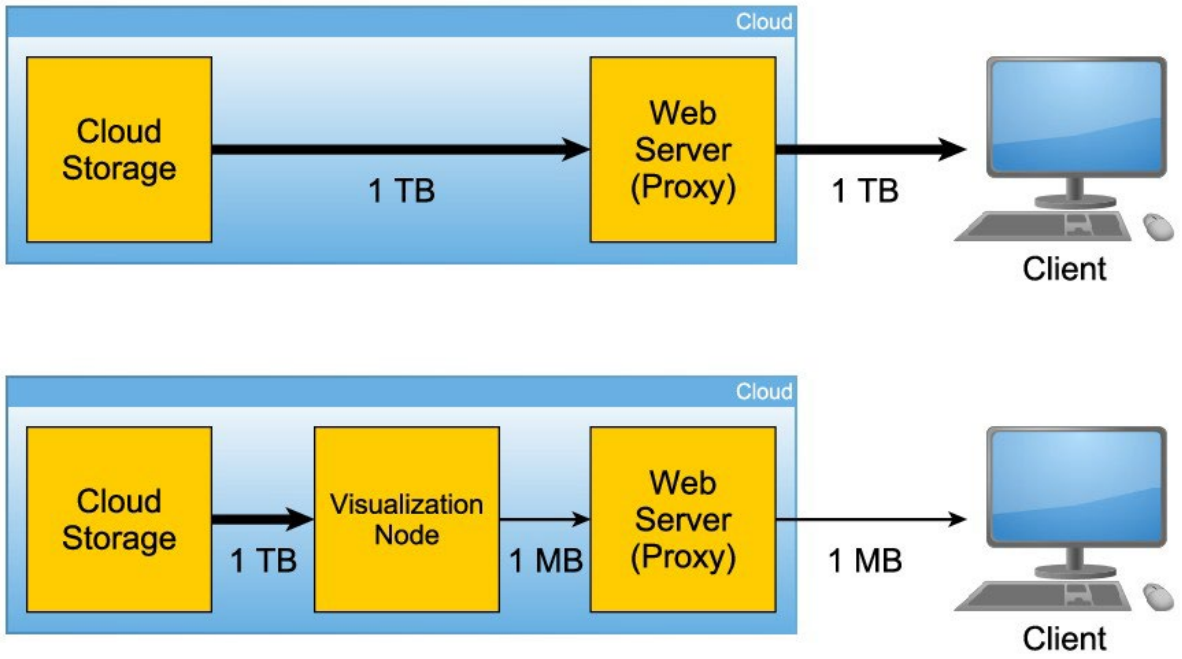


Abbildung 1: Workflow einer Remote-Visualisierung

Der dabei neue entstehende Workflow soll so konzipiert werden, dass sowohl eine gemeinsame als auch eine getrennte Vermarktung des Ergebnisses sichergestellt ist.

3 Projektrahmen

In diesem Kapitel werden die am Projekt beteiligten Unternehmen und ihre Rolle, die sie im Projekt eingenommen haben, erläutert. Darüber hinaus wird auf die Ressourcen eingegangen, die für die Projektdurchführung benötigt wurden.

3.1 Konsortium und Rollen

Im nachfolgenden Abschnitt werden die am Projekt beteiligten Unternehmen vorgestellt und ihre Rolle im Projekt erläutert.

3.1.1 Falquez, Pantle und Pritz GbR (FPPG)

Die Falquez, Pantle und Pritz GbR (FPPG) ist ein junges Ingenieurbüro für die Auslegung, Entwicklung und Simulation strömungstechnischer Anlagen. Herzstück ist die eigene Software-Entwicklung: Mit dem Nuberisim-Solver werden integriert Strömungssimulation und Aero-Akustik-Simulation von Strömungsmaschinen (Lüftern, Turbinen u. ä.) und anderen strömungsführenden Geräten und Anlagen durchgeführt und lärmarme Geräte entwickelt. Die eigene Cloud-Plattform Nuberisim stellt jedem Nutzer Werkzeuge für umfangreiche Detailsimulationen zur Verfügung, wobei auch Hochleistungsrechenressourcen genutzt werden können.

Die FPPG ist Entwicklungspartner im Konsortium mit dem Ziel, die Nuberisim-Plattform und das Angebot dort zwecks besserer Vermarktung weiter zu entwickeln. Hierzu gehört die Einbindung eines leistungsfähigen Remote-Visualisierers, hier in Form des IGD-Visualisierers.

3.1.2 Fraunhofer-Institut für Graphische Datenverarbeitung IGD

Das Fraunhofer IGD ist durch die Abteilung Interaktive Engineering Technologien (IET) in diesem Projekt vertreten. Die Abteilung verfolgt angewandte Forschung auf dem Gebiet der Geometrieverarbeitung und Design, schneller interaktiver Simulation, als auch Visualisierung und Analyse. Eines der Hauptziele der Abteilung ist die Integration von Design, Simulation und Visualisierung mit dem Ziel der Interaktivität, d. h. eine Technologie, die es den Benutzern ermöglicht mit den virtuellen Modellen und der Visualisierung zu interagieren.

Die IET-Abteilung ist der Entwickler des Remote-Visualisierers, welcher zur Anzeige und Analyse der Simulationsergebnisse verwendet wird. Außerdem soll die Lauffähigkeit des Visualisierers auf weiteren Plattformen sichergestellt werden. Zusätzlich soll langfristig der Funktionsumfang des Visualisierers erweitert und evaluiert werden.

Das IGD hat die Problematik des Datendownloads von Simulationsergebnissen auch bei klassischen Hochleistungsrechner bereits erkannt und einen ersten Remote-Visualisierer mit schlankem Rendering entwickelt. Dieser bietet technisch das Potential der Einbindung in die Nuberisim-Plattform, ist prinzipiell geeignet für Strömungssimulationen und wird weiterentwickelt bzw. angepasst.

3.1.3 Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO

Im Zuge der Projektdurchführung unterstützte das Fraunhofer IAO im Rahmen des Förderprojekts Cloud Mall BW beim Projektmanagement und bei der Aufbereitung, Dokumentation und Verbreitung der Ergebnisse in Form der Transferdokumentation. Schwerpunkt dabei waren die Kommunikation sowie Wegbereitung für eine erfolgreiche Kooperation und Integration der beteiligten Partner.

3.2 Notwendige Ressourcen sowie Kompetenzen

Mit dem Kick-Off Mitte November 2019 sollte das Projekt im April 2020 beendet werden. Aufgrund insbesondere der Verzögerungen bei der Bereitstellung der GPU-VM und der Corona-bedingten Umorganisationen der Arbeitsabläufe (s. Abschnitt 4.5) betrug die Projektlaufzeit am Ende jedoch ca. 12 Monate.

Über den gesamten Projektzeitraum hindurch waren aufseiten FPPG zwei gesellschaftsführende Gesellschafter beteiligt, während das IGD mit einem Wissenschaftlichen Mitarbeiter am Projekt gearbeitet hat.

Neben der Organisation der GPU-VM, die die FFPG durchgeführt hat, bestand der Großteil der Arbeit in der technischen Integration der zwei Systeme (IGD-Visualisierer und Nuberisim-Plattform). Die IST-Analyse der beiden Systeme beim Kick-Off sowie das im Anschluss daran von der FPPG entwickelte Integrationskonzept führten dazu, dass die Integrationsarbeiten relativ schnell beginnen konnten. Da das Detailwissen über die Systeme bei den jeweiligen Partnern liegt, wurden die diesbezüglichen Abstimmungen oft direkt zwischen der FPPG und dem IGD durchgeführt.

Das Fraunhofer IAO führte das Projektmanagement durch und war für die Aufbereitung, Dokumentation und die Verbreitung der Ergebnisse in Form der Transferdokumentation zuständig.

4 Lösungsbeschreibung

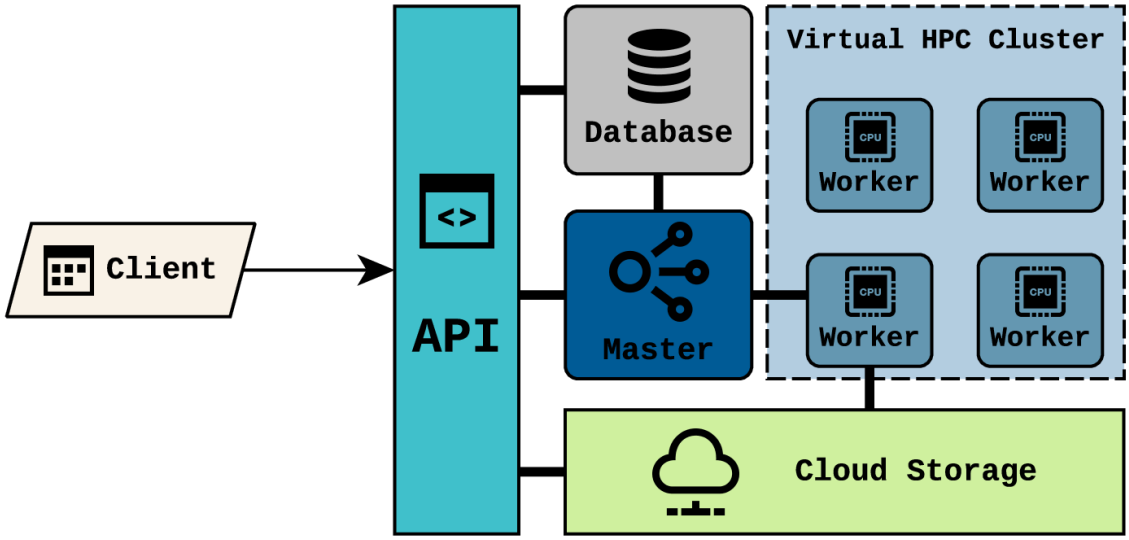
In diesem Abschnitt werden die Anforderungen erläutert, die vom IST-Zustand der zu integrierenden Systeme abgeleitet wurden. Neben der letztendlichen Benutzeroberfläche wird auch auf mögliche Geschäftsmodelle eingegangen, die die Integration der beiden Systeme in sich birgt. Weiter wird der im Zuge der Entwicklung entstandene Prototyp näher beschrieben. Darüber hinaus wird auch auf den Prototypen, die Umsetzungsherausforderungen sowie Endnutzererfahrungen eingegangen.

4.1 Anforderungen

Beim Kick-Off des Projekts wurden allen Beteiligten die zu integrierenden Lösungen zunächst vorgestellt. Ausgehend vom IST-Zustand wurde der SOLL-Zustand abgeleitet.

4.1.1 IST-Zustand

Die Nuberisim-Plattform (Falquez, Pantle und Pritz GbR) wird derzeit über die Amazon Web Services (AWS) betrieben und stellt die Datenschnittstelle für den Remote-Visualisierer bereit. Folgende Datenformate werden unterstützt:



- Tecplot
- EnSight
- VTK
- CGNS

Die Visualisierungslösung des Fraunhofer IGD erlaubt es, Simulationsergebnisse im Browser zu visualisieren. Zusätzlich werden Post-Processing-Funktionalitäten wie Schnittflächen oder Stromlinien unterstützt (vgl. Abbildung 2).

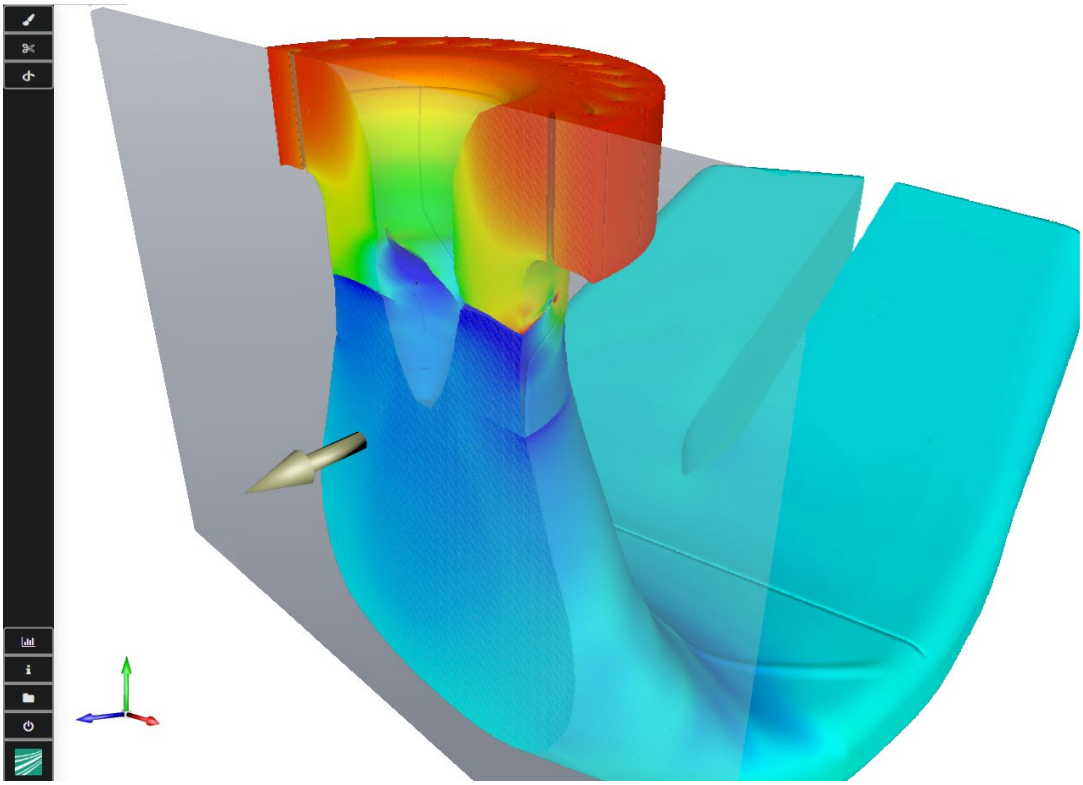


Abbildung 2: Browser-basierte Visualisierungslösung des Fraunhofer IGD. Die Visualisierung zeigt den Geschwindigkeitsbetrag einer Strömungslösung in einer Wasserkraftturbine farblich kodiert von blau (niedrig) nach rot (hoch).

Das Rendering ist GPU-basiert und wird serverseitig mittels OpenGL und clientseitig mittels WebGL realisiert. Die Kommunikation zwischen der Client- und Serverkomponente findet über ein vom Fraunhofer IGD entwickeltes Format zur Datenkompression statt und verwendet protobuf als Kommunikationsprotokoll. Zusätzlich dazu existiert eine lokale Geometrie, sodass Visualisierungsinteraktionen in Echtzeit möglich sind.

Der Visualisierer befindet sich im Bereitschaftsmodus und wird für jede Visualisierungssitzung separat instanziiert. Dabei ist die Instanz an den zu bearbeitenden Datensatz bzw. Pfad zum Datensatz gebunden. Es existiert keine Kommunikation zwischen dem Visualisierer und den Datenbanken, die die zu visualisierenden Daten enthalten. Mit Beendigung der Visualisierungssitzung („Session“) durch den Benutzer wird auch die zugehörige Instanz beendet.

Als Ergebnis produziert der Visualisierer nicht direkt fertige Bilder, sondern spezielle Visualisierungsinformationen, sogenanntes vorgerendertes 2.5D. Dabei hängt die Größe dieser Informationen nicht vom zu visualisierenden Datensatz ab, sondern von der Darstellungsauflösung und der Menge an gefüllten Pixel. Ein 2.5D-Frame liegt im Bereich von 500 kB – 5 MB. Wird mit der Visualisierung interagiert (z. B. die Kameraposition geändert), so manipuliert der Nutzer während der Interaktion nur die lokale 2.5D-Visualisierung. Erst nach Beendigung der Interaktion (Loslassen der Maustaste) wird ein neuer 2.5D-Frame vom Visualisierer angefordert. Dieses Vorgehen hält die Übertragungskosten gering und ermöglicht volle Interaktivität auch bei niedrigen Bandbreiten.

4.1.2 SOLL-Zustand

Der Remote Post-Processor (RPP / IGD-Visualisierer) basiert bereits auf einer Client-Server-Architektur. Der RPP-Server (Abbildung 3 – rechts) ist eine Konsolenapplikation und basiert auf C++ und OpenGL in Verbindung mit dem Qt-Framework. Hierbei hat er die Aufgabe die Simulationsdaten zu laden und in eine interne Datenstruktur zu überführen. Diese Daten werden dann entweder direkt gerendert oder zunächst durch Post-Processing-Algorithmen bearbeitet um dann anschließend durch ein Offscreen-Renderingverfrahren visualisiert zu werden. Die gerenderten Informationen werden anschließend enkodiert, komprimiert und über die WebSocket-Verbindung an den Client geschickt.

Der Web-Client (Abbildung 3 – links) basiert auf JavaScript/AngularJS und WebGL (in Abbildung 3 blau dargestellt). Der Kern der Applikation wird in C++ geschrieben und mit emscripten in performantes WebAssembly konvertiert (in Abbildung 3 orange dargestellt). Über eine WebSocket-Verbindung erhält der Client das gerenderte Bild in dekodierter Form und stellt es in einem lokalen WebGL-Fenster dar. Hierbei hat der Nutzer die Möglichkeit, sowohl in der Szene zu navigieren als auch die Parameter für das Post-Processing über die Benutzeroberfläche anzupassen. Diese Veränderungen werden an den Server wiederum per WebSocket kommuniziert, um anschließend ein neues Bild darzustellen. Das Update der Bilder erfolgt immer beim Loslassen der Maustaste (mouse-release).

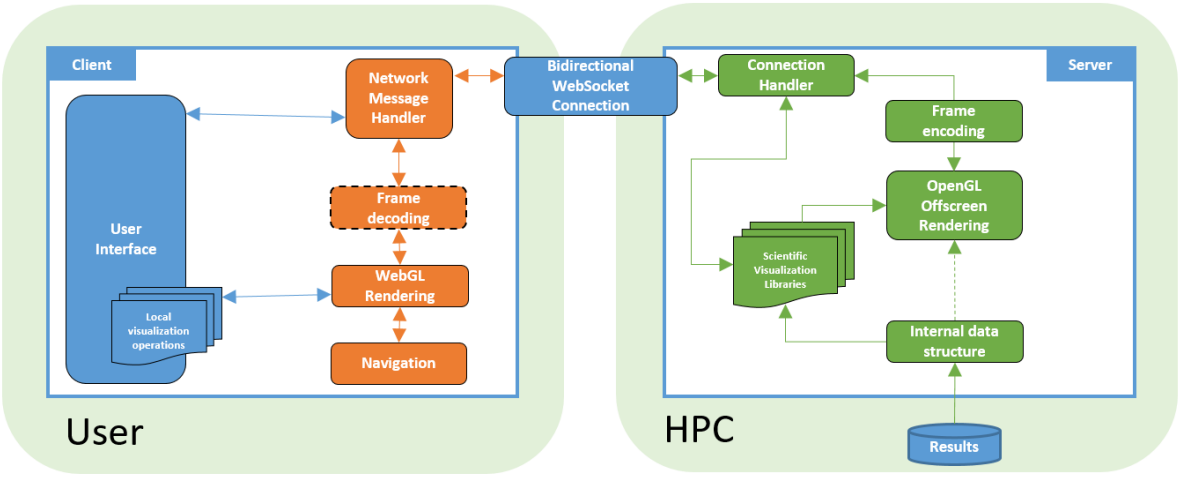


Abbildung 3: Architektur des IGD-Visualisierers aufgeteilt in Client und Server

Der IGD-Visualisierer ist eine gekapselte Komponente, die dynamisch allokierbar ist, auf dem JavaScript-Framework AngularJS basiert und vom Master verwaltet wird. Sie enthält alle für die Visualisierung notwendigen Bibliotheken und kann unabhängig von der Nuberisim-Plattform gewartet werden. Darauf aufbauend soll er in die Nuberisim-Plattform integriert werden.

Hierfür muss der Client dahingehend angepasst werden, dass er in die Lage versetzt werden soll, Nachrichten über Callbacks zu senden und zu empfangen, da in der Nuberisim-Platform die WebSocket-Verbindung von außerhalb erstellt wird und der Visualisierer unabhängig von dieser arbeitet. Der Server wird in einer Docker-Umgebung auf Basis von Ubuntu 16.04 betrieben. Weiter wird eine Cuda-fähige NVIDIA-GPU zum Einsatz kommen. Ein Client-Hoster startet den RPP-Server und teilt dem Client die WebSocket-IP des Servers mit. Er übernimmt die Instanziierung des Visualisierers und das Auflösen der übergebenen Dateipfade. Die Verwaltung der Datenflüsse vom Client zur Nuberisim-Plattform und

damit zum Visualisierer findet zentral im Nuberisim-Master-Node statt; der IGD-Visualisierer würde somit zum Visualization-Node.

Die Darstellung der Simulationsergebnisse erfolgt in Form von Polyedernetzen (Tet, Hex, Mixed) und Oberflächennetzen (Dreiecke, Quads). In diesem Kontext sind die gängigsten Dateiformate:

- CFD General Notation System (CGNS)
- Visualization Toolkit (VTK)

Zwar haben sowohl FPPG als auch IGD das CGNS-Format implementiert, der unterschiedliche Versionsstand führte jedoch dazu, dass nur das VTK-Format Berücksichtigung fand. Eine Anpassung der beiden CGNS-Versionen aneinander hätte zu viele Ressourcen gebunden, die besser in der Plattformintegration aufgehoben waren. Abbildung 4 stellt die Integration des IGD-Visualisierers in die Nuberisim-Plattform dar.

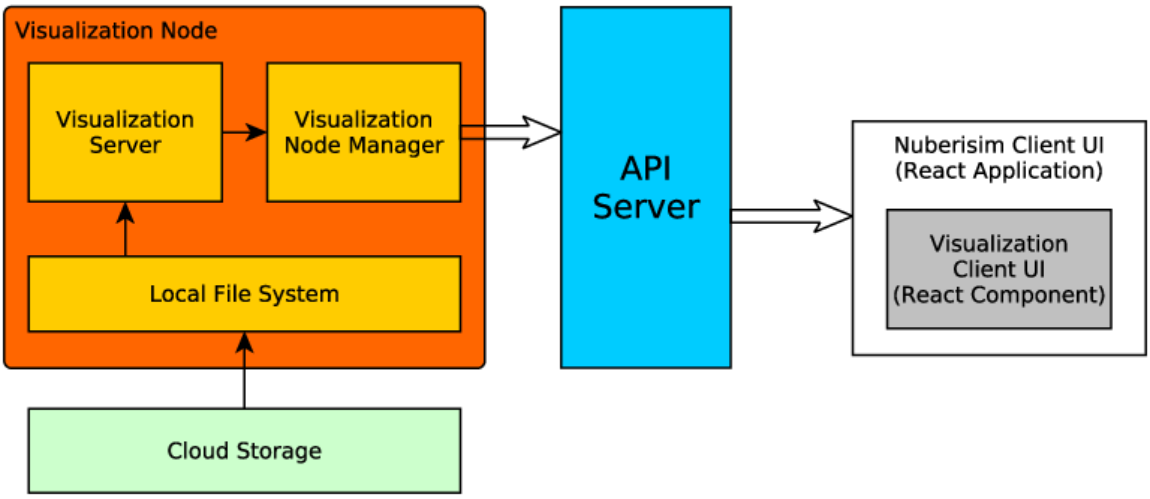


Abbildung 4: Schematische Darstellung der Integration des IGD-Visualisierers

4.2 Graphische Benutzeroberfläche

Die Benutzung der Nuberisim-Plattform und damit auch des IGD-Visualisierers erfolgt im Browser. Der Visualisierer wurde als separater Menüeintrag eingefügt (s. Abbildung 5).

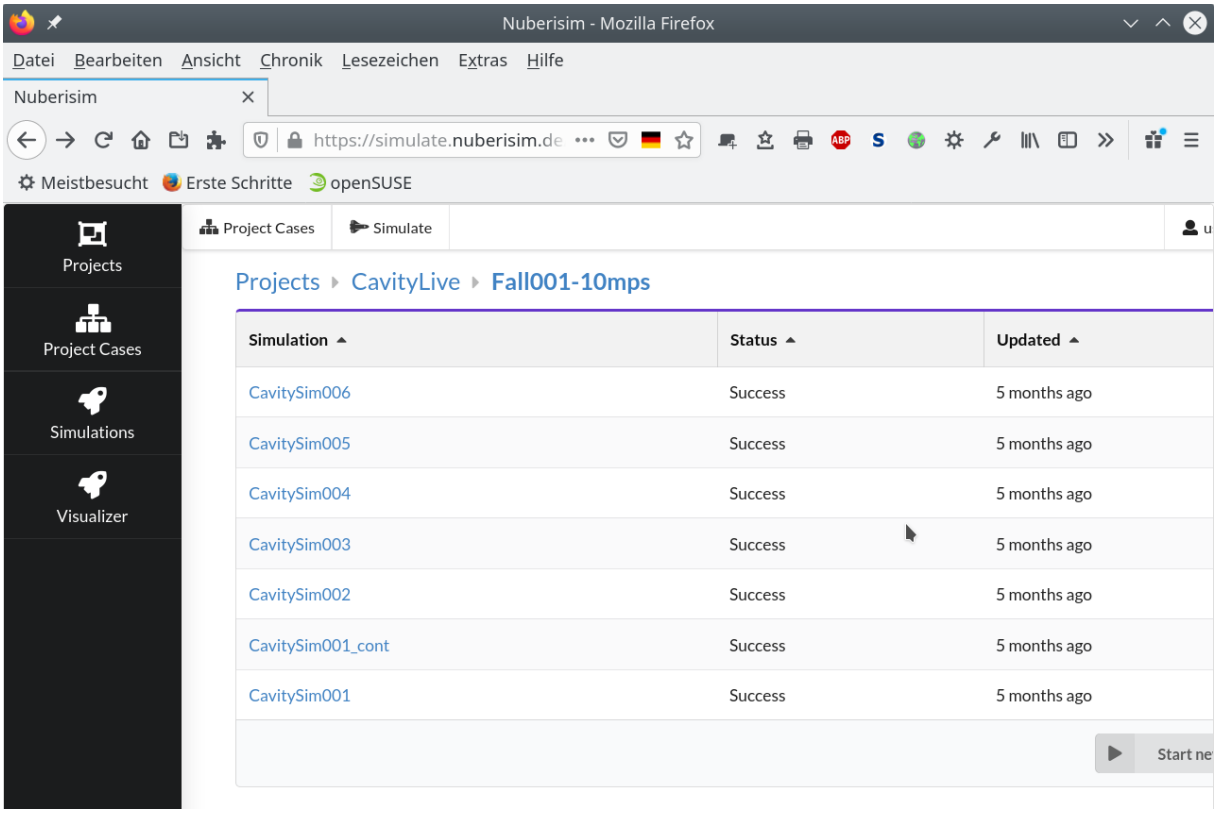


Abbildung 5: Der Visualisierer ist über die Benutzeroberfläche im Browser erreichbar (unterste Schaltfläche im linken Panel)

Wird der Visualizer-Menüeintrag ausgewählt, erscheint das in Abbildung 6 dargestellte Fenster. Während im oberen Teil (*Main Viewer Results Files per Multi-Grid Level*) bereits visualisierte Daten heruntergeladen werden können, können im unteren Teil (*Additional Viewer Results Files per Multi-Grid-Level*) weitere VTK-Daten ausgewählt werden, die visualisiert werden sollen.

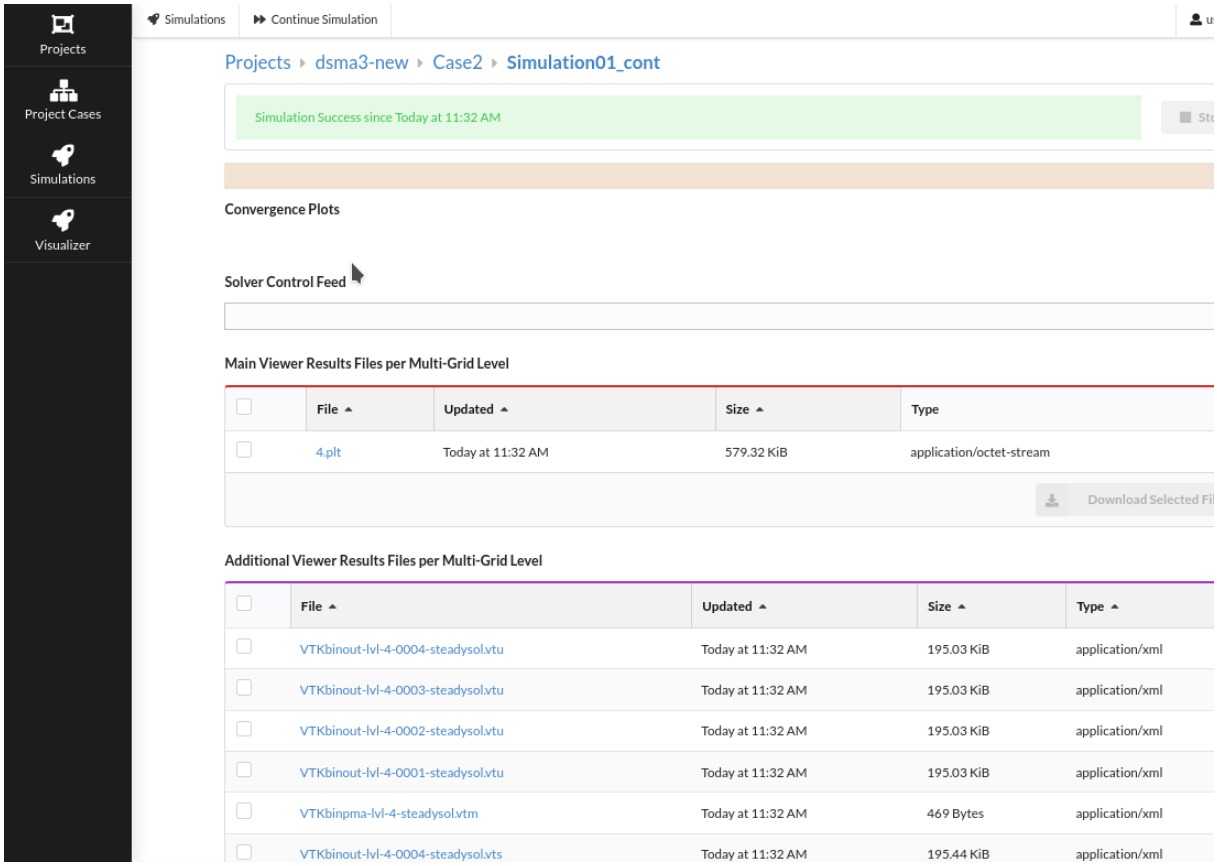


Abbildung 6: Anzeige nach der Auswahl eines Simulationsfalles über die Benutzeroberfläche: Die Liste „Additional Viewer Results Fules per Multi-Grid Level“ enthält diejenigen Ergebnisdateien, die vom IGD-Visualisierer verarbeitet werden.

Eine über die Nuberisim-Plattform gestartete Remote-Visualisierung mittels des integrierten IGD-Visualisierers ist in Abbildung 7 zu sehen.

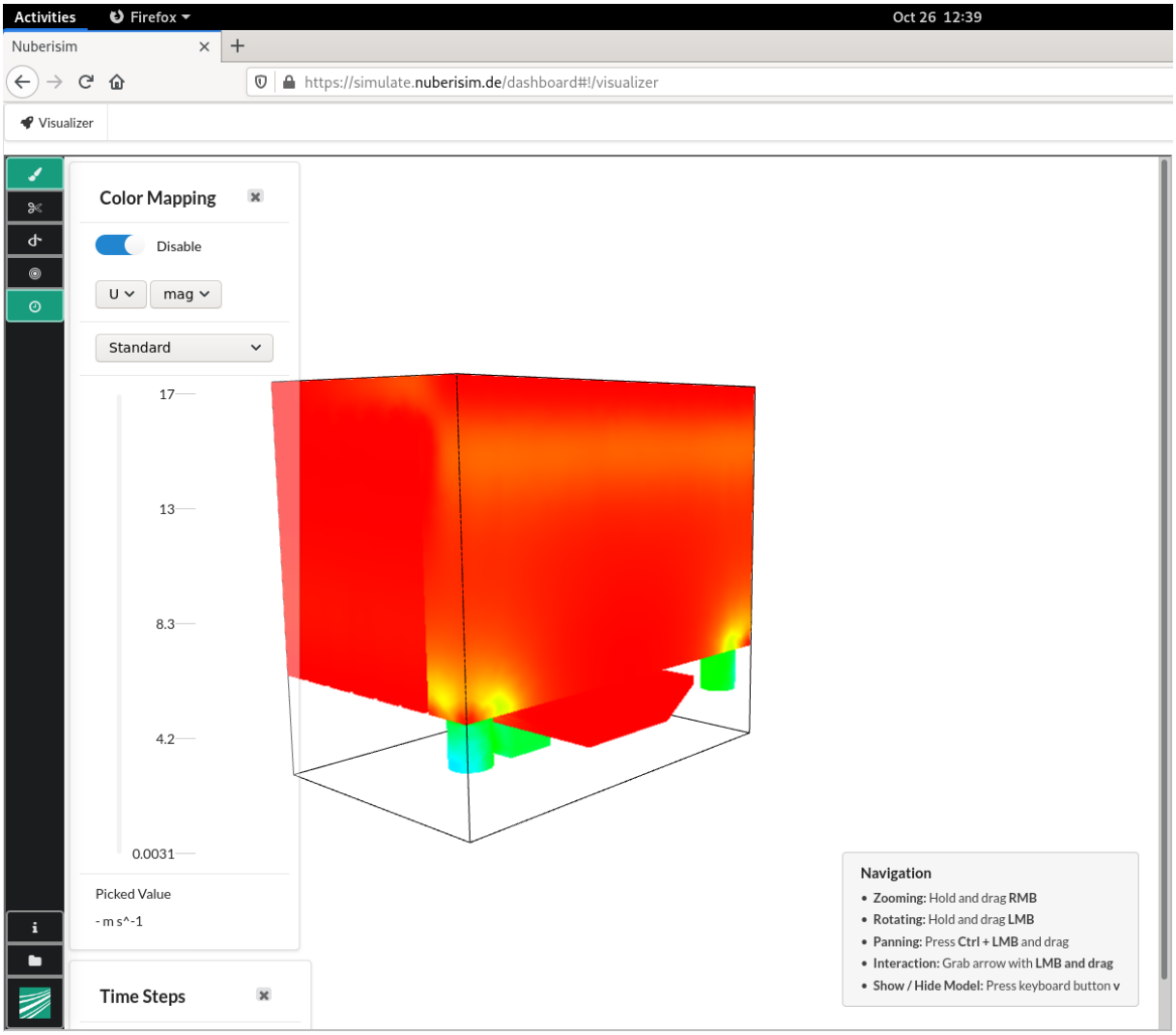


Abbildung 7: Visualisierungsergebnis der Strömungssimulation eines industriellen Ofens. Die Auswahl „U mag“ (U-Magnitude) im Panel „Color Mapping“ bezeichnet den Betrag der Strömungsgeschwindigkeit, der hier mit farblicher Kodierung von blau (niedrig) nach rot (hoch) dargestellt ist. Weiterhin zeigt das „Color Mapping“ den der Farbkodierung zugeordneten Wertebereich des Geschwindigkeitsbetrages.

Die bisherige Integration des IGD-Visualisierers in die Nuberisim-Web-Oberfläche konzentrierte sich auf die Grundfunktionen (z. B. Laden von Simulationsergebnissen und Starten einer Visualisierung). Eine Funktionserweiterung könnte folgendermaßen aussehen:

1. Die Interaktion mit der Visualisierung erfolgt im Browser:
    a. Erstellung und Speicherung von Szenen (z. B. durch Makros oder Layouts)
    b. Übertragung von Szenen zu einer anderen Simulation derselben Konfiguration.
    c. Erstellung und Speicherung von Snapshots.
    d. Erstellung (und Speicherung) von Animationen.
2. Während einer Session sollen folgende Informationen gespeichert werden:
    a. Nutzerdaten,

   b. Simulationsmetadaten (z. B. Zugehörigkeit einer Simulation zu einer Konfiguration und Projekt).
3. Eine beendete Simulation kann mehrfach parallel visualisiert werden (z. B. in mehreren gleichzeitig geöffneten Browser-Tabs).

4.3 Geschäftsmodelle

Bei der Verwendung des IGD-Visualisierers über die Nuberisim-Plattform kommen unterschiedliche Geschäftsmodelle infrage:

Im Rahmen eines Marketplace-Modells eines Cloud-Anbieters stehen sowohl der Visualisierer als auch die Nuberisim-Plattform zur Verfügung und können miteinander verwendet werden. Jede Software für sich hinterlegt im Marketplace ein Preismodell. Die Abrechnung erfolgt nach individueller Nutzung (CPU/GPU-Zeit) oder durch den Verbrauch eines zuvor hinterlegten Budgets. Die Herausforderung besteht jedoch in der Abstimmung mit dem Marketplace und der Anpassung an deren Abrechnungsmodelle.

Alternativ steht der Visualisierer im Hintergrund innerhalb der Nuberisim-Plattform zur Verfügung und die Abrechnung erfolgt über Nuberisim auf Basis der genutzten GPU-Zeit. Dieses Modell erfordert eine intensive Abstimmung mit IGD, da hier zunächst ein Preismodell von IGD entwickelt werden muss. Außerdem ist der bürokratische Aufwand der Abrechnung für FPPG höher. Daher wird prinzipiell das Marketplace-Modell favorisiert.

4.4 Prototyp

FPPG hat im Dezember 2019 einen Prototypen als Vorlage für die Integration des Visualisierers über die Web-Socket-Kommunikation entwickelt und zur Verfügung gestellt.

Der Prototyp ist ein eigenständiges, kompilierbares Programmpaket zu Demonstrationszwecken. Er stellt einen Webserver zzgl. einer einfachen java-script/react-Visualisierer-Anwendung, einen 3-dimensional dargestellten Würfel, bereit. Das Beispiel verdeutlicht den prinzipiellen Aufbau des Web- bzw. Applikationsservers, wie er in der Nuberisim-Plattform umgesetzt ist. Er verdeutlicht die Kanalisierung der Anwendung über einen bestimmten http-Port und die Web-Socket-Kommunikation (s. Abbildung 8).

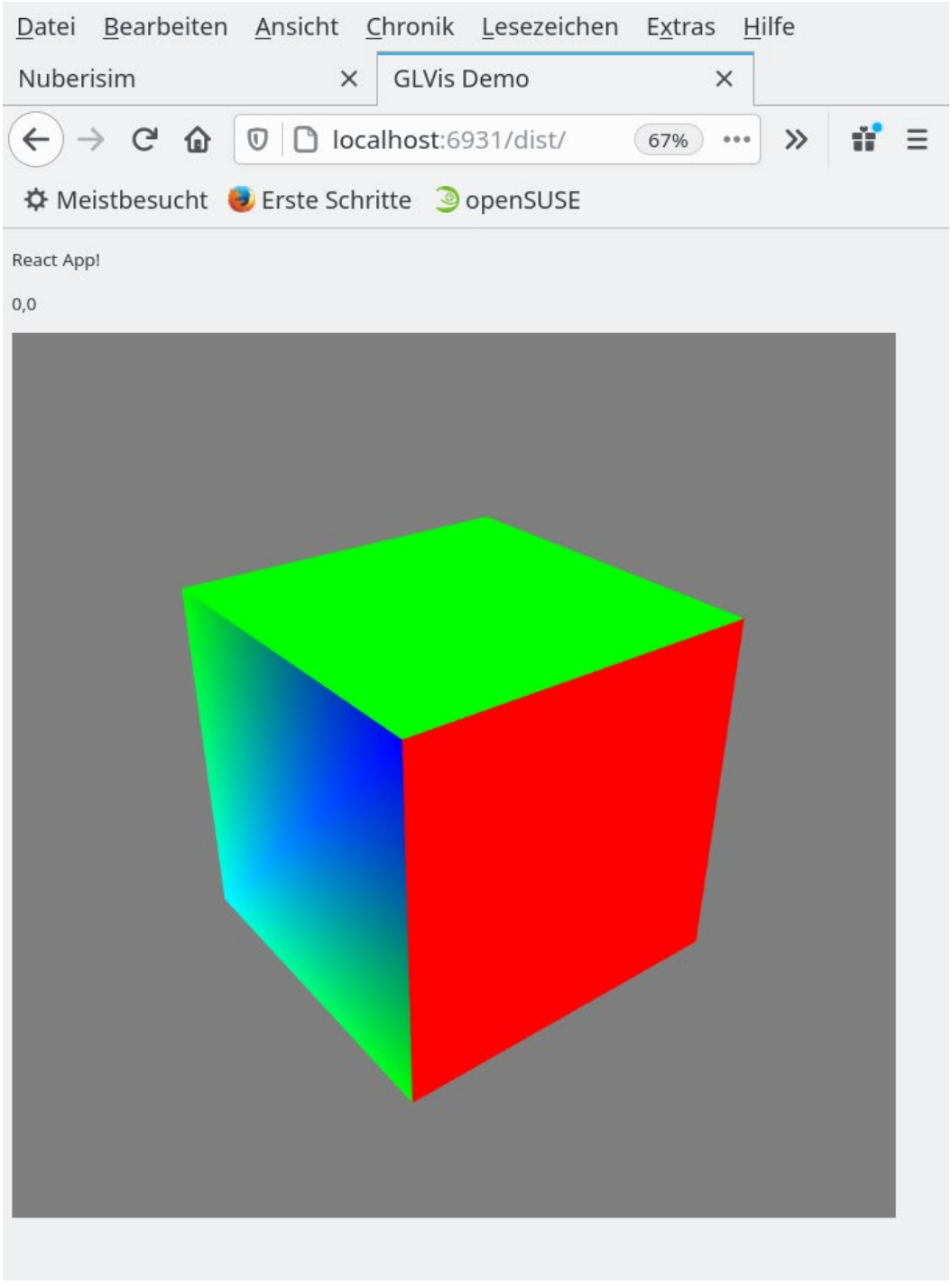


Abbildung 8: CloW-Viz-Prototyp, worin zu Demonstrationszwecken ein Web-GL-Demo in Form eines rotierbaren Würfels integriert wurde. Die Farbkodierung des Würfels basiert auf den Würfelflächen zugeordneten Werten zzgl. eines Farbverlaufs auf Basis perspektivischer Beleuchtung. Zweck des Prototypen ist es, einerseits ein Schnittstellenbeispiel für den zu integrierenden IGD-Visualisierer zur Nuberisim-Plattform bereit zu stellen, das andererseits bereits einfache Visualisierungselemente enthält, womit die Interaktion von Mausbewegung und Darstellung im Browser demonstriert werden soll.

4.5 Herausforderungen bei der Umsetzung

In Bezug auf die erforderliche GPU-VM wurde nach einer kostengünstigen Lösung gesucht, um ausreichend Entwicklungszeit zur Verfügung zu haben. Die Anforderungen waren zum einen, dass ein Cloud-System zur Verfügung steht, worauf die Nuberisim-Plattform aufgespielt werden kann, und zum anderen die Verfügbarkeit einer NVIDIA-GPU, die dort eingebunden werden kann. Hier wurde ein externer Partner eines öffentlichen Trägers gefunden. Mit Beginn der Corona-Pandemie, der daraus resultierenden Umorganisation der Arbeitsabläufe und der anfangs schlechteren Erreichbarkeit der Ansprechpartner sowie der zeitgleichen Hacker-Angriffe auf Hochleistungsrechner in Deutschland, wovon auch der externe Partner betroffen war, erfuhr die Bereitstellung jedoch eine deutliche zeitliche Verzögerung. Die Lösung stand letzten Endes erst Mitte Mai 2020 zur Verfügung.

Die Bereitstellung von Beispieldaten durch FPPG zum Abgleich der lesbaren Ergebnis-Datenformate erfolgte planmäßig. Dabei wurde deutlich, dass zunächst nur das VTK-Format weiterverfolgt werden sollte, da Visualisierer und Nuberisim-Solver jeweils unterschiedliche CGNS-Standards implementiert haben. Der im Nuberisim-Solver implementierte Standard nutzt das HDF5-Format (Hierarchical Data Format), ist damit neuer und parallelisierbar, wogegen der im Visualisierer implementierte Standard noch das ADF-Format (Advanced Data Format) nutzt, damit etwas älter ist und den neueren HDF5-Kern nicht interpretieren kann.

In Bezug auf das VTK-Format ergaben sich weitere Herausforderungen. Die prinzipielle Lesbarkeit der VTK-Ergebnisdateien wurde im Vorfeld verifiziert. Problematisch erwies sich auch hier der Ansatz für die Parallelisierbarkeit, der schon beim CGNS-Format Probleme bereitete, bei VTK aber einer anderen Systematik folgt: Der Solver nutzt die Parallelisierbarkeit des VTK-Formats, indem jeder Prozess seine Teilbereiche (Domain Decomposition Blocks) eigenständig in eigene Dateien schreibt. Im Falle einer nicht-Remote-Visualisierung, z. B. mit dem VTK-Standard-Visualisierer Paraview, können diese Einzeldateien als Liste eingelesen und visualisiert werden. Dies beschleunigt einerseits den Schreibprozess der Ergebnisse, andererseits kann der Visualisierungsprozess parallelisiert werden. Bislang ist der IGD-Visualisierer jedoch nur in der Lage, eine einzige Datei einzulesen und darzustellen. Mittelfristig soll das Einlesen mehrerer Dateien jedoch möglich sein.

Weitere Herausforderungen betrafen die Authentifizierung sowie die statisch vorgegebenen IP-Adressen. Während erstere dadurch gelöst wurde, dass dies die aufrufende Applikation übernimmt, war es bei letzterer durch die Verwendung von iframes.

Hinsichtlich der eigentlichen Implementierungen wurde schnell Konsens erzielt. Von Bedeutung war dabei, dass die Skript-Dialekte sich ähneln und dass der Aufruf des Remote-Visualisierers über eine Web-Socket-Anfrage abgewickelt werden konnte. Ein prototypischer Entwurf dazu wurde von FPPG bereitgestellt. Die nachfolgenden Integrationsarbeiten orientierten sich an diesem Prototyp. Das IGD hatte diesbezüglich die Aufgabe, die entsprechenden Anpassungen im Remote-Visualisierer durchzuführen.

4.6 Erfahrungen von Endanwendern

Die Nuberisim-Plattform wird regelmäßig an der Dualen Hochschule Baden-Württemberg in Karlsruhe in der Lehre eingesetzt. In diesem Zusammenhang ist geplant – idealerweise im Wintersemester 2020/2021 und im Sommersemester 2021 – den Einsatz des Visualisierers auf den im Rahmen dieses Projekts zur Verfügung stehenden Ressourcen zu testen. Die Kursgröße umfasst ca. 20 - 30 Studierende, die den in die Nuberisim-Plattform integrierten Visualisierer dann zeitgleich verwenden.

5 Integration und Kooperation zwischen den beteiligten Unternehmen

Nachfolgend wird die Organisation des Praxispiloten und die technische Zusammenarbeit zwischen FPPG und IGD beschrieben. Weiter wird kurz auf das Strategische und die rechtlichen Rahmenbedingungen dieses Projekts eingegangen.

5.1 Organisatorisches

Die Organisation der Projektpartner wurde überwiegend vom Fraunhofer IAO übernommen und fand zunächst per Telefon, später über MS Teams statt. Da der Fokus dieses Praxispiloten vor allem auf der technischen Integration der beiden Lösungen lag, haben sich FPPG und IGD bei Bedarf auch bilateral ausgetauscht. Als gemeinsame Dateiablage wurde die Fraunhofer ownCloud-Lösung verwendet.

5.2 Technisches

Die Arbeit und Abstimmung zwischen IGD und FPPG erfolgte hauptsächlich durch Regeltelefonkonferenzen. Es wurden regelmäßig Updates bezüglich des Fortschrittes und der Architektur ausgetauscht. Darüber hinaus wurde für die Integration des IGD-Visualisierers in die Nuberisim-Plattform von FPPG ein Prototyp mit ausführlicher Dokumentation an das Team des IGD gegeben, welcher zur Orientierung für die Art und Weise der Integration genutzt wurde. Im Gegenzug lieferte das IGD Dokumentationen mit Codebeispielen für das Einrichten der GPU-VM, der Betreibung des RPP-Servers und der beispielhaften Integration des RPP-Clients zusammen mit dem dazugehörigen Code.

Der Austausch von Codes wurde nicht über ein Versionierungssystem durchgeführt, da die beiden Codebasen in getrennten Repositories gelagert sind.

Das IGD benutzt zur Versionierung SVN und entwickelt unter Zuhilfenahme von Visual Studio 2019 und Visual Studio Code. FPPG nutzt die git-Versionierung und -Repositories, im Allgemeinen unter Zuhilfenahme von IntelliJIDEA, wobei in der Vergangenheit jedoch auch regelmäßig Visual Studio eingesetzt wurde.

5.3 Strategisches

Strategische Planungen für eine gemeinsame Verwertung werden aktuell insbesondere durch die Ausgestaltung passender Geschäftsmodelle unternommen.

5.4 Rechtliches

Für die Zusammenarbeit innerhalb des Praxispiloten wurden von allen Beteiligten die Cloud Mall BW Teilnahmebedingungen unterzeichnet. Diese Regeln umfassen die Zusammenarbeit im Projekt und Aspekte der Vertraulichkeit, den Umgang mit Ergebnissen und Einräumung von Nutzungsrechten, die Vorgehensweise bei Veröffentlichungen sowie Gewähr und Haftung bei der Durchführung des Praxispiloten. Darüber hinaus existiert eine beidseitige Geheimhaltungsvereinbarung zwischen IGD und FPPG.

6 Resümee

In diesem Abschnitt wird die Rolle der Cloud in diesem Praxispiloten näher beleuchtet. Darüber hinaus werden die Lessons Learned beschrieben und ein Ausblick gegeben.

6.1 Rolle der Cloud

Die Rolle der Cloud in diesem Projekt ist essenziell. Sowohl die Simulationsberechnung als auch die Visualisierung läuft in der Cloud ab, sodass die i. d. R. mit der Cloud verbundenen Vorteile nun auch hier nutzbar gemacht werden können:

- Kosteneinsparungen durch bedarfsgenaue Nutzung
- praktisch uneingeschränkte
- parallele und daher zeitlich effiziente Abarbeitung

Die Übertragung der in der Cloud gespeicherten Simulationsergebnisse und damit auch die damit verbundenen Datentransferkosten entfallen. Mit der Projektdurchführung ist zudem die Basis für eine ganzheitliche Simulationslösung im Sinne einer Software as a Service (SaaS) gelegt.

6.2 Lessons Learned

Da der Fokus dieses Praxispiloten sehr stark auf der technischen Integration zweier Systeme lag, sind die bei der Projektdurchführung gewonnenen Erkenntnisse (Lessons Learned) nachfolgend aus der Sicht der beiden Integrationspartner beschrieben.

Lessons Learned aus Sicht von FPPG

- Speziell die Anpassung eines zunächst für Hochleistungsrechner entwickelten Software-Pakets an die Anforderungen eines mandantenfähigen Systems stellt eine Herausforderung dar. Der Workflow und alle darin abrufbaren Anwendungen müssen sich diesem Aspekt unterordnen und dementsprechend angepasst werden, und zwar hinsichtlich Zugriffsrechten, verschiedener Stufen der Datentrennung und -geheimhaltung, des Storage-Managements und Datenabrufs. Dies geht i. d. R. über die Abfrage von Daten bei Hochleistungsrechner innerhalb eines bestimmten Nutzerverzeichnisses hinaus und der Anpassungsaufwand muss bei der Zeit- und Mittelplanung berücksichtigt werden.
- Datenformat ist nicht gleich Datenformat: Besonders die im Nuberisim-Solver umgesetzten Konzepte zur Parallelisierung, die sich bis in den Aufbau und Kern der Ergebnis-Dateien fortsetzt, führt dazu, dass ein und dasselbe Datenformat – obwohl dem vorgegebenen Standard folgend – im Detail unterschiedlich realisiert sein kann. Dies führt z. B. zu der noch fortbestehenden Problematik beim Einlesen einer großen Menge partitionierter Dateien.

Lessons Leaned aus Sicht vom IGD

- VMs mit GPU sind nicht immer trivial aufzusetzen. Visualisierungen in diesem Kontext (Stichwort Offscreen Rendering auf VMs) erfordern Erfahrung bei der Konfiguration der VM und der Installation der richtigen Tools und Abhängigkeiten. Standardmäßig sind z. B. keine GPU Treiber auf den jeweiligen VM installiert.
- Die Integration von zwei verschiedenen Applikationen ist aufgrund von unterschiedlichen Technologie-Stacks selten trivial. Diese Gegebenheiten sollten bei Zeit- und Mittelplanung berücksichtigt werden.

6.3 Ausblick

Das Postprocessing mit dem IGD-Remote-Visualisierer kann in Zukunft in Richtung verschiedener Dimensionen erweitert werden.

Zum einen sind Erweiterungen in der Kommunikation zwischen der Nuberisim-Plattform und dem Visualisierungsserver sinnvoll. Dazu gehört z. B. die strikte Mandantentrennung oder auch die Entkopplung der Server und Instanzen. Zusätzliche Entwicklungen zur Stärkung von Datentrennung und Geheimhaltung spielen dabei eine wesentliche Rolle.

Auch ist eine Erweiterung der Softwarefunktionalität, insbesondere um Funktionen für Strömungssimulationen, denkbar, z. B. die Speicherung von Szenen zur wiederholten Visualisierung bei verschiedenen Simulationsfortschritten oder zwecks Bearbeitung von Sequenzen von transient erzeugten Ergebnisdateien mit dem Ziel, Animationen zu erzeugen.

Auch wäre eine Datenformatangleichung sinnvoll, um die bereits beschriebenen Hürden bei der parallelen Erzeugung und Verarbeitung von Ergebnisdateien zu überwinden (sowohl im CGNS- als auch im VTK-Format).

Weiterhin ist es perspektivisch sinnvoll, die Integration des Visualisierers zu erweitern, um eine bessere Kontrolle bzw. Steuerung von außen zu gewährleisten und auch die implementierungstechnischen Kommunikationswege effizienter gestalten zu können, wodurch eine bessere Anpassung an die Plattform ermöglicht und die Mandantenfähigkeit unterstützt wird.

Zuletzt sollten in Zukunft Integrations- und Belastungstest durchgeführt werden, um zu evaluieren, inwiefern die Robustheit der Applikation unter Realbedingungen gewährleistet ist. Solche Tests erscheinen vor dem Hintergrund einer potenziellen Vermarktung als besonders sinnvoll und sind auch bereits geplant, aber noch nicht realisiert. Vor der Vermarktung muss auch das Lizenzmodell zwischen IGD und FPPG detailliert ausgearbeitet und abgestimmt werden.

7 CMBW-Projektdarstellung

Im Gemeinschaftsprojekt Cloud Mall Baden-Württemberg (Cloud Mall BW) werden Potenziale und Möglichkeiten von Cloud Computing für den Mittelstand in Baden-Württemberg identifiziert und ausgeschöpft. Kleinen und mittleren Cloud-Serviceanbietern und -anwendern wird ein Rahmen geboten, um untereinander Kooperationen zu schließen, das eigene Netzwerk zu stärken und dadurch aktiv Wettbewerbsvorteile auszubauen. Kooperative Ideen kleiner und mittlerer Cloud-Services oder Cloud-Plattformanbieter werden gezielt in Praxispiloten vorangetrieben und personell und fachlich vom Cloud Mall BW-Projektteam unterstützt.

Das Gemeinschaftsprojekt wird vom Ministerium für Wirtschaft, Arbeit und Wohnungsbau Baden-Württemberg gefördert. Beteiligt sind das Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, das Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, sowie das Institut für Enterprise Systems an der Universität Mannheim (InES) und bwcon research GmbH (bwcon). Unterauftragnehmer des Projekts sind Trusted Cloud und das Institut für Arbeitswissenschaften und Technologiemanagement IAT der Universität Stuttgart. In der Projektzeit ist die Durchführung von bis zu vierzig Praxispiloten geplant.

8 Kontakt

Gerne können die Vertreter der Praxispilotpartner bei Fragen und Anmerkungen zum Praxispilot oder zu Inhalten direkt angesprochen werden:

Falquez, Pantle und Pritz Gbr

Iris Pantle

pantle@nuberisim.de

Fraunhofer Institut für Graphische Datenverarbeitung IGD

Christian Altenhofen

christian.altenhofen@igd.fraunhofer.de

CMBW - Projektleiter des Praxispiloten

CMBW Partner

Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO

Dimitri Evcenko

dimitri.evcenko@iao.fraunhofer.de

Weitere Information zum Thema Praxispiloten finden Sie unter der Projektwebsite:
https://cloud-mall-bw.de/