

Kasztelnik et al.: Digital Twin Simulation Development and Execution on HPC Infrastructures

Gliederung

1. Einführung und Motivation
2. Anforderungen an die IT Infrastruktur
3. MEE Infrastruktur
4. Verwendung von GIT
5. Anwendungsfall “The BoneStrength Application”
6. Proof of Concept
7. Andere Use Cases und Lösungsansätze
8. Fazit
9. Quellen

Einführung und Motivation

Digital Twins profitieren von



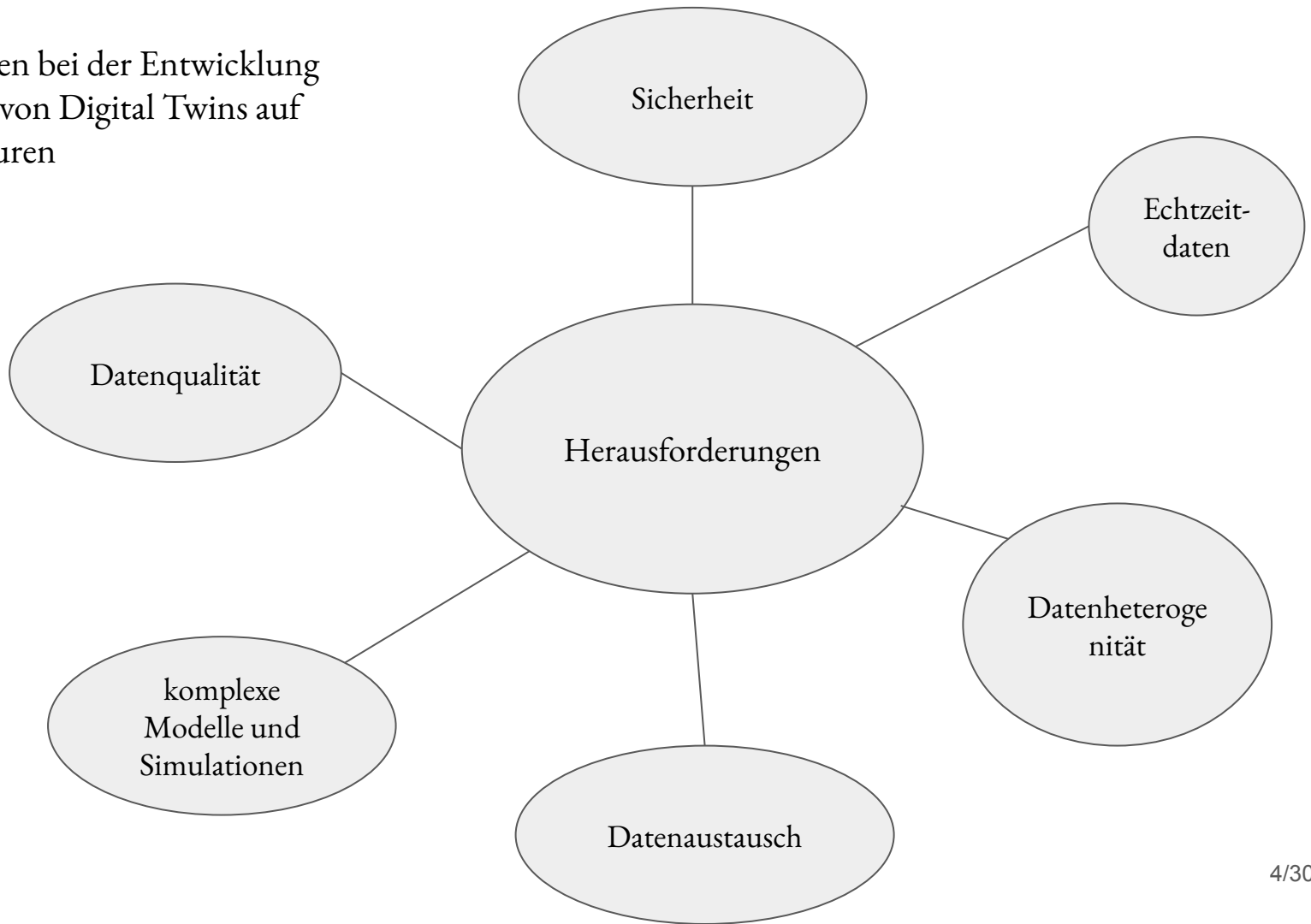
HPC Infrastrukturen bieten

Leistungsstarke Rechenleistung

Große Speicherkapazitäten

Echtzeit Analysefähigkeiten

Herausforderungen bei der Entwicklung und Ausführung von Digital Twins auf HPC Infrastrukturen



Verschiedene Use Cases - Verschiedene Schwerpunkte

Gesundheitswesen anders als E-Commerce

Art der Daten (Sensibel oder Firmengeheimnis)

Compliance (Wettbewerb oder Zusammenarbeit)

Typische Systemanforderungen nach Kasztelnik et al.

- Alle Arten von Daten verstehen
- Umfassendes Repository
- Zugang zu Recheninfrastruktur
- Folgende Arten von Rechenressourcen:

Eigenständige Server	Simulationen in kleinem Maßstab
Cloud-Computing-Infrastrukturen	Moderat große Menge, komplexere Algorithmen
HPC (Bspw. Rechencluster)	Verarbeitung großer Datenmengen

Weitere Funktionelle Anforderungen nach Kasztelnik et al.

Verfügbarkeit von Hardware Komponenten

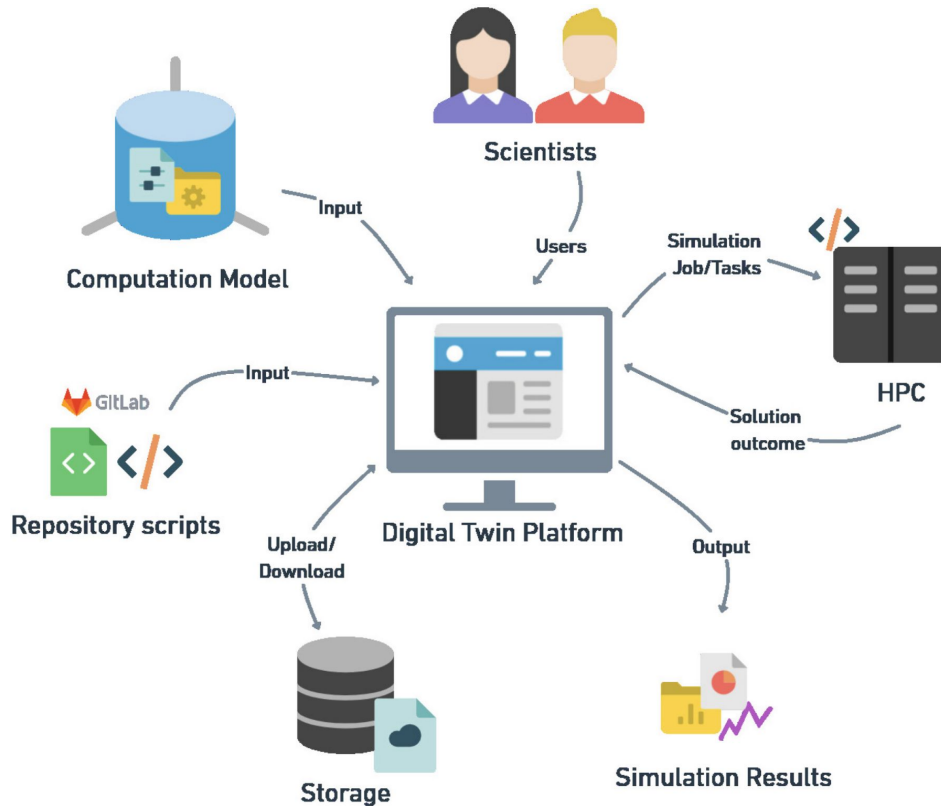
Verfügbarkeit von spezifischen Software packages/Libraries

Sicherstellung der Sicherheit von Daten und Rechenmodellen

Kritik an Vorhandenen Lösungen

Eigenständige Server	/	- Kaum Rechenstärke
Cloud-Computing-Infrastrukturen	- Schneller Zugang - Mehrmandantenfähig	- Kaum Flexibilität - Nicht jedes Model ausführbar
Klassische HPC Rechencluster	- Sehr gute Rechenstärke - Speichervorteile	- Probleme bei der Sicherheit
Arvados Workflow System	- weit entwickelte CWL - bietet einfache Integration von Infrastrukturen	- Modifikation der Anmelde/Berechnungsknoten schwierig
Pegasus	- Robust - Berechnungen auf weiter Palette von Infrastrukturen möglich	- Gesamtkomplexität aktuell zu hoch - Spezifische Funktionen müssen implementiert werden
LEXIS Project	- Berechnungen auf weiter Palette von Infrastrukturen möglich - Sichere Zugänge	- Customization extrem aufwendig, aber nötig - seit 2022 kein Alien4Cloud

Universelle Plattform für Digital Twins



Model Execution Environment (MEE)

- Prototyp Implementation
- Mehrere Projekte: EurValve, PRIMAGE, ISW
- Spezialisiert auf Gesundheitswesen
- Über GSI-SSH und Proxy-Zertifikatsdelegation mithilfe OpenID-Mechanismus integriert
- PLGrid Infrastruktur bietet Zugang zu 8PFlops Rechenleistung
- Mehrere Petabyte Speicher

Ziele:

- Komplexität von HPC verbergen
- Einheitliche Methode zur Speicherung und Pflege von Patienten/Fall Daten einführen



3R Vision

-Repeatability

Eigene Berechnungen wiederholen - Selbes Team, selbe Bedingungen

-Replicability

Fremde Berechnungen erreichen - Anderes Team, selbe Bedingungen

-Reproducibility

Vergleichbare Ergebnisse erzielen - Anderes Team, andere Bedingungen

Modelversionierung wichtig für Durchsetzung

MEE - Wie erfüllt es Prinzipien und Anforderungen?

Modellversionierung

- Integration von Git-Repositories
- Code in Repository pushen
- Modellversion angeben und dann berechnen

Integration mit HPC

- Mit PLGrid-Infrastruktur integriert
- Benutzerrechte von MEE an HPC-Supercomputer delegierbar

3R Vision

- Durch Integration von Git-Repositories
- Lauf protokolliert Modellversion (Git-SHA)
- SHA nutzen um Berechnungen zu wiederholen

Organisation von Patienten-Daten

- Dedizierter Speicherplatz für jeden Patienten(HPC,S3)
- Speicherplatz für spez. Eingaben und Ergebnisse d. Berechnung
- Kohorten für Simulationskampagnen

MEE - Nutzung und Funktionsweisen

Zusammensetzung von Steps, Flows, Patientendaten und Pipelines

Steps:

- Anmeldeinformationen werden bereitgestellt
- Zugriff auf Code-Repository der Simulation ermöglichen

Flows:

- Mehrere Steps werden zu einem Flow
- Vorlage zur Erstellung von Pipeline für best. Patienten

MEE - Nutzung und Funktionsweisen

Pipelines:

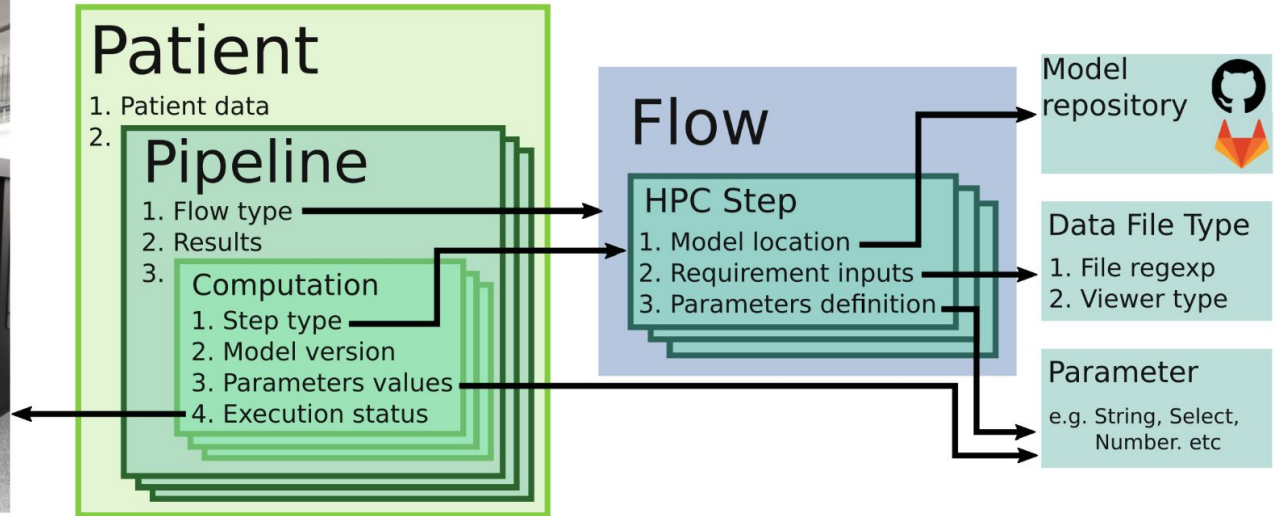
- Definiert Ausgabeort und
- Definiert Liste von auf HPC ausgeführten Berechnungen
- Bestehen aus separaten Berechnungen (Pipeline steps)

Patientendatenstruktur:

- Legt Speicherort für personalisierte Eingabedaten fest
- Enthält Liste von auf Patientendaten ausgeführte Pipelines

Jede Berechnung muss vor dem Start konfiguriert werden. Dazu gehören die Auswahl einer bestimmten Version der Simulation und Festlegung der Eingabeparameter

Struktur der Daten



Verwendung von Git

Ausschlaggebend für Realisierung der 3R Vision und Modelversionierung

Pipeline steps können für Integration in kollaboratives Versionskontrollprojekt konfiguriert werden

Git Repository speichert Template vom WKS und Simulations Sourcecode

Für Berechnung aus Repository Modelversion selektieren

Nach selektierung der Startparameter:

WKS Datei runtergeladen

In endgültige Konfigurationsdatei für Warteschlangensystem umgewandelt (Hier Slurm-Startskript)

Verwendung von Git

Um im Repository zurecht zu finden gibt es folgende “Helfer”:

clone_repo: Zum klonen des Simulations Repository in best. Version

stage_in input-file-type: Sucht input file in Ergebnissen vergangener Berechnungen
pipeline inputs und patient input directories

stage_out file-path: Lädt Simulations Ergebnisse im pipeline output directory hoch

value_of parameter-key: Injiziert Wert von Nutzer in eine Berechnung

Beispiel Anwendung von GIT

```
1 #!/bin/bash -l
2 #SBATCH -N 2
3 #SBATCH --ntasks-per-node=2
4 #SBATCH --mem-per-cpu=3GB
5 #SBATCH --time=04:00:00
6 #SBATCH -A {% value_of grant_id %}
7 #SBATCH -p plgrid-gpu
8 #SBATCH --gres=gpu:2
9
10 {% clone_repo %}
11
12 module load plgrid/apps/cuda/10.1
13 module load plgrid/tools/gcc/8.2.0
14
15 nvidia-smi
16
17 {% stage_in amb-input amb-input.json %}
18
19 ./amb-input Models/Prototype_v2.0/ABM13.4/FGPU_NB --in amb-input.json --
    primage amb-output.json
20
21 {% stage_out amb-output.json %}
```

The BoneStrength Application

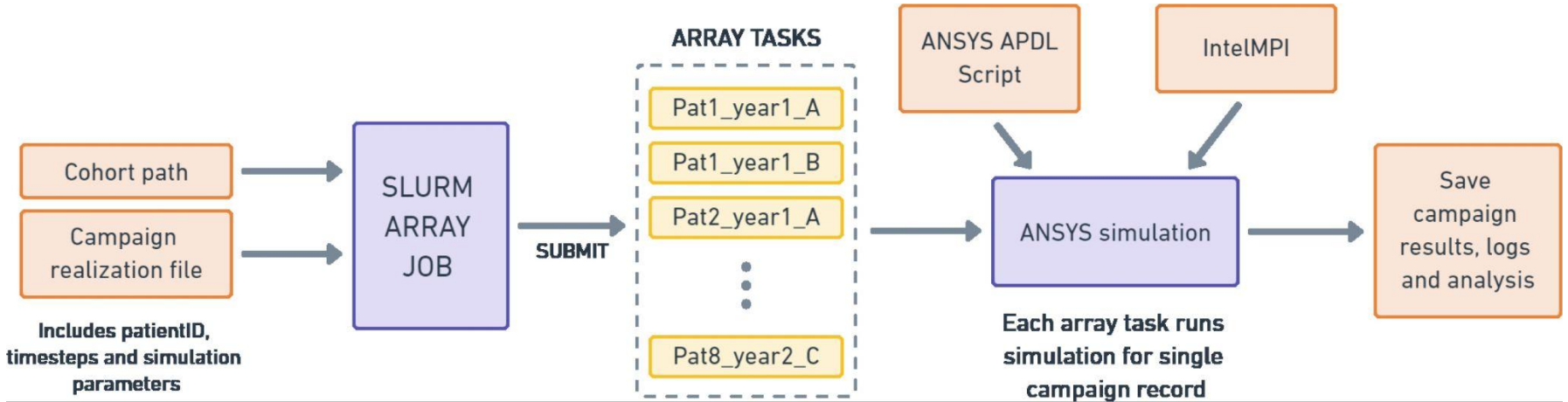
2 Arten von Modellen:

- Finite Element Model
- Stochastisches Patienten-spezifisches Modell

Simulationen um Wahrscheinlichkeit einer Fraktur bei Sturz vorherzusagen

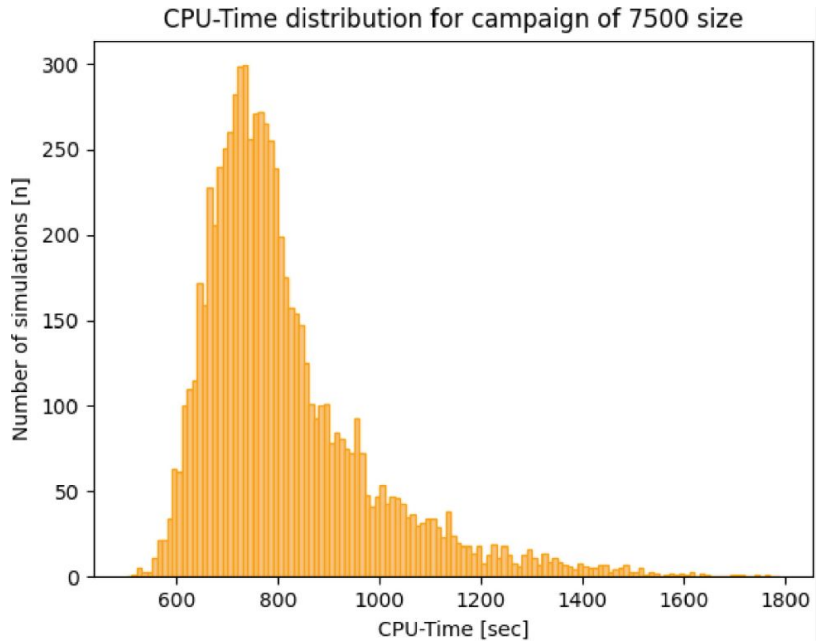
- ANSYS
Mechanical APDL
- MPI Parallelismus
- 5-6GB
Speicherplatz
- 20GB RAM
- 4-8 Kerne
- Dauert 1-2
Stunden

Simulationsablauf



Proof of Concept

- MEE auf BoneStrength Application getestet
- Kohorte von 1080 Patienten
- Insgesamt 7500 Simulationen per Modelversion
- Jede einzelne Campaign-Task 28GB Ram, 4CPUs

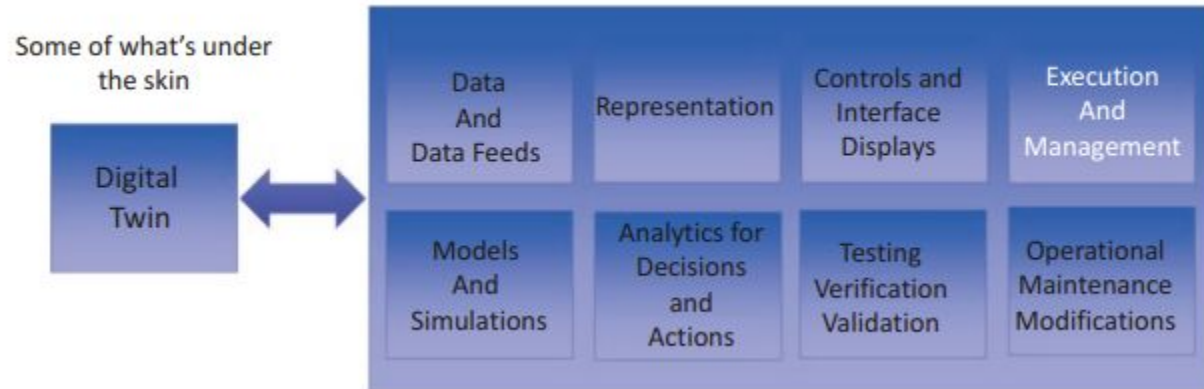


	CPU-Time	Memory used	Efficiency
Mean	813.2 sec	6448.9 MB	94.5 %
Median	769.8 sec	6447.0 MB	92.9 %

Andere Use-Cases (1)

„Infrastructure for Digital Twins: Data, Communications, Computing, and Storage”

-Flavio Bonomi, Adam T. Drobot



Daraus entstehen 4 wichtigsten Blöcke: Datenspeicherung und Pflege, Rechenleistung, Kommunikation und Softwareplattformen und Frameworks

Andere Use-Cases (2)

Neue Erkenntnisse:

- Kuration von Experten (Data and Feeds)
- Repräsentation bei diesem Use Case extrem wichtig (Representation)
- Echtzeit Operationen wichtig (Control, Interfaces and Displays)
- Menschliche Einschätzungen identifizieren können (Analytics for Decisions and Action)
- Testen auf jeder Ebene (Testing, Verification, Validation)
- Wartung und Modifikationsfunktionen (Operational Maintenance and Modification)

Andere Use-Cases (3)

Vielzahl von Use Cases

5 Kategorien

1. Fertigung
2. Produkte
3. Dienstleistungen
4. Prozesse
5. Abstrakte Konzepte und Ideen

Jedoch: Die Wichtigsten Anforderungen sind immer Sicherheit, Wirtschaftlichkeit und Gesellschaftlicher Wert

Andere Use-Cases(4)

Nutzung von Edge und Cloud

Infrastruktur hostet Digital Twins als Softwareanwendung

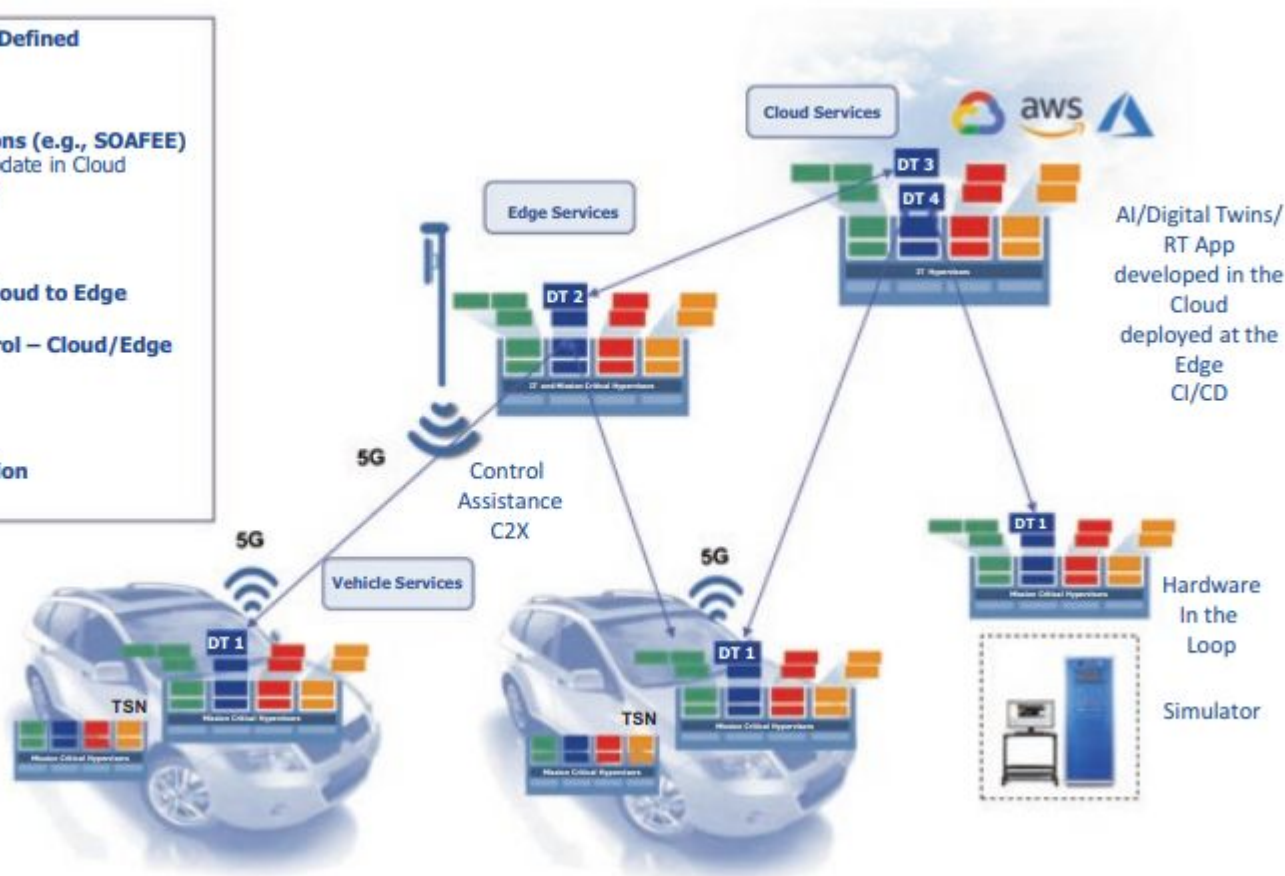
Vehicle Twins (DT1) bspw. Echtzeitfahrzeugsteuerung

Edge Digital Twin (DT2) bspw Intersection Management

Anspruchsvollere Digital Twins werden in der Cloud gespeichert (DT3/DT4)

Implications of Software Defined Automotive:

- **CI/CD for RT Applications (e.g., SOAFEE)**
 - Develop/validate/update in Cloud
 - Deploy at the Edge
- **HIL/SIL**
- **AI/Digital Twin from Cloud to Edge**
- **Software Defined Control – Cloud/Edge Assisted**
- **App Store**
- **Intelligent Transportation**



Weiterer helfender Lösungsansatz

„Toward Building a Digital Twin of Job Scheduling and Power Management on an HPC System” -Tatsuyoshi Ohmura et al.:

Inhalt vom Paper:

- Digital Twin eines Job Schedulers und der Energiesteuerung eines HPC-Systems erstellen
- Bspw Slurm
- Effizienz des HPC Systems verbessern
- Anwendung auf AOBA
- Reduzierung von Wartezeit um 70% möglich, Energieverbrauch um 1,2%
- Bspw auf MEE anwenden um Effizienz zu bewerten

Fazit

- Ausführung von DTs auf HPC Infrastrukturen zieht Herausforderungen mit sich
- Viele Anforderungen an Systeme
- Daten Management, Sicherheit und Heterogenität wichtig
- Verschiedene Use-Cases - Verschiedene Schwerpunkte
- MEE für Medizinische Zwecke sinnvoll - nicht leicht auf alles beziehbar
- Grundrisse könnten Nützlich sein, jedoch ist der Use-Case entscheidend

Quellen/Literatur

Abbildungen Folie 9-22: Kasztelnik et al.: Digital Twin Simulation Development and Execution on HPC Infrastructures

Abbildungen Folie 23-27: Flavio Bonomi , Adam T. Drobot: Infrastructure for Digital Twins: Data, Communications, Computing, and Storage

Kasztelnik et al.: Digital Twin Simulation Development and Execution on HPC Infrastructures

Tatsuyoshi Ohmura et al.: Toward Building a Digital Twin of Job Scheduling and Power Management on an HPC System

Flavio Bonomi , Adam T. Drobot: Infrastructure for Digital Twins: Data, Communications, Computing, and Storage