27.10.2025 (Workshop Hagen)

- Kommunikation über Zulip
- Dokumentation in GitLab

- Treffen abwechselnd mit/ohne Betreuer

- Projektleitung zuerst Andreas (wechselt)
  ↳ Stefan aufnehmen

- offenes Zoom-Meeting für Treffen

- Architektur:
  ↳ Brainstorming Minimalziele
  ↳ Quanten-computing als Blackbox ansehen (Input → Output)
  ↳ kleine Datensätze möglich ⇒ Vision, Proof of Concept
  ↳ fehlerbehaftete Maschinen ⇒ Simulation (mit Noise)
  ↳ Paper zur Architektur (Draft) → Vision Paper
  ] Ausblick

- Entwicklungsumgebung PyCharm Pro für Studenten kostenlos (1 Jahr)
  ↳ auch gut für qiskit ⇒ YouTube Kanal
- Anaconda für Jupyter Notebooks
- Docker - Umgebung² ⇒ eventuell einrichten

- Lehrbuch zum Einarbeiten + YouTube Tutorials] zu wenig Zeit !
  ↳ viel Mathe (Lineare Algebra)

- TO DO: Docker besprechen

- Minimalziele:
  ↳ Algorithmus ≙ Python ≙ Schaltkreis
  ↳ Mechanismus (Template) Schaltkreise für verschiedene Probleme
    ⇒ verschiedene Use cases: Sortieren / Clustern / ...
    ⇒ unterschiedlich codierte Datensätze
  ↳ objektorientiert ?

  ↳ Architektur: + Coder Name für "Create Circuit"
    - generische Architektur + Konfiguration
    - Microservices (Komponenten für sich) → lose gekoppelt (Dummy Prozesse)
      ⇒ nicht ein Koordinator ↳ Technologie-Recherche
      ⇒ Decision Service → nur für Quantencomputer
    - Prozesse kommunizieren miteinander

  ↳ Applikation:
    - einfache Schaltkreise (wenige Algorithmen)
    - GUI → funktional anstatt schön
    - Create Circuit: wählt passenden Schaltkreis aus (JSON, ...)
      ⇒ Iterationen erkennen
      ⇒ Circuit erstellen
    - Python Schnittstelle (Problem, Elemente ⇒ Template)
    - Daten aus Datenbank ziehen
    - Workflow

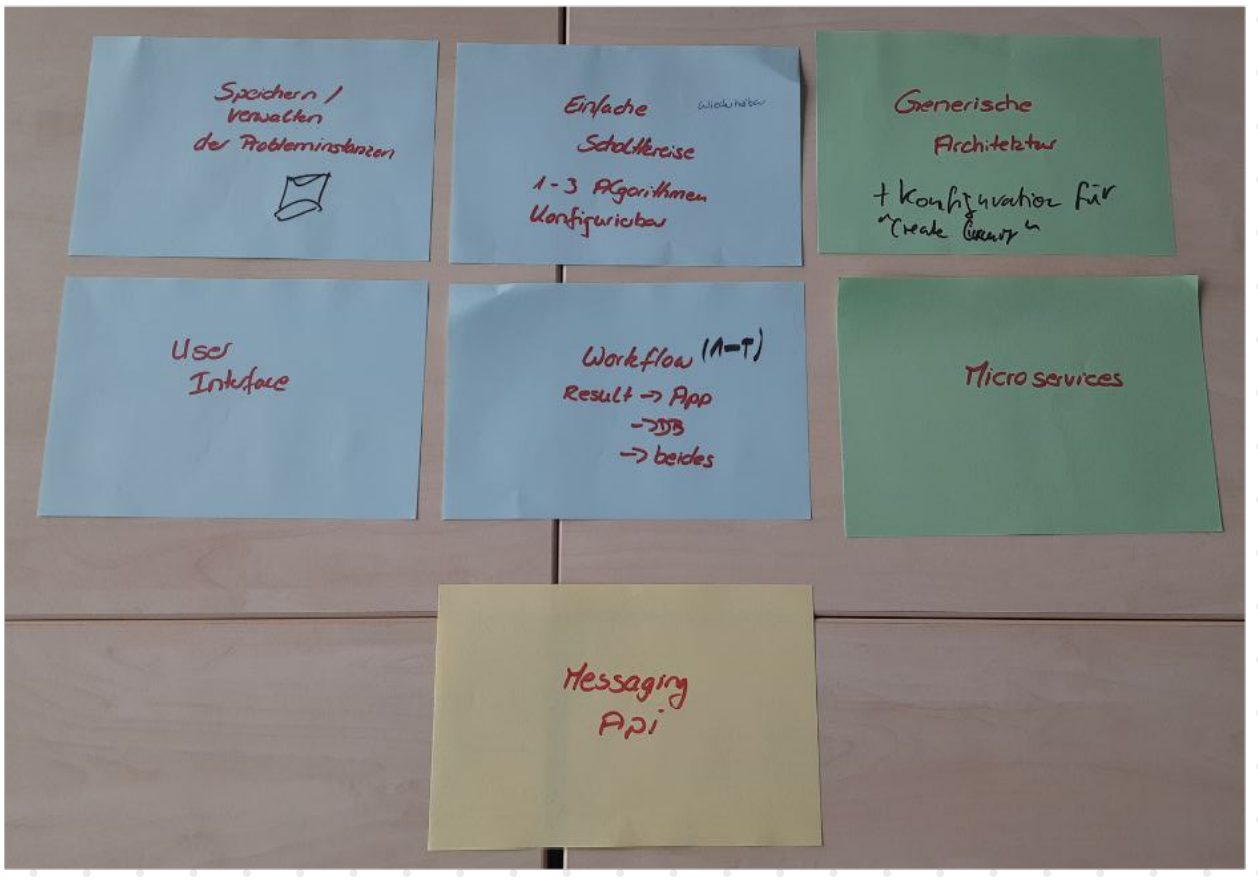
- Data Wrangling (Daten abbilden)
  ↳ Probleminstanzen speichern in Datenbank
  ↳ Bsp: Routenplanung (NP-schwer) → Logistik
- Verwaltung von Probleminstanzen
  ↳ Push-Trigger (automatische Anwendung Circuit) → Vision
  ↳ pull-basiert: Minimallösung
- Monitoring des Job-Status
  ↳ Jobs in GUI und/oder Datenbank anzeigen
  ↳ Result: Database | App | beides
- Workflow (Schritte 1-5)
- nur notwendige Daten codieren (Data Constraints)
- Frameworks für Microservices nutzen (↳ mögliche recherchieren)
  ↳ genutzte Technologien

↳ Kommunikation | Komponenten:
- Messaging API
- Microservices aufbrechen => möglichst klein
- verschiedene Varianten von Algorithmen => Konfiguration
- Bsp: Graph-Probleme (Grover-Algorithmus)
  => Codieren, Colouring (-> Verschränkung)
- Bsp: Codierung von Zahlen
- Anzahl Ergebnisse auslesen
- Bsp: Münzwurf (-> mehrfach werfen, Histogramm speichern)
- Bsp: Suche | Sortierung
- Komplizierte Query auf Quantencomputer => Datenbank

[=> Kurse: Hasso Plattner Institut]

- Simulation ideal bis ~30 Qubits
- Bsp: Clustern (automatische Aktualisierung in der Datenbank)
- Bsp: QDE: Circuit Generation

- TO DO: Minimalziele in GitLab-Board eintragen



- Buch für Quanten-Grundlagen
- Buch Python-Grundlagen

- Programmiersprache
  ↳ Python wegen Qiskit ? => Python und Java verknüpfen ?

- TASK: Architektur - Patterns für/mit Python

- Microservice - Architektur: Micronaut vs SpringBoot => Java oder Python ?
- Recherche:
  ↳ Messaging APIs / Rest API -> Flask ? (Kommunikation)
  ↳ Kafka => Kommunikation ? Logging ? (OPTIONAL)
  ↳ Auswahl: Micronauts / Flask / SpringBoot
  ↳ Chron - Jobs ?
  ↳ Tests ? (Nose test ?) => Test-Framework für Python
    => später festlegen
    => Integrationstest, nur für eine Komponente

- Dokumentation:
  ↳ Doc - Generierung für Python ?
  ↳ Style - Guide / Programmierrichtlinien
    => Einrückung
    => Variablennamen
    => Entwicklungsumgebung / IDE

- Docker Container ?! => nicht zwingend
- GitLab-Verzeichnisse erstellen
- requirements.txt pflegen
- Dokumentation: Installationsanleitung ! -> Ausführung durch Externe
- Cluster-Nodes ?

- Projektleitung organisieren: Anzahl Wochen ÷ 4

- Framework mit Polymorphismus
  ↳ Objektorientierung in Python ?
  ↳ Template-Methoden => Codierung ? Konfiguration ?
  ↳ Quanten Circuits als Blackbox

- passende Datenbank aussuchen (relational, Graph, ...)
- Grafiken mit: → draw.io → Visio (Microsoft 365 über Fernuni)
  → plantUML → staruML
- git-Client für Ubuntu suchen ?

- Aufgabenteilung: => später: Gruppen
  ↳ Maik: - Architekturpatterns Python
    - Technologien: Flask
    - Aufgaben in GitLab eintragen -> evtl. anderes Board
  ↳ Nina: - Test-Umgebung für Python recherchieren
    - Docker zum Laufen bekommen
  ↳ Stefan: - Generierung der Dokumentation in Python
  ↳ Andreas: - Docker zum Laufen bekommen
    - Micronauts, Springboot

- nächste Schritte:
  - ↳ Styleguide festlegen
  - ↳ Workflow modellieren (generische Architektur)
    - => Schaltkreis generieren + zurückgeben
    - => Messaging
    - => Microservices
  - ↳ Konzept für Python
    - => Template
  - ↳ mögliche Schaltkreise auswählen
    - Münzwurf
    - Codierung Graphen
  - ↳ Datenbank auswählen (nicht Fokus)

=> nächstes Treffen (mit Markus)
02.11.2023, 16:30 - 17:30

- Anwendung: -> sinnvoll für Quantenrechner
  - ↳ Input triggern
  - ↳ Codierung (im Simulator)
  - ↳ Rechg. über berechnungen
    - => Applikation und/oder User
  - ↳ Ergebnis in GUI anzeigen/ausgeben

- Showcase überlegen
  - ↳ Münzwurf } gemeinsam BSP
  - ↳ Graph Coloring } durchspielen

- Superposition und Verschränkung verstehen