Anlage zum Verwendungsnachweis

Sachbericht zum Verwendungsnachweis

TEIL I

Teil I: Kurzbericht (wird veröffentlicht)

1.1 Aufgabenstellung und wissenschaftlich-technischer Ausgangspunkt

Das Verbundprojekt QCStack (Förderkennzeichen 13N15921) hatte zum Ziel, einen hardwareagnostischen, technologieübergreifenden Middleware-Softwarestack für gatterbasierte Quantencomputer zu entwickeln – die entscheidende Schicht zwischen dem Quantenprozessor (QPU) und den höheren Anwendungsebenen. Das Konsortium umfasste drei Partner: Fraunhofer ITWM (Verbundkoordinator, Schwerpunkt Compiler und Scheduler), Qruise GmbH (Schwerpunkt Kalibrierung, Charakterisierung und Kontrolle, Hauptverwertungspartner) sowie die Freie Universität Berlin (FU Berlin, Schwerpunkt optimale Quantensteuerung).

Zum Zeitpunkt des Projektbeginns im Februar 2022 fehlte ein universell einsetzbarer, hardware-unabhängiger Quantencomputer-Softwarestack vollständig. Existierende Lösungen waren entweder herstellerspezifisch und proprietär (IBM Qiskit, AWS Braket) oder deckten nur Teilbereiche des Stacks ab. Zentrale Komponenten wie automatisierte Kalibrierung, kontinuierliche Rekalibrierung und QPU-Scheduling galten als Geschäftsgeheimnis der großen Cloud-Anbieter und waren als eigenständige Produkte nicht erhältlich. Das C³-Framework (Combined Characterization and Control), das die Gründer von Qruise entwickelt hatten, bildete den wissenschaftlichen Ausgangspunkt für das Teilvorhaben.

1.2 Ablauf des Vorhabens

Das Projekt lief über vier Jahre (Februar 2022 bis Dezember 2025) mit einer kostenneutralen Laufzeitverlängerung um zwölf Monate, die Mitte 2024 beantragt und bewilligt wurde. Der Halbzeitmeilenstein (MS2, Monat 18) wurde am 31. Juli 2023 erfolgreich beim Projektträger präsentiert und formal bestätigt. Die technische Arbeit gliederte sich in fünf Arbeitspakete (AP0–AP5). Der Qruise-Anteil konzentrierte sich auf AP2 (Inbetriebnahme und Rekalibrierung), AP3 (Compiler/Scheduler, unterstützende Rolle), AP1 (optimale Steuerung, Algorithmen-Schnittstellen) sowie AP4 (Systemintegration und Evaluation).

1.3 Wesentliche Ergebnisse

Qruise hat im Rahmen des QCStack-Projekts zwei Kernprodukte entwickelt, die den MVP-Status erreicht haben: QruiseML, eine JAX-basierte Simulations- und Modellierungsbibliothek (qruise-toolset 2.0), und QruiseOS, ein vollautomatisiertes Kalibrierungs- und Betriebssystem für Quantencomputer. Beide Produkte wurden auf sechs verschiedenen QPU-Plattformen und drei Kontrollhardware-Ökosystemen validiert. Für 2-Qubit-Gatter wurden Fidelitäten von ca. 98 % durch Closed-Loop-Kalibrierung erzielt. Die entwickelten Produkte werden zukünftig durch die Qruise GmbH kommerzialisiert.

TEIL II

Teil II: Eingehende Darstellung (wird veröffentlicht)

2.1 Ausgangssituation und Einordnung des Teilvorhabens

Der Aufbau und Betrieb eines Quantencomputers erfordert weit mehr als die Entwicklung der eigentlichen Quantenprozessor-Hardware. Zwischen dem physikalischen Chip und den Anwendungsalgorithmen liegt eine komplexe Softwareschicht, die im klassischen Computing einer Kombination aus Betriebssystem, Compiler-Toolchain, Gerätetreibern und Diagnose-Software entspricht. Für Quantencomputer existierte zu Projektbeginn keine generische, plattformunabhängige Lösung für diese Schicht. Jeder QPU-Entwickler war gezwungen, diese Software von Grund auf selbst zu implementieren – ein erheblicher Aufwand, der mehrere Personenjahre erfordert und außerhalb der Kernkompetenz der meisten Hardware-Teams liegt.

Das Teilvorhaben von Qruise GmbH adressierte genau diese Lücke. Unter dem Projekttitel „Softwarepaket für Kalibrierung, Charakterisierung und Kontrolle“ war Qruise für den zentralen Baustein des QCStack-Stacks verantwortlich: die Software, die einen Quantencomputer von der Hardware bis zur betriebsbereiten, kalibrierten Maschine führt und ihn dort kontinuierlich hält. Qruise brachte hierfür die Vorarbeiten aus dem Helmholtz-Validierungsfonds-Projekt und das C³-Framework (Combined Characterization and Control) ein, das experimentell an supraleitenden Qubits am Forschungszentrum Jülich validiert worden war und bereits Beschleunigungen von Quantengattern sowie die Identifikation unerwarteter Fehlerquellen demonstriert hatte.

Im Konsortium arbeiteten die drei Partner in enger Abstimmung zusammen: Die FU Berlin (Arbeitsgruppe Koch) war verantwortlich für die theoretisch-algorithmische Entwicklung optimaler Quantensteuerungsmethoden und lieferte Algorithmen für Quantendynamik und optimale Gattersynthese, die Qruise in seine Toolchain integrierte. Fraunhofer ITWM entwickelte den Compiler-Stack und den Scheduler und nutzte dabei die von Qruise bereitgestellten Hardware-Modelle und Kalibrierungsdaten als Eingabe für rauschempfindliche Kompilierung. Qruise übernahm als Haupt-Integrator die Rolle, alle Softwarekomponenten auf realer QPU-Hardware zu testen und zu einem funktionsfähigen Gesamtstack zusammenzuführen.

2.2 Durchgeführte Arbeiten nach Arbeitspaketen

2.2.1 AP0: Projektmanagement

Das Projektmanagement wurde durch das Fraunhofer ITWM als Verbundkoordinator geleitet. Qruise leistete hierfür zwei Personenmonate und trug aktiv zur Koordination und Qualitätskontrolle bei. Ein wesentliches Element war die Einrichtung wesentlicher technischer Arbeitstreffen zwischen den Partnern sowie eines gemeinsamen Messaging-Systems für schnelle Abstimmungen. Die halbjährlichen Konsortialtreffen und die Präsentation beim Halbzeitmeilenstein am 31. Juli 2023 wurden von Qruise mit vorbereitet. Der Halbzeitmeilenstein wurde vom Projektträger VDI Technologiezentrum mit Schreiben vom 25. August 2023 formal bestätigt; gegen eine Weiterführung des Projekts bestanden keine Bedenken.

2.2.2 AP1: Konzeption der optimalen Steuerung

In AP1, das unter der Leitung der FU Berlin durchgeführt wurde, war Qruise mit 15 Personenmonaten an allen drei Unterprojekten beteiligt. Die Zusammenarbeit basierte auf einem klar definierten Schnittstellenvertrag: Qruise stellte der FU Berlin detaillierte QPU-Modelle sowie eine Python-basierte Simulationsumgebung über die Konsortium-JupyterHub-Plattform zur Verfügung; die FU Berlin lieferte

implementierte Algorithmen in einem vereinbarten Standardformat zurück, die Qruise in sein Toolset integrierte.

In UP1.1 integrierte Qruise alle wesentlichen klassischen Techniken der modellbasierten optimalen Quantensteuerung in eine einheitliche Algorithmen-Library. Nutzer können darin verschiedene Pulsparametrisierungs-, State-Propagations- und Optimierungsroutinen beliebig kombinieren. Die Algorithmen wurden auf supraleitende Transmon-Bauelemente, Rydberg-Atome und weitere Hardwareplattformen angewandt und validiert. UP1.2 lieferte die Testumgebung: Qruise stellte der FU Berlin ein detailliertes Modell einer der QCStack-Referenz-QPUs zur Verfügung, das als Testbed für die in UP1.1 entwickelten Algorithmen diente. Insbesondere wurde der semi-globale Algorithmus für Quantendynamik in das Qruise-Toolset integriert.

Im Rahmen von UP1.3 wurden mittels optimaler Steuerung Pulssequenzen mit minimalem Gatterfehler berechnet. Diese optimal-gesteuerten Pulse dienen als »warmer Start« für die modellfreie Closed-Loop-Kalibrierung in AP2: Anstatt mit zufälligen oder manuell entworfenen Pulsen zu beginnen, erhält die Kalibrierungsroutine einen durch optimale Steuerung berechneten Ausgangspunkt, der bereits nahe am optimalen Lösungsraum liegt und die Konvergenz erheblich beschleunigt. Für die Generierung dieser Pulse benötigt UP1.3 als Eingabe präzise Modellinformationen aus der Charakterisierung (UP2.2), was die enge Verknüpfung beider Arbeitspakete verdeutlicht.

2.2.3 AP2: Software zur Inbetriebnahme und Rekalibrierung

AP2 bildete den Kern des Qruise-Teilvorhabens mit 62 von insgesamt 146 Qruise-Personenmonaten im Projekt. Die hier entwickelte Software ist direkt in die beiden Produkte QruiseML und QruiseOS eingeflossen und bildet die technische Grundlage für die kommerzielle Verwertung.

UP2.1 – Skalierbare parallele Simulation

Die Entwicklung eines physikalisch genauen, skalierbaren Simulators für Quantenprozessoren war eine der zentralen technischen Herausforderungen des Projekts. Eine vollständige Simulation einer großen QPU ist prinzipiell unmöglich, da der Zustandsraum exponentiell mit der Qubit-Zahl wächst. Der in QCStack verfolgte Ansatz nutzt überlappende Kacheln (Tiles): Die QPU wird in überlappende Teilbereiche unterteilt, jede Kachel separat simuliert, und die Ergebnisse werden zusammengeführt. Dieser Ansatz ermöglicht skalierbare optimale Steuerung und Charakterisierung für 1- und 2-Qubit-Gatter auf großen Geräten, ohne die vollständige QPU simulieren zu müssen.

Im Projektverlauf wurde die Simulations- und Optimierungsbibliothek qruise-toolset vollständig neu entwickelt und im Herbst 2024 als Version 2.0 veröffentlicht. Der entscheidende technologische Schritt war die Migration auf JAX, die wissenschaftliche Rechenbibliothek von Google. JAX ermöglicht JIT-Kompilierung (Just-in-Time Compilation), sodass Python-Code zu hochperformantem Maschinencode kompiliert wird, der auf CPUs, GPUs und TPUs läuft. Besonders wichtig ist die native Unterstützung für automatische Differenzierung: Gradientenberechnungen, die für optimale Steuerung und Modelllernen benötigt werden, erfolgen automatisch und exakt, ohne manuelle Ableitung. Die neue Bibliothek ist vollständig kompatibel mit der verbreiteten Open-Source-Bibliothek QuTiP und ermöglicht Nutzern die Wiederverwendung bestehenden Codes. In direkten Vergleichen mit konkurrierenden Simulationspaketen auf dem Markt zeigte qruise-toolset 2.0 vergleichbare oder bessere Leistung für die meisten Anwendungsszenarien bis zu ca. zehn Qubits.

Die parallelisierte Simulation für ein 2-Qubit-Gatterparameter erreichte eine Modellgenauigkeit von 99,6 % Übereinstimmung mit experimentellen Daten, was die Qualität des Simulationsansatzes bestätigt. Im späten Projektverlauf (2025) wurde mit der Entwicklung von qruise-toolset 3.0 begonnen, das auf einer Julia-Laufzeitumgebung aufbaut. Julia bietet native JIT-Kompilierung auf LLVM-Niveau, hochperformante automatische Differenzierung und umfassende Unterstützung für sparse Matrizen. Ziel ist die Simulation von ca. 15-Qubit-Einheitszellen, die notwendig ist, um QPU-Übersprechen und korrelierte Fehler korrekt zu modellieren. Erste Benchmarks zeigen, dass ein probabilistisches Hamilton-System für 15 Qubits mit 350-ns-Gate-Protokollen und 128 Anfangsbedingungen in ca. acht Stunden

simuliert werden kann. Eine Alpha-Version mit Python-Frontend und QuTiP-Kompatibilität wurde am Projektende bereitgestellt.

UP2.2 – Charakterisierung von mehreren Qubits

Die Charakterisierung eines supraleitenden Quantenprozessors ist ein komplexer mehrstufiger Prozess: Hunderte von Messexperimenten müssen in der richtigen Reihenfolge durchgeführt werden, da spätere Messungen auf den Ergebnissen früherer aufbauen. Bislang wurde dieser Prozess manuell geplant und ausgeführt. Ein wesentliches Ziel von UP2.2 war die vollständige Automatisierung dieses Prozesses mithilfe eines gerichteten azyklischen Graphen (DAG, Directed Acyclic Graph), der die Abhängigkeiten zwischen den Experimenten kodiert.

Qruise entwickelte ein Framework, in dem der Charakterisierungs-DAG automatisch aus den Eigenschaften der QPU generiert wird – für 1- und 2-Qubit-Experimente, ohne manuelle Eingriffe. Der DAG enthält alle Schritte, die für das vollständige Bring-up eines supraleitenden Transmon-Prozessors erforderlich sind, darunter Resonator-Spektroskopie, Qubit-Frequenz- und Anharmonizitätsbestimmung, T1- und T2*-Messungen, 1-Qubit-Gate-Charakterisierung mittels Quantenprozess-Tomographie (QPT) und Randomized Benchmarking (RB), 2-Qubit-Gate-Charakterisierung (QPT), Multi-Qubit-Charakterisierung einschließlich Kopplung und Crosstalk, Rauschspektroskopie der Qubit-Ansteuerung sowie Cryoscope-Messungen zur Analyse von Signalleitungsverzerrungen.

Ab Mitte 2023 wurde der vollständige Charakterisierungs-DAG täglich auf einer fünf-Qubit-Referenz-QPU ausgeführt. Automatisierte Fehleranalyse validiert die experimentell bestimmten Systemparameter gegen Schwellenwerte und verfolgt Änderungen im detaillierten QPU-Modell im Zeitverlauf, was die Grundlage für kontinuierliche Rekalibrierung bildet. Im weiteren Projektverlauf wurden die Charakterisierungsexperimente um Multi-Level-Auslesediskriminierung, qubit-qubit Fluss-Crosstalk, korrelierte Auslesefehler-Mitigation und vorkompensierende CZ-Chevron-Messungen erweitert. Die Parallelisierung der Charakterisierungsexperimente für nicht-benachbarte Qubits ermöglicht eine drastische Reduktion der gesamten Bring-up-Zeit um mehrere Größenordnungen gegenüber sequentiellem Vorgehen.

UP2.3 – Fehlerbudget und Modelllernen

Das Fehlerbudget eines Quantengatters beschreibt, welche physikalischen Effekte in welchem Maß zur Gesamtfehlerrate beitragen. Dieses Wissen ist für QPU-Entwickler von zentralem Interesse, da es die gezielte Verbesserung der Hardware ermöglicht: Wenn bekannt ist, dass 60 % des Gatterfehlers auf T1-Relaxation und 30 % auf Kontrollelektronik-Rauschen zurückzuführen sind, kann der Entwicklungsfokus entsprechend gesetzt werden.

Die ursprünglich geplante direkte Zerlegung von Betriebsstörungen in Modellursachen erwies sich als wissenschaftlich deutlich anspruchsvoller als erwartet. Die Identifikation trennbarer Fehlerquellen in Multi-Qubit-Systemen stellte sich als offenes Forschungsproblem heraus, das über den ursprünglichen Planungsstand hinausging. Qruise investierte erheblich mehr Ressourcen in diesen Bereich als geplant und entwickelte einen alternativen Ansatz, der über Modelllernen (Model Learning) arbeitet: Aus experimentellen Daten – insbesondere aus CPMG-Rauschspektroskopie, Flussabhängigkeit der Dephasierung und CZ-Chevron-Daten – werden Kopplungsparameter und Rauschbeiträge in den digitalen Zwilling der QPU eingelernt. Der verfeinerte digitale Zwilling ermöglicht dann die Quantifizierung der statistischen Distanz zwischen Modell und Experiment sowie die Extrapolation auf das Fehlerbudget. Bis Ende 2025 war dieser Ansatz abgeschlossen und validiert.

UP2.4 – Kalibrierung

Das Kalibrierungspaket implementiert die modellfreie, geschlossene Regelkreis-Optimierung (Closed-Loop Calibration) von Quantengattern direkt auf der Hardware. Im Gegensatz zur modellbasierten optimalen Steuerung benötigt die Closed-Loop-Kalibrierung kein präzises Gerätemodell: Sie optimiert Pulsparameter iterativ auf der Grundlage von Messungen am realen Gerät und konvergiert so zum optimalen Puls unabhängig von Modellungenauigkeiten. Damit ist sie besonders wertvoll für

Produktionsumgebungen, in denen schnelle Anpassung an Gerätevariabilität wichtiger ist als tiefes physikalisches Verständnis.

Qruise implementierte zunächst Closed-Loop-Kalibrierung für schnelle 1-Qubit-Gatter; die Ergebnisse wurden am APS March Meeting 2025 präsentiert. Parallel wurde eine randomisierte Benchmarking-Kalibrierung zur Feinabstimmung von 1-Qubit-Pulsformen implementiert. Der wesentlich anspruchsvollere Schritt war die Ausweitung auf 2-Qubit-Gatter: Auf einer QSolid QPU (Kontrollhardware: Zurich Instruments) wurde ein iSWAP-Gatter durch Closed-Loop-Kalibrierung auf ca. 98 % Fidelität optimiert; auf einer QuantWare Contralto QPU (Kontrollhardware: Quantum Machines/Qblox) wurde ein CZ-Gatter auf ebenfalls ca. 98 % Fidelität kalibriert. Diese Ergebnisse demonstrieren, dass QruiseOS plattformunabhängig hochwertige 2-Qubit-Gatter durch automatisierte Kalibrierung erzielen kann.

UP2.5 – Kontinuierliche Rekalibrierung und Geräteüberwachung

Quantenprozessoren unterliegen zeitlichen Drifts: Qubit-Frequenzen, Anharmonizitäten und Kopplungsstärken ändern sich auf Zeitskalen von Minuten bis Tagen. Ein produktionstüchtiger Quantencomputer muss kontinuierlich überwacht und rekalibriert werden, ohne den Betrieb für Nutzer-Jobs zu unterbrechen. UP2.5 implementierte diese Fähigkeit als eigenständigen Dienst in QruiseOS.

Das System verfolgt Systemparameter wie Qubit-Frequenz, Anharmonizität, T1- und T2*-Zeiten über mehrere Tage hinweg und löst automatisch Rekalibrierungsroutinen aus, wenn Parameter definierte Schwellenwerte überschreiten. Die Rekalibrierung ist mit dem Scheduler (AP3/UP3.3) verknüpft: Rekalibrierungsaufgaben mit hoher Priorität werden zwischen Nutzer-Jobs verschachtelt, sodass die QPU-Auslastung maximiert wird ohne die Nutzererfahrung zu beeinträchtigen. Kontinuierliche Kalibrierung und Geräteüberwachung wurden bis Mitte 2025 weit verbreitet getestet und in Produktions- und Testumgebungen bei mehreren QPU-Anbietern eingesetzt.

UP2.6 – Benutzeroberfläche und Deployability

Die Benutzeroberfläche von QruiseOS wurde konsequent auf die Anforderungen von QPU-Betriebsteams ausgerichtet. Das Echtzeit-Dashboard wurde vollständig neu gestaltet: Es bietet schnellen Zugriff auf die wichtigsten Betriebskennzahlen (Qubit-Status, aktuelle Kalibrierungsqualität, Scheduler-Auslastung, Drift-Indikatoren) und ermöglicht die gezielte Einschränkung von Workflows auf bestimmte Qubits oder Koppler. Ergänzend wurde eine neue Befehlszeilenschnittstelle (CLI) entwickelt, die flexible Automatisierung und Skripting für fortgeschrittene Nutzer ermöglicht.

Ein besonderer Fokus lag auf der Portabilität der Deployment-Lösung. Die Softwarearchitektur wurde vereinfacht, sodass QruiseOS über Docker Compose bereitgestellt werden kann – auch auf ressourcenbeschränkten Geräten mit minimaler Netzwerkkonfiguration. Dies reduziert den Installationsaufwand erheblich und senkt die Hürde für neue Anwender. Die Vereinfachung der Architektur hat zugleich den Wartungsaufwand in Produktionsumgebungen erheblich reduziert.

2.2.4 AP3: Compiler und Scheduler

Der Compiler und Scheduler stellte das Kernarbeitspaket des Partners Fraunhofer ITWM dar; Qruise war mit 48 Personenmonaten in unterstützender und leitender Funktion beteiligt. Die Zusammenarbeit folgte einer klaren Arbeitsteilung: ITWM verantwortete die Compiler-Kernimplementierung, während Qruise die physiknahen Aspekte beisteuerte und den Scheduler leitete.

In UP3.1 definierten ITWM und Qruise gemeinsam die Eingabe- und Ausgabe-Spezifikationen des Compilers. Qruise spezifizierte dabei insbesondere die Parameter, die direkt aus der zugrunde liegenden Physik stammen: die verfügbaren nativen Gatter der Hardware, den Konnektivitätsgraphen der QPU sowie die aktuellen Gatter-Fidelitäten aus der Kalibrierung. Diese Informationen übermittelt Qruise an ITWM im standardisierten Qiskit-Format (BackendProperties und BackendConfiguration), was eine nahtlose Integration ermöglicht. Die Schnittstelle stellt sicher, dass der Compiler in seiner endgültigen

Version rauschempfindliche Kompilierung durchführen kann: Er berücksichtigt dabei die aktuellen Gatterqualitäten und bevorzugt bei der Schaltkreisoptimierung leistungsfähigere Qubits und Koppler.

Qruise übernahm die Leitung der Scheduler-Implementierung (UP3.3), da der Scheduler eng mit der QPU-Steuerelektronik verknüpft ist und tiefes Wissen über die Hardware-Zeitskalen erfordert. Der QruiseOS-Scheduler implementiert eine priorisierte Task-Warteschlange und stellt sicher, dass die QPU maximal ausgelastet bleibt. Er enthält Mechanismen für Benutzerauthentifizierung und -autorisierung, Schaltkreis-Validierung sowie Vorab-Kompilierung von Jobs zur Minimierung der Latenz. Besonderes Augenmerk wurde auf die Verknüpfung mit der Rekalibrierungsfunktion gelegt: Rekalibrierungsaufgaben werden als hochpriore Jobs in die Warteschlange eingesteuert und zwischen Nutzer-Jobs ausgeführt. Ein Mehrbenutzer-Scheduler mit dieser Funktionalität wurde bereits 2023 in Betrieb genommen. Die Integration mit High-Level-Frameworks (Qiskit, Cirq) und der Steuerungshardware (UP3.4) wurde unter der Leitung von Qruise abgeschlossen.

2.2.5 AP4: Evaluation und Systemintegration

Qruise übernahm die Rolle des Haupt-Integrators für den QCStack-Softwarestack. Die Schnittstellendefinitionen zwischen allen Softwarekomponenten (UP4.1) wurden bis zum Halbzeitmeilenstein abgeschlossen. Die Schnittstellen umfassten die API zwischen den Algorithmen zur optimalen Steuerung (FUB) und der Kalibrierungssoftware (Qruise) sowie die Schnittstellen zwischen Compiler/Scheduler (ITWM) und dem Kalibrierungsstack (Qruise).

Für die Hardware-Software-Integration (UP4.2) stellte Qruise digitale Zwillinge von Referenz-QPUs bereit, die für Tests und Benchmarking ohne reale Hardware genutzt werden konnten. Ein dreistelliges Qubit-Subset der Referenz-QPU mit Rx-, Rz- und CZ-Gattern wurde als digitaler Zwilling implementiert; FUB und Qruise bestätigten die Übereinstimmung der Simulationsergebnisse für geschlossene Systemdynamik. ITWM nutzte die von Qruise bereitgestellten Gate-Informationen für die vollständige Compiler-Pipeline einschließlich physikalischer Qubit-Platzierung und Gatter-Routing.

Als Referenzschaltkreis für das Benchmarking wurde der Shor-Algorithmus verwendet. Die Evaluation (UP4.3) zeigte messbare Verbesserungen durch den ITWM-Compiler gegenüber einer Basisimplementierung in Gatteranzahl und Schaltkreistiefe. Implementierte Benchmarking-Tools umfassen Quanten-Prozess-Tomographie (QPT), Randomized Benchmarking (RB), interleaved Randomized Benchmarking (iRB) und Quantum Volume (QV), das auch auf Rydberg-Hardware getestet wurde. Der vollständige Kalibrierungs-, Charakterisierungs-, Gate-Optimierungs- und Scheduler-Stack wurde erfolgreich mit folgenden Kombinationen aus QPU und Kontrollhardware evaluiert: Rigetti QPU, QuantWare Contralto (Quantum Machines / Qblox), XeedQ, QSolid/Fraunhofer (Zurich Instruments), Chalmers University sowie TNO.

2.3 Wichtigste Positionen des zahlenmäßigen Nachweises

Die folgende Tabelle fasst die wesentlichen Finanzdaten des Qruise-Teilvorhabens gemäß dem finalen Verwendungsnachweis zusammen.

Position	Betrag (EUR)
Entstandene Kosten gesamt (bis einschl. 2025)	1.924.315,58
davon Personalkosten (Pos. 0837)	1.837.040,86
davon Reisekosten (Pos. 0838)	85.922,70
davon Sonstige direkte Kosten (Pos. 0850)	1.352,02
Eigenanteil Qruise (30 %)	577.294,67

Bewilligte Zuwendung (BMBF)	1.506.233,00
Gezahlte Bundesmittel	1.351.403,02
Verbleibender Anteil des Bundes (berechnet)	1.347.020,91
Überzahlung (zur Rückerstattung)	4.382,11

Die entstandenen Gesamtkosten von 1.924.315,58 EUR liegen um ca. 10,3 % unterhalb der Gesamtvorkalkulation von 2.151.762,00 EUR. Dies ist in erster Linie darauf zurückzuführen, dass keine vorhabenspezifischen Gegenstände oder Investitionsgüter angeschafft wurden und die Entwicklung ausschließlich durch Personaleinsatz erbracht wurde. Reisekosten in Höhe von 1.352,02€ entstanden im Wesentlichen für Projekttreffen. Da diese überwiegend remote ausgeführt wurden, entstanden hier kaum Kosten. Die bewilligte Zuwendung des BMBF betrug 1.506.233,00 EUR bei einer Förderquote von 70 %. Die tatsächlich gezahlten Bundesmittel belaufen sich auf 1.351.403,02 EUR; der berechnete verbleibende Anteil des Bundes beträgt 1.347.020,91 EUR, woraus eine geringfügige Überzahlung von 4.382,11 EUR resultiert, die zurückzuerstatten ist.

2.4 Notwendigkeit und Angemessenheit der geleisteten Projektarbeiten

Mit 95,5 % entfiel der weitaus größte Anteil der Projektkosten auf Personalkosten, was bei einem reinen Software-Forschungs- und Entwicklungsvorhaben ohne Investitionsgüter vollständig angemessen und üblich ist. Die Entwicklungsintensität war durch die Komplexität der technischen Aufgaben begründet: Die vollständige Neuentwicklung der Simulations- und Kalibrierungssoftware auf einer modernen, differenzierbaren Rechenbasis (JAX), die Entwicklung des automatisierten Charakterisierungs-DAG, die Implementierung von Closed-Loop-Kalibrierung für 2-Qubit-Gatter sowie die Integration und Validierung auf sechs verschiedenen QPU-Plattformen sind sämtlich Aufgaben, die tief spezialisiertes Fachwissen an der Schnittstelle von Quantenphysik, numerischer Mathematik und Softwareentwicklung erfordern.

Im Jahr 2022 wurden Arbeiten leicht vorgezogen, da Qruise-Personal temporär Ressourcen aus anderen Projekten freigestellt hatte. Dies führte zu einer leichten Überschreitung des kassengemäßen Rahmens im ersten Jahr (372.674,24 EUR verausgabt gegenüber 331.229,89 EUR kassengemäß verfügbar), was jedoch über die Gesamtlaufzeit ausgeglichen wurde. Eine kostenneutrale Laufzeitverlängerung um zwölf Monate (bis 31. Dezember 2025) war notwendig und wurde bewilligt: Sie ermöglichte es, Verzögerungen im Compiler-Arbeitspaket des Partners Fraunhofer ITWM, die auf Personalprobleme zurückzuführen waren, aufzufangen und die Gesamt-Stack-Integration mit dem ITWM-Compiler vollständig abzuschließen. Die zusätzliche Zeit wurde auch genutzt, um die 2-Qubit-Gate-Kalibrierung und das Fehlerbudget-Modelllernen abzuschließen und auf weiteren QPU-Plattformen zu validieren.

2.5 Fortschritte auf dem Gebiet des Vorhabens bei anderen Stellen

Die Wettbewerbslandschaft im Bereich automatisierter QPU-Kalibrierung und Inbetriebnahme-Software hat sich während der Projektlaufzeit erheblich weiterentwickelt und ist zum Projektende deutlich belebter als zu Beginn – was die Richtigkeit des von Qruise adressierten Problems unterstreicht.

Der direkteste Wettbewerber ist QuantrolOx. QuantrolOx hat mit Quantum EDGE einen neuen Industriestandard gesetzt und erreicht vollautomatisierte QPU-Charakterisierung und Inbetriebnahme mit Erstellung eines 2-Qubit-Gatters in unter 25 Minuten – eine Aufgabe, die bisher mehrere Experten auf PhD-Niveau bis zu einer Woche beschäftigt hat. Quantum EDGE kann sowohl on-premise als auch über die Cloud bereitgestellt werden. QuantrolOx fokussiert sich jedoch ausschließlich auf automatisiertes

Bring-up und Charakterisierung und bietet weder physikalisch exakte Simulation noch Modelllernen oder Fehlerbudgetierung – Kernbestandteile von QruiseML.

Quantum Machines (Israel) ist primär ein Anbieter von Quantencontroll-Hardware (OPX-Serie), hat aber mit QUAlibrate ein vollständig Open-Source-Kalibrierungsframework veröffentlicht, das vollständige Qubit-Inbetriebnahme in unter einer Minute und parallelisierte Multi-Qubit-Kalibrierung ermöglicht – allerdings ausschließlich für Systeme, die auf Quantum Machines-Hardware laufen. QUAlibrate ist damit kein hardware-agnostisches Produkt, sondern ein Ökosystem-Tool zur Kundenbindung. QruiseOS wurde 2026 erfolgreich am Israeli Quantum Computing Center (IQCC) auf Basis von Quantum Machines OPX1000 integriert und automatisierte das Bring-up eines 21-Qubit-QPUs in ca. 15 Minuten – Qruise und Quantum Machines sind damit in der Praxis eher komplementär als direkt kompetitiv.

Q-CTRL (Australien) hat seinen Fokus im Projektverlauf erweitert: Im Mai 2025 lancierte Q-CTRL mit Boulder Opal Scale Up eine autonome Kalibrierungslösung für On-Premise-Quantencomputer, die in Partnerschaft mit QuantWare und TreQ Push-Button-Inbetriebnahme von QPUs ermöglicht. Damit ist Q-CTRL nun direkter Wettbewerber im Kalibrierungssegment. Q-CTRL setzt dabei auf einen physics-informed-AI-Ansatz und bleibt ohne eigene tiefe Simulationsschicht, wie sie QruiseML bietet. LinkedIn

IBM hat Qiskit weiterentwickelt, bleibt aber strukturell an die eigene Hardware gebunden. Riverlane und Horizon Quantum Computing fokussieren sich auf Fehlerkorrektur-Middleware, eine Ebene oberhalb des in QCStack adressierten Stacks. ParityQC adressiert die Kompilierung für Parity-basierte Hardware.

Zusammenfassend hat sich das Wettbewerbsfeld zwischen 2022 und 2025 stark belebt, was die strategische Relevanz des Marktsegments bestätigt. Kein Wettbewerber kombiniert jedoch zum Projektende alle fünf Kernfähigkeiten der Qruise-Produkte in einer hardware-agnostischen Plattform: physikalisch exakte skalierbare QPU-Simulation (QruiseML), vollautomatisierte DAG-basierte Charakterisierung, Closed-Loop-Kalibrierung für 1Q- und 2Q-Gatter, Modelllernen und Fehlerbudgetierung sowie kontinuierliche Rekalibrierung mit integriertem Geräte-Monitoring. Die Nachfrage nach dieser integrierten Lösung wird durch die aktive Nutzung durch externe Partner bestätigt.

2.6 Voraussichtlicher Nutzen und Verwertbarkeit der Ergebnisse

Die im Projekt entwickelten Softwareprodukte adressieren einen schnell wachsenden Markt. Der weltweite Quantencomputing-Gesamtmarkt wurde für 2025 auf ca. 3,5 Milliarden US-Dollar geschätzt und soll bis 2030 auf über 20 Milliarden US-Dollar anwachsen, bei jährlichen Wachstumsraten von über 40 %. Staatliche Investitionen allein sind bis Anfang 2025 auf ca. 10 Milliarden US-Dollar an zugesagten Mitteln gestiegen. Für QPU-Hardware-Entwickler – die primäre Zielgruppe von Qruise – stellt die Entwicklung eines eigenen Betriebsstacks eine erhebliche Ablenkung von ihrer Kernkompetenz dar. Qruise schätzt den Aufwand auf mehrere Personenjahre pro Hersteller; QruiseML und QruiseOS ermöglichen es, diesen Aufwand drastisch zu reduzieren und stattdessen auf eine markterprobte Lösung zurückzugreifen.

Qruise plant drei Geschäftsmodell-Säulen: Direktlizenzierung der Softwarepakete an QPU-Hersteller, wiederkehrende Lizenzeinnahmen für OS-Scheduler und Rekalibrierungskomponenten im Rahmen eines „QCStack inside“-Modells sowie professionelle Dienstleistungen für Bring-up, Kalibrierung und kundenspezifische Anpassungen. Erste Kundenbeziehungen und Testpartnerschaften, die aus der Projektarbeit entstanden sind (QuantWare, QSolid, QuTech TNO, Chalmers), bilden die Grundlage für die unmittelbar folgende Kommerzialisierungsphase. Die Gründer von Qruise sind seit über einem Jahrzehnt in der Quantencomputing-Community bekannt und respektiert, was den Zugang zu potenziellen Kunden erheblich erleichtert.

Neben der kommerziellen Verwertung sind die Open-Source-Komponenten der entwickelten Bibliotheken (qruise-toolset, PyPI) unmittelbar als Forschungsinfrastruktur nutzbar. Andere BMBF-geförderte Quantencomputing-Projekte sowie EU-Flaggschiffprojekte können auf diesen Komponenten aufbauen und ihre QPU-Hardware schneller in Betrieb nehmen.

2.7 Veröffentlichungen und Dissemination

Im Rahmen des QCStack-Projekts bzw. in engem Zusammenhang damit wurden folgende Aktivitäten zur Verbreitung der Ergebnisse durchgeführt. Auf dem APS March Meeting 2025 wurde die Closed-Loop-Kalibrierung für schnelle 1-Qubit-Gatter in einem Fachvortrag präsentiert. Das qruise-toolset wird als Open-Source-Bibliothek auf PyPI bereitgestellt und umfasst umfangreiche Dokumentation sowie Anwendungsbeispiele für supraleitende, Ionenfallen-, Rydberg- und NV-basierte Quantencomputer. Technische Blog-Beiträge und API-Dokumentation zu QruiseOS und QruiseML wurden veröffentlicht. Die Fortschritte des Projekts wurden halbaljährlich in Konsortialtreffen präsentiert.