

MQV 2026 Annual Meeting Implementation Workshops

Session 1 — 6th Oct 13:55-15:20 (85')

Workshop 1 · SQQC systems & applications on SQQC QPU

Workshop lead: Klaus Liegener

Moderators: Max Werninghaus (SQQC systems) · Klaus Liegener (applications)

For: NP1/SQQC, the hardware team and the algorithm/application researchers. **Required:** Ignacio Cirac, Michael Knap, Michael Hartmann, Markus Müller, Jeanette Lorenz and Wolfgang Maurer.

Abstract: Two parts. (1) SQQC systems (~10 min, Max Werninghaus): a short pitch of the superconducting systems SQQC plans to expose from January 2027 — which demonstrator cryostats/QPUs (scaling toward 50 qubits) and further available systems — and in which access format (cloud access through the Munich Quantum Portal, exposed via MQSS/QDMI as QaaS / HPCQCaaS backends). (2) Applications: which novel quantum-algorithm developments from NP5 can run on the NP1 devices available from January 2027.

Keywords: superconducting QPU, algorithms, co-design

Guiding questions

- SQQC systems: Which systems will be exposed, in which access format, and on what timeline (from Jan 2027)?
- Applications: Which novel quantum-algorithm developments from NP5 could run on the NP1 devices available from Jan 2027?

Intended outcome: A shortlist of concrete first applications for the 2027 devices, mapped to the exposed SQQC systems.

Workshop 2 · QDMI, Compiler Infrastructure & Quantum Pathway Program

Workshop leads: Lukas Burgholzer & Yannick Stade

Moderators: Yannick Stade & Lukas Burgholzer (QDMI) · Damian Rovara & Lukas Burgholzer (MLIR/QIR) · Marco De Pascale & Jorge Echavarria (QPP)

For: NP4/Software, NP1, NP2, MQSC, planqc, Peak Quantum, LRZ. **Required:** Ercüment Kaya / Mortaza Aghtar (LRZ), Marco De Pascale, Akshay Bhosale (LRZ), Luigi Iapichino (LRZ), Berti Florea (TUM/DLR), Martin Schulz (TUM/LRZ).

Abstract: This workshop covers three closely linked threads on software interfaces and access to MQV quantum resources: QDMI, the MLIR/QIR-based compiler infrastructure, and the Quantum Pathway Program (QPP). QDMI is the hardware-abstraction interface used to expose superconducting and neutral-atom systems; key questions are which hardware constraints physicists and experimentalists must expose to the compiler, and what is missing to support broader use cases such as integrating entire compute centers as a single exchangeable device, building on a draft RFC for a modular revamp of QDMI. On the compiler side, an existing pipeline is already capable of emitting QIR; the central question is which restrictions the QIR output should carry, or more broadly, what a suitable hardware exchange format looks like, tying back to what QDMI must expose. QPP aims to give industry straightforward, low-barrier access to MQV resources through a QDMI-powered federated network — the open question is what makes that access genuinely straightforward in practice.

Keywords: software-hardware interface, compiler infrastructure, hardware access, QDMI, exchange format

Guiding questions

- QDMI (priority): How to evolve the interface? Which bits are missing?
- MLIR/QIR/compiler: How to connect to the MQV hardware platforms? Which bits are missing?
- QPP: Giving industry access to quantum computing via the QDMI-powered federated network — what is missing?

Intended outcome: A prioritized list of QDMI gaps, the compiler-infrastructure requirements to close them, and the missing pieces for federated industry access via QPP.

Session 2 — 7th Oct 17:05-18:20 (75')

Workshop 3 · QEC control stack & SC qubit software

Workshop leads: Max Werninghaus · Jonas Winklmann & Burak Mete · Johannes Zeiher

Moderators: Max Werninghaus (SC software stack) · Jonas Winklmann & Burak Mete (control HW) · Johannes Zeiher (compilation)

For: NP1–4, all gGmbH units, MQSC, Peak Quantum, planqc, Zurich Instruments. **Required:** Petr Zapletal, Jorge Echavarria, Burak Mete, Akshay Bhosale.

Abstract: The full control-and-software stack for the superconducting systems, in three parts. SW stack: where the line runs between the MQSS and the backend software, and how modules move across it. Control HW for QEC: the classical support architecture for modular and hierarchical error correction. Compilation for error-corrected QC: strategies and challenges of compiling for logical operation. Its main aim is to map out the strawman (classical support) architecture for QEC.

Keywords: quantum error correction, quantum software architecture, compilation

Guiding questions

- Software stack: What is the line between MQSS and backend software? How do we move modules across?
- Control HW: How to design the classical support architecture for modular and hierarchical error correction? (strawman)
- Compilation: Strategies and challenges for compilation for error-corrected QC.

Intended outcome: An agreed MQSS ↔ backend boundary, a strawman control architecture for QEC, and the compilation requirements that follow.

Workshop 4 · Chip integration & electronics

Workshop lead: Christian Schneider (NP1)

Contributing part-leads: Elias Meltzer & Hans Adel (packaging & wiring) · Matthias Althammer (on-chip routing) · Thomas Thönes & Erkan Isa (cryo-electronics)

Moderator: Christian Schneider

For: NP1/SQQC, Peak Quantum, Quint.

Abstract: This workshop covers physical integration and electronics. It is organized around three efforts and driven by the requirements of the QPU scaling roadmap. We start from the scaling targets (how many

and which qubits are expected when, and with which performance requirements) and derive specifications for three areas:

- (1) Packaging & wiring: functionalities, line count, superconducting readout lines (important for error correction), flex cabling and magnetic shielding, towards easier-to-use next-generation packages for 100+ qubits.
- (2) On-chip signal routing: filters, non-reciprocal devices and amplification, including routes enabled by new materials.
- (3) Cryogenic control: cryo-CMOS and SFQ/AQFP for flux-bias and qubit control, focusing on the interfaces to the qubit chip and to the room-temperature control and software stack.

Keywords: magnetic shielding, packaging & wiring, flex cables, on-chip routing, filtering, non-reciprocal devices, cryogenic control, cryo-CMOS, SFQ, AQFP, flux-bias control, qubit-per-line scaling

Guiding questions

- **Packaging & wiring:** Which functionalities, line counts and heat-load budgets do next-generation packages need, inside and outside the cryostat, to reach 100+ qubits? Where do PCB, flex cabling, cryostat wiring and room-temperature wiring set the limits?
- **On-chip routing:** Which filters, non-reciprocal devices and amplifiers must be integrated on chip or in the package, and what do new materials enable that discrete components cannot? What are realistic timelines?
- **Cryogenic control:** What are the requirements for flux-bias and qubit control with cryo-CMOS and SFQ/AQFP, including power output, resolution, noise, channel count and dissipation? At which generation do they enter the roadmap? How are the interfaces defined between cryo-control and the qubit chip, and between cryo-control and the room-temperature control and software stack?

Intended outcome: A consolidated requirement/interface specification across packaging, wiring, on-chip components and cryo-electronics.

Session 3 — 8th Oct 11:10-12:25 (75')

Workshop 5 · How to work as an ecosystem

Workshop lead: Thomas Luschmann

Moderators: Amit Devra · Max Werninghaus · Rui Pereira / Christopher Eichler

For: The whole ecosystem — NP leads and researchers, gGmbH units, MQV start-ups, industry partners, RTOs.

Abstract: A shared workshop on how MQV partners can work together as an ecosystem on an operational level. The goal is to understand the role of, and services offered by, the gGmbH, the MQV start-ups, and the industry partners and RTOs. What collaboration models and interfaces between the partners can be established to enable joint projects and demonstrators? We also aim to develop a framework for infrastructure usage: where the interfaces between start-ups, RTOs and NP researchers are (fabrication, characterization, tool access), what user-access rules and practical formats make the distributed infrastructure usable, and how to formalize the start-up-MQV relationship (including who qualifies as an MQV start-up).

Keywords: collaboration, services, joint projects, shared infrastructure

Guiding questions

- How does the gGmbH operating model work (common across pillars, with small specifics), and how does the ecosystem engage with it?
- What does the ecosystem require from the gGmbH units?
- Which collaboration models and interfaces are offered or planned to be offered by start-ups and RTOs to enable the demonstrators?
- How is the distributed infrastructure used across the ecosystem, and can we formalize access rules, formats and the start-up-MQV relationship?

Intended outcome: A shared, concrete picture of how the ecosystem works together — the gGmbH offering, co-development models, and infrastructure-access rules/formats.