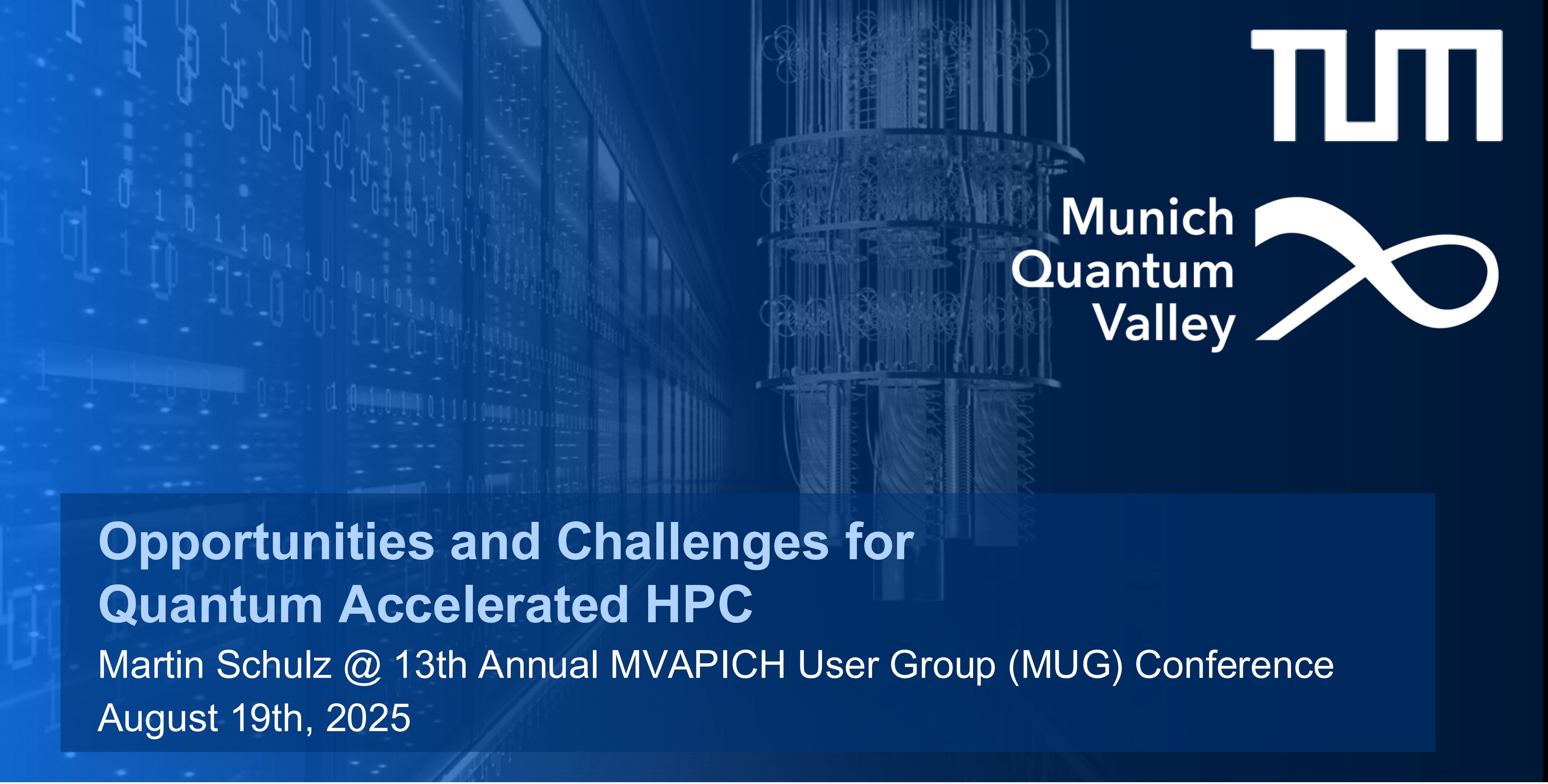


Opportunities and Challenges for Quantum Accelerated HPC

Martin Schulz @ 13th Annual MVAPICH User Group (MUG) Conference

August 19th, 2025

The Promise of Quantum Computing

System working on quantum mechanical principles

- Superposition = Multiple States at one time (until measured)
- Entanglement = Correlation between states in a system
- Probabilistic

Basic unit = 1 Qubit

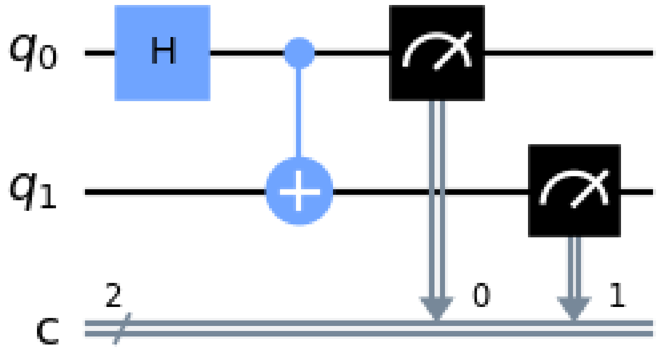
- Can be in superposed and/or entangled state
- When read: 0 or 1 only (rest collapses)

Programming with gates

- Operations on qubits
- Different systems have different gates

Quantum Advantage

- Some quantum algorithms require exponentially less steps than classical algorithms



From: https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.QuantumCircuit

What is all the Fuss About?

Applications for Quantum Computing

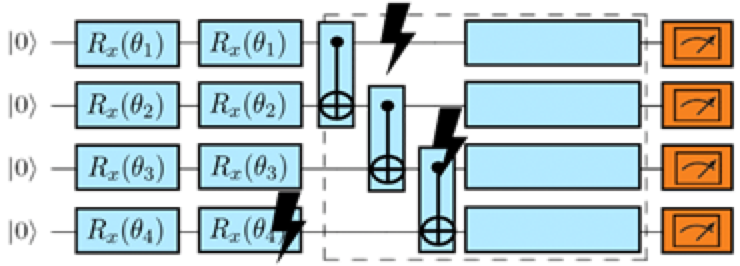
Main Application Domains

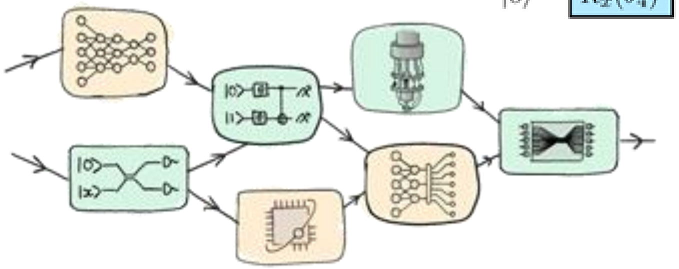
- Quantum simulation of quantum systems
- Quantum optimization
- Quantum machine learning
- Quantum linear systems

Challenges:

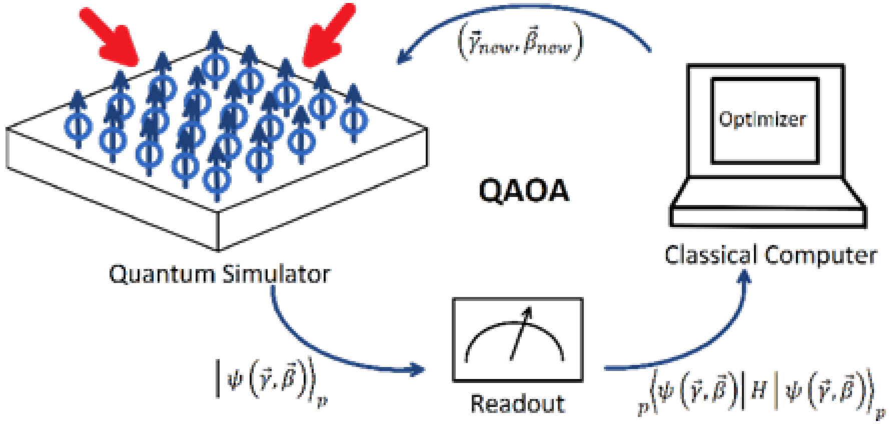
- Small number of qubits
- Noisy systems
- Only few working algorithms
- Specialized programming
- Still treated as physics experiments
- Need for HPC integration

$$\mathcal{N}(\rho) = \sum_{i=0}^{n} K_i \rho K_i^\dagger$$





PENNYLANE



$(\vec{\gamma}_{new}, \vec{\beta}_{new})$
Optimizer
QAOA
Quantum Simulator
Classical Computer
$|\psi(\vec{\gamma}, \vec{\beta})\rangle_p$
Readout
${}_p\langle\psi(\vec{\gamma}, \vec{\beta})|H|\psi(\vec{\gamma}, \vec{\beta})\rangle_p$

The Munich Quantum Valley initiative develops quantum computation and quantum technologies in Bavaria.

Superconducting. Ion. Neutral Atom. Quantum-HPC.

BAdW BAYERISCHE AKADEMIE DER WISSENSCHAFTEN DLR FAU Friedrich-Alexander-Universität Erlangen-Nürnberg Fraunhofer LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN MAX PLANCK GESELLSCHAFT Technische Universität München TUM

Mission of MQV

The primary goal of the Munich Quantum Valley initiative is **developing and operating competitive quantum computers** in close cooperation with strong industry partners and visionary start-ups and making them available for a broad range of applications.

BAdW BAYERISCHE AKADEMIE DER WISSENSCHAFTEN

DLR

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

Fraunhofer

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

MAX PLANCK GESELLSCHAFT

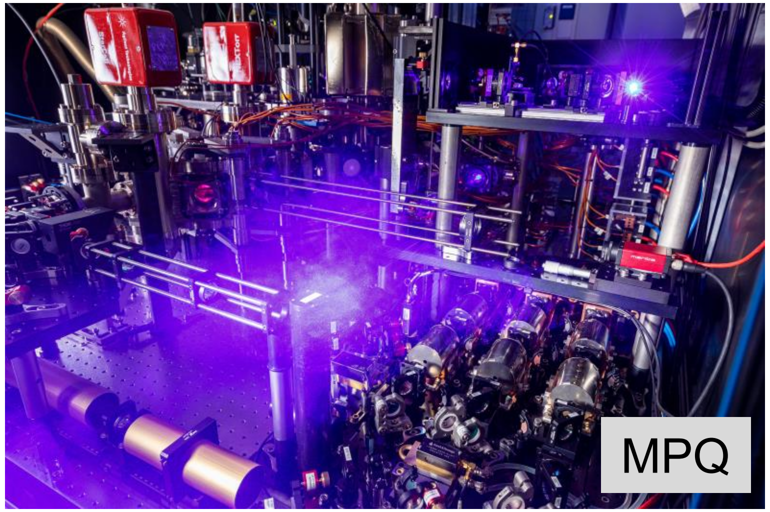
Technische Universität München TUM

Munich Quantum Valley

Full Stack Quantum Computing

Towards a Sustainable Eco-System

Neutral Atoms



MPQ

planqc

creating quantum computers atom by atom

Our processors use atoms in artificial crystals of light: the fastest way to scale to useful quantum advantage.

Superconducting

WMI

Peak Quantum

IQM

Software Stack

TUM / LRZ

MUNICH QUANTUM SOFTWARE COMPANY

BAdW BAYERISCHE AKADEMIE DER WISSENSCHAFTEN

DLR

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

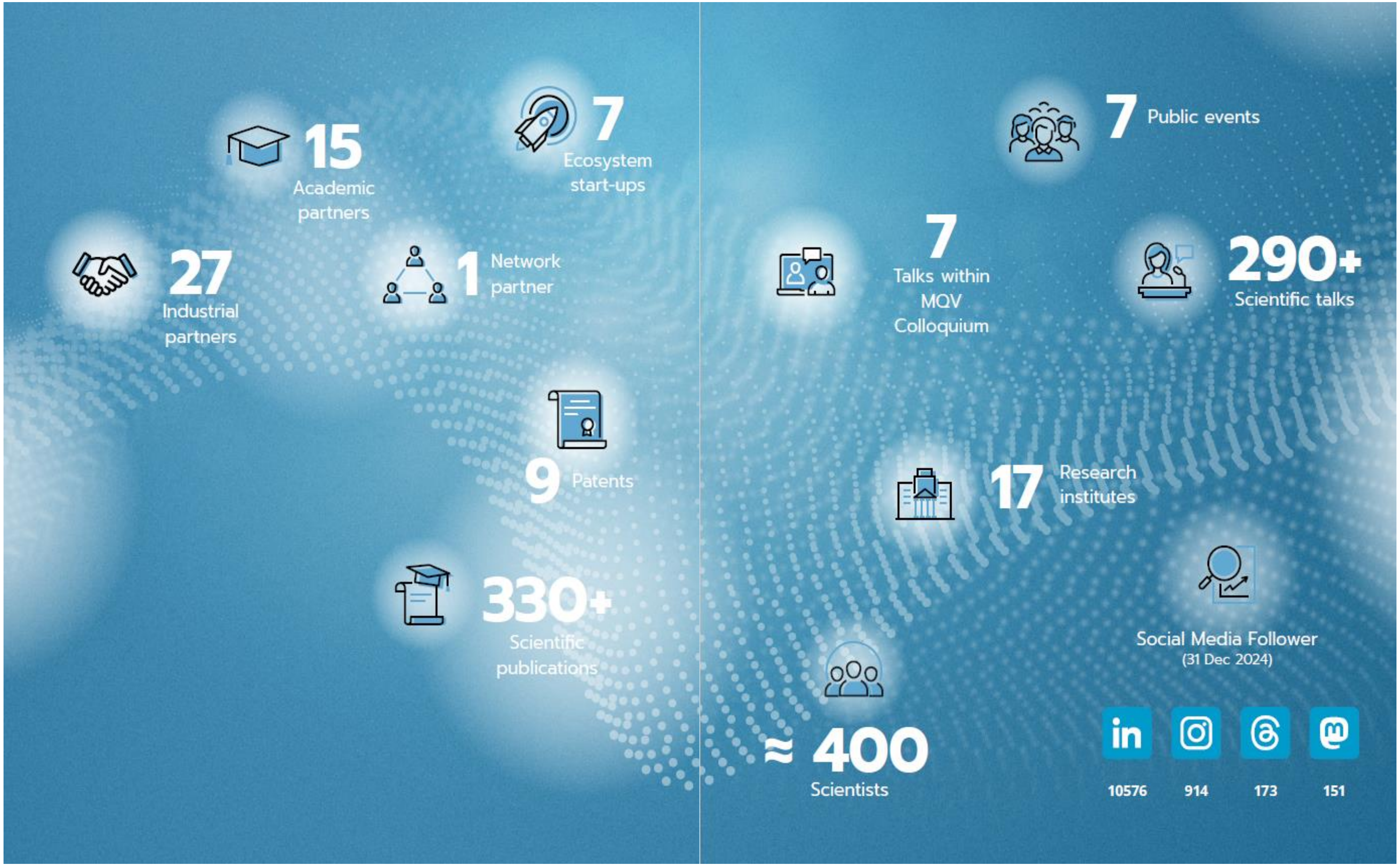
Fraunhofer

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

MAX PLANCK GESELLSCHAFT

Technische Universität München TUM

The MQV in Numbers



15 Academic partners

7 Ecosystem start-ups

27 Industrial partners

1 Network partner

9 Patents

330+ Scientific publications

7 Public events

7 Talks within MQV Colloquium

290+ Scientific talks

17 Research institutes

Social Media Follower (31 Dec 2024)

in	Instagram	Threads	Mastodon
10576	914	173	151

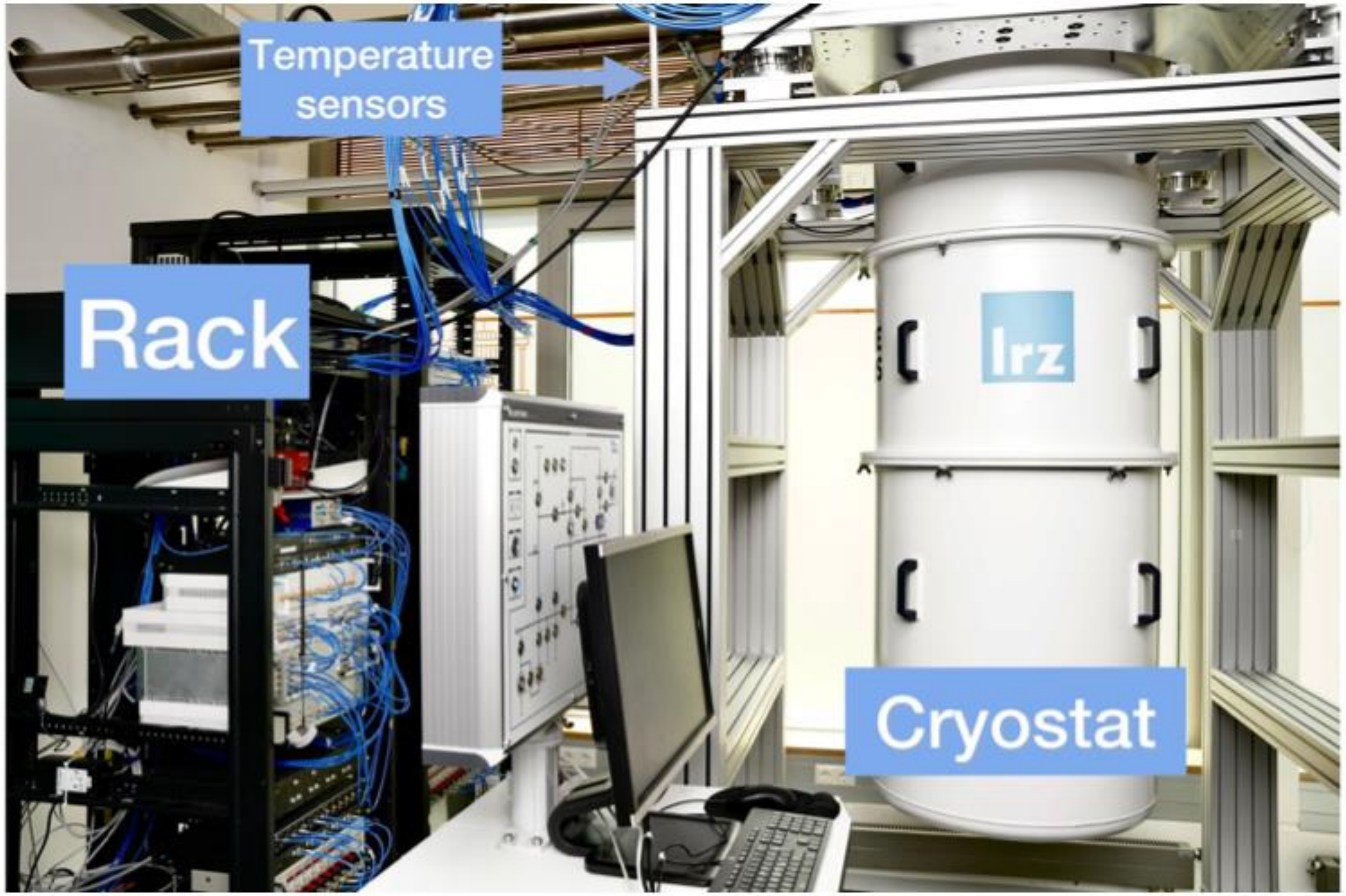
≈ 400 Scientists

… and growing

BAdW BAYERISCHE AKADEMIE DER WISSENSCHAFTEN · DLR · FAU Friedrich-Alexander-Universität Erlangen-Nürnberg · Fraunhofer · LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN · MAX PLANCK GESELLSCHAFT · Technische Universität München TUM

First ”Commercial” Systems at a Compute Center

Examples in the LRZ Quantum Integration Centre (QIC)



IQM @ LRZ

AQT @ LRZ

Quantum computers will not replace HPC.

⚠

Quantum computers will not replace HPC.

Quantum *accelerators* are HPC.

Why HPC-QC?

Quantum
Computing
=
High-Performance
Computing

New compute capability that adds to the supercomputing portfolio.

Strategy: Quantum Computing as Accelerator for HPC

Quantum Computing as a stand-alone system not viable at growing scales

- Complex control systems
- Data staging and post-processing
- Targeted towards very specific workloads and kernels
- Tight interactions needed for variational algorithms and some coupled workflows
- Complex compilation and runtime environment with high demands need HPC

Usage as accelerator for HPC workloads

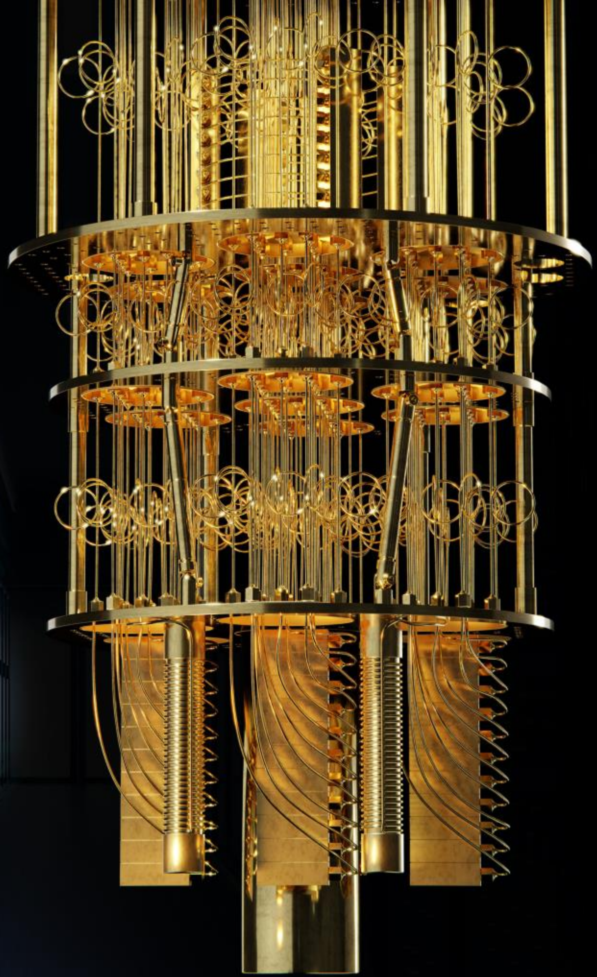
- Intended for fine-grained kernels within larger applications or workloads
- Similar to other accelerators, on-node (like GPUs, FPGAs) or dissaggregated (like AI HW)

Consequences: HPC and QC as a single HPCQC system

- Requires tools and models to extract application components relevant for acceleration
- Requires easy access for HPC community to QC programming (or library usage)
- Requires integration into a single programming environment
- Requires unified user access/management/experience
- Requires close hardware integration (single system) for latencies and management

Challenges

- Hardware Integration of the QC control system

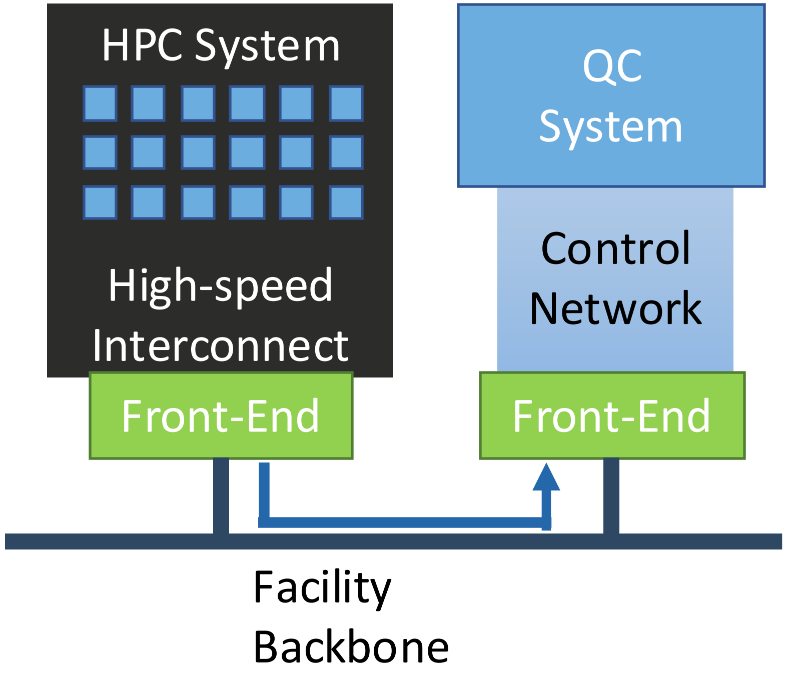


Why HPC-QC?

Quantum
Computing
=
High-Performance
Computing

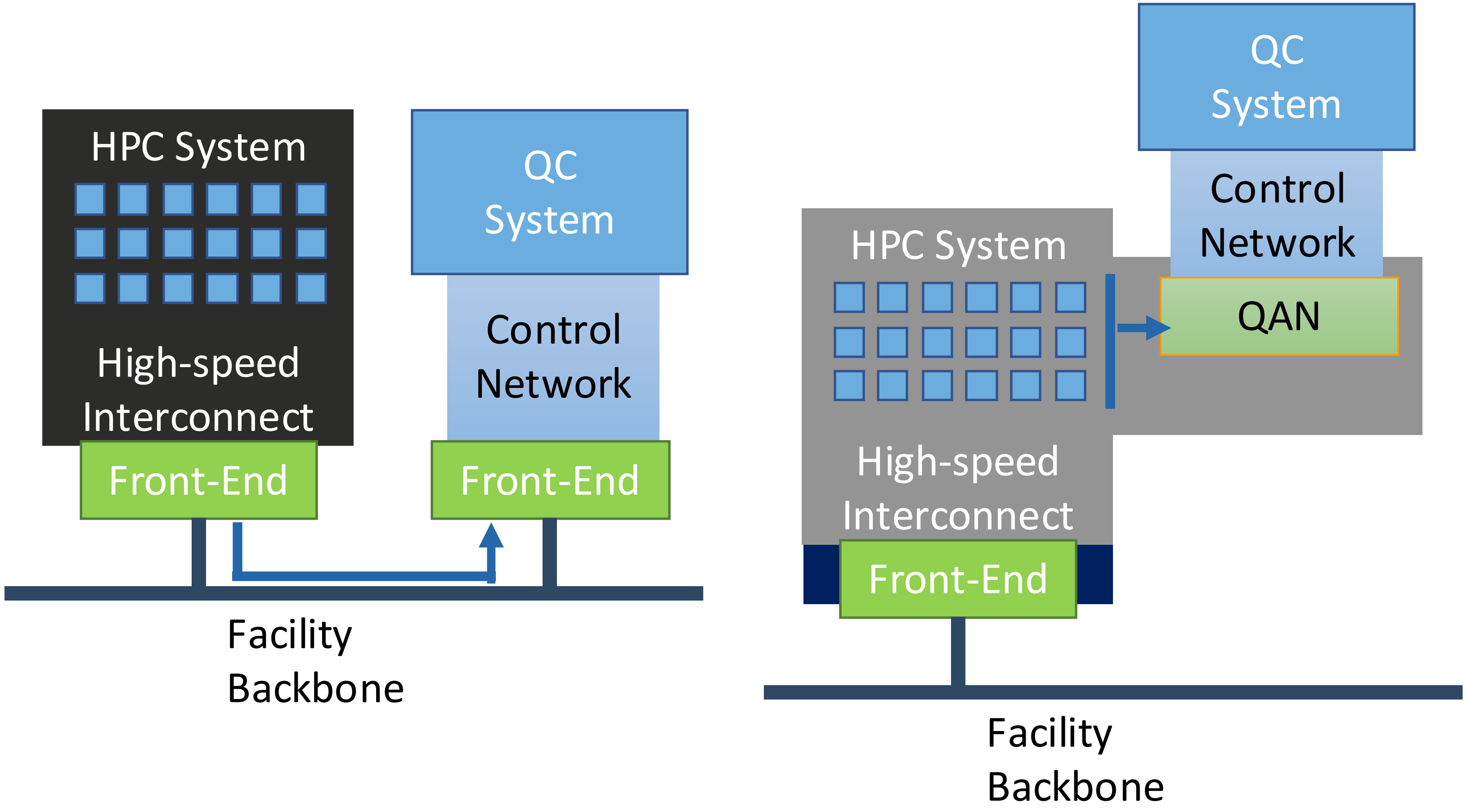
New compute capability that adds to the supercomputing portfolio.

Reducing the Gap Between Host and Accelerator



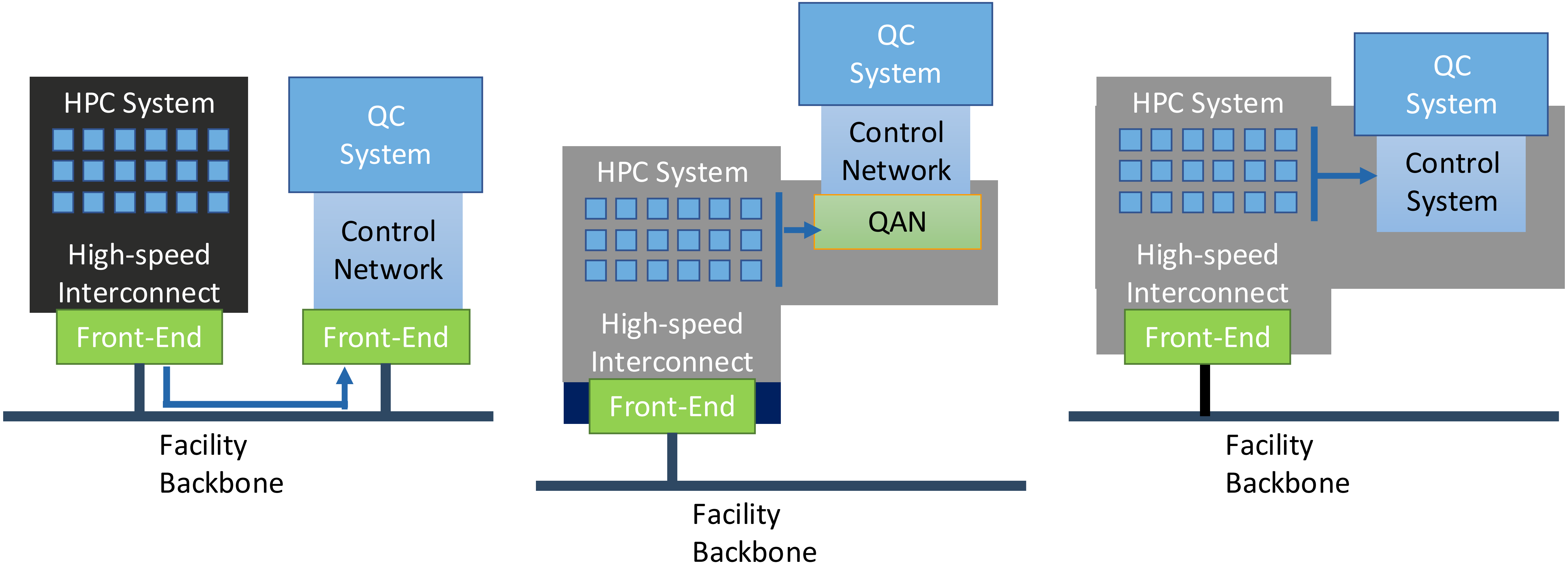
Evolution from network integration

Reducing the Gap Between Host and Accelerator



Evolution from network integration to system image integration

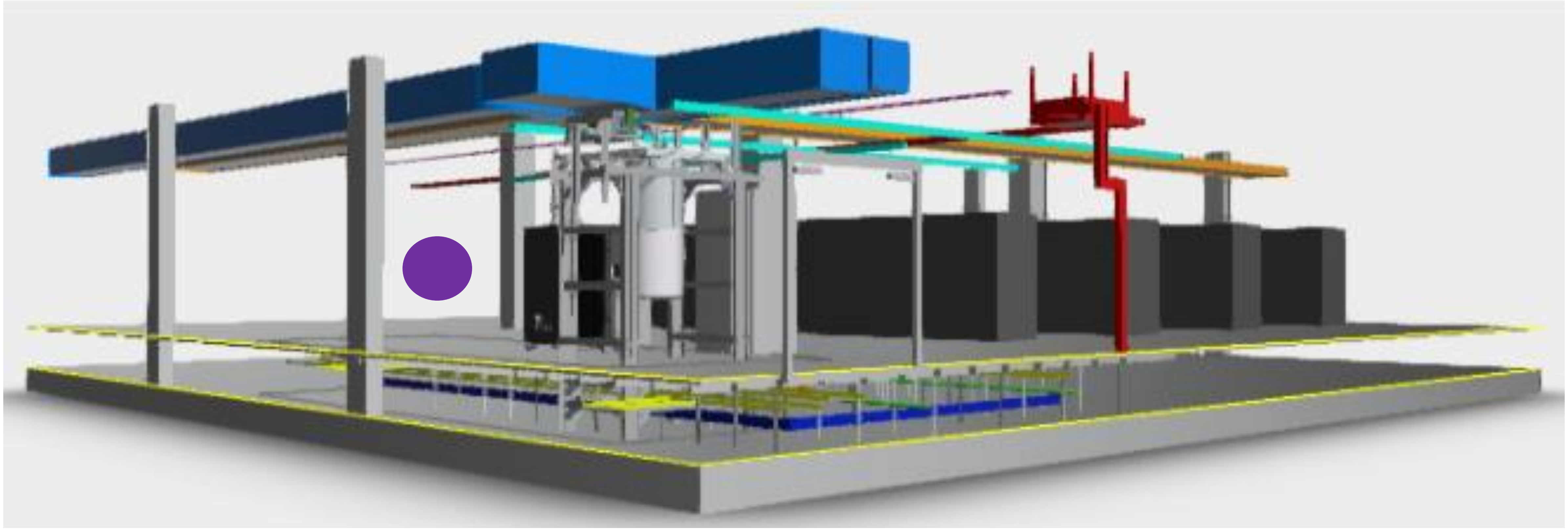
Reducing the Gap Between Host and Accelerator



Evolution from network integration to system image integration to on-node integration

LRZ QC and HPCQC Integration

Location of Production QC systems in LRZ Compute Cube



Planned location of System Q-Exa (shown) and Euro-Q-Exa system (to the left of Q-Exa, purple dot).

SPONSORED BY THE

Federal Ministry
of Education
and Research

SPONSORED BY THE

Federal Ministry of Education and Research

Q-Exa

German QC Demonstrator
20 Qubits built by IQM

@ Leibniz Supercomputing Centre
on the same floor as our HPC systems

Inaugurated June 18, 2024

Started with early user phase

Transitioning to standard operations now

SPONSORED BY THE

Federal Ministry of Education and Research

Q-Exa

Friendly User Pilot Phase
July - October 2024

Quantum Computing and Technologies Department
Leibniz Supercomputing Centre

lrz Leibniz Supercomputing Centre
of the Bavarian Academy of Sciences and Humanities

June 18, 2024
Inauguration of Q-Exa

Quantum-Accelerated HPC
underway

Achieved first submissions of hybrid application (VQE) via HPC compute node to quantum system using co-located, on prem HPC and QC systems.

EuroHPC‘s Euro-Q-Exa, hosted at LRZ

lrz Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

EURO-Q-EXA

PRACTICAL, SCALABLE AND RELIABLE EUROPEAN
QUANTUM-ACCELERATED EXASCALE-CLASS COMPUTING

EuroHPC Joint Undertaking

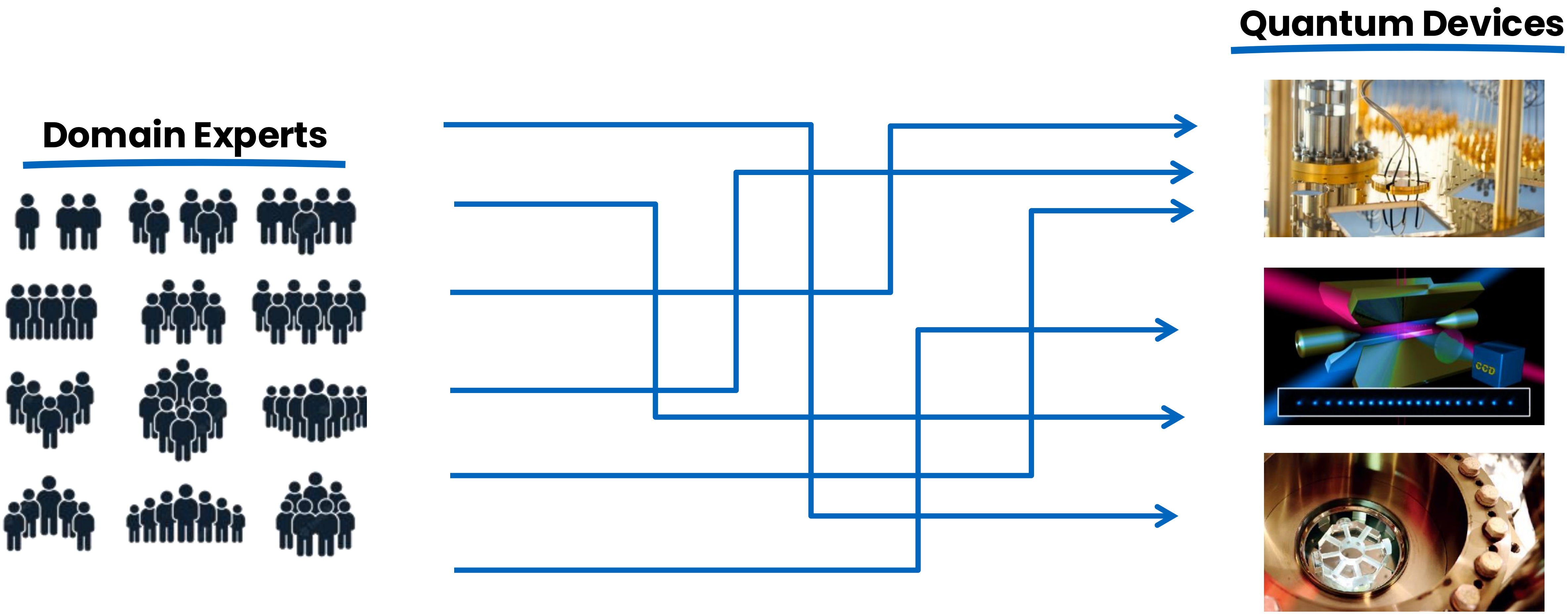
SPONSORED BY THE
Federal Ministry of Education and Research

Bayerisches Staatsministerium für Wissenschaft und Kunst

- Two-stage system delivery, all superconducting
  - 54 qubit in 2025
  - 150+ qubit in 2026
- Progressively tighter HPCQC integration and demands on vendor
- Academic and industrial use

What about the Software Stack?

The Role of Software



Quantum Devices

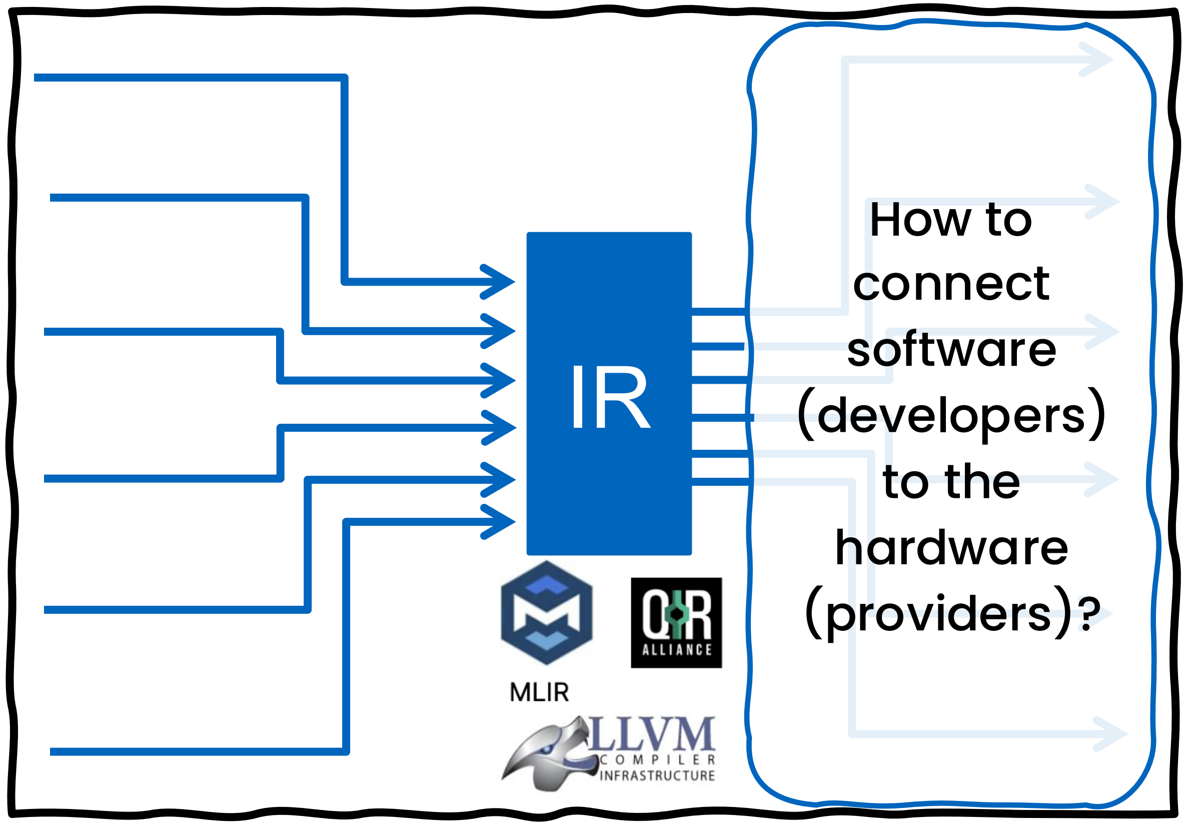
Domain Experts

The Role of Software

MQSS Munich Quantum Software Stack

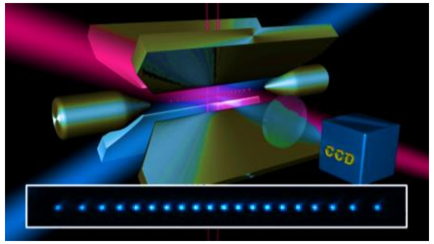
Domain Experts

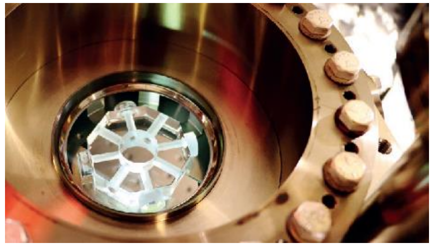
Munich Quantum Software Stack (MQSS)



How to connect software (developers) to the hardware (providers)?

Quantum Devices

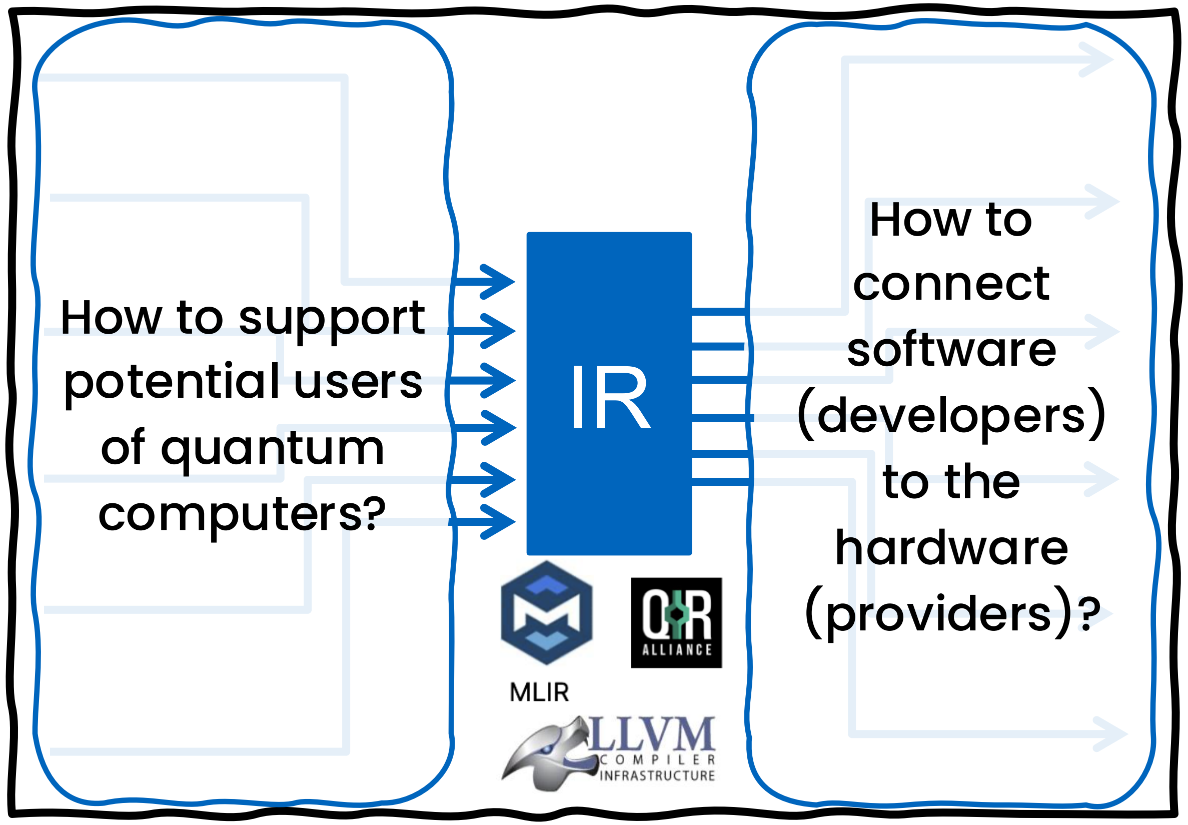




The Role of Software

Domain Experts

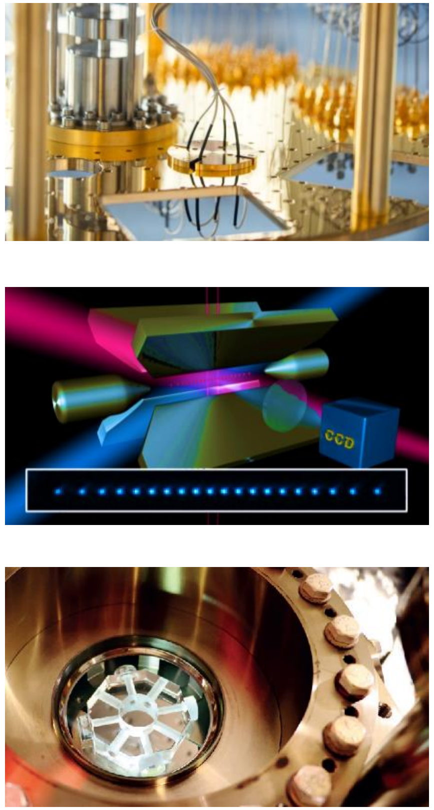
Munich Quantum Software Stack (MQSS)



How to support potential users of quantum computers?

How to connect software (developers) to the hardware (providers)?

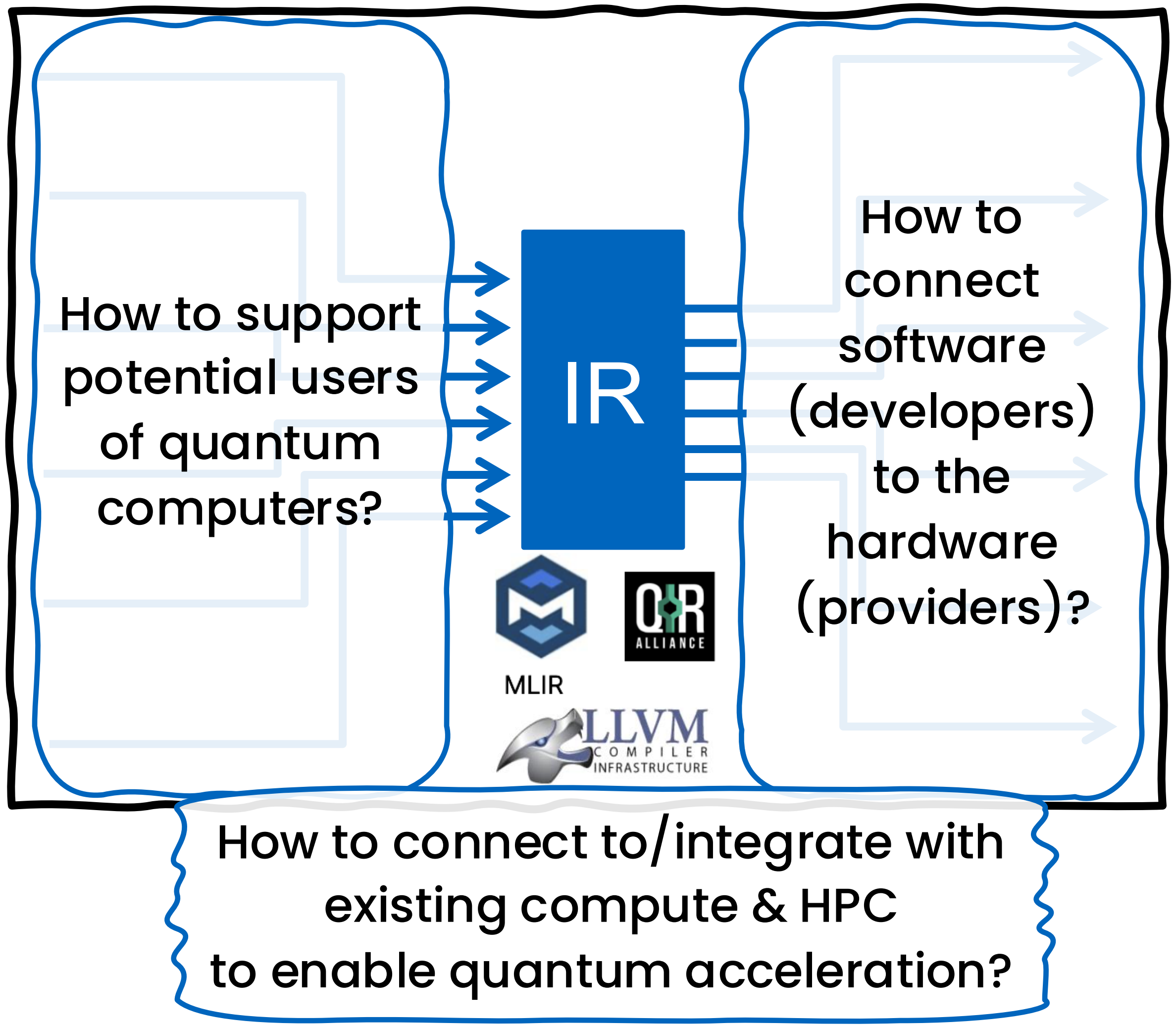
Quantum Devices



The Role of Software

Domain Experts

Munich Quantum Software Stack (MQSS)

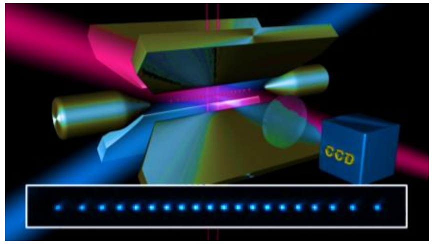


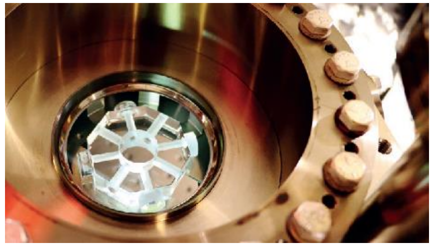
How to support potential users of quantum computers?

How to connect software (developers) to the hardware (providers)?

How to connect to/integrate with existing compute & HPC to enable quantum acceleration?

Quantum Devices





MQSS Requirements

Support a wide range of QC modalities

- Find common abstraction among modalities
- Query properties from concrete systems where needed

Support a wide range of programming models and abstractions

- Decouple programming model from support stack, compiler and back-end
- Enable models from existing Python/Scripting models to high-performance approaches

Support HPCQC integration

- Integrate QC into the existing HPC ecosystem as a specialized accelerator
- Hybrid programming abstractions building on/working with traditional HPC MPI+X approaches

Support efficient resource utilization

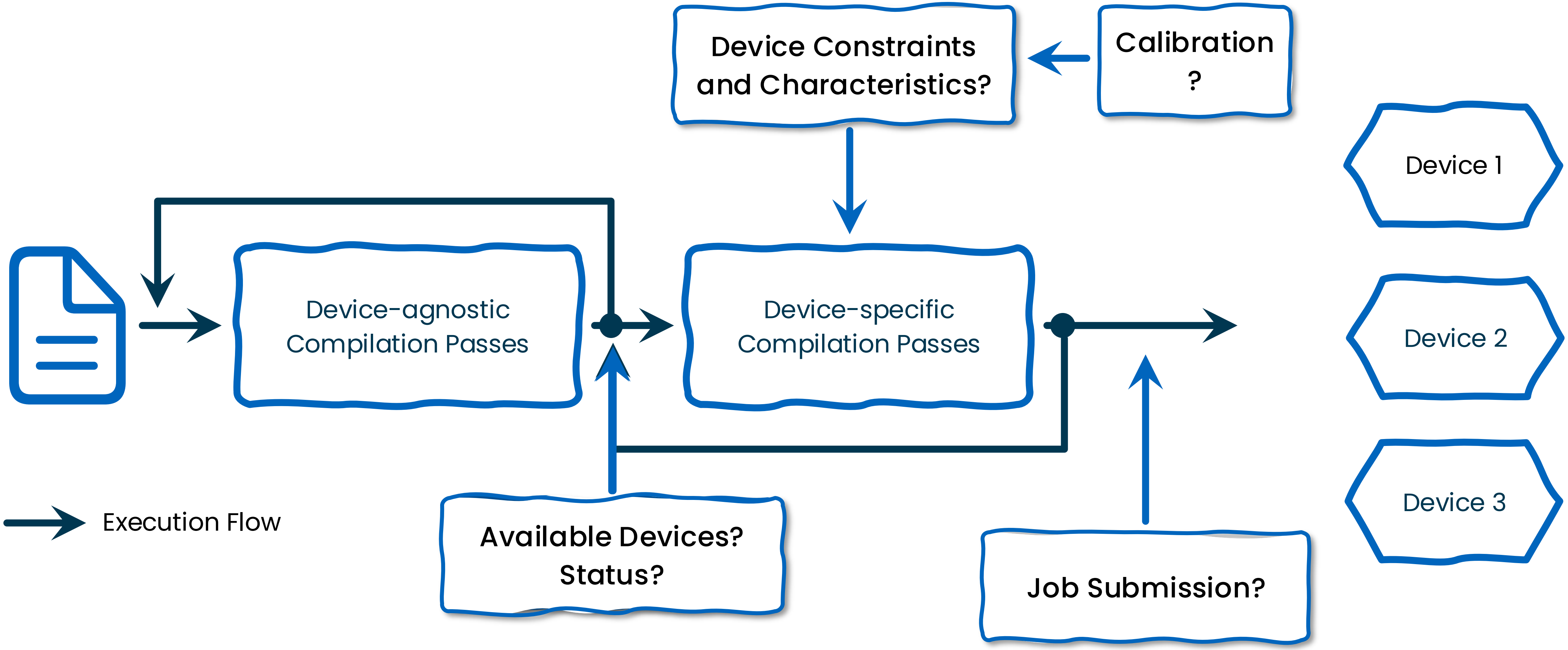
- Efficient, hierarchical scheduling combined with resource and runtime predictions
- Optimize quantum compilation and optimization on modern hardware

Support easy system management

- Comprehensive monitoring and Online Data Analytics
- Easy user and system management through a configurable, modern portal interface

Support internal and external users and installations

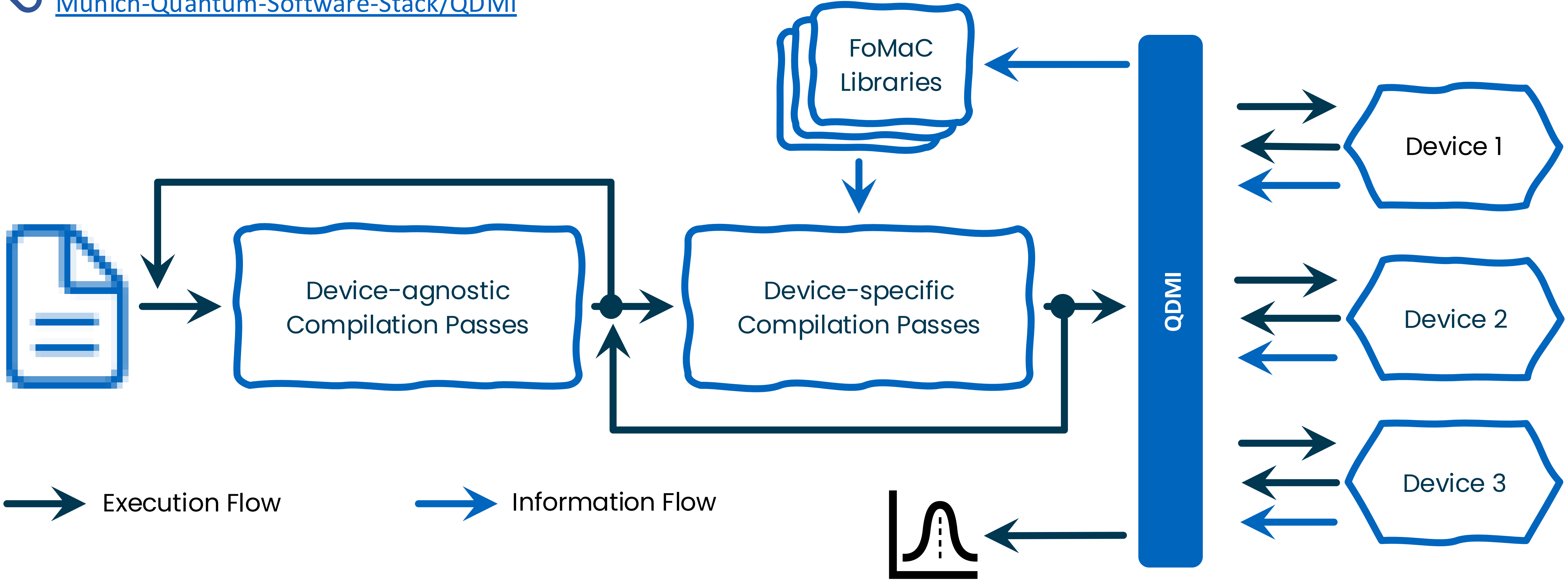
Software 🤝 Hardware


Device Constraints and Characteristics?
Calibration ?
Device-agnostic Compilation Passes
Device-specific Compilation Passes
Device 1
Device 2
Device 3
Available Devices? Status?
Job Submission?
Execution Flow

Software 🤝 Hardware

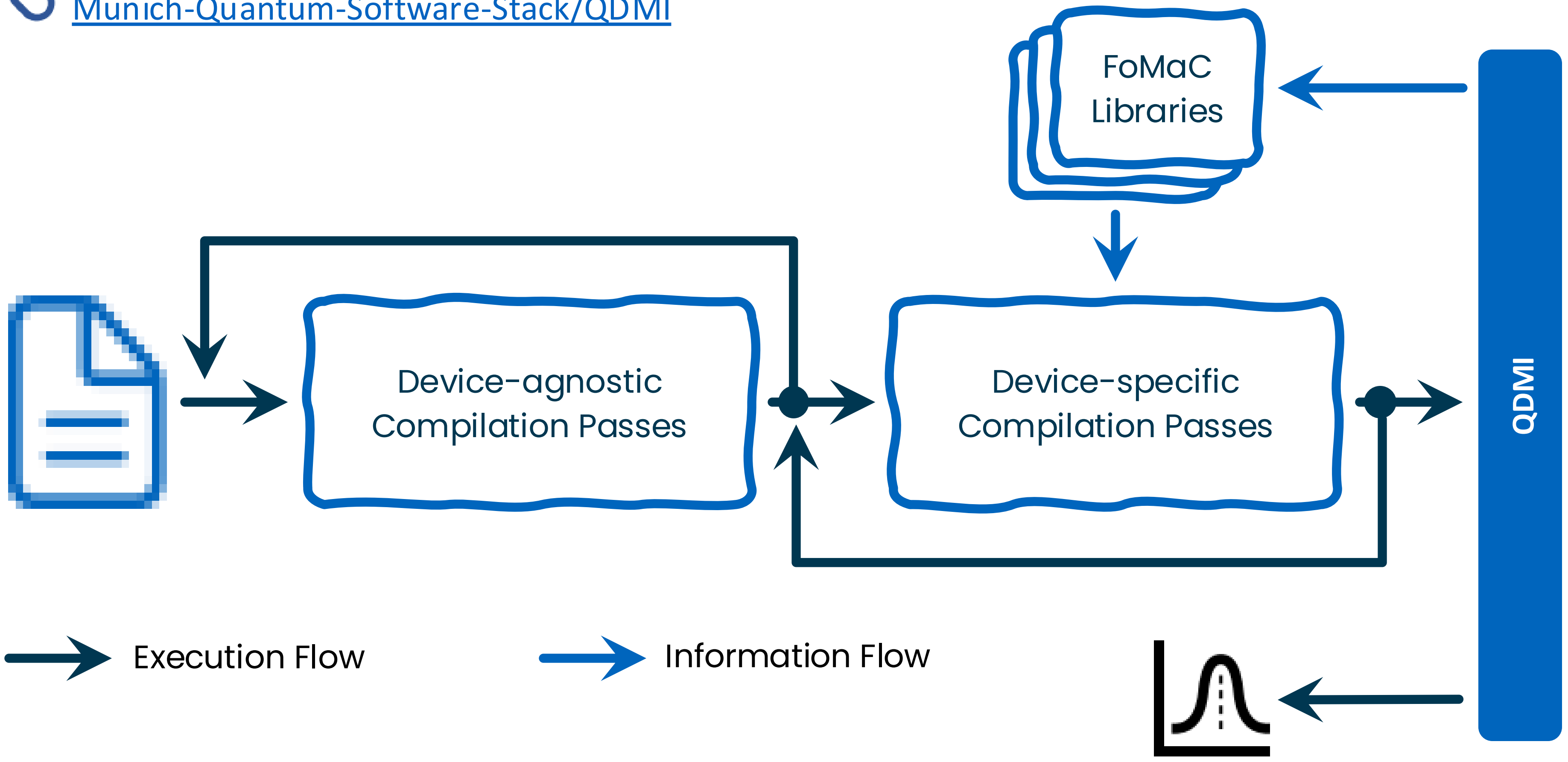
MQSS Munich Quantum Software Stack

[Github.com/ Munich-Quantum-Software-Stack/QDMI](Github.com/Munich-Quantum-Software-Stack/QDMI)



Software 🤝 Hardware

[Github.com/ Munich-Quantum-Software-Stack/QDMI](https://github.com/Munich-Quantum-Software-Stack/QDMI)


FoMaC Libraries
Device-agnostic Compilation Passes
Device-specific Compilation Passes
QDMI
Execution Flow
Information Flow

MQV System Back-ends

- WMI / K1
- MPQ / K3

- PlanQC
- PeakQ

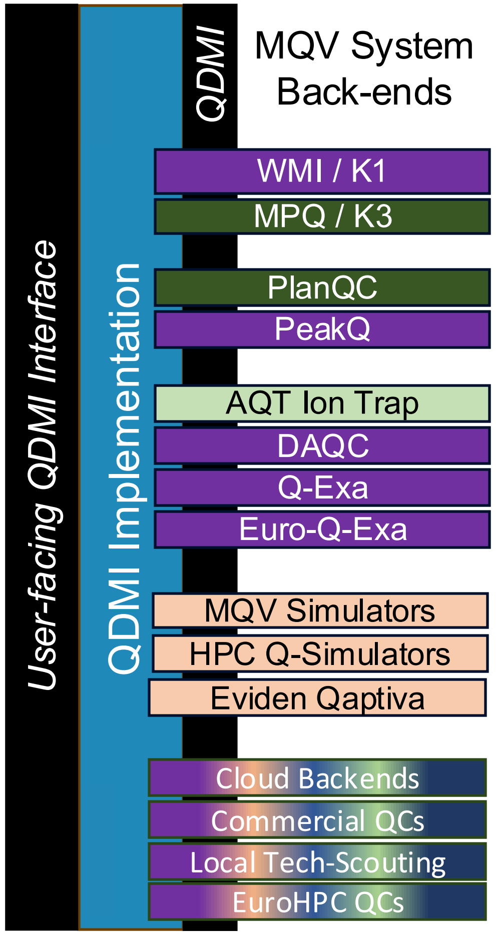
- AQT Ion Trap
- DAQC
- Q-Exa
- Euro-Q-Exa

- MQV Simulators
- HPC Q-Simulators
- Eviden Qaptiva

- Cloud Backends
- Commercial QCs
- Local Tech-Scouting
- EuroHPC QCs

Integration of a Wide Range of Backends

Domain Experts



MQSS Core: Back-End *Interfaces /APIs* *Plugins by Modality*

QDMI Interface / Properties

C-API for all 3 parts

- Inspired by MPI / OpenCL
- Versioned and intended as backwards compatible

Intended connections

- Vendors as providers
- The quantum software stack as users

Plugin concept for backends

- Selection interface to pick backend
- Multiple existing backends in the works

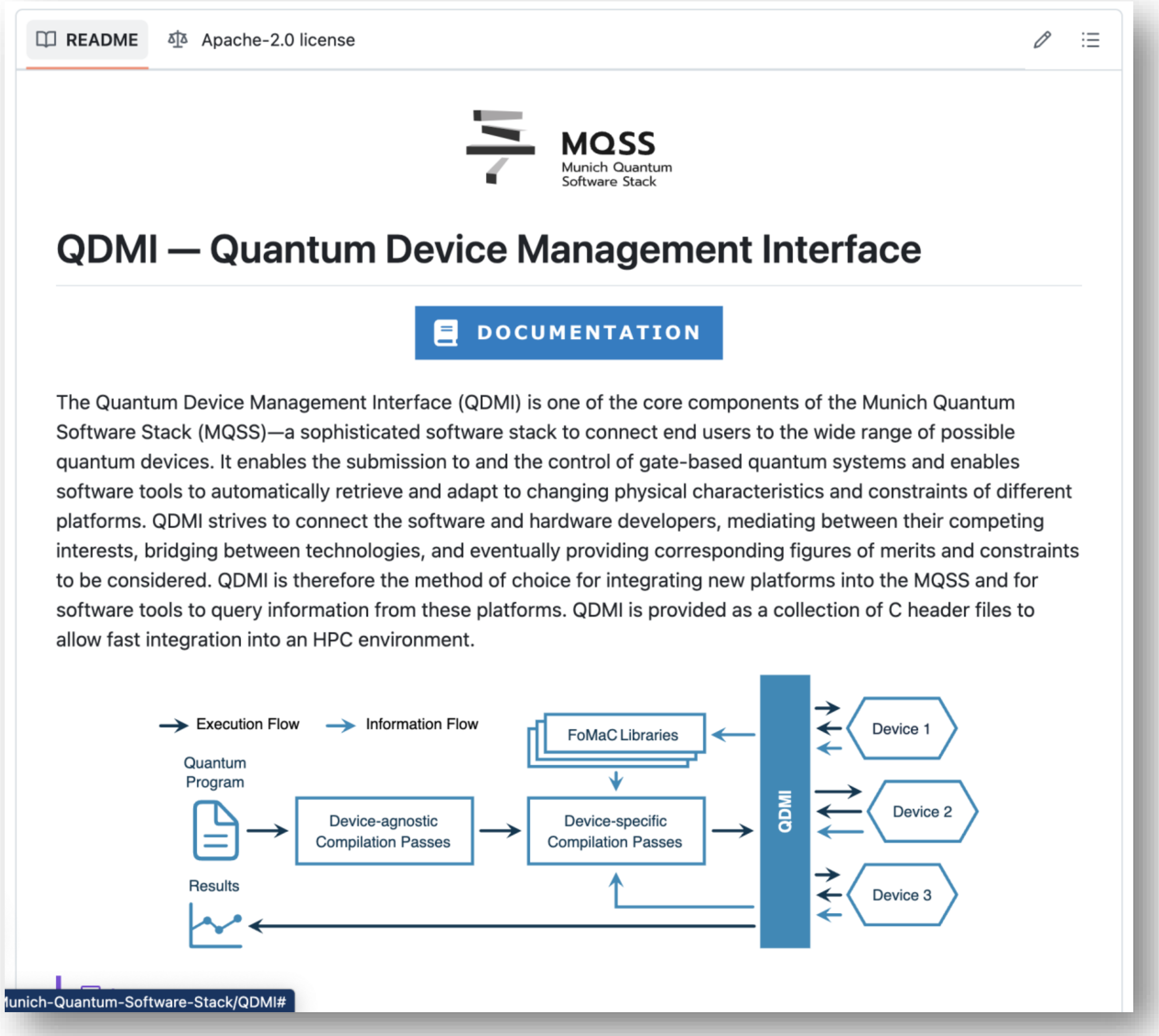
Flexibility in the APIs

- Key/Value-like abstraction for QC properties
- Support multiple levels of abstraction

Completing CI/CD setup

- Intention is for easy testing
- External contributions via github welcome!

Contact: mqss@munich-quantum-valley.de

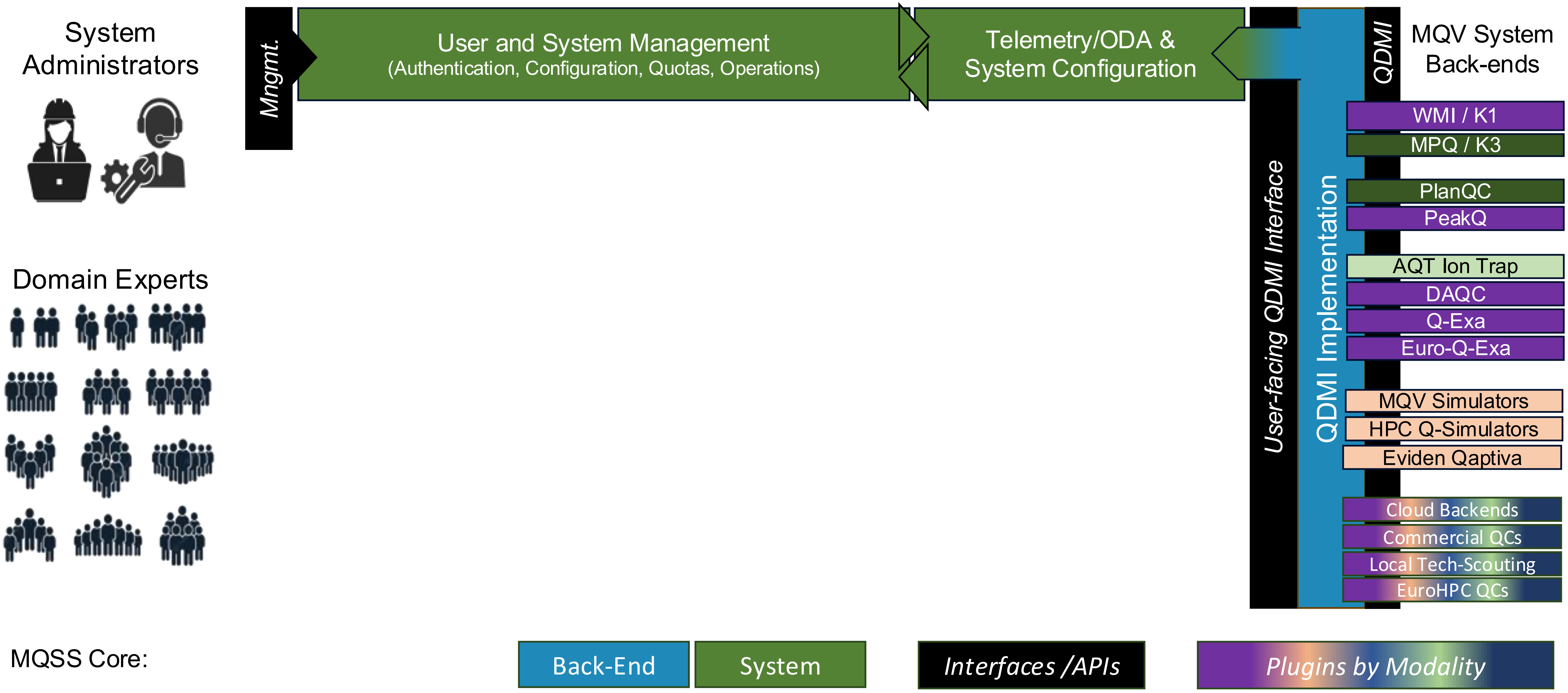

README
Apache-2.0 license
MQSS Munich Quantum Software Stack
DOCUMENTATION
Execution Flow
Information Flow
Quantum Program
Results
Device-agnostic Compilation Passes
FoMaC Libraries
Device-specific Compilation Passes
QDMI
Device 1
Device 2
Device 3
unich-Quantum-Software-Stack/QDMI#

QDMI — Quantum Device Management Interface

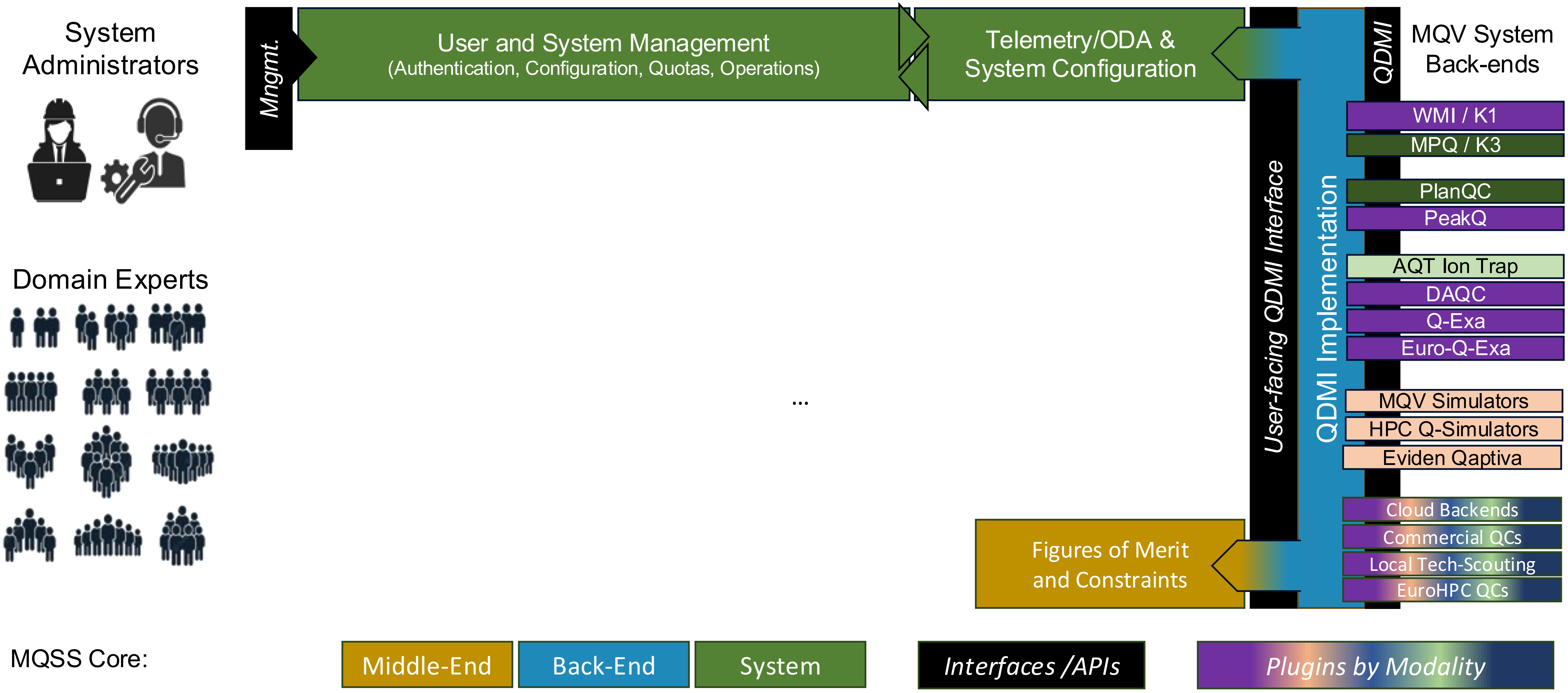
The Quantum Device Management Interface (QDMI) is one of the core components of the Munich Quantum Software Stack (MQSS)—a sophisticated software stack to connect end users to the wide range of possible quantum devices. It enables the submission to and the control of gate-based quantum systems and enables software tools to automatically retrieve and adapt to changing physical characteristics and constraints of different platforms. QDMI strives to connect the software and hardware developers, mediating between their competing interests, bridging between technologies, and eventually providing corresponding figures of merits and constraints to be considered. QDMI is therefore the method of choice for integrating new platforms into the MQSS and for software tools to query information from these platforms. QDMI is provided as a collection of C header files to allow fast integration into an HPC environment.

https://github.com/Munich-Quantum-Software-Stack/QDMI

MQSS Architecture



MQSS Architecture



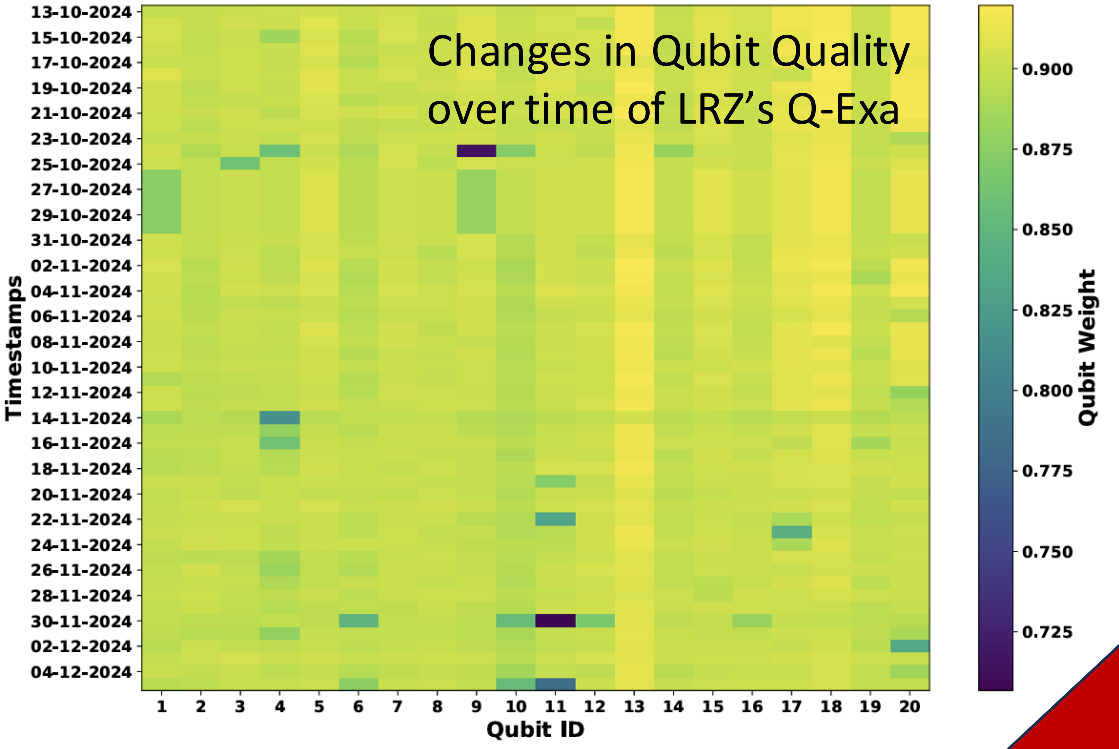
Extracting Topology Information with Sys-Sage

Topology data critical for both HPC and QC

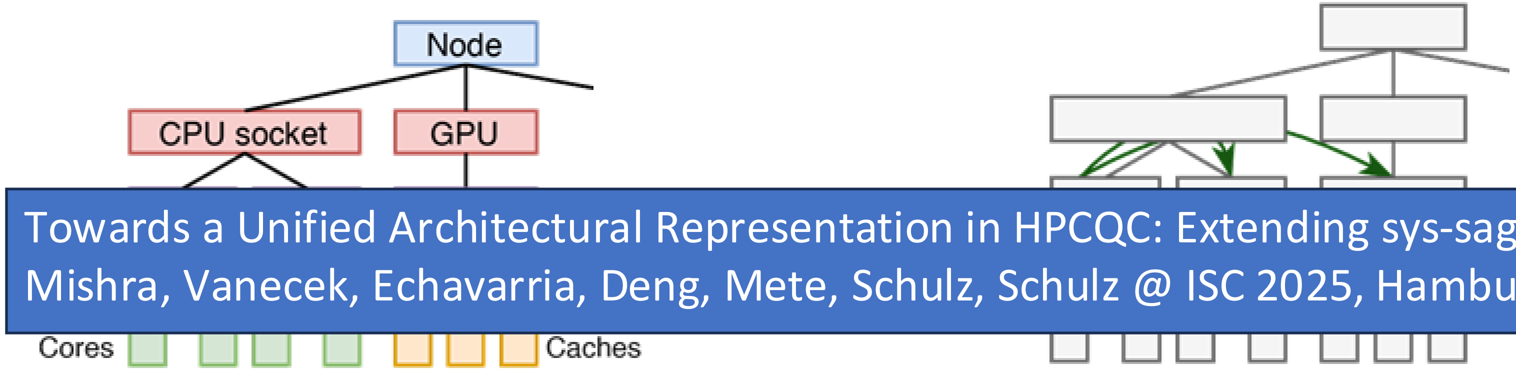
- Sys-sage tracks dynamic topology data for HPC systems and applications
- Crucial for optimizations and mapping of applications to systems

Why not extend to Quantum Topologies?

- Track qubit topologies (static property)
- Track qubit quality (dynamic property)
- Map both data into shared data structures



Changes in Qubit Quality over time of LRZ's Q-Exa

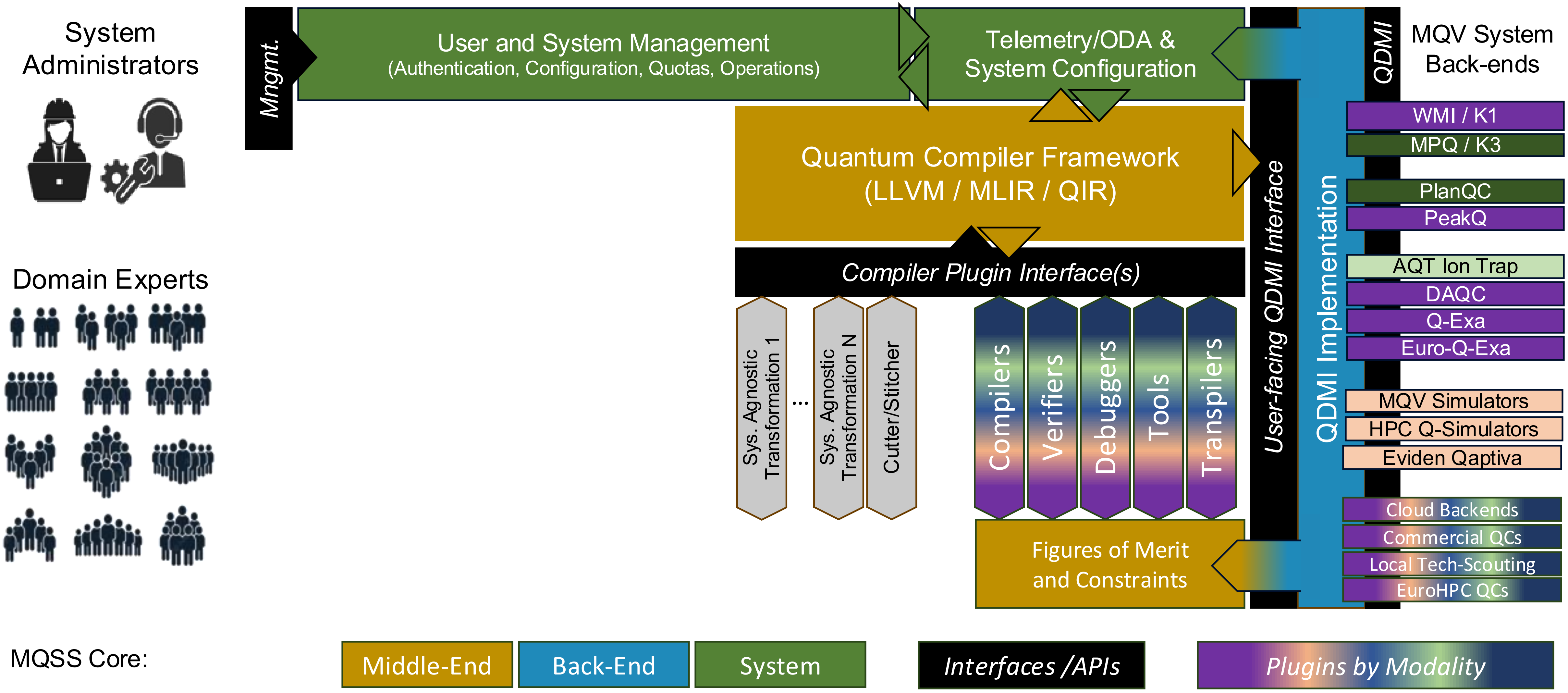


Towards a Unified Architectural Representation in HPCQC: Extending sys-sage for Quantum Technologies
Mishra, Vanecek, Echavarria, Deng, Mete, Schulz, Schulz @ ISC 2025, Hamburg

HANS MEUER AWARD WINNER 2025
best paper @ ISC25

MQSS Architecture

MQSS
Munich Quantum Software Stack



MQSS Components Catalog

Front-End

QPI: Hybrid Programming from C/C++
- LRZ/LS & TUM/MS: Ercüment Kaya

FPQA Compiler for Max3SAT problems
- TUM/PB: Oğuzcan Kırmemiş

qTPU: Large circuits as tensor networks
- TUM/PB: Nathaniel Tornow

ISV Job execution for Spin Hamiltonians
- LRZ/LS: Burak Mete and Tobias Bauer

MQT QECC: EC quantum circuit preparation
- TUM/RW: Lucas Berent

Parallel circuit extraction from ZX Diagrams
- LMU/DK: Karl Führlinger

GA4QCD: Application-specific synthesis
- LMU/CLP: Leo Sünkel

qcd-gym: Circuit builder/optimizer using RL
- LMU/CLP: Philipp Altmann

Middle-End

MQT Predictor: Predict suitable back-ends
- TUM/RW: Nils Quetschlich

MILQ: Assigning circuits backends
- TUM/CM: Philipp Seitz and Manuel Geiger

AI-based compiler path selection
- LRZ/LS & TUM/MS: Aleksandra Świerkowska

MQT QMAP: Topology mapping of circuits
- TUM/RW: Lukas Burgholzer

MQT QCEC: Tool for equivalence checking
- TUM/RW: Lukas Burgholzer

MQT Qudits: Compilation for multistate Qbits
- TUM/RW: Kevin Mato

Quantum constant propagation
- TUM/HS: Yanbin Chen

Mid-Circuit measurement reduction
- TUM/HS: Innocenzo Fulginiti

Back-End

Hardware backend development with partners
- LRZ/LS: Jorge Echavarria

FoMaCs via Sys-Sage tool library
- TUM/MS: Stepan Vanecek

Unified Quantum Platform (UQP)
- TUM/MS: Amr Elsharkawy

Quantum Control Processor (QCP) and ISA
- TUM/MS: Xiaorang Guo

Simulator: MQT DDSIM
- TUM/RW: Lukas Burgholzer

Simulator: Tensor networks
- TUM/CM: M. Geiher and Q. Huang

Simulator: Parallel Clifford+T
- LMU/DK: Florian Kroetz

Simulator: Back-ends for HPC simulators
- LRZ/LS: Marco De Pascale

System

Munich Quantum Portal (MQP) and plugins
- LRZ/LS: Marco De Pascale

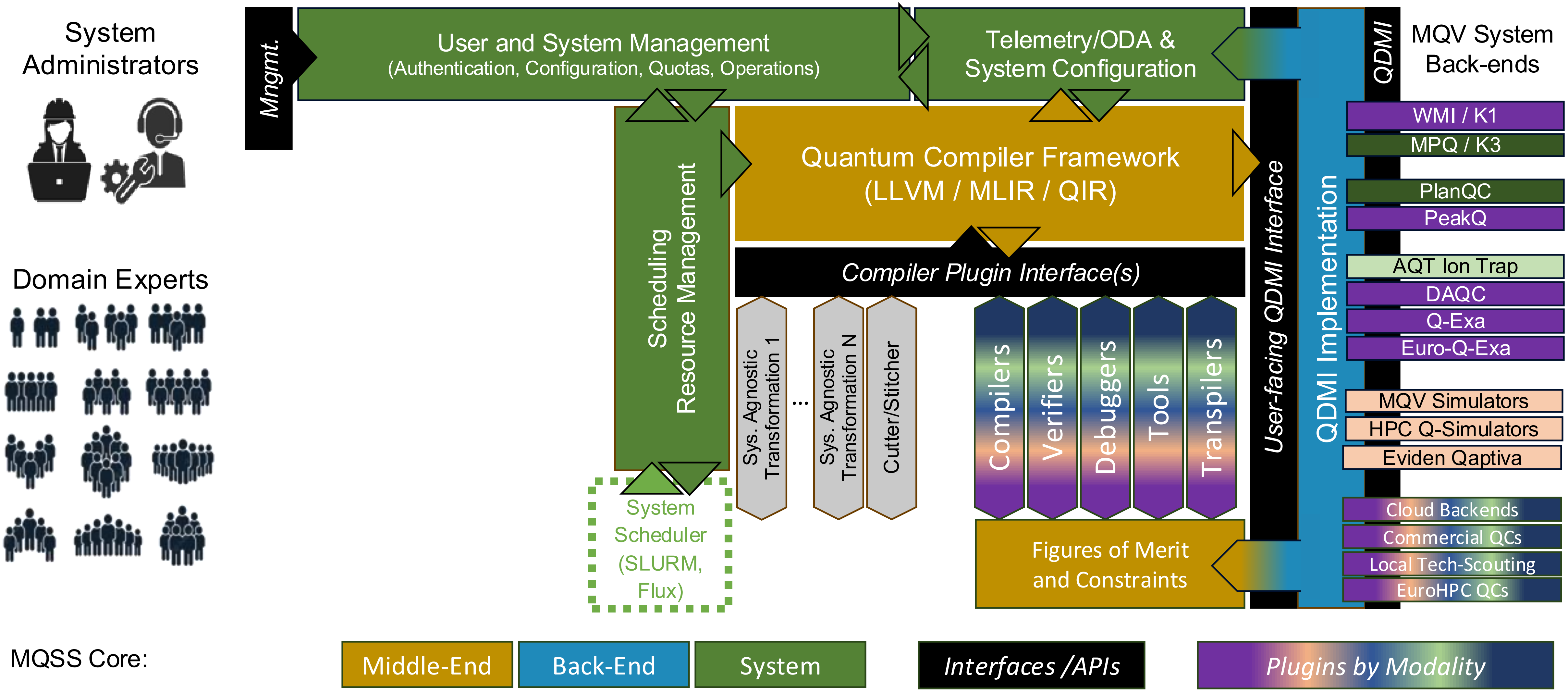
Resource prediction and circuit scheduler
- LRZ/LS: Minh Chung

IoT Environment / ODA / Digital Twins
- LRZ/LS & TUM/MS: H. Ahmed and Y. Gambo

HPC scheduling
- LRZ/LS & TUM/MS: Nufail Farooqi

Operations, Configuration, Calibration
- LRZ/LS: Matt Tovey and Xiaolang Deng

MQSS Architecture



System Administrators

Domain Experts

Mngmt.

User and System Management
(Authentication, Configuration, Quotas, Operations)

Telemetry/ODA & System Configuration

Scheduling Resource Management

Quantum Compiler Framework (LLVM / MLIR / QIR)

Compiler Plugin Interface(s)

Sys. Agnostic Transformation 1

…

Sys. Agnostic Transformation N

Cutter/Stitcher

Compilers

Verifiers

Debuggers

Tools

Transpilers

System Scheduler (SLURM, Flux)

Figures of Merit and Constraints

User-facing QDMI Interface

QDMI Implementation

QDMI

MQV System Back-ends

WMI / K1

MPQ / K3

PlanQC

PeakQ

AQT Ion Trap

DAQC

Q-Exa

Euro-Q-Exa

MQV Simulators

HPC Q-Simulators

Eviden Qaptiva

Cloud Backends

Commercial QCs

Local Tech-Scouting

EuroHPC QCs

MQSS Core:

Middle-End

Back-End

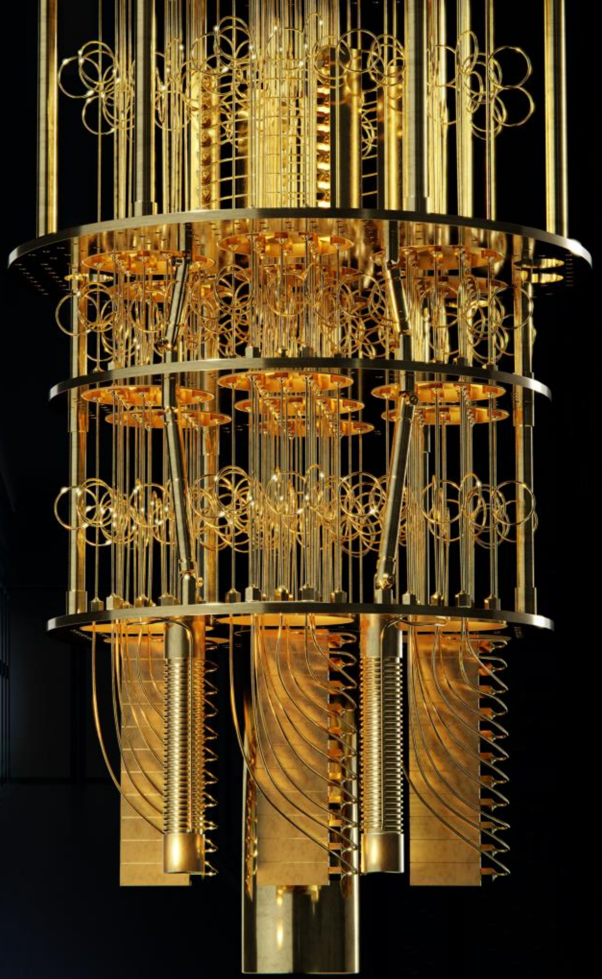
System

Interfaces /APIs

Plugins by Modality

Challenges

- Hardware Integration of the QC control system
- Dynamic scheduling of hybrid workflows
- Hybrid programming approaches/models

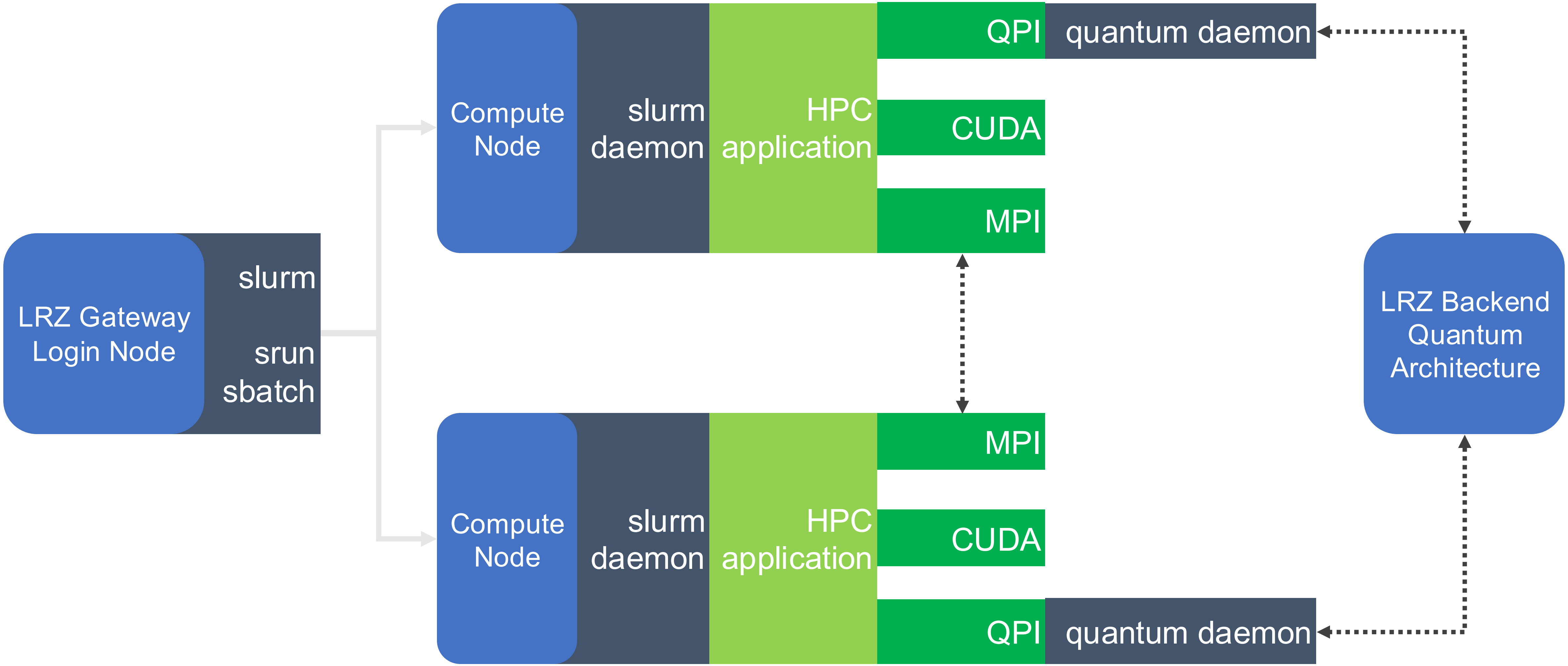


Why HPC-QC?

Quantum Computing
=
High-Performance Computing

New compute capability that adds to the supercomputing portfolio.

HPC Accessing QC via QPI



Quantum Programming Interface

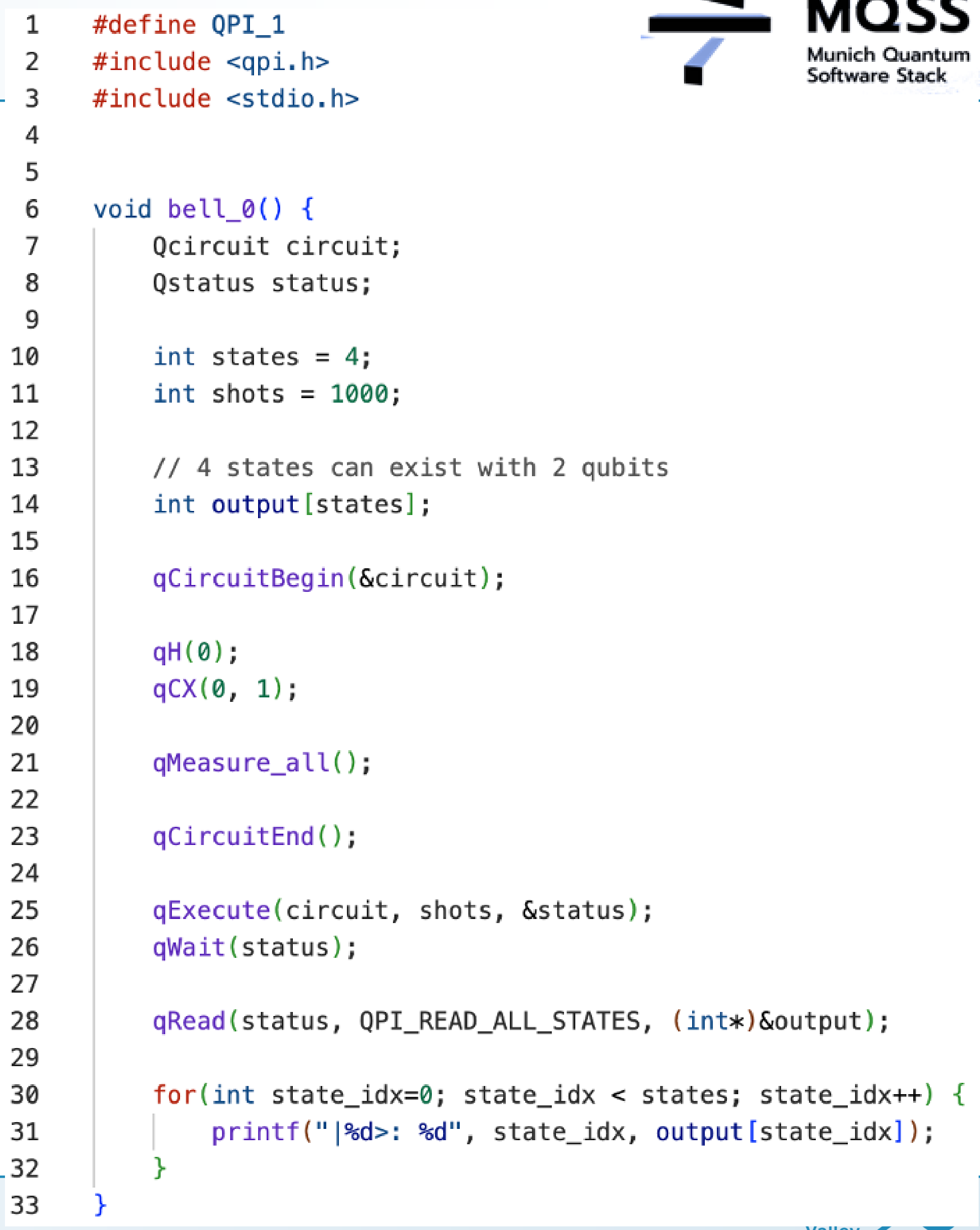
Aims to provide similar abstraction to Qiskit

- Abstracts architecture
- Vendor neutral

Users are legacy HPC applications

- C-based API
- Accelerator concept

Maps well to task-offload model and doesn’t force data structure



```c
#define QPI_1
#include <qpi.h>
#include <stdio.h>


void bell_0() {
    Qcircuit circuit;
    Qstatus status;

    int states = 4;
    int shots = 1000;

    // 4 states can exist with 2 qubits
    int output[states];

    qCircuitBegin(&circuit);

    qH(0);
    qCX(0, 1);

    qMeasure_all();

    qCircuitEnd();

    qExecute(circuit, shots, &status);
    qWait(status);

    qRead(status, QPI_READ_ALL_STATES, (int*)&output);

    for(int state_idx=0; state_idx < states; state_idx++) {
        printf("|%d>: %d", state_idx, output[state_idx]);
    }
}
```

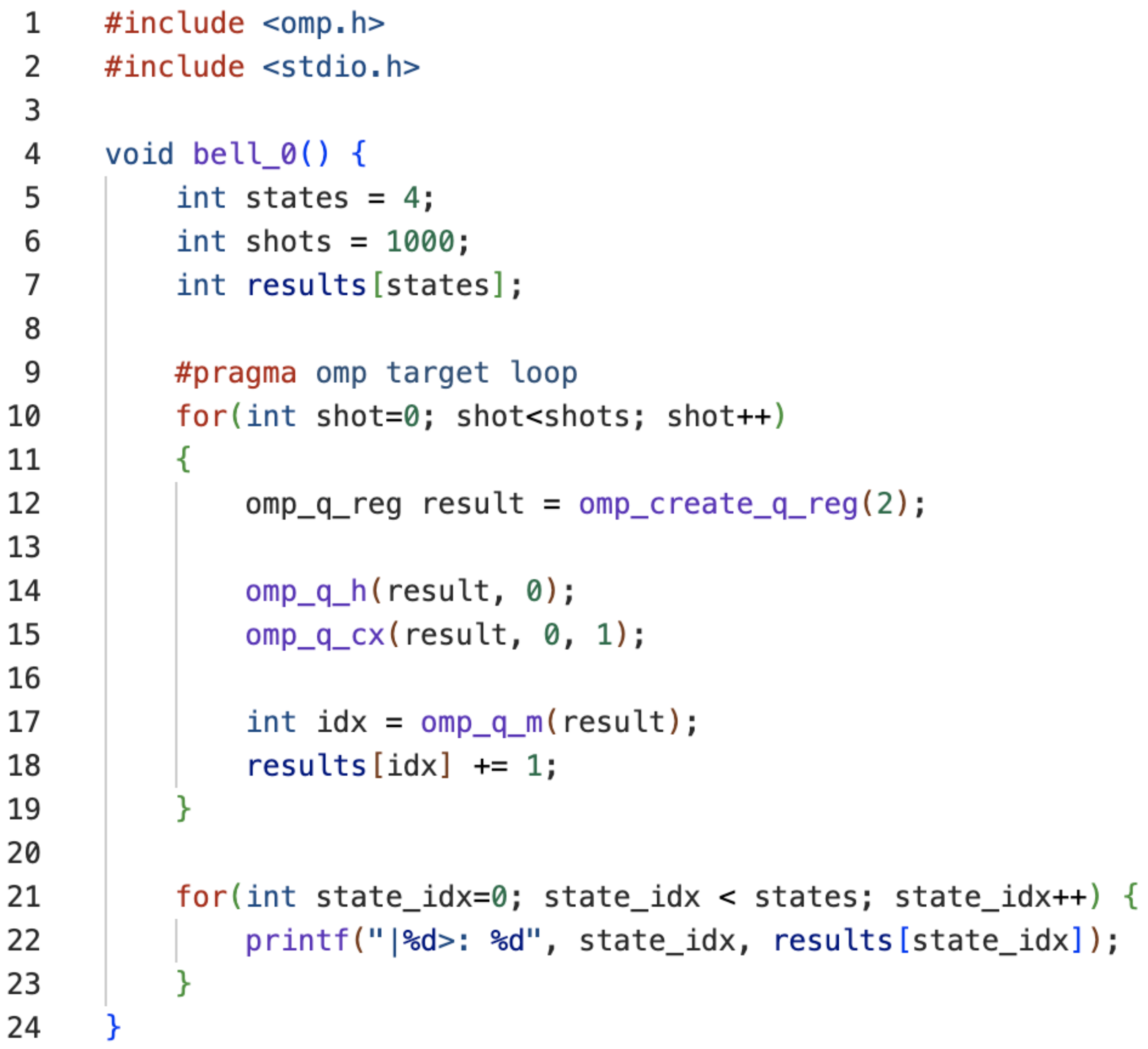
OpenMP Quantum Tasks Integration

Lower learning curve for HPC users

Benefits from compiler level information instead of a library level

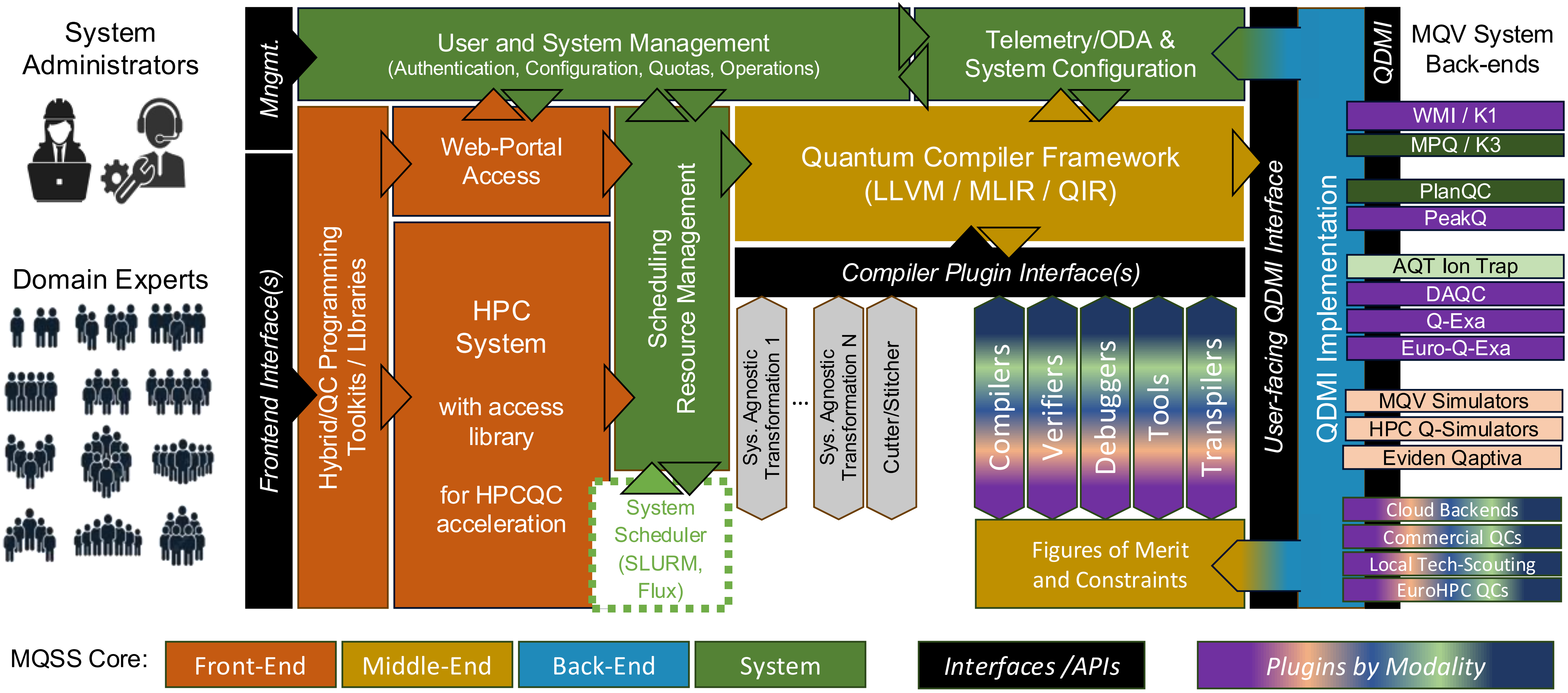
Possibility to includee offloading classical task to “nearby compute”

Quantum Task Offloading with the OpenMP API
Joseph KL Lee, Oliver T Brown, Mark Bull, Martin Ruefenacht, Johannes Doerfert, Michael Klemm, Martin Schulz
Posters at SC23



```
#include <omp.h>
#include <stdio.h>

void bell_0() {
    int states = 4;
    int shots = 1000;
    int results[states];

    #pragma omp target loop
    for(int shot=0; shot<shots; shot++)
    {
        omp_q_reg result = omp_create_q_reg(2);

        omp_q_h(result, 0);
        omp_q_cx(result, 0, 1);

        int idx = omp_q_m(result);
        results[idx] += 1;
    }

    for(int state_idx=0; state_idx < states; state_idx++) {
        printf("|%d>: %d", state_idx, results[state_idx]);
    }
}
```

MQSS Architecture



Looking Forward: Where Next for MQSS …

Stabilization of MQSS

- Continued Growth
- CI & CD support
- User Interfaces
- Transfer to Operation

“Evolution” of MQSS

- New backends/systems
- Better optimizations
- Easy-to-use abstractions
- User support components

“Revolution” of MQSS

- Fault tolerance
- Error mitigation
- Basic research work
- Crosscut for entire MQSS

The MQSS Core Team

Martin Schulz
martin.w.j.schulz@tum.de

Lukas Burgholzer
lukas.burgholzer@tum.de

Robert Wille
robert.wille@tum.de

Jorge Echavarria
Jorge.Echavarria@lrz.de

Alumni: Laura Schulz
Now collaborator at
Argonne NL

Contact:
mqss@quantum-valley.de

Thank you to our Groups!

CAPS Team @ TUM

CDA Team @ TUM

QCT Team @ LRZ

Contacts:

Martin Schulz
schulzm@in.tum.de

Robert Wille
robert.wille@tum.de

Contact the MQSS Team:
mqss@munich-quantum-valley.de

Github for MQSS:
https://github.com/Munich-Quantum-Software-Stack

Looking Forward: Where Next for MQSS …

Stabilization of MQSS

- Continued Growth
- CI & CD support
- User Interfaces
- Transfer to Operation

“Evolution” of MQSS

- New backends/systems
- Better optimizations
- Easy-to-use abstractions
- User support components

“Revolution” of MQSS

- Fault tolerance
- Error mitigation
- Basic research work
- Crosscut for all Q-DESSI

QC@LRZ HPCQC MQT

Enabling Quantum Acceleration in Hybrid HPCQC Workflows for and beyond NISQ