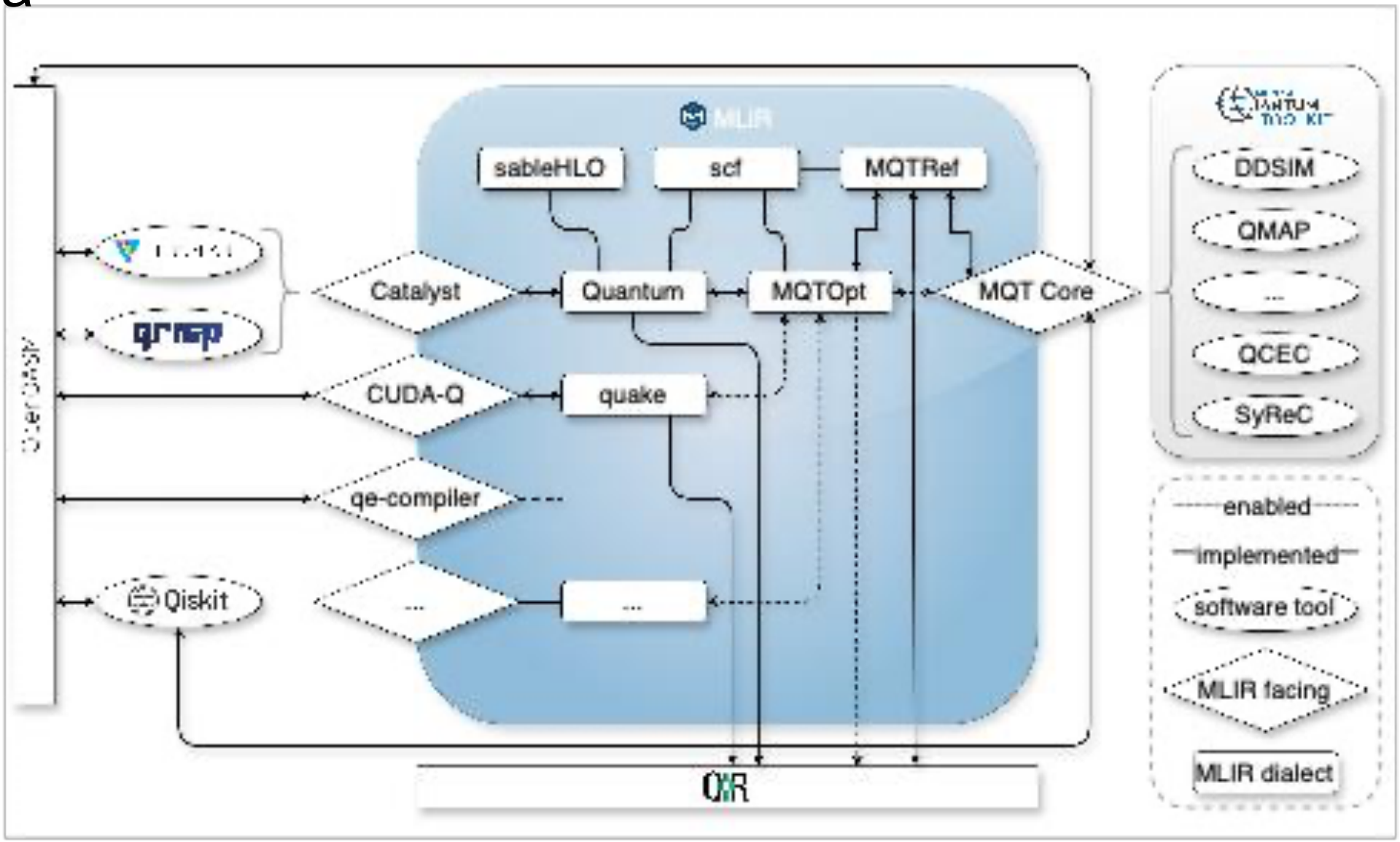
Connecting Quantum Software Tools with(in) MLIR

Patrick Hopf
Technical University of Munich
patrick.hopf@tum.de
www.cda.cit.tum.de

Agenda


MLIR
sableHLO
scf
MQTRef
Catalyst
Quantum
MQTOpt
MQT Core
CUDA-Q
quake
qe-compiler
Qiskit
...
QIR
DDSIM
QMAP
...
QCEC
SyReC
enabled
implemented
software tool
MLIR facing
MLIR dialect

The Munich Quantum Toolkit

Application

- Workflow from classical problem to quantum solution
- Automated encoding, execution & decoding

Compilation

- Automatic device selection
- Compiler optimization
- Technology-specific compilation
- Reversible synthesis

Error Correction

- Decoding algorithms
- Fault-tolerant state preparation
- Automated code construction and numerical simulations

Application → Simulation → Compilation → Verification → Error Correction → Hardware

Simulation

- Classical simulation of quantum circuits based on decision diagrams
- Includes sampling, noise-aware simulation, Hybrid Schrödinger Feynman approaches, approximation strategies, expectation value computations, etc.

Verification

- Equivalence checking
- Verifying compilation results

Hardware

- Application specific physical design for superconducting platform

Data Structures & Core Methods

- Efficient data structures
- Dedicated core methods (optimal and heuristic)
- Based on C++ and Python

Decision Diagrams · Tensor Networks · ZX-Calculus · SAT/SMT Solvers · Machine Learning · Heuristics

Check it out!

https://mqt.readthedocs.io

The Munich Quantum Toolkit

All tools are available as open-source repositories on GitHub

MQT ProblemSolver — Application
A Tool for Solving Problems Using Quantum Computing
github.com/cda-tum/mqt-problemsolver

MQT Bench — Application
A Quantum Circuit Benchmark Suite
www.cda.cit.tum.de/mqtbench
github.com/cda-tum/mqt-bench

MQT Quantum Auto Optimizer — Application
An Automatic Framework for Solving Optimization Problems
github.com/cda-tum/mqt-qao

MQT QUBOMaker — Application
A Framework for the Automatic Generation of QUBO Formulations
github.com/cda-tum/mqt-qubomaker

MQT DDSIM — Simulation
A Tool for Classical Quantum Circuit Simulation based on Decision Diagrams
github.com/cda-tum/mqt-ddsim

MQT Predictor — Compilation
A Tool for Determining Good Quantum Circuit Compilation Options
github.com/cda-tum/mqt-predictor

MQT IonShuttler — Compilation
A Tool for Generating Shuttling Schedules for QCCD Architectures
github.com/cda-tum/ion-shuttler

MQT Qudits — Compilation
A Tool for Compiling High-Dimensional Quantum Systems
github.com/cda-tum/mqt-qudits

MQT SyReC — Compilation
A Tool for the Synthesis of Reversible Circuits/Quantum Computing Oracles
github.com/cda-tum/mqt-syrec

MQT QMAP — Compilation
A Tool for Quantum Circuit Mapping And Clifford Circuit Optimization/Synthesis
github.com/cda-tum/mqt

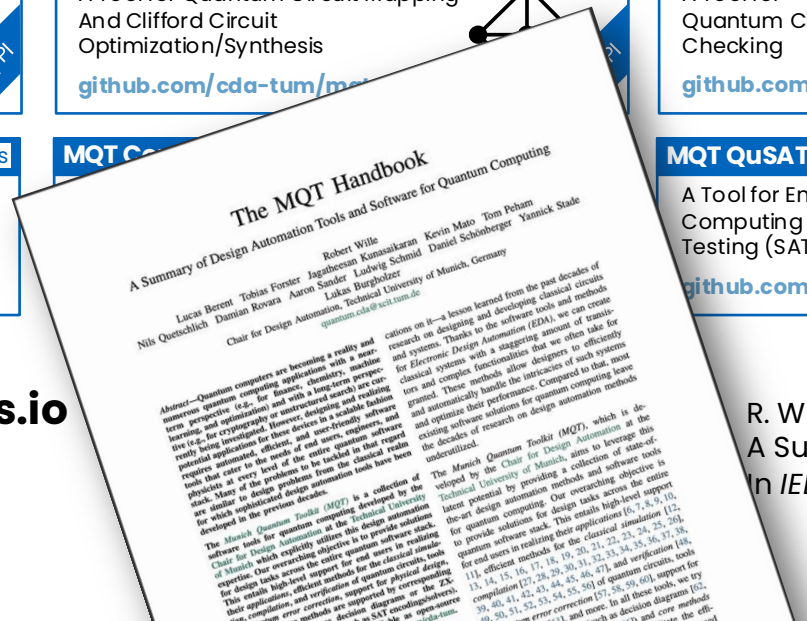
MQT QCEC — Verification
A Tool for Quantum Circuit Equivalence Checking
github.com/cda-tum/mqt-qcec

MQT DASQA — Hardware
A Tool for Designing Alternative Superconducting Quantum Architectures
github.com/cda-tum/mqt-dasqa

MQT DDVis — Data Structures
A Web-Application visualizing Decision Diagrams for Quantum Computing
www.cda.cit.tum.de/app/ddvis

MQT QuSAT — Core Methods
A Tool for Encoding Quantum Computing using Satisfiability Testing (SAT) Techniques
$F \wedge (x_1 \wedge \neg x_2)$
$F \wedge (x_3 \wedge x$
github.com/cda-tum/mqt-qusat

MQT QECC — QECC
A Tool for Quantum Error Correcting Codes
github.com/cda-tum/mqt-qecc

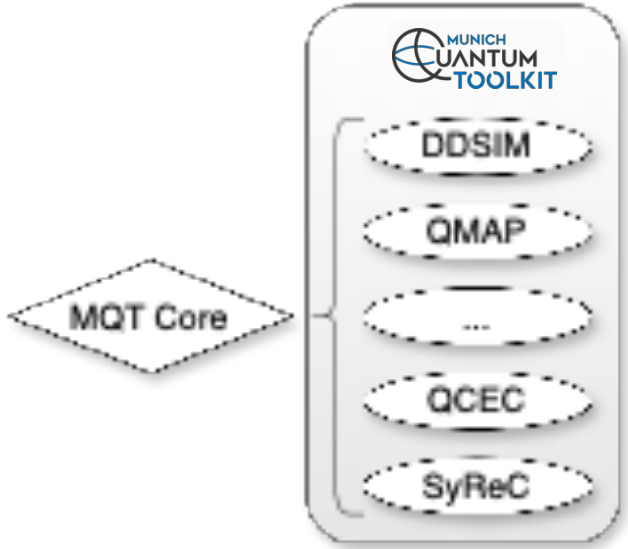

The MQT Handbook
A Summary of Design Automation Tools and Software for Quantum Computing

https://mqt.readthedocs.io

R. Wille et al. The MQT Handbook:
A Summary of Design Automation Tools and Software for Quantum Computing.
In *IEEE International Conference on Quantum Software (QSW)*. 2024

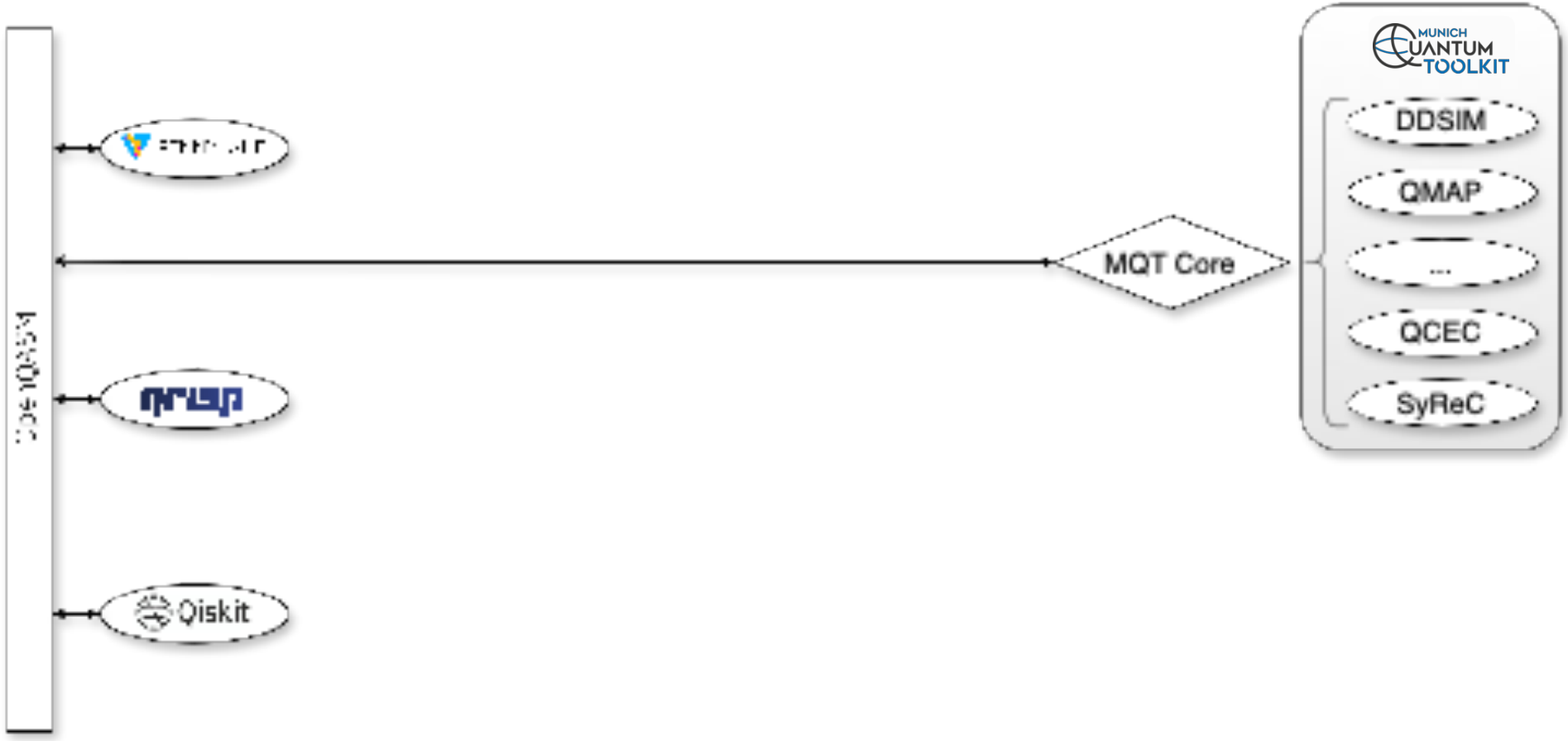
Intermediate Representation

```cpp
#include <mqt-core/ir/QuantumComputation.hpp>
#include <memory>
#include <vector>

int main() {
    qc::QuantumComputation circ(2, 1); // 2 qubits, 1 classical bit

    qcircuit.h(0); // Apply Hadamard to qubit 0

    // Create CNOT operation: control q0, target q1
    auto cnot = std::make_unique<qc::StandardOperation>(
        std::vector<qc::Qubit>{0, 1}, qc::OpType::X
    );
    circ.emplace_back(std::move(cnot));
}
```



Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

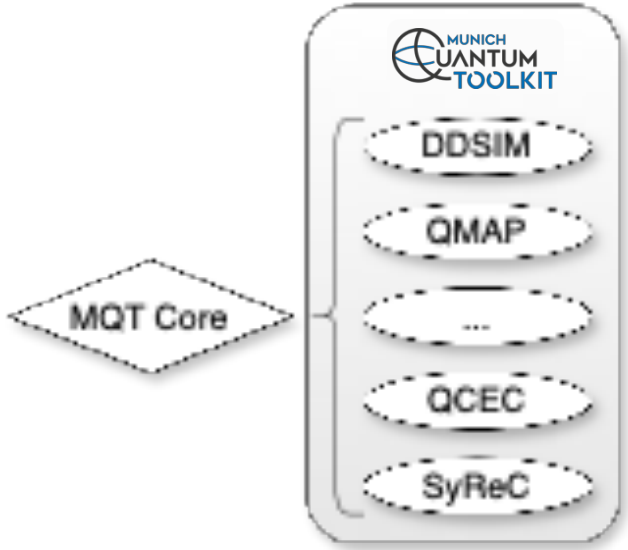
Intermediate Representation



Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representation – QASM (quantum-first)

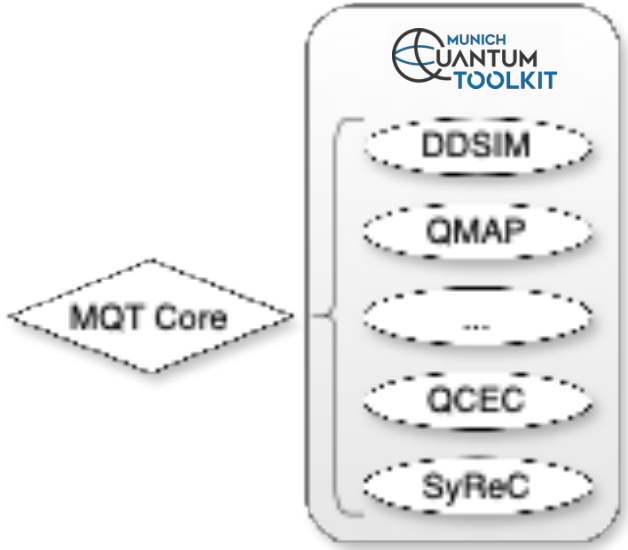
```cpp
#include <mqt-core/ir/QuantumComputation.hpp>
#include <mqt-core/qasm2/Importer.hpp>
#include <string>

std::string qasm_str = R"(OPENQASM 2.0;
include "stdgates.inc";
      qubit[2] q;
      h q[0];
      cx q[0], q[1];
)";



qc::QuantumComputation circ = qasm3::Importer::imports(qasm_str);
```



Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representation – QASM (quantum-first)

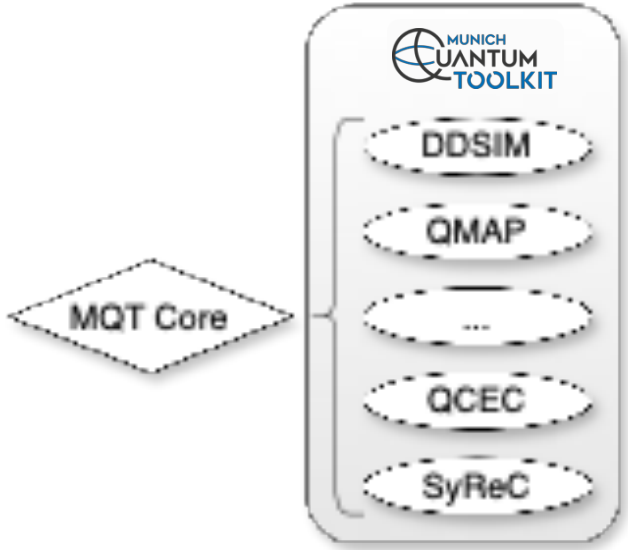
```
#include <mqt-core/ir/QuantumComputation.hpp>
#include <mqt-core/qasm3/Importer.hpp>
#include <string>

std::string qasm_str = R"(OPENQASM 3.0;
include "stdgates.inc";
      qubit[2] q;
      bit c[1];
      h q[0];
      c[0] = measure q[0];
      if (c[0]) {
          cx q[0], q[1];
      }
)";
qc::QuantumComputation circ = qasm3::Importer::imports(qasm_str);
```



Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

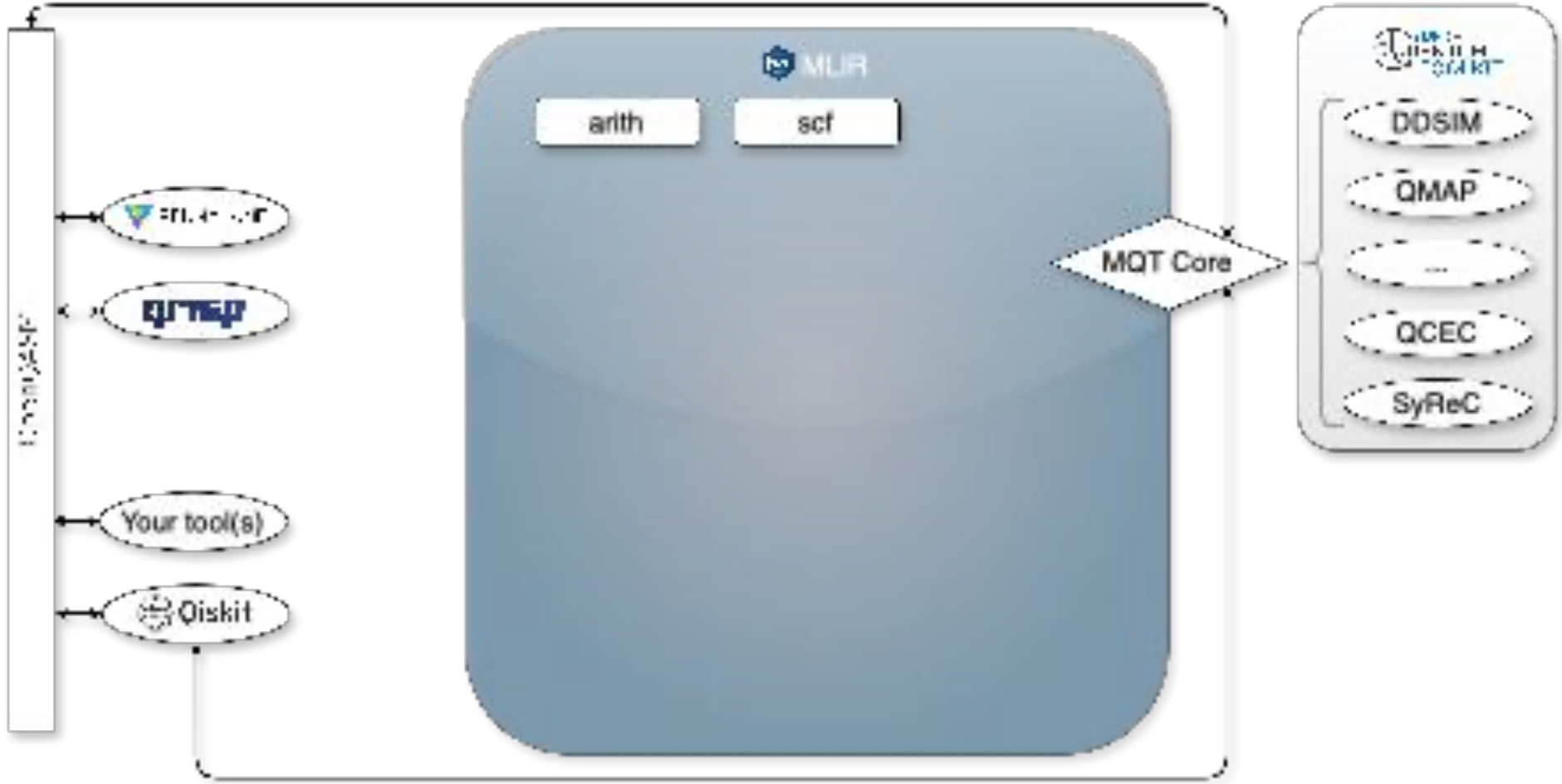
Intermediate Representation

```cpp
#include <mqt-core/ir/QuantumComputation.hpp>
#include <memory>
#include <vector>

int main() {
qc::QuantumComputation qcircuit(2, 1);

circ.h(0);
circ.measure(0, 0);
auto cnot = std::make_unique<qc::StandardOperation>(
                  std::vector<qc::Qubit>{0, 1}, qc::OpType::X);

circ.ifElse(std::move(cnot), nullptr, /*then and no else*/
                  /*control bit=*/0, /*expected value=*/true);
}
```



Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

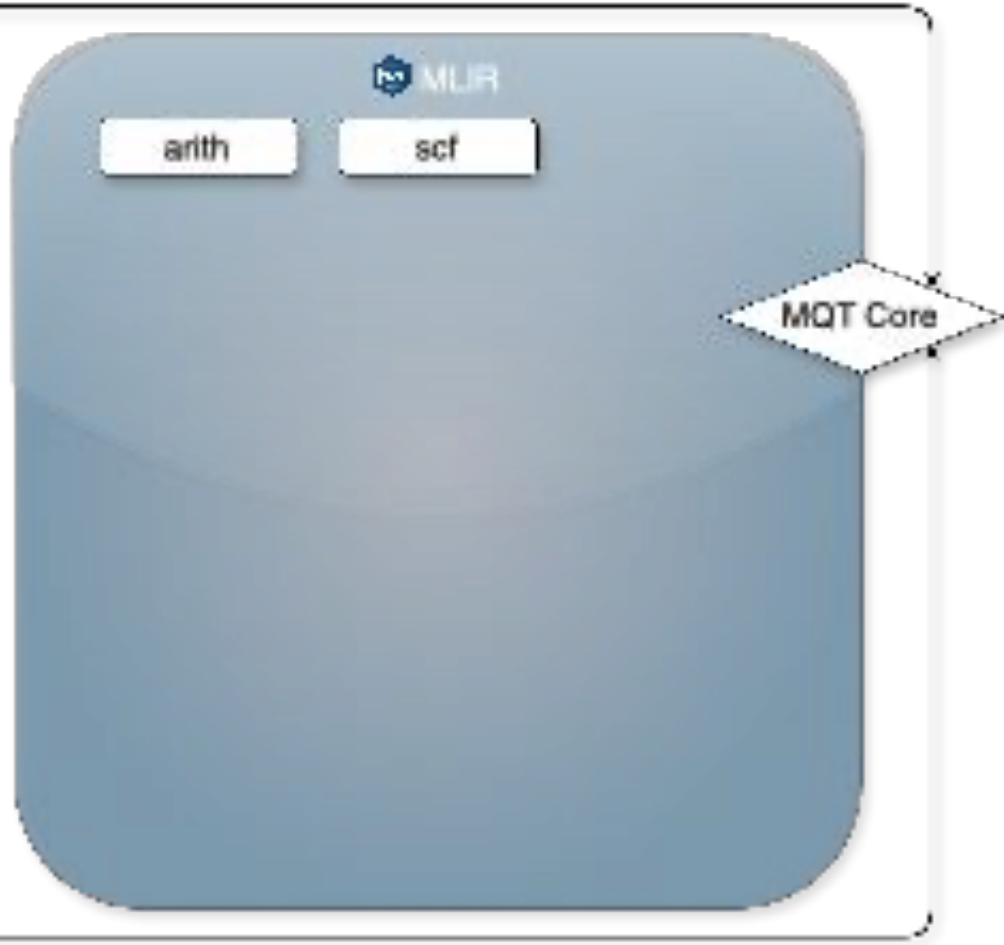
Intermediate Representations – MLIR (classical-first)



Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representations – MLIR

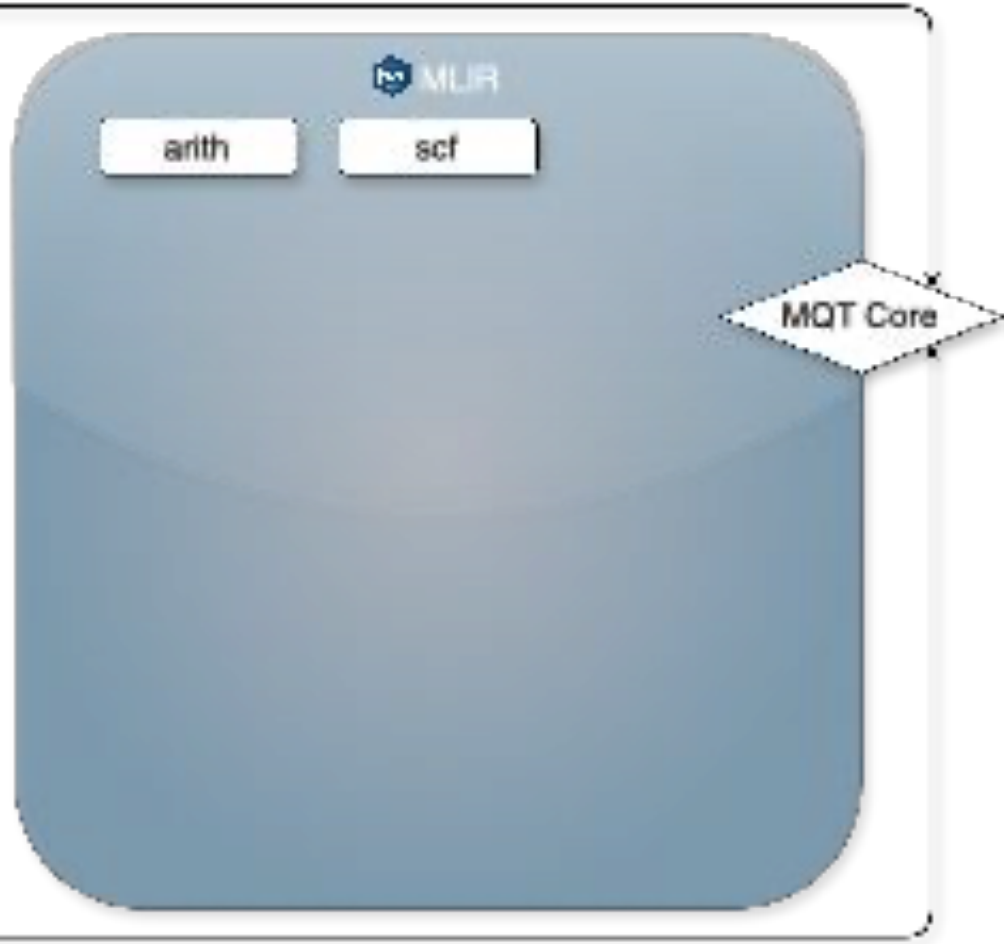
```
module {
  func.func @sum(%n: index) -> index {
    %c0 = arith.constant 0 : index
    %c1 = arith.constant 1 : index
    %res = scf.for %i = %c0 to %n step %c1
          iter_args(%acc = %c0) -> index {
                %acc2 = arith.addi %acc, %i : index
                scf.yield %acc2
          }
    return %res : index
  }
}
```



Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representations – MLIR

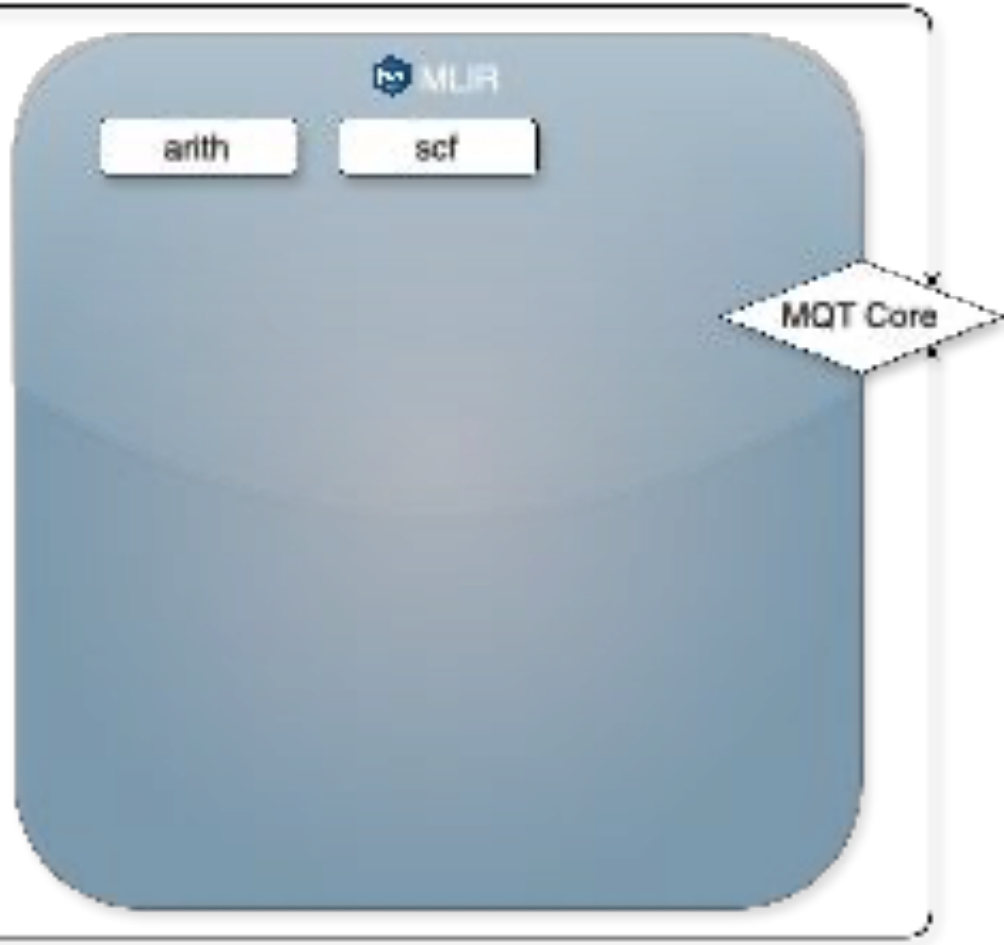
```
module {
  func.func @sum(%n: index) -> index {
    %c0 = arith.constant 0 : index
    %c1 = arith.constant 1 : index
    %res = scf.for %i = %c0 to %n step %c1
          iter_args(%acc = %c0) -> index {
                %acc2 = arith.addi %acc, %i : index
                scf.yield %acc2
          }
    return %res : index
  }
}
```



Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

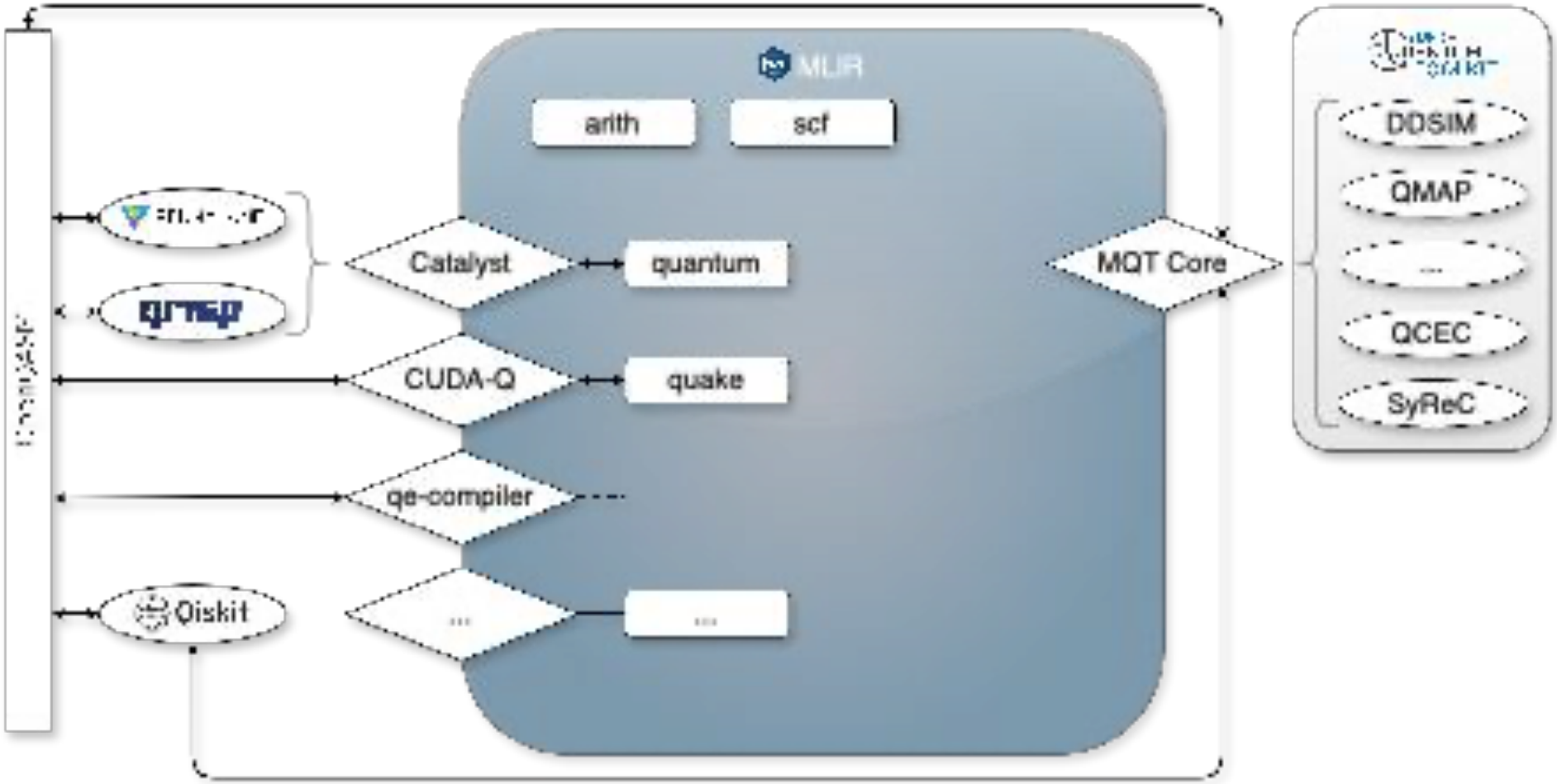
Intermediate Representations – MLIR

```
module {
  func.func @sum(%n: index) -> index {
    %c0 = arith.constant 0 : index
    %c1 = arith.constant 1 : index
    %res = scf.for %i = %c0 to %n step %c1
          iter_args(%acc = %c0) -> index {
                %acc2 = arith.addi %acc, %i : index
                scf.yield %acc2
          }
    return %res : index
  }
}
```



Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

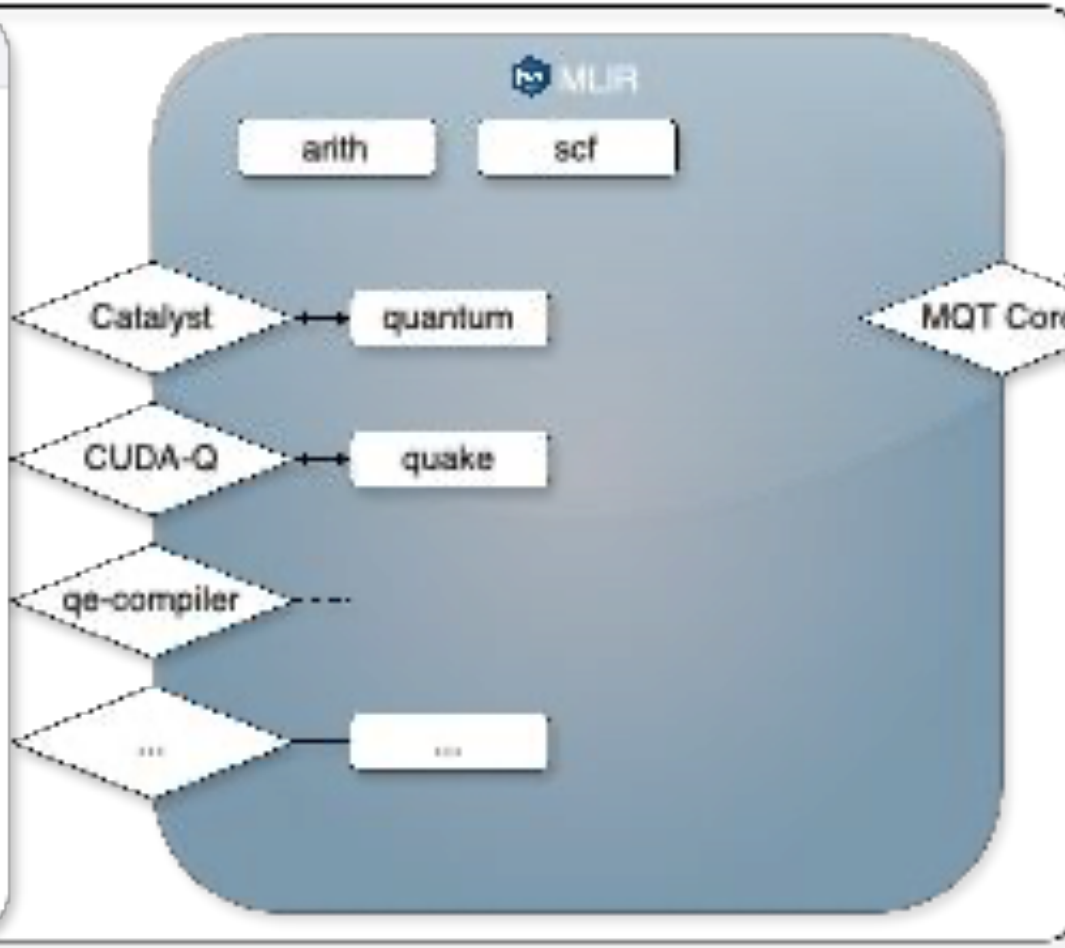
Intermediate Representations – MLIR



Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representations – MLIR

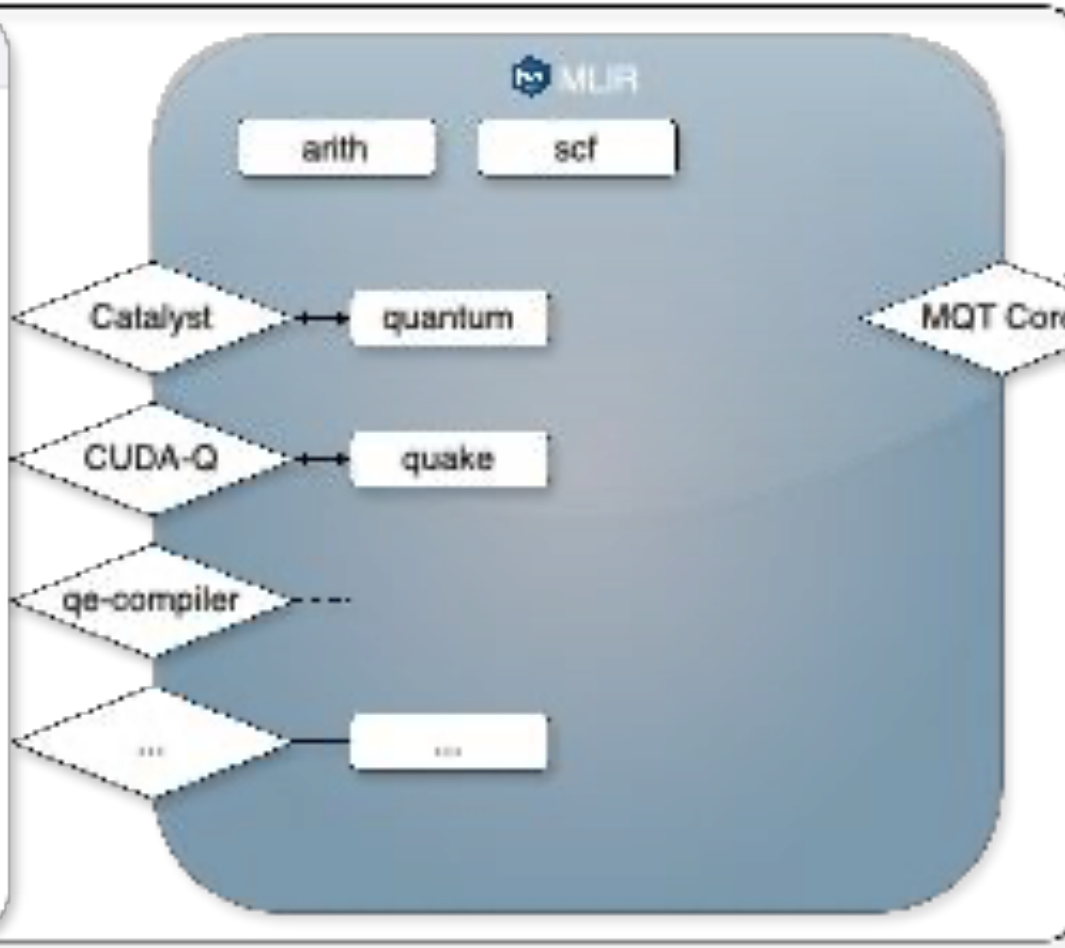
```
module  {
func.func public @circuit() attributes {…} {
      …
      %0 = quantum.alloc( 2) : !quantum.reg
      %1 = quantum.extract %0[ 0] : !quantum.reg -> !quantum.bit
      %2 = quantum.extract %0[ 1] : !quantum.reg -> !quantum.bit
      %out_qubits = quantum.custom "Hadamard"() %1 : !quantum.bit
      %out_qubits_0:2 = quantum.custom "CNOT"() %2, %out_qubits
                                    : !quantum.bit, !quantum.bit
      %3 = quantum.insert %0[ 0], %out_qubits_0#1 : !quantum.reg, …
      %4 = quantum.insert %3[ 1], %out_qubits_0#0 : !quantum.reg, …
      quantum.dealloc %4 : !quantum.reg
      return
   }
}
```



Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representations – MLIR

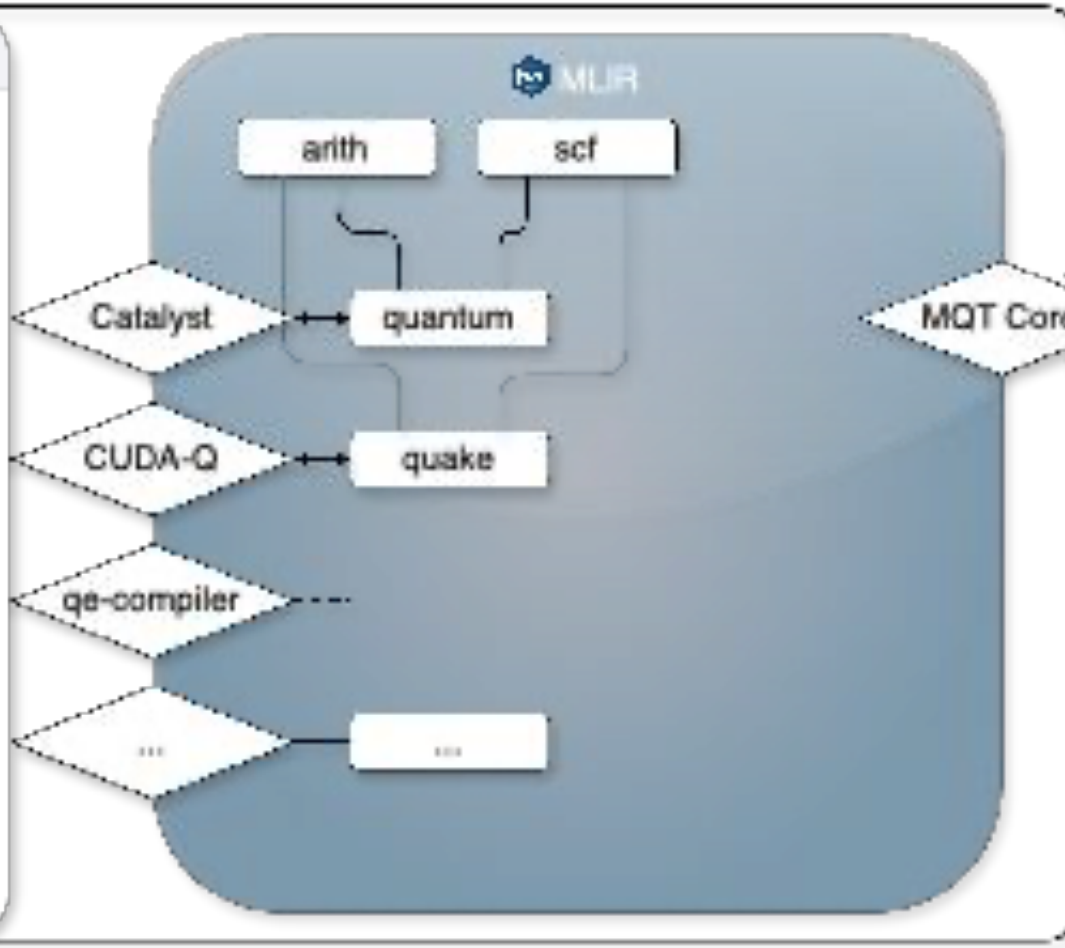
```
module  {
func.func public @circuit() attributes {…} {
      …
      %0 = quantum.alloc( 2) : !quantum.reg
      %1 = quantum.extract %0[ 0] : !quantum.reg -> !quantum.bit
      %2 = quantum.extract %0[ 1] : !quantum.reg -> !quantum.bit
      %out_qubits = quantum.custom "Hadamard"() %1 : !quantum.bit
      %out_qubits_0:2 = quantum.custom "CNOT"() %2, %out_qubits
                                    : !quantum.bit, !quantum.bit
      %3 = quantum.insert %0[ 0], %out_qubits_0#1 : !quantum.reg, …
      %4 = quantum.insert %3[ 1], %out_qubits_0#0 : !quantum.reg, …
      quantum.dealloc %4 : !quantum.reg
      return
   }
}
```



Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

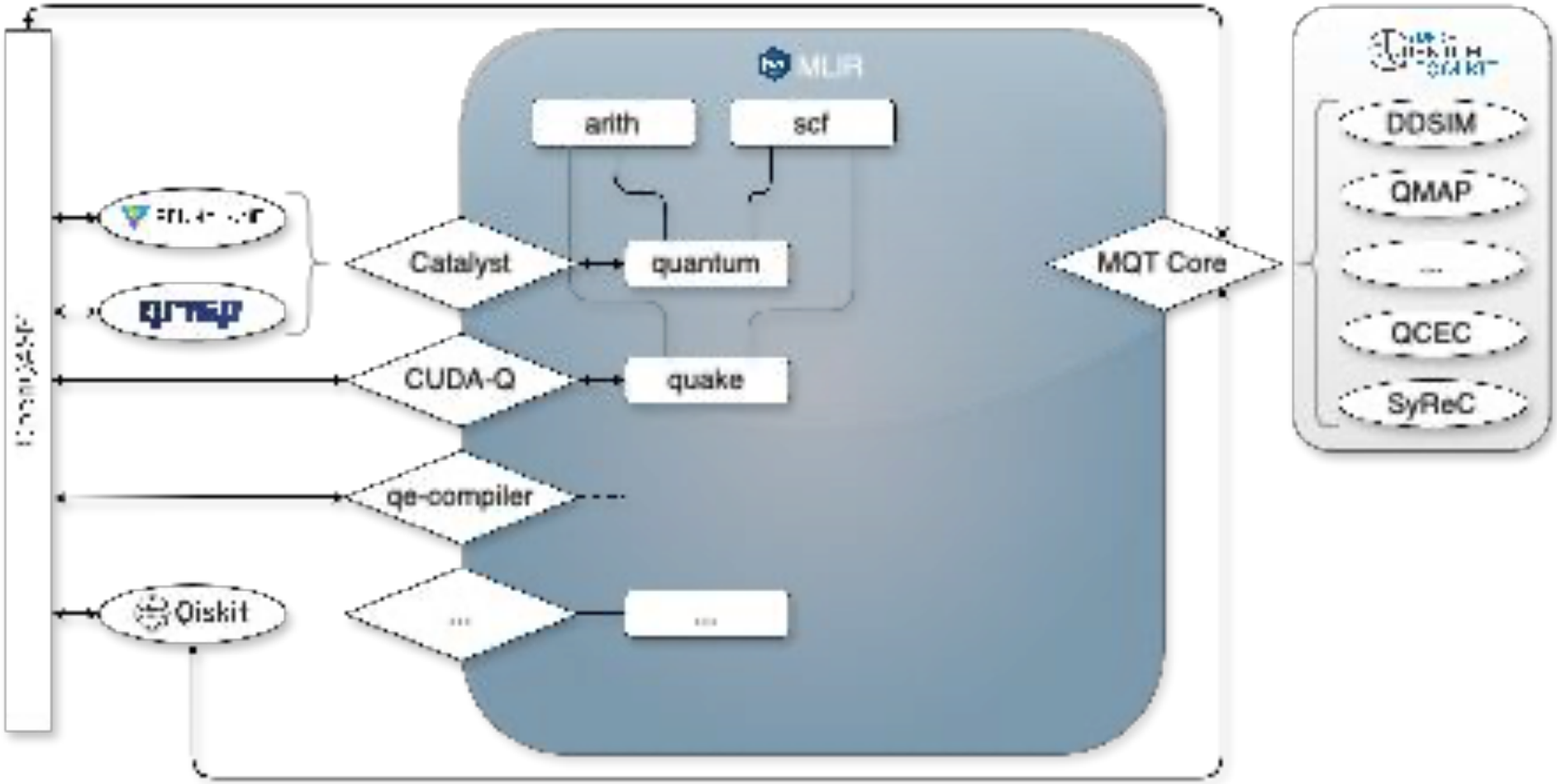
Intermediate Representations – MLIR

```
module  {
func.func public @circuit(%cond: i64) {
    %qreg = quantum.alloc(1) : !quantum.reg
    %q0 = quantum.extract %qreg[0] : !quantum.reg -> !quantum.bit

    scf.if %cond {
      %q0_h = quantum.custom "Hadamard"() %q0 : !quantum.bit
      %qreg1 = quantum.insert %qreg[0], %q0_h : !quantum.reg, …
    } else {
      %qreg1 = %qreg
    }

    quantum.dealloc %qreg1 : !quantum.reg
    return
  }
}
```



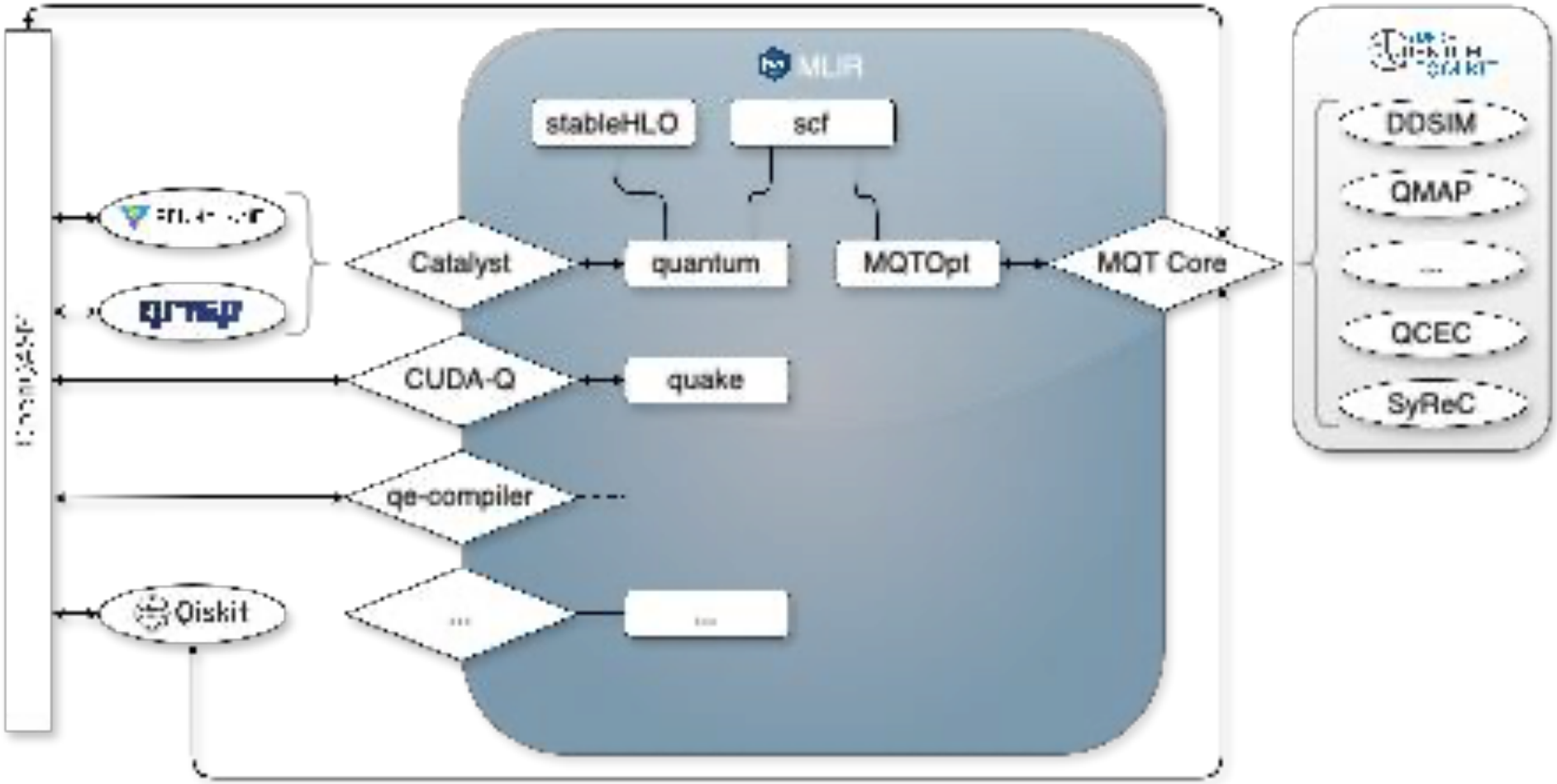
Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representations – MLIR



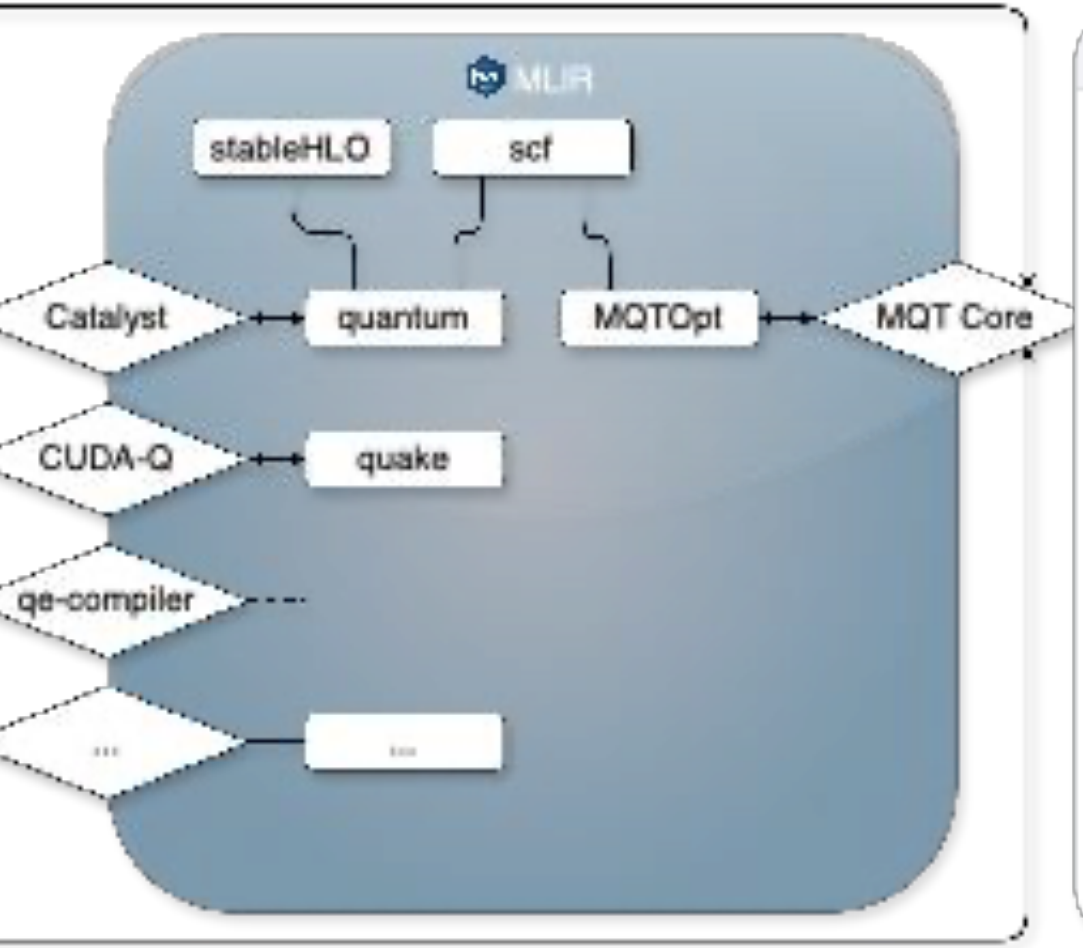
Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representations – MLIR



Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

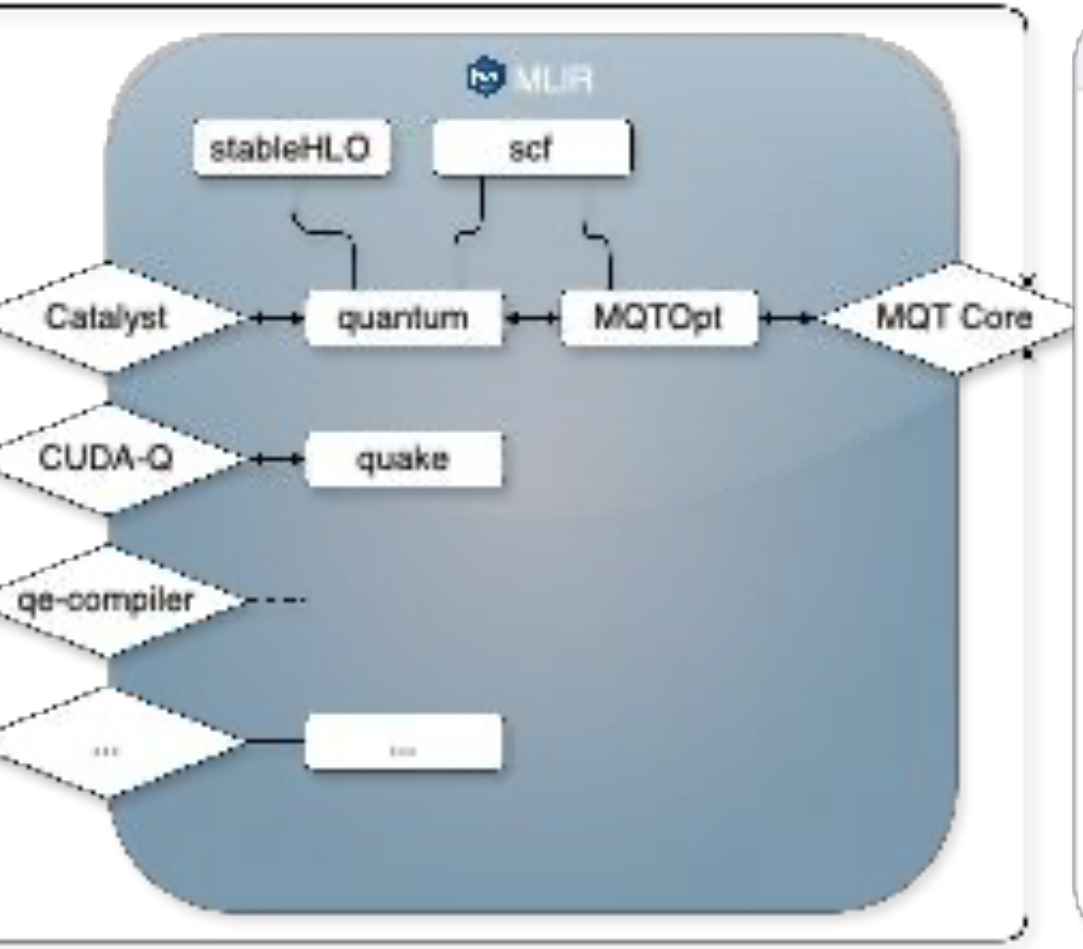
Intermediate Representations – MLIR




```
module {
 func.func public @circuit(%cond: i64) {
  %alloc = memref.alloc() : memref<2x!mqtopt.Qubit>
  %c0 = arith.constant 0 : index
  %c1 = arith.constant 1 : index
  %0 = memref.load %alloc[%c0] : memref<2x!mqtopt.Qubit>
  %1 = memref.load %alloc[%c1] : memref<2x!mqtopt.Qubit>
  %out_qubits = mqtopt.h(static [] mask []) %0 : !mqtopt.Qubit
  %out_qubits_0, %pos_ctrl_out_qubits = mqtopt.x(static [] mask [])
         %1 ctrl %out_qubits : !mqtopt.Qubit ctrl !mqtopt.Qubit
  %c0_1 = arith.constant 0 : index
  %c1_2 = arith.constant 1 : index
  memref.store %pos_ctrl_out_qubits, %alloc[%c0_1] : memref<2x!...
  memref.store %out_qubits_0, %alloc[%c1_2] : memref<2x!...
  memref.dealloc %alloc : memref<2x!mqtopt.Qubit>
  quantum.device_release
  return}
}
```

Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representations – MLIR Plugin




```
lib/
├── Conversion/
│   ├── CatalystQuantumToMQTOpt/
│   │   └── CatalystQuantumToMQTOpt.cpp
│   └── MQTOptToCatalystQuantum/
│       └── MQTOptToCatalystQuantum.cpp
└── mqt-plugin.cpp

include/mlir/
└── Conversion/
    ├── CatalystQuantumToMQTOpt/
    │   ├── CatalystQuantumToMQTOpt.h
    │   └── CatalystQuantumToMQTOpt.td
    └── MQTOptToCatalystQuantum/
        ├── MQTOptToCatalystQuantum.h
        └── MQTOptToCatalystQuantum.td
```

Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representations – MLIR Plugin

```
lib/
├── Conversion/
│   ├── CatalystQuantumToMQTOpt/
│   │   └── CatalystQuantumToMQTOpt.cpp
│   └── MQTOptToCatalystQuantum/
│       └── MQTOptToCatalystQuantum.cpp
└── mqt-plugin.cpp

include/mlir/
└── Conversion/
    ├── CatalystQuantumToMQTOpt/
    │   ├── CatalystQuantumToMQTOpt.h
    │   └── CatalystQuantumToMQTOpt.td
    └── MQTOptToCatalystQuantum/
        ├── MQTOptToCatalystQuantum.h
        └── MQTOptToCatalystQuantum.td
```

```
include "mlir/Pass/PassBase.td"

def MQTOptToCatalystQuantum : Pass<"mqtopt-to-catalystquantum"> {
  let summary = "Convert MQT's `MQTOpt` to Catalyst's `Quantum`.";

  let description = [{
    This pass converts MQT's `MQTOpt` to Catalyst's `Quantum`.
  }];
  let dependentDialects = [
    "::catalyst::quantum::QuantumDialect",
    "::mlir::arith::ArithDialect",
    "::mlir::func::FuncDialect",
    "::mlir::memref::MemRefDialect",
    "::mqt::ir::opt::MQTOptDialect"
  ];
}
```

Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representations – MLIR Plugin

```
lib/
├── Conversion/
│   ├── CatalystQuantumToMQTOpt/
│   │   └── CatalystQuantumToMQTOpt.cpp
│   └── MQTOptToCatalystQuantum/
│       └── MQTOptToCatalystQuantum.cpp
└── mqt-plugin.cpp

include/mlir/
└── Conversion/
    ├── CatalystQuantumToMQTOpt/
    │   ├── CatalystQuantumToMQTOpt.h
    │   └── CatalystQuantumToMQTOpt.td
    └── MQTOptToCatalystQuantum/
        ├── MQTOptToCatalystQuantum.h
        └── MQTOptToCatalystQuantum.td
```

```
struct ConvertMQTOptSimpleGate final : OpConversionPattern<MQTGateOp> {
  using OpConversionPattern<MQTGateOp>::OpConversionPattern;

  LogicalResult matchAndRewrite(MQTGateOp op,
                  typename MQTGateOp::Adaptor adaptor,
                  ConversionPatternRewriter& rewriter) const override {

auto newOp = rewriter.create<catalyst::quantum::CustomOp>(
        /*gate=*/ op.getLoc(),
        /*gate=*/ adaptor.gateName,
        /*in_qubits=*/ adaptor.inQubits,
        /*in_ctrl_qubits=*/ adaptor.ctrlInfo.ctrlQubits,
        /*in_ctrl_values=*/ adaptor.ctrlInfo.ctrlValues,
        /*params=*/adaptor.getParams(),
        /*adjoint=*/false);

    rewriter.replaceOp(op, newOp.getResults());
    return success(); ...}
```

Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representations – MLIR Plugin

```
lib/
├── Conversion/
│   ├── CatalystQuantumToMQTOpt/
│   │   └── CatalystQuantumToMQTOpt.cpp
│   └── MQTOptToCatalystQuantum/
│       └── MQTOptToCatalystQuantum.cpp
└── mqt-plugin.cpp

include/mlir/
└── Conversion/
    ├── CatalystQuantumToMQTOpt/
    │   ├── CatalystQuantumToMQTOpt.h
    │   └── CatalystQuantumToMQTOpt.td
    └── MQTOptToCatalystQuantum/
        ├── MQTOptToCatalystQuantum.h
        └── MQTOptToCatalystQuantum.td
```

```
class MQTOptToCatalystQuantumTypeConverter final : public TypeConverter {
public:
  explicit MQTOptToCatalystQuantumTypeConverter(MLIRContext* ctx) {

    addConversion([ctx](MemRefType memrefType) -> Type {
      if (auto qubitType =
              dyn_cast<opt::QubitType>(memrefType.getElementType())) {
        return catalyst::quantum::QuregType::get(ctx);
      }
      return memrefType;
    });

    addConversion([ctx](opt::QubitType /*type*/) -> Type {
      return catalyst::quantum::QubitType::get(ctx);
    });
  }
};
```

Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

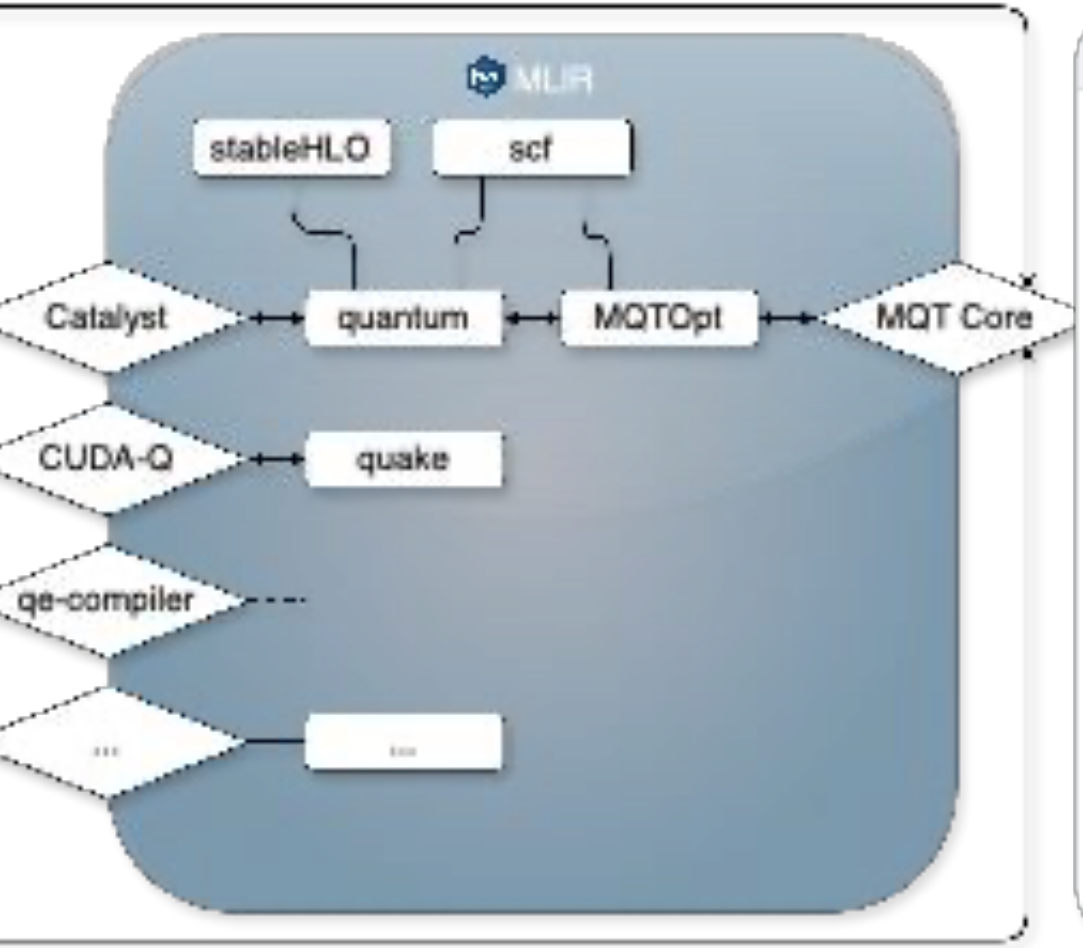
Intermediate Representations – MLIR Plugin

```
lib/
├── Conversion/
│   ├── CatalystQuantumToMQTOpt/
│   │   └── CatalystQuantumToMQTOpt.cpp
│   └── MQTOptToCatalystQuantum/
│       └── MQTOptToCatalystQuantum.cpp
└── mqt-plugin.cpp

include/mlir/
└── Conversion/
    ├── CatalystQuantumToMQTOpt/
    │   ├── CatalystQuantumToMQTOpt.h
    │   └── CatalystQuantumToMQTOpt.td
    └── MQTOptToCatalystQuantum/
        ├── MQTOptToCatalystQuantum.h
        └── MQTOptToCatalystQuantum.td
```

```cpp
extern "C" LLVM_ATTRIBUTE_WEAK DialectPluginLibraryInfo
mlirGetDialectPluginInfo() {
 return {.apiVersion = MLIR_PLUGIN_API_VERSION,
         .pluginName = "MQTOpt",
         .pluginVersion = LLVM_VERSION_STRING,
 .registerDialectRegistryCallbacks = [](DialectRegistry* registry) {
   registry->insert<::mqt::ir::opt::MQTOptDialect>();
}};}

extern "C" LLVM_ATTRIBUTE_WEAK PassPluginLibraryInfo
mlirGetPassPluginInfo() {
  return {.apiVersion = MLIR_PLUGIN_API_VERSION,
          .pluginName = "MQTOptPasses",
          .pluginVersion = LLVM_VERSION_STRING,
          .registerPassRegistryCallbacks = []() {
    mqt::ir::conversions::registerCatalystQuantumToMQTOptPasses();
    mqt::ir::conversions::registerMQTOptToCatalystQuantumPasses();
}};}
```

Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

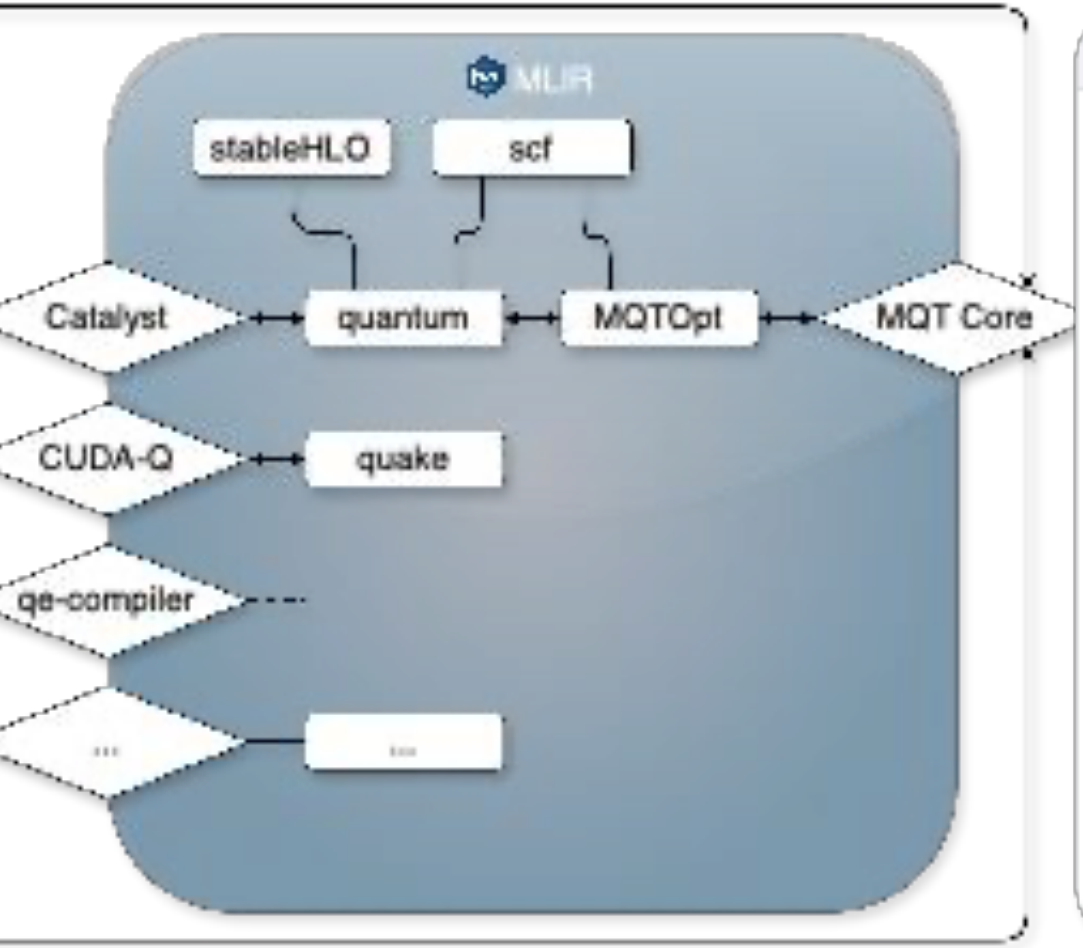
Intermediate Representations – MLIR Plugin




```
mlir-opt \
 --load-pass-plugin=/build/lib/mqt-core-plugins-catalyst.dylib \
 --load-dialect-plugin=/build/lib/mqt-core-plugins-catalyst.dylib \
 --mqtopt-to-catalyst-quantum \
 quantum_program.mlir \
 -o mqtopt_program.mlir

mlir-opt \
 --load-pass-plugin=/build/lib/mqt-core-plugins-catalyst.dylib \
 --load-dialect-plugin=/build/lib/mqt-core-plugins-catalyst.dylib \
 --catalyst-quantum-tomqtopt \
 mqtopt_program.mlir \
 -o quantum_program.mlir
```

Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

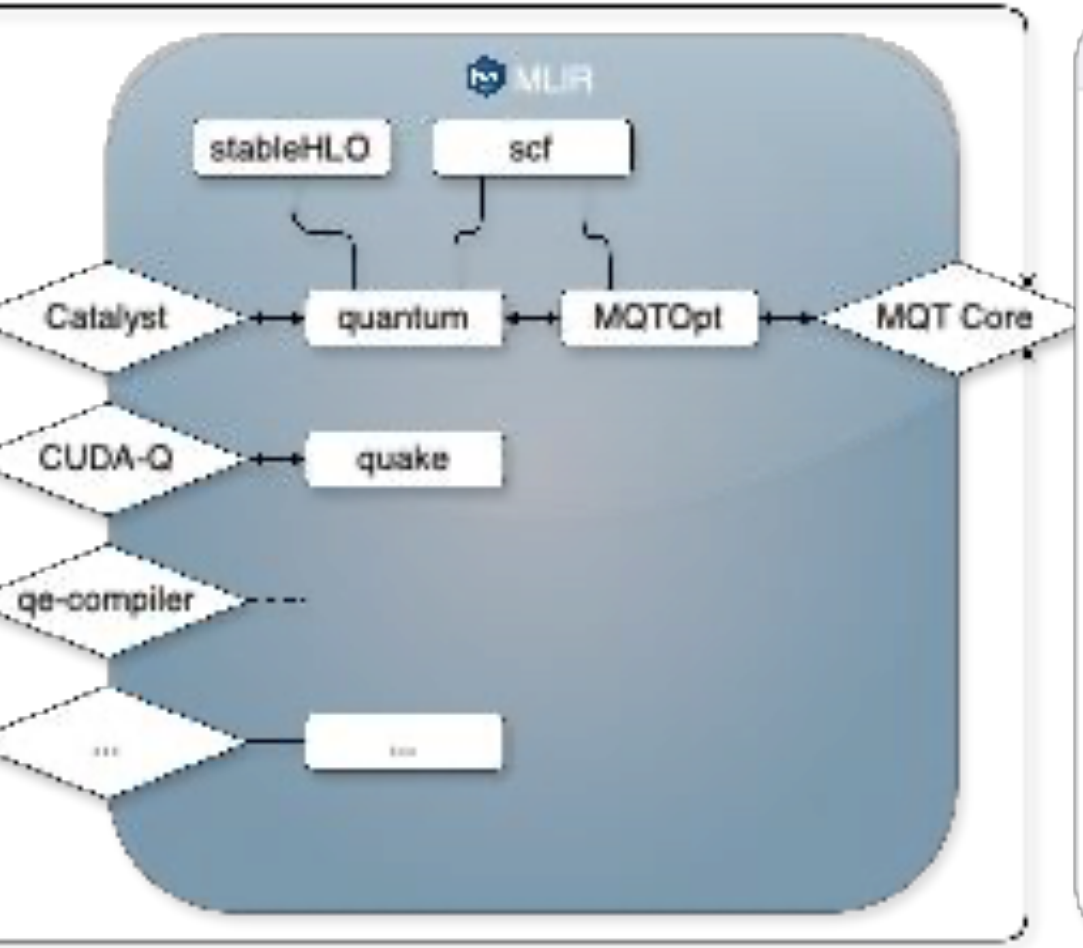
Intermediate Representations – MLIR Plugin




```
mlir-opt \
 --load-pass-plugin=/build/lib/mqt-core-plugins-catalyst.dylib \
 --load-dialect-plugin=/build/lib/mqt-core-plugins-catalyst.dylib \
 --mqtopt-to-catalyst-quantum \
 --catalyst-quantum-to-mqtopt \
 quantum_program.mlir \
 -o quantum_program_roundtrip.mlir
```

Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

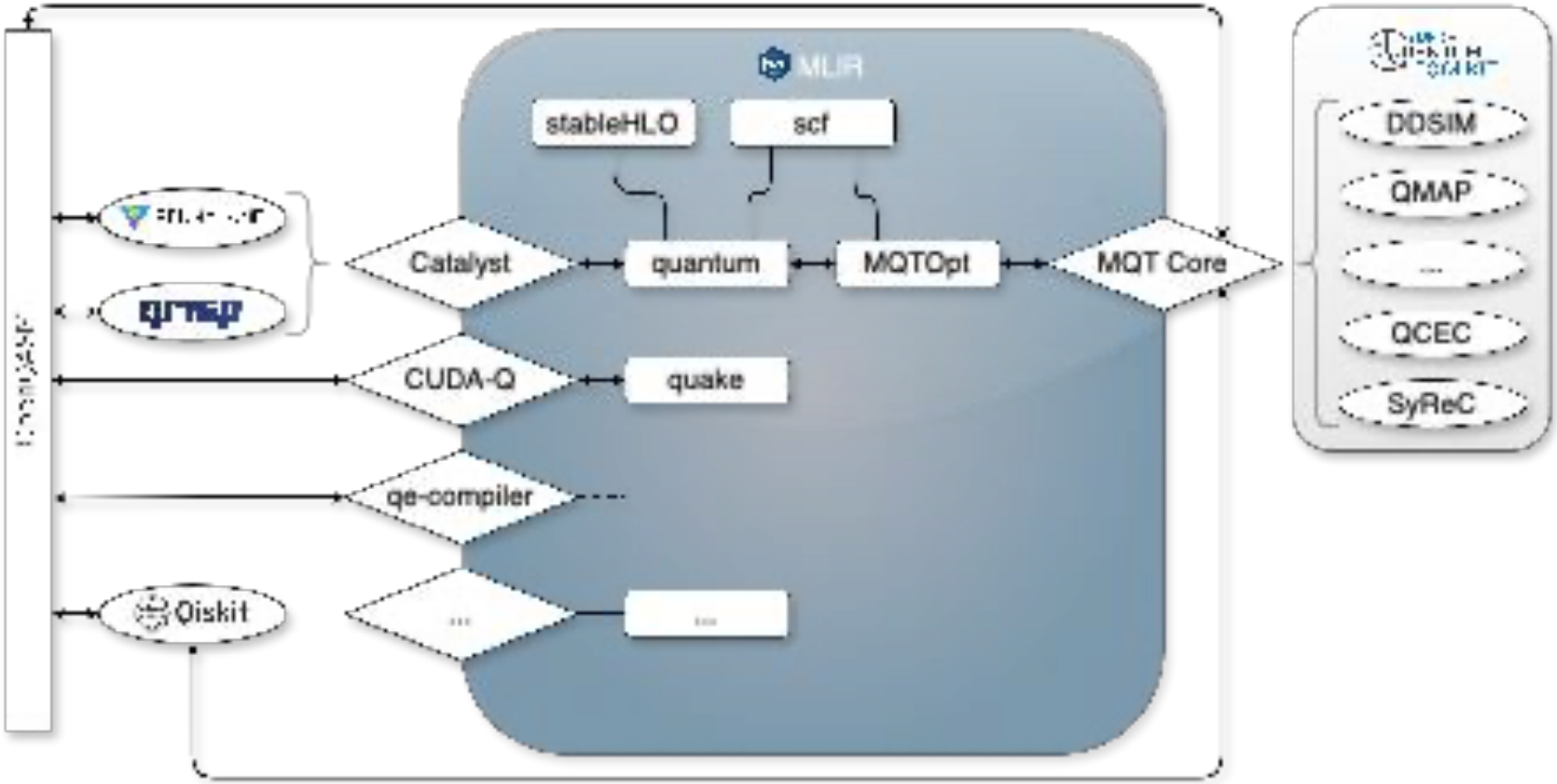
Intermediate Representations – MLIR Plugin




```
mlir-opt \
 --load-pass-plugin=/build/lib/mqt-core-plugins-catalyst.dylib \
 --load-dialect-plugin=/build/lib/mqt-core-plugins-catalyst.dylib \
 --mqtopt-to-catalyst-quantum \
 --mqt-qmap \
 --catalyst-quantum-to-mqtopt \
 quantum_program.mlir \
 -o quantum_program_roundtrip.mlir
```

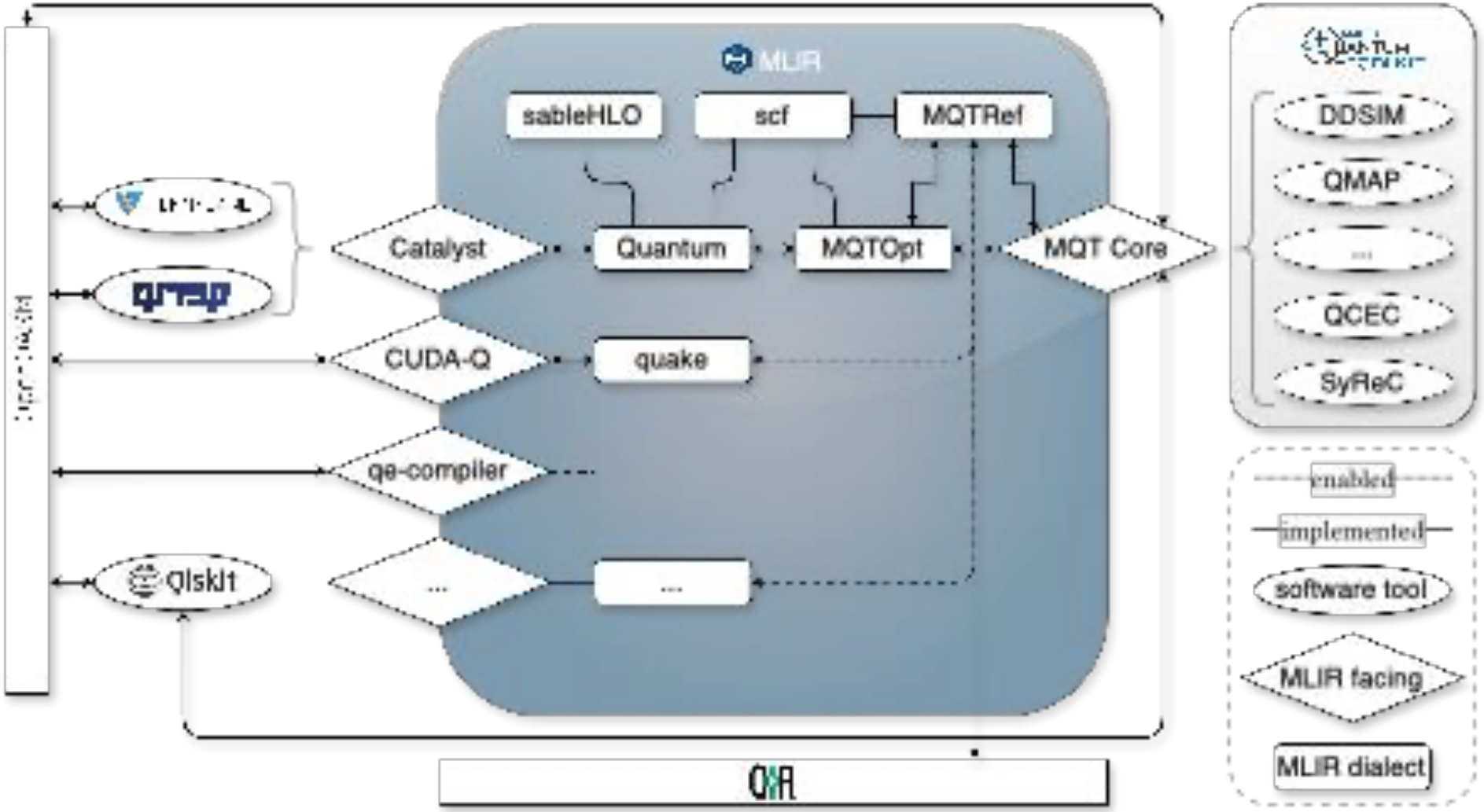
Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representations – MLIR Plugin



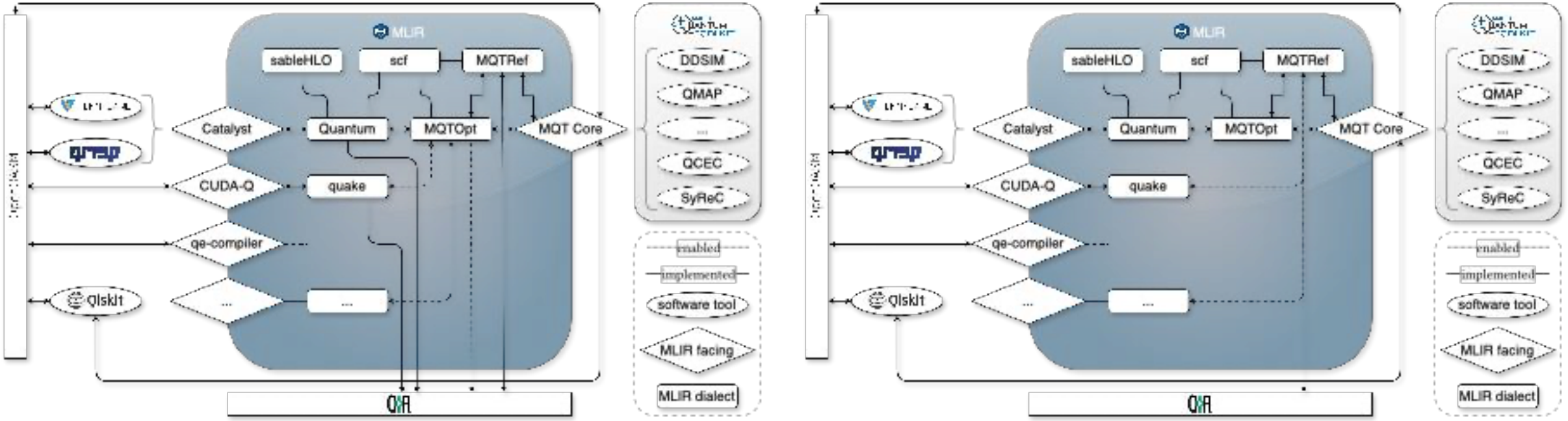
Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representations – MLIR Ecosystem



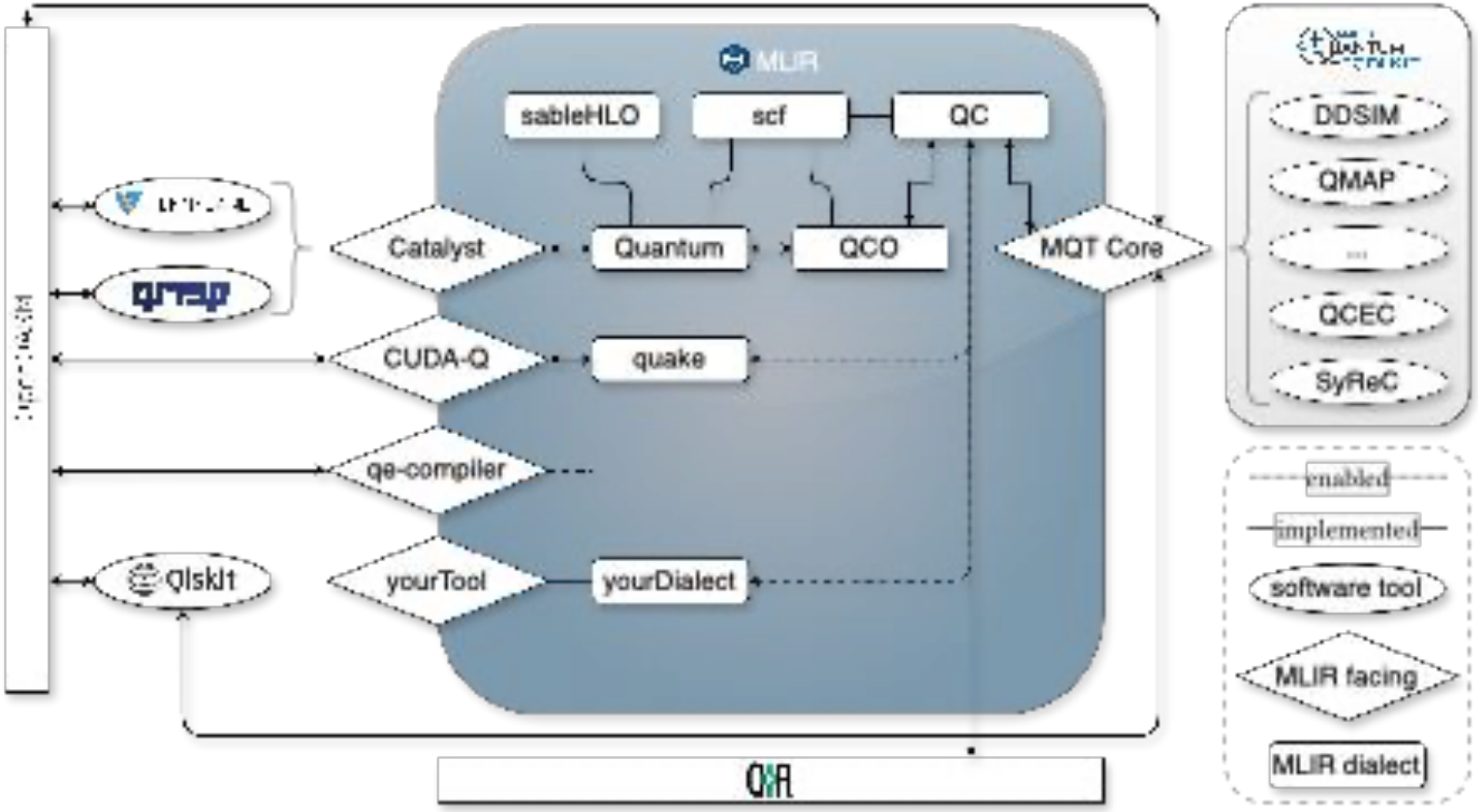
Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representations – MLIR Ecosystem



Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Intermediate Representations – MLIR Ecosystem



Hopf, P., Ochoa Lopez, E., Stade, Y., Rovara, D., Quetschlich, N., Florea, I., Izaac, J., Wille, R., & Burgholzer, L. (2026). Integrating Quantum Software Tools with(in) MLIR. In *SCA/HPCAsia 2026: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. Association for Computing Machinery.

Summary and Outlook

Version Compatibility
Can we align LLVM versions?

Packaging & Distribution
How to ship all of this?

Uniform Exchange Format
Can we agree on some common format?

Steep Learning Curve
Complex MLIR concepts
C++

MUNICH QUANTUM SOFTWARE COMPANY

UBC

Unitary Foundation

QuEra

QUANTINUUM

BERKELEY LAB

XANADU

JEFF BRIDGES COMPILATION

github.com/unitaryfoundation/jeff

Q&A


```
pip install mqt-core-plugins-catalyst
```

github.com/munich-quantum-toolkit/core-plugins-catalyst