Service-oriented Quantum Applications

IAAS Research

University of Stuttgart

Martin Beisel, Benjamin Weder

[beisel, weder]@iaas.uni-stuttgart.de

Institute of Architecture of Application Systems

PlanQK SeQuenC EniQmA

Tutorial Structure

- Session 1 (09:00 - 10:30): An Introduction to Quantum Computing
- Session 2 (11:00 - 12:30): Quantum Software Engineering
- **Session 3 (14:00 - 15:30): Quantum Workflows**
  - Quantum Workflows
  - Service-oriented Quantum Applications
  - Introduction to Hands-On Session
  - Hands-On Session Part 1
- Session 4 (16:00 - 17:30): Operation of Hybrid Quantum Applications

Tutorial Structure

- Session 1 (09:00 - 10:30): An Introduction to Quantum Computing
- Session 2 (11:00 - 12:30): Quantum Software Engineering
- **Session 3 (14:00 - 15:30): Quantum Workflows**
  - Quantum Workflows
  - **Service-oriented Quantum Applications**
  - Introduction to Hands-On Session
  - Hands-On Session Part 1
- Session 4 (16:00 - 17:30): Operation of Hybrid Quantum Applications

Motivation

Motivation

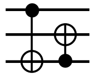
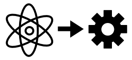
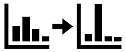
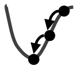
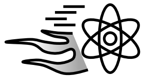
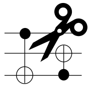
- Many quantum applications are implemented as monoliths
- Low reusability, understandability, scalability, maintainability, …

- Implement quantum applications as service-oriented architectures
  - Identify core functionalities
  - Conceptualize micro service architecture

IAAS Research

Quantum Service Ecosystem

Modularization

- Identify typical steps of a quantum application
  - Circuit Generation 
  - Circuit Execution 
  - Error Mitigation 
  - Optimization 
  - Result Evaluation 
  - Warm-Starting 
  - Circuit-Cutting 

- Separate application based on identified steps

Circuit Generation Service

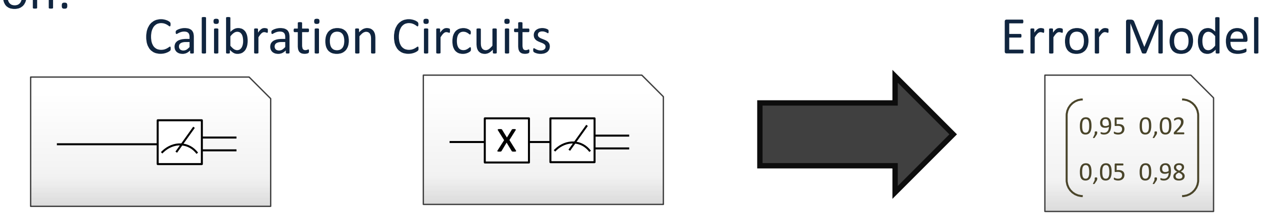
- Supports commonly used encodings, circuit fragments and entire circuits
  - Encodings:
    - Basis encoding, amplitude encoding, angle encoding, …
  - Circuit fragments:
    - Quantum phase estimation, quantum fourier transformation, …
  - Circuits:
    - QAOA, HHL, VQE, …

Circuit Execution Service

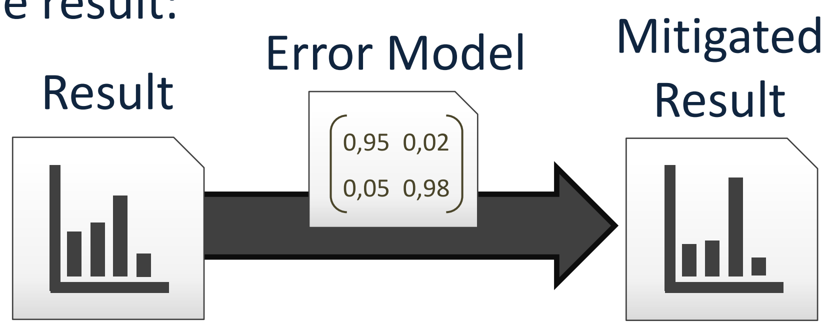
- Execution for various providers
  - IBM, IonQ: https://github.com/UST-QuAntiL/qiskit-service
  - Execution via AWS: https://github.com/UST-QuAntiL/braket-service
    - Rigetti, OQC, IonQ, QuEra, Xanadu
  - Q#: https://github.com/UST-QuAntiL/qsharp-service
  - Google: https://github.com/UST-QuAntiL/cirq-service

Error Mitigation Service

- Reduce impact of errors based on data about the quantum computer
- Example: Readout-error mitigation using the calibration matrix
  - Data collection:



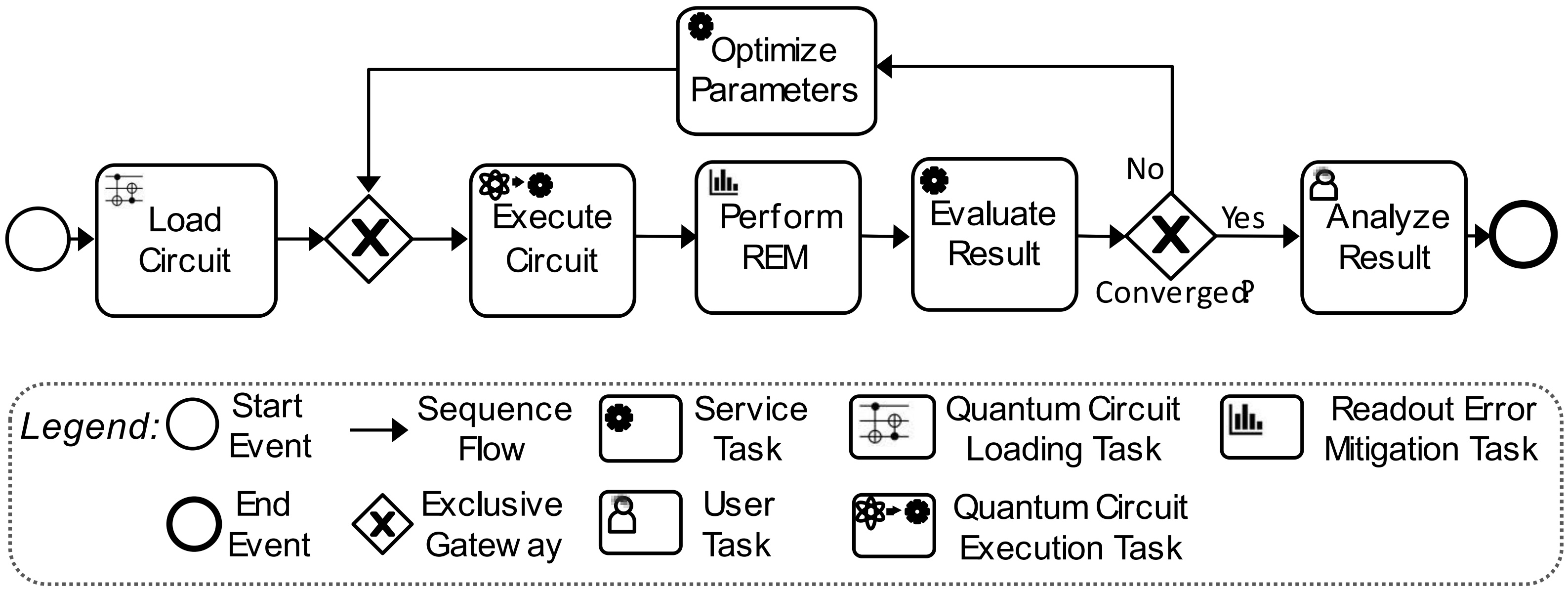
  - Mitigating the result:



- Error mitigation service enables generating, saving, reusing and sharing error models

Parameter Optimization Service

- Variational quantum algorithms require the optimization of circuit parameters



- Integrates various optimization algorithms provided by:
  - SciPy
  - Qiskit

Result Evaluation Service

- Many quantum algorithm require a classical evaluation of the circuit execution result

  Example:

  $$\{"\mathbf{01}": 700, "\mathbf{10}": 200, "\mathbf{11}": 100\}$$

  $$5 * 0.7 + 9 * 0.2 + 10 * 0.1 = 6.3$$

- Service supports
  - Objective functions, such as expectation value, Gibbs, Cvar
  - Cost functions, such as maximum cut problem, traveling salesman problem

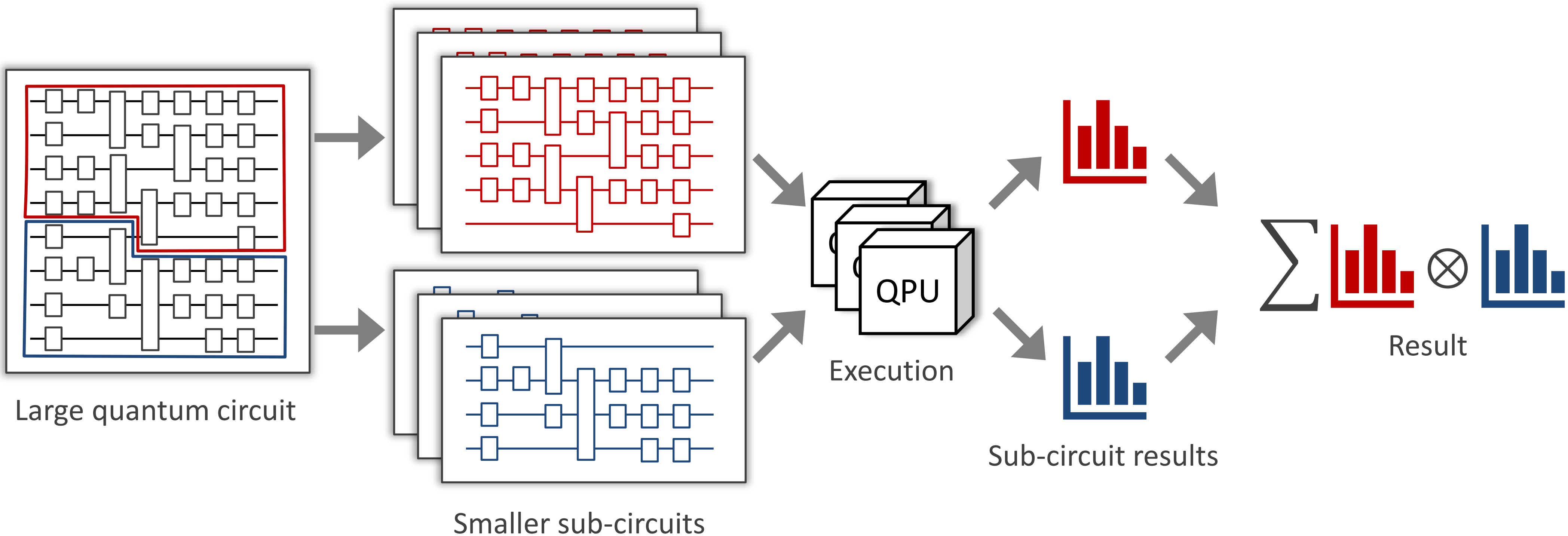
Warm-Starting Service

- Warm-Starting can improve the performance of quantum algorithms by:
  - Efficiently approximating classical solutions that can be encoded into the quantum circuit
  - Pre-computing initial parameters for the circuit optimization process

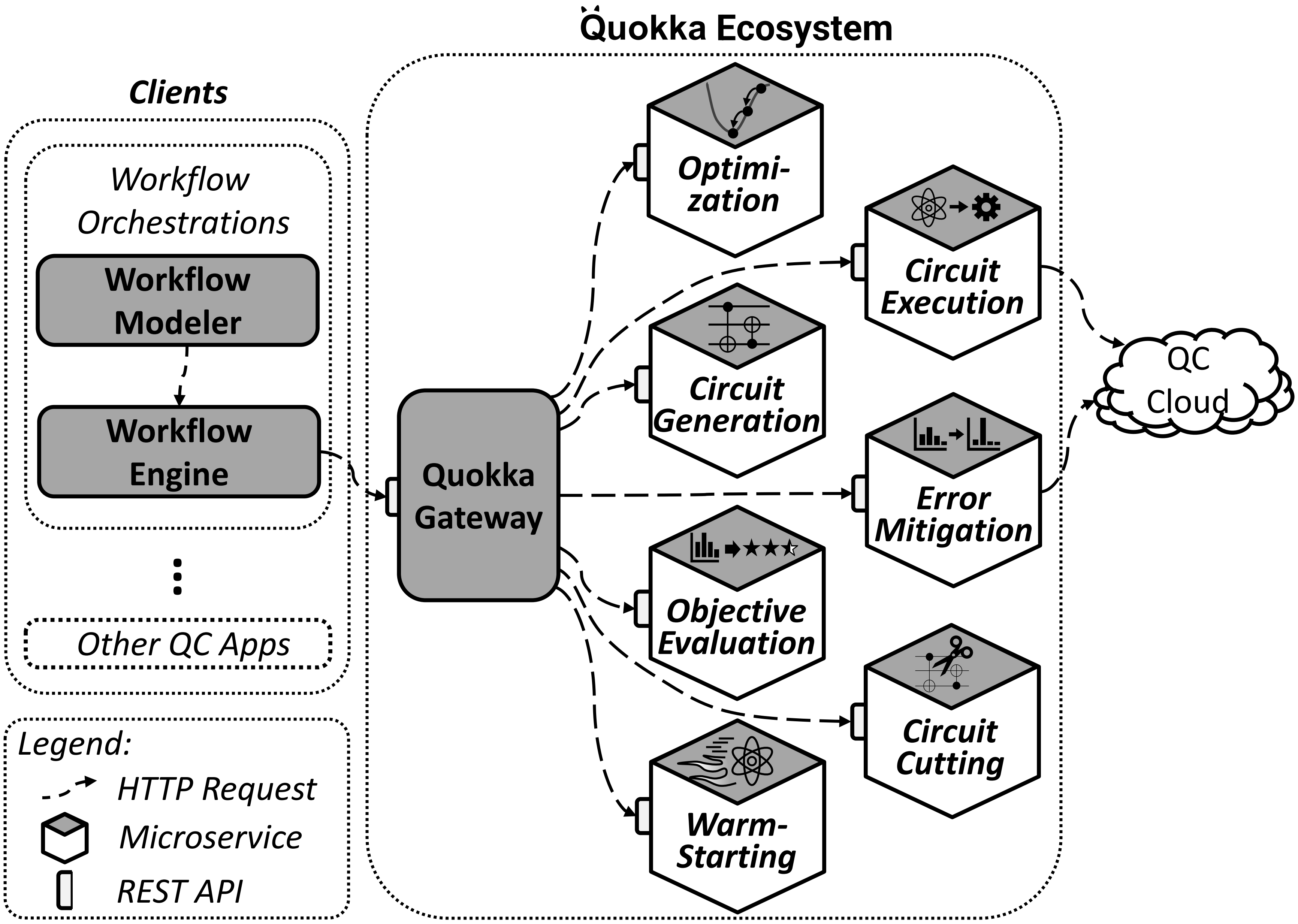
- Warm-Starting services enables users to easily warm-start their quantum algorithm

Quantum Circuit Cutting Service

- Quantum circuits might be too large (width, depth) to retrieve good results
- Execute multiple smaller circuits
  ➔ Classical post-processing to combine results



System Architecture



Hands-On Session