



iQD SDK –Optimization for Quantum Circuits

Ver 1.0.0 | May 2025

www.iqore.com | team@iqore.com

1. Introduction

The iQore SDK enables developers, researchers, and enterprises to apply secure, cloud-hosted optimization logic to quantum circuits without modifying existing hardware or algorithm logic. Using iQD, the SDK optimizes circuits in real-time, improving quantum gate efficiency, entanglement, and fidelity.

Our solution injects the optimization logic securely, never exposing proprietary code, and supports both simulators and IBM QPUs.

2. Key Features:

- **Secure Cloud Injection**
All optimization logic is securely injected from the cloud, never touching your machine or source code.
- **Qiskit-Compatible**
Works seamlessly with any valid QuantumCircuit, enabling quantum optimization without complex setup.
- **Non-Invasive Workflow**
Simply call `optimize()`, save your token, and run your circuit. No extra setup needed.
- **Backend-Agnostic**
Supports simulators like `aer_simulator` and real quantum backends from IBM Quantum, including legacy systems.

3. Quickstart Guide

Install the SDK

```
pip install iqore
```

© 2025 iQore, Inc. All rights reserved.

The iQore SDK and all related technologies are provided under a proprietary license. This documentation is intended for public informational use by developers and partners integrating iQore systems. Redistribution of the SDK or reverse engineering of proprietary components is prohibited. For licensing inquiries, visit www.iqore.com or contact team@iqore.com.

Authenticate with Your Token

```
from iQore import iQoreRuntimeService

# Save a token
iQoreRuntimeService.save_account("Your_Token_Here")

# Retrieve the token
token = iQoreRuntimeService.get_token()
```

Optimize Your Circuit

```
from qiskit import QuantumCircuit
from qiskit_aer import AerSimulator
from iQore import iQD
import numpy as np

# Quantum backend and matrix
backend = AerSimulator()
matrix = np.random.rand(4, 4)
qc = QuantumCircuit(4)

# Optimize the quantum circuit
iQD.optimize(
    qc,
    qubit_count=4,
    backend=backend,
    matrix=matrix,
    enable_iQD_dtc=True
)

# Sample user logic
qc.cx(0, 1) # Apply a user-defined gate
qc.measure_all() # Measure all qubits

# Display the optimized circuit
print(qc.draw("text"))
```

© 2025 iQore, Inc. All rights reserved.

The iQore SDK and all related technologies are provided under a proprietary license. This documentation is intended for public informational use by developers and partners integrating iQore systems. Redistribution of the SDK or reverse engineering of proprietary components is prohibited. For licensing inquiries, visit www.iqore.com or contact team@iqore.com.



4. Core Components

iQoreRuntimeService

Handles token management and securely stores user authentication.

iQD.Optimize

Performs quantum circuit optimization via secure, cloud-delivered logic.

5. Security & Token Architecture

How It Works

1. Authenticate Securely

You sign in using your personal iQore token — no passwords, no keys to manage.

2. Connect to the iQore Optimizer Cloud

When you call `iQD.optimize()`, your circuit is sent through a secure, encrypted request.

3. Get Your Upgraded Circuit

You receive a fully optimized Qiskit circuit — ready to run — with zero risk of your code or our IP being exposed.

6. Circuit Optimization API

Parameter	Type	Description
qc	QuantumCircuit	Qiskit circuit to optimize
qubit_count	int	Number of active qubits
backend	str or Backend	Backend used for execution or optimization guide
matrix	np.ndarray (<i>Optional</i>)	Internal enhancement tuning matrix
enable_iQD_dtc	bool (<i>Optional</i>)	Enables Dynamic Tensor Controller (DTC) mode

© 2025 iQore, Inc. All rights reserved.

The iQore SDK and all related technologies are provided under a proprietary license. This documentation is intended for public informational use by developers and partners integrating iQore systems. Redistribution of the SDK or reverse engineering of proprietary components is prohibited. For licensing inquiries, visit www.iqore.com or contact team@iqore.com.



7. Real-World Use Cases

- **Quantum Circuit Research:** Enhance algorithms like QAOA and VQE
- **Simulators:** Optimize for statevector, Aer, and other simulation models
- **IBM QPUs:** Submit optimized circuits directly to Qiskit Runtime or legacy IBMQ systems
- **Fidelity Improvement:** Increase the coherence and fidelity of quantum systems with minimal changes to user code

8. Support & Licensing

The iQore SDK is licensed under a proprietary license. Redistribution, reverse engineering, or extraction of embedded logic is prohibited.

For commercial licensing or enterprise integration, contact:

- **Email:** team@iqore.com
- **Website:** <https://iqore.com>

About iQore

iQore develops quantum-classical optimization technologies that help developers, researchers, and enterprises extract better outcomes from quantum computing systems. With a focus on secure, high-fidelity optimizations, iQore delivers smarter performance and more efficient execution for today's quantum hardware.

© 2025 iQore, Inc. All rights reserved.

The iQore SDK and all related technologies are provided under a proprietary license. This documentation is intended for public informational use by developers and partners integrating iQore systems. Redistribution of the SDK or reverse engineering of proprietary components is prohibited. For licensing inquiries, visit www.iqore.com or contact team@iqore.com.