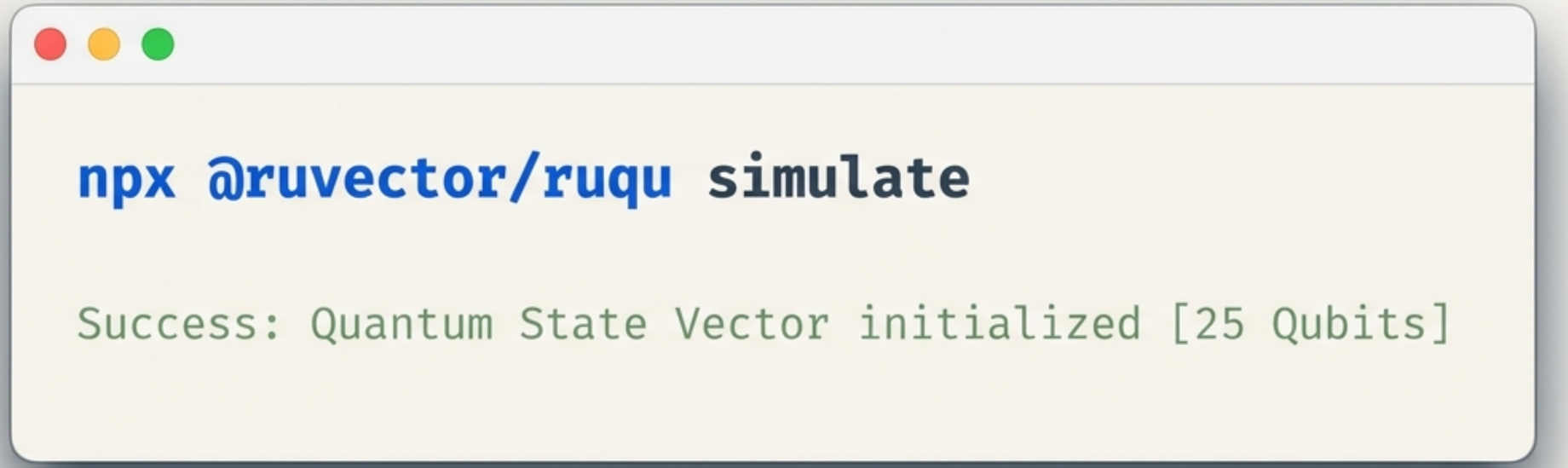


The Quantum Computer in Your Browser

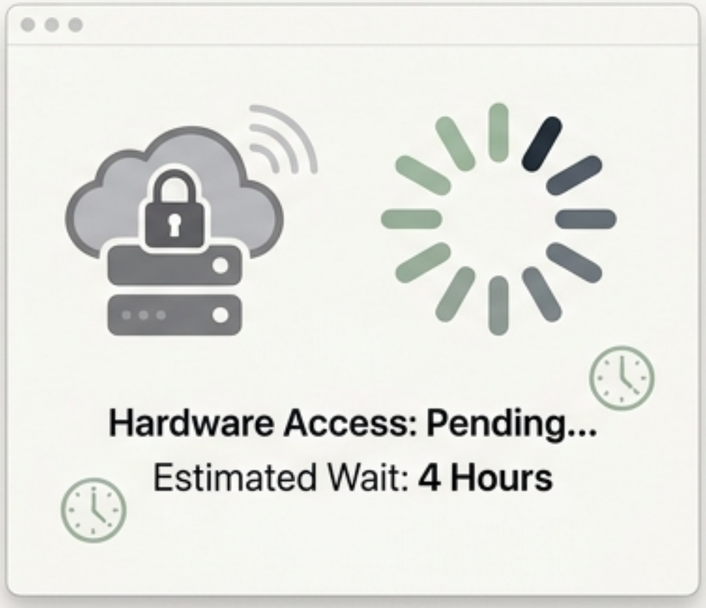
A pure Rust, zero-hardware quantum sandbox that runs locally.
No cloud queues. No physics degree required.

The Friction of the Status Quo

The Dependency Hell

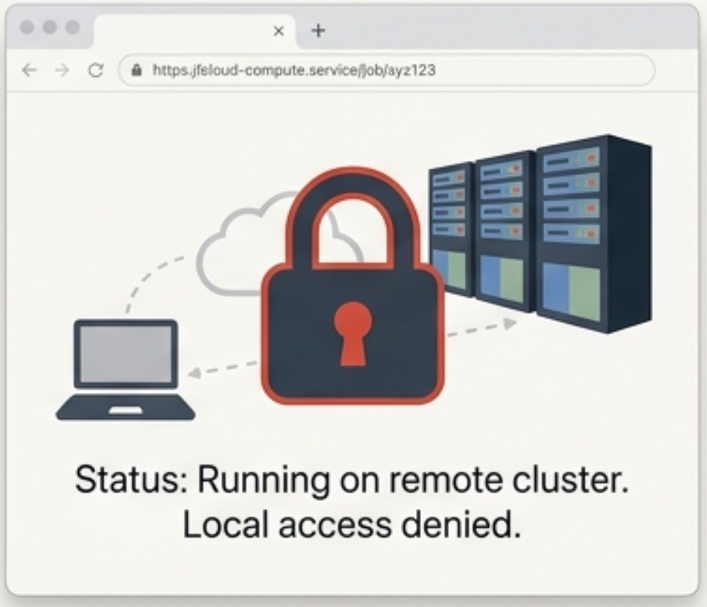
Heavy Python stacks fail to install and break environments.

The Hardware Queue



Real QPU access is gatekept, expensive, and slow.

The Black Box



Computations happen on distant servers, impossible to run purely in local web apps.

Instant Quantum Execution, Anywhere

Zero install. Zero hardware. The whole class touches quantum in the first five minutes.

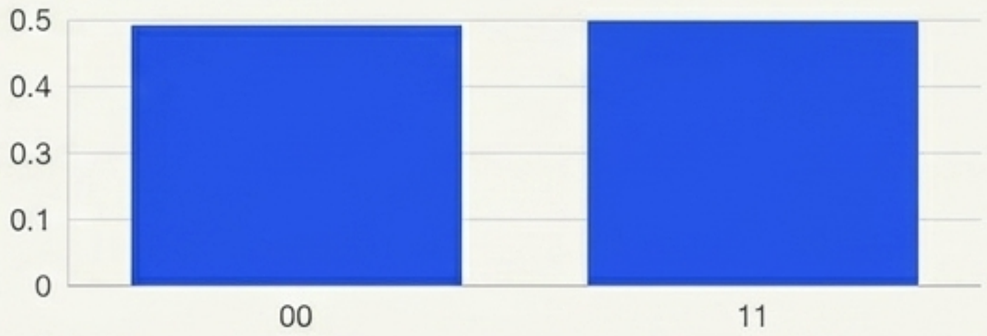
The Terminal

```
npx @ruvector/ruqu simulate --qubits 2

{"00": 0.499, "11": 0.501}
```

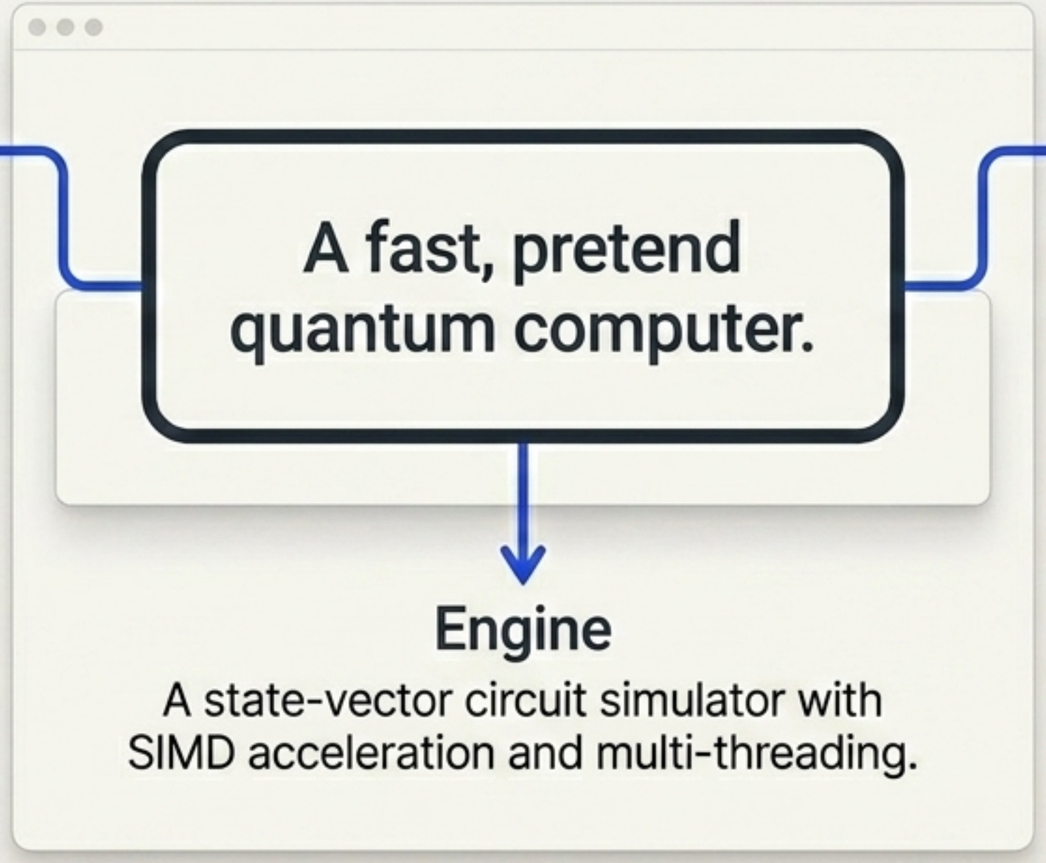
The Browser

```
const sim = new Simulator();
let circuit = new Circuit(2);
circuit.h(0);
circuit.cnot(0,1);
sim.run(circuit);
```



What exactly is ruqu?

It perfectly imitates a real quantum machine so you can build and test quantum ideas before the expensive hardware exists.



Language

Written from scratch in Pure Rust. No FFI. No C++. No Python dependencies.

Target

Compiles directly to Native and WebAssembly (wasm32).

Engine

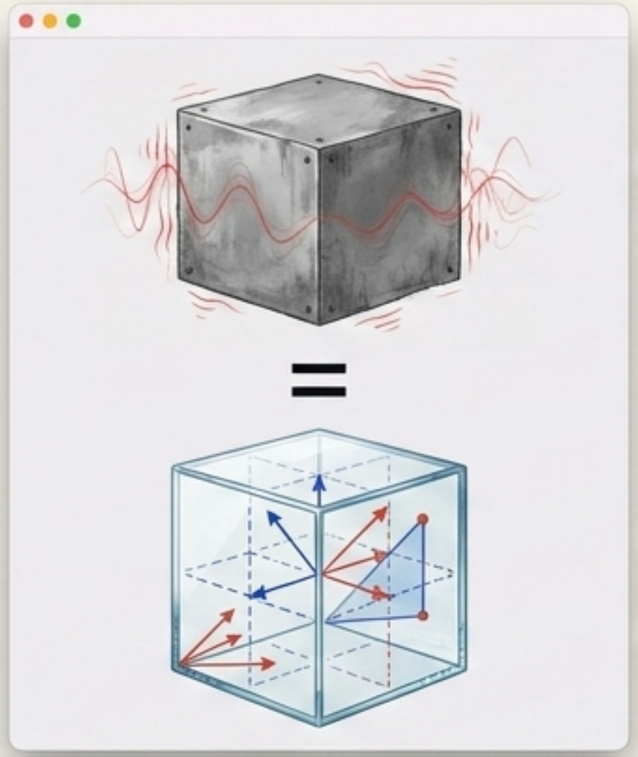
A state-vector circuit simulator with SIMD acceleration and multi-threading.

The Lightweight Alternative

Dimension	Legacy Python Stacks	ruqu
Setup Time	Minutes / Hours	**0 seconds (npx)**
Dependencies	Heavy Python tree	**None (Pure Rust)**
Browser Support	Server-only	**Native WASM**
Execution Engine	Python-bound	**SIMD-Accelerated**
Hardware Required	Needs Cloud QPU	**Zero Hardware**

How the 'Pretend' Engine Works

ruqu holds the full quantum state in standard computer memory. It applies every gate using exact linear algebra.



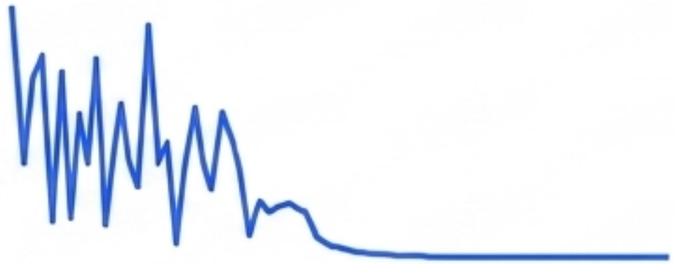
Real Quantum Hardware
(Noisy, physical, error-prone)

ruqu Engine
(Clean math, perfect theoretical accuracy)

It tracks every single possibility simultaneously, giving you the perfect theoretical answer up to 32 qubits.

Batteries Included: Production Algorithms

VQE (Chemistry)

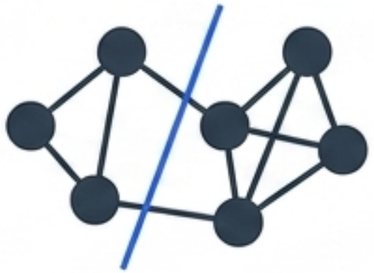


Finds ground-state energies of molecular Hamiltonians without hardware.

Grover (Search)

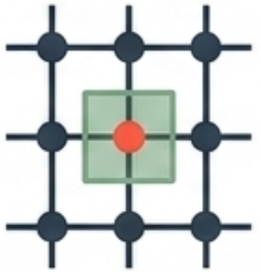
O(√N) quadratic speedup for unstructured search via amplitude amplification.

QAOA (Optimization)



Approximate solutions to combinatorial optimization problems like `MaxCut`.

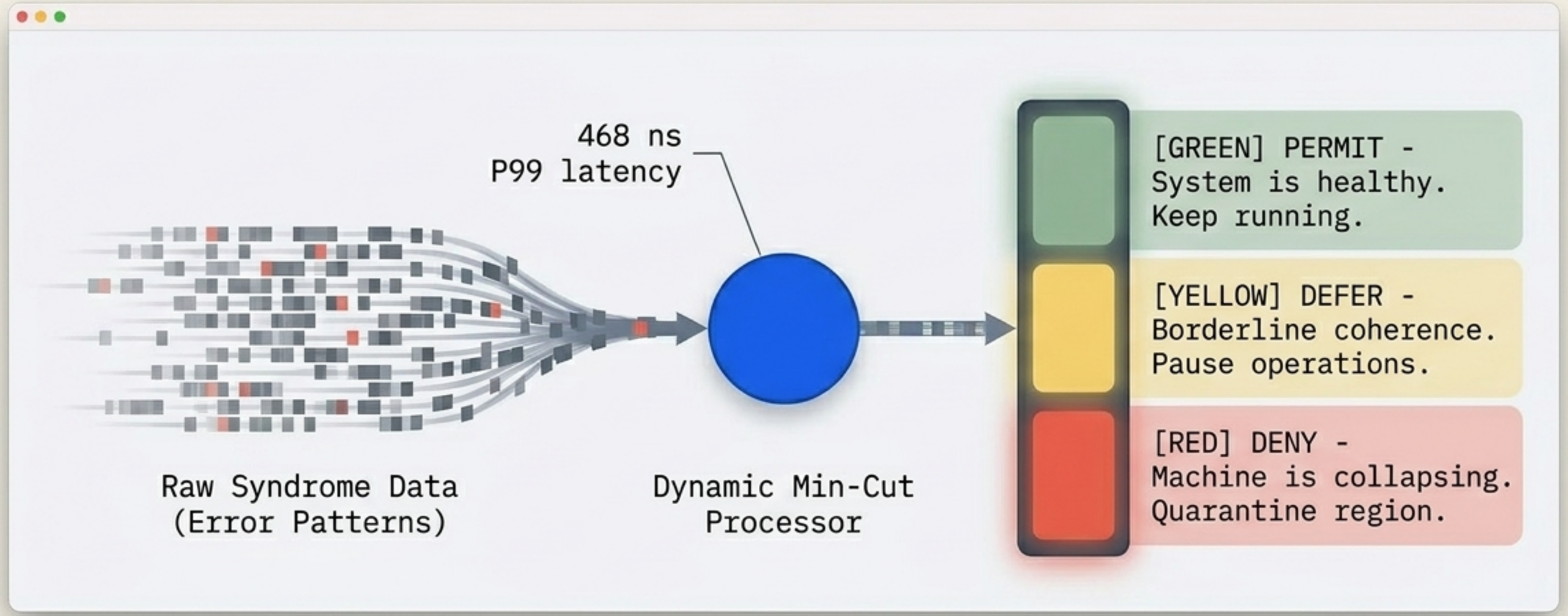
Surface Code (Correction)



Distance-3 quantum error correction with syndrome decoding.

The Nervous System: ruQu Coherence

A classical health monitor that catches structural breakdowns before they wipe out your progress.



The 5-Crate Architecture

ruqu-wasm & CLI
WebAssembly bindings and the npx CLI agent wrapper.

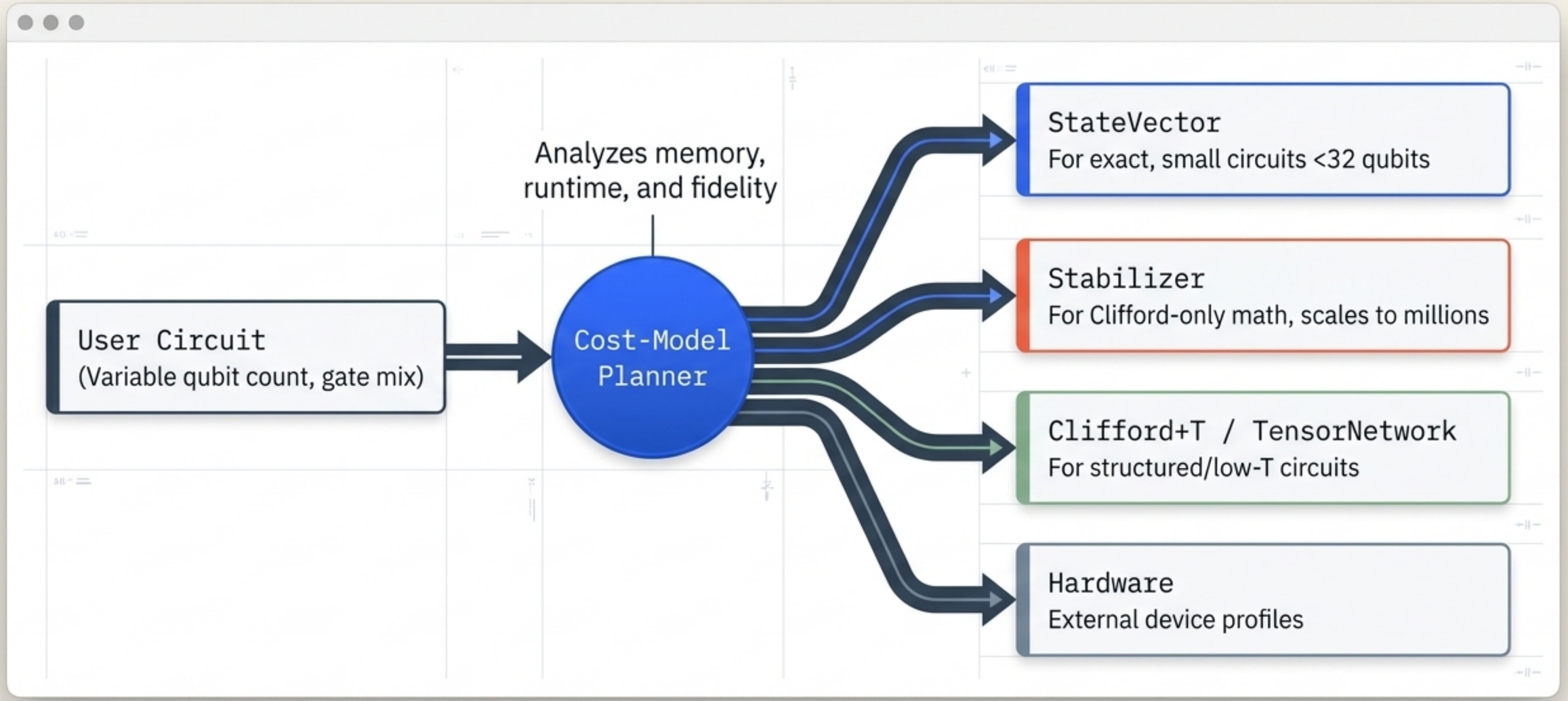
ruQu
The coherence engine and real-time safety gate.

ruqu-exotic
Experimental quantum-inspired tools for AI (memory decay, swarm interference).

ruqu-algorithms
VQE, Grover, QAOA, Surface Code.

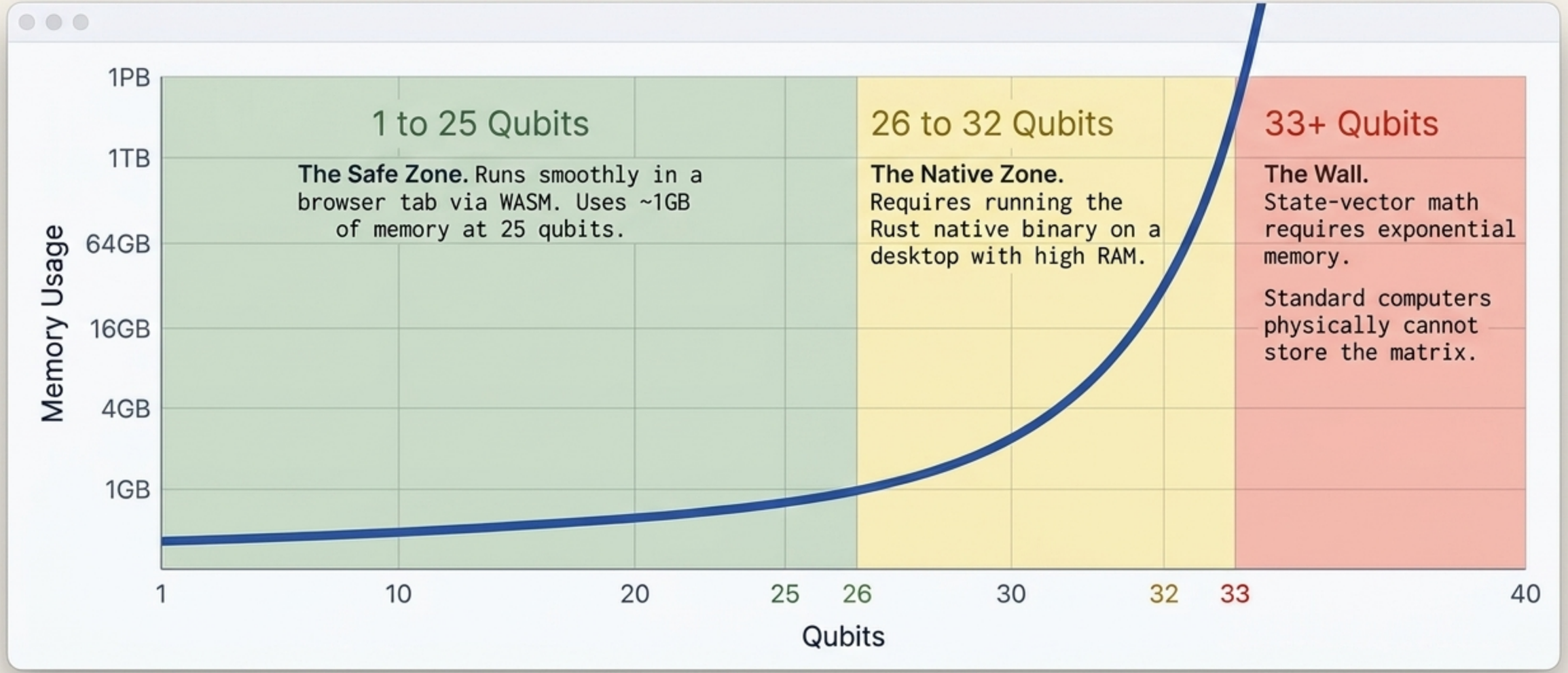
ruqu-core
High-performance state-vector simulator. SIMD, multi-threading, realistic noise.

The Smart Router: Auto-Selecting Backends

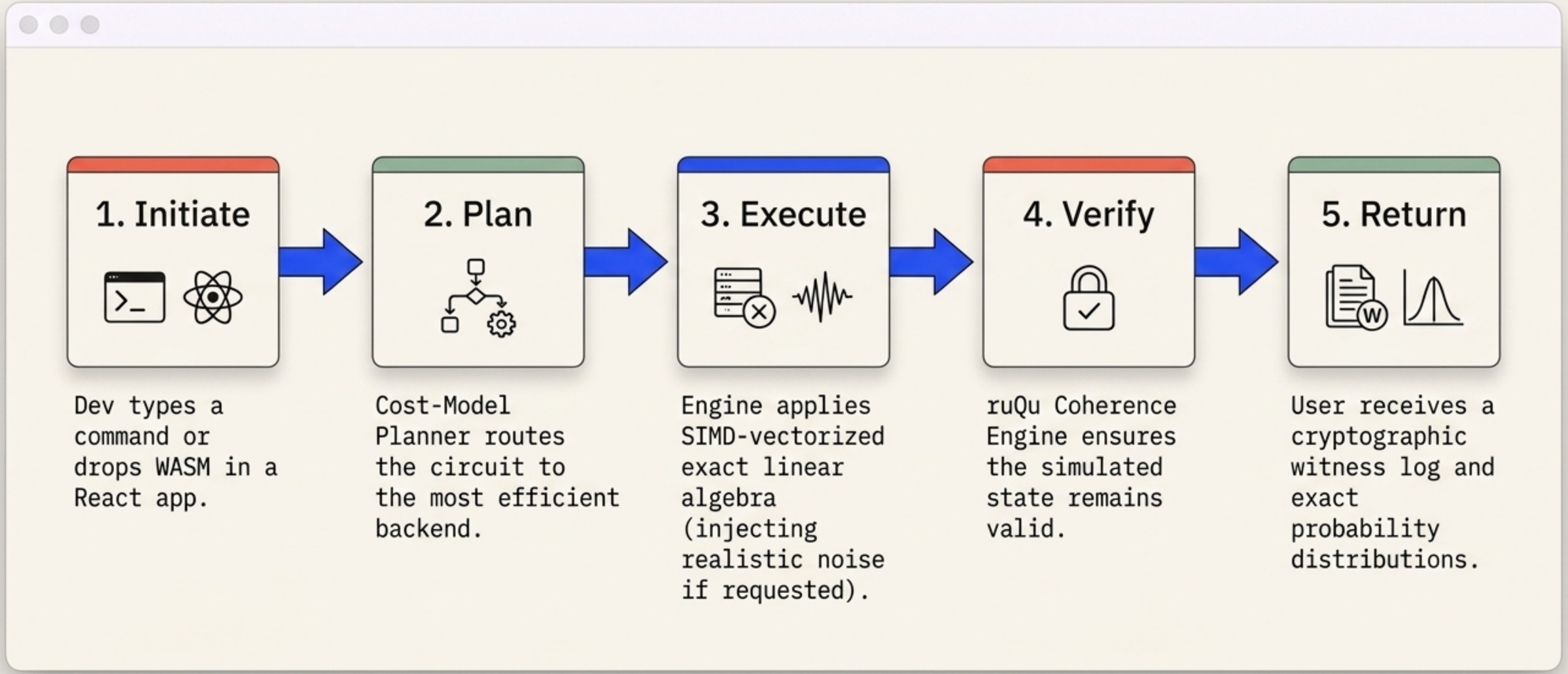


Honest Limits: The Boundary Map

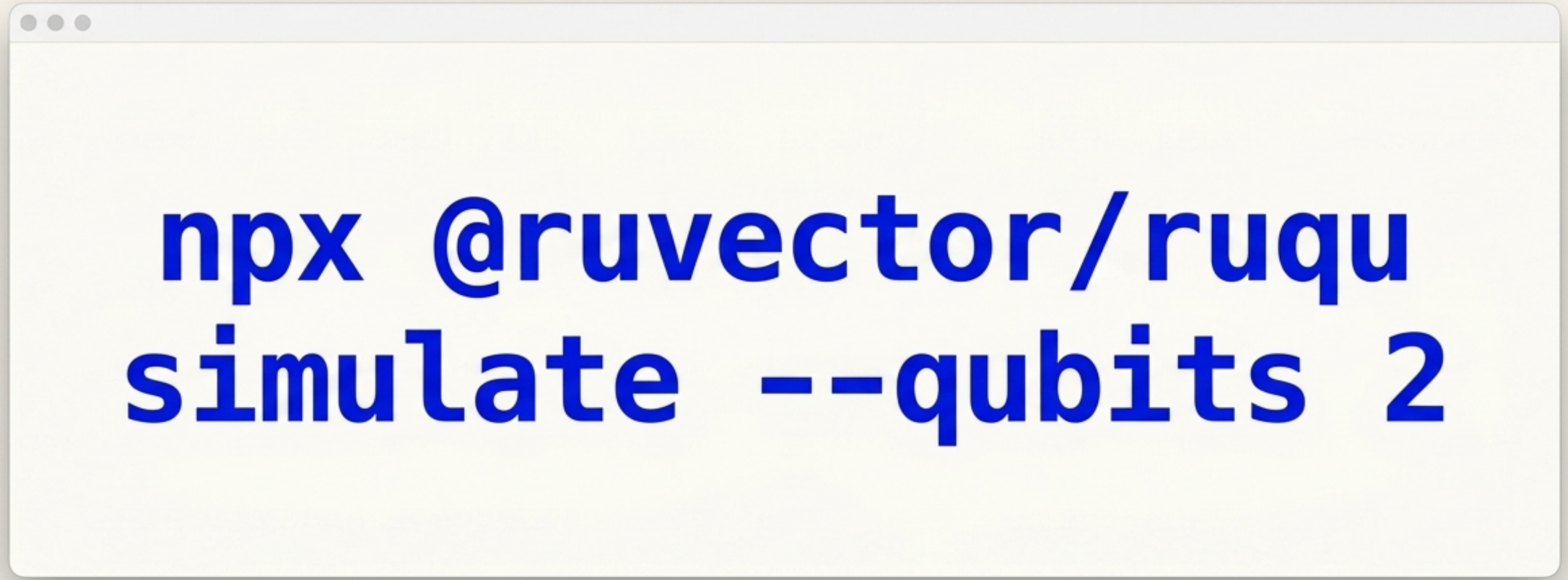
Takeaway: It is a simulator. It models noise, but hardware limits dictate its capacity.



From Command to Quantum Output



Start your pretend quantum computer right now.



Open your terminal. No installation required.