

# Schema Driven Experiment Portals - Quantum R&D

Low-code UI and pipeline generation transforms quantum research workflows

By

Vamsi Praveen Karanam

Senior Software Engineer



# The Notebook Problem

## Constant Change

Hardware, error-mitigation, parameters evolve rapidly

## Hidden Dependencies

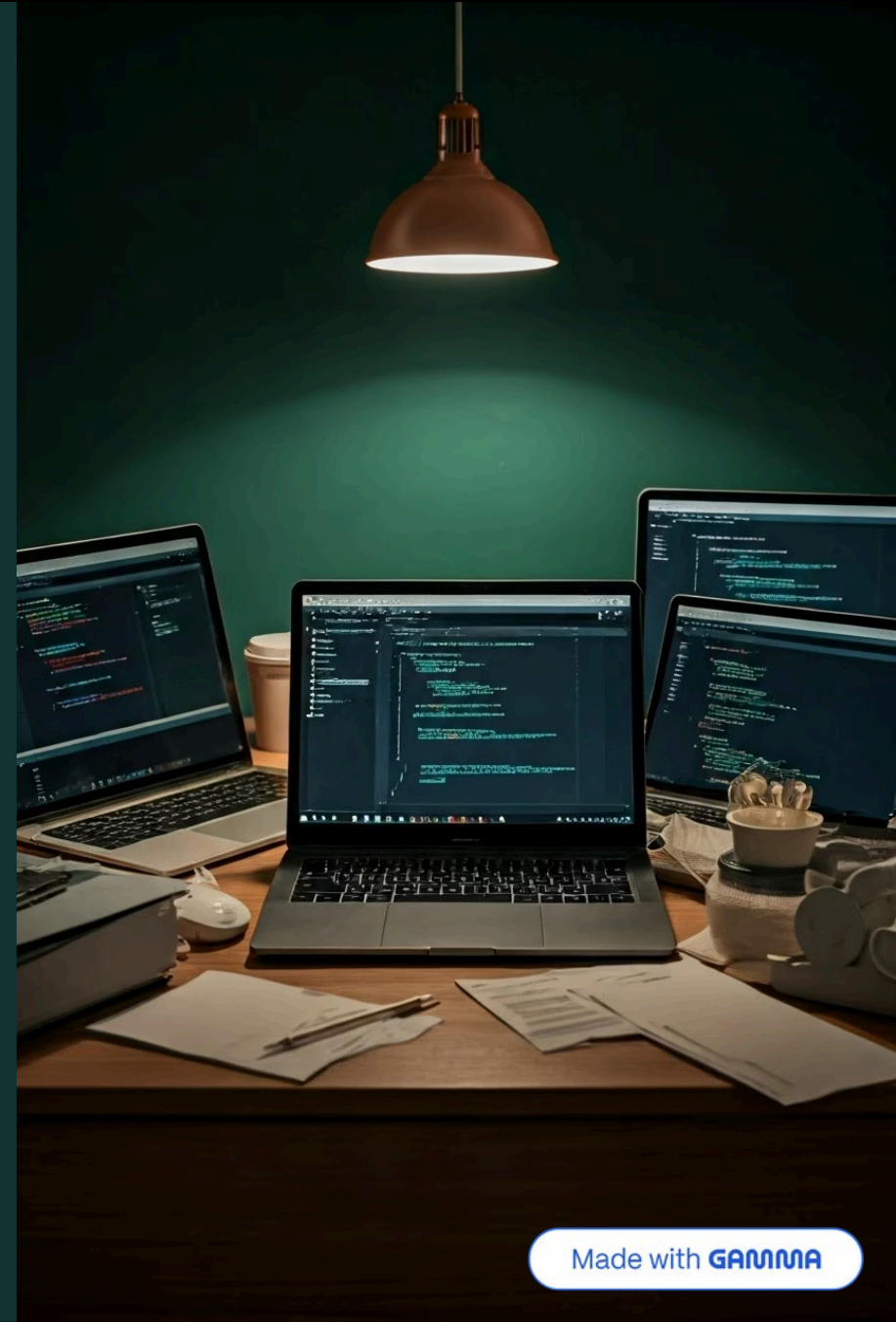
Notebook copies accumulate errors

## Documentation Drift

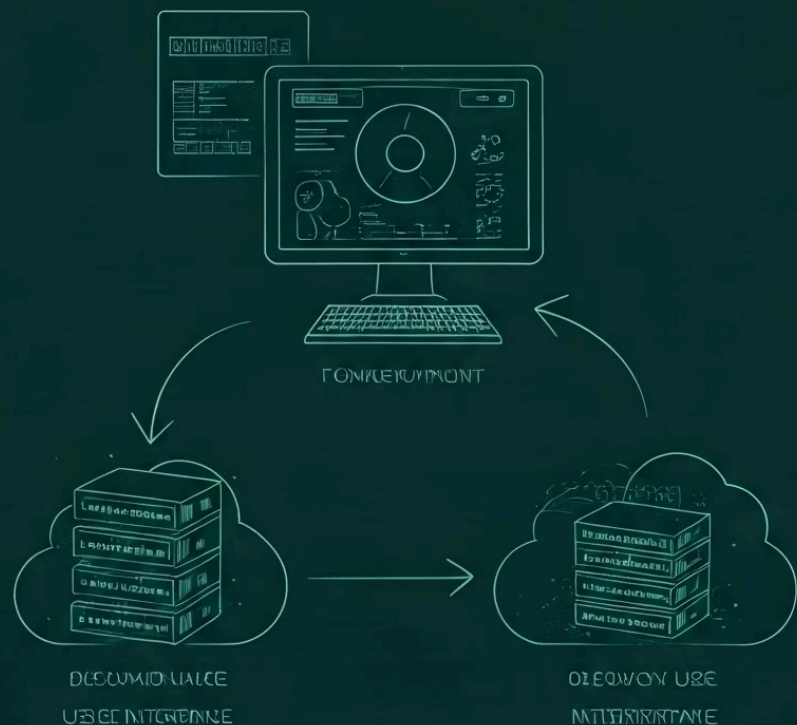
Code diverges from documentation

## Compliance Barriers

Audit trails blocked by inconsistency



# Schema-Driven Solution



## JSON Schema

Single source of truth for experiment parameters



## React UI Generation

Automatic control panel creation



## Step Functions Workflow

Distributed parallel execution



## Audit Trail

Complete lineage and reproducibility

# System Architecture

## Schema Registry

S3 bucket with Object Lock for immutable versions

## Front-End Generator

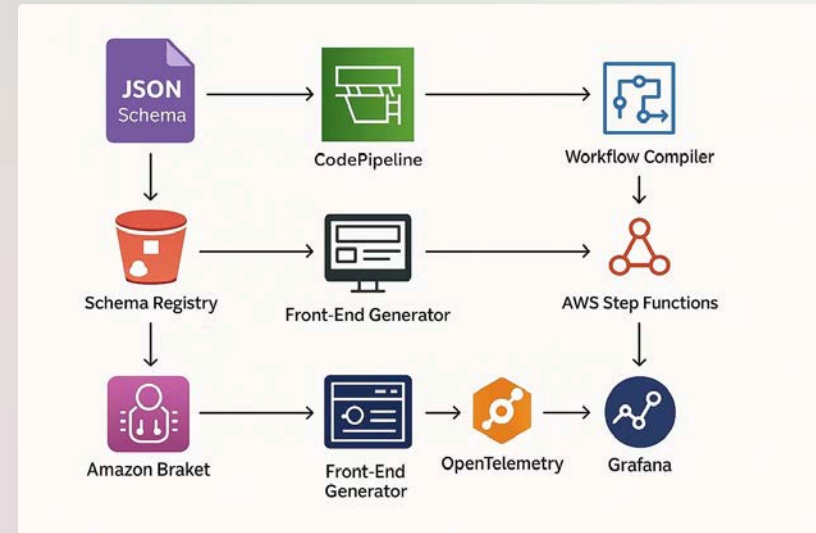
TypeScript tool creates React components

## Workflow Compiler

CDK construct produces Step Functions definition

## Observability

OpenTelemetry spans with automatic dashboards



# Schema vs. Notebook

## Traditional Notebook

```
import braket.aws as braket
from my_postprocess import summarise

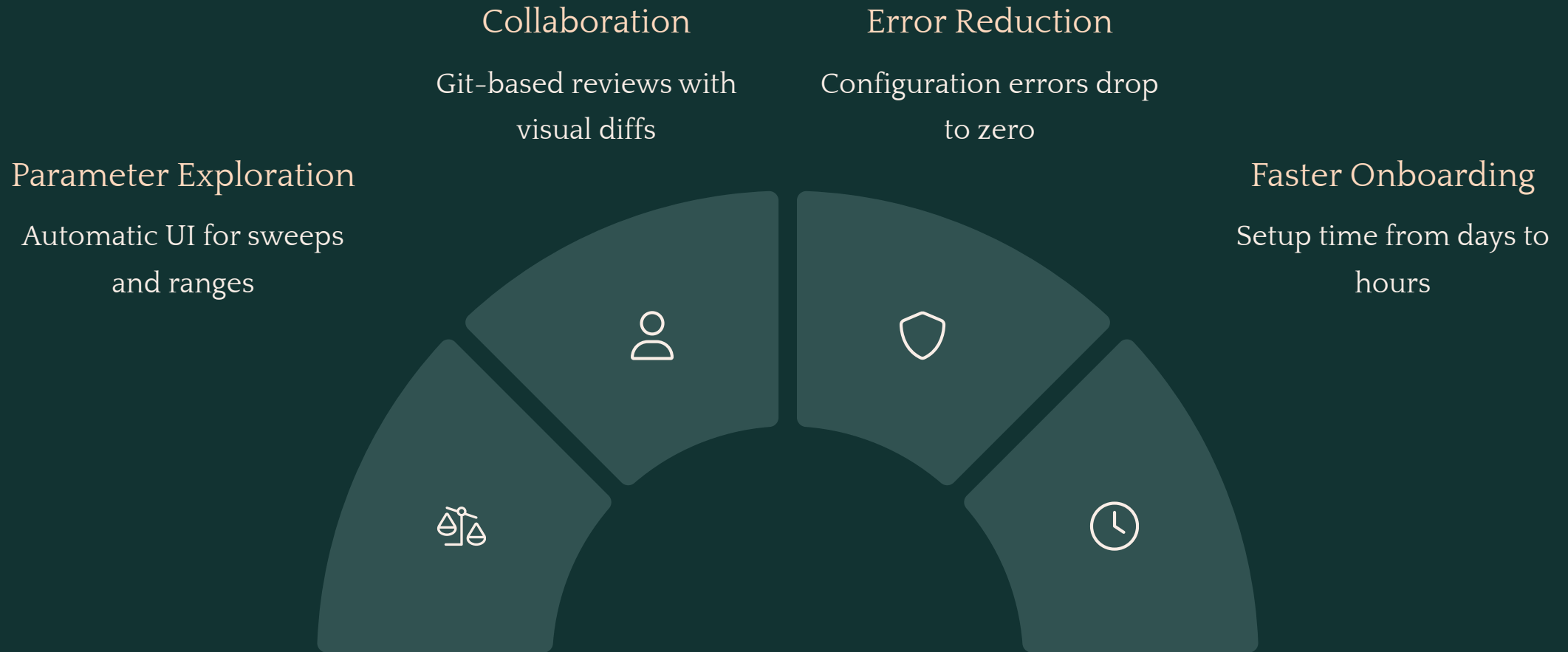
device = braket.AwsDevice(
    "arn:aws:braket:::device/qpu/ionq/ionQdevice"
)
qasm = open("vqe_depth3.qasm").read()

task = device.run(
    program=qasm,
    shots=10_000,
    poll_timeout_seconds=7_200
)
result = task.result()
summarise(result, output_path="s3://lab-data/vqe/run42/")
```

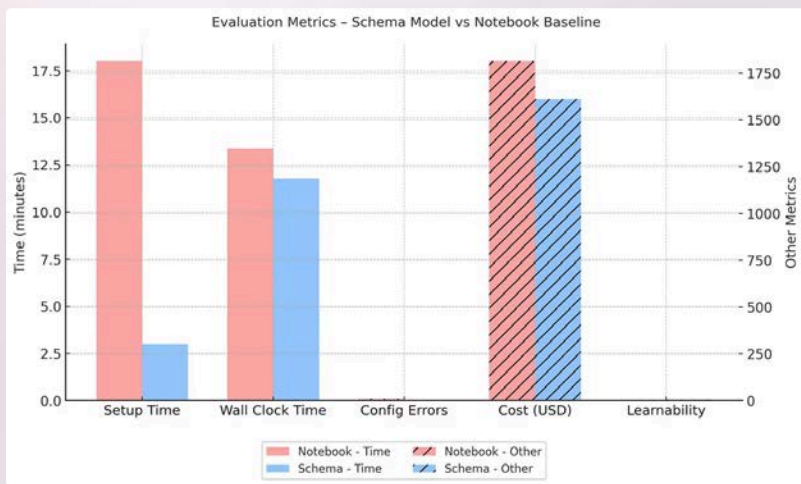
## JSON Schema Approach

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "VQE Sweep",
  "type": "object",
  "properties": {
    "circuit": { "const": "vqe_depth3.qasm" },
    "deviceArn": {
      "enum": [
        "arn:aws:braket:::device/qpu/ionq/ionQdevice",
        "arn:aws:braket:::device/quantum-simulator/amazon/sv1"
      ]
    },
    "shots": { "type": "integer", "minimum": 1, "maximum": 100000 },
    "outputPath": { "type": "string", "format": "uri" }
  },
  "required": ["deviceArn", "shots", "outputPath"]
}
```

# Key Benefits



# Performance Evaluation



Metric	Notebook	Schema
Setup time per variant	18 min	3 min
Wall clock (1M shots)	13h 22m	11h 47m
Configuration errors	7	0
AWS charges	\$1,814	\$1,612
User learnability (1-5)	3.1	4.8



# Implementation Details

## CDK Project

Three nested stacks for frontend, pipeline, observability

## Artifacts

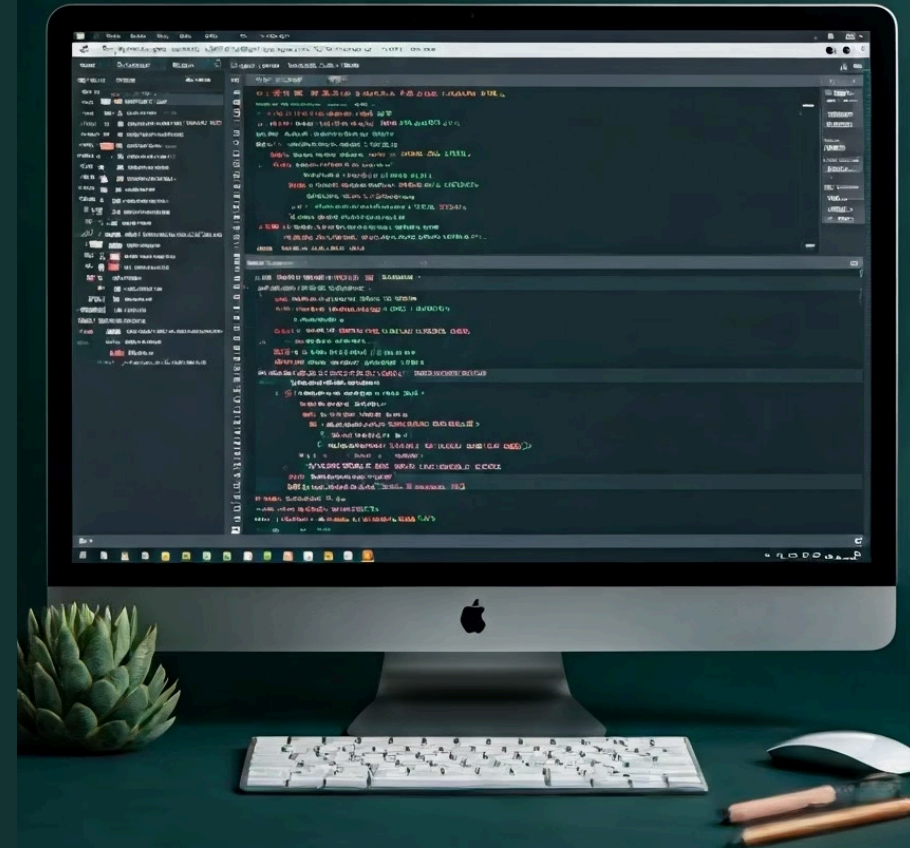
375-byte schema, 1.2KB React bundle, 6KB state machine

## Testing

Localstack validates generator output before deployment

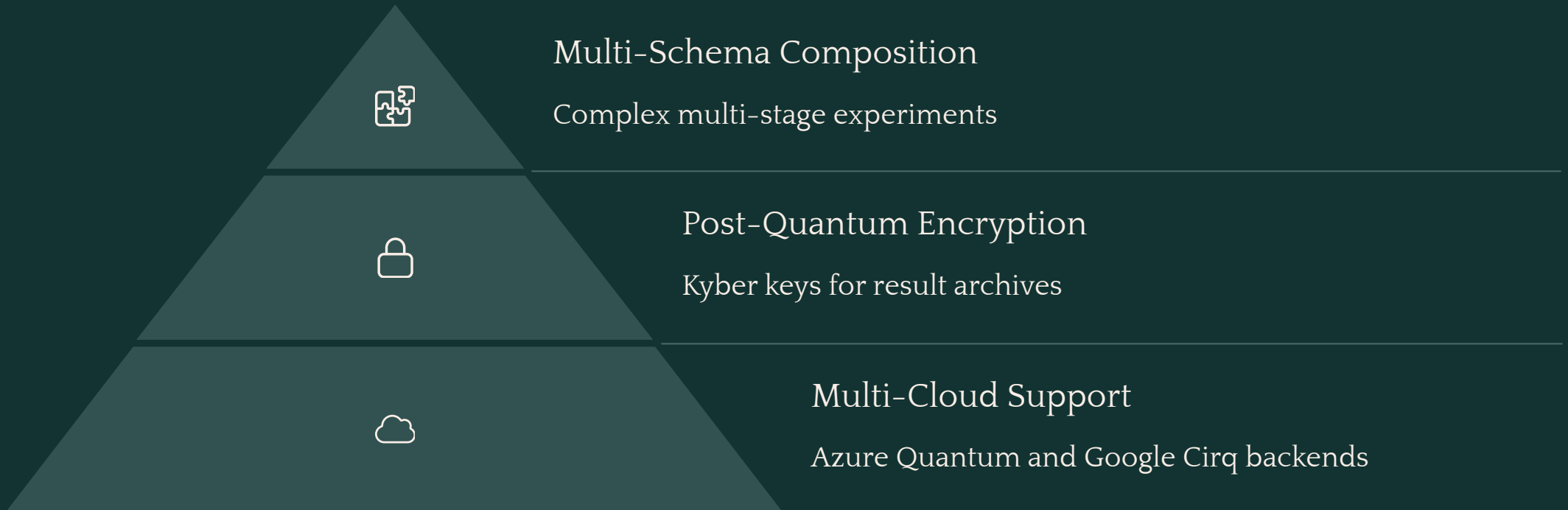
## Cost

Development within AWS free tier





# Future Directions



# Thank You