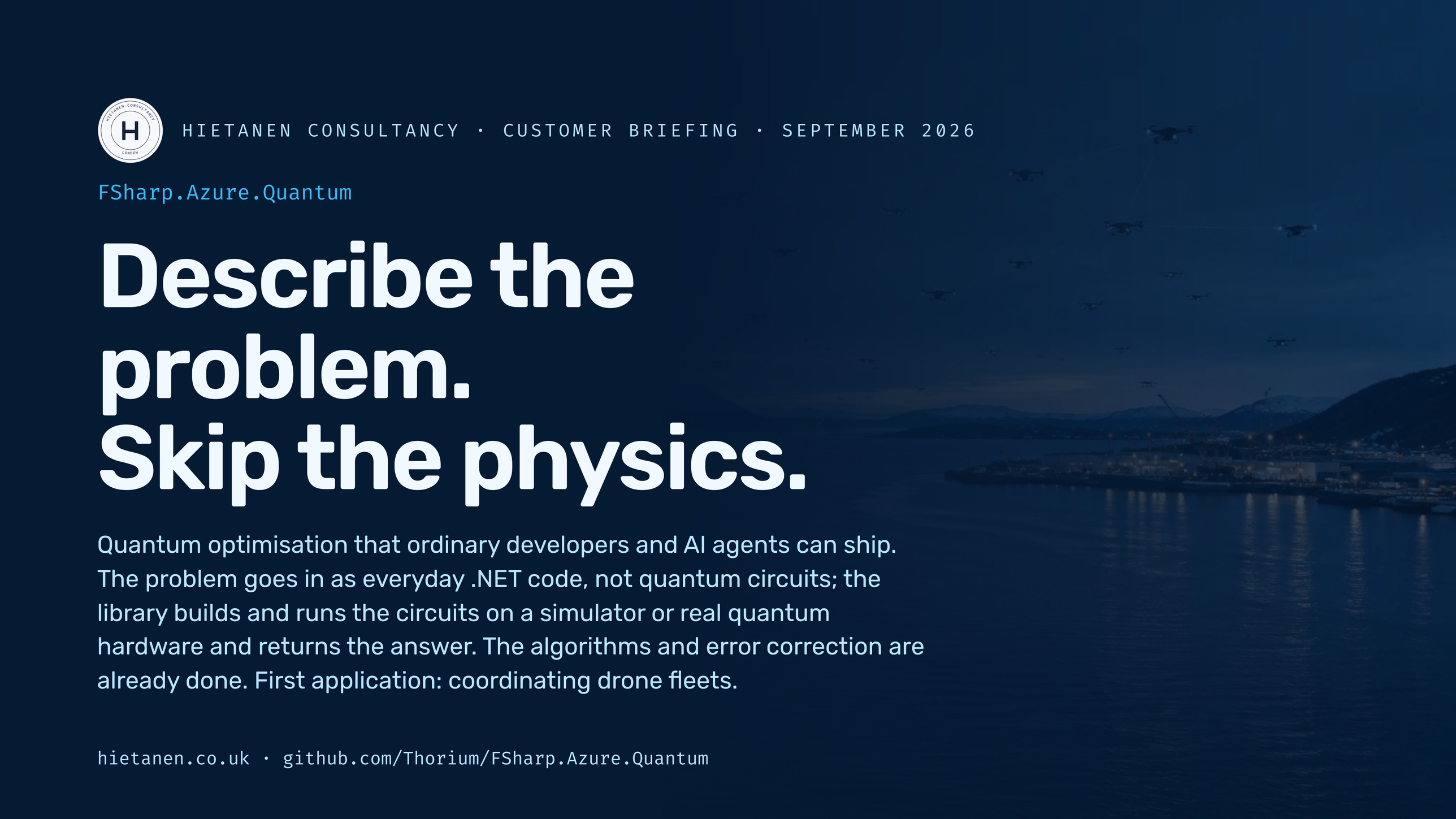


HIETANEN CONSULTANCY · CUSTOMER BRIEFING · SEPTEMBER 2026

FSharp.Azure.Quantum

Describe the problem. Skip the physics.

Quantum optimisation that ordinary developers and AI agents can ship. The problem goes in as everyday .NET code, not quantum circuits; the library builds and runs the circuits on a simulator or real quantum hardware and returns the answer. The algorithms and error correction are already done. First application: coordinating drone fleets.

hietanen.co.uk · github.com/Thorium/FSharp.Azure.Quantum

THE PROBLEM

Coordinating a fleet is a combinatorial problem that grows faster than the fleet.

Routing and assignment are NP-hard

Which drone flies where, in what order, within range and airspace limits. Add one failed unit or a new task and the whole plan has to be recomputed, in seconds.

Quantum tooling is written for physicists

Qiskit, Cirq and Q# expect gates, Hamiltonians and hand-encoded QUBOs. A fleet-software team has none of that vocabulary and no way to hire it.

Hardware is arriving without a developer path

Azure Quantum, D-Wave and Braket sell machine time today. Nothing lets an ordinary .NET or C# application describe a business problem and run it there.

THE PRODUCT

One NuGet package. Business problem in, quantum-computed answer out.

A .NET developer — or the agent working alongside them — describes routing, scheduling, assignment or portfolio problems in ordinary F# or C# and solves them on a simulator or on real hardware: the same code, every backend.

- On NuGet; F# computation expressions and a C# fluent API
- Seven optimisation builders, business builders for scheduling, fraud and churn, and quantum ML
- Backends: local simulator, Azure Quantum (IonQ, Pasqal, Quantinuum, Rigetti), D-Wave, AWS Braket, topological simulator

What it changes for a vendor: multi-aircraft planning from the developers it already has, in a twelve-week project, in place of an optimisation team it would have to hire.

FleetPlanning.fs

```
let plan = tsp {
    cities waypoints
    distances haversine
    minimize TotalFlightTime
}
let result =
    HybridSolver.solve plan backend
// splits the tour into range-safe
// sorties across the whole fleet
```

HOW IT WORKS

Three layers. Developers touch only the top one.

Layer 1

Builders — the business vocabulary

TSP, scheduling, assignment, knapsack, portfolio, network flow; fraud, churn, workforce. Written by .NET developers or by their agents.

Layer 2

Solvers — the algorithms

QAOA for optimisation, VQE and ADAPT-VQE for chemistry, Grover, QFT-based arithmetic and phase estimation.

Layer 3

Backends — one interface, every machine

Local simulator, gate-based QPUs through Azure Quantum and Braket, D-Wave annealers, a topological simulator. Circuits arrive as OpenQASM from Qiskit and leave as OpenQASM, QIR or a MAVLink mission.

What an integrator writes

One builder, in F# or C#. The backend is an argument: a laptop simulator today, an annealer or a QPU later, with no rewrite.

Quantum-first, honestly

A quantum solver never silently hands back a classical answer: if the hardware cannot solve it, the caller gets an error. HybridSolver routes to a classical baseline only when asked.

DRONE-FIRST · WHAT FLIES TODAY

Four runnable fleet examples, all four flown in ArduPilot's own simulator.

FleetPathPlanning

Fleet path planning

Waypoints become a tour, split into range-safe sorties across the fleet, altitudes checked against the regulatory ceiling.

SwarmTaskAllocation

Swarm task allocation

Tasks with dependencies and priorities scheduled across drones with capacity limits, minimising mission time.

SwarmChoreography

Swarm choreography

Formation transitions solved with QAOA, exported to Crazyflie indoors and as ArduPilot missions outdoors.

FireAirBridge

Wildfire air bridge

Water as flow on corridors between lake and fire, capped by fill slots, spacing and drop slots. The planner opens corridors as the wind turns.

Where the hardware is. Every example exports ArduPilot missions and a launcher that checks, flies and draws them. All four have flown in ArduPilot's simulator (SITL): the real autopilot firmware on a simulated airframe. Next comes one real aircraft.

WHY NOW

The rules already allow swarms. The software is what is missing.

One pilot, many aircraft, already lawful

An EASA authorisation records a pilot-to-aircraft ratio: 1:5, or three pilots to ten aircraft.

Deployment is the bottleneck

UK BVLOS flies today in rail, power-line and wind-farm corridors. One U-space airspace is live in the EU.

Quantum hardware is on the cloud

Azure Quantum, Braket and D-Wave sell machine time today.

AI agents need a typed API

Agents can write a builder expression; they cannot invent a QUBO encoding.

One timing argument: the rules, the hardware and the agents are here before the fleets are. That is the window to build the planning layer, and it sells today on classical hardware with the quantum path behind the same API.

COMPETITION

Physicist SDKs on one side, classical optimisers on the other. We sit between.

Capability	FSharp.Azure.Quantum	Quantum SDKs	Classical optimisers	Quantum startups
Business-level problem builders	Yes	No, gates and circuits	Yes, modelling languages	Per engagement
Usable from .NET and C#	Yes	Python, Q# only	Yes	Rarely
Runs on every quantum backend	Yes, one interface	Own vendor mostly	No	Own stack
Solves large fleets today	Hybrid path	No	Yes, the benchmark	Claimed
Evidence for a multi-aircraft authorisation	Constraint audit trail	No	No	Per engagement
Drone flight-controller export	Built in	No	No	No
Classical fallback disclosed	Always	Not applicable	Not applicable	Often not
Circuits move in and out	OpenQASM 1.0–3.0, QIR	Yes, OpenQASM	Not applicable	Own format

Quantum SDKs: Qiskit, Cirq, Q#, D-Wave Ocean. Classical optimisers: Gurobi, OR-Tools, CPLEX. Quantum startups: application-specific consultancies built on one vendor.

TEAM AND METHOD

One engineer, AI-assisted, shipped the library. The company grows as vendors sign.



Tuomas Hietanen — Founder and CEO

CTO and co-founder of Finular Group, nine years building regulated lending platforms in the UK and US. Lead engineer at Basware on purchase-to-pay at global scale. Maintainer of SQLProvider and LINQKit, published .NET author, and the author of FSharp.Azure.Quantum.

Who owns what	Today	Next hire
Sales	Founder	Head of sales
Customer integrations	Founder and network	Engineers
Flight testing	Founder, in simulator	Drone engineer
Quantum algorithms	Founder	Researcher

Identified, and named once agreements are in writing: a second senior team member and partners for data and compliance. The network is engineers we have worked with, .NET engineers from Ukraine among them, on contract. The first hire is an engineer, when two vendors ship the planner.

Key-person risk, said out loud. One person holds the code; delivery already draws on the network. The code is public domain and under test, and every algorithm is documented against its source paper.

WORKING WITH US

Bring us a fleet-planning problem.

01 · Project

Twelve weeks, fixed scope

Your fleet or your product, your constraints, a planner in production.

02 · Supported builds

For vendors who ship it

Signed builds, long-term support, an SLA and private builders.

03 · What stays yours

Aircraft and authority

Every mission is a file your pilot-in-command inspects before anything flies.

Ask for a live demo: four drones, one laptop, a quantum solver, twenty minutes.

Tuomas Hietanen • hello@hietanen.co.uk • hietanen.co.uk/projects/quantum/

Speaker notes • Quantum Fleet Planning Customer Briefing

01 • Describe the problem. Skip the physics.

Describe the problem, skip the physics. The quantum algorithms are implemented and tested inside the library, so a customer's developers describe the problem and never touch a gate. Drones are where the combinatorics bite first.

02 • Coordinating a fleet is a combinatorial problem that grows faster than the fleet.

Coordinating a fleet is a combinatorial problem that grows faster than the fleet. The buyer is the vendor whose software the fleet already runs on, the pain is re-planning under constraints, and that gap exists today, whenever quantum advantage arrives.

03 • One NuGet package. Business problem in, quantum-computed answer out.

One NuGet package: a fleet developer describes the problem in a builder they can read in ten seconds and gets a quantum-computed plan back. No gates, no Hamiltonians.

DETAIL

- Outcome: multi-aircraft planning from the vendor's own developers in a twelve-week, fixed-scope project.
- The benchmark publishes the comparison at 50 to 200 aircraft against a classical baseline.
- Also in the package: OpenQASM 1.0 to 3.0 in and out, QIR emission, MAVLink missions, error mitigation and qubit routing.

04 • Three layers. Developers touch only the top one.

Three layers, and developers touch only the top one. The plan is written once, in the language the vendor's developers already use; moving it from a simulator to an annealer or a QPU changes one argument.

DETAIL

- No silent classical fallback: the rule is in the README and enforced in the tests.

ANSWERS

- Is this real quantum execution? Yes: no classical fallback hides behind the API, and the tests enforce it.

05 • Four runnable fleet examples, all four flown in ArduPilot's own simulator.

Four runnable fleet examples, and all four have flown in ArduPilot's own simulator: the real autopilot firmware on a simulated airframe. One real aircraft is the next step.

DETAIL

- FleetPathPlanning, SwarmTaskAllocation, SwarmChoreography and FireAirBridge export ArduPilot missions; the launcher flies them and draws the flight against the plan.
- The simulator proves parameter names, mission upload, arming and timing, and flying it found and fixed real defects. Airframe, wind, GPS error and radio loss need the real aircraft.
- The air bridge plans corridors, with drones as units of flow on them; the classical planner is the oracle the quantum one has to beat. Its customers are aerial-services contractors for fire and rescue.
- Choreography runs four drones on sixteen qubits, and a hundred drones on the same sixteen qubits, four at a time.

ANSWERS

- Payload? A constraint the planner models: camera overlap, spray-tank mass against endurance, refill stops.

06 • The rules already allow swarms. The software is what is missing.

The rules already allow swarms; deployment is what is missing, and the software has to be ready first. That is the window this company is in.

DETAIL

- Permission: an EASA operational authorisation records a remote-pilot-to-aircraft ratio, capped by the automation the flight manual documents.
- Deployment: UK BVLOS flies in atypical air environments under a handful of authorisations; since January 2023 the EU has designated one U-space airspace (San Salvo, Italy).
- Hardware: Azure Quantum sells IonQ, Pasqal, Quantinuum and Rigetti time; D-Wave annealers handle thousands of variables; the library already simulates Microsoft's Majorana machine.
- Agents: a typed, high-level library is how quantum reaches the code agents write.

ANSWERS

- Why sell to vendors now? The developer layer arrives before the fleets, so today's buyer is the vendor shipping the fleet software.
- When does quantum win? The benchmark will publish the problem size at which the hybrid planner beats a classical baseline.

07 • Physicist SDKs on one side, classical optimisers on the other. We sit between.

Physicist SDKs on one side, classical optimisers on the other; we sit between. For a fifty-unit fleet today Gurobi is faster, and customers get that through the hybrid path with no code change, and the quantum path when it wins.

08 • One engineer, AI-assisted, shipped the library. The company grows as vendors sign.

One engineer directing AI coding agents shipped the library, with a network of engineers for projects. The company grows as vendors sign, and a typed API is how quantum reaches the code that agents write.

DETAIL

- Today the founder owns sales, customer integrations, flight testing in ArduPilot's simulator, and the algorithms; a test suite in the thousands, the whole source public.
- The network joins projects on contract, including .NET engineers from Ukraine; names on request.
- Next: an integrations engineer when two vendors ship the planner, then a head of sales, a quantum-algorithms researcher and a drone engineer for flight testing.
- Civil work only: defence or dual-use enquiries go through the export-control review, and counsel checks what may be shared.

ANSWERS

- Who else? A second senior team member and partners for data and compliance are identified; names go on the slide once agreements are in writing.

09 • Bring us a fleet-planning problem.

Bring us a fleet-planning problem: a twelve-week, fixed-scope project, with the boundary stated plainly. We are a planning tool, not a flight-control system.

DETAIL

- Project: a vendor's or operator's fleet or product, their constraints, a planner in production; the hybrid solver runs today on classical hardware, with quantum backends behind the same API.
- Supported builds: signed builds, long-term support, an SLA and private builders for vendors shipping the planner.
- What stays yours: the aircraft, the authorisation, the pilots and the liability. Payload is a constraint we model, not hardware we ship.

ANSWERS

- A hosted planning API? It will be built with the first vendors who want to route their customers' planning through it.
- Who does the work? The founder with a network of engineers, .NET engineers from Ukraine among them, per project; civil work only.
- What does the library cost? Nothing: it is public domain on NuGet. What is bought is the project, the supported build and somebody answerable.