ALICE & BOB

Bringing the Cat home

Installing on-prem superconducting cat qubits

QUEST-IS // TQCI

RÉMI DE LA VIEUVILLE

DECEMBER 4TH, 2025

01

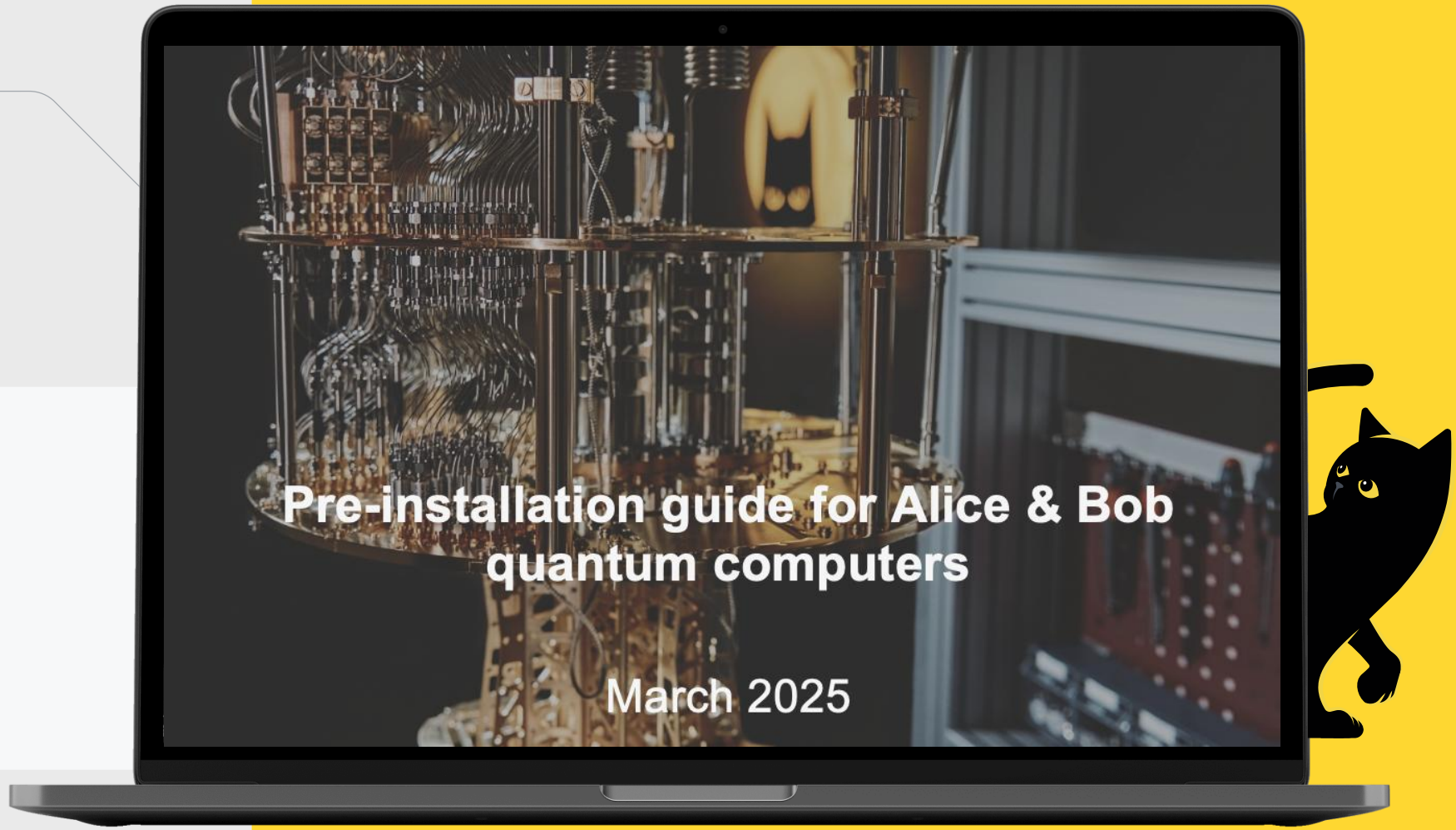
//SECTION 01

HARDWARE INSTALLATION

Pre-installation guide

- System description
- Room requirements
- Utilities
- Maintenance

Also a Site Evaluation Checklist



System description

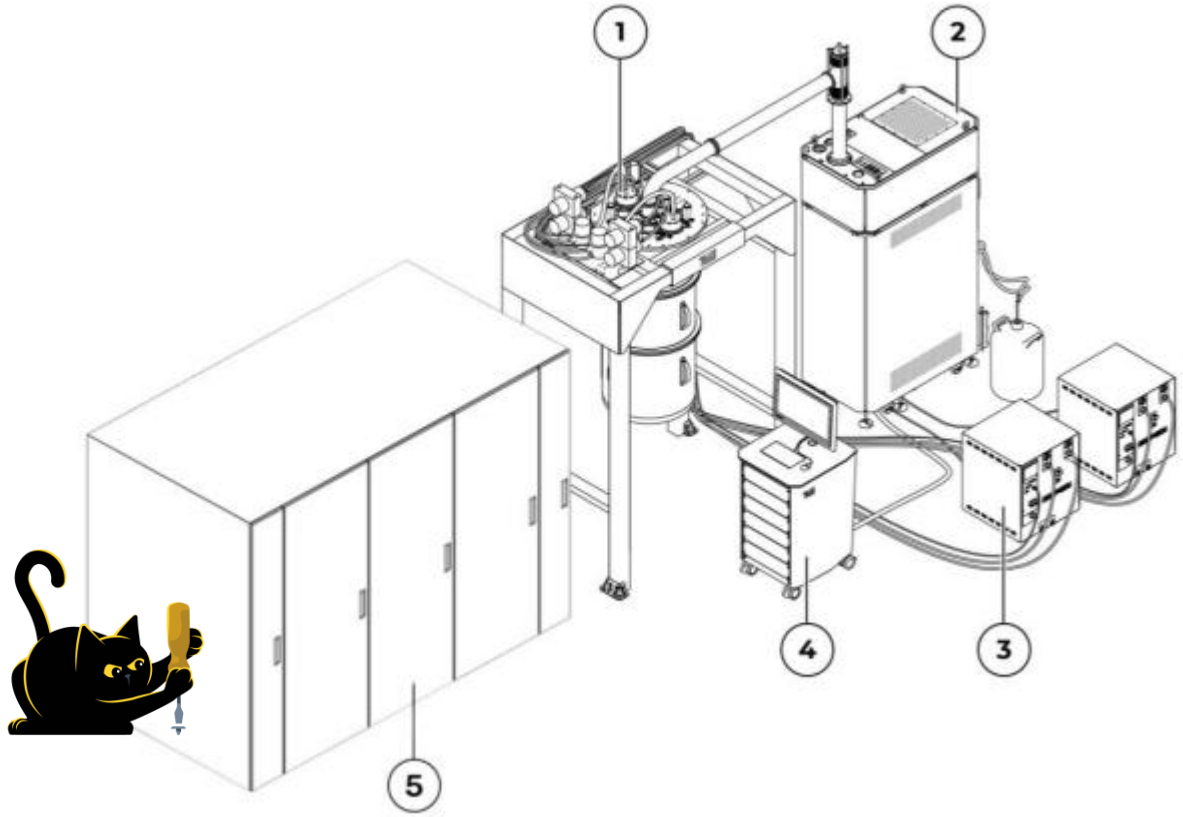
1) Cryostat

2) Gas Handling System

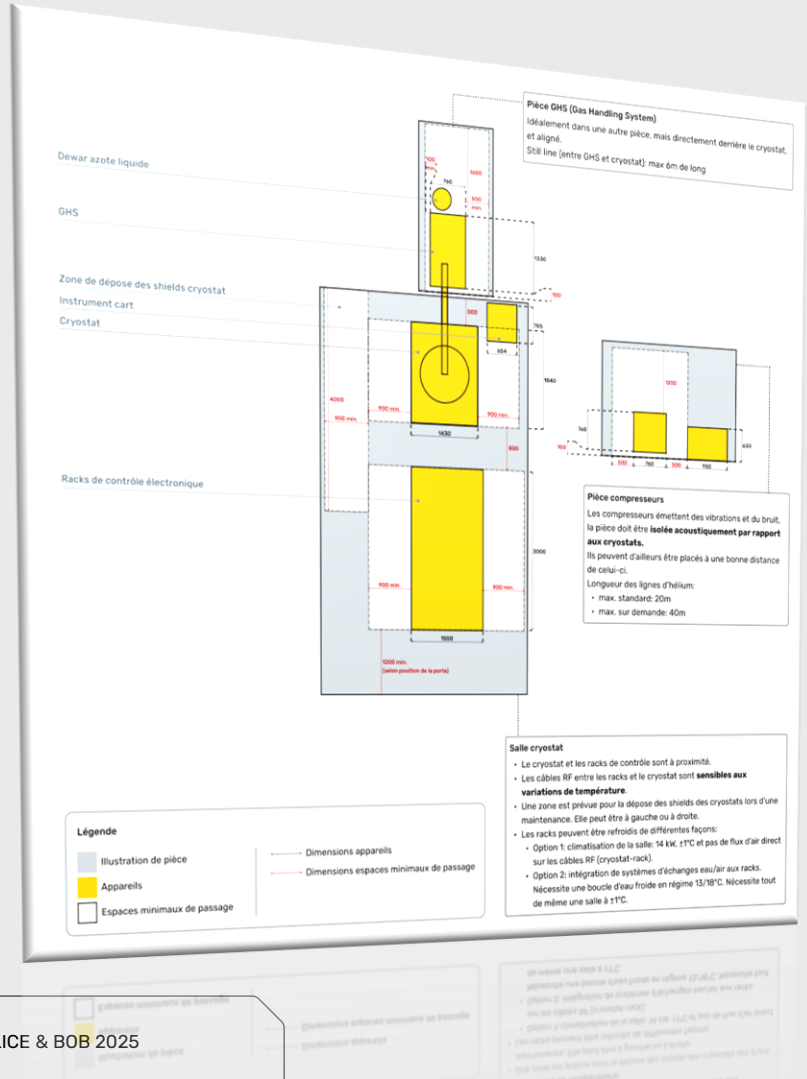
3) Pulse-tubes compressors for Helium

4) Instrument cart

5) Control Electronics rack



Room requirements



1 Dimensions — Including height (4m)

Weight — Cryostat 1200 kg

2 Temperature — 18-25°C +/- 1°

Humidity — 45 - 65%

Dirt

3 Vibrations — VC-D

Noise — < 50 dB (empty)

4 CEM

NF C 15-100
CEI 60364-4- 44
LED lighting

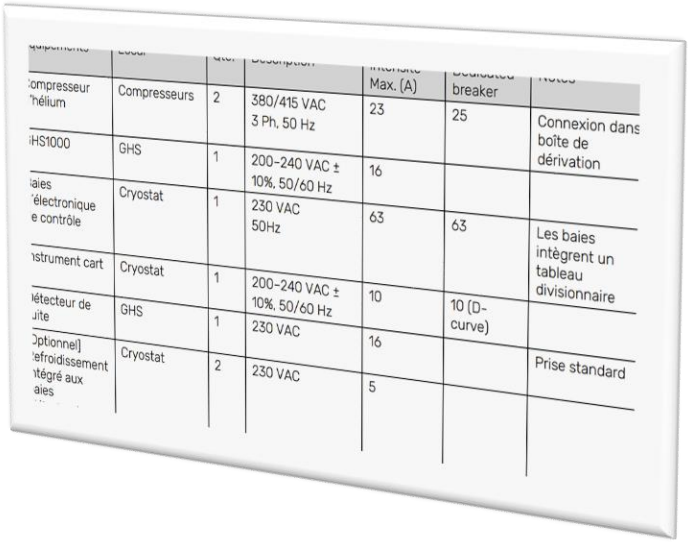
5 Safety

Fire detection

Anoxy detection

Utilities

Electricity



				Max. (A)	breaker	
ompresseur hélium	Compresseurs	2	380/415 VAC 3 Ph, 50 Hz	23	25	Connexion dans boîte de dérivation
iHS1000	GHS	1	200-240 VAC ± 10%, 50/60 Hz	16		
aies électronique e contrôle	Cryostat	1	230 VAC 50Hz	63	63	Les baies intègrent un tableau divisionnaire
strument cart	Cryostat	1	200-240 VAC ± 10%, 50/60 Hz	10	10 (D-curve)	
létecteur de uite	GHS	1	230 VAC	16		
Optionnel] efroidissement tégré aux aies	Cryostat	2	230 VAC	5		Prise standard

Liquid Nitrogen

20L / week

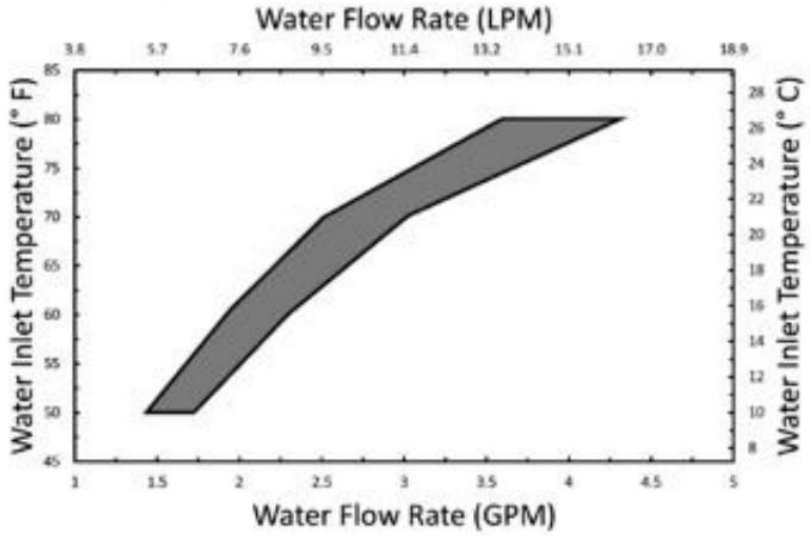
Compressed Air

Water

Necessary for:

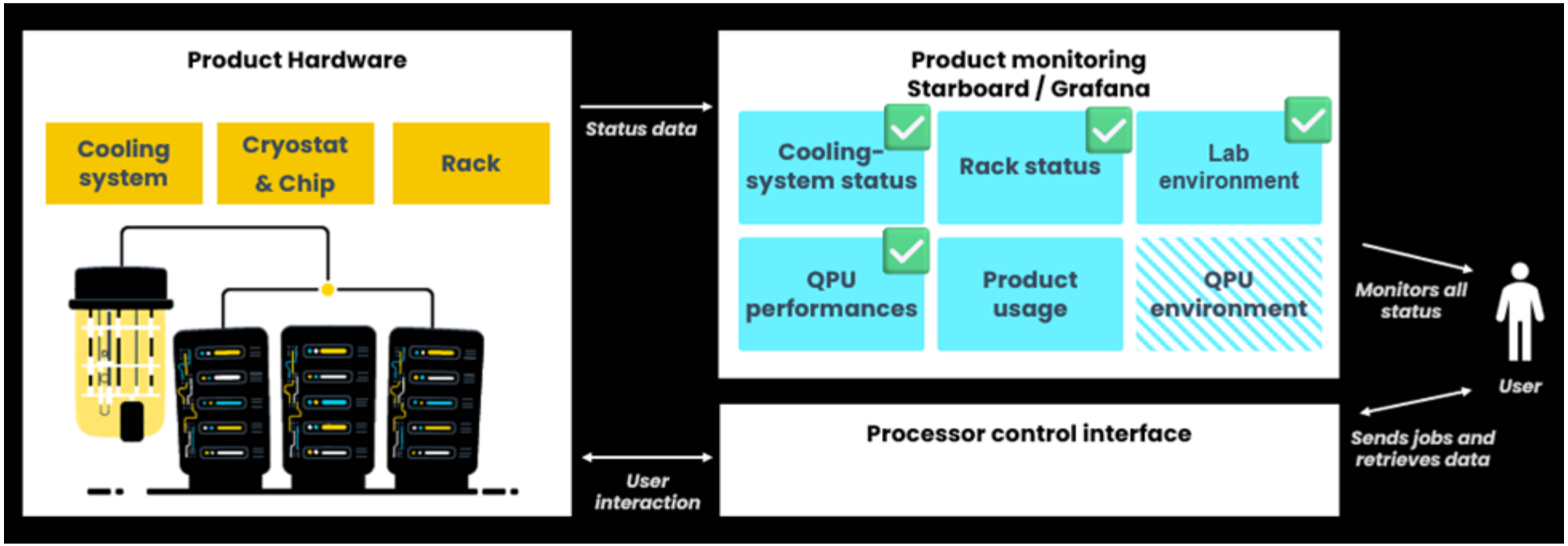
- Helium compressors for Pulse tubes
- Turbomolecular pumps

Cooling water temperature and flow rate must fall within the shaded area of the chart shown in Figure 47.



Maintenance

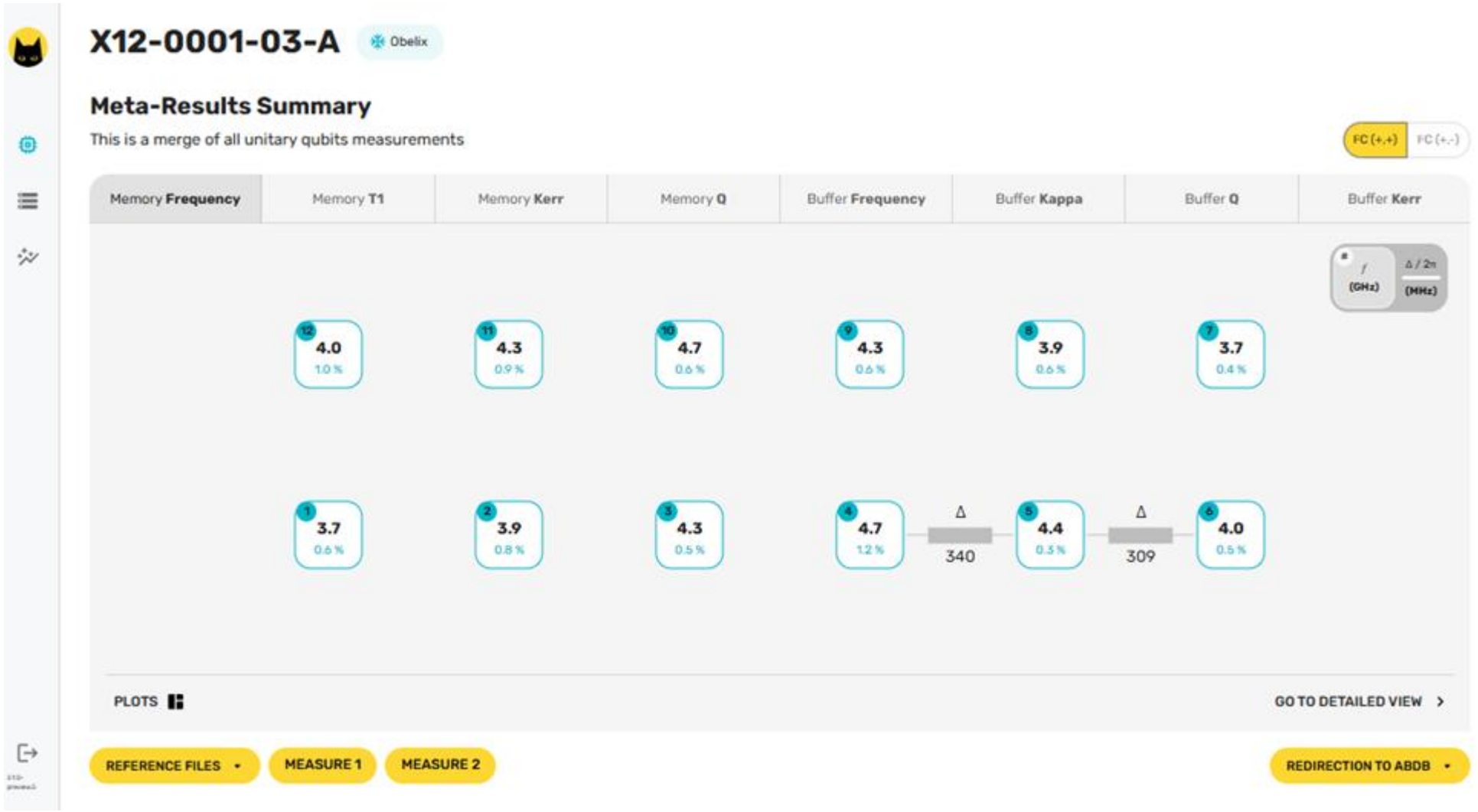
Monitoring



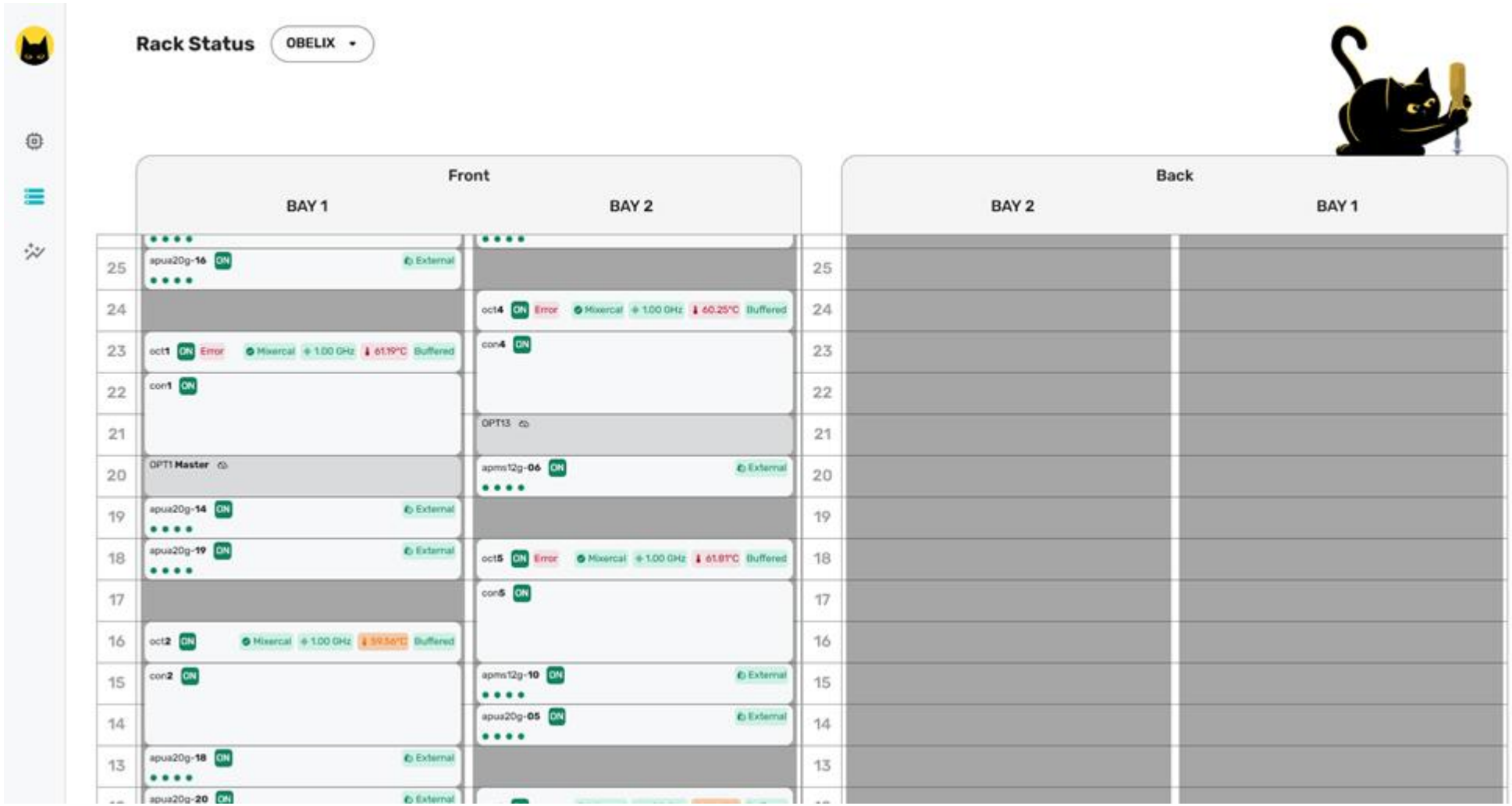
Liquid Nitrogen

Cryostat

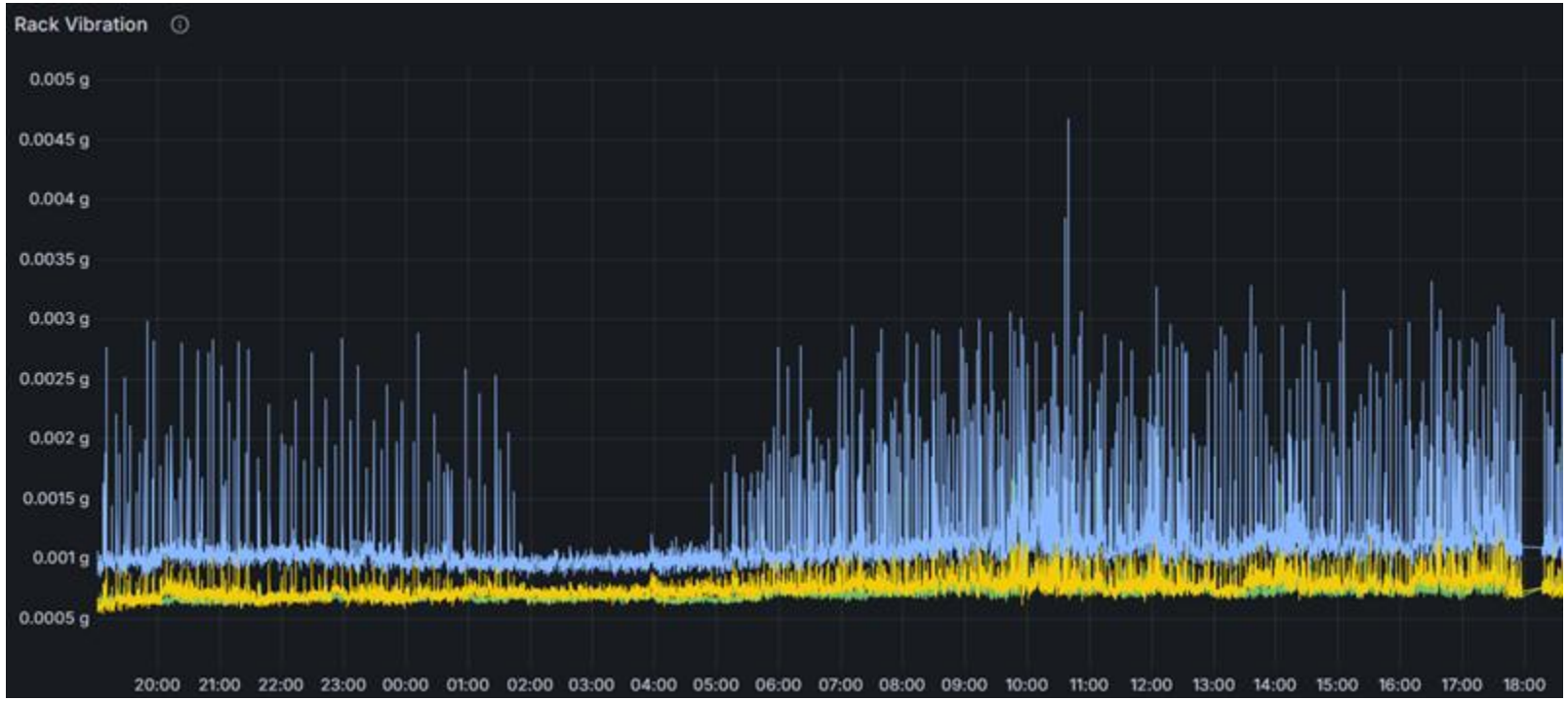
[Monitoring] QPU



[Monitoring] Rack



[Monitoring] Room



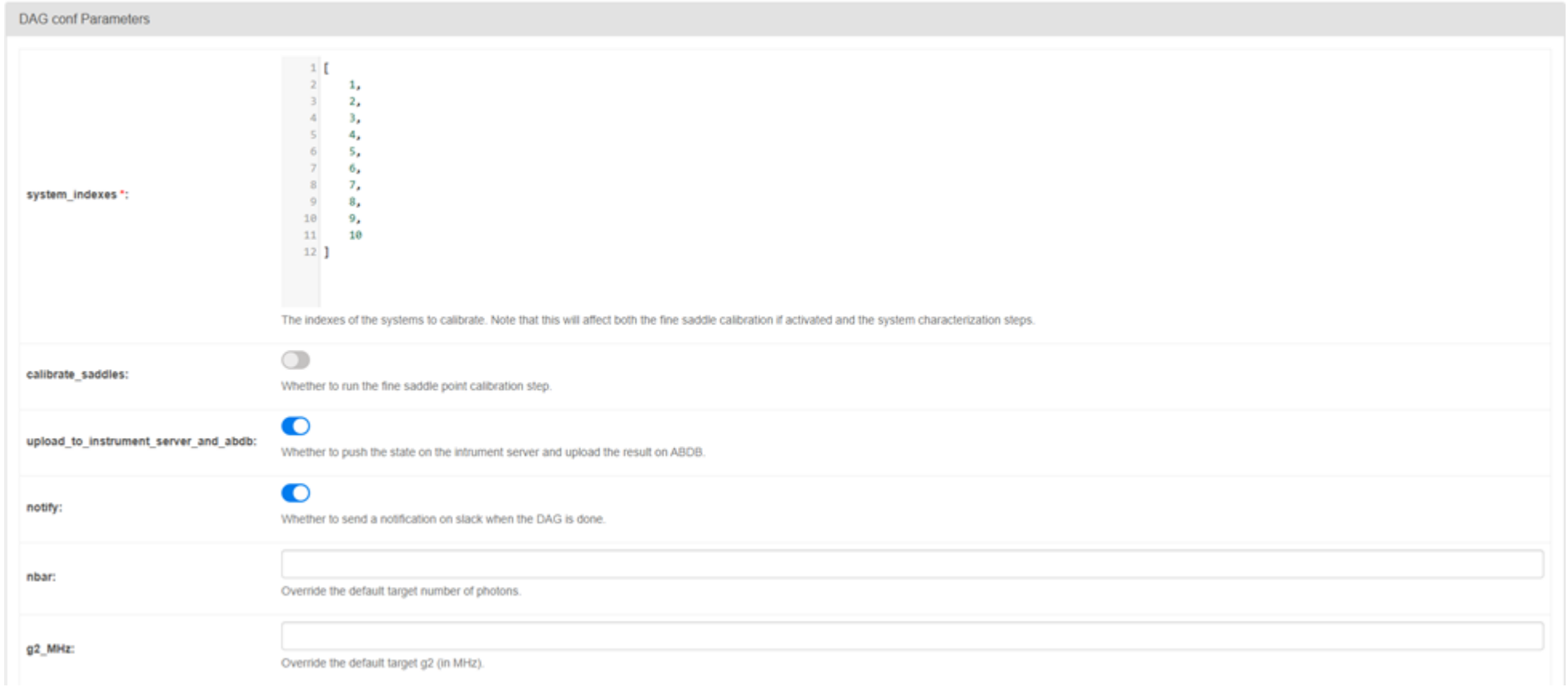
[Monitoring] Usage

02

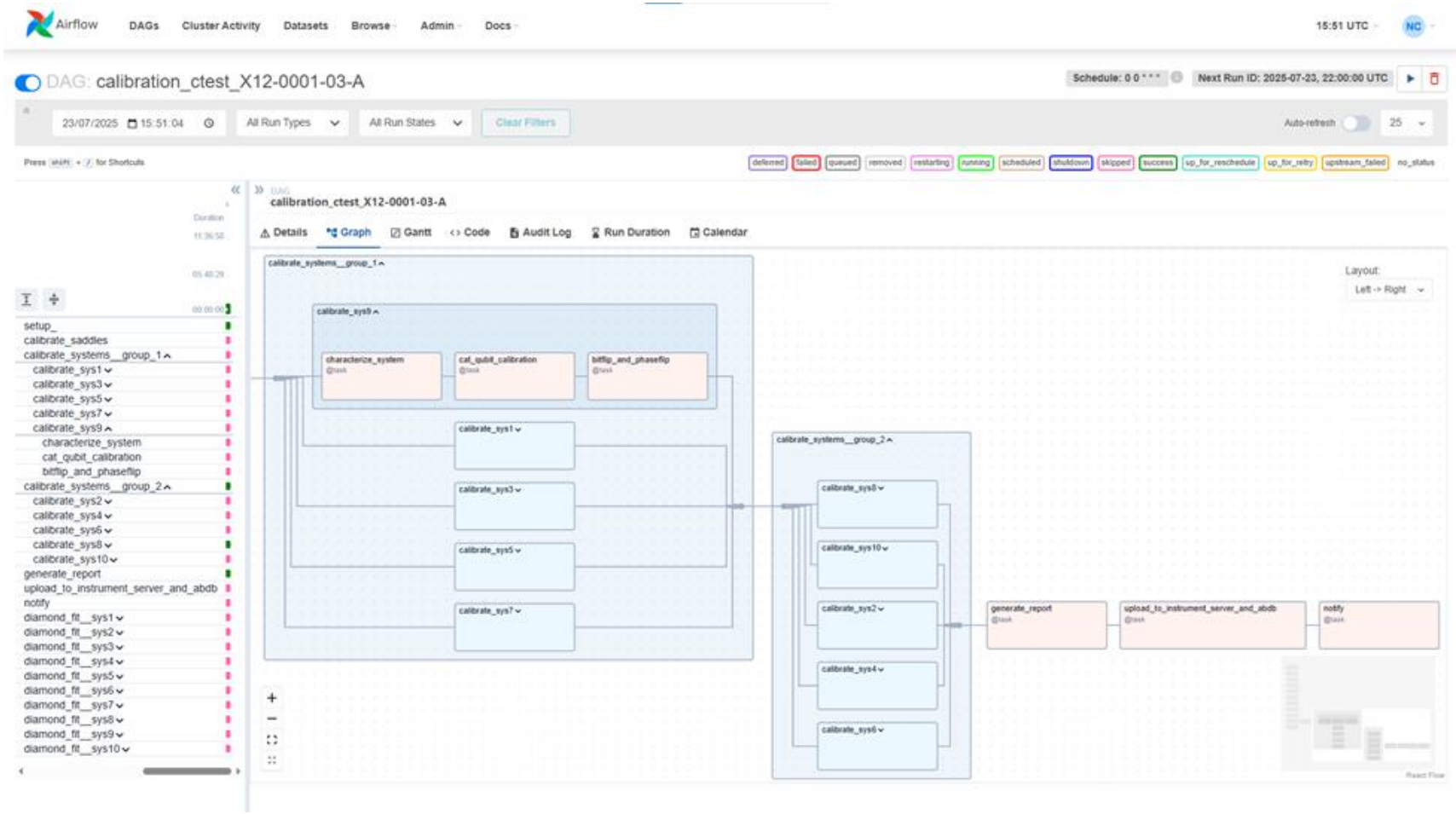
//SECTION 02

FIRMWARE CALIBRATION

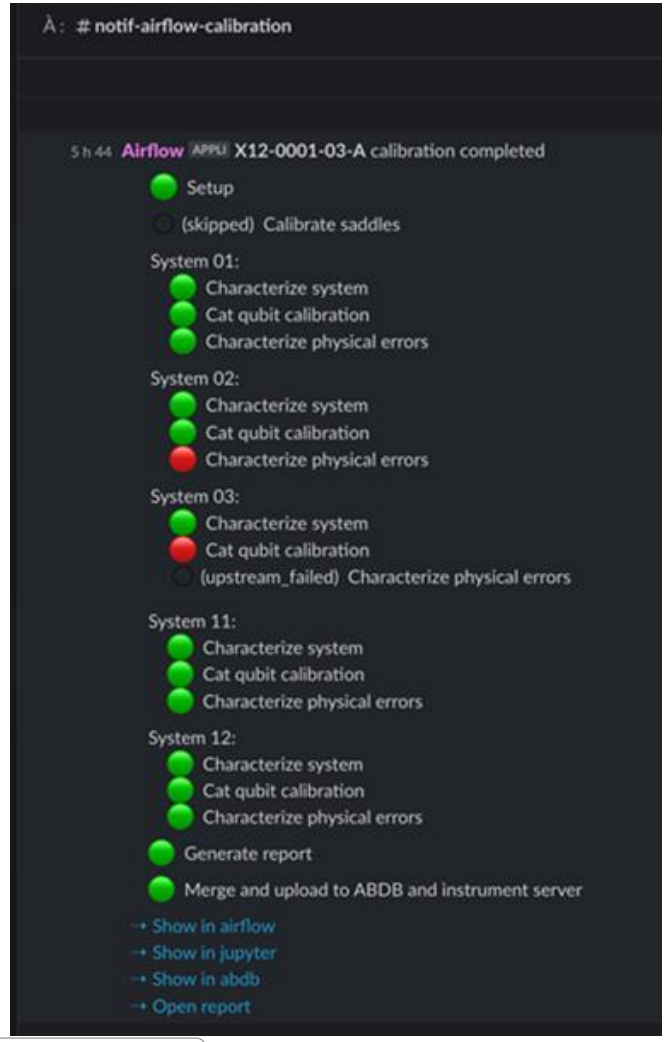
[Calibration] Configuration panel

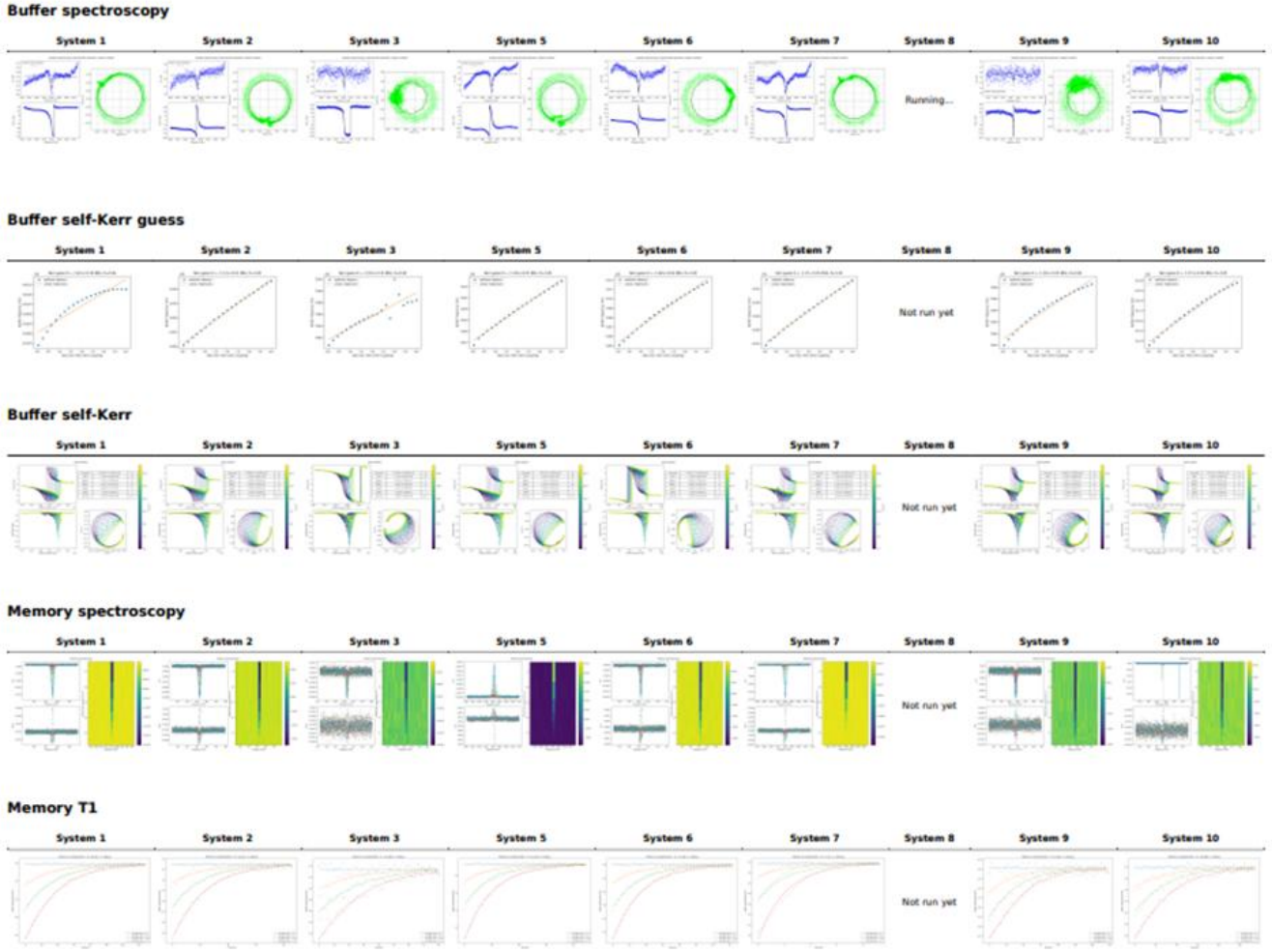


[Calibration] Workflow

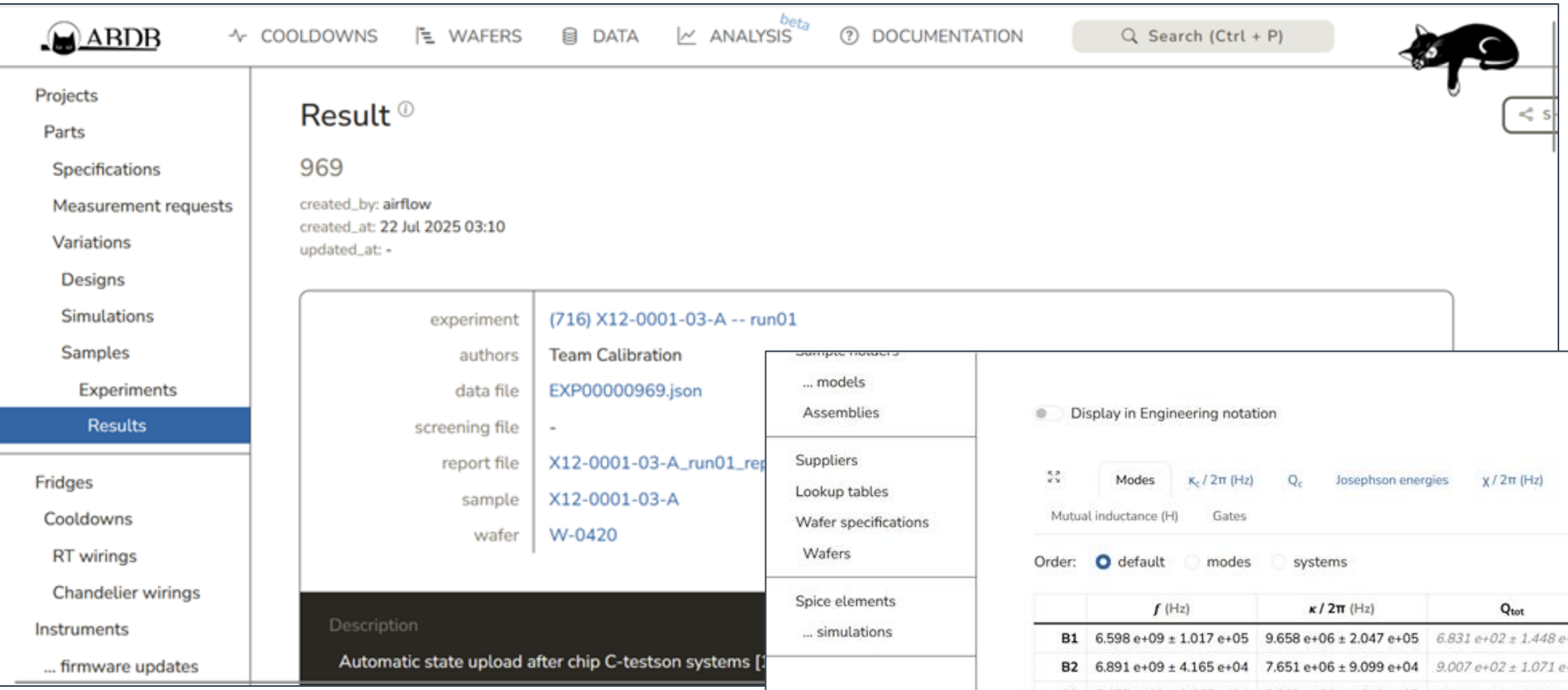


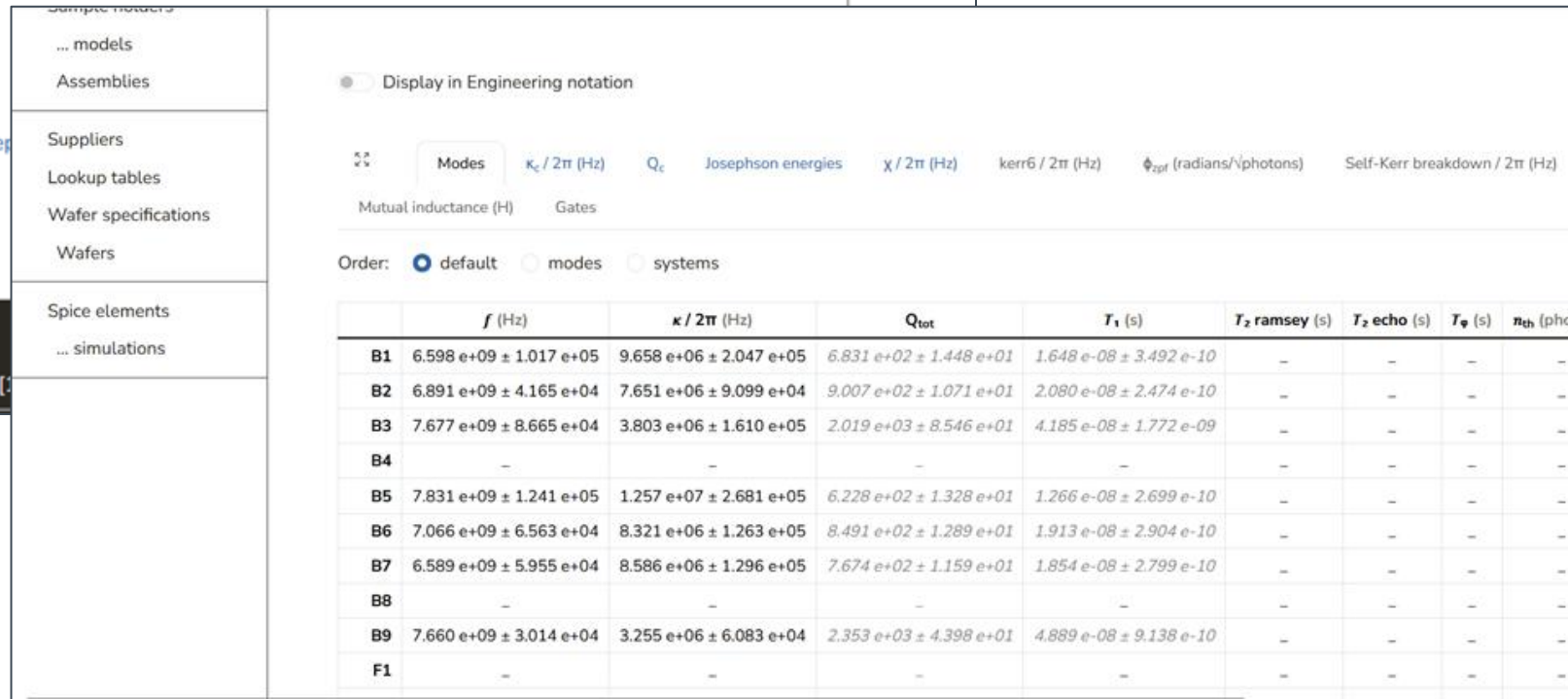
[Calibration] Summary





[Calibration] Full reports



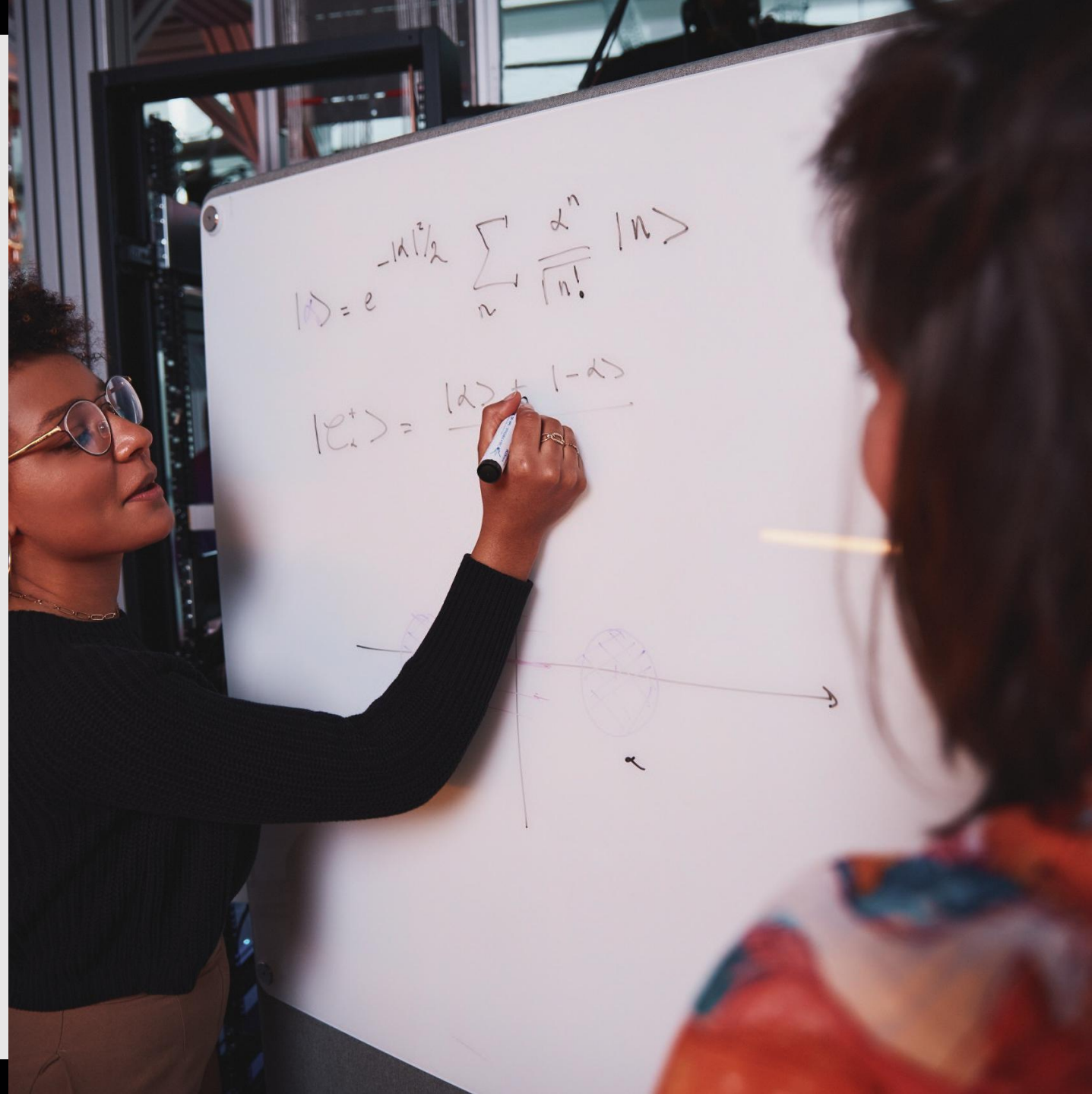


	f (Hz)	κ / 2π (Hz)	Q_{tot}	T_1 (s)	T_2 ramsey (s)	T_2 echo (s)	T_φ (s)	n_{th} (pho
B1	6.598 e+09 ± 1.017 e+05	9.658 e+06 ± 2.047 e+05	6.831 e+02 ± 1.448 e+01	1.648 e-08 ± 3.492 e-10	–	–	–	–
B2	6.891 e+09 ± 4.165 e+04	7.651 e+06 ± 9.099 e+04	9.007 e+02 ± 1.071 e+01	2.080 e-08 ± 2.474 e-10	–	–	–	–
B3	7.677 e+09 ± 8.665 e+04	3.803 e+06 ± 1.610 e+05	2.019 e+03 ± 8.546 e+01	4.185 e-08 ± 1.772 e-09	–	–	–	–
B4	–	–	–	–	–	–	–	–
B5	7.831 e+09 ± 1.241 e+05	1.257 e+07 ± 2.681 e+05	6.228 e+02 ± 1.328 e+01	1.266 e-08 ± 2.699 e-10	–	–	–	–
B6	7.066 e+09 ± 6.563 e+04	8.321 e+06 ± 1.263 e+05	8.491 e+02 ± 1.289 e+01	1.913 e-08 ± 2.904 e-10	–	–	–	–
B7	6.589 e+09 ± 5.955 e+04	8.586 e+06 ± 1.296 e+05	7.674 e+02 ± 1.159 e+01	1.854 e-08 ± 2.799 e-10	–	–	–	–
B8	–	–	–	–	–	–	–	–
B9	7.660 e+09 ± 3.014 e+04	3.255 e+06 ± 6.083 e+04	2.353 e+03 ± 4.398 e+01	4.889 e-08 ± 9.138 e-10	–	–	–	–
F1	–	–	–	–	–	–	–	–

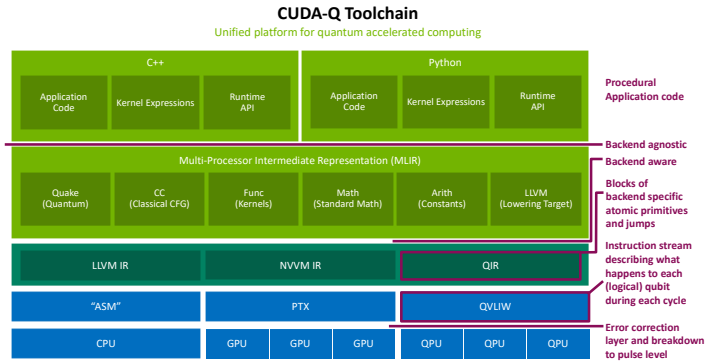
03

//SECTION 03

FIRMWARE FOR FTQC



Many stacks,
But how is a quantum program executed?



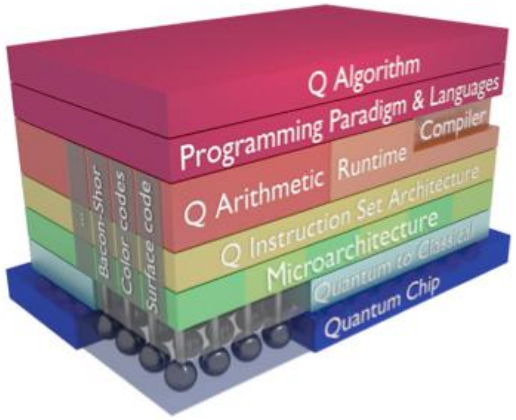
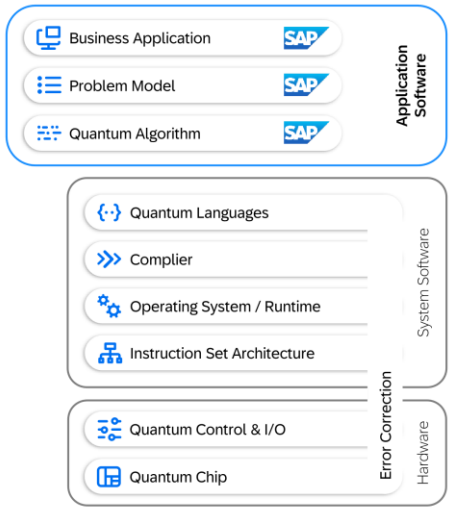
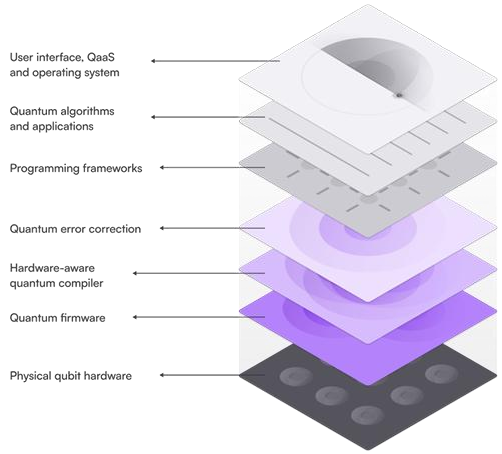
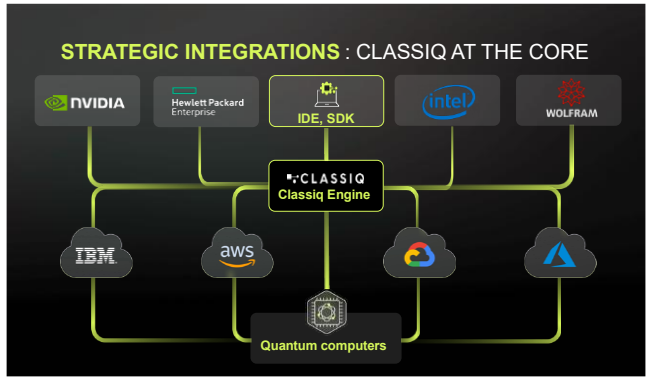


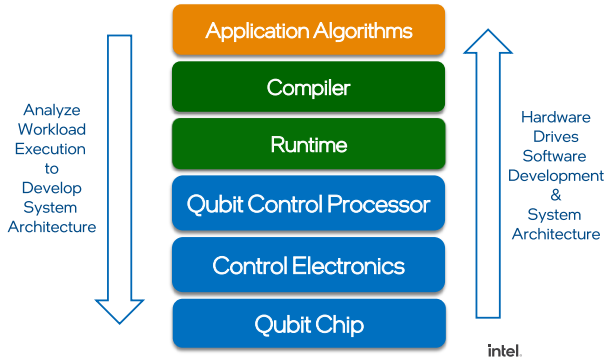
Figure 1: Overview of the quantum computer system stack from [2].

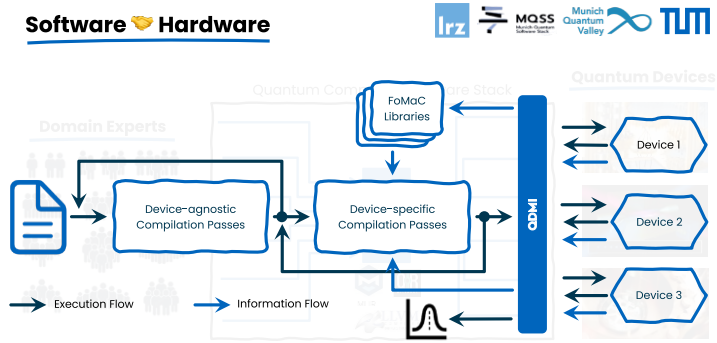


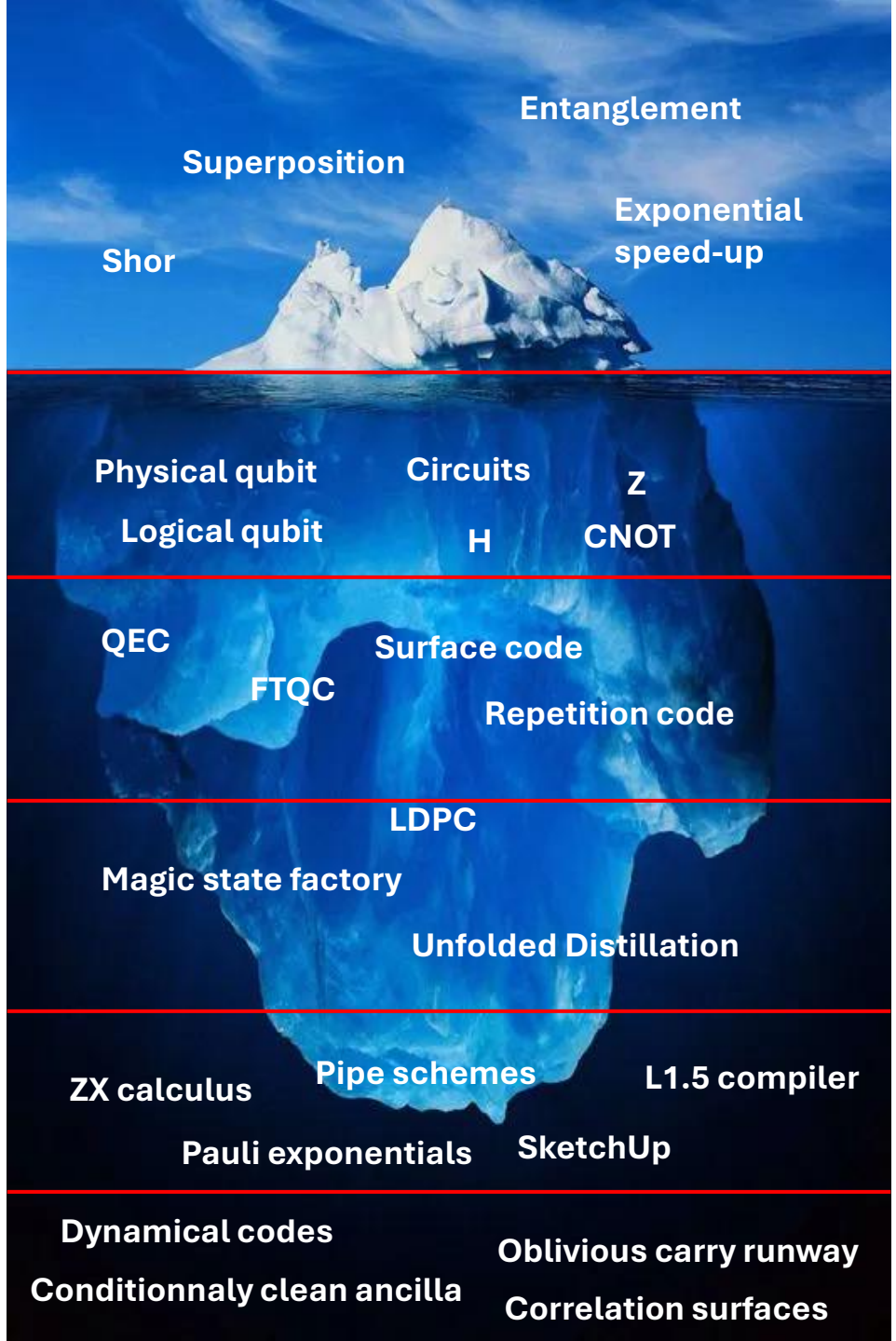
Quantum Computing Stack



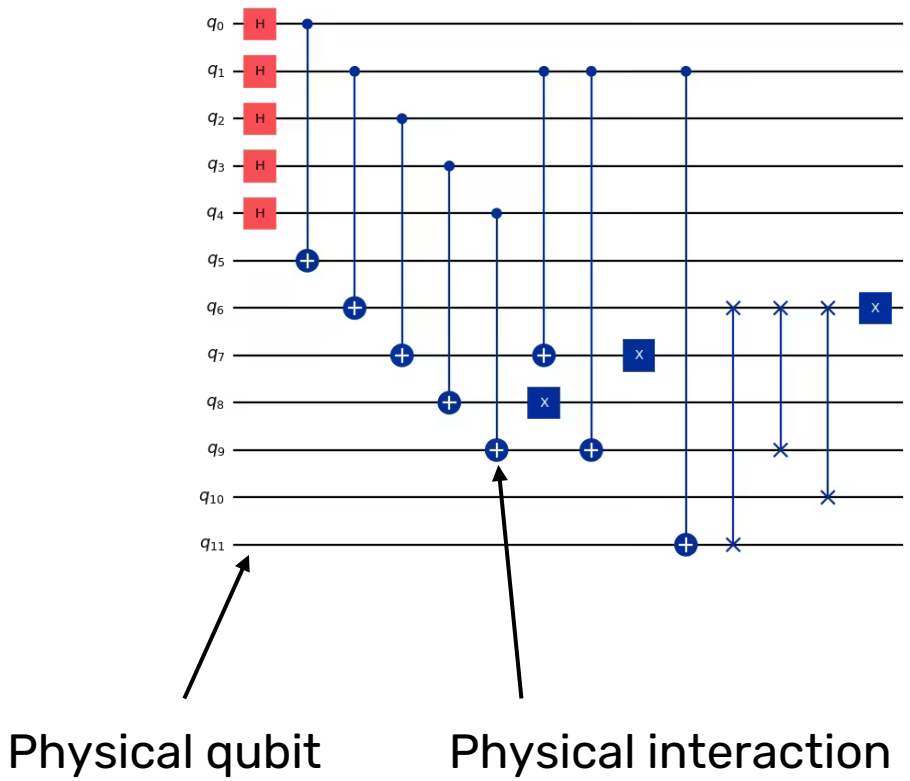







Entanglement
Superposition
Exponential speed-up
Shor
Physical qubit
Logical qubit
Circuits
H
Z
CNOT
QEC
FTQC
Surface code
Repetition code
LDPC
Magic state factory
Unfolded Distillation
ZX calculus
Pipe schemes
L1.5 compiler
Pauli exponentials
SketchUp
Dynamical codes
Conditionnaly clean ancilla
Oblivious carry runway
Correlation surfaces

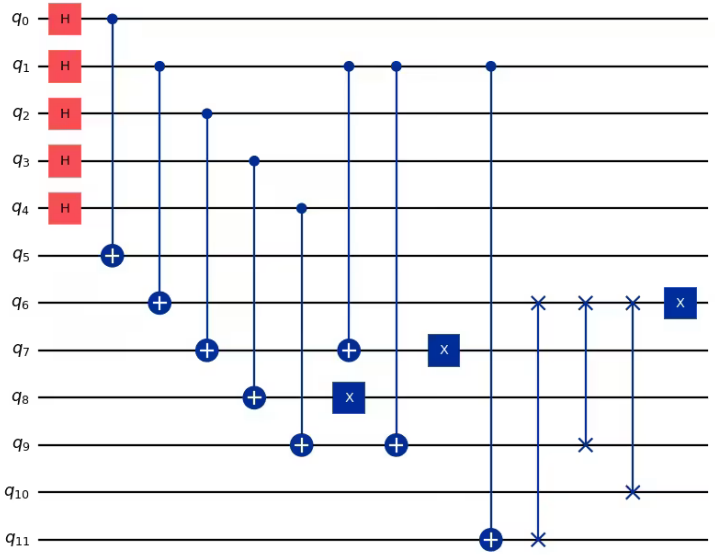
Quantum computing on a noisy device (*not* fault-tolerant)



- Processors 100-1000 qubits
- Continuous gate sets: exact compilation
- Usually no live feedback

- Qubit / gate errors $10^{-2} - 10^{-4}$: limited depth
- No known useful applications at this depth

Simulating a “perfect” device with noisy qubits



Each algorithmic qubit is encoded in a group of physical qubits

Error of the physical qubit $\sim 10^{-2} - 10^{-4}$

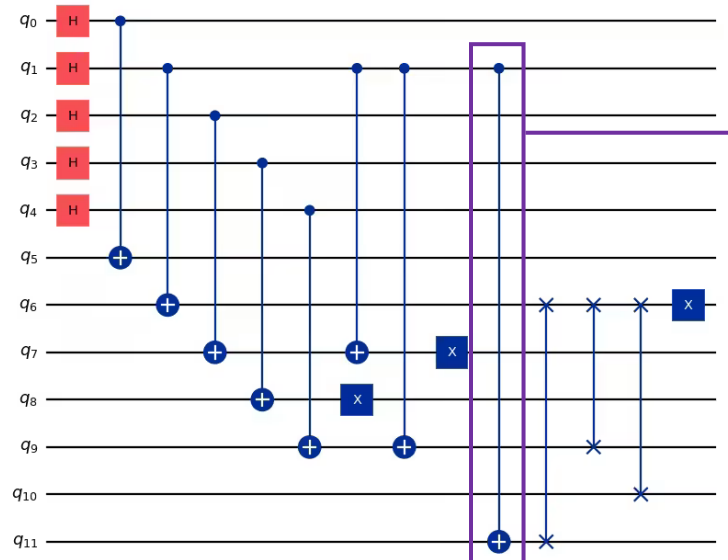
$$p_L \propto \left(\frac{p}{p_{th}}\right)^{d/2}$$

$\propto$ physical qubits per algorithmic qubit

Error of the algorithmic qubit $\sim 10^{-8} - 10^{-15}$

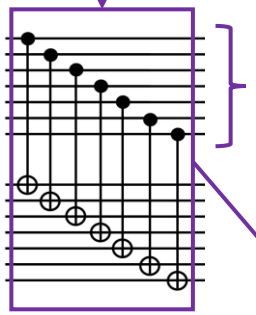
Minimal quality requirement on the physical qubit $p \leq p_{th}$

Simulating a “perfect” device with noisy qubits



A logical gate

Each algorithmic qubit is encoded in a group of physical qubits

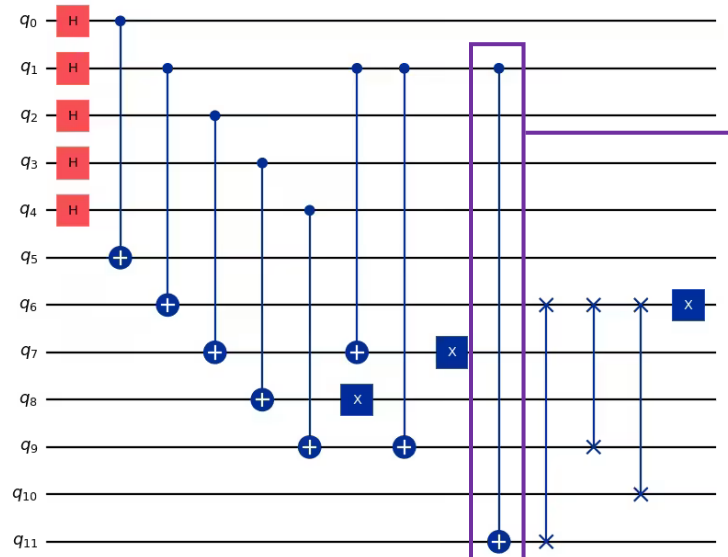


A single logical qubit

A logical gate or “gadget”: a physical circuit that manipulates the algorithmic information

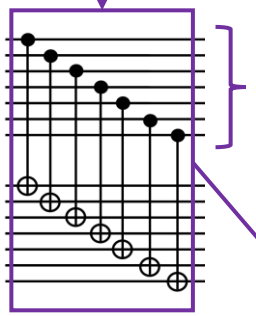
If the “gadgets” are *fault-tolerant*, the logical information is manipulated with fidelity p_L even though each (physical gate) in the circuit has fidelity p !

Simulating a “perfect” device with noisy qubits



A logical gate

Each algorithmic qubit is encoded in a group of physical qubits



A single logical qubit

A logical gate or “gadget”: a physical circuit that manipulates the algorithmic information

- Arbitrary quality = infinite depth! $p_L \propto \left(\frac{p}{p_{th}}\right)^{d/2}$
- Processors are $10\times - 1000\times$ bigger
- Processors are $10\times - 100\times$ slower
- Decoding problem to solve (live)
- Discrete gate sets: no exact compilation

A&B FTQC abstraction layers

L5: application
Specification of the problem, as expressed in the fields it comes from

Example: RSA number to factorize, electronic Hamiltonian coefficients

L4: high level algorithm
Manipulating interpreted data types with high-level operations

Example: "Shor in 3 lines"

L3: register operation
Low level algorithm, manipulating simple data types with elementary operations

Example: Addition on a register, QFT

L2: logical operation
Operations on individual logical qubits, better when each correspond to a fault-tolerant gadget

Example: logical circuit, ZX-calculus

L1.5: time-space layered logical operation
Time-space positioned logical qubits, with code parameters

Example: pipe schemes, ZX with space-time coordinates

L1: physical circuit
Full physical circuit, including synchronization

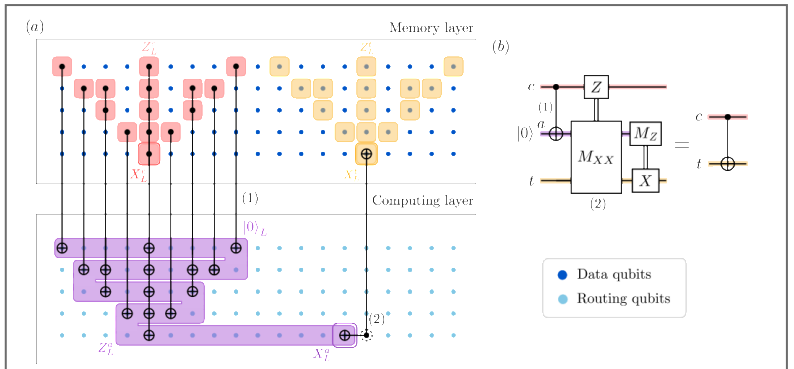
Example: QASM, Stim

L0: bosonic mode levels

L4: "Shor in 3 lines"

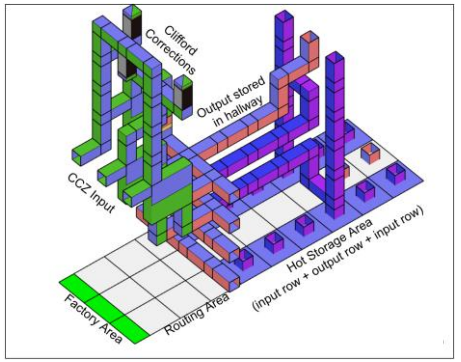
```
def find_order(a, N):
    qg = QuantumModulus(N)
    qg[:] = 1
    qpe_res = QuantumFloat(2*qg.size+1)
    h(qpe_res)
    qg *= pow(a, qpe_res, N)
    QFT(qpe_res, inv=True)

    mes_res = qpe_res.get_measurement()

    return extract_order(mes_res, a, N)
```

L2: logical CNOT



Ruiz, Guillaud, Leverrier *et al.*, Nat Commun (2025)

L1.5: Adder Compute Step



SLURM integration

SPANK plugins which give users access to quantum resources through a "Quantum Resource Management Interface" (QRMI)

WITH HARTREE CENTRE (UK)