Software collaboration ETH + QuTech

19-5-2016

Thursday
Goal: Understand requirements, capabilities and potential for collaboration.
8:30 - 9:00arrival at QuTech + Coffee
9:00 - 10:00 Presentations by Johannes and Adriaan
    Current/desired framework Johannes Heinsoo
    Current Delft framework (PycQED + QCodes)
11:00 - 12:00 Discussion to
1. Summarize conclusions of initial presentations
2. Identify sub-goals for rest of the visit

12:00- 13:00 lunch
13:00 - 15:00 Talk with other lab-members on data acquisition software (Ramiro and Chris in particular)
17:00 - 18:00 Meeting Johannes and Adriaan, draw conclusions from the day.
19:00 - Dinner, opportunity to discuss software in a more relaxed/less structured setting

Friday
Goal: plan for software collaboration in QuSurf + decision on platform
9:00 - 10:00 Meeting Johannes + Adriaan, goal: a plan for structural software collaboration
12:00 - 12:30 Werkbespreking lunch
12:30 - 13:30 Werkbespreking
16:00 Meeting with Leo, Johannes, Niels and Adriaan to discuss next steps on software platform. Additonally present plan for collaboration.
17:00 Celebrate @ TPKV!

Goal of this presentation

Understand

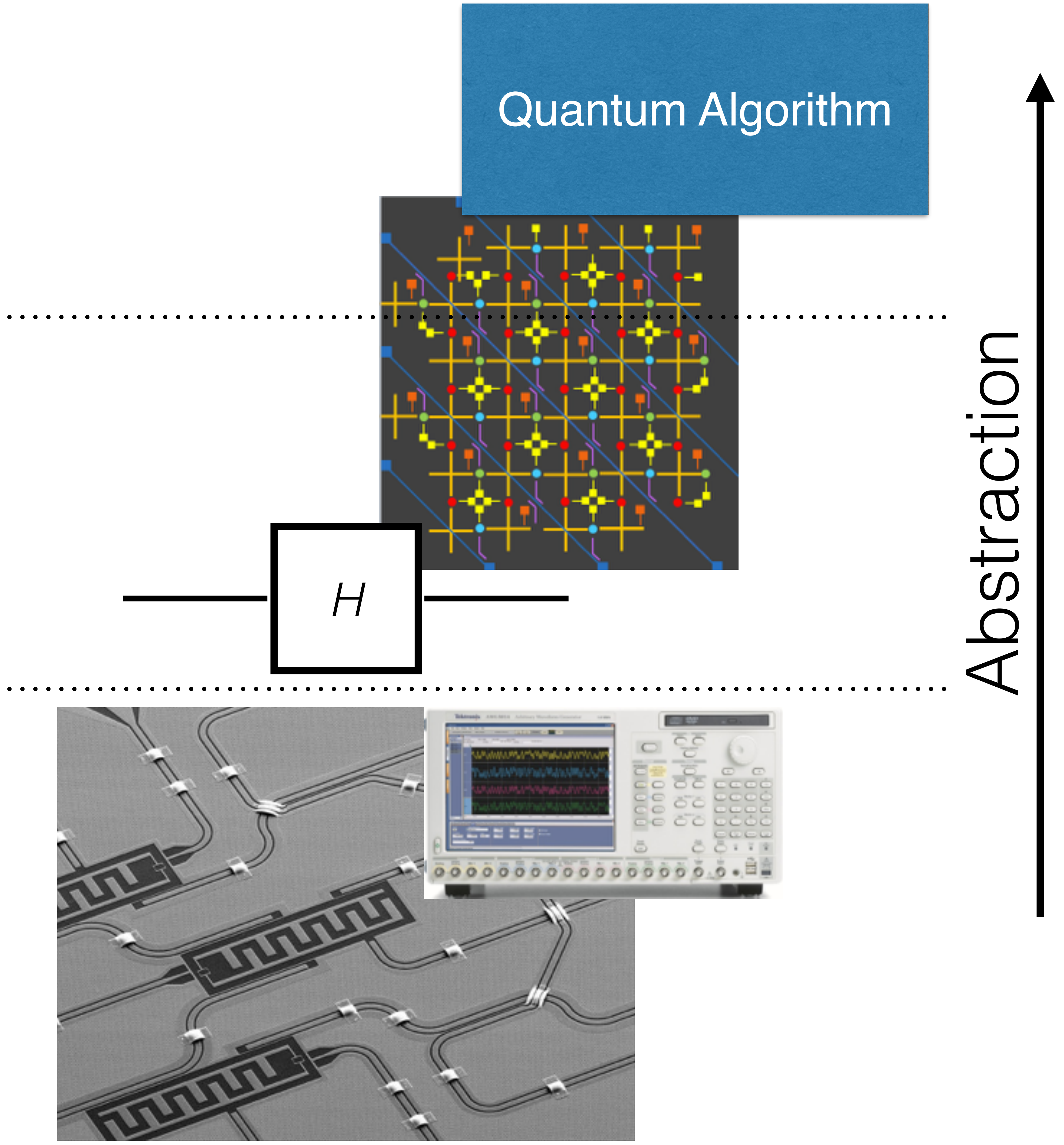
- **Requirements** of data acquisition software
- **Current capabilities** and platform
- Potential for **collaboration**

Requirements

- Controls the setup, logs and plots the data
- Automatable and extensible
- Easy to learn/use
- Grows with newest experimental insights

Current Platform: Building layers of abstraction

- High level - ‘Quantum Compiler’
    - Solid?
- Mid level - System specific bridge to abstraction
    - PycQED
- Low level - Instrument control software
    - QCodes/QTlab



The current framework is build around a minimal set of concepts

Core concepts

- Parameter
  - Sweep function
  - Detector function
- Instrument
- MC/Loop
- Analysis

Derived concepts

- Composite parameter
- hard/soft sweeps/parameters
- Meta-instrument
- Adaptive measurements
- Qubit-object

Every experiment consists of a Loop

Example: Heterodyne experiment

Loop:
1. Some **parameter**(s) is/are varied
2. Some **parameter** is measured
3. Data is saved and analyzed

PycQED syntax

```
MC.set_sweep_function(source.frequency)
MC.set_sweep_points(np.linspace(start, stop, steps))
MC.set_detector_function(HeterodyneDetector())
MC.run(name='Heterodyne')
ma.MeasurementAnalysis()
```

QCodes syntax

```
data = qc.Loop(source.frequency[start:stop:step]).each(
    Heterodyne.IQ).run(name='Heterodyne')
qc.QTPlot(data.IQ)
```

The Loop/MC takes care of standardised datasaving, logging and live plotting

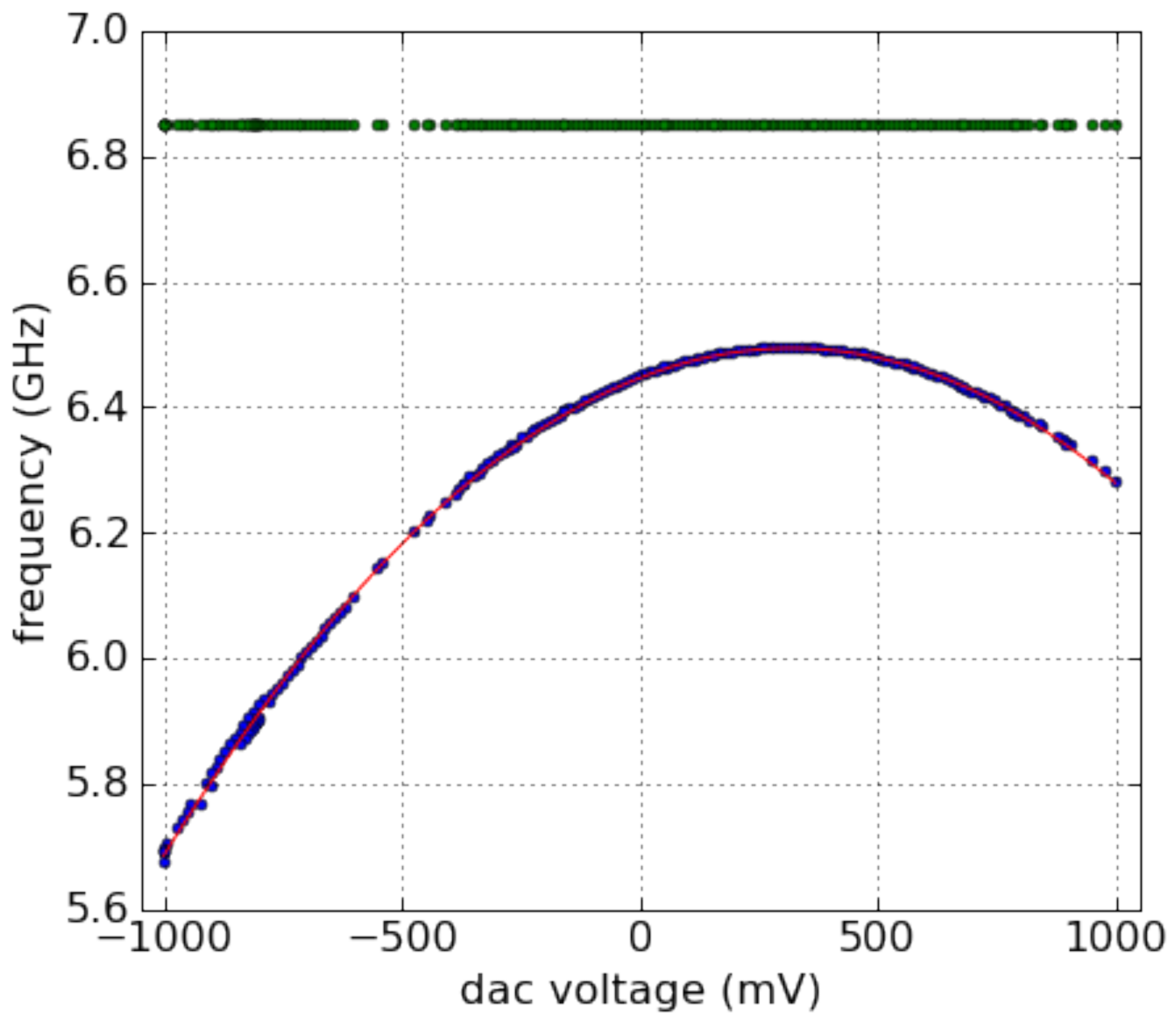
The notion of a parameter is sufficiently abstract that it allows nested/composite measurements

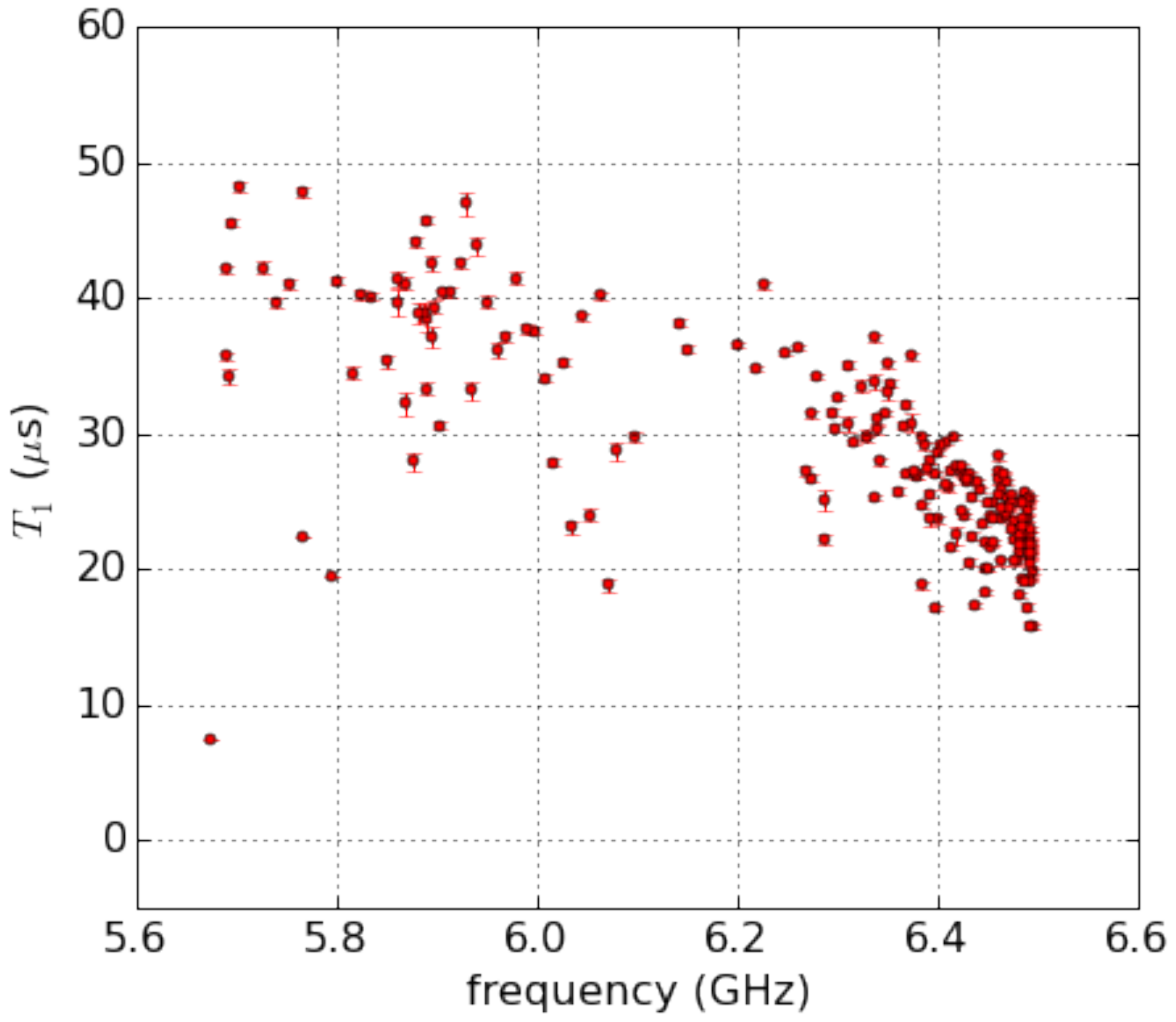
Example: T1 measurement

```
MC.set_sweep_function(swf.Flux_Control_mV(dac_channel))
MC.set_sweep_points(virtual_dac_voltages)
MC.set_detector_function(cdet.T1_Detector(qubit))
MC.run(measurement_name='T1_Mux0_Bottom_Qubit',
       debug_mode=False)
```

Consists of :
1. Finding the resonator
2. Finding the qubit
3. Calibrating the pulse amplitude
4. Performing a T1 measurement

A **composite parameter** can contain other parameters and/or complete loops with analysis

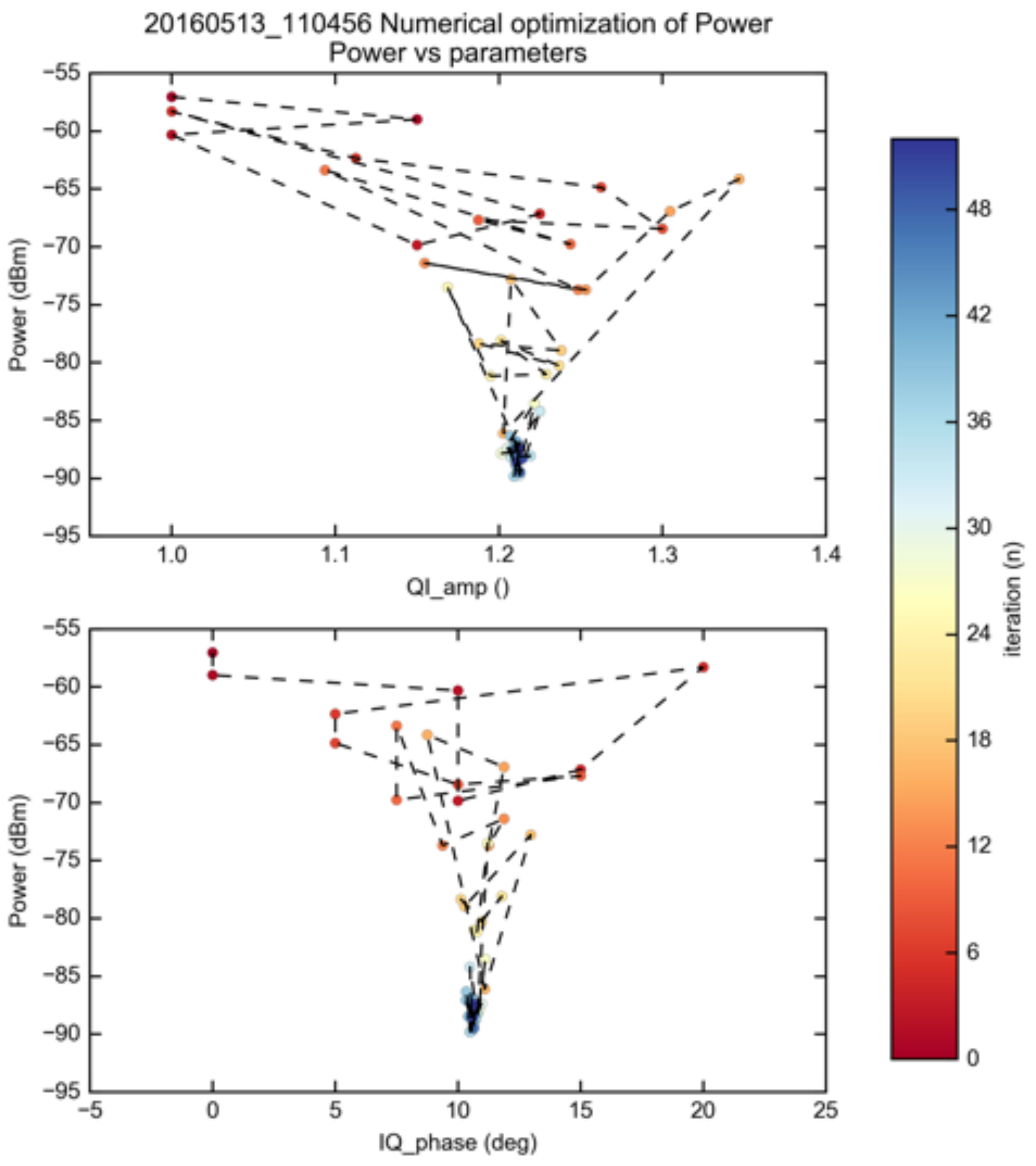




The concept of the loop is sufficiently flexible that it allows adaptive measurements

Example: Numerically optimised mixer skewness

```
ad_func_pars = {'adaptive_function': nelder_mead,
                'x0': [1.0, 0.0],
                'initial_step': [.15, 10],
                'no_improv_break': 5,
                'minimize': True,
                'maxiter': 500}
MC.set_sweep_functions([S1, S2])
MC.set_detector_function(d)
MC.set_adaptive_function_parameters(ad_func_pars)
MC.run(name=name, mode='adaptive')
a = MA.OptimizationAnalysis()
```



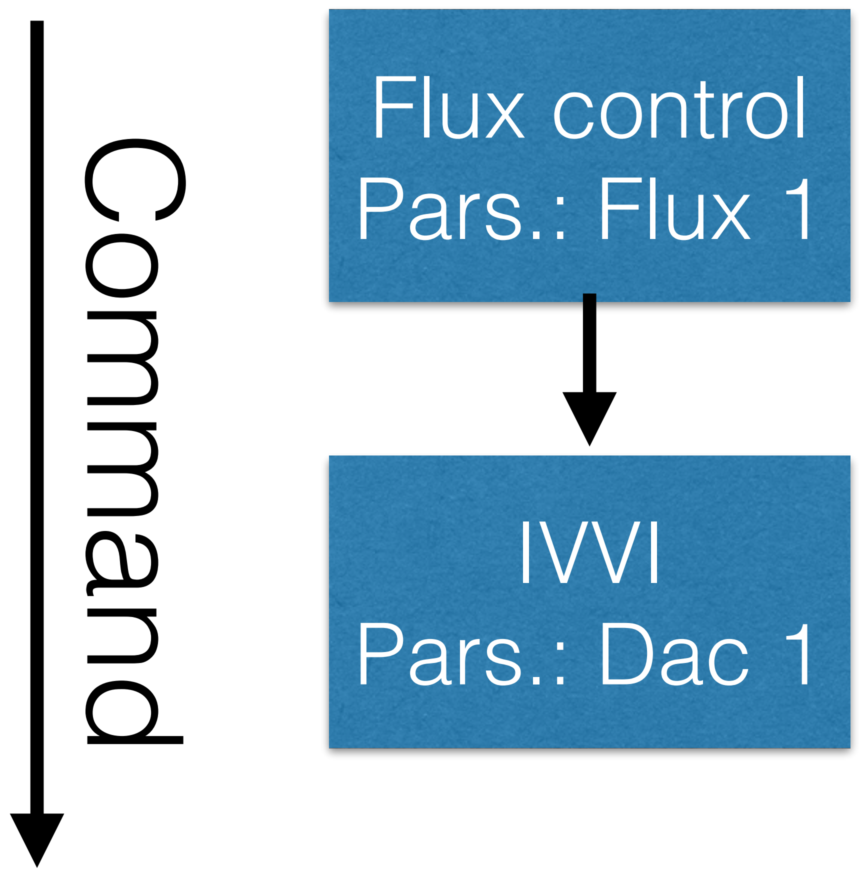
The adaptive loop can use **any adaptive function** on the results of anything we can quantify/measure

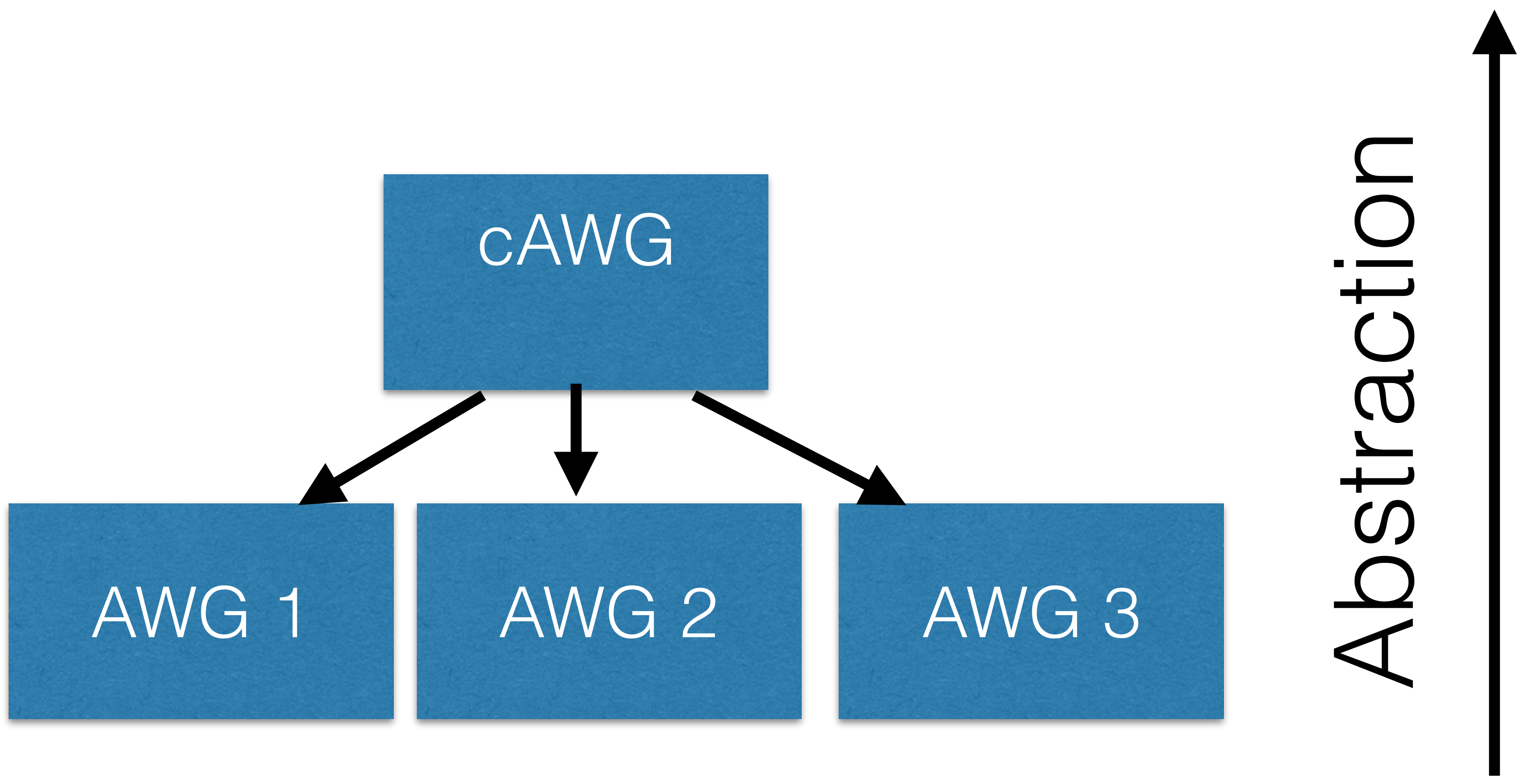
Meta-instruments allow layers of abstraction

Example: Flux-control
Converts fluxes to dac-voltages
using a calibrated correction matrix

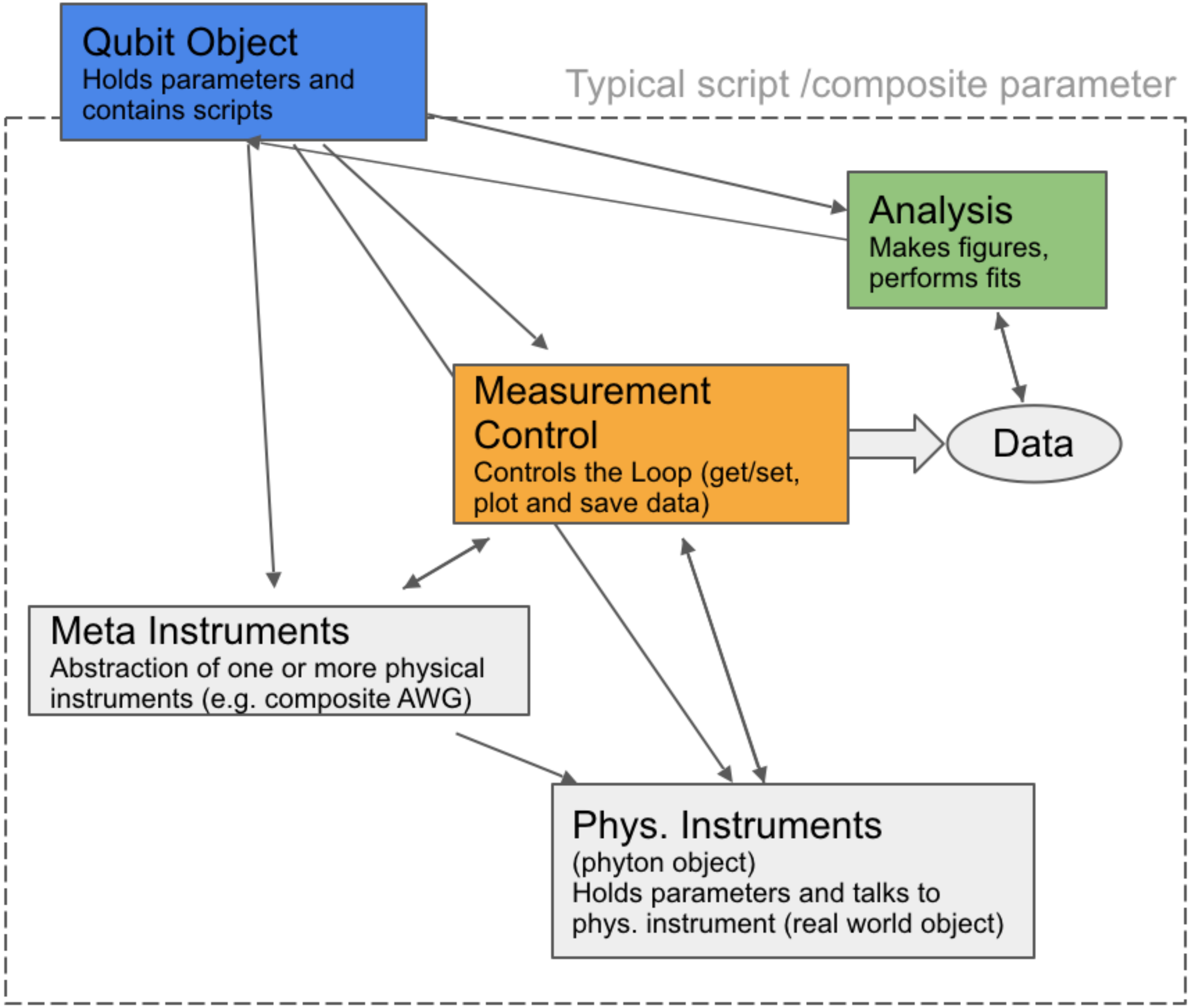
Example: Composite AWG
Acts as a single multi-channel AWG
but talks to underlying instruments

A **meta-instrument** can contain other instruments but acts like a regular instrument





The **qubit object** is a special meta-instrument that executes small scripts (e.g. find frequency using Ramsey) and holds parameters



Potential for collaboration

PyCQED
Specific to cQED

- Waveform sequencing
- Meta Instruments
- Standard experiments and routines
- Measurement Control → Loop
- Specific analysis
- Basic analysis and fitting → Basic analysis and fitting

QCoDeS
General to experimental physics

- Physical Instruments
- Live Plotting
- Loop
- Waveform sequencing?
- Basic analysis and fitting
- Data handling

Software collaboration ETH + QuTech

19-5-2016

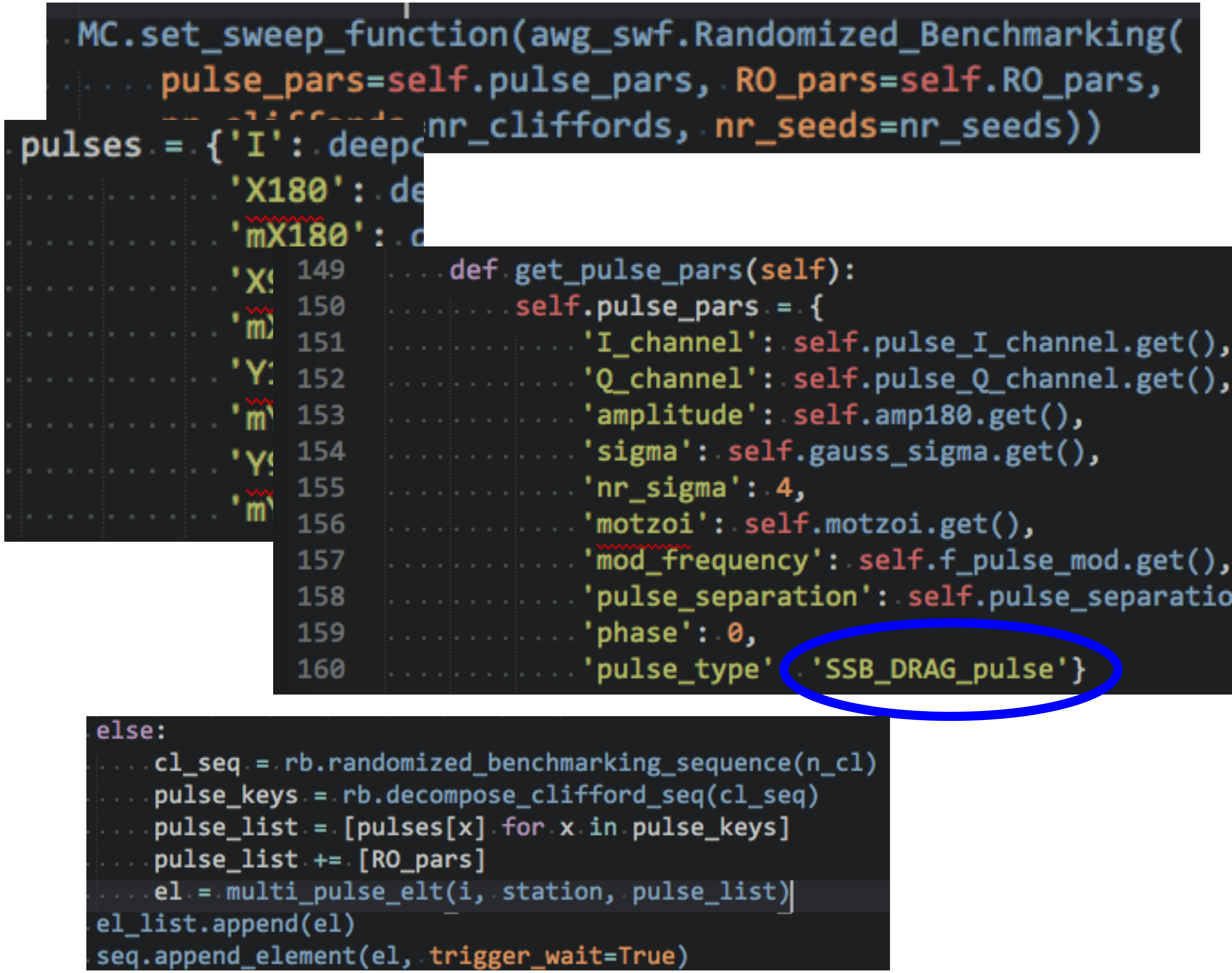
AWG sequencing (made easy)

Pulses(dict) containing pulses by “name” (X180 etc)

Pulse_pars (dict)

AWG segment is made by adding pulses (by key) in a for loop

Pulse-definition is contained in pulse-lib (currently only RO pulse and SSB Drag pulse)



- Includes fix-point correction
- Automated sideband modulation
- Allows for pulse definitions including markers
- In-prinicple extensible to arbitrary nr of channels