http://cs.adelaide.edu.au/~markus/

An Academic Optimisation Problem in Logistics and Two Real-World Optimisation Problems Related to Energy

Markus Wagner
markus.wagner@adelaide.edu.au

University: T

OPTIMISATION AND LOGISTICS

Renewable Energy

Coordinator:
Dr Markus Wagner

Algorithmic Game Theory

Coordinator:
Dr Mingyu Guo

Foundations of Heuristics

Coordinator:
Prof Frank Neumann

Staff Profile:

6 faculty members
2 postdocs
8 PhD students

Search-based Software Engineering

Coordinator:
Dr Bradley Alexander

Supply Chain Management

Coordinator:
Dr Sergey Polyakovskiy

OPTIMISATION AND LOGISTICS

Supply Chain Management (Australian Research Council funded)

- Large scale industrial optimisation problems with many interacting components.

Dynamic Constraints (ARC funded)

- Algorithms for problems with dynamically changing constraints.

Dynamic Adaptive Software Configurations (ARC funded)*

- Self-adapt system configurations to changing conditions.

Lots of other knowledge, either in-house or via international collaborations, e.g. more theory, system modelling, speed-up of simulations (algorithmically or using machine learning)…

SOME OF THE ACTIVITIES OF OPTIMISATION AND LOGISTICS 2016-2018

- ACM Genetic and Evolutionary Computation Conference 2016
  (General Chair: Frank Neumann)
- NII Shonan Meeting on "Computational Intelligence for Software Engineering, Shonan Village Centre, Japan.
  Organizers: Hong Mei (Peking), Frank Neumann (UoA),
  Xin Yao (Birmingham)
- Dagstuhl Seminar on "Automatic Algorithm Selection and Configuration", Schloss Dagstuhl, Germany
  Organizers: Heike Trautmann (Muenster), Holger Hoos (Vancouver), Frank Neumann (UoA).
- NII Shonan Meeting on "Data-Driven Search-Based Software Engineering", Shonan Village Centre, Japan.
  Organizers: Markus Wagner (UoA), Leandro Minku (Leicester), Ahmed E. Hassan (Queens U),
  John Clark (York)
- Australasian Conference on Artificial Life and Computational Intelligence 2018
  (General Chair: Markus Wagner)
- International Workshop on Benchmarking of Computational Intelligence Algorithms, BOCIA, http://iao.hfuu.edu.cn/bocia18
  (Co-Chair: Markus Wagner)

MARKUS WAGNER

2003-2009 — UNIVERSITÄT KOBLENZ · LANDAU

2006-2007 — The University of Georgia

2010-2013 — max planck institut informatik

2013 — THE UNIVERSITY of ADELAIDE — Senior Lecturer

Summary: 80+ papers/co-authors/reviews/events/…
1 best paper/poster/presentation/keynote/medal/…
2nd time in Hefei ☺

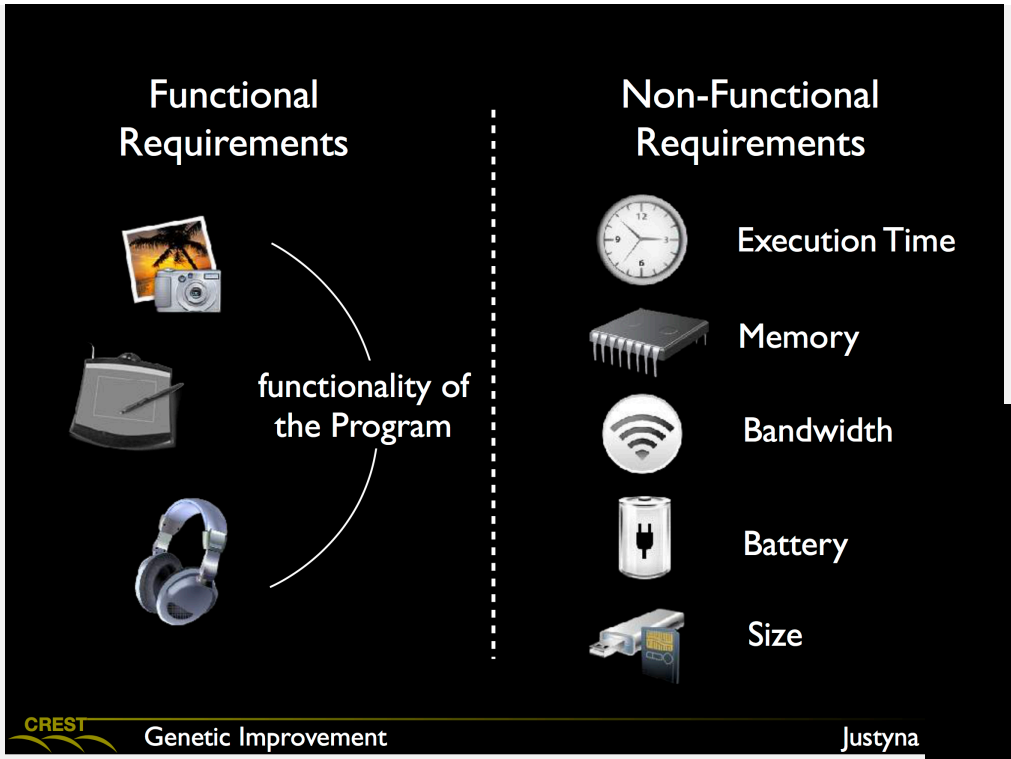
IEEE CIS: Chair University Curricula 2016/2017
Chair Educational Material Subcommittee 2014/2015
Founding Chair of Task Force ”CI in the Energy Domain”

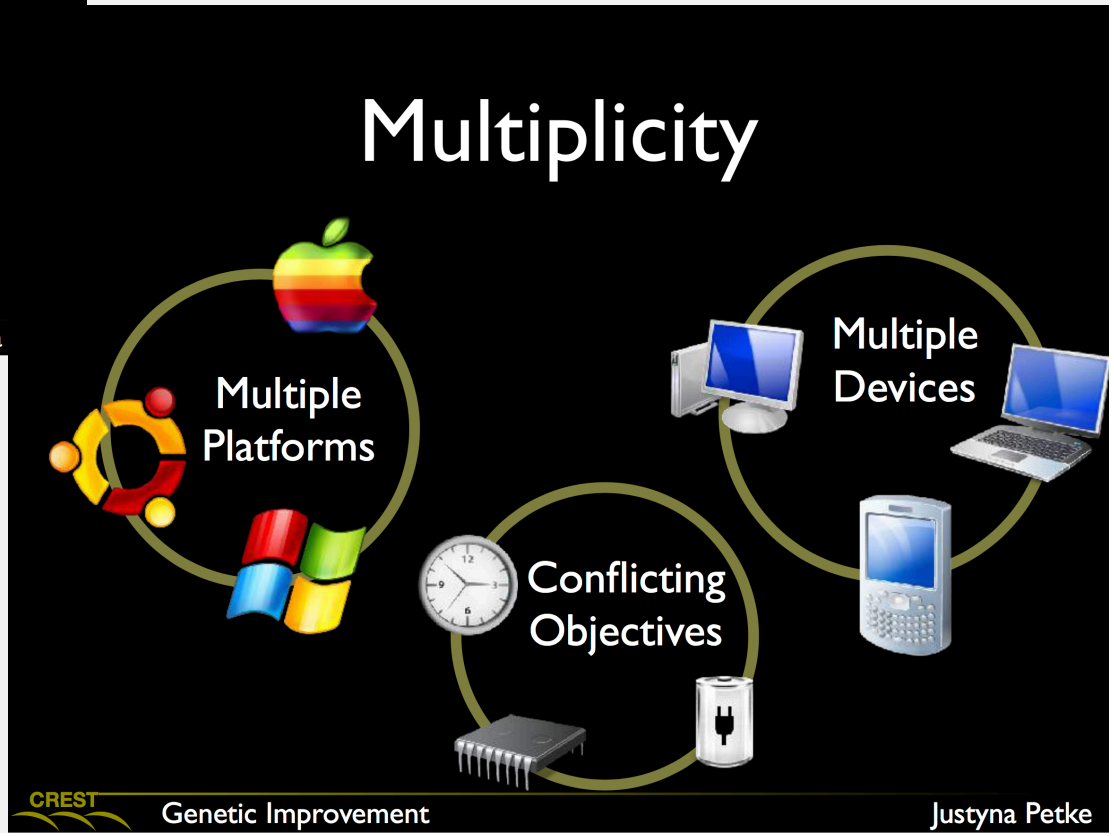
TODAY: change in plans!

The ”logistics problem” will come later.
Let us look at APPLICATIONS first.

Programming is Hard!

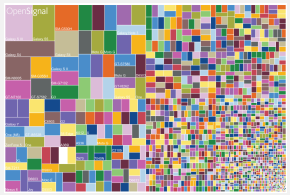
http://phdopen.mimuw.edu.pl/zima15/petke-slides/gi.pdf





Note: over 24,000 different Android phones in 2015

https://qz.com/472767/there-are-now-more-than-24000-different-android-devices/



Genetic Improvement of Software

human writes code

computer improves it

DEEP PARAMETER OPTIMISATION ON ANDROID SMARTPHONES FOR ENERGY MINIMISATION - A TALE OF WOE AND A PROOF-OF-CONCEPT -

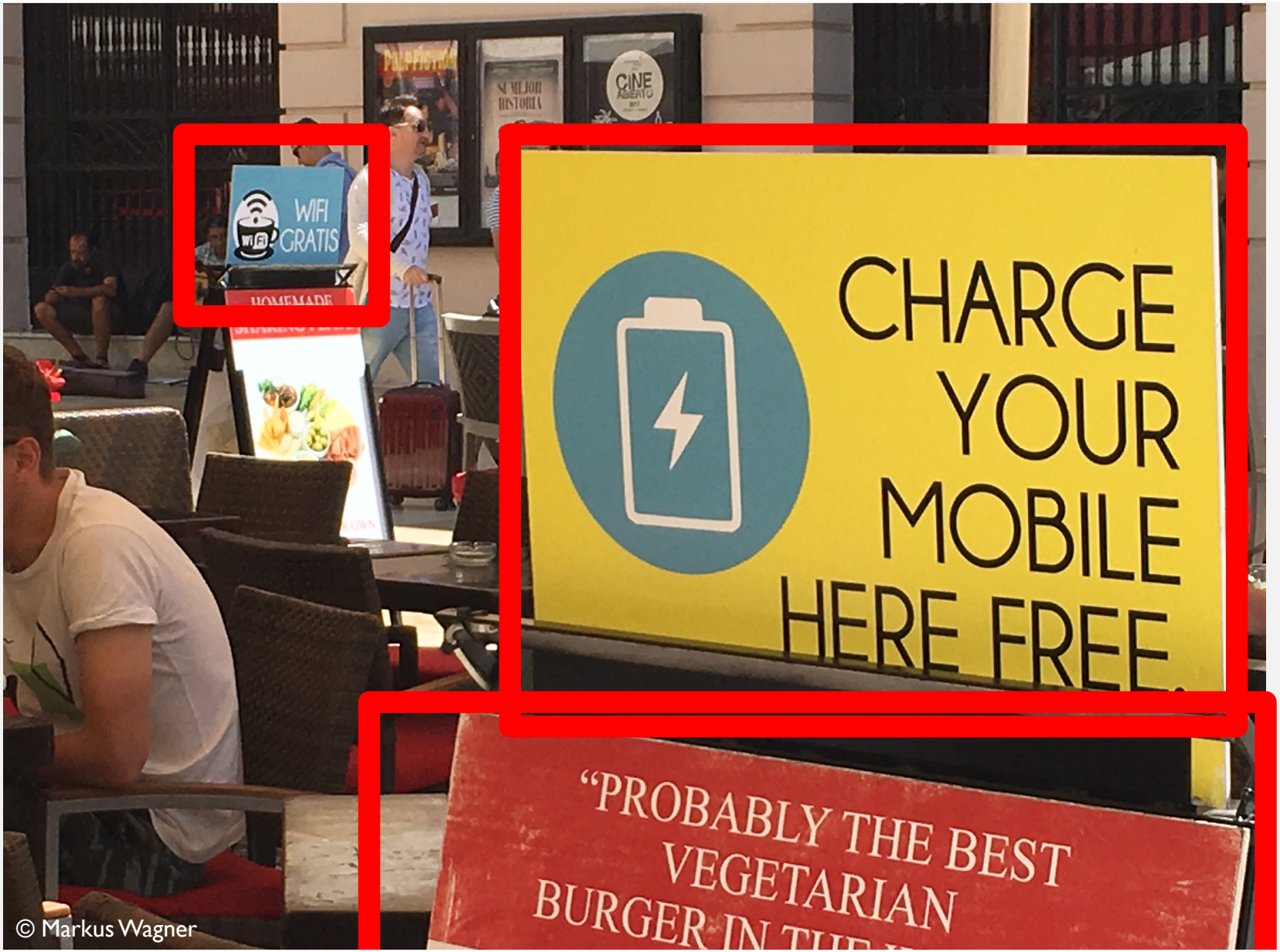
Mahmoud A. Bokhari, Bobby R. Bruce (UCL),
Brad Alexander and Markus Wagner

Full paper: http://cs.adelaide.edu.au/~markus/pub/2017gecco-deepandroid.pdf

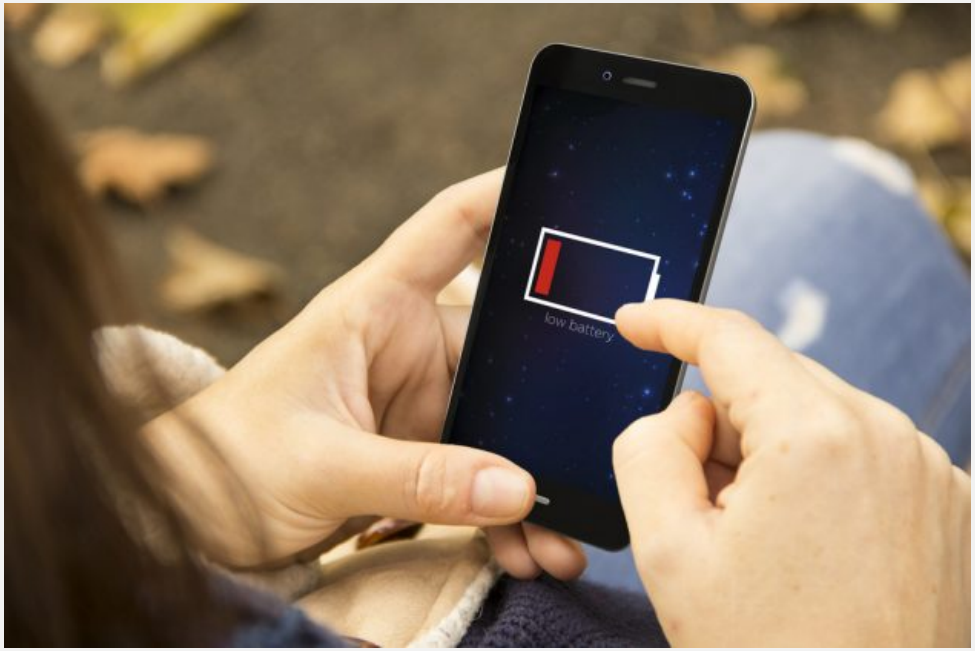
Maslow's hierarchy of needs 2.0

Maslow's hierarchy of needs

Photo taken in Malaga, July 2017



as soon as this appears



shock

anger

denial

the need to get power

MOTIVATION

Number of smartphone users ~2 billion

Users expect

Reality

there is a simple solution

Genetic Improvement

there is a ~~simple~~ solution

Genetic Improvement

MOTIVATION

What do programmers know about the energy consumption of software?

CANDY PANG, ABRAM HINDLE, BRAM ADAMS, AHMED E. HASSAN

How Do Code Refactorings Affect Energy Usage?*

Cagri Sahin
University of Delaware
United States
cagri@udel.edu

Lori Pollock
University of Delaware
United States
pollock@udel.edu

James Clause
University of Delaware
United States
clause@udel.edu

RELATED WORK

Reducing Energy Consumption Using Genetic Improvement

Bobby R. Bruce
University College London
London
United Kingdom
r.bruce@cs.ucl.ac.uk

Justyna Petke
University College London
London
United Kingdom
j.petke@ucl.ac.uk

Mark Harman
University College London
London
United Kingdom
mark.harman@ucl.ac.uk

Optimizing Energy Consumption of GUIs in Android Apps: A Multi-objective Approach

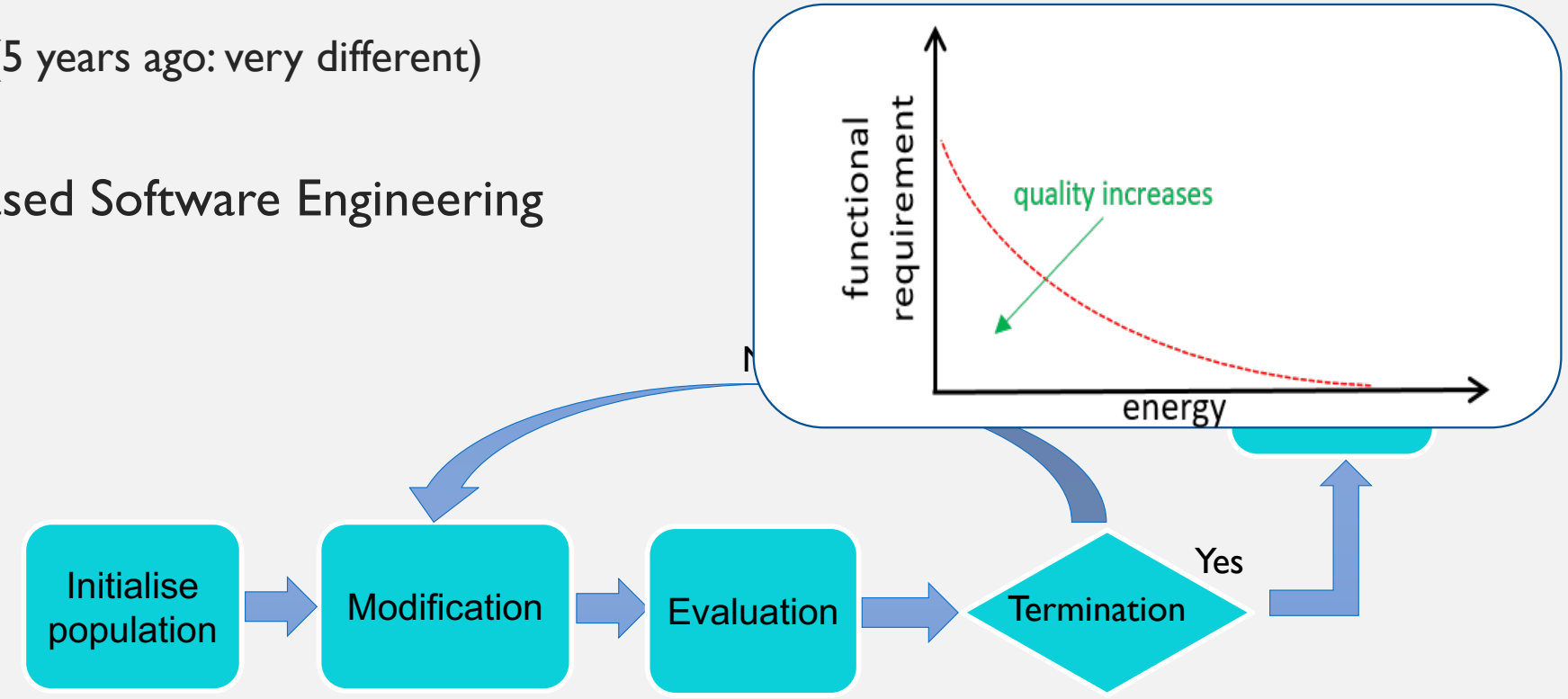
Mario Linares-Vásquez[1], Gabriele Bavota[2], Carlos Bernal-Cárdenas[1]
Rocco Oliveto[3], Massimiliano Di Penta[4], Denys Poshyvanyk[1]
[1]The College of William and Mary, Williamsburg, VA, USA — [2]Free University of Bozen, Bolzano, Italy
[3]University of Molise, Pesche (IS), Italy — [4]University of Sannio, Benevento, Italy

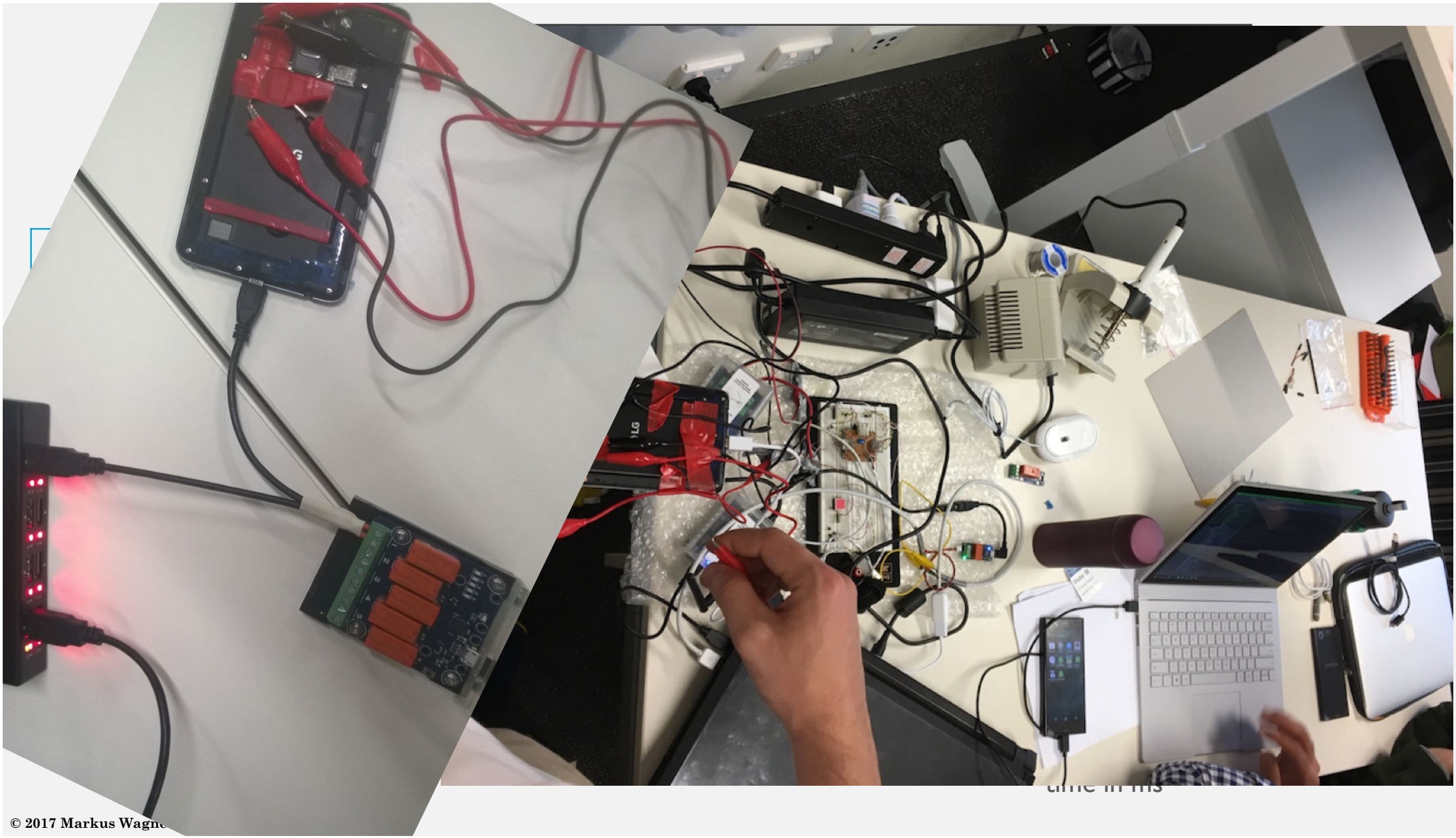
ABSTRACT

The wide diffusion of mobile devices has motivated research towards optimizing energy consumption of software systems—including apps—targeting such devices. Besides efforts aimed at dealing with various kinds of energy bugs, the adoption of Organic Light-Emitting Diode (OLED) screens has motivated research towards reducing energy consumption by *choosing an appropriate color palette. Whilst past research* have been aimed at engineering energy-friendly hardware components in mobile devices, some recent research has also focused on energy-aware development practices for reducing the energy consumption in mobile apps. For instance, common energy bugs in mobile apps have been identified and catalogued [27, 28, 32, 34, 42], as well as typical hot spots [39] together with energy greedy APIs [26,33]. In addition, several infrastructures and methods have been proposed to measure

METHODOLOGY

- Energy Measurement
  - External.
  - Internal. (5 years ago: very different)
- Search Based Software Engineering





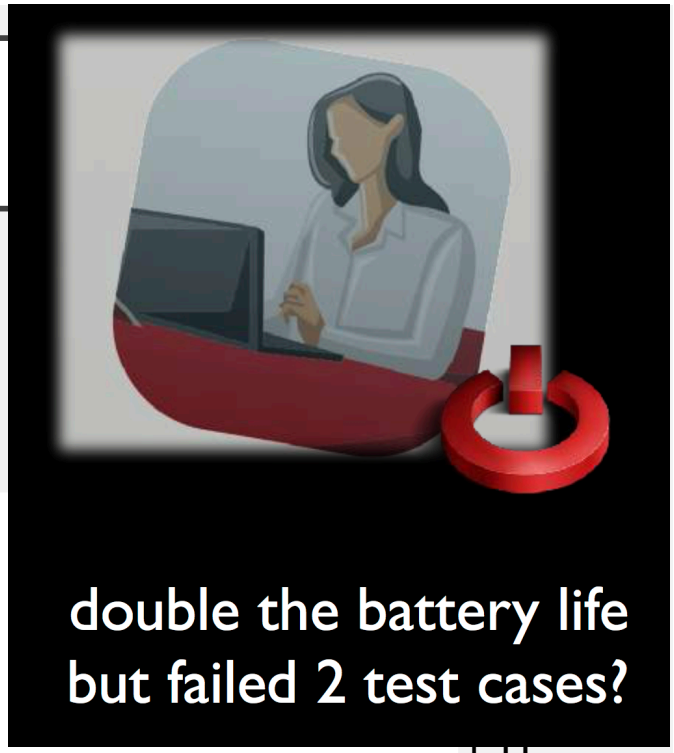
time in his

TARGET APP

- Requirements:
  - Open source.
  - Widely used.
  - Energy-hungry app.
  - Test suite.

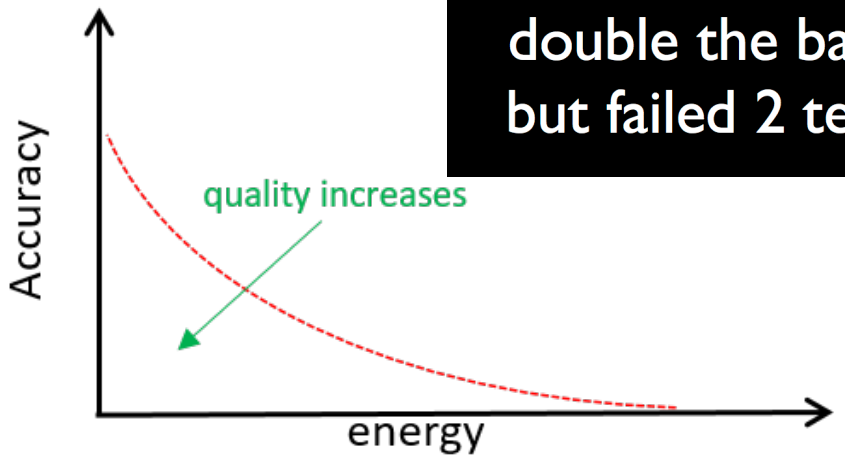
TARGET APP

- Rebound Library:
  - Open source.
  - Widely used.
  - Energy-hungry app.
  - 44 test cases.
    - Speed and position of springs.



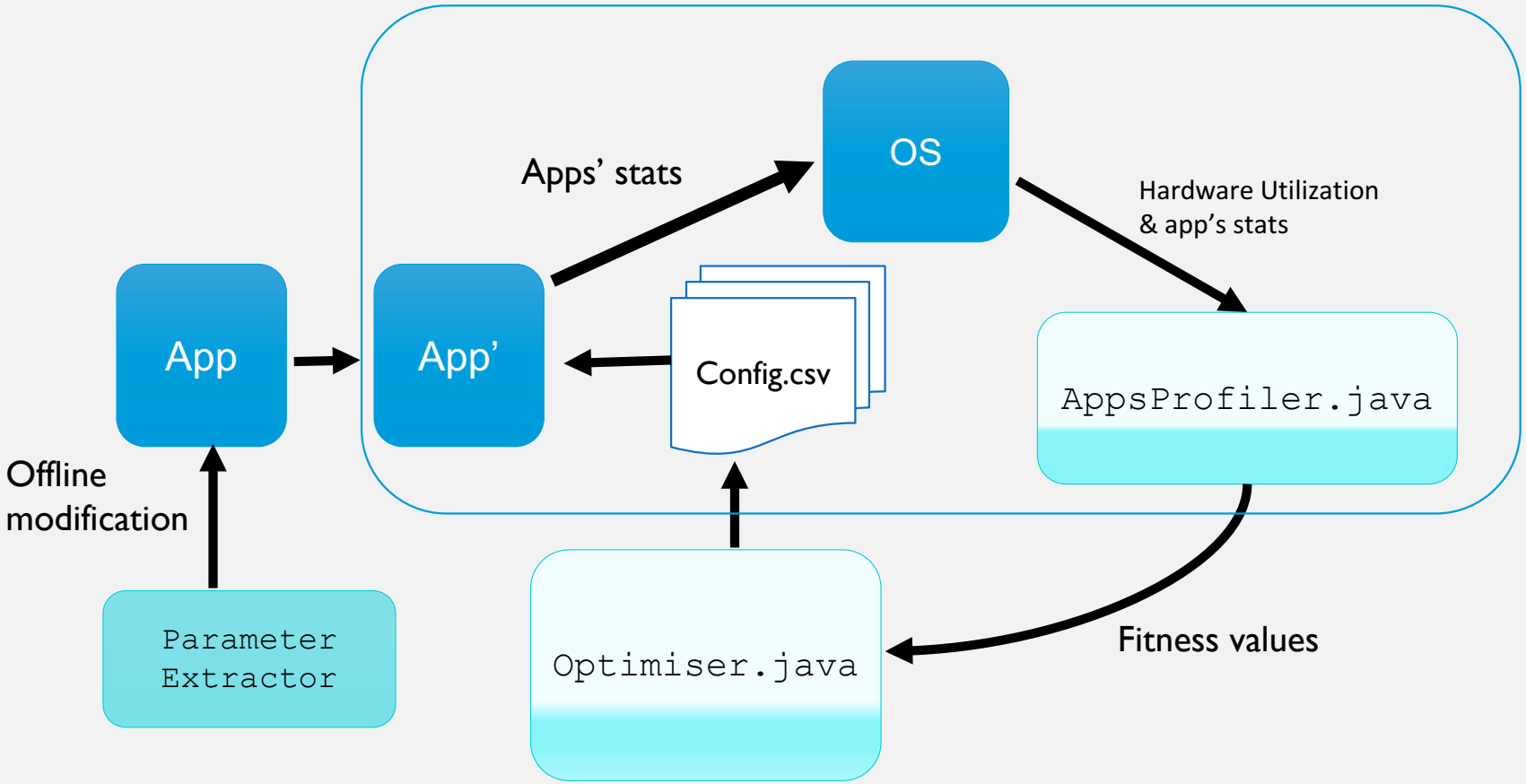
http://face

ok Home



FRAMEWORK

- Deep Parameter Tuning. ➔ Parameter tuning of the physics calculations!



FRAMEWORK

- Deep Parameter Tuning. ➔ Parameter tuning of the physics calculations!

SpringConfigRegistryTest.java × | SpringTest.java × | rebound-core × | Spring.java × | MainActivity.java × | rebou

Spring

```
1   /.../
10
11  package com.facebook.rebound;
12
13  import java.util.concurrent.CopyOnWriteArraySet;
14
15  /**
16   * Classical spring implementing Hooke's law with configurable friction and tension.
17   */
18  public class Spring {
19
20    // unique incrementer id for springs
21    private static int ID = 0;
22
23    // maximum amount of time to simulate per physics iteration in seconds (4 frames at 60 FPS)
24    private static final double MAX_DELTA_TIME_SEC = 0.064;
25    // fixed timestep to use in the physics solver in seconds
26    private static final double SOLVER_TIMESTEP_SEC = 0.001;
27    private SpringConfig mSpringConfig;
28    private boolean mOvershootClampingEnabled;
```

FRAMEWORK

- Deep Parameter Tuning. ➔ Parameter tuning of the physics calculations!

ResultRecorder.java × | SpringTest.java × | Define.java × | Spring.java × | ModifiedRebound × | gradle.properties × | InexactComparis

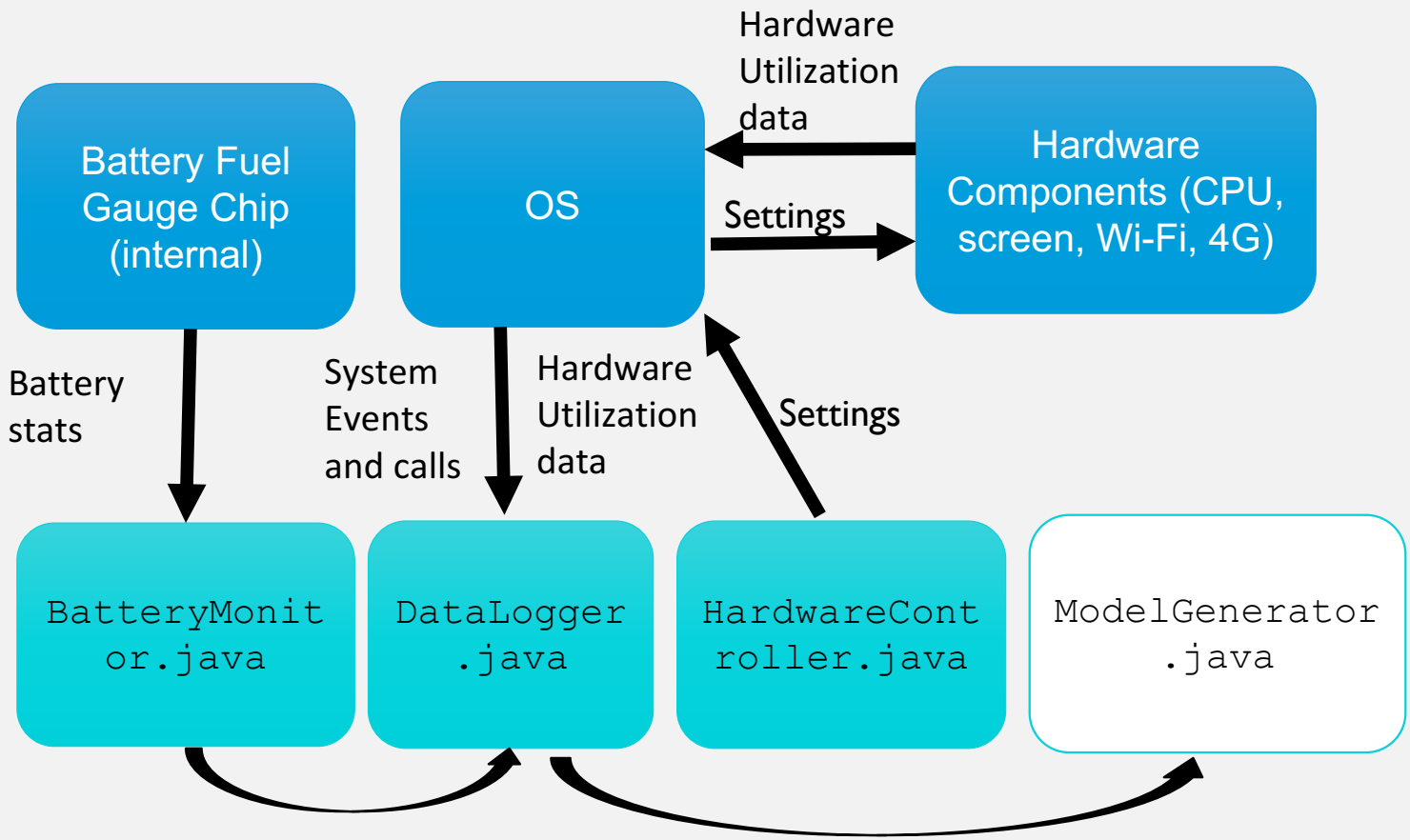
Spring

```
12
13  import java.util.concurrent.CopyOnWriteArraySet;
14
15
16
17  /**
18   * Classical spring implementing Hooke's law with configurable friction and tension.
19   */
20  public class Spring {
21
22      // unique incrementer id for springs
23      private static int ID = Define.Spring_INTEGER_21_1_0;
24
25      // maximum amount of time to simulate per physics iteration in seconds (Define.Spring_INTEGER_23_1_4 frames at Def
26      private static final double MAX_DELTA_TIME_SEC = Define.Spring_DOUBLE_24_1;
27      // fixed timestep to use in the physics solver in seconds
28      private static final double SOLVER_TIMESTEP_SEC = Define.Spring_DOUBLE_26_1;
29      private SpringConfig mSpringConfig;
30      private boolean mOvershootClampingEnabled;
```

19	Spring_INTEGER_48_1_0	0
20	Spring_INTEGER_23_2_60	57
21	Spring_INTEGER_23_1_4	4
22	Spring_INTEGER_21_1_0	0
23	Spring_INTEGER_2_1_2013	2013
24	Spring_DOUBLE_481_1	0
25	Spring_DOUBLE_377_3	2
26	Spring_DOUBLE_377_2	6
27	Spring_DOUBLE_377_1	1
28	Spring_DOUBLE_376_3	2
29	Spring_DOUBLE_376_2	6
30	Spring_DOUBLE_376_1	1
31	Spring_DOUBLE_366_1	0.5
32	Spring_DOUBLE_365_1	0.5
33	Spring_DOUBLE_361_1	0.5
34	Spring_DOUBLE_360_1	0.5
35	Spring_DOUBLE_47_1	0.005
36	Spring_DOUBLE_46_1	0.005
37	Spring_DOUBLE_26_1	0.001
38	Spring_DOUBLE_24_1	0.064

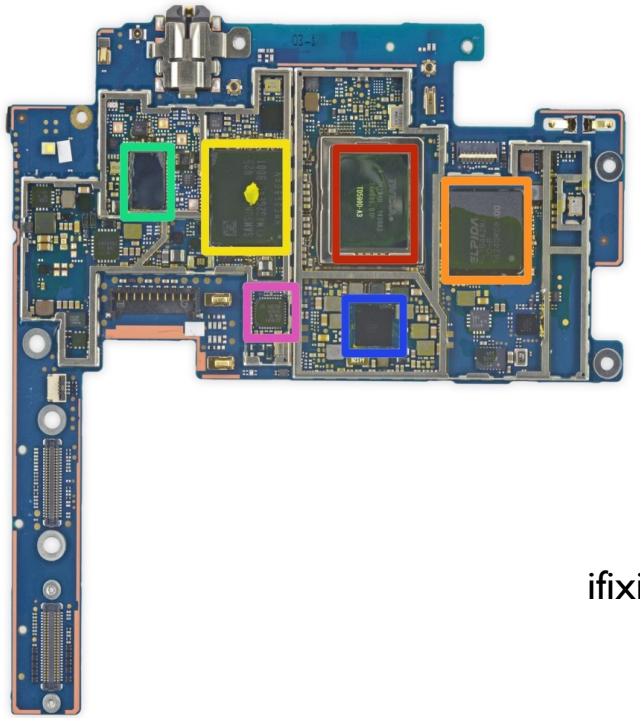
METHOD

Hardware Data:
CPU freq., Screen brightness, transmitted packets, signal strength …
36 dimensions (CPU, Screen, Network)
Software data:
Apps utilisation.

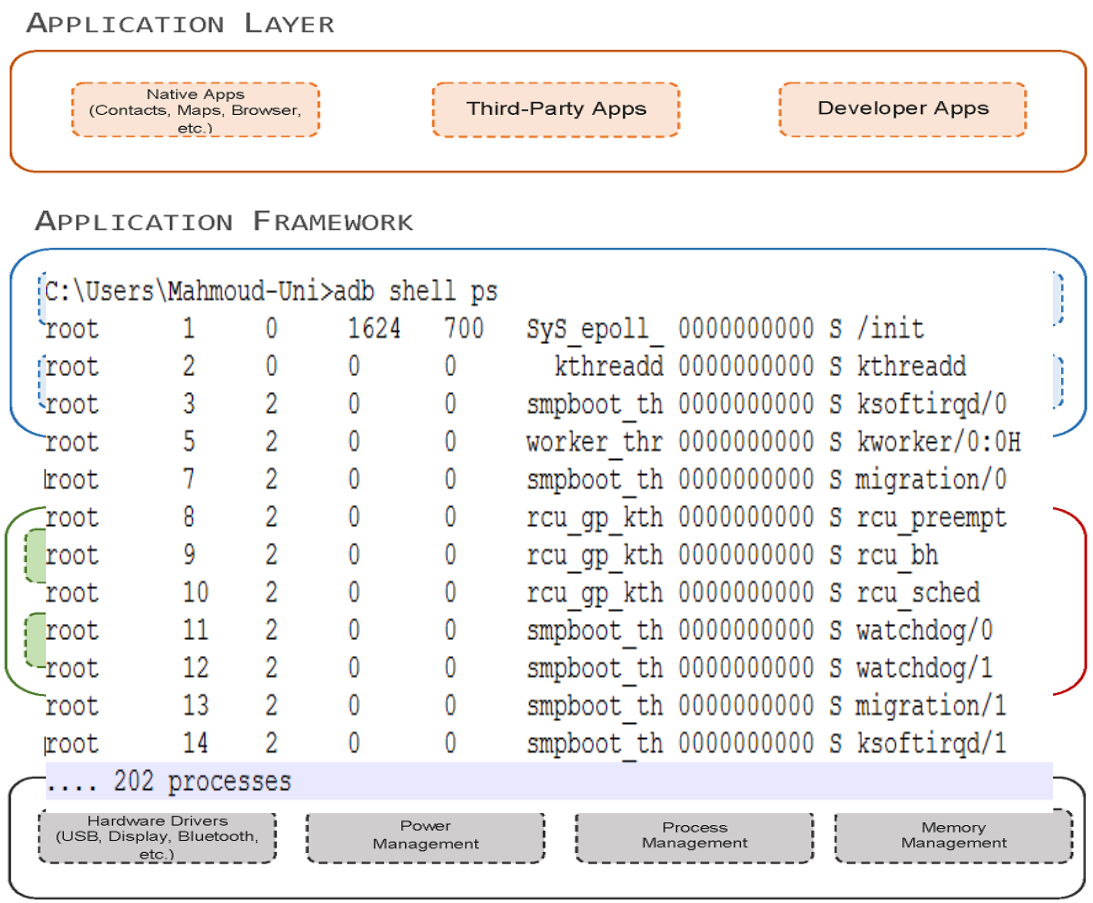


CHALLENGES

- Noise.
  - System Architecture and Behaviour.



ifixit.com

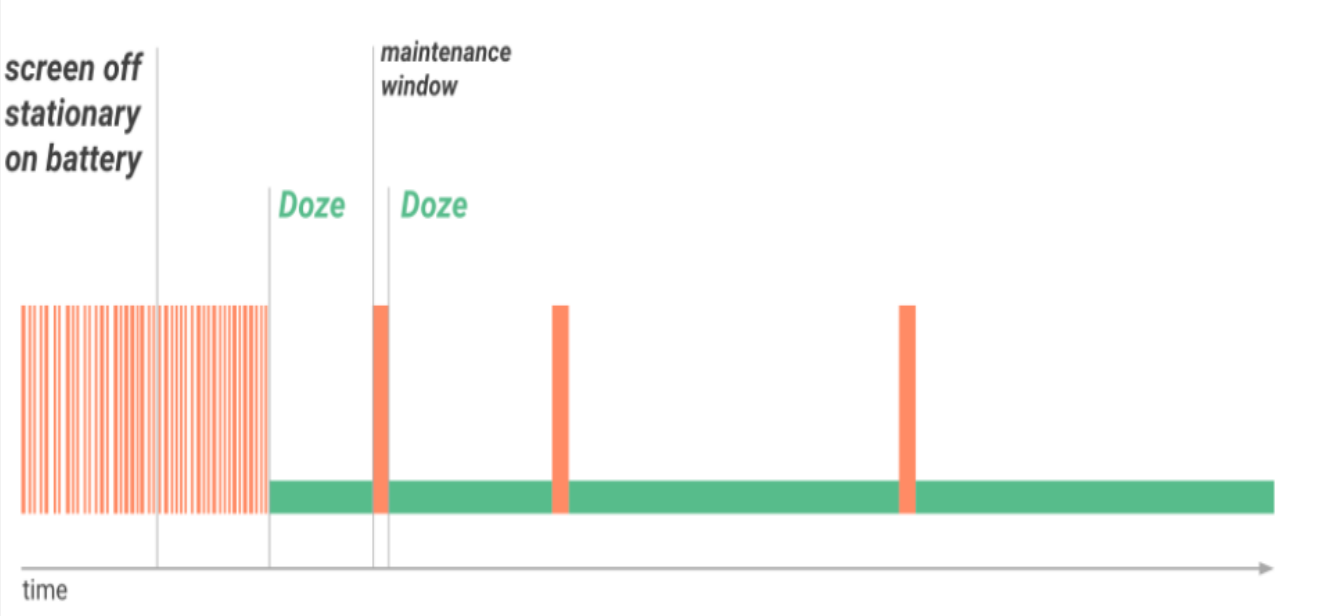


CHALLENGES

- Noise.
  - System Architecture and Behaviour.
    - Solution.
      - Airplane mode.
      - Changing the CPU governor.
      - Fixing CPU frequency.
      - Turning the screen off.

CHALLENGES

- Noise.
  - System Architecture and Behaviour.
    - Doze Mode.

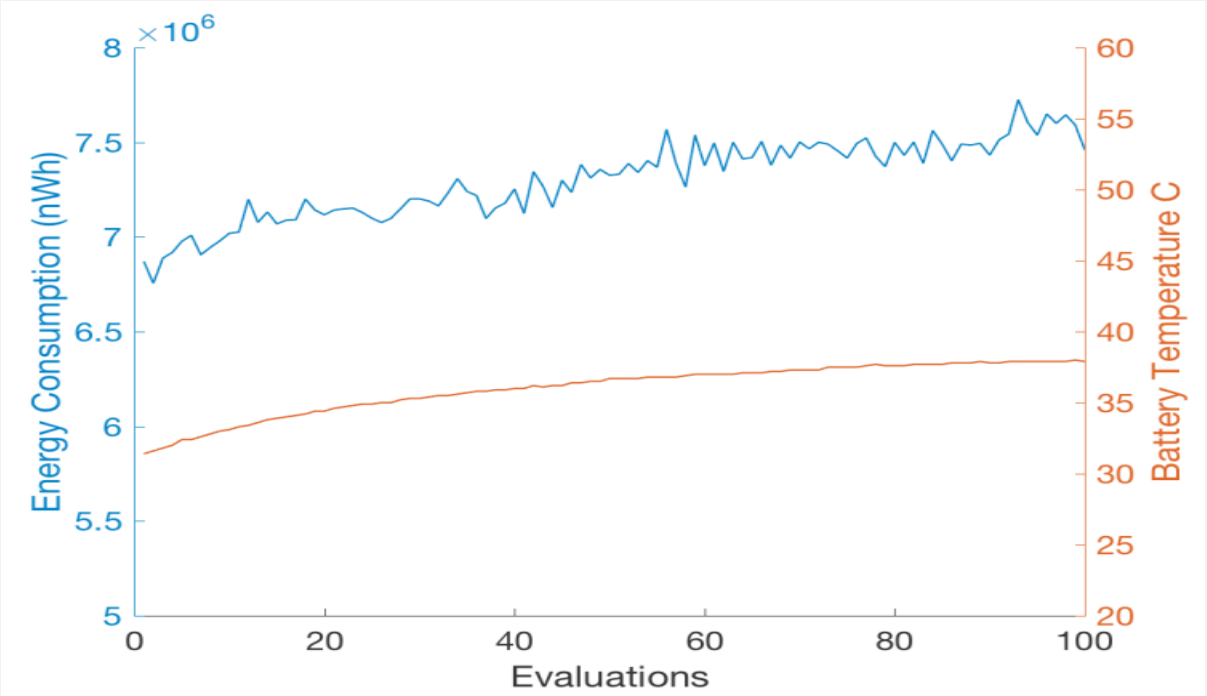


CHALLENGES

- Noise.
  - System Architecture and Behaviour.
    - Doze Mode.
    - Solution.

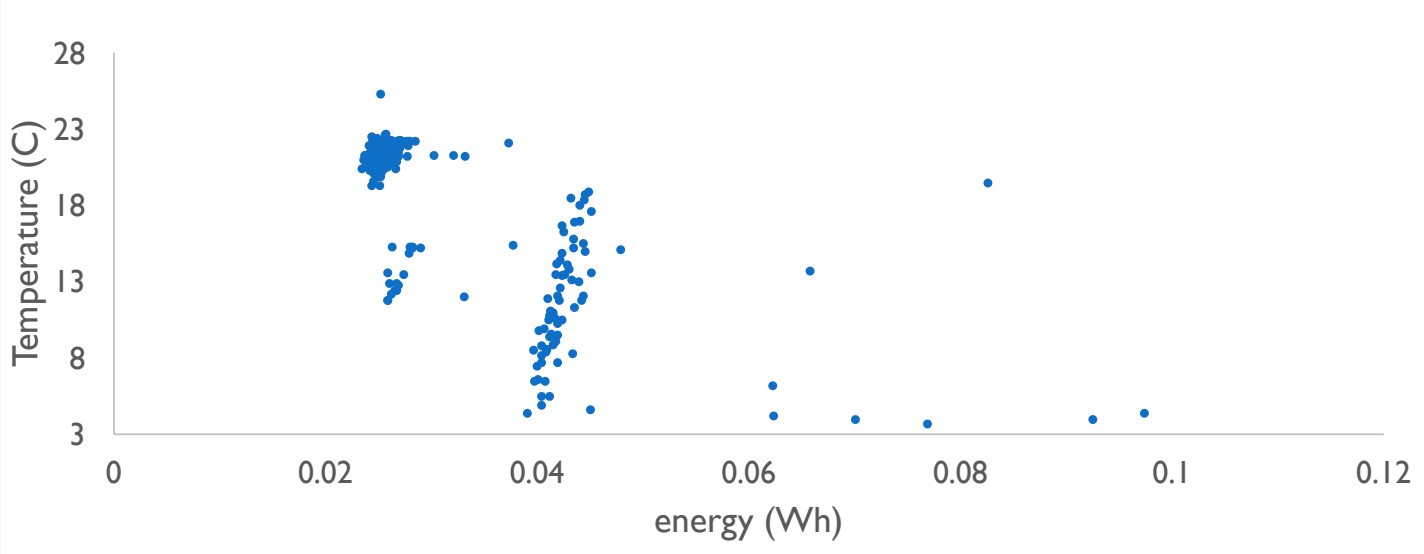
CHALLENGES

- Noise.
  - Temperature.



CHALLENGES

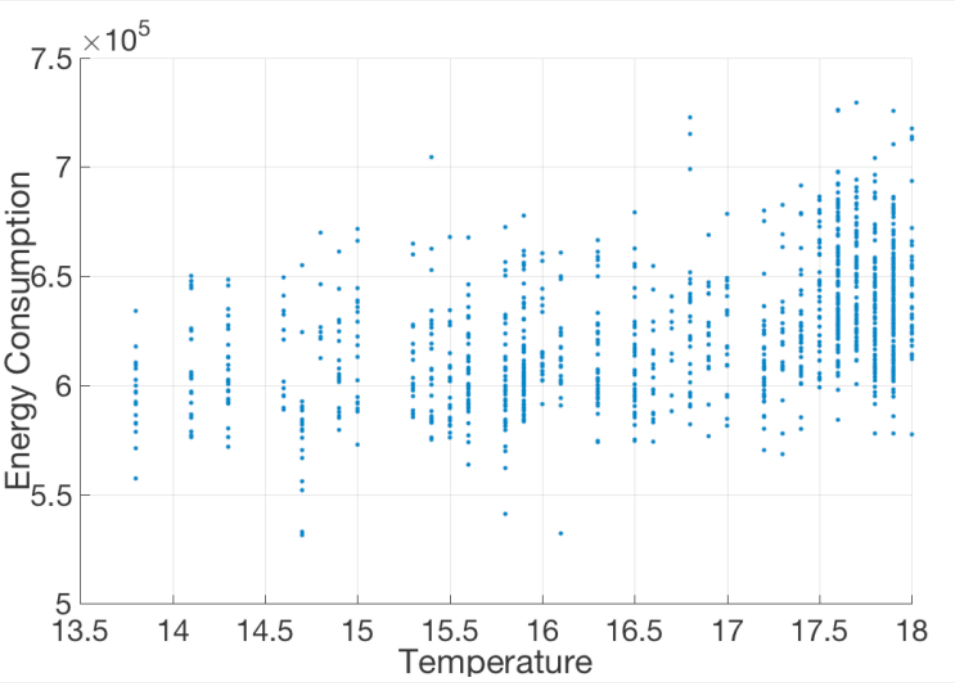
- Noise.
  - Temperature.
    - Fridge.

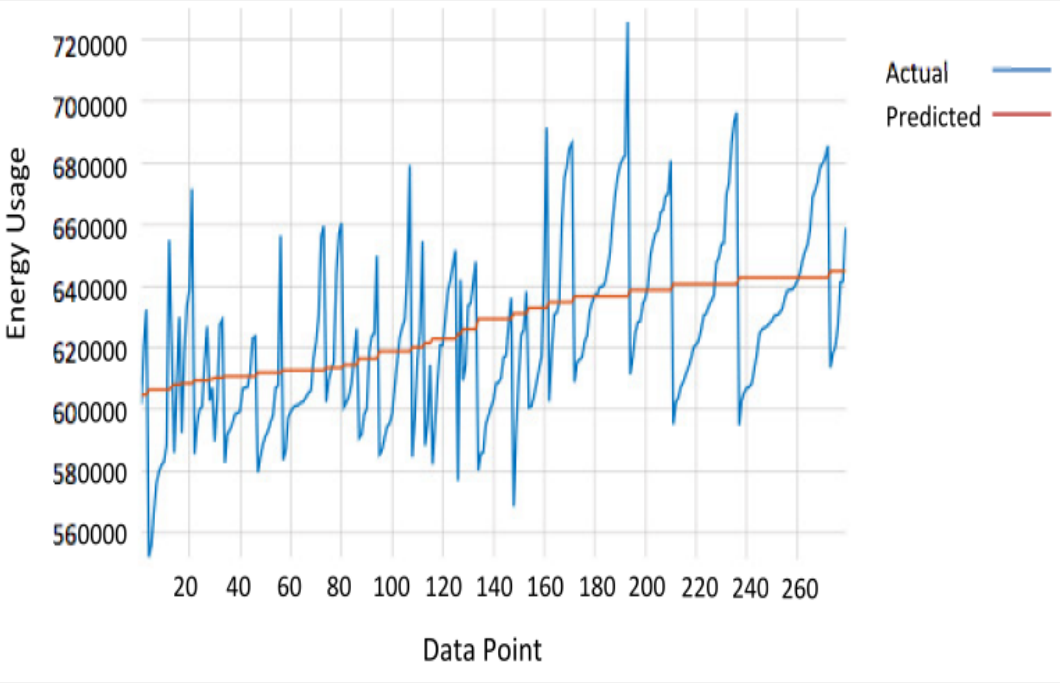


CHALLENGES

- Noise.
  - Temperature.
    - Fridge.
    - Model & compensate?

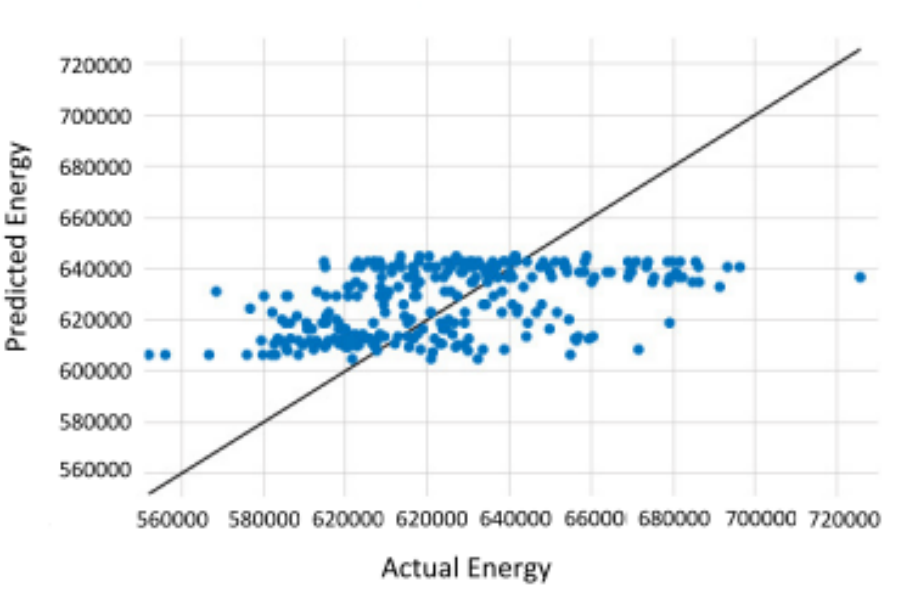
$$e = 357t^2 - 6180.5t + 608000$$





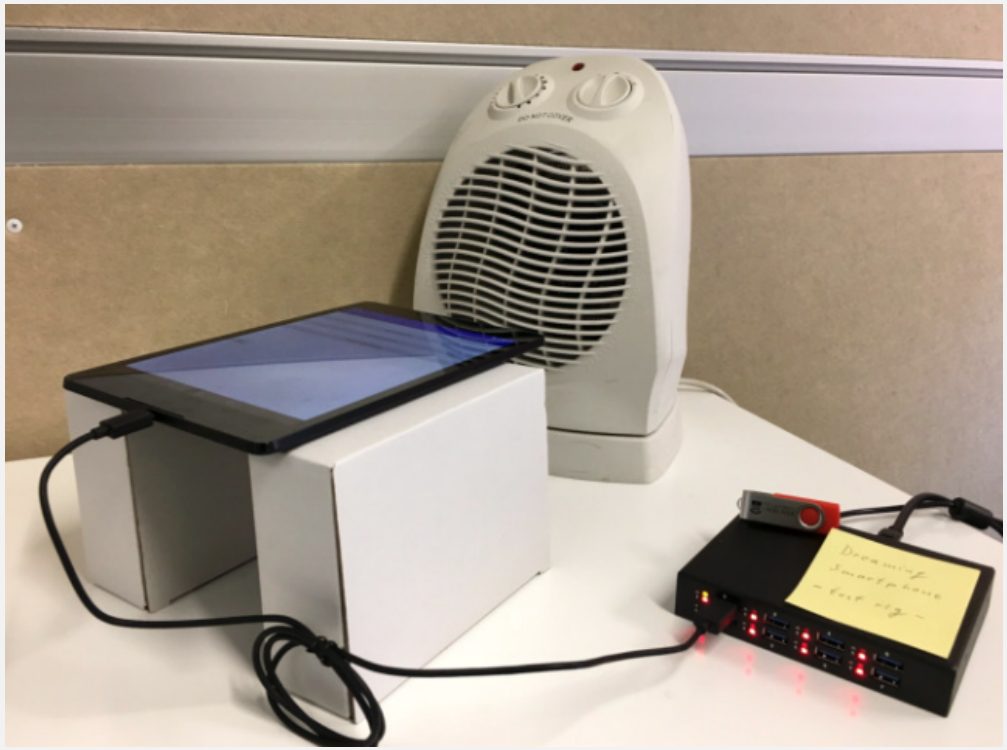
CHALLENGES

- Noise.
  - Temperature.



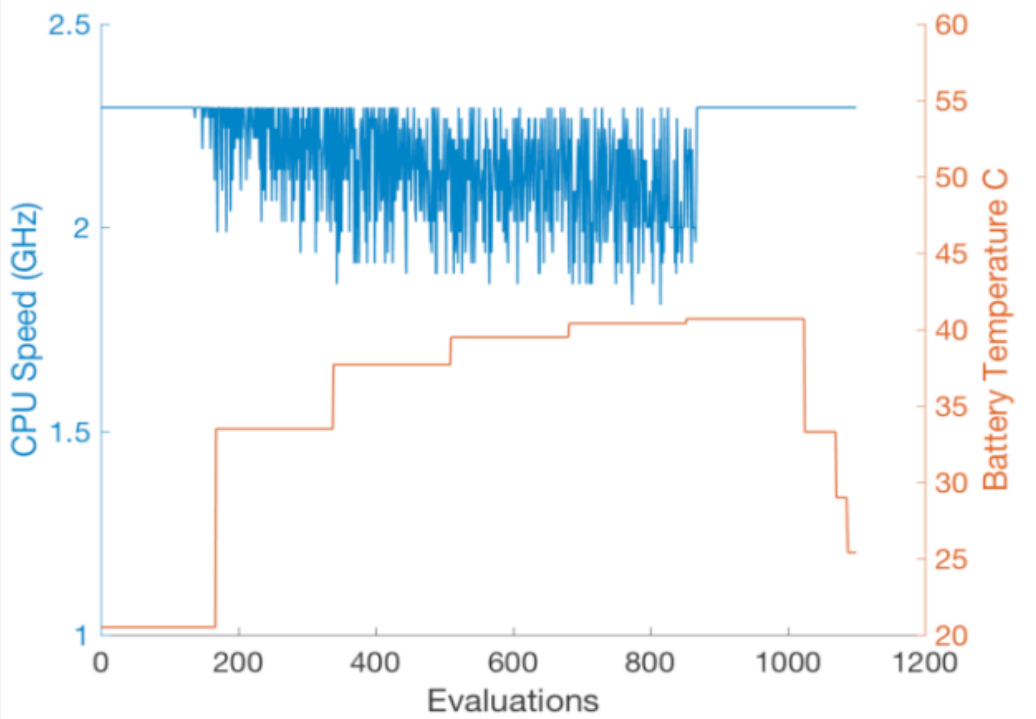
CHALLENGES

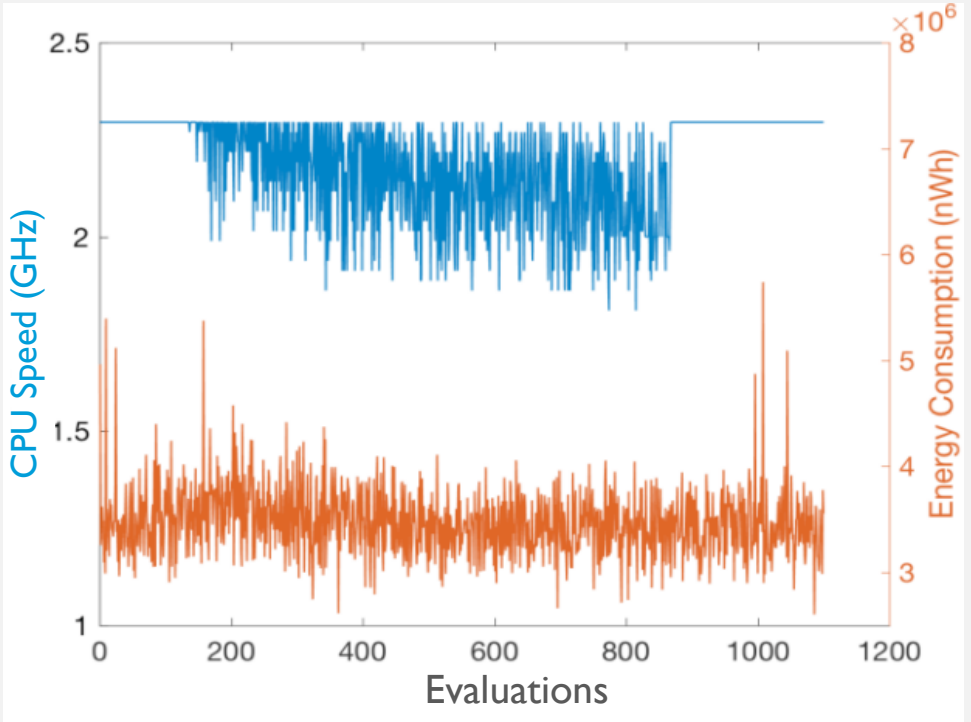
- Noise.
  - Temperature.
  - Solution.



CHALLENGES

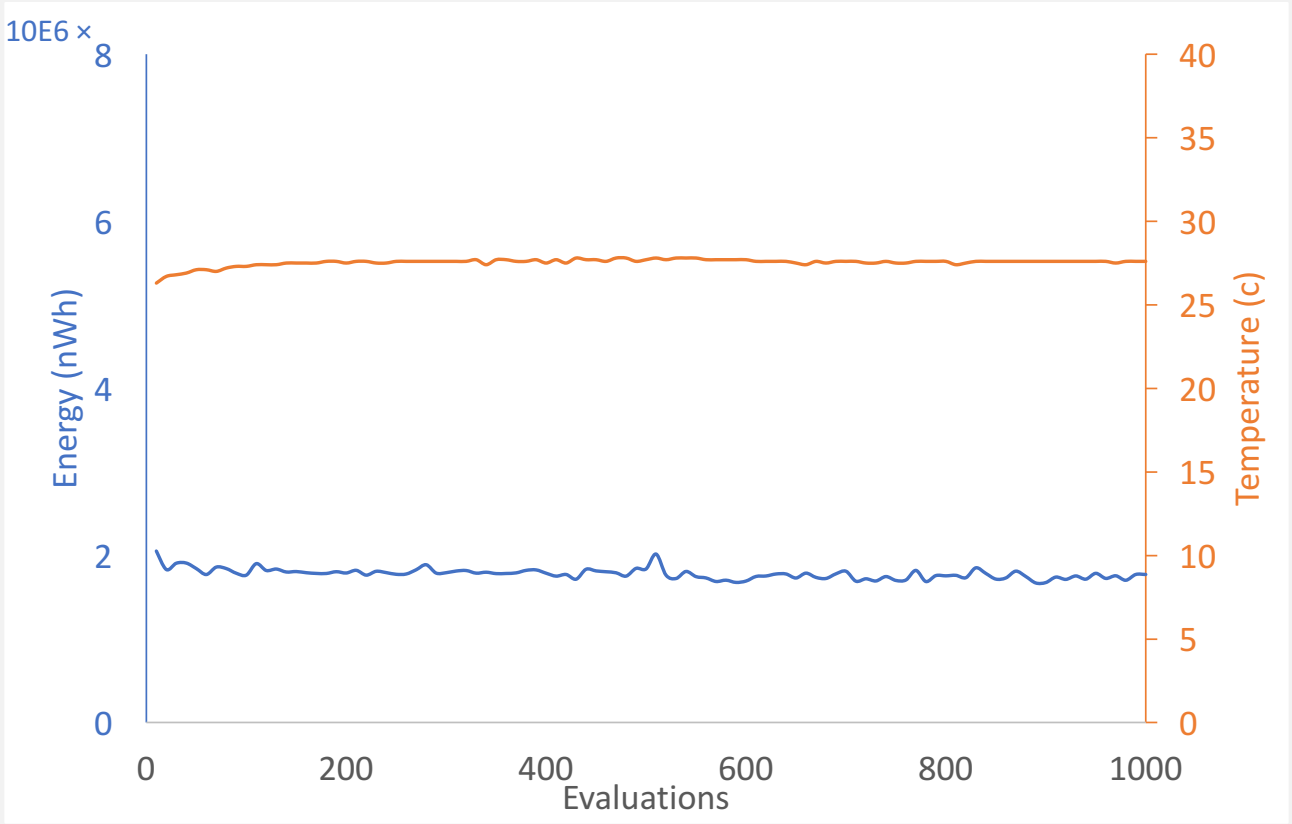
- Noise.
  - CPU Throttling.





CHALLENGES

- Noise.
  - CPU Throttling.
  - Solution.
    - Median frequency: 1.4 GHz.

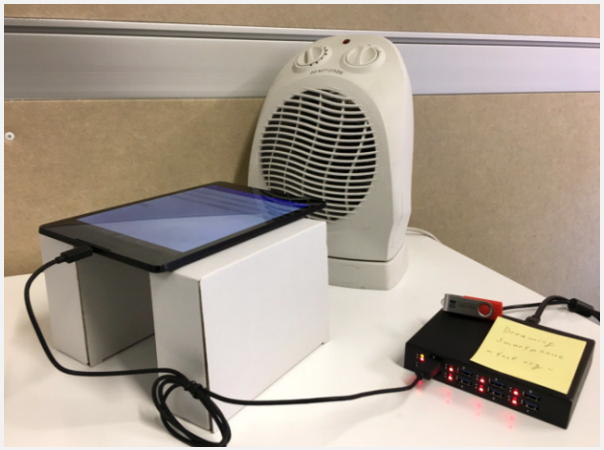


CHALLENGES

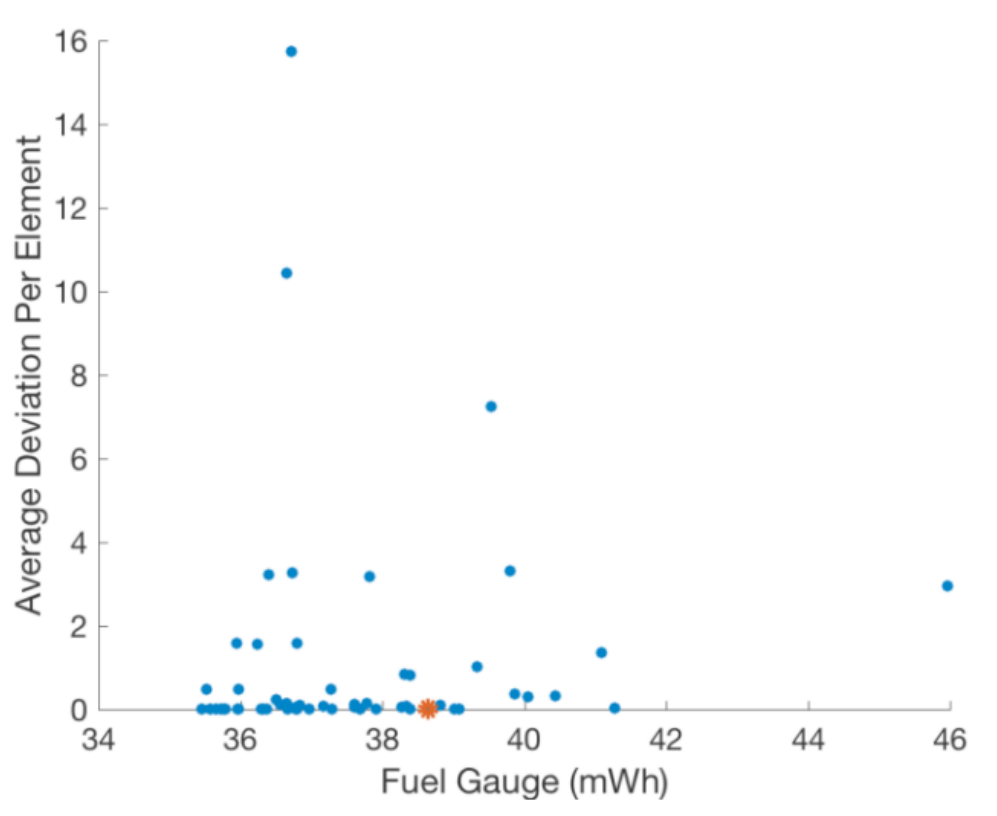
- Android Debug Bridge.
  - Deployment time.
  - Instability.
  - Connectivity Issues.

EXPERIMENT SETUP

- Population size: 20.
  - First population.
- Generations: 17.
- Objectives: Energy & Accuracy.
- Timeout: 2 mins.
- Recharge: 20 mins.
- Measurement frequency: 4Hz.

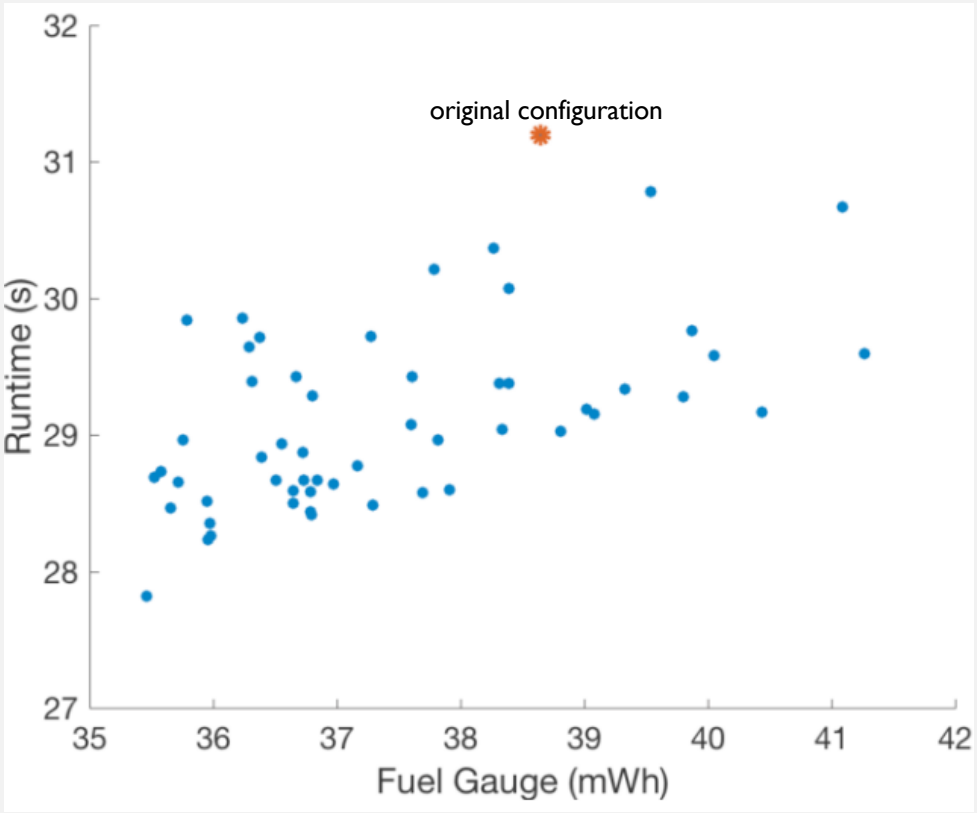


RESULT



Orange: original configuration

RESULT



Summary:

- 1 year of coding and learning
- super "smart" OS and Android tools
- internal sensor: super noisy in short evaluations, pretty good for longer evaluations