MDROID:
An Energy-Aware Mutation Testing Framework for Android

Seminar: Software Testing

Outline

1) Current Problem

2) Basics
   - Mutation Testing
   - Android Activity Lifecycle

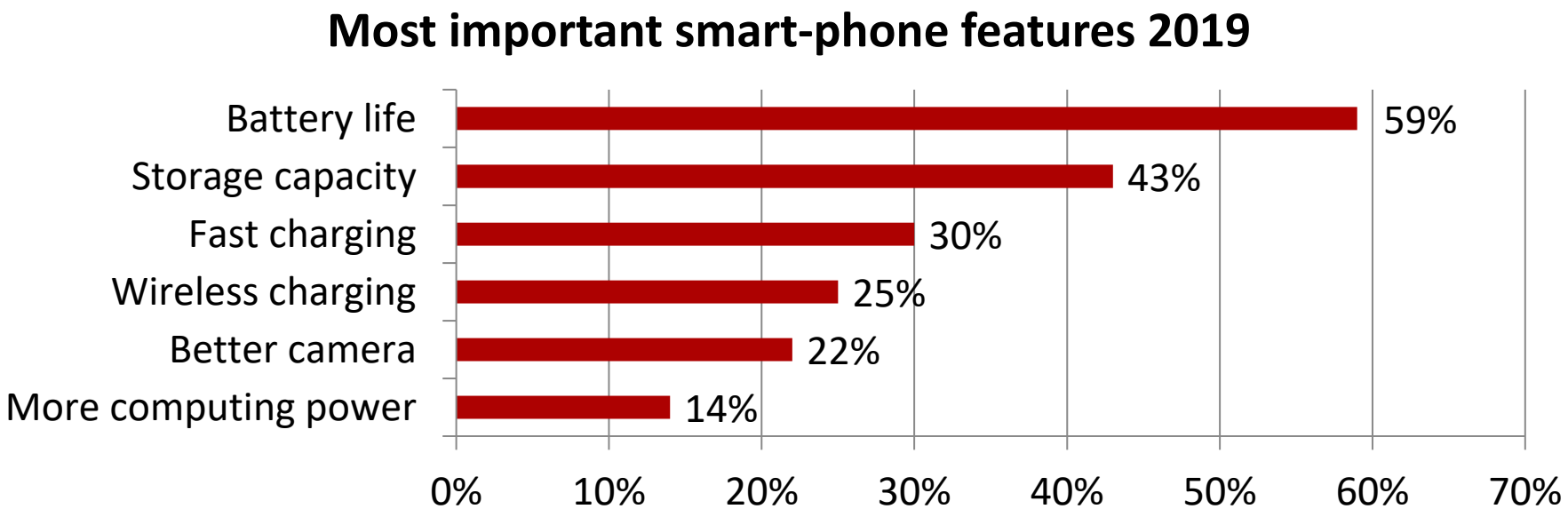
3) Framework Overview

4) Components of Framework

5) Evaluation of µDROID

Current Problem

- Rising popularity of mobile apps
  - Energy-efficiency major concern
- No tools for energy-testing
  - Functional correctness test used

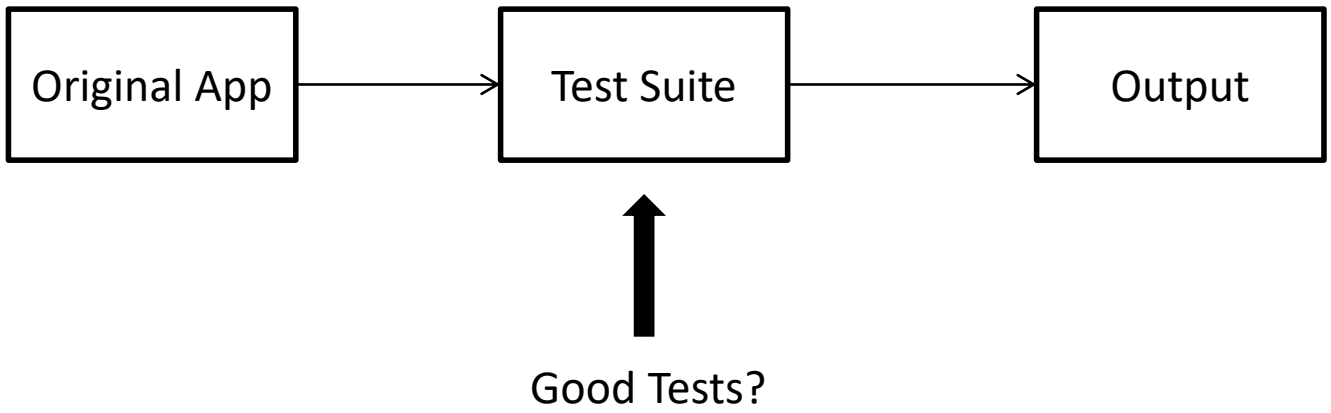


Source: statista.com 15.11.2019

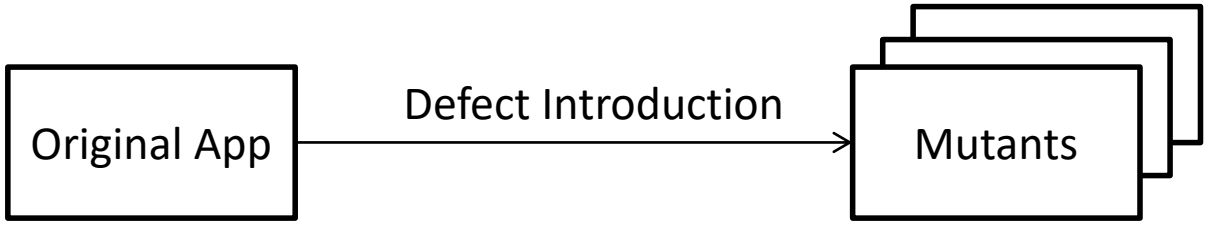
Solution: µDROID

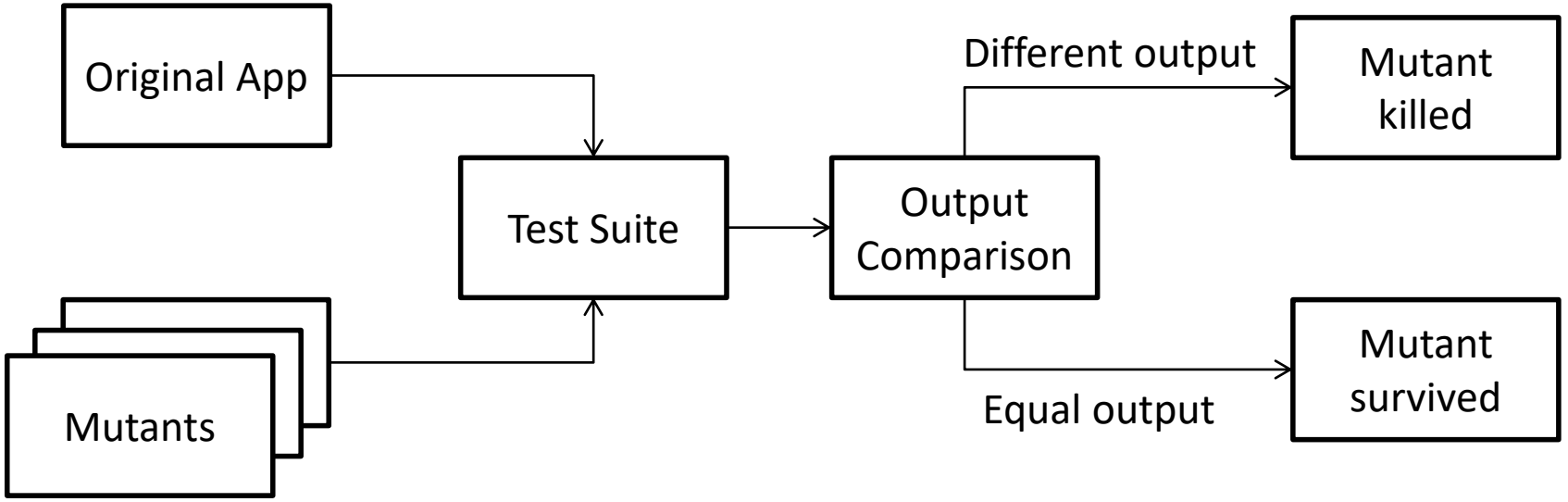
- Testing framework for Android
- Evaluates your test suite
  - Revealing energy-bugs
- Automatic application

Mutation Testing

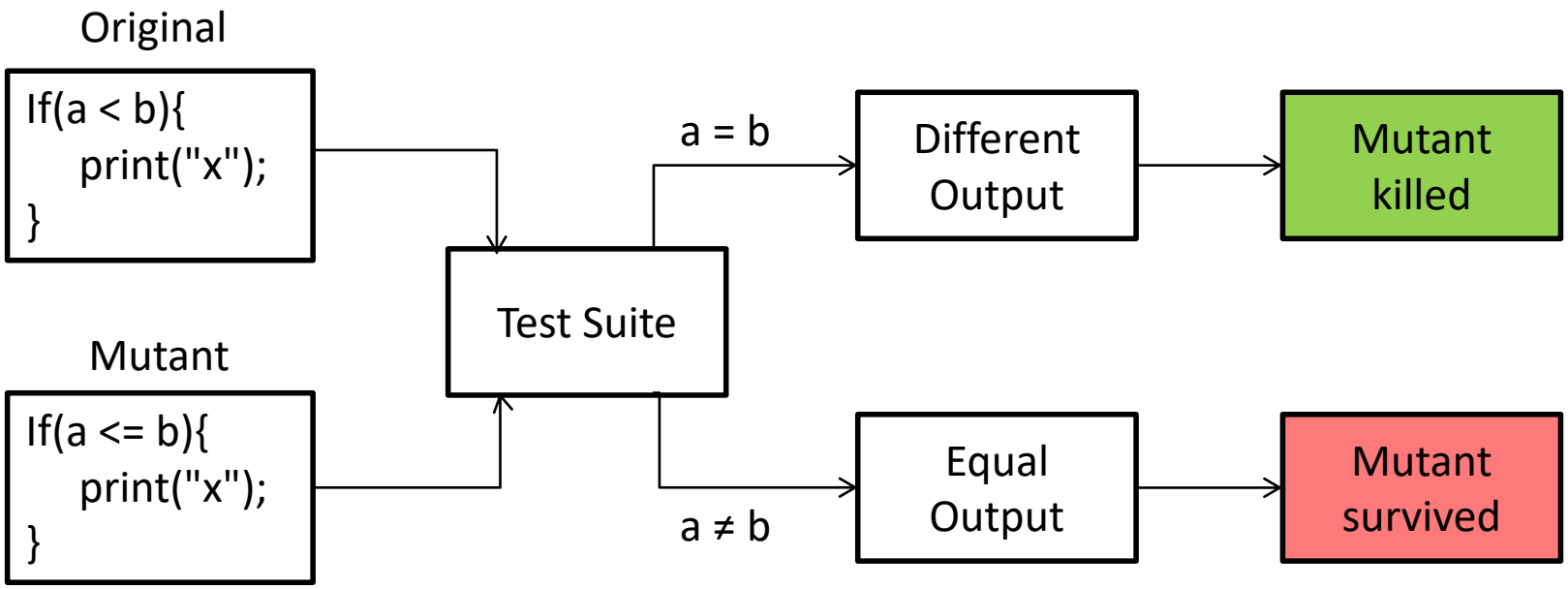


Mutation Testing

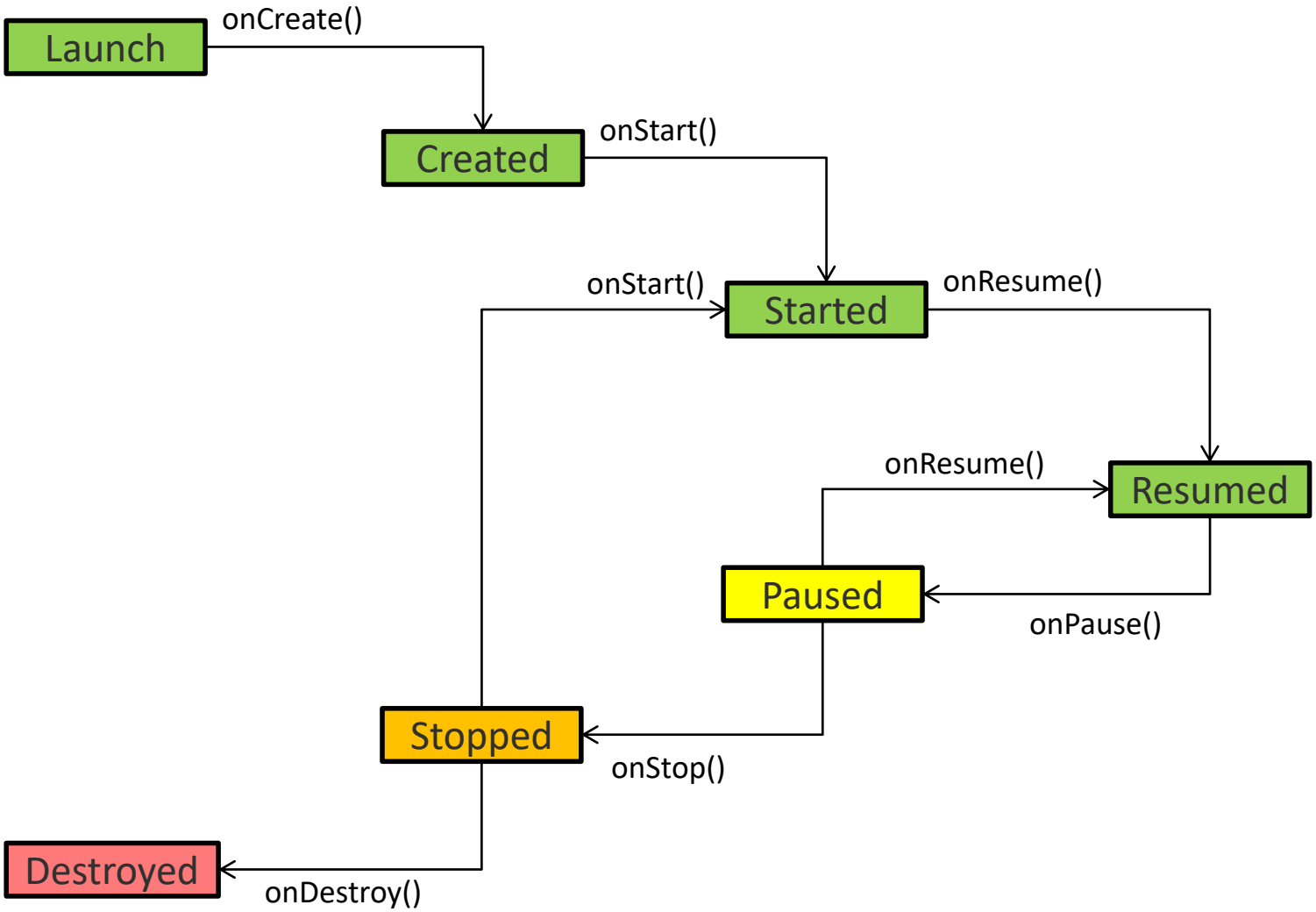




Mutation Testing

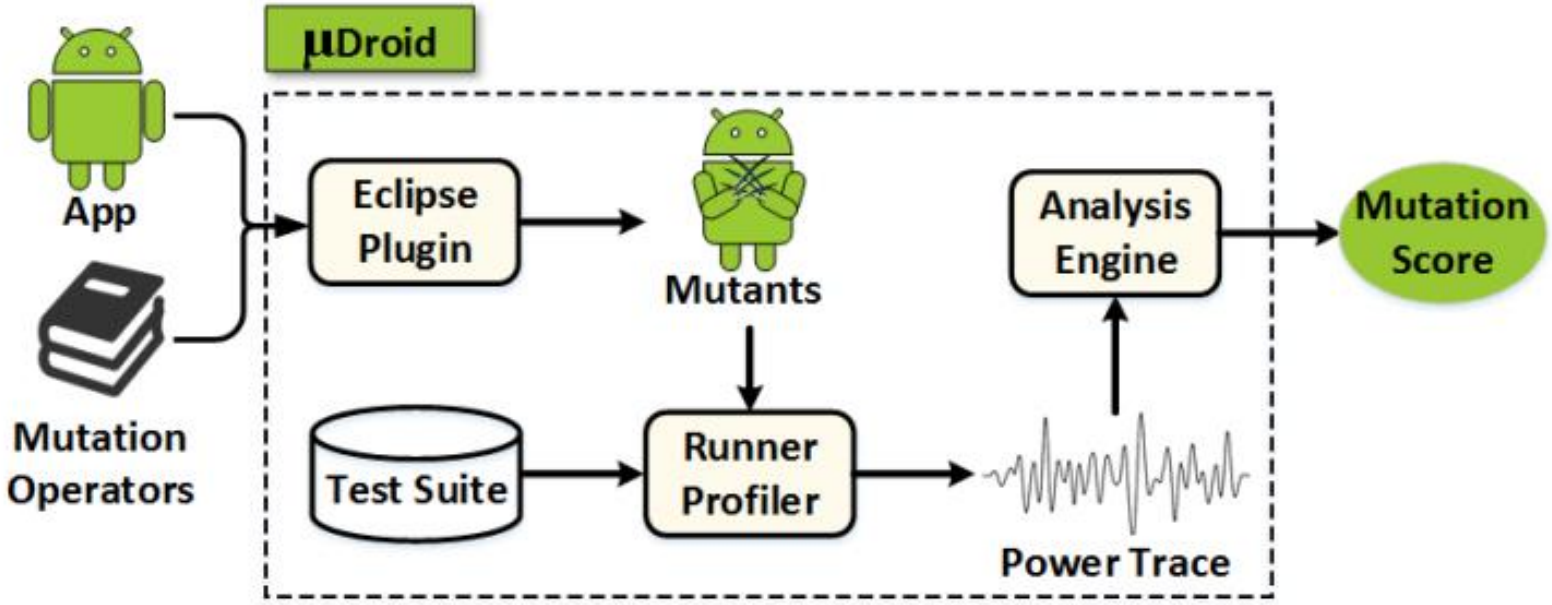


Android Activity Lifecycle



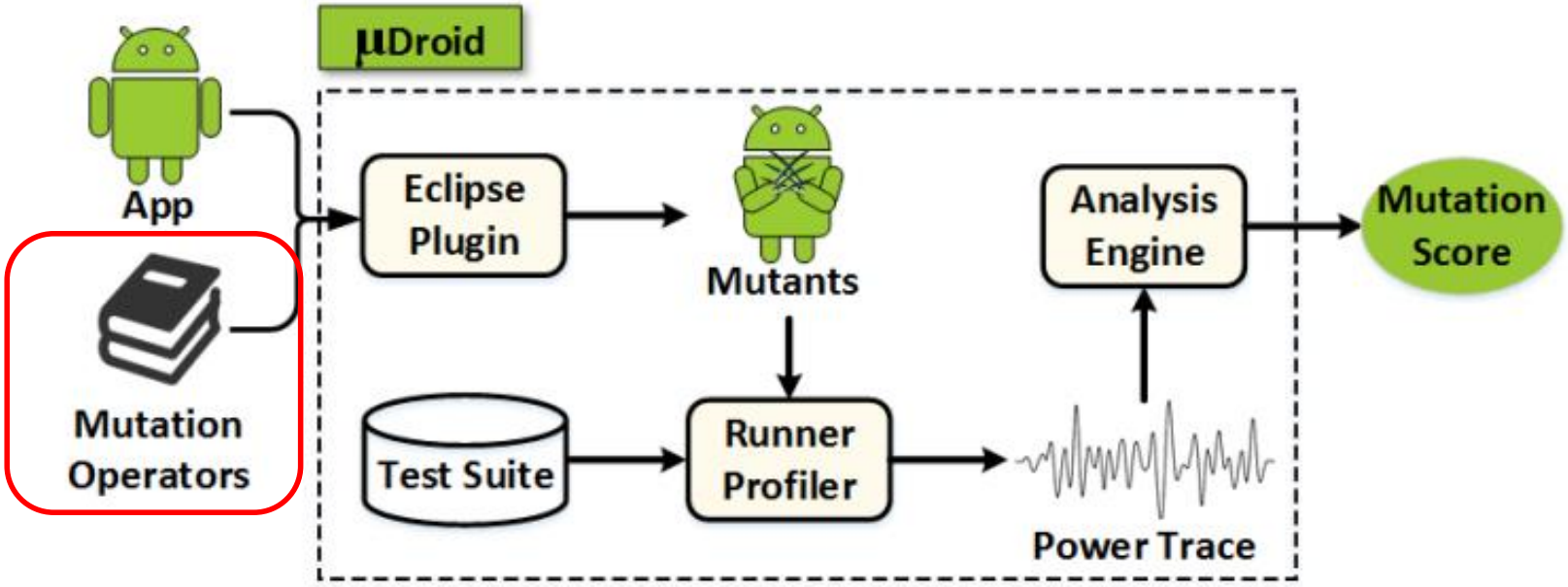
Based on: https://developer.android.com/guide/components/images/activity_lifecycle.png 19.11.2019

Framework Overview



Source: Jabbarvand, R., & Malek, S. (2017, August). µDroid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM.
Figure 1

Overview



Source: Jabbarvand, R., & Malek, S. (2017, August). µDroid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM.
Figure 1

Mutation Operators

1) Connectivity Mutation Operators
2) Location Mutation Operators
3) Wakelock Mutation Operators
4) Display Mutation Operators
5) Recurring Callback and Loop Mutation Operators
6) Sensor Mutation Operators

1) Connectivity Mutation Operators

- Network-related

- Mobile data power usage > WiFi power usage

Original App Code

```
1 WifiManager manager = getSystemService("WIFI_SERVICE");
2 if(manager.isWifiEnabled()){
3     // download big file
4 }
```

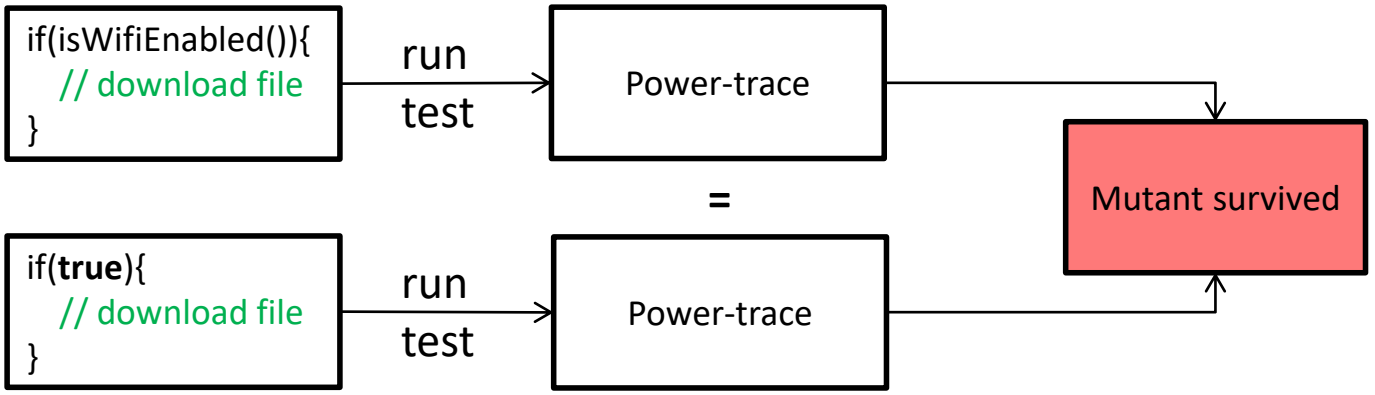
Mutant: remove check for WiFi

```
1 WifiManager manager = getSystemService("WIFI_SERVICE");
2 if(true){
3     // download big file
4 }
```

Based on: Jabbarvand, R., & Malek, S. (2017, August). µDroid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM. Figure 3

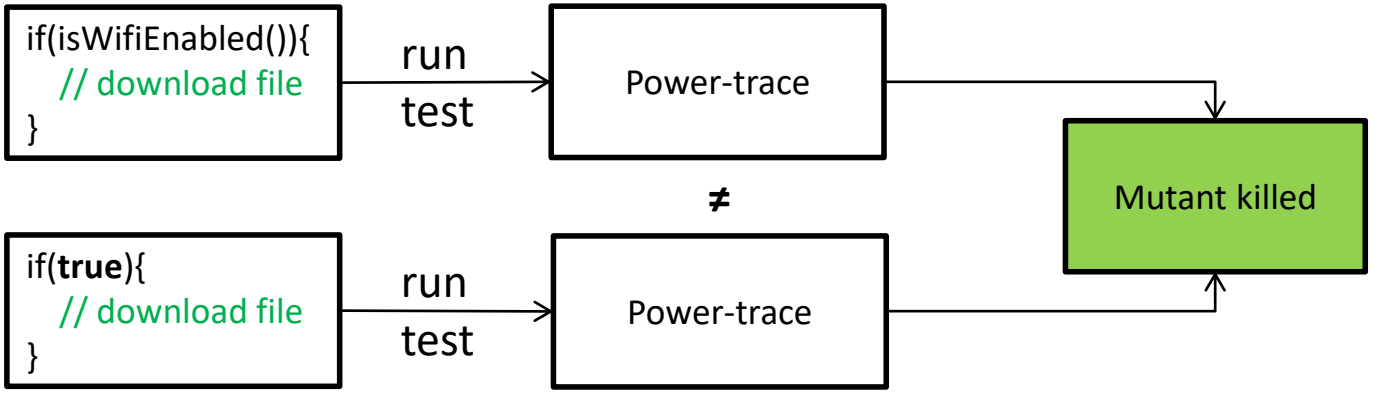
1) Connectivity Mutation Operators

WiFi always enabled



WiFi disabled

Mobile data enabled



2) Location Mutation Operators

Original App Code

```
1 public class TrackActivity extends Activity{
2     private LocationManager manager;
3     private LocationListener listener;
4     protected void onCreate(){
5         manager = getSystemService("LOCATION_SERVICE");
6         listener = new LocationListener(){
7             public void onLocationChanged(){
8                 // process location information
9             }
10        };
11        manager.requestLocationUpdates("NETWORK", 2*60*1000, 20, listener);
12    }
13    protected void onDestroy(){
14        super.onDestroy();
15        manager.removeUpdates(listener);
16    }
17 }
```

Based on: Jabbarvand, R., & Malek, S. (2017, August). µDroid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM. Figure 2

2) Location Mutation Operators

Mutant: increase update frequency

```
1  public class TrackActivity extends Activity{
2      private LocationManager manager;
3      private LocationListener listener;
4      protected void onCreate(){
5          manager = getSystemService("LOCATION_SERVICE");
6          listener = new LocationListener(){
7              public void onLocationChanged(){
8                  // process location information
9              }
10         };
11         manager.requestLocationUpdates("NETWORK", 2*60*1000, 1, listener);
12     }
13     protected void onDestroy(){
14         super.onDestroy();
15         manager.removeUpdates(listener);
16     }
17 }
```

Based on: Jabbarvand, R., & Malek, S. (2017, August). µDroid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM. Figure 2

2) Location Mutation Operators

Mutant: remove listener unregistration

```
1 public class TrackActivity extends Activity{
2     private LocationManager manager;
3     private LocationListener listener;
4     protected void onCreate(){
5         manager = getSystemService("LOCATION_SERVICE");
6         listener = new LocationListener(){
7             public void onLocationChanged(){
8                 // process location information
9             }
10        };
11        manager.requestLocationUpdates("NETWORK", 2*60*1000, 20, listener);
12    }
13    protected void onDestroy(){
14        super.onDestroy();
15    }
16 }
```

Based on: Jabbarvand, R., & Malek, S. (2017, August). µDroid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM. Figure 2

3) Wakelock Mutation Operators

- App keeps device awake
- CPU, Sensors, WiFi, etc. stays on
- Needs explicit release
- Mutation Operators delete release

4) Display Mutation Operators

- Most power-consuming component

Original App Code

```
1 Settings.System.putInt(getContentResolver(),
2     "SCREEN_OFF_TIMEOUT", 1*60*1000);
```

Mutant: increase display timeout

```
1 Settings.System.putInt(getContentResolver(),
2     "SCREEN_OFF_TIMEOUT", Integer.MAX_VALUE);
```

Based on: Jabbarvand, R., & Malek, S. (2017, August). µDroid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM.

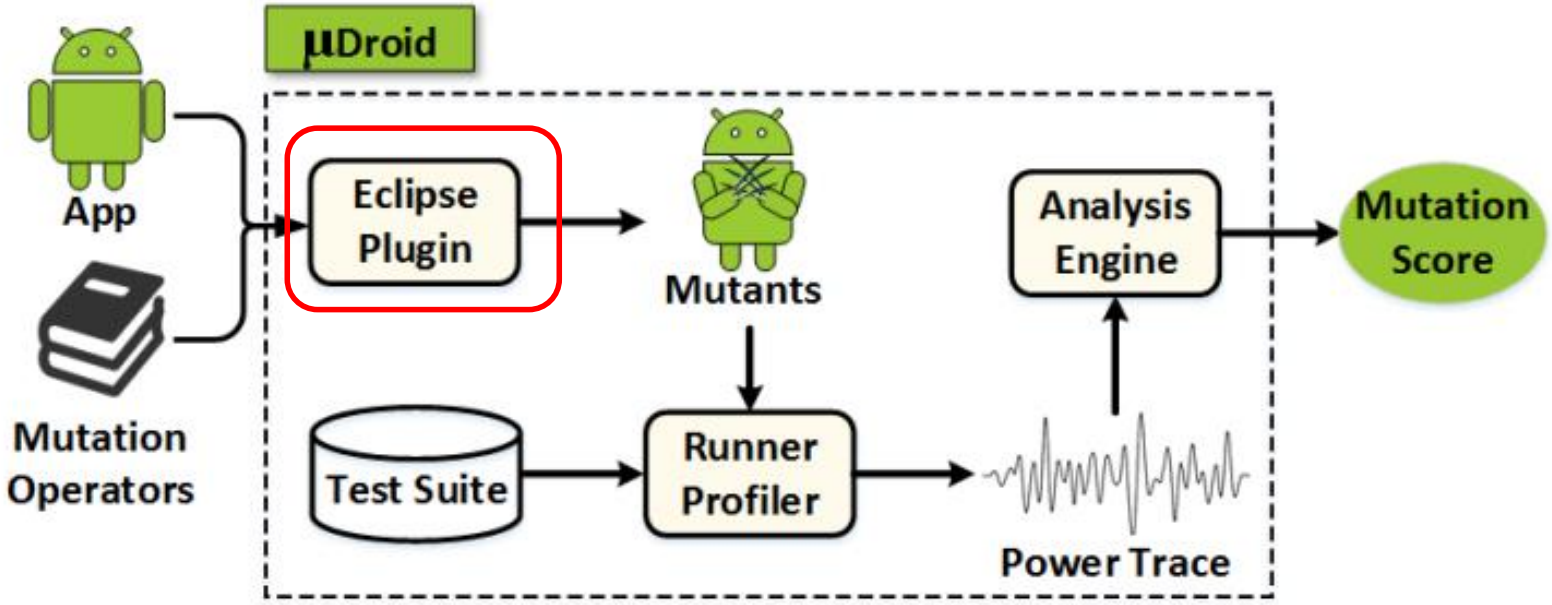
5) Recurring Callback and Loop Mutation Operators

- Repeating tasks
- Frequency relates to power level
- Mutant Operators remove frequency decrease
- Mock battery level tests

6) Sensor Mutation Operators

- Like Location Mutation Operator
- 1. Mutation Operator increases scan-frequency
- 2. Mutation Operator deletes release

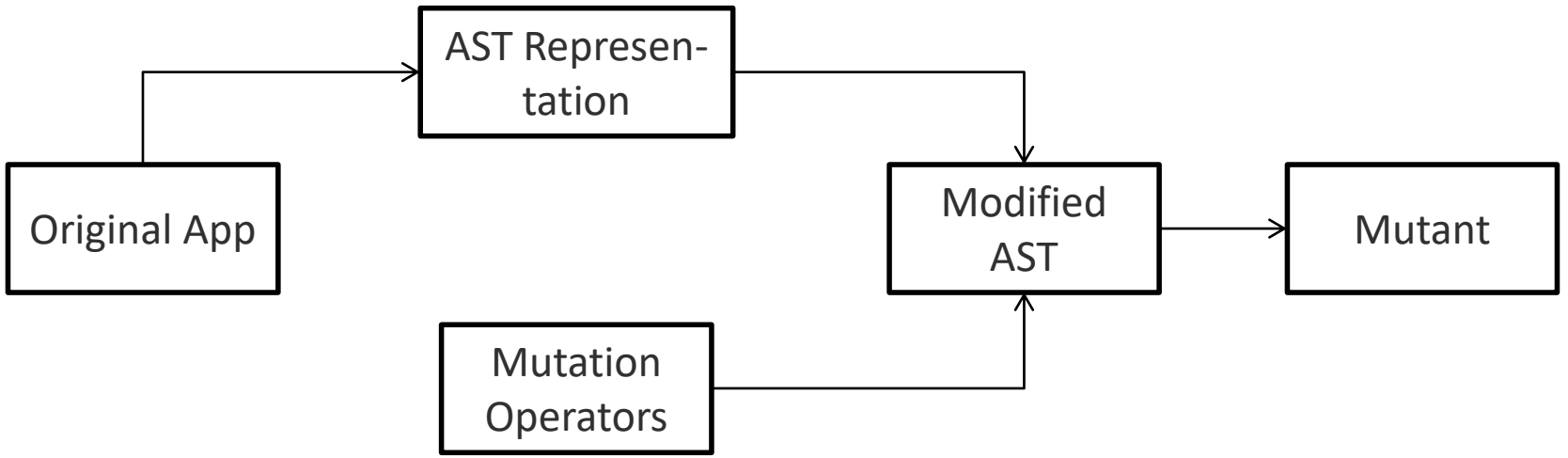
Overview



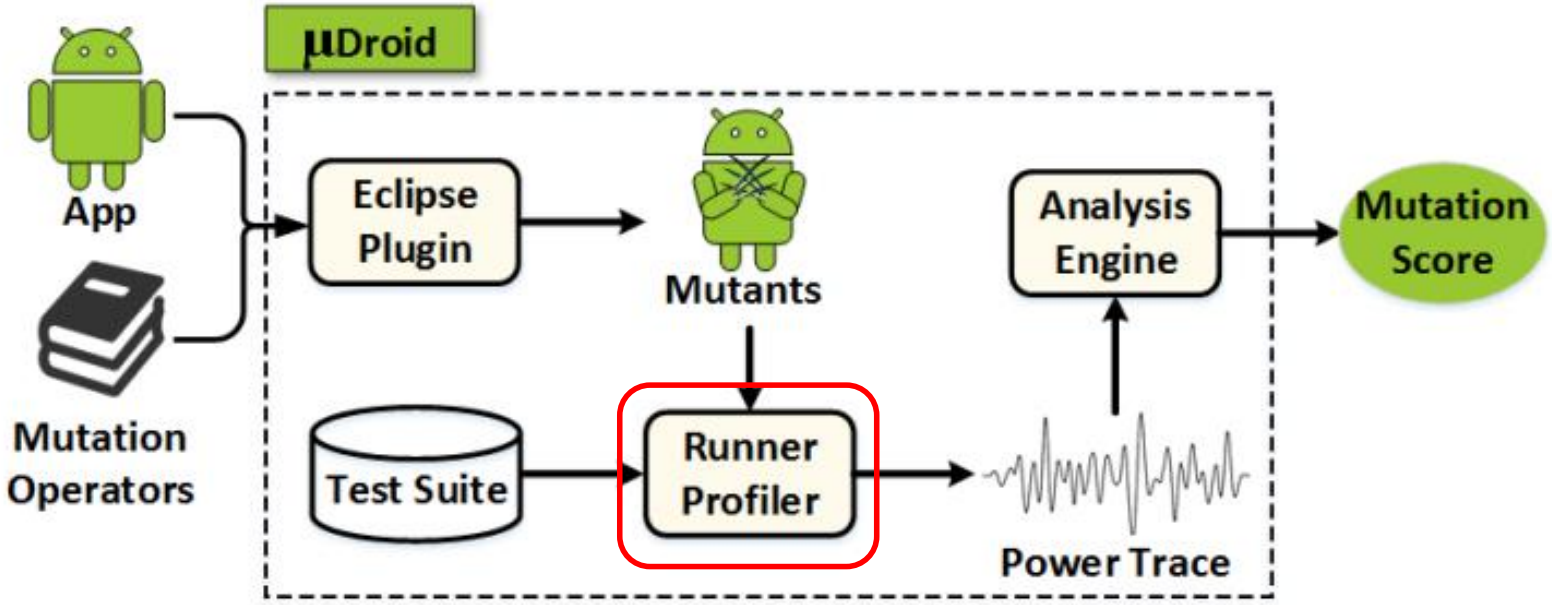
Source: Jabbarvand, R., & Malek, S. (2017, August). µDroid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM.
Figure 1

Eclipse Plugin

- Mutant generation:
  1. Create Abstract Syntax Tree (AST)
  2. Introduce Faults from Anti-Patterns
  3. Generate implementation from AST



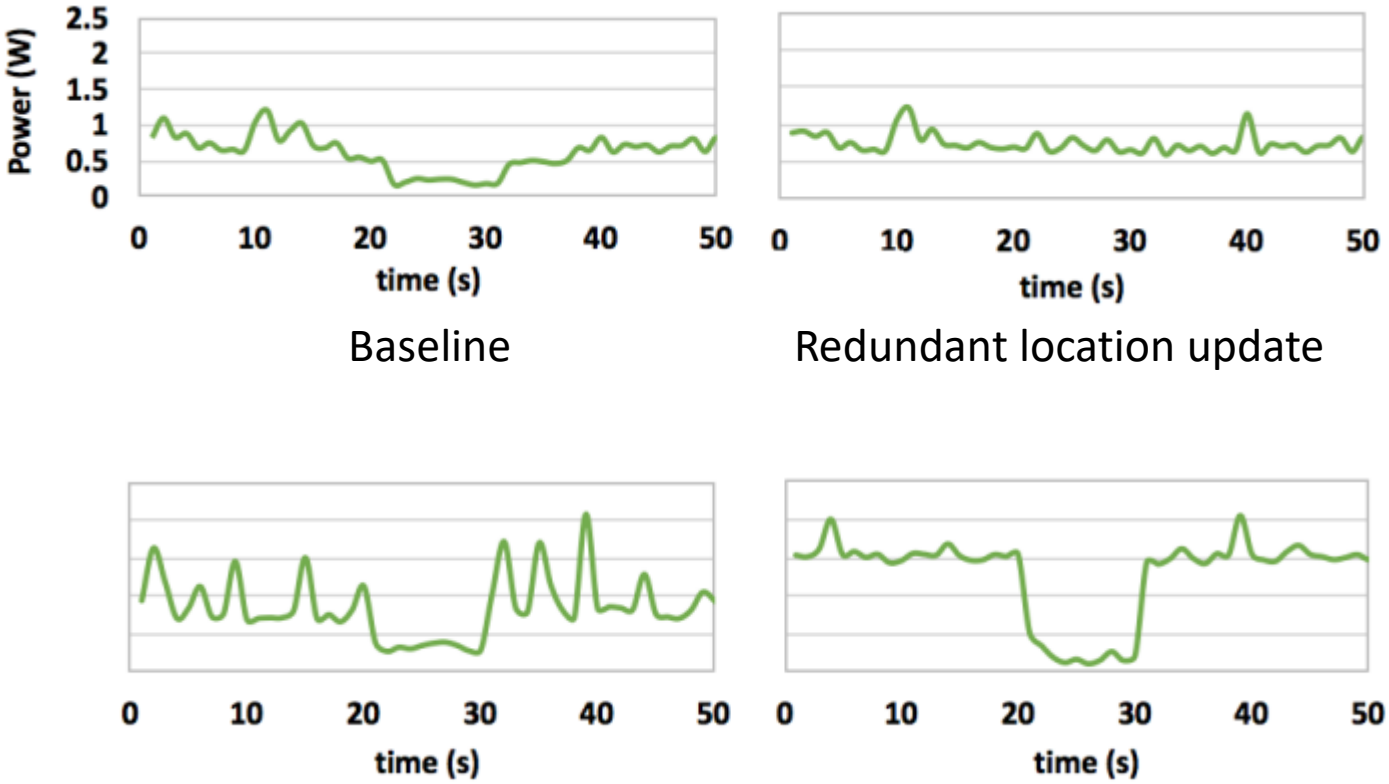
Overview



Source: Jabbarvand, R., & Malek, S. (2017, August). µDroid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM.
Figure 1

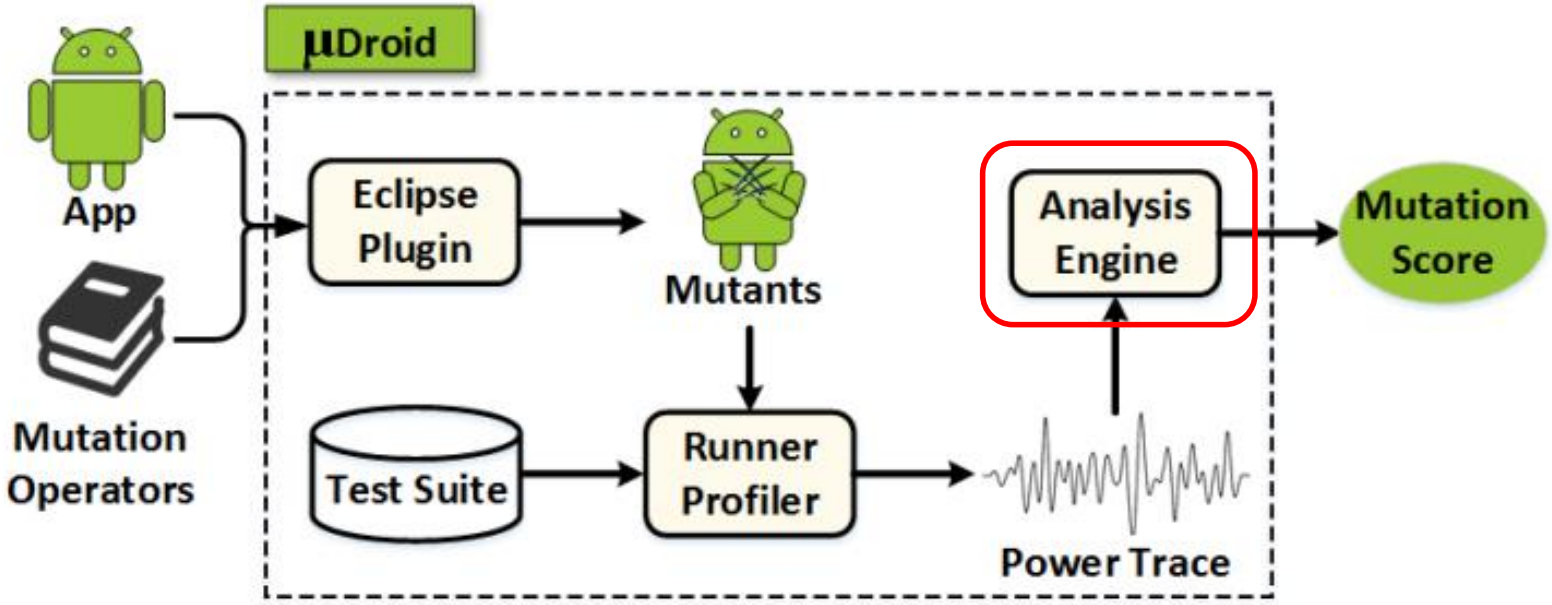
Power Traces

- Power traces of Sensorium App



Source: Jabbarvand, R., & Malek, S. (2017, August). µDroid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM. Figure 6

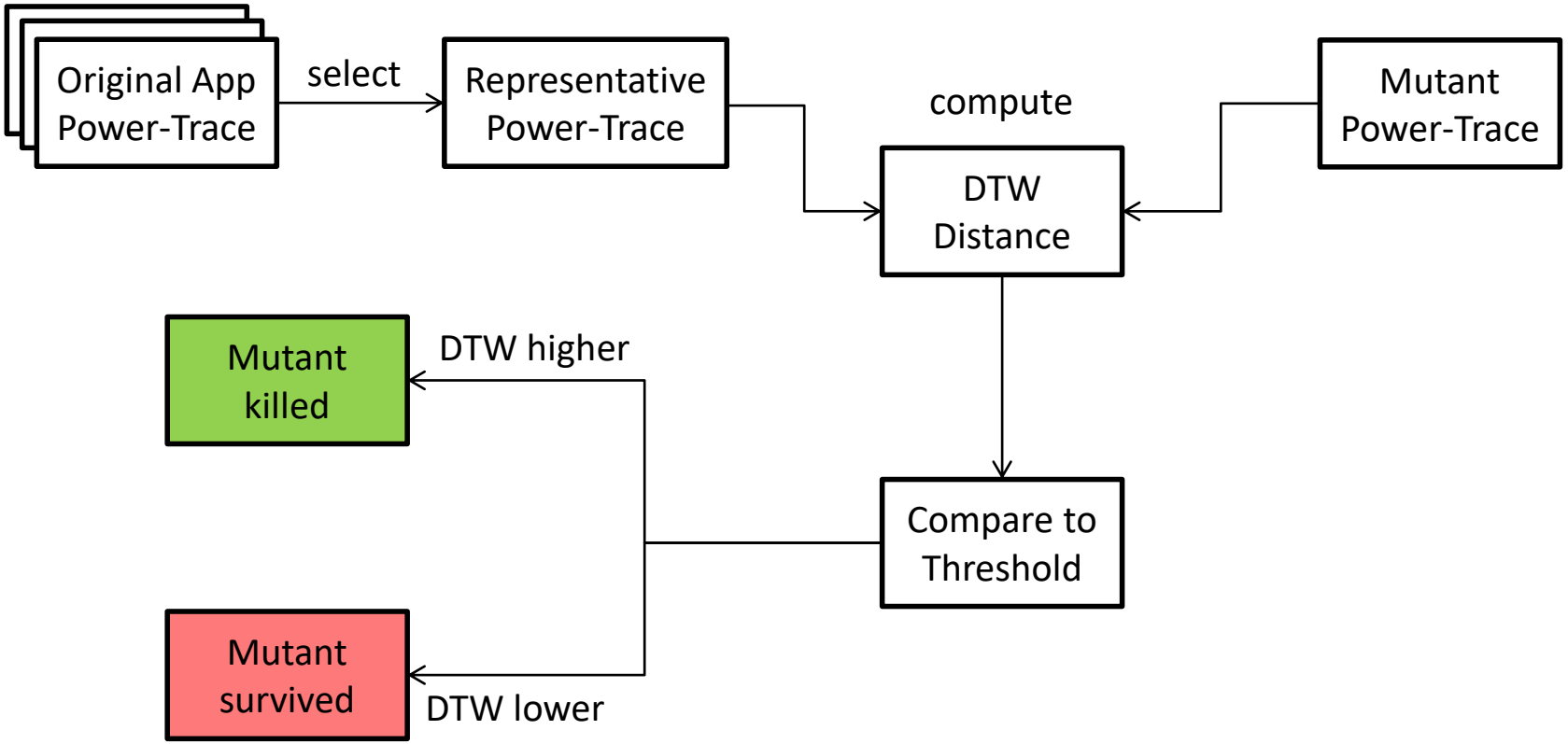
Overview



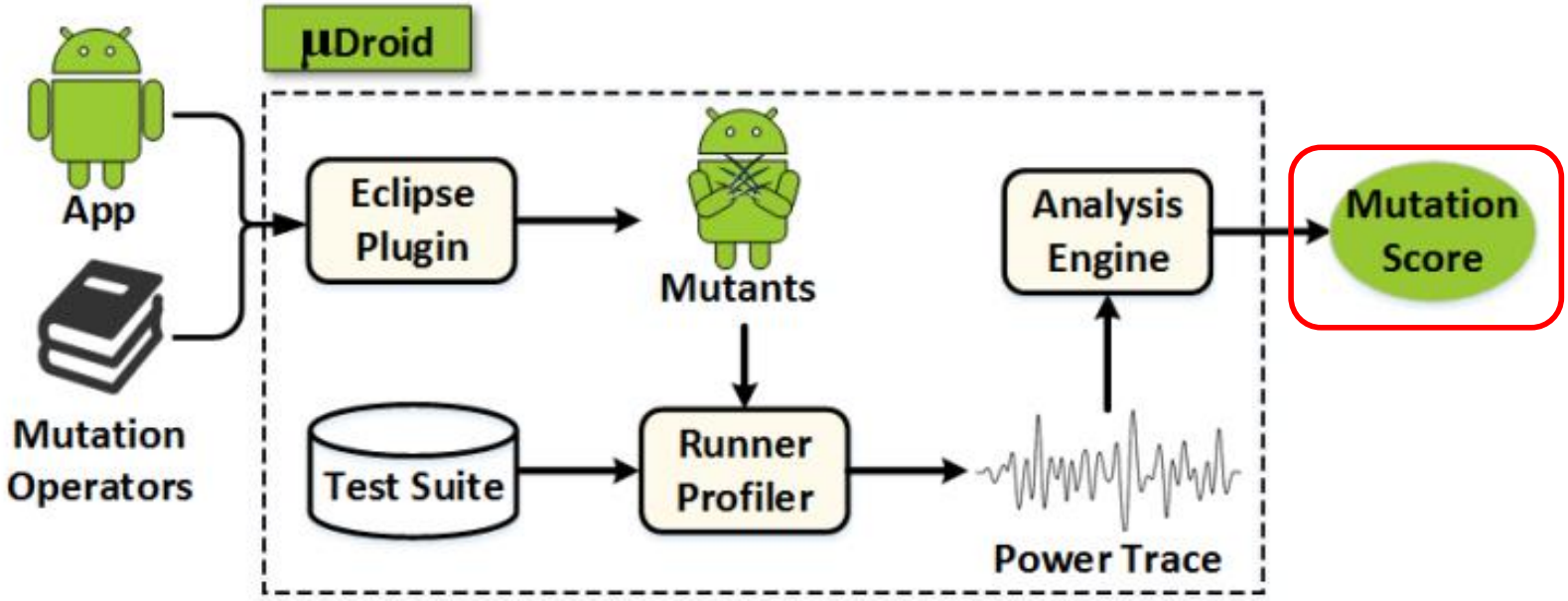
Source: Jabbarvand, R., & Malek, S. (2017, August). µDroid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM. Figure 1

Analysis Engine

For each test in test suite:



Overview



Source: Jabbarvand, R., & Malek, S. (2017, August). µDroid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM. Figure 1

Mutation Score

$$Mutation\ Score = \frac{\#\ Killed\ Mutants}{\#\ Mutants}$$

Evaluation of µDROID

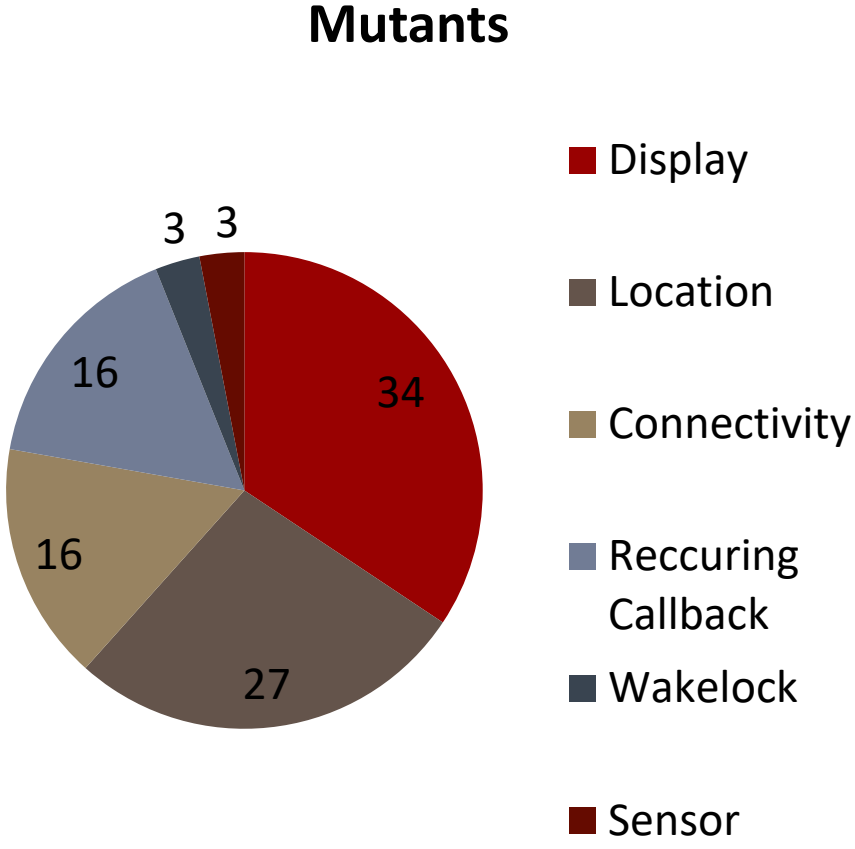
- RQ1) Prevalence, Quality and Contribution
- RQ2) Effectiveness
- RQ3) Association to real faults
- RQ4) Accuracy of Oracle
- RQ5) Performance

Experimental Setup

- 100 randomly selected Apps
  - Containing known energy-bug
- **Hardware:**
  - Android 6.0.1
  - Google Nexus 6
- **Measurement:**
  - Trepn (Qualcomm)
- **Test Cases:**
  - Hand-written tests
  - Random tests by Android Monkey

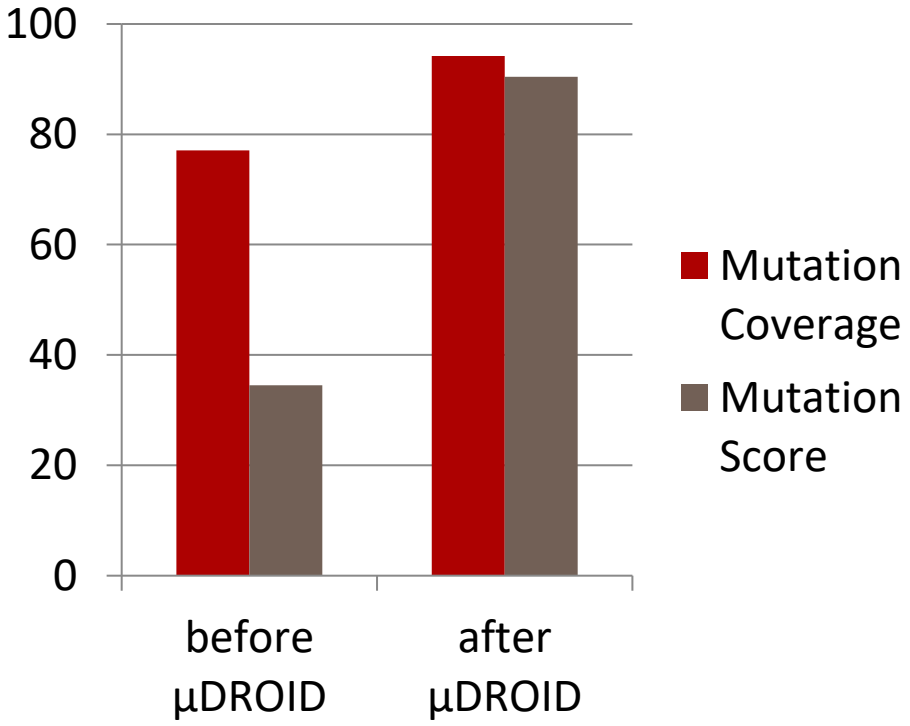
RQ1) Prevalence, Quality and Contribution

- 28 Mutants on average
- Real need in Domain
- Every Operator useful



RQ2) Effectiveness

- Tests created by developers
- Tests had many deficiencies
- Example missing test cases:
  - Poor WiFi signal
  - Different battery-levels
  - Mocking locations
  - Long tests (file download)



RQ3) Association to real faults

- High mutation score = high uncovering rate of energy bugs?

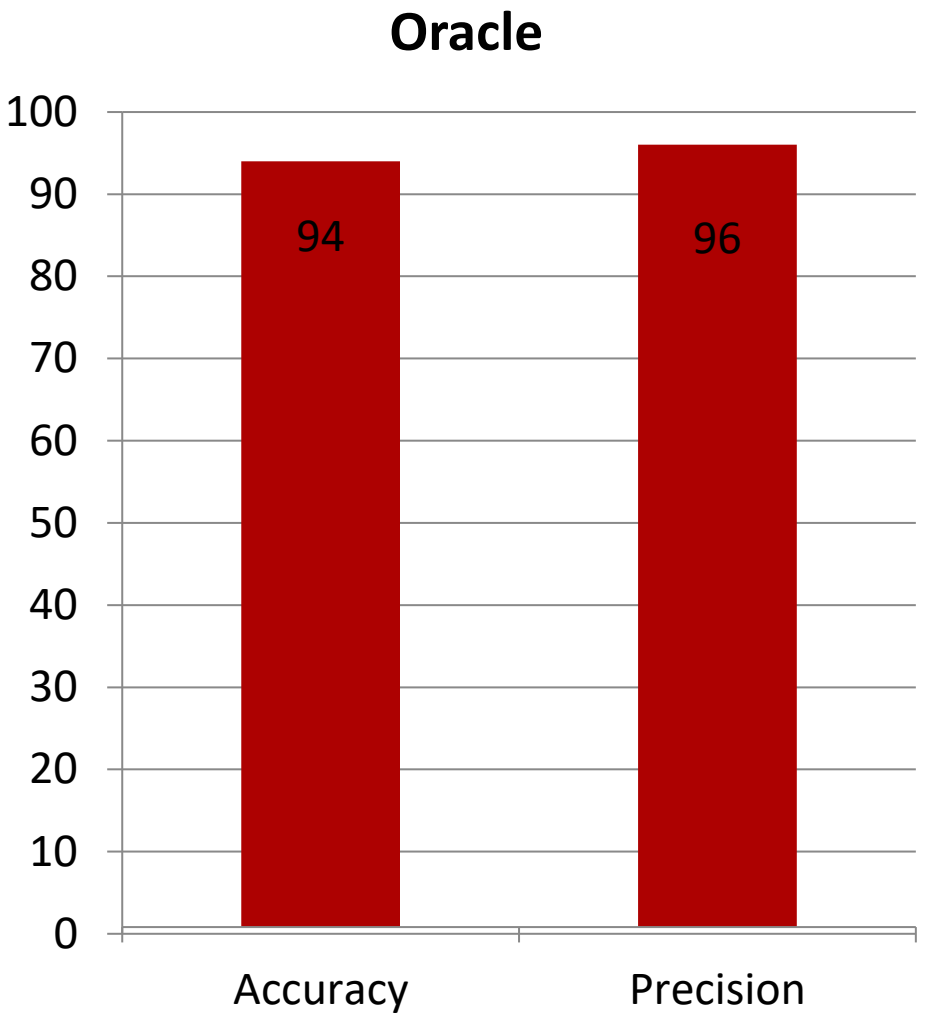
- Tests on energy-bug containing app
  - 20 Test suites
  - 20 Test cases

- Test suites that uncovered real bug:

  77% higher Mutation Score

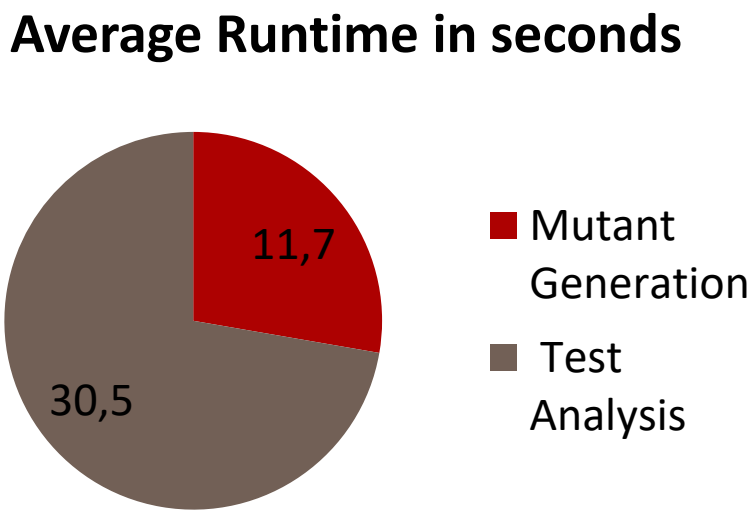
RQ4) Accuracy of Oracle

- Acceptable accuracy for use in practice



RQ5) Performance

- **Test System:**
  - Intel I7 2.2 GHZ
  - 16GB DDR3 RAM
- 0.4s per Mutant
- 0,83s per Test Analysis



Summary

- **µDROID:**
  - Mutation testing framework for android
  - Shows energy-related deficits in tests
- **Workflow of µDROID:**
  - Mutation Operators
  - Mutant generation
  - Running tests
  - Analyzation
- **Evaluation:**
  - Good results for use in practice

Future work: COBWEB

- From same authors
- Search-based energy-testing framework
  - For Android
- Evolutionary Algorithm
- Automatically generated tests

Time for questions!

Backup slides

Sources

- Jabbarvand, R., & Malek, S. (2017, August). µDroid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM.
- Jabbarvand, R., Lin, J. W., & Malek, S. (2019, May). Search-based energy testing of Android. In *Proceedings of the 41st International Conference on Software Engineering* (pp. 1119-1130). IEEE Press.
- Jabbarvand, R., & Malek, S. (2017, May). Advancing energy testing of mobile applications. In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)* (pp. 491-492). IEEE.
- Jia, Y., & Harman, M. (2010). An analysis and survey of the development of mutation testing. *IEEE transactions on software engineering*, *37*(5), 649-678.
- https://developer.android.com/guide/components/activities/activity-lifecycle Access-date: 19.11.2019
- https://de.statista.com/statistik/daten/studie/180389/umfrage/gruende-fuer-kaufentscheidung-von-smartphones/ Access-date: 15.11.2019

Advancing Energy Testing of Mobile Applications

- Previous work of authors

- Three stepped approach
  1. Energy-Aware Test Generation
  2. Energy-Aware Test-Suite Adequacy Assessment
  3. Energy-Aware Test-Suite Minimization

- µDROID = 2. step

- Started collection of Energy Anti-Patterns
  - Mutation Operators based on

1) Connectivity Mutation Operators

- Bluetooth-related

- Discovery = Heavyweight Task

Original App Code

```
1  public void discover(int scan_interval){
2      BluetoothAdapter blue = BluetoothAdapter.getDefaultAdapter();
3      private Runnable discovery = new Runnable(){
4          public void run(){
5              blue.startDiscovery();
6              handler.postDelayed(this, scan_interval);
7          }
8      };
9      // code for processing bluetooth data
10     handler.removeCallbacks(blue);
```

Based on: Jabbarvand, R., & Malek, S. (2017, August). µDroid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM. Figure 4

1) Connectivity Mutation Operators

Mutant: increase scan frequency

```
1 public void discover(int scan_interval){
2     BluetoothAdapter blue = BluetoothAdapter.getDefaultAdapter();
3     private Runnable discovery = new Runnable(){
4         public void run(){
5             blue.startDiscovery();
6             handler.postDelayed(this, 0);
7         }
8     };
9     // code for processing bluetooth data
10    handler.removeCallbacks(blue);
```

Mutant: delete removeCallback call

```
1 public void discover(int scan_interval){
2     BluetoothAdapter blue = BluetoothAdapter.getDefaultAdapter();
3     private Runnable discovery = new Runnable(){
4         public void run(){
5             blue.startDiscovery();
6             handler.postDelayed(this, scan_interval);
7         }
8     };
9     // code for processing bluetooth data
10
11 }
```

Based on: Jabbarvand, R., & Malek, S. (2017, August). µDroid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM. Figure 4