

# How to Validate Mobile Crowdsourcing Design? Leveraging Data Integration in Prototype Testing.

---

Chu Luo, Miikka Kuutila, Simon Klakegg, Denzil Ferreira,  
Huber Flores, Jorge Goncalves, Vassilis Kostakos, Mika Mäntylä

Center for Ubiquitous Computing & M3S  
University of Oulu, Finland  
<http://ubicomp.oulu.fi/>



# Mobile Crowdsourcing

- Mostly on smartphone



# Context: Understand Workers



# Processing Mobile Context



# Testing is Challenging



# Related Work

- Testing Tools
- Record and Replay Tools

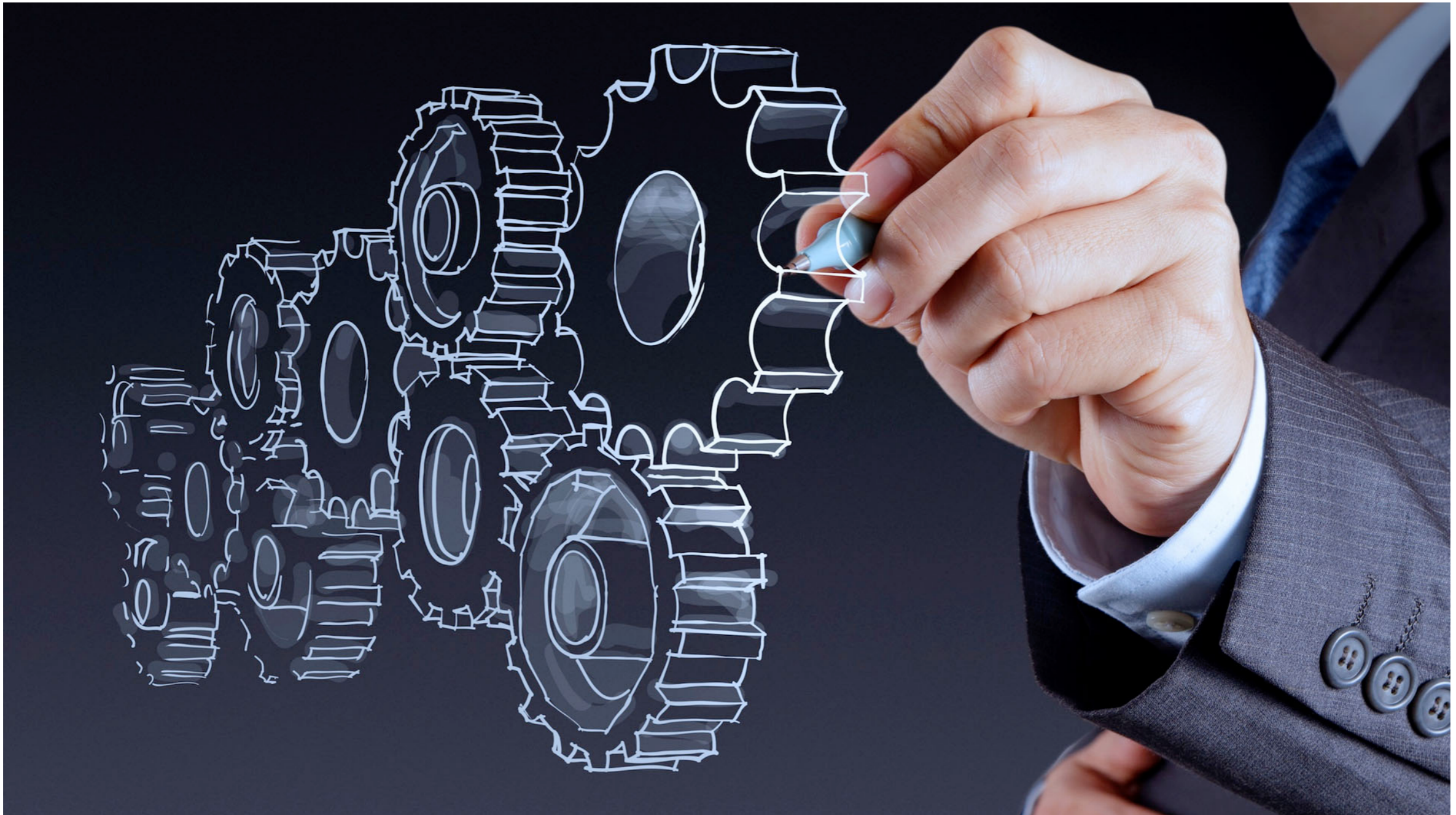


# Challenges

- Participant Recruitment.
- Timeliness of Historical Data
- Uncommon Context
- Device Heterogeneity

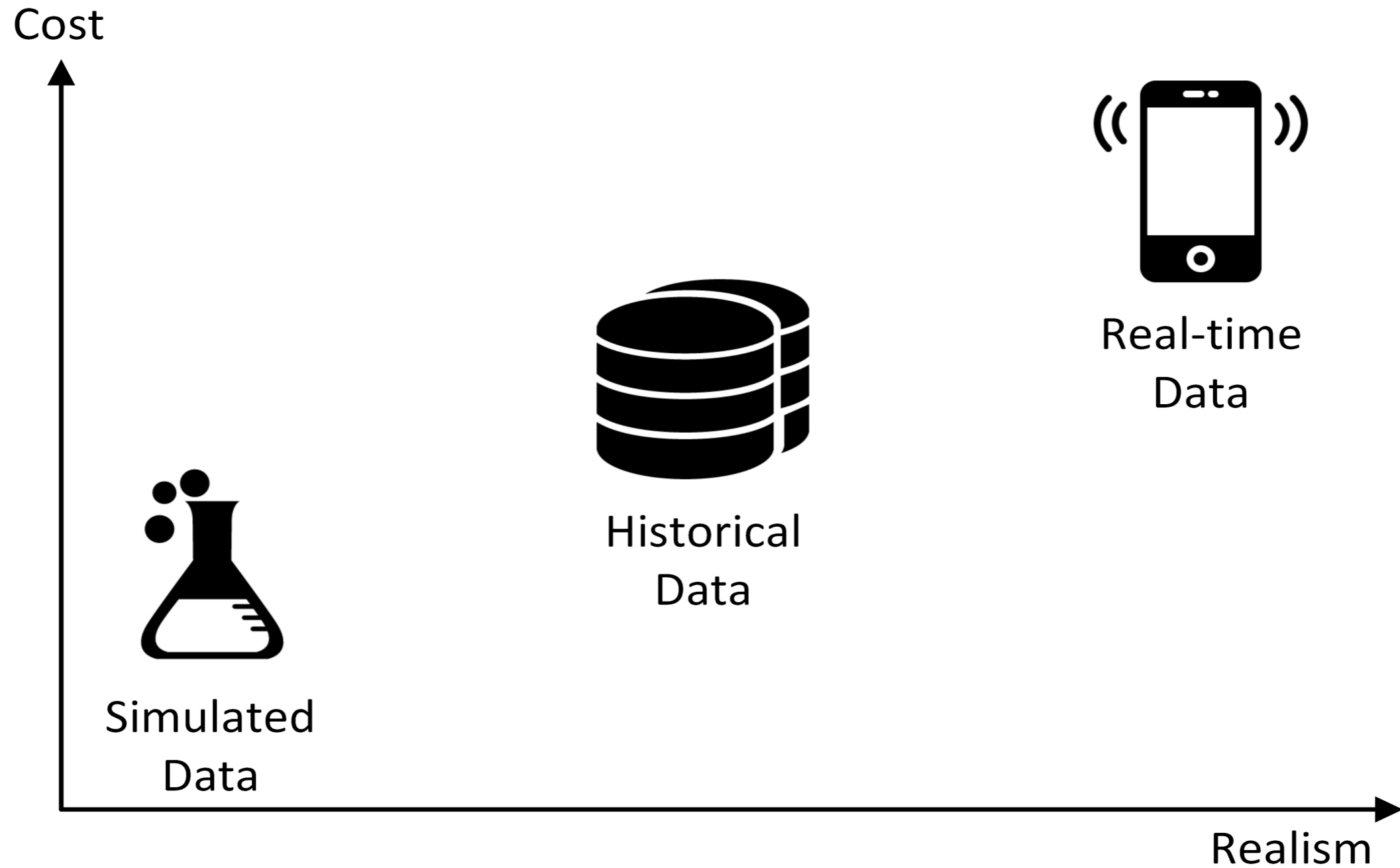


# Data Makes Mobile Crowdsourcing Testable

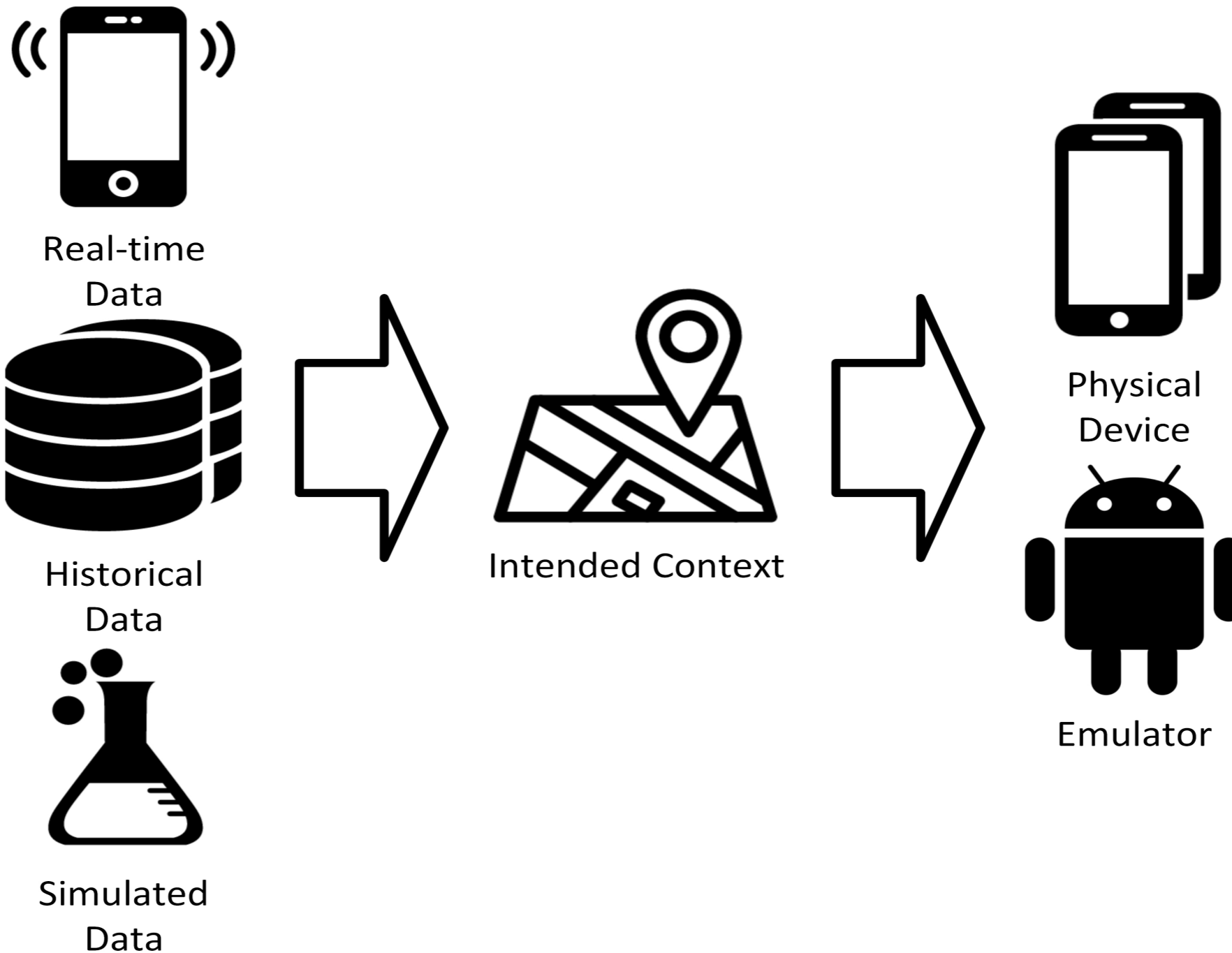




# Comparison of Data Sources



# Our Work: Data Integration



# Real-time Data: AWARE

<http://www.awareframework.com/>



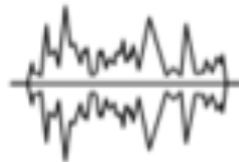
Hardware



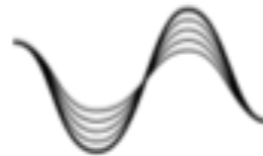
Software



Human



Data



Analysis



AWARE  
plugins



AWARE  
user studies



AWARE  
applications



Abstraction



Validation



Reusability



# Historical Data

## Context Simulator

<https://bitbucket.org/sbobek/context-simulator>



# Simulated Data

## Use the way we fake data

```
public void createApplicationsCrashes(long timestamp, String device_id, String package_name,
                                     String application_name, long application_version,
                                     String error_short, String error_long,
                                     int error_condition, int is_system_app) {
    ContentValues data = new ContentValues();
    data.put(Applications_Crashes.TIMESTAMP, timestamp);
    data.put(Applications_Crashes.DEVICE_ID, device_id);
    data.put(Applications_Crashes.PACKAGE_NAME, package_name);
    data.put(Applications_Crashes.APPLICATION_NAME, application_name);
    data.put(Applications_Crashes.APPLICATION_VERSION, application_version);
    data.put(Applications_Crashes.ERROR_SHORT, error_short);
    data.put(Applications_Crashes.ERROR_LONG, error_long);
    data.put(Applications_Crashes.ERROR_CONDITION, error_condition);
    data.put(Applications_Crashes.IS_SYSTEM_APP, is_system_app);
    Intent applications_crashes = new Intent();
    applications_crashes.setAction(ACTION_AWARE_PLUGIN_DUMMYAWARE);
    applications_crashes.putExtra(EXTRA_DATA, data);
    sendBroadcast(applications_crashes);
    getContentResolver().insert(Applications_Crashes.CONTENT_URI, data);
    Log.d("DUMMAYAWARE", "138");
}
```



# Black Box + White Box Testing



# Deliver the Best App to Workers with the Help of Testing



# Thank You! Questions?



**Chu Luo <chu.luo@oulu.fi>**

**Center for Ubiquitous Computing, University of Oulu, Finland**

**<http://ubicomp.oulu.fi/>**

