

Watt's Happening!!



Alexis Fisher

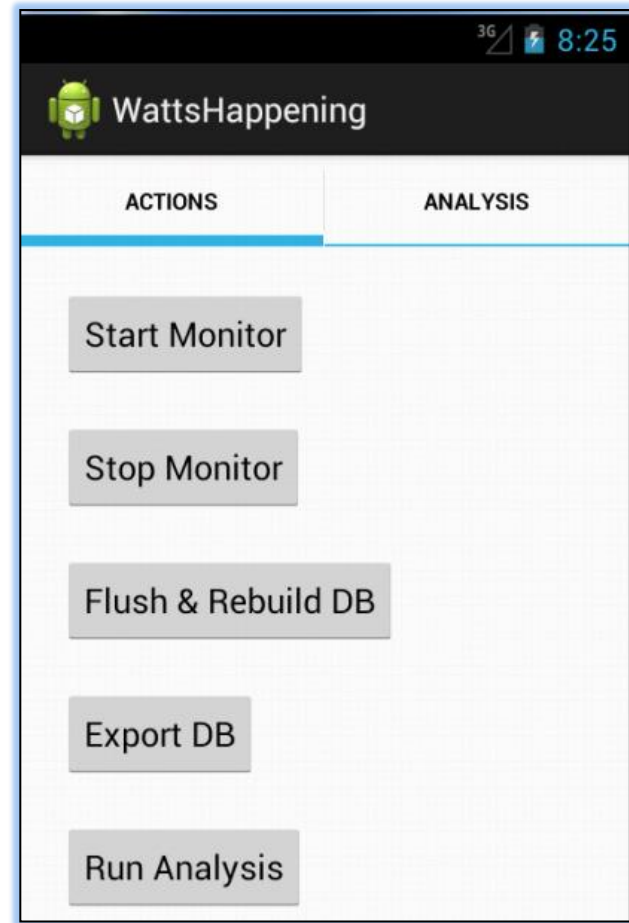
Ben Bramble

Adam Vail

Nick Burek

Overview

- Motivation
- Logging
- Analyzing
- Recommending
- Future Work
- Questions



Logging

- Structure of Code
- Android uses SQLITE
- How to access tables
- How often logging (alarm ensures constant logging every x minutes)
- Example tables (What data and how to get it)

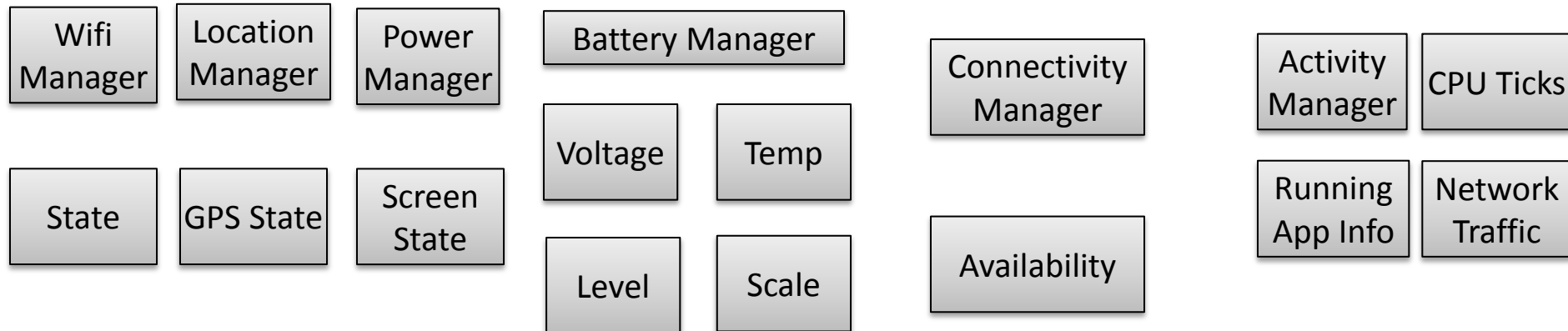
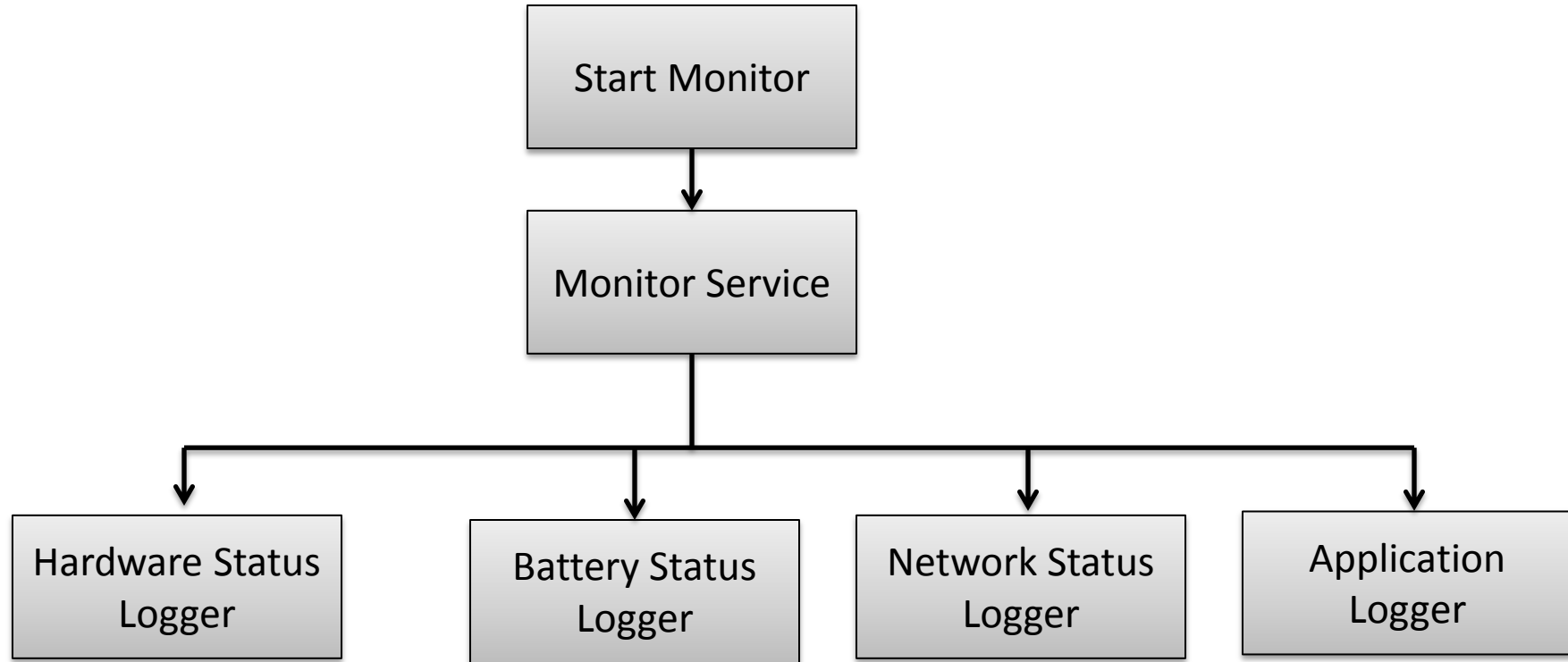
Motivation



Logging Decisions

- Pull vs continuously monitoring
- Alarm system
- Time between pulls
- Passive pull (GPS, bluetooth)
- Short running apps (see paper)

Code Structure



ID	Timeslice_ID	Voltage	Temp	Percentage
1	1	4144.0	220.0	0.9
2	2	4131.0	220.0	0.9

**Battery
Table**

ID	Timeslice_ID	Name	Enabled	Status
1101	276	Bluetooth	0	, Scanmode: None
1102	276	WIFI	0	Disabled

**Hardware
Table**

ID	Timeslice_ID	Name	State	Connection
3320	277	Ethernet	IDLE	IsAvailable: 1 IsConnected: 0
3321	277	Mobile_supl	IDLE	IsAvailable: 1 IsConnected: 0

**Network
Table**

ID	Timeslice_ID	Name	App_ID	CPU	RX_Bytes	TX_Bytes
13053	277	Maps	10008	481.0	4933.0	2120.0
13054	277	Contacts	10005	310.0	0.0	0.0

App Table

ID	App_UID	Hist_CPU	Hist_Net	Hist_HW	#_of_Updates
210	10207	0.0	0.0	0.0	120
211	10221	0.0	19908.0	0.0	98

**Aggregate
App Table**

We have to estimate time remaining before bat dies b/c goal

2 issues: over and under est

Overestimation: too much time

Under est: dies sooner

Short term = (only % point granularity) why 5 minutes (only logging every 5 minutes, don't want to go shorter, haven't tried longer) Long time = miss too many apps, shorter = not yield significantly more information

Long term = minimal usage or idle

We make this calc by looking at bat level over time

Naïve way: look at change over time (graphs with slopes) leads to over estimation

We must look at most recent usage: currently using it = good chance you will continue to use

Assumption short term will continue

If we have data in short term window, use solely as basis in prediction for a conservative estimate

ELSE, no short term data so we fall back on long term (past usage = future usage)

Analyzing Application Usage

- How do we know which applications are more intensive?
- Rank based on hardware usage
 - CPU usage
 - Network traffic usage

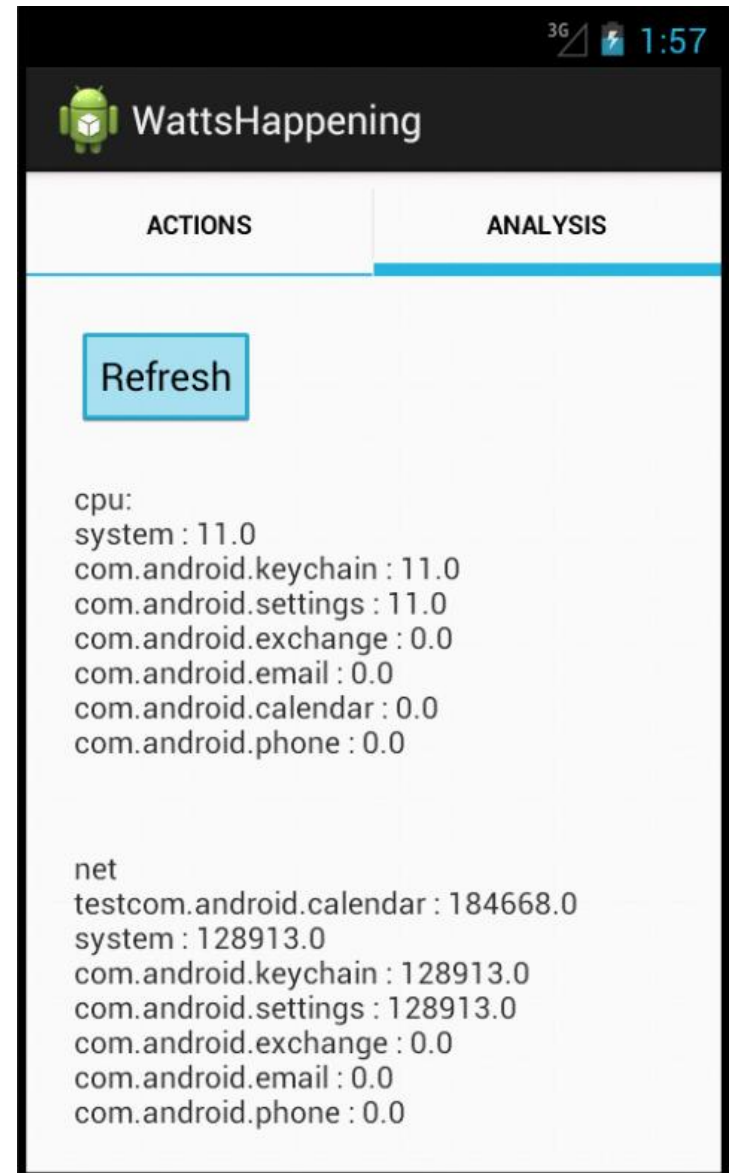


Our Metrics

- CPU Usage
 - CPU ticks used by application
 - CPU ticks used in total
 - Gives the % of CPU used by the application over that time
 - Track that average over long periodsc
- Network Usage
 - Number of RX and TX bytes
 - Which network is currently active
 - Gives the data transfer rates over time
 - Possible to correlate this with which network is active

Showing High Usage Applications

- Rank by CPU usage
- Rank by Network Usage



Possible Application

- Power Model of Hardware
 - Could give us battery usage of an application
- Determine a user's historic usage model to recommend hardware settings
- Recommend which applications to kill to make it to a certain time without your battery dying

Questions?