09. Mobile Devices and Permissions Models

Blase Ur and Mainack Mondal

April 23rd, 2018

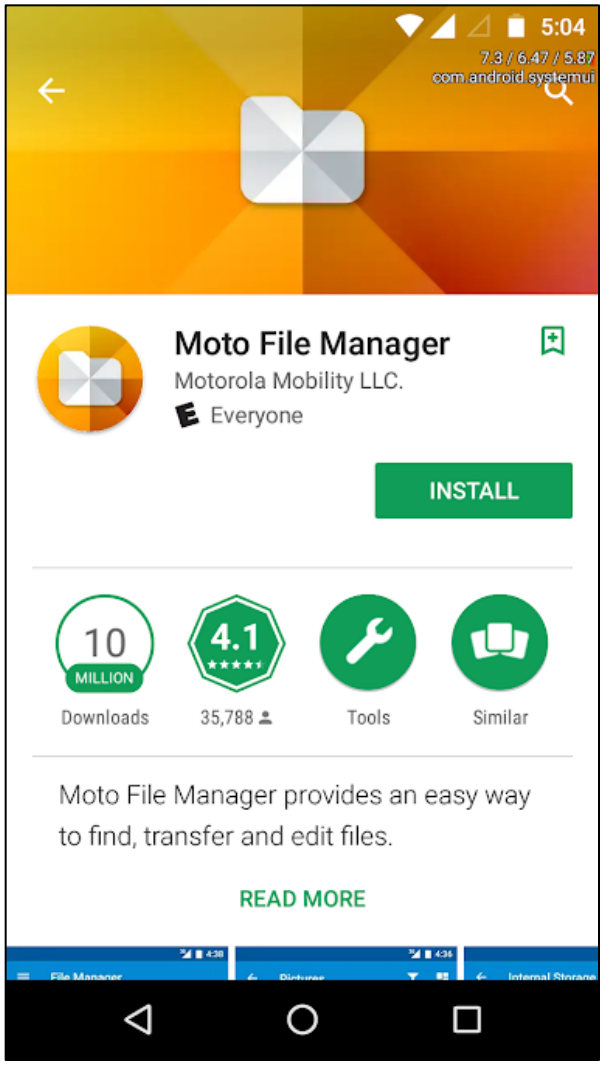
CMSC 23210 / 33210

THE UNIVERSITY OF CHICAGO

Security, Usability, & Privacy Education & Research

Mobile devices run apps

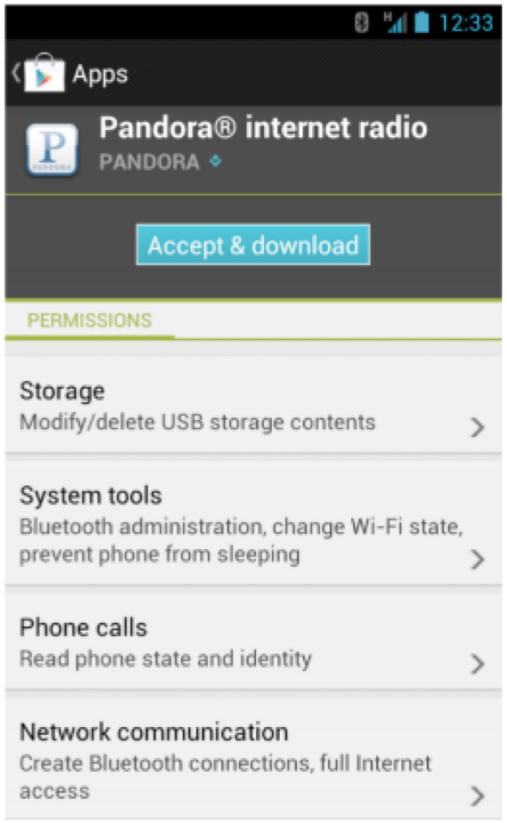
- Apps require permissions
- Permission
  - Android defined many permissions that developers can ask in their apps
  - Users have to grant these permissions

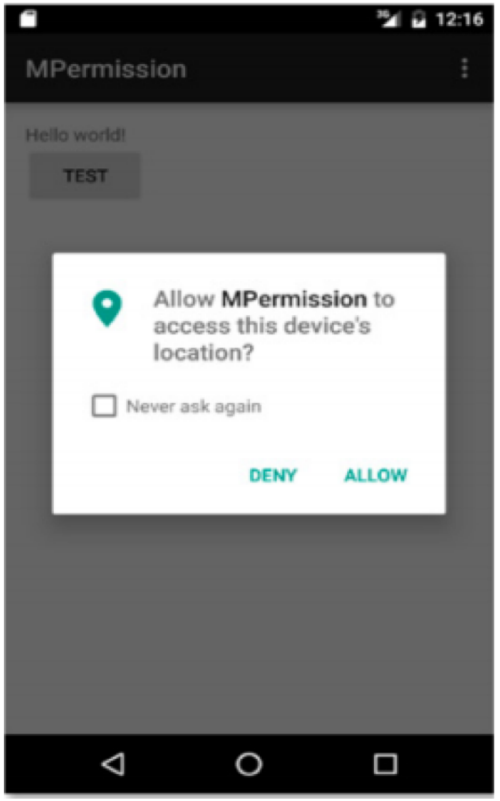


Protection level of permissions in Android

- Normal: Very little risk, default value
  - Automatically granted by playstore
- Dangerous: For high-risk protected operations
  - E.g., user's private data
  - Must be explicitly granted by the user
- Which is which:
  - read your wifi network information, use camera , set wallpaper, use bluetooth

When to ask permissions from users?





Prior to Android 6.0

All permissions were asked at **install-time**

Android 6.0

All permissions were asked at **run-time**

Understanding usability of permissions (Prior to Android 6.0)

- Quantitative survey [Felt et al. SOUPS‘12]
  - “The last time you downloaded an Android application, what did you look at before deciding to download it?”
  - 308 users

- Qualitative survey of 25 users [Felt et al. SOUPS‘12]
  - think-aloud experiment
  - semi-structured interviews

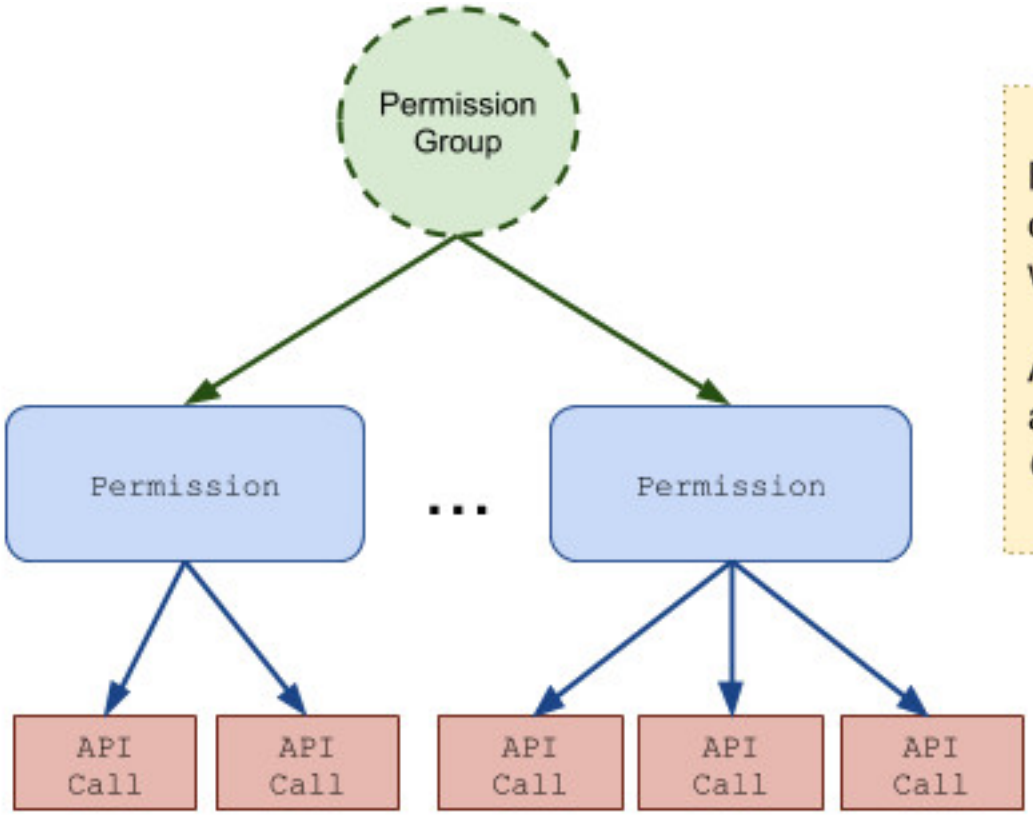
Usability of permissions : Problem

- Users were unaware
  - Too late in the installation process
- Users became habituated
  - If too many permission requests

- How about asking fewer permissions?
  - More coarse-grained permissions?
  - Permission groups are also allowed

    https://developer.android.com/guide/topics/permissions/overview.html

Permission groups in Android



Related API Calls and access to object properties are associated with a specific permission request.

And related permission requests are rolled up into a *Permissions Group*.

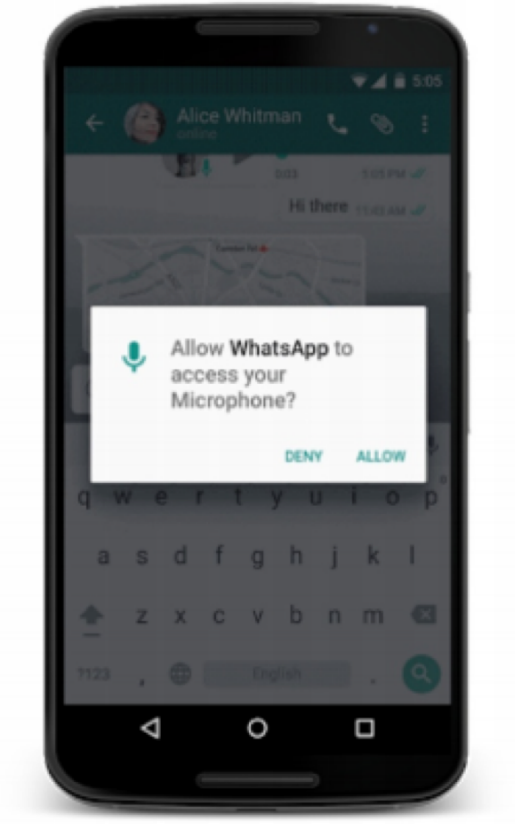
Access is granted to entire groups!

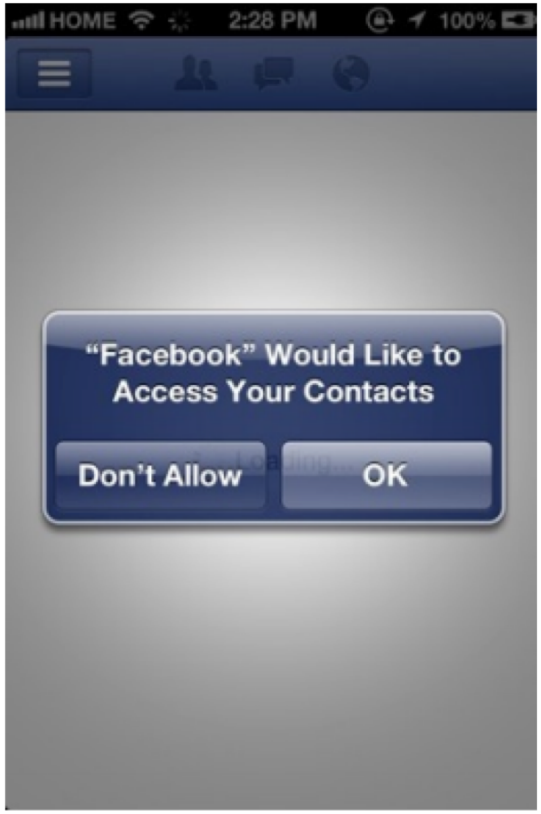
Permission groups

- In Group PHONE:
  - READ_PHONE_NUMBERS

- *Also* in Group PHONE:
  - CALL_PHONE: Call any phone
  - PROCESS_OUTGOING_CALLS: Allows an application to see the number being dialed during an outgoing call with the option to redirect the call to a different number or abort the call altogether ➔ *Bad idea!*

Does run-time permission model fix issues?





These prompts are missing **contexts**

How often apps access the resources they ask for? [Wijesekera et al., USENIX '15]

- Dynamic (run-time) analysis of app usage
  - modified Android OS
  - logged all API methods which involved sensitive data access
- 36 users' 6,048 hour Android usage data
  - 27m permission request from the apps
  - 213 requests per hour
  - location, read/write sms, browser history

Users want to block some permissions [Wijesekera et al., USENIX '15]

- Exit survey result
  - on average, users wanted to block 35% of all requests
  - However asking each time is infeasible

Lessons about user's Android permission usage

- Visibility of application requesting permission is a strong contextual cue for the users
  - invisible means users are quite likely to block
- Frequency at which requests occur makes it impractical to prompt user on every case
- ask-on-first-use on its own is insufficient and needs to be extended.

How to leverage context to provide more private permission model?

- Nissenbaum's privacy as contextual integrity
- Contextual information norms are modeled by
  - data subject (the user)
  - Sender
  - Receiver
  - Information type
  - Transmission principle (constraints)

Apply contextual integrity to make a better permission model

Don't prompt users when data flows are appropriate – prevent habituation

Prompt users when the data flows are unknown/inappropriate – aware users

Use Machine learning to predict desired preferences

Feature Group	Feature	Type
Behavioral Features (B)	Number of times a website is loaded to the Chrome browser.	Numerical
	Out of all visited websites, the proportion of HTTPS-secured websites.	Numerical
	The number of downloads through Chrome.	Numerical
	Proportion of websites requested location through Chrome.	Numerical
	Number of times PIN/Password was used to unlock the screen.	Numerical
	Amount of time spent unlocking the screen.	Numerical
	Proportion of times screen was timed out instead of pressing the lock button.	Numerical
	Frequency of audio calls.	Numerical
	Amount of time spent on audio calls.	Numerical
	Proportion of time spent on silent mode.	Numerical
Runtime Features (R1)	Application visibility (True/False)	Categorical
	Permission type	Categorical
	User ID	Categorical
	Time of day of permission request	Numerical
Aggregated Features	Average denial rate for (A1) application:permission:visibility	Numerical
	Average denial rate for (A2) application$_F$:permission:visibility	Numerical

How good are the ML algorithms?
[Wijesekera et al., OAKLAND'17]

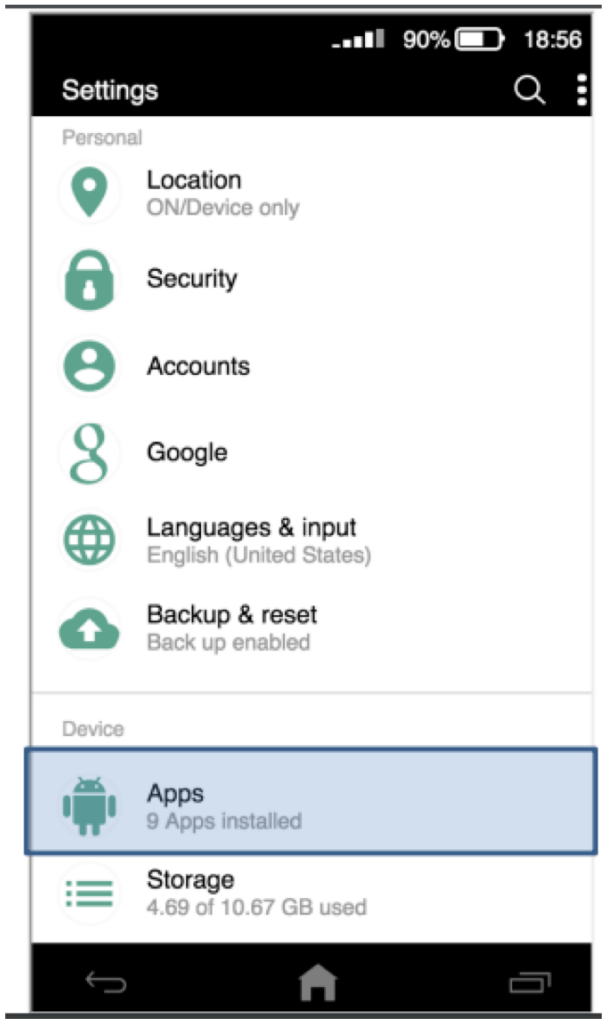
	Error rate	Average prompts/user
Ask on first use (Android/iOS)	15.5%	12.34
ML model (SVM with RBF kernel)	3.2%	12.00

Usable permission: What if the classifier mess up?

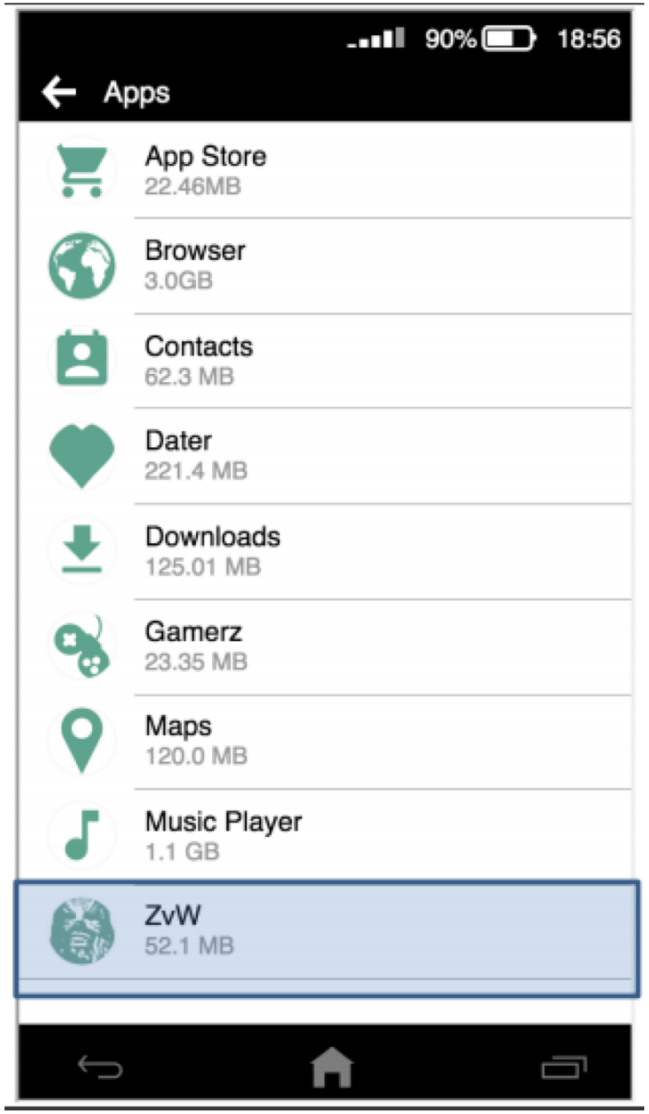
Users need go to app settings page

How hard can that be?

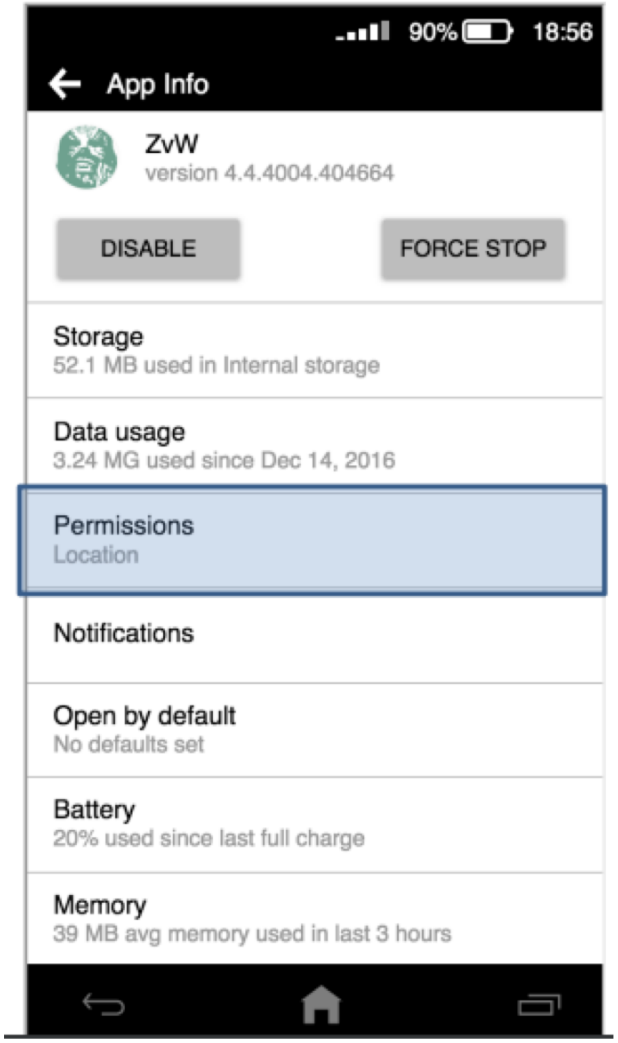
Correcting error in per-app permission



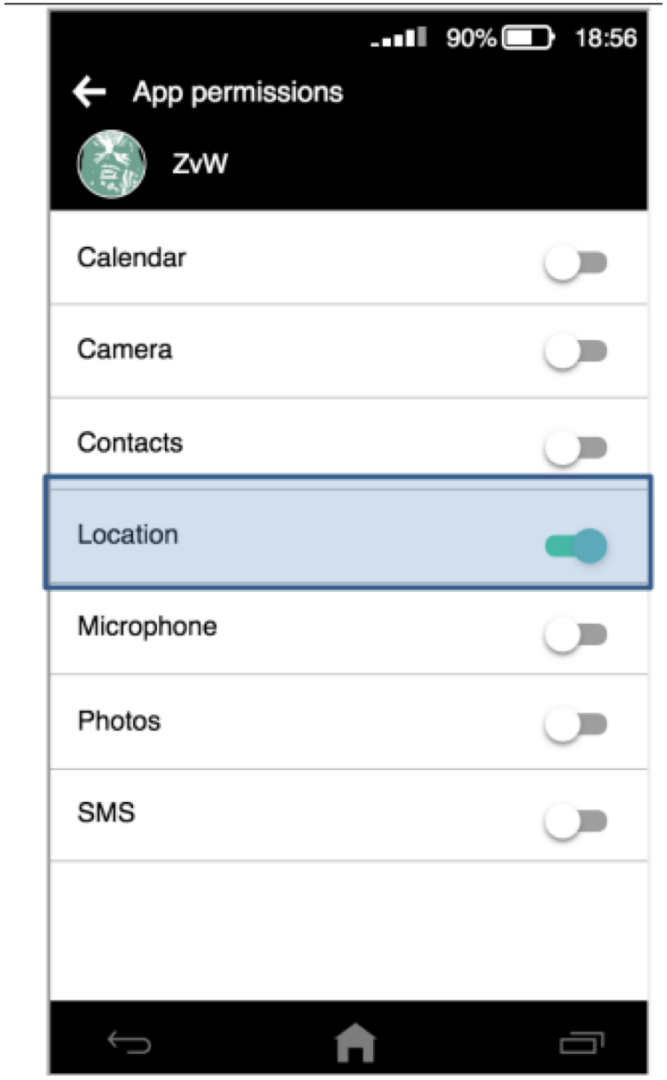
Correcting error in per-app permission



Correcting error in per-app permission



Correcting error in per-app permission



Issues (even) with an automated permission classifier

1. No overall view on what the apps have accessed

2. Per-app information is hard to access

3. Permissions are non-contextual

How do improve usability further?

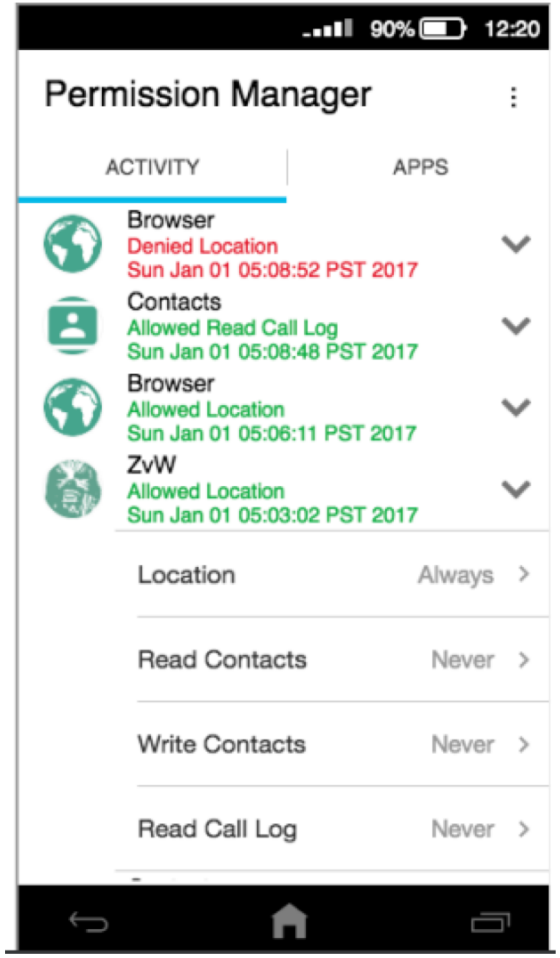
Idea: How about a privacy dashboard

- More control to the user

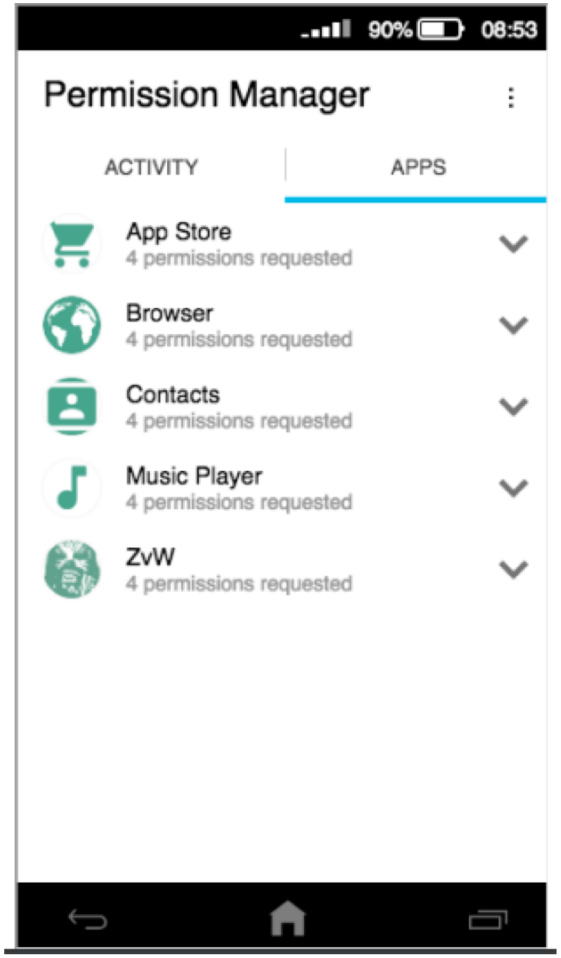
The dashboard should help to

- Understand automated decisions
- Change incorrect decisions

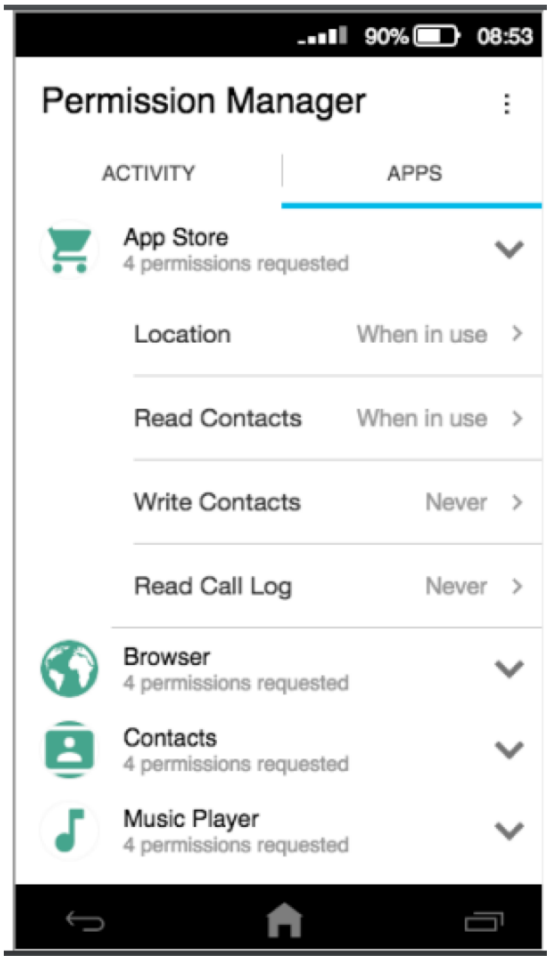
TurtleGuard



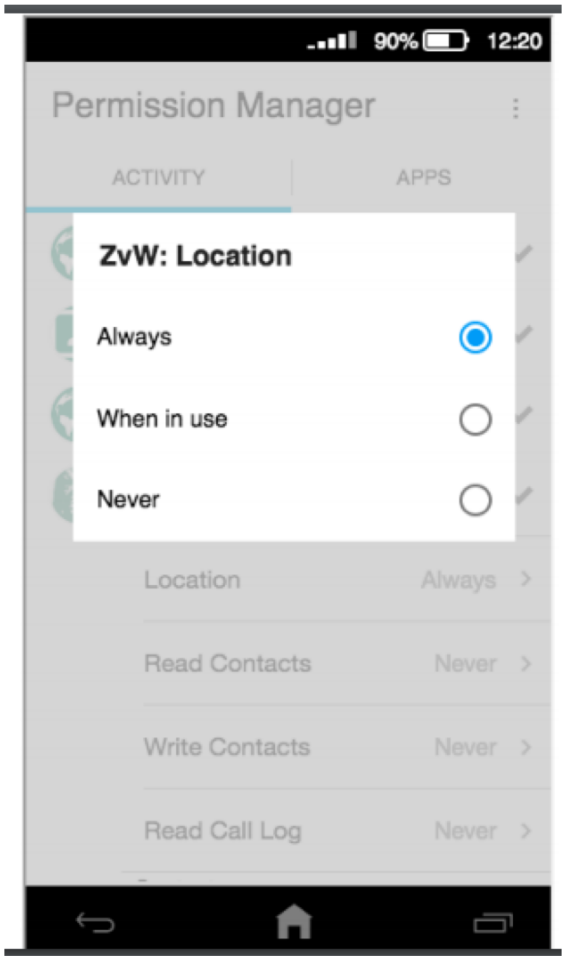
TurtleGuard



TurtleGuard



TurtleGuard



TurtleGuard : Usability

- Deploy and give users tasks to complete
- Task 1 : what are the two most recent apps that accessed the device's location?

- Task 2: currently, which of the following data types can be accessed by the ZvW app?

TurtleGuard : Usability

- Deploy and give users tasks to complete
- Task 1 : what are the two most recent apps that accessed the device's location?

- Task 2: currently, which of the following data types can be accessed by the ZvW app?

TurtleGuard users performs worse than control!

TurtleGuard : Usability

- Task 3 : is the ZvW app able to access location data even when it is not actively being used?

- Task 4: prevent it from doing so…or explain whether it is even possible

TurtleGuard : Usability

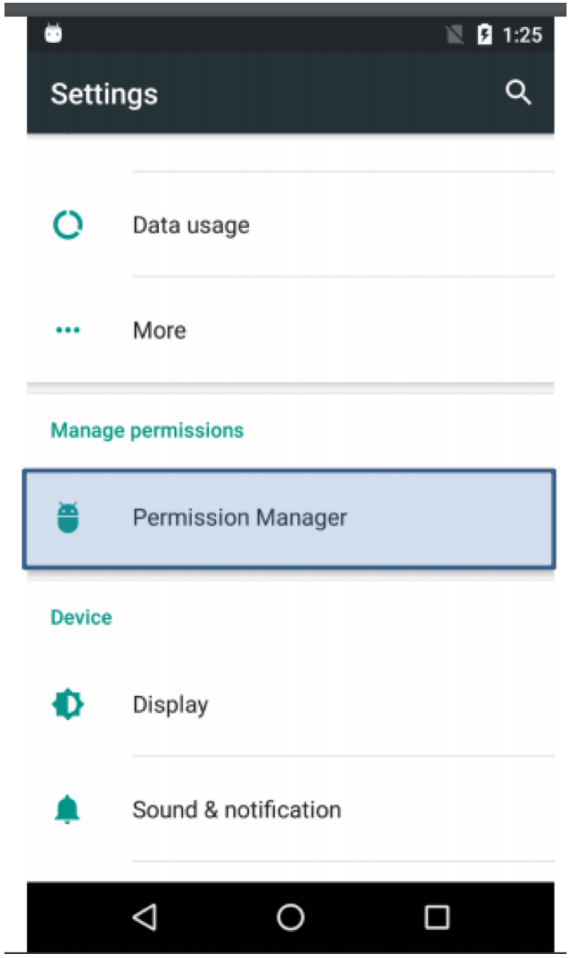
- Task 3 : is the ZvW app able to access location data even when it is not actively being used?

- Task 4: prevent it from doing so…or explain whether it is even possible

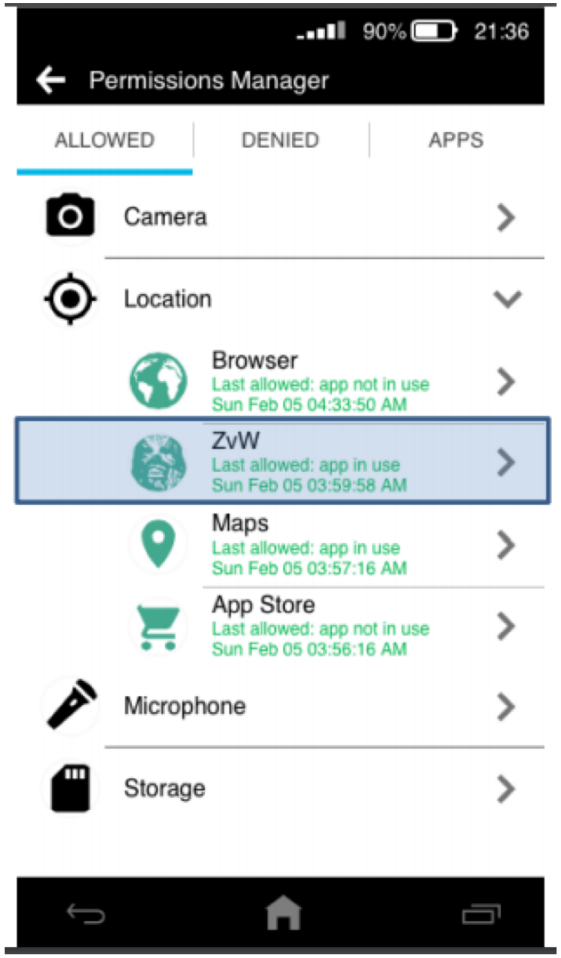
TurtleGuard helps the users in these tasks

ITERATIVE DESIGN

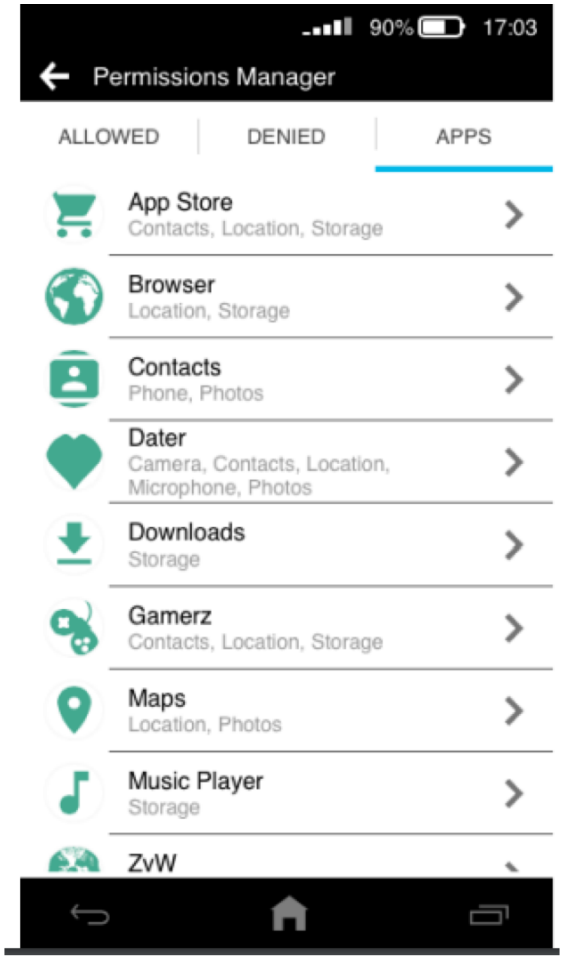
Improve usability of TurtleGuard



Improve usability of TurtleGuard



Improve usability of TurtleGuard



TurtleGuard++ : Usability

- Task 1,2,3,4 :
  - TurtleGuard significantly helps the users in these tasks

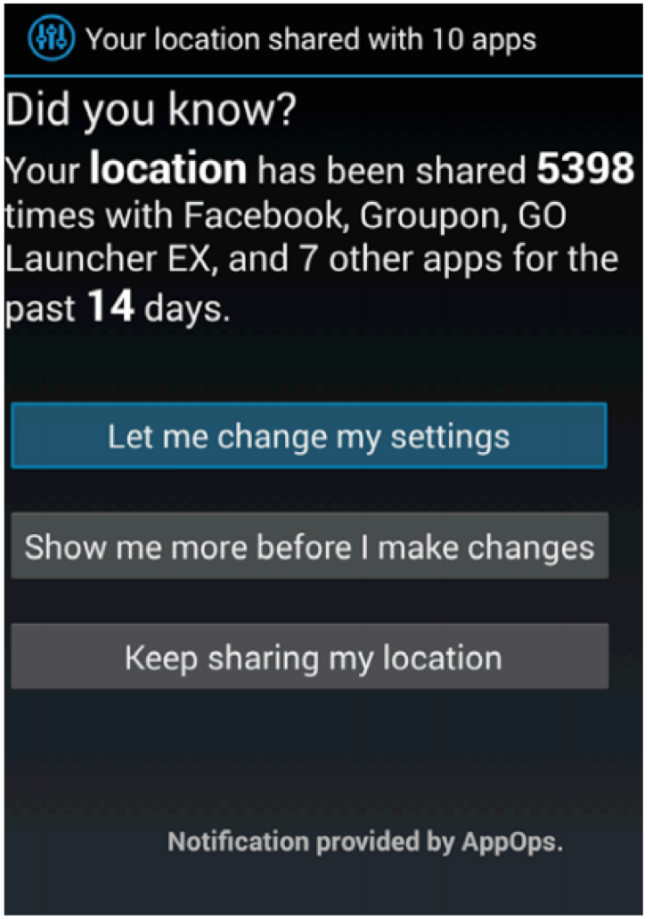
- Permission models in Mobiles
  - Help users to automate the permission
  - Create a dashboard to give users more control

NUDGING

www.normsadeh.com/file_download/185 (slide 15 to 35)

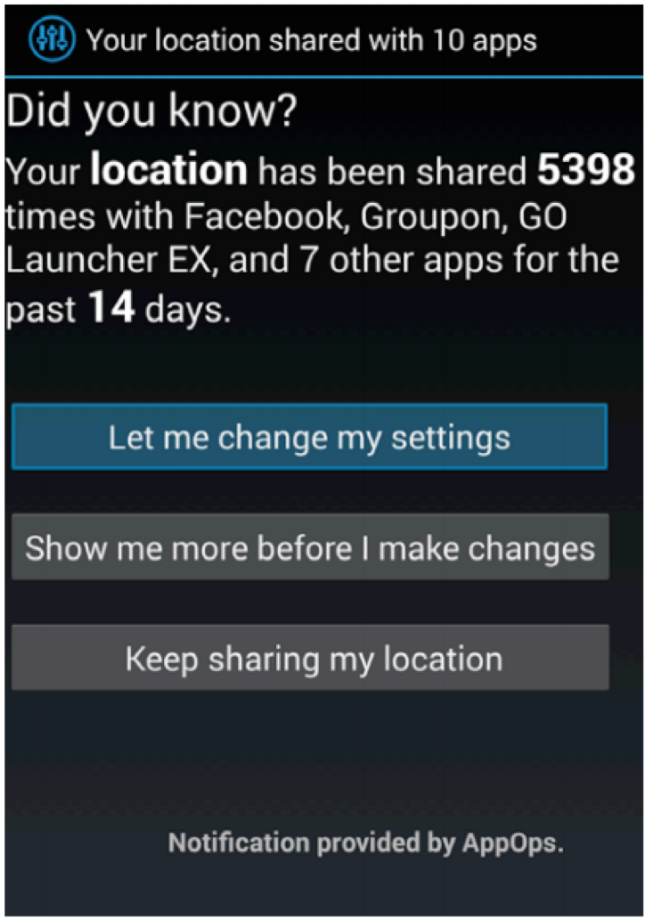
How to “Nudge” people

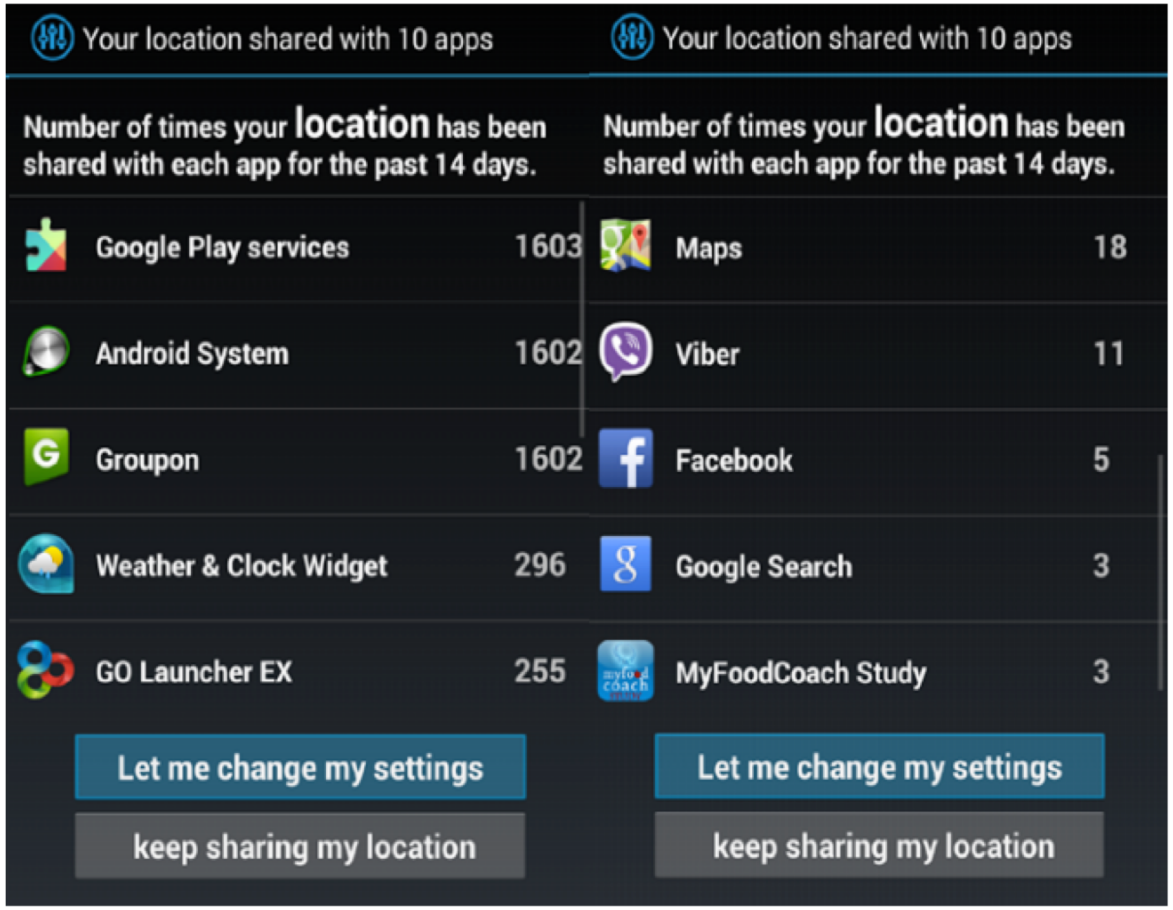
[Almuhmedi et al., CHI’15]



Counter **cognitive** and **behavioral** bias

From “Nudge” to “details”





Designing Nudges

- What kind of nudges?
- How often?
- How do we evaluate the impact?