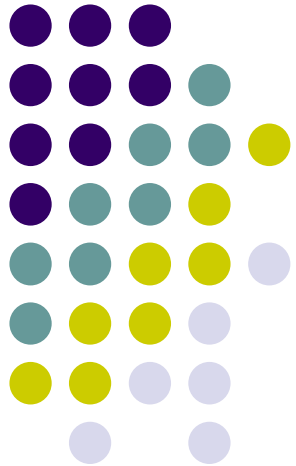
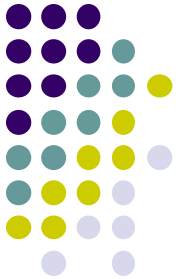
CS 528 Mobile and Ubiquitous Computing Lecture 6a: Other Android UbiComp Components, Tech Talk, Final Project Proposal & Smartphone Sensing

Emmanuel Agu



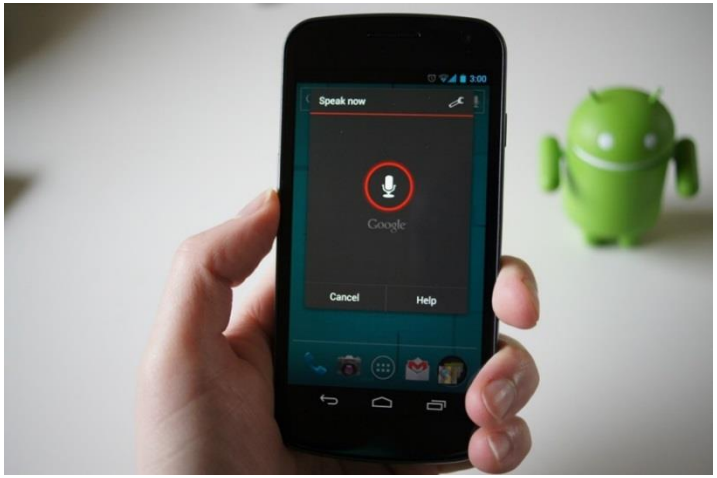


What other Android APIs may be useful for Mobile/ubicomp?

Speaking to Android

http://developer.android.com/reference/android/speech/SpeechRecognizer.html
https://developers.google.com/voice-actions/

- **Speech recognition:**
  - Accept inputs as speech (instead of typing) e.g. dragon dictate app?
  - Note: Requires internet access
- Two forms
  1. **Speech-to-text**
     - Convert user's speech to text. E.g. display voicemails in text
  2. **Voice Actions:** Voice commands to smartphone (e.g. set alarm)

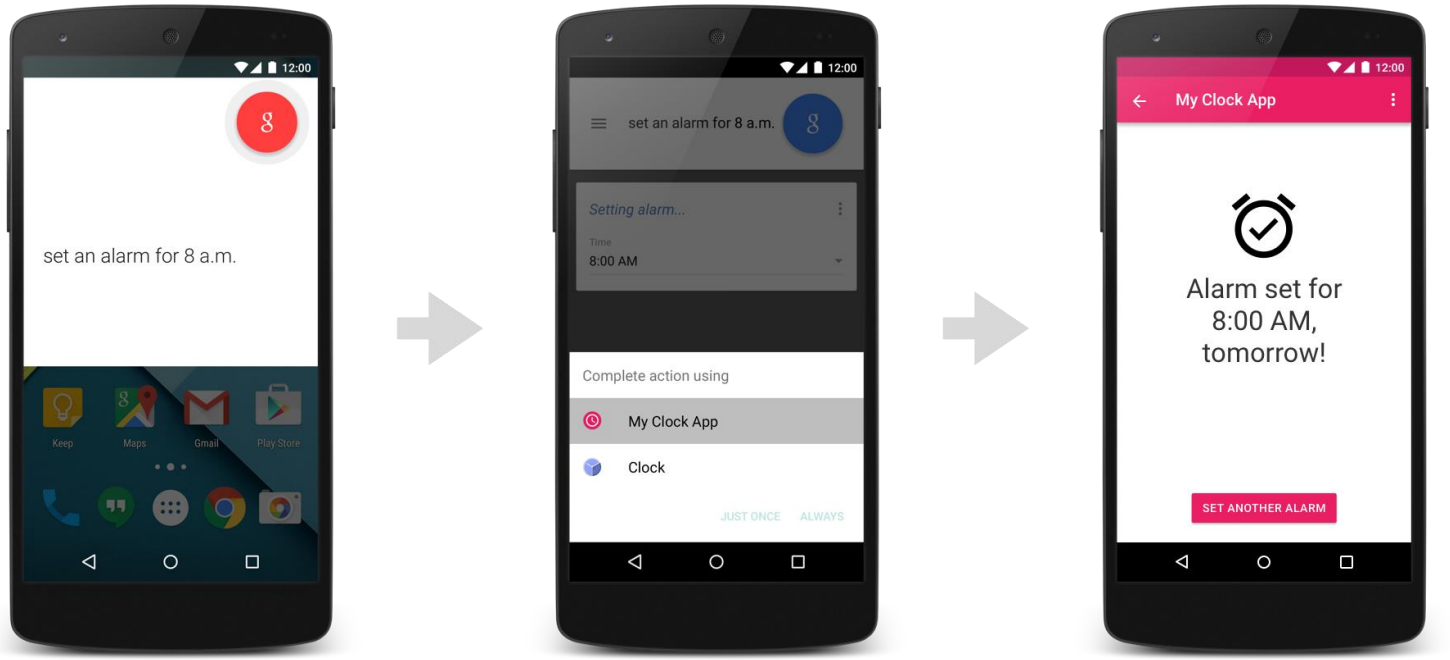


Speech to text

Google Voice Actions

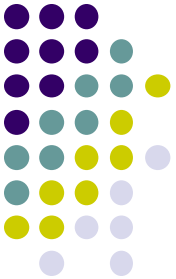
https://developers.google.com/voice-actions/

- E.g. Tell Google to set an alarm

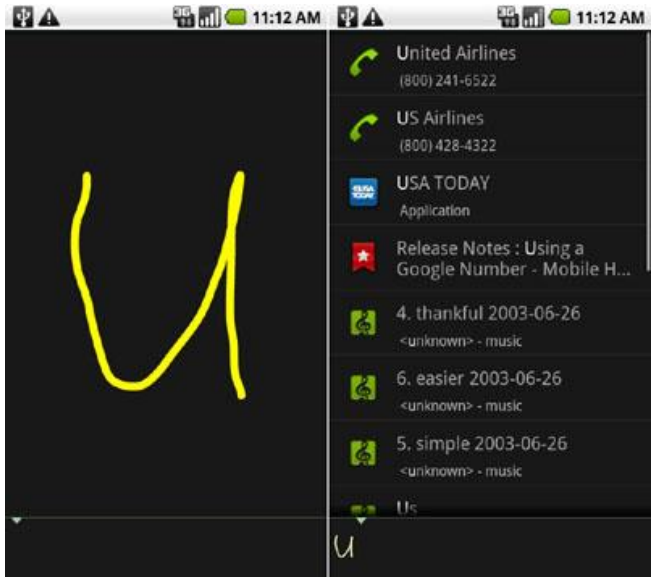


Gestures

https://developer.android.com/training/gestures/index.html
http://www.computerworld.com/article/2469024/web-apps/android-gestures--3-cool-ways-to-control-your-phone.html

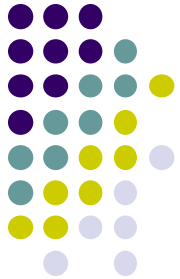


- **Gesture:** Hand-drawn shape on the screen
- Example uses:
  - Search your phone, contacts, etc by handwriting onto screen
  - Speed dial by handwriting first letters of contact's name
  - Multi-touch, pinching



More MediaPlayer & RenderScript

http://developer.android.com/guide/topics/renderscript/compute.html
https://developer.android.com/reference/android/media/MediaRecorder

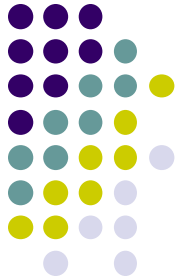


- MediaRecorder is used to **record** audio
  - Manipulate raw audio from microphone/audio hardware, PCM buffers
    - E.g. if you want to do audio signal processing, speaker recognition, etc
    - **Example:** process user's speech, detect emotion, nervousness?
  - Can playback recorded audio using MediaPlayer

- **RenderScript**
  - High level language for computationally intensive tasks/GPGPU,
  - Can be used to program phone CPU, GPU in a few lines of code
  - Use Phone's Graphics Processing Unit (GPU) for computational tasks
  - Useful for heavy duty tasks. E.g. image processing, computational photography, or computer vision

Wireless Communication

http://developer.android.com/guide/topics/connectivity/bluetooth.html
http://developer.android.com/reference/android/net/wifi/package-summary.html



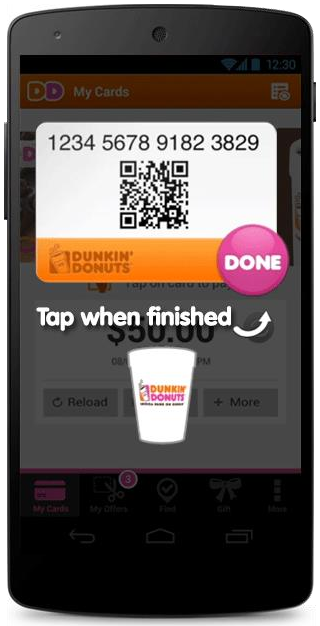
- Bluetooth
  - Discover, connect to nearby bluetooth devices
  - Communicating over Bluetooth
  - Exchange data with other devices

- WiFi
  - Scan for WiFi hotspots
  - Monitor WiFi connectivity, Signal Strength (RSSI)
  - Do peer-to-peer (mobile device to mobile device) data transfers

Wireless Communication

http://developer.android.com/guide/topics/connectivity/nfc/index.html

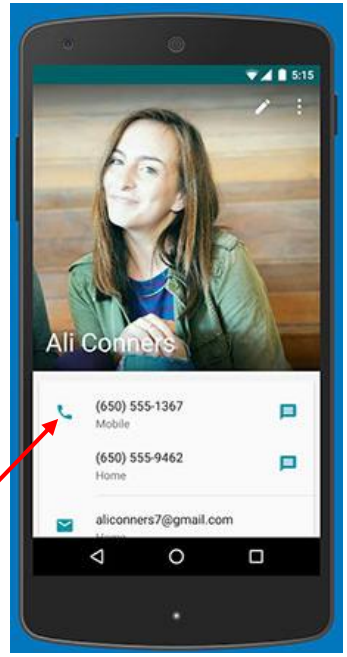
- NFC:
  - Contactless, transfer small amounts of data over short distances
  - **Applications:** Share spotify playlists, Google wallet
  - **Android Pay**
    - Store debit, credit card on phone
    - Pay by tapping terminal



Telephony and SMS

http://developer.android.com/reference/android/telephony/package-summary.html
http://developer.android.com/reference/android/telephony/SmsManager.html

- **Telephony:**
  - Initiate phone calls from within app
  - Access dialer app, etc
- **SMS:**
  - Send/Receive SMS/MMS from app
  - Handle incoming SMS/MMS in app



Dialer

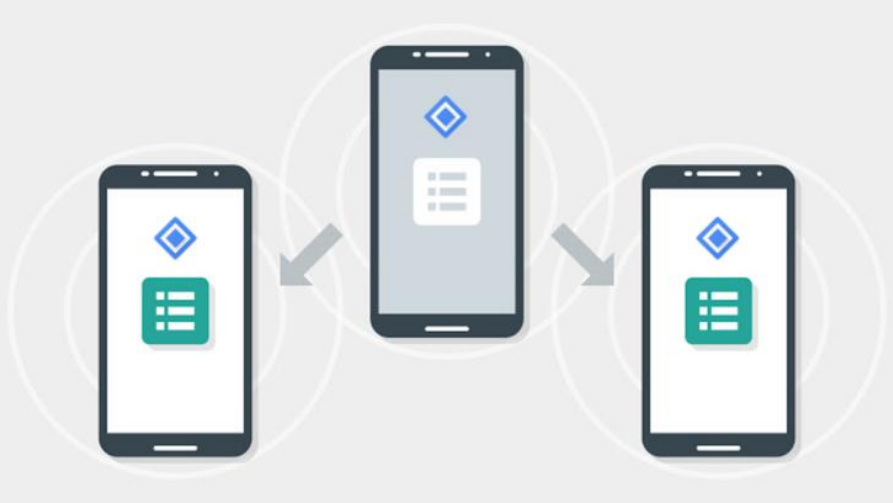
SMS

Google Play Services: Nearby Connections API

https://developers.google.com/nearby/connections/overview

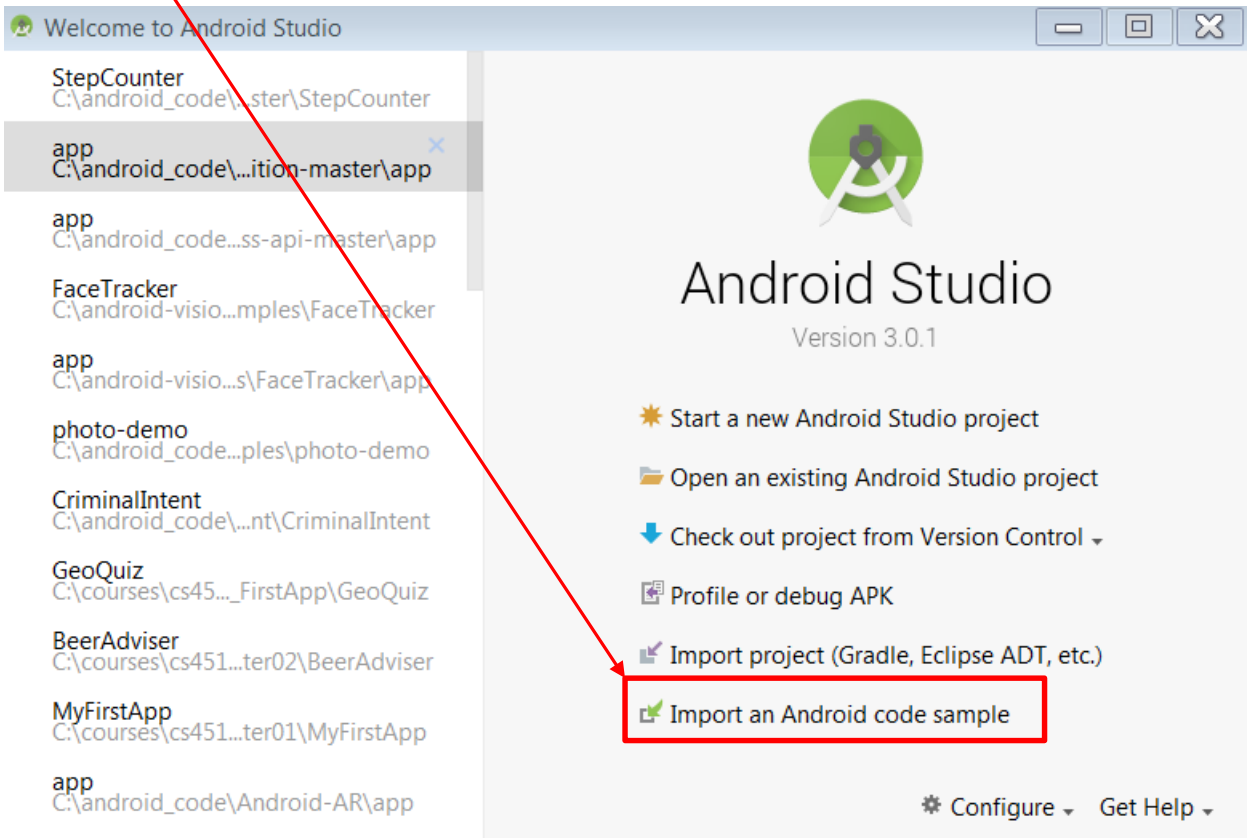
- Peer-to-peer networking API, allows devices communicate over a LAN
- Allows one device to serve as host, advertise
- Other devices can discover host, connect, disconnect
- **Use case:** Multiplayer gaming, shared virtual whiteboard
- Good tutorial by Paul Trebilcox-Ruiz

https://code.tutsplus.com/tutorials/google-play-services-using-the-nearby-connections-api--cms-24534?_ga=2.245472388.1231785259.1517367257-742912955.1516999489



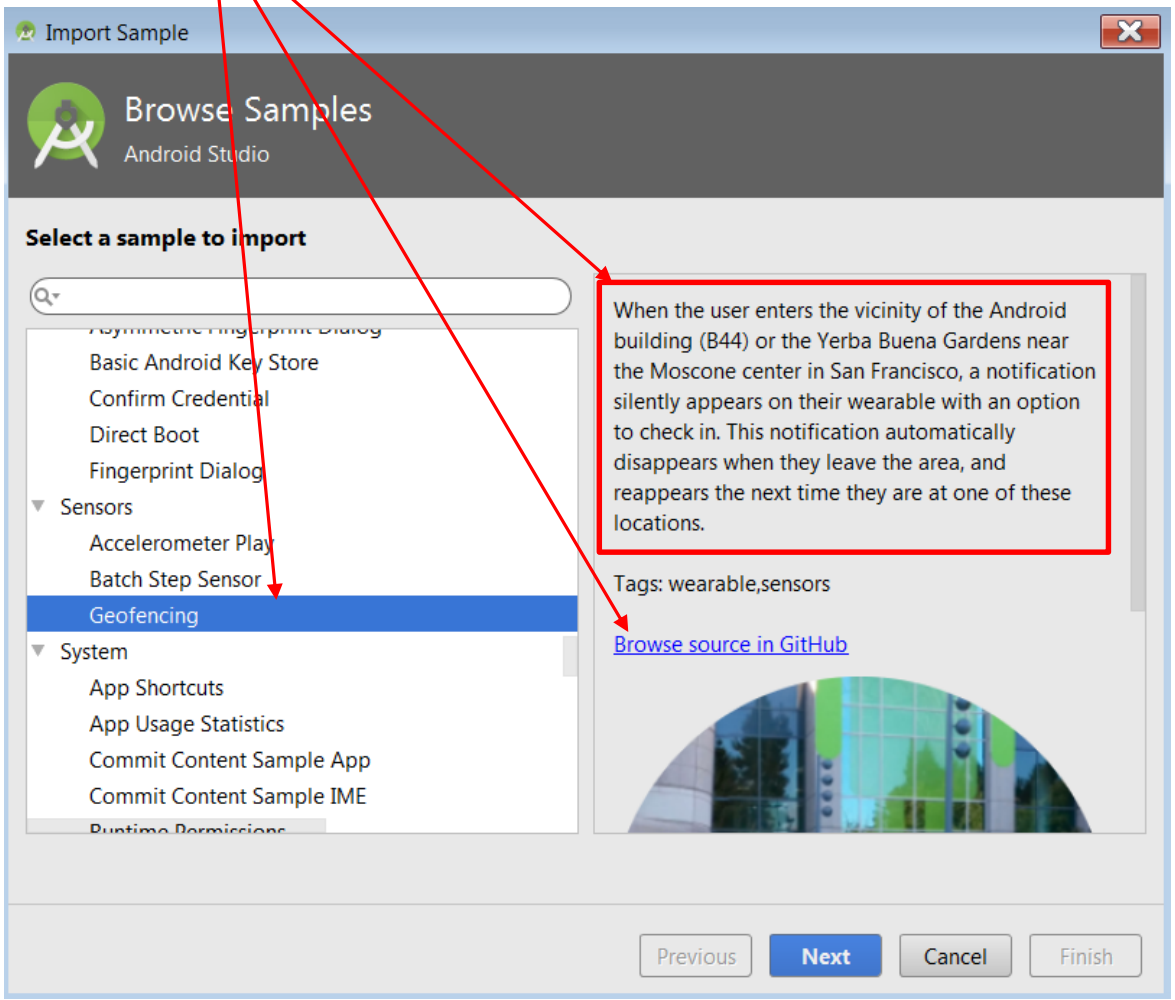
Google Android Samples

- Android Studio comes with many sample programs
- Just need to import them



Google Android Samples

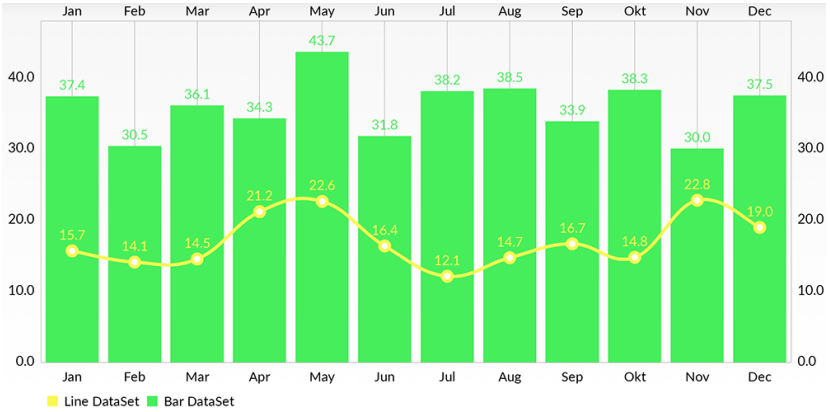
- Can click on any sample, read overview
- Source code available on github
- Tested, already working



Other 3rd Party Stuff

http://web.cs.wpi.edu/~emmanuel/courses/ubicomp_projects_links.html
https://developer.qualcomm.com/software/trepn-power-profiler

- **MPAndroid:** Add charts to your app



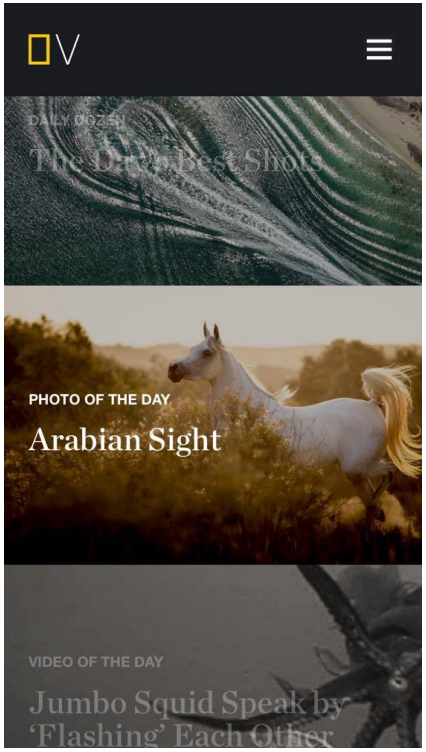
- **Trepn:** Profile power usage and utilization of your app (CPU, GPU, WiFi, etc)
  - By Qualcomm

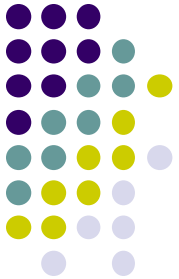
Other 3rd Party Stuff

http://web.cs.wpi.edu/~emmanuel/courses/ubicomp_projects_links.html

- **Programmable Web APIs:** 3rd party web content (e.g RESTful APIs) you can pull into your app with few lines of code
  - **Weather:** Weather channel, yahoo weather
  - **Shared interests:** Pinterest
  - **Events:** Evently, Eventful, Events.com
  - **Photos:** flickr, Tumblr
  - **Videos:** Youtube
  - **Traffic info:** Mapquest traffic, Yahoo traffic

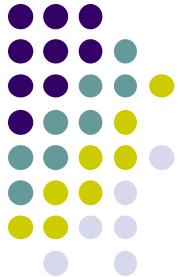
- **E.g. National Geographic:** picture of the day





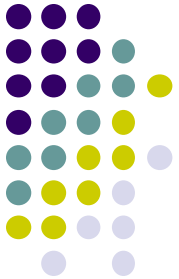
Student Presentation: Mobile Technologies

Talk: Mobile Technology



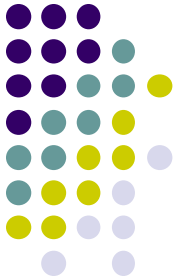
- GROUP to research, master and present on any TWO mobile technologies.
- Your talk should cover:
  - Background on the technology (tell a story about its history, etc)
  - Specific problems it's designed to solve
  - Typical example use case: When is it typically used?
  - Real world examples of where it is being used. E.g. by XYZ company for ABC
  - Overview of how it works?
  - Code snippet: Walk through a simple program that uses the technology including how to compile it and how to run it.

Talk on Mobile Technology



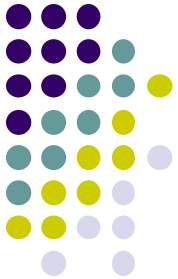
- Submit talk slides + working code
- To avoid duplicate presentations, each group email me their TWO topics by November 1, 2018
- This talk is 15% of your grade!
- The idea is to become expert, help any groups that need your help on that technology

Talk on Mobile Technology

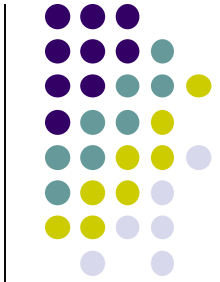


- Mobile programming/develpment:
  - Kotlin
  - iPhone development
  - 3rd part libraries: E.g. Xamarin
  - Mobile web programming
  - PhoneGap
  - AppInventor
  - Mobile game development tools: Unity,
- Machine/Deep Learning:
  - Deep Learning/machine learning in Android: Tensorflow, etc
  - Mobile machine/deep learning support in MATLAB
  - Keras support for Android Deep learning
  - Neural Networks API (NNAPI)

Talk on Mobile Technology



- More Google APIs (that could be used by mobile devices):
  - Analytics
  - Google Drive
  - Google Fit
  - Google Cast
  - Advertising: E.g. Adwords, Admobs
- More Android APIs:
  - Firebase (database, messaging, authentication, analytics, etc)
  - Speaking to Android (Speech recognition, Voice Actions)
  - Renderscript
  - Media Recorder
  - Wireless Communication: Bluetooth, WiFi, NFC, etc
  - Android Pay
  - Telephone/SMS
  - Nearby Connections API
  - Depth Sensing: Project Tango
  - Augmented Reality: ARtoolkit, vuforia, EasyAR



Final Project Proposal

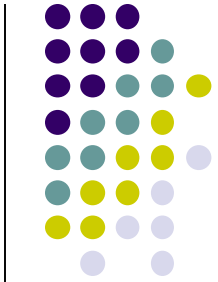
Final Project Proposal

- While working on projects 3 & 4, also brainstorm on final project
- Nov 1, Propose mobile/ubicomp app, solves WPI problem or Machine learning
- Proposals should include:
  1. **Problem you intend to work on**
     - Solve WPI/societal problem (e.g. walking safe at night)
     - Use at least 3 mobile/ubicomp components (e.g. location, sensor or camera)
     - If games, must gamify solution to real world problem
  2. **Why this problem is important**
     - E.g. 37% of WPI students feel unsafe walking home
  3. **Related Work:** What prior solutions have been proposed for this problem
  4. **Summary of envisioned mobile app (?) solution**
     1. E.g. Mobile app automatically texts users friends when they get home at night

Final Project Proposal

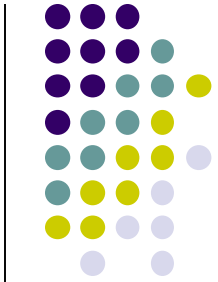
- Can also do Machine learning project that classifies/detects analyzes a dataset of builds a real-time app to classify some human sensor data. E.g. Classifies
  - A speaker's voice to determine if nervous, sad, etc
  - A user's accelerometer data and recognizes their walk from 5-10 other people
  - A picture of a person's face and determines their mood
  - Data from a person's phone to measure their sleep duration or/and quality
  - Video of a person's face to detects their heart rate
  - A person's communication/phone usage patterns to detect their mood
- See project difficulty points rubric
- Also propose evaluation plan
  - E.g. Small user study to evaluate app.
  - Can trade with another team: you review our app, we review yours
  - Machine learning performance metrics (e.g. classification accuracy, cross validation, etc)
- Can bounce ideas off me (email, or in person)
- Can change idea any time

Rubric: Grading Considerations



- **Problem (10/100)**
  - How much is the problem a real problem (e.g. not contrived)
  - Is this really a good problem that is a good fit to solve with mobile/ubiquitous computing? (e.g. are there better approaches?)
  - How useful would it be if this problem is solved?
  - What is the potential impact on the community (e.g. WPI students) (e.g. how much money? Time? Productivity.. Would be saved?)
  - What is the evidence of the importance? (E.g. quote a statistic)
- **Related Work (10/100)**
  - What else as been done to solve this problem previously
- **Proposed Solution/Classification (10/100)**
  - How good/clever/interesting is the solution?
  - How sophisticated and how are the mobile/ubiquitous computing components (high level) used? (e.g. location, geofencing, activity recognition, face recognition, machine learning, etc)

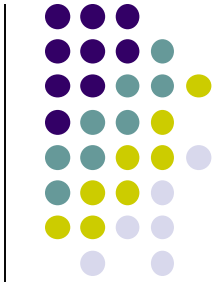
Rubric: Grading Considerations



- **Implementation Plan + Timeline (10/100)**
  - Clear plans to realize your design/methodology
  - Android modules/3rd party software used
  - Software architecture,
  - Screenshots (or sketches of UI), or study design + timeline

- **Evaluation Plan (10/100)**
  - How will you evaluate your project.
  - E.g. small user studies for apps
  - Machine learning cross validation, etc

- 50 more points allotted for your slides + presentation

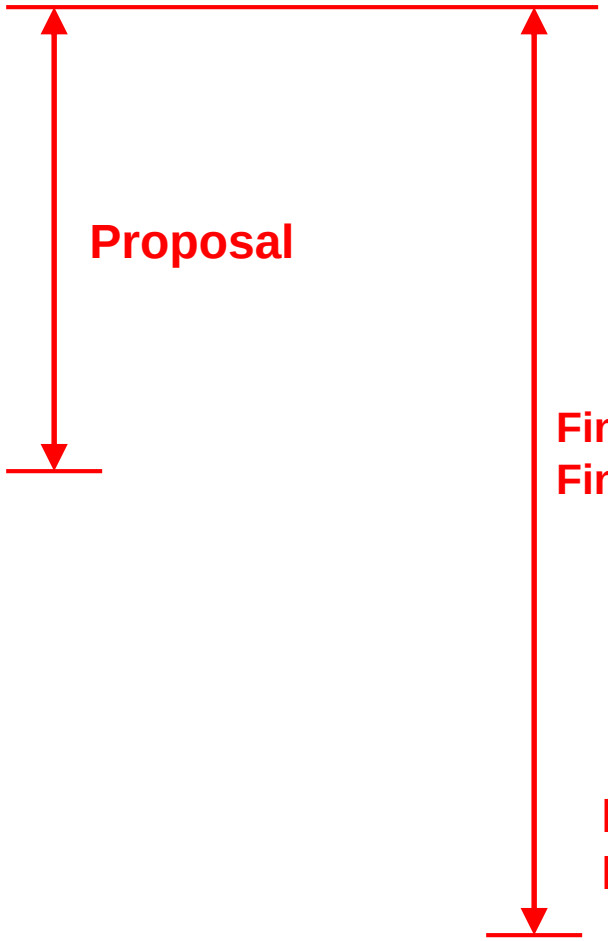


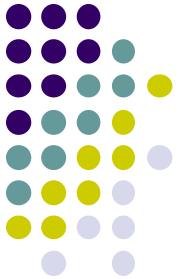
Final Project: Proposal Vs Final Submission (Presentation + Paper)

Final Project Proposal Vs Final Submission

- Introduction
- Related Work
- Approach/methodology
- Implementation
- **Project timeline**

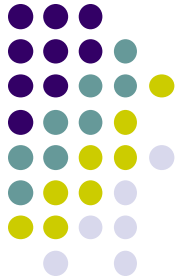
- Evaluation/Results
- Discussion
- Conclusion
- Future Work





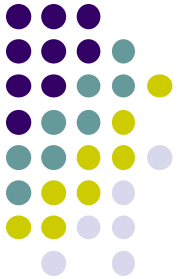
The Rest of the Class

The Rest of this class



- **Part 1: Course and Android Introduction**
  - Introduce mobile computing, ubiquitous Computing, Android,
  - Basics of Android programming, UI, Android Lifecycle
- **Part 2: Mobile and ubicomp Android programming**
  - mobile Android components (location, Google Places, maps, geofencing)
  - Ubicomp Android components (camera, face detection, activity recognition, etc)
- **Part 3: Mobile Computing/Ubicomp Research**
  - Machine learning (classification) in ubicomp
  - Ubicomp research (smartphone sensing examples, human mood detection, etc) using machine learning
  - Mobile computing research (app usage studies, energy consumption, etc)

Next!!



Smartphone Sensing

Smartphone Sensors

- Typical smartphone sensors today
  - accelerometer, compass, GPS, microphone, camera, proximity
- Use machine learning to classify sensor data

Future sensors?

- Heart rate monitor,
- Activity sensor,
- Pollution sensor,
- etc

Growth of Smartphone Sensors

- Every generation of smartphone has more and more sensors!!

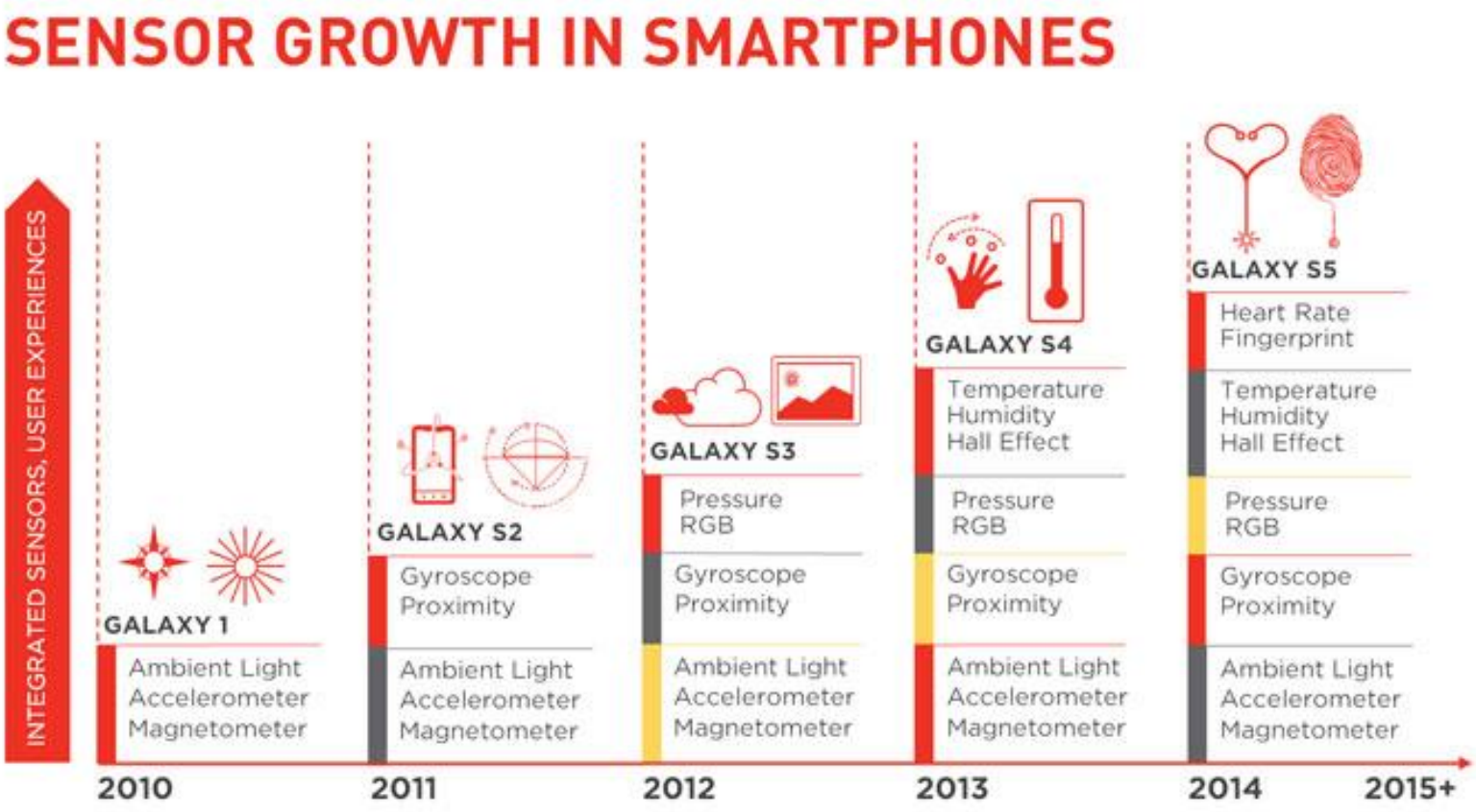


Image Credit: Qualcomm

Smartphone Sensing: What Can We Detect/Infer using These Sensors

24/7 detection, in natural settings

Smartphone Sensor data

Machine Learning

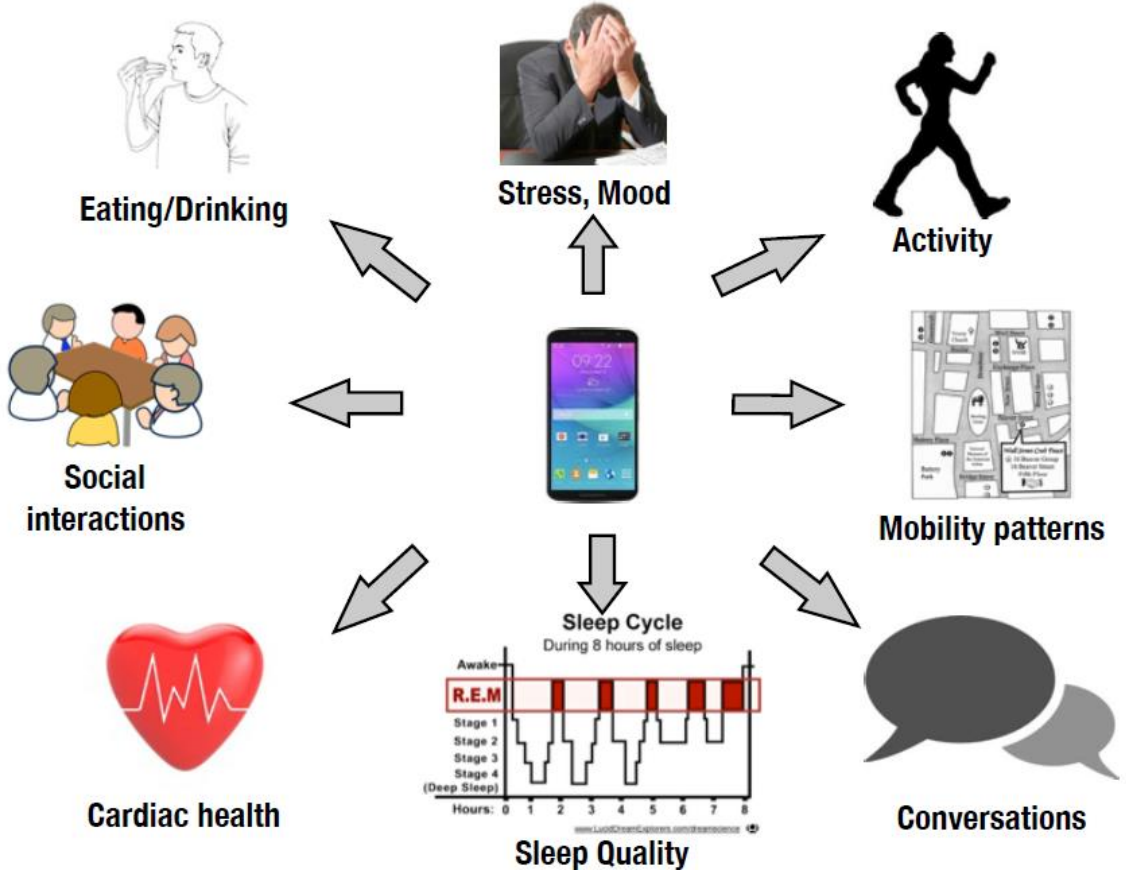


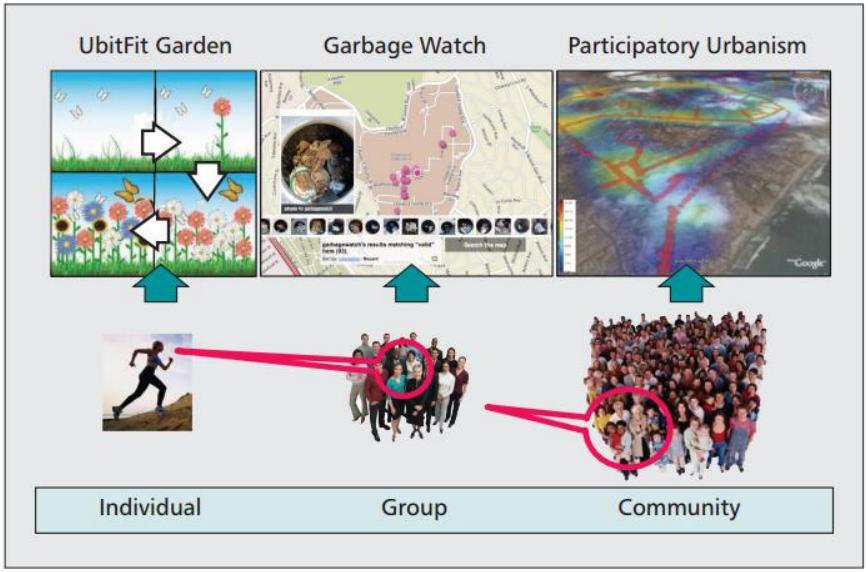
Image Credit: Deepak Ganesan, UMass

Sense What?

- **Environmental:** pollution, water levels in a creek
- **Transportation:** traffic conditions, road conditions, available parking
- **City infrastructure:** malfunctioning hydrants and traffic signs
- **Social:** photoblogging, share bike route quality, petrol price watch
- **Health and well-being:**
  - Share exercise data (amount, frequency, schedule),
  - share eating habits and pictures of food

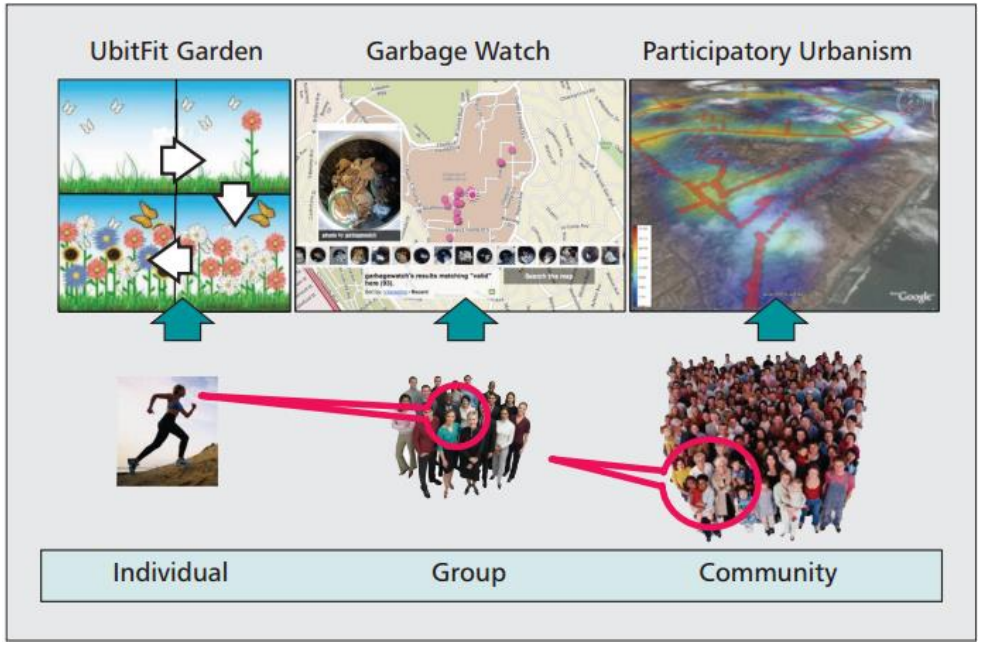
Mobile CrowdSensing

- **Mobile CrowdSensing:** Sense collectively
- **Personal sensing:** phenomena for an individual
  - E.g: activity detection and logging for health monitoring
- **Group:** friends, co-workers, neighborhood
  - E.g. GarbageWatch recycling reports, neighborhood surveillance



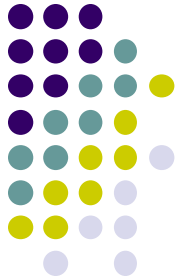
Mobile CrowdSensing

- **Community sensing (mobile crowdsensing):**
  - Large-scale phenomena monitoring
  - Many people contribute their individual readings
  - **Examples:** Traffic congestion, air pollution, spread of disease, migration pattern of birds, city noise maps



Mobile Crowd Sensing Types

- Many people cooperate, share sensed values
- 2 types:
  1. ***Participatory Sensing:*** User manually enters sensed values (**active** involvement)
     - E.g. Comparative shopping: Compare price of toothpaste at CVS vs Walmart
  2. ***Opportunistic Sensing:*** Mobile device automatically senses values (**passive** involvement)
     - E.g. Waze crowdsourced traffic



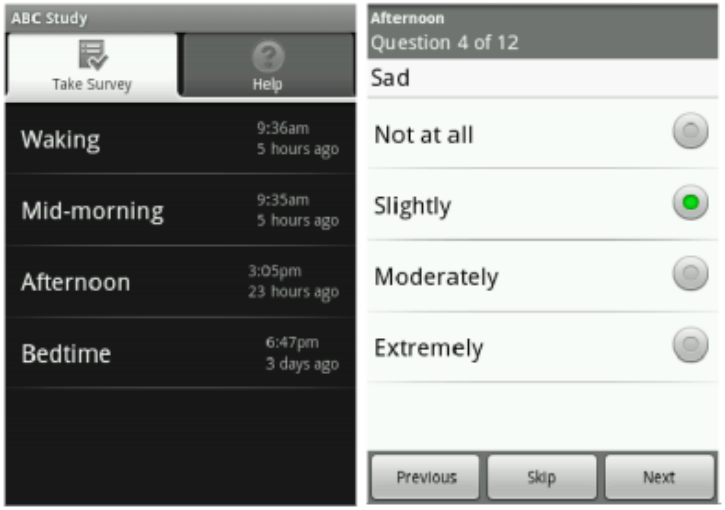
Smartphone Sensing Examples

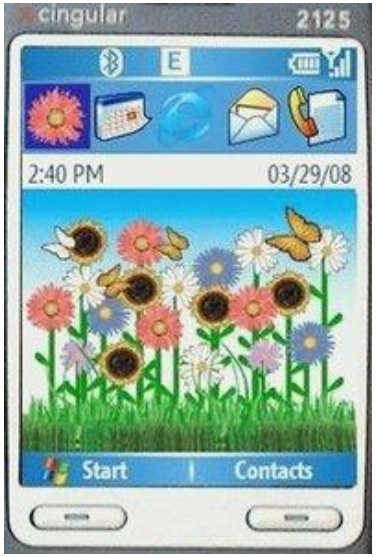
Personal Sensing

- Personal monitoring
- Focusing on user's daily life, physical activity (*Khan et al.*)
- Basically like Fitbit on your phone

Other Examples of Personal Participatory Sensing

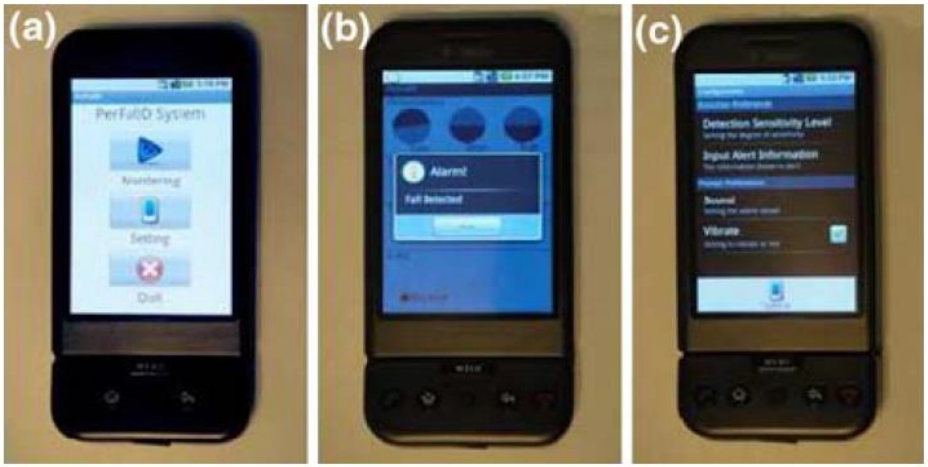
- AndWellness
  - “Personal data collection system”
  - Active user-triggered experiences and surveys
  - Passive recording using sensors
- UbiFit Garden
  - Uses smartphone sensors , real-time tracking, statistical modeling, and a personal, mobile display to encourage regular physical activity





Personal Opportunistic Sensing

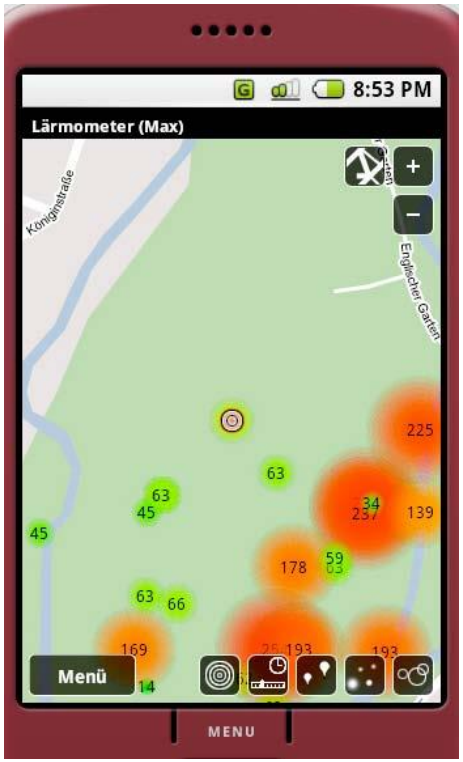
- PerFalld
  - Detects if user falls using sensor
  - Starts a timer if it detects that someone fell
  - If individual does not stop timer before it ends, emergency contacts are called



User interfaces in PerFallD: (**a**) bright, large virtual buttons on operating screen (**b**) clear alert window (**c**) simple, non-confusing preference screen

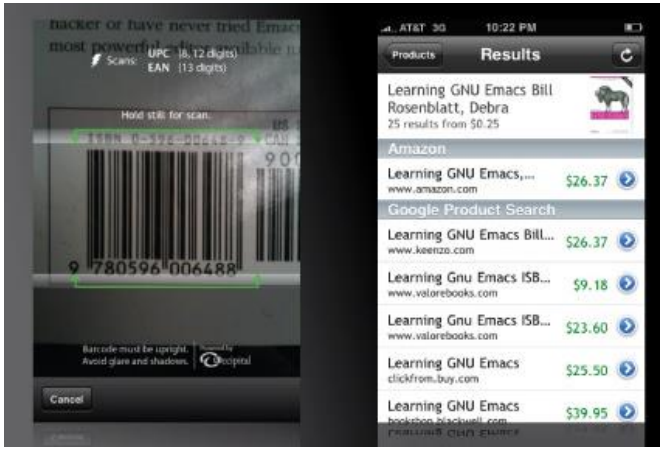
Public Sensing

- Data is shared with everyone for public good
- Traffic
- Environmental
  - Noise levels
  - Air pollution



Public Participatory Sensing

- **LiveCompare**
  - User-created database of UPCs and prices
  - GPS and cell tower info used to find nearby stores
- **PetrolWatch**
  - Turns phone into fully automated dash-cam
  - Uses GPS to know when gas station is near

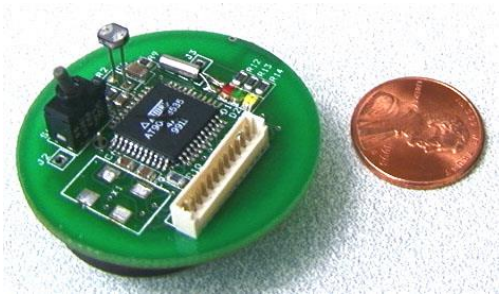


Public Participatory Sensing

- **Pothole Monitor**
  - Combines GPS and accelerometer
- **Party Thermometer**
  - Asks you questions about parties
  - Detects parties through GPS and microphone

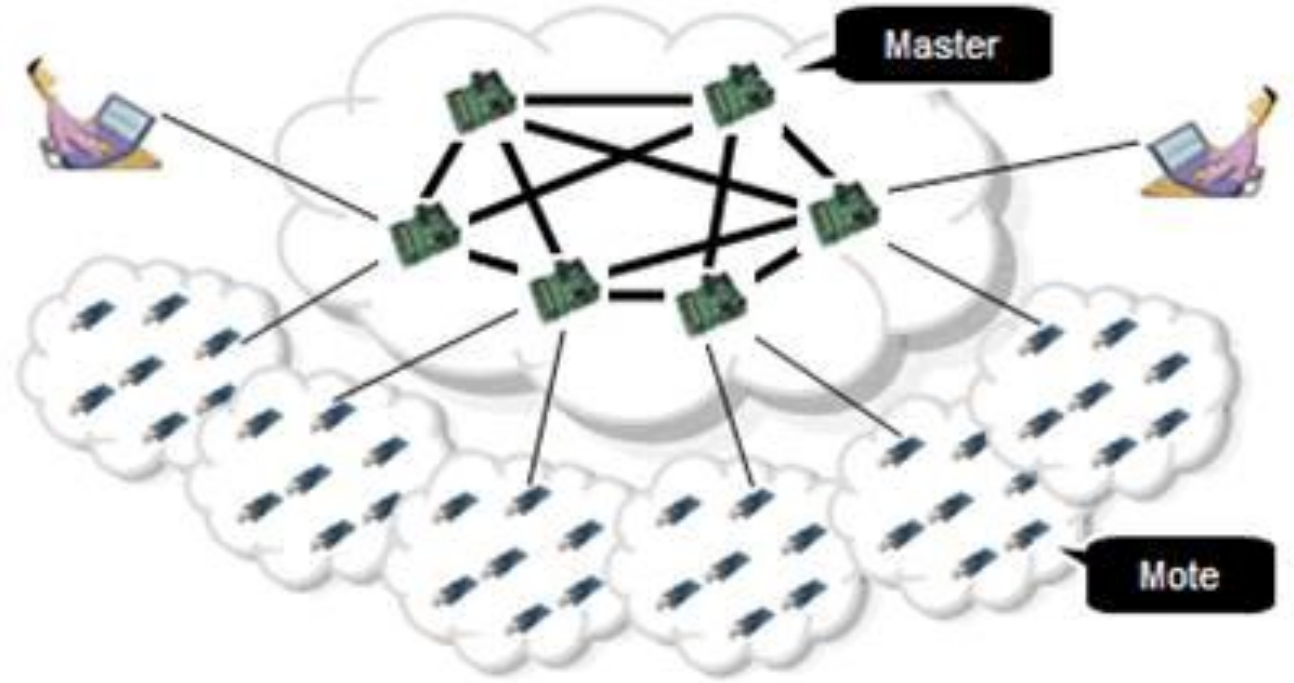
Smartphone Sensing vs Dedicated Sensors

VS

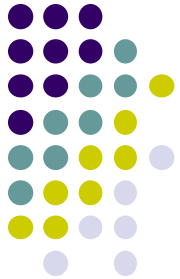


Background: Wireless Sensors for Environment Monitoring

- Embedded in room/environment
- Many sensors cooperate/communicate to perform task
- Monitors conditions (temperature, humidity, etc)
- User can query sensor (What is temp at sensor location?)



Sensing with Smartphones vs Dedicated Sensors

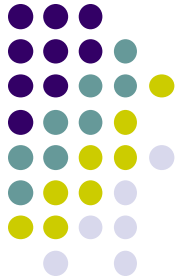


- **More resources:** Smartphones have much more processing and communication power

- **Easy deployment:** Millions of smartphones already owned by people
  - Instead of installing sensors in road, we detect traffic congestion using smartphones carried by drivers
  - Makes maintance easier. E.g. owner will charge their phone promptly

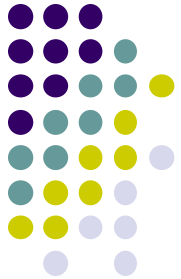
- **Time-varying data:** population of mobile devices, type of sensor data, accuracy changes often due to user mobility and differences between smartphones

Sensing with Smartphones vs Dedicated Sensors



- **Reuse of few general-purpose sensors:** While sensor networks use dedicated sensors, smartphones reuse relatively few sensors for wide-range of applications
  - E.g. Accelerometers used in transportation mode identification, pothole detection, human activity pattern recognition, etc

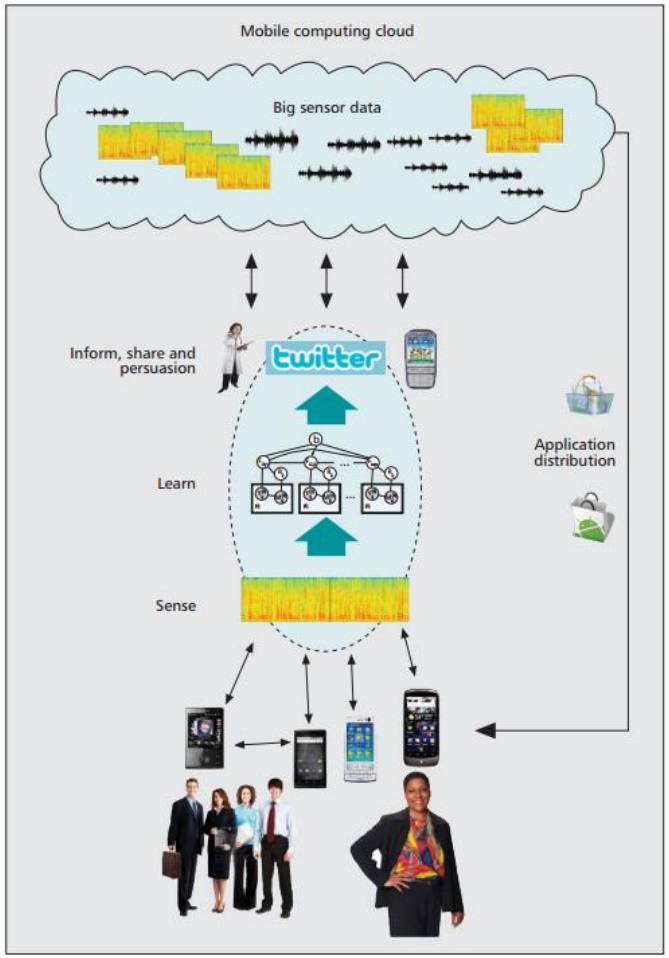
- **Human involvement:** humans who carry smartphones can be involved in data collection (e.g. taking pictures)
  - Human in the loop can collect complex data
  - Incentives must be given to humans



Smartphone Sensing Architecture

Smartphone Sensing Architecture

- Paradigm proposed by Lane *et al*
- **Sense:** Phones collect sensor data
- **Learn:** Information is extracted from sensor data by applying **machine learning and data mining** techniques
- **Inform, share and persuasion:** inform user of results, share with group/community or persuade them to change their behavior
  - **Inform:** Notify users of accidents (Waze)
  - **Share:** Notify friends of fitness goals (MyFitnessPal)
  - **Persuasion:** avoid speed traps (Waze)



References

1. ***A Survey of Mobile Phone Sensing.*** Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, Andrew T. Campbell, In IEEE Communications Magazine, September 2010
2. ***Mobile Phone Sensing Systems: A Survey,*** Khan, W.; Xiang, Y.; Aalsalem, M.; Arshad, Q.; , Communications Surveys & Tutorials, IEEE , vol.PP, no.99, pp.1-26