



UNIVERSITY OF CALIFORNIA  
**MERCED**

# Mobile Computing

CSE 162  
Fall 2025

Hua Huang

Department of Computer Science and Engineering

# Logistics

- Lecture time: 6-7:15 pm, Wednesday and Friday.
  - Location: Student Services Building | Room 130
- Office Hour:
  - Hua Huang: 1-2pm, Wednesday. SE2 275
  - Rahul Hoskeri: 2-3pm, SE2 lobby

# Textbooks

- Required: Head First Android Development, 2nd Edition (access the book through the university library: [link](#))
- (Optional): Manish J. Gajjar: *Mobile Sensors and Context-Aware Computing*, 2015; ISBN: 978-0-12-801660-2, (<https://www.sciencedirect.com/book/9780128016602/mobile-sensors-and-context-aware-computing>)
- (Optional): Raj Kamal: *Mobile Computing*, Oxford University Press, 2019; ISBN: 9780199455416(Any edition is fine, <https://archive.org/details/mobilecomputing000kama>)

# Textbook

- Download the book for free.
  - Head First Android Development, **2nd Edition** (access the book through the university library: [link](#))
- Actually read the book
  - It's not a manual for a quick answer look up. Read and there is much to learn.
- We are using the second edition.



BOOK  
**Head first Android development : a brain-friendly guide**  
Griffiths, Dawn, author.; Griffiths, David, author.  
Sebastopol, CA : O'Reilly Media, Incorporated; 2017; Second edition.  
Available Online >

# Grading

- Labs: 35%
  - Extra credits available
- Three Mid-term exams: 30%
- Final Exams: 30%
- Attendance: 5%

# Lab Overview

- A series of android programming projects to familiarize with mobile programming
  - UI
  - Sensor
  - Location aware services
  - Mobile AI
  - etc



# Question

- Do you?
  1. Own Android phones
  2. Can borrow android phones
  3. Do not own and cannot borrow
- Either phones or virtual machine implementations are fine.

# Lab Schedule

- Approximately One lab project every two weeks:
  - Basic tasks: implement an app. Follow the instruction of the TA
  - Bonus tasks: explore and complete an additional feature of the app
- Lab delivery:
  - Demo the features. Demonstrate that the prescribed features are up and running
  - Show your program. Be prepared to answer questions about your program.
  - Submit your functioning program through Catcourse.

# Plagiarism Policy

- Don't cheat.
  - These are exercises. You don't get punished by writing bugs. Instead, you gain experience and prepare for your future jobs.
  - We are here to help you finish them.
  - Understand every line of codes you write.
- If get caught, the consequence is grave
  - Zero grade for the assignment, fail the class, or worse

# Plagiarism Policy for Labs

- Create each app from scratch
- Naming standards required
- What are not cheating:
  - Codes generated by the IDE
  - Discussion with the TA or classmates and find out how to implement it.
  - Search tutorials about how to do it.

# Attendance Policy

- Physical attendance to the classes and labs are required
  - Lecture notes cannot replace lectures
  - In-class quizzes are not announced before hand
  - Exam questions are often discussed in lectures
- If you cannot attend, contact the instructors before hand
- Grading:
  - 80%+ attendance: 10pt
  - Between 30%-80%: 0-10pt
  - Below 30%: 0pt

# AI Usage

- The wrong way: AI is doing development, you help with debugging
  - AI can generate solutions to existing classical problems, which are also available in stackoverflow.com. It often has poor solution to new and unique problems.
  - AI-generated codes often seem to be convincing, while in fact includes hard-to-find bugs
    - You could end up spending more time debugging than coding on your own
    - You get into trouble in your job interview and your job.

# AI Usage

- The right way: You do the software development. Use AI to guide you about new features
  - Understanding of the program is required
    - Think about your interview
  - Debugging is required
    - If you are to develop anything meaningful, which is anything more complex than a leetcode question, you likely need to do the heavy lifting
    - This class offer you excellent opportunities to practice, with minimal consequence for errors
    - We are here to help with challenges
  - Treat AI as an improved stackoverflow: it finds you answers quickly. But you still need to verify the answers.

# AI Usage

- There are many noises. But AI cannot replace human in software development

[https://www.wsj.com/finance/softwares-death-by-ai-has-been-greatly-exaggerated-b639c0cd?st=icF2Qw&reflink=desktopwebshare\\_permalink](https://www.wsj.com/finance/softwares-death-by-ai-has-been-greatly-exaggerated-b639c0cd?st=icF2Qw&reflink=desktopwebshare_permalink)

# Research Projects

- Research projects are available
  - In the general topics covered by the course: networking, sensing, computing

# Introduction

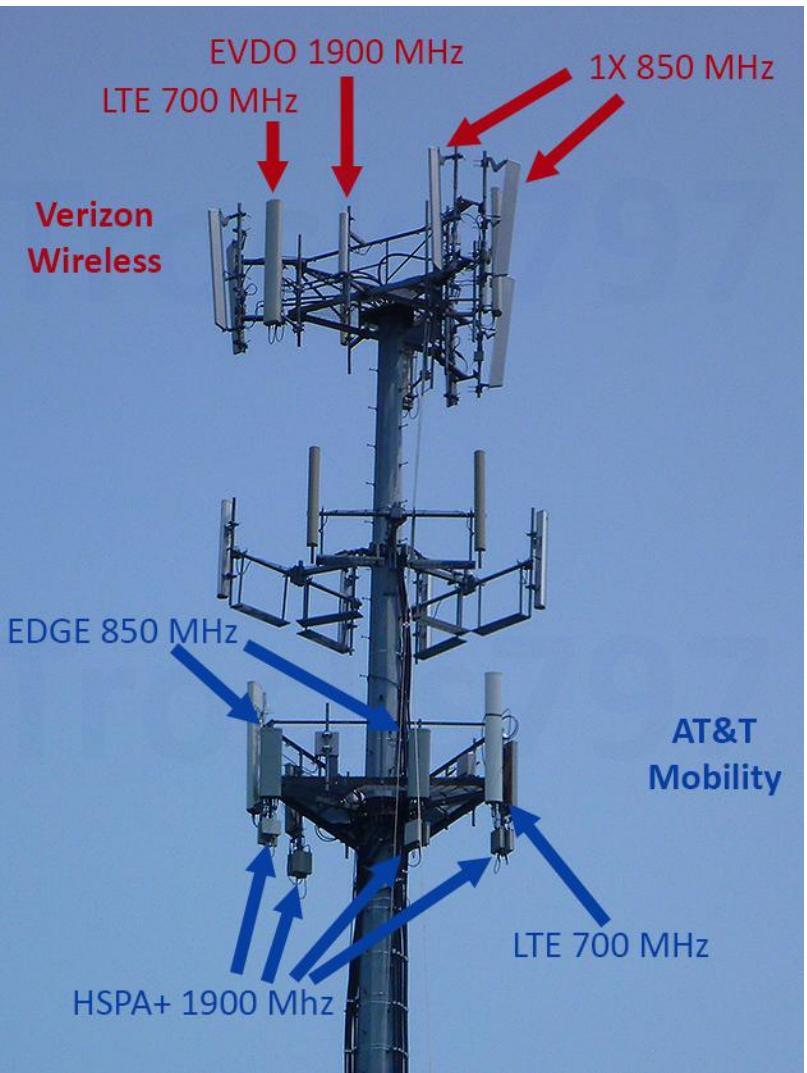
# History of Mobile Computing



1970s to early 2000:  
Consistent innovation in wireless/mobile communication



- On April 3 1973, Motorola engineer Marty Cooper made the first cell phone call
- The DynaTAC phone weighed about 2.2 pounds and was 10 inches long



- Ubiquitous voice. different frequency bands
  - Global System for Mobile communication (GSM)
  - Code Division Multiple Access (CDMA)
- Data connectivity: 3G, 4G LTE, 5G



# Mobile Phones were just phones for a while...

- Then things began to change
  - More and more people own mobile phones
  - Batteries got better, form factors improved, coverage improved, plans were better...
  - New applications besides communication become available
  - The handset manufacturers didn't want to write all the applications for these new phones
  - However... they didn't want to open up their platform...
  - The first mobile web platform was born (client-server model)

# WAP

- Wireless Application Protocol
- Basically it's a stripped-down HTTP that was meant to be better at transmitting over the unreliable mobile network



# Mobile Phone Economies

- Before there were app stores, purchases were made through SMS
  - Send a text message to a pay-per-text number and they would respond with a ringtone or wallpaper or something else
- Some purchases could be made through platform holder services
  - V-Cast is a mobile application that allows users access and download various forms of entertainment and media from their cell phones.

# When there's money to be made...

- The Internet was full of media that people wanted to consume on the go
- Other handheld devices were selling like gangbusters (Game Boy)
- Phones seemed like an obvious next step
  - A computer that everyone carried with them and was always connected

# Bigger and bigger players get involved

- Nokia had a large portion of the mobile phone market early on
- Other players like Blackberry, Samsung, HTC, etc. also were involved
- Each had (basically) their own operating system, which made third-party development tricky
- What changed with phones?
  - Phones started running existing operating systems (Windows CE and Linux)
  - Mobile carriers started to relax the constraints on what phones could do

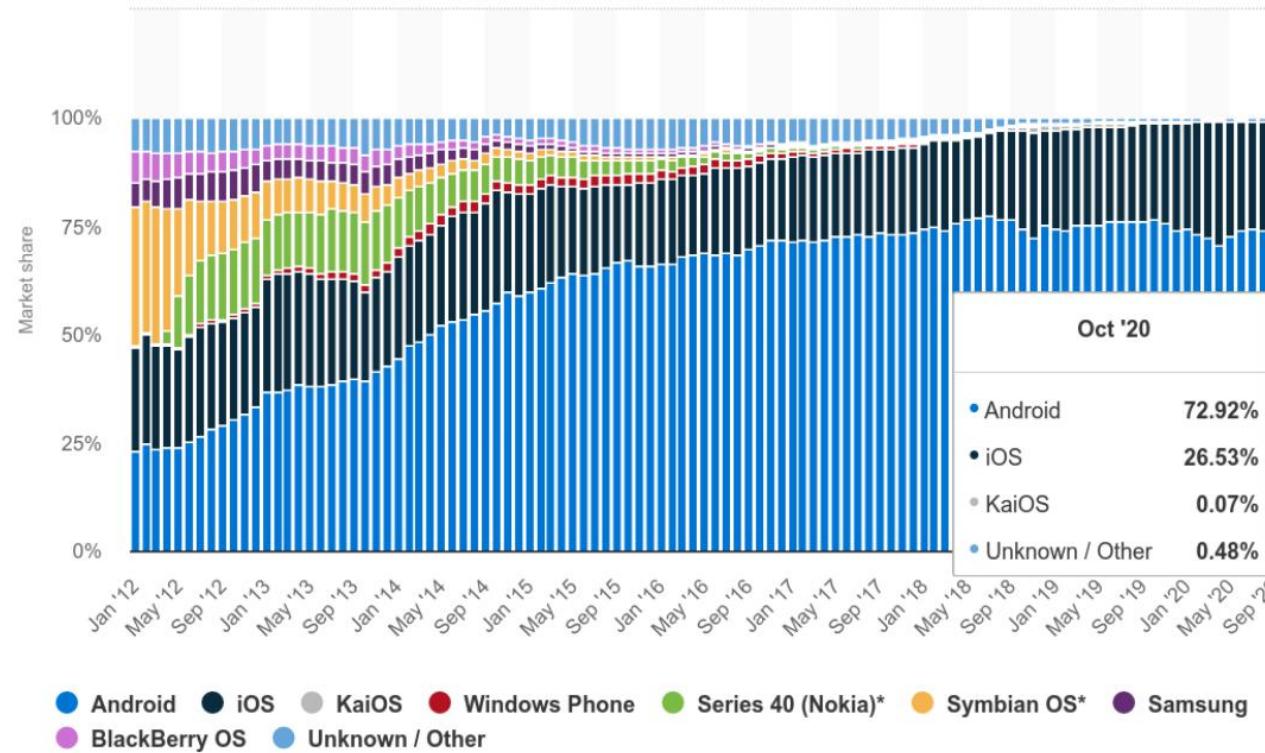
# The Market Fractures

- Microsoft
  - Tried to leverage “write once, run on any Windows device” a bit
  - Worked for a while with PDAs, but never really caught on with phones
- Apple
  - Started off as a phone that had a web browser and iPod bolted on
  - Evolved into much more once the App Store opened
- Google
  - Just provided the OS and let others build the devices (for a while)
  - Open source OS + no developer fees = lots of interest and apps

# Two Main Operating Systems Remain

- iOS
  - iPhones and iPads only
  - Objective-C or Swift using Xcode OR third party platforms (Unity, Xamarin, etc.)
  - Tightly controlled
- Android
  - Thousands of different devices of all shapes and sizes
  - Java using Android Studio OR third party platforms (Unity, Cordova, etc.)
  - Open Source, available to everyone

# Mobile OS Market Share



- Android: 72.9%
- IOS: 26.5%



Android Tablet



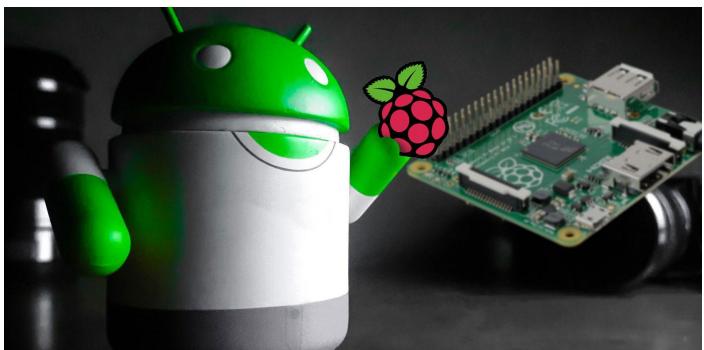
Android Auto



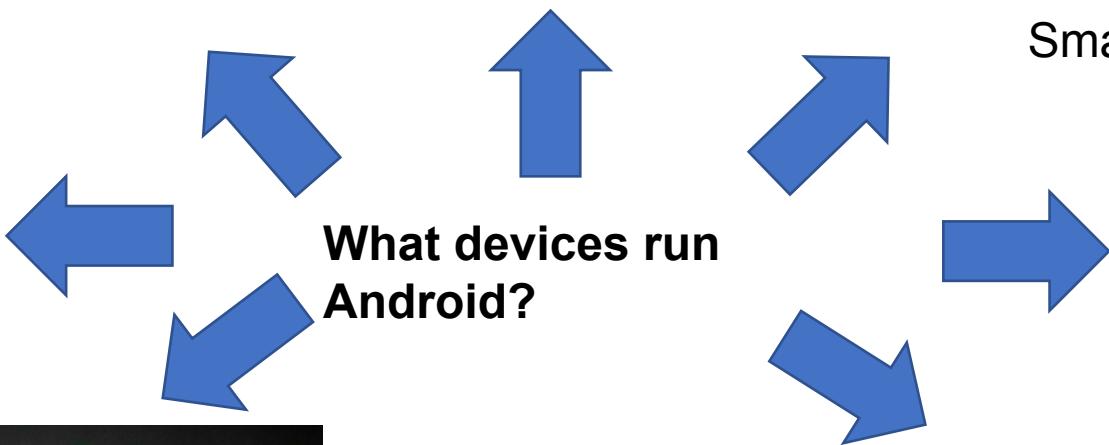
Smartwatch



Google Glass



Embedded devices  
(e.g. Raspberry Pi)



Android TV



Smartphone

# Why Android?

- Contains rich mobile and ubicomp programming modules
  - Sensors: GPS, microphone, camera, IMU, ...
  - Data processing: machine learning
  - Application: activity recognition, bio-sign monitoring, speech recognition, ...

# Mobile Computing Concepts

# mo·bile

*adjective*

/'mōbəl, 'mō,bīl/

1. able to move or be moved freely or easily.

"he has a major weight problem and is not very mobile"

*synonyms:* able to move (around), **moving**, walking; **motile**; **ambulant**

# Main Classes of Mobile Computing

- Mobile Phones
  - Initially focused on voice calls
  - Increasingly adding computational capacities



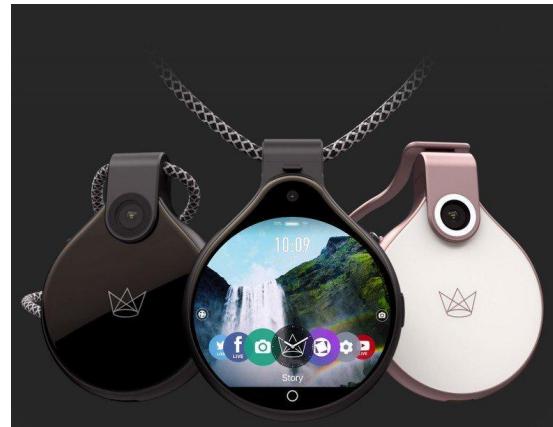
# Main Classes of Mobile Computing

- Mobile Phones
- Portable Computers
  - Portable computers are devices with only essential computing components and input/output devices
  - E.g., laptops, notebooks, tablets, notepads
  - lighter in weight than desktops, through removal of nonessential input/output devices like disc drives, use of compact hard drives, and so on.



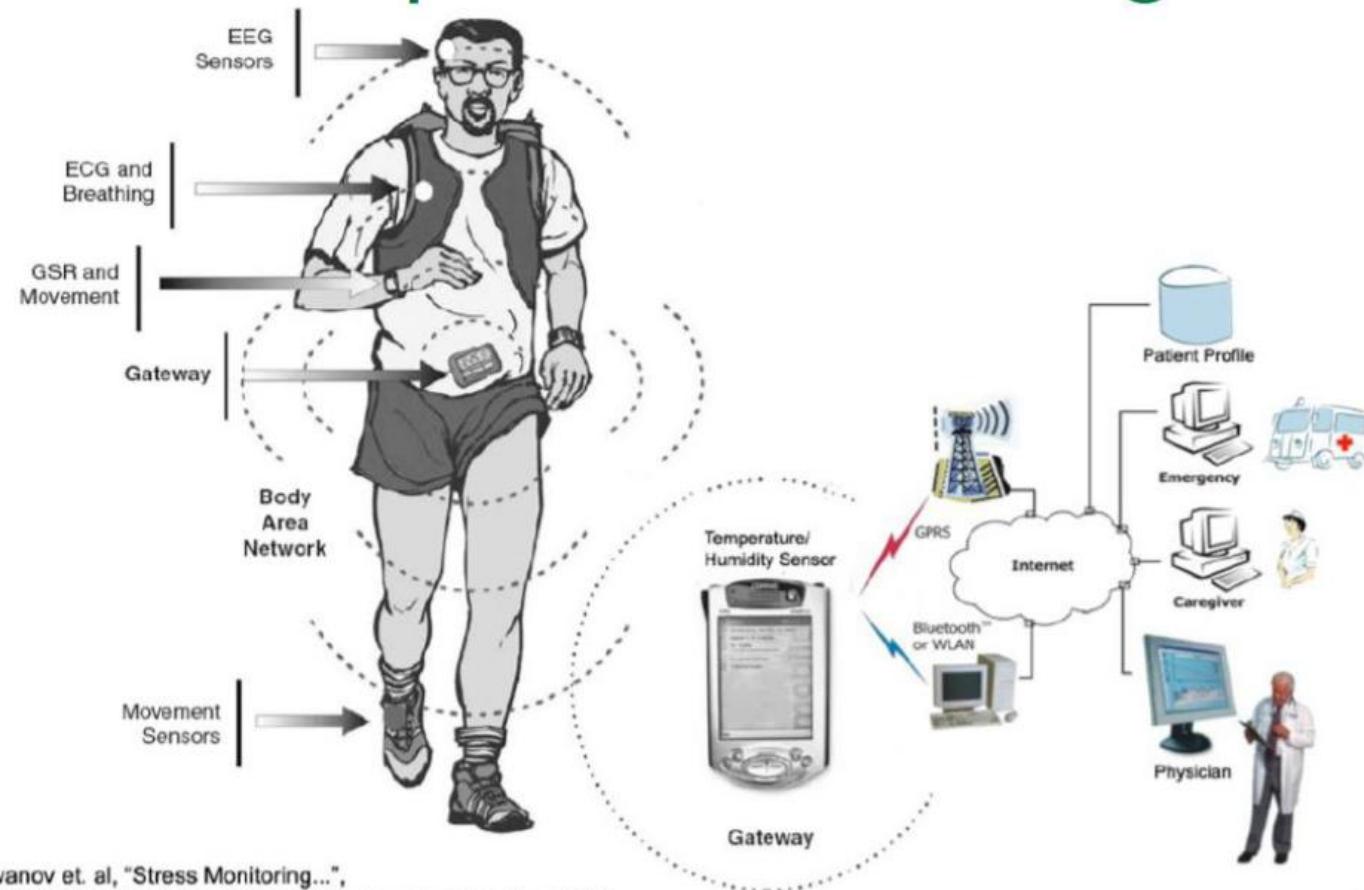
# Main Classes of Mobile Computing

- Mobile Phones
- Portable Computers
- Wearable computers
  - Devices put on the body for computation/connection as well as fashion
  - Capable of sensing, computing, reporting, and connecting
  - E.g., watches, necklaces, implants



# Ubiquitous Computing: Wearable sensors for Health

## remote patient monitoring



# More Mobile Computing Devices



*Body Worn  
Activity Trackers*



*Bluetooth  
Wellness  
Devices*

# Smart Mobile Computing

- Smart = Networking + Computing + Sensing
  - **Sensors:** Camera, video, location, temperature, heart rate sensor, etc
  - **Computing:** Java apps, JVM, apps
    - Powerful processors: Quad core CPUs, GPUs
  - **Communication:** Talk, text, Internet access, chat

# Computing

# SmartPhone Computing Hardware

- Google Pixel XL phone: Quad core 1.6 GHz Snapdragon CPU, Adreno 530 GPU, 4GB RAM
  - A PC in your pocket!!
  - Multi-core CPU, GPU
  - Runs OpenGL ES, OpenCL and now Deep learning (Tensorflow)

# Sensing

# Smartphone Sensors

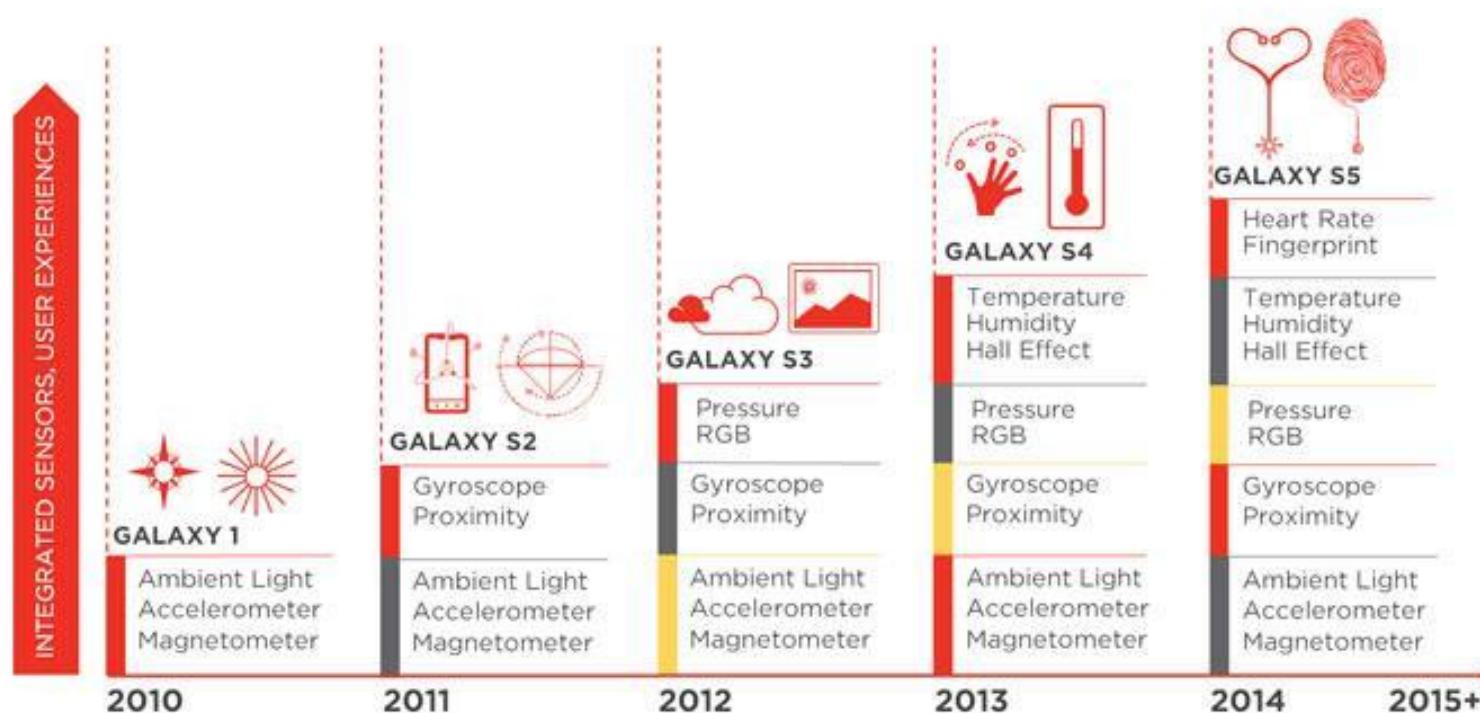
- Typical smartphone sensors today
  - accelerometer, compass, GPS, microphone, camera, proximity
- Can sense physical world, inputs to intelligent sensing apps
  - E.g. Automatically turn off smartphone ringer when user walks into a class



# Growth of Smartphone Sensors

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

## SENSOR GROWTH IN SMARTPHONES



More:

- LIDAR
- mmWave
- pollution sensor

# Sensor

- **Example:** E.g. door senses only human motion, opens
- **Sensor:** device that can sense physical world, programmable, multi-functional for various tasks (movement, temperature, humidity, pressure, etc)
- Device that can take inputs from physical word
  - Also includes camera, microphone, etc
- Ubicomp uses data from sensors in phone, wearables (e.g. clothes), appliances, etc.



(courtesy of MANTIS  
project, U. of Colorado)



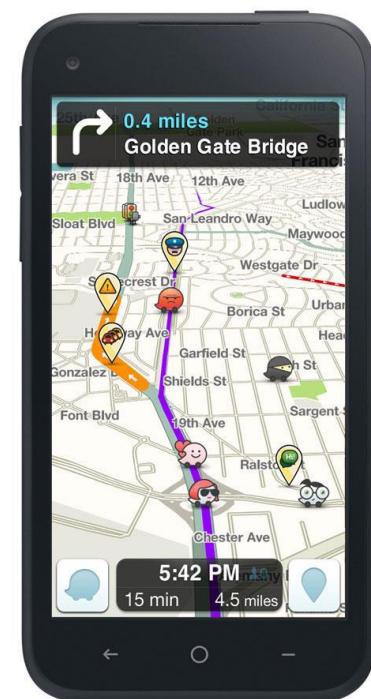
RFID tags



Tiny Mote Sensor,  
UC Berkeley

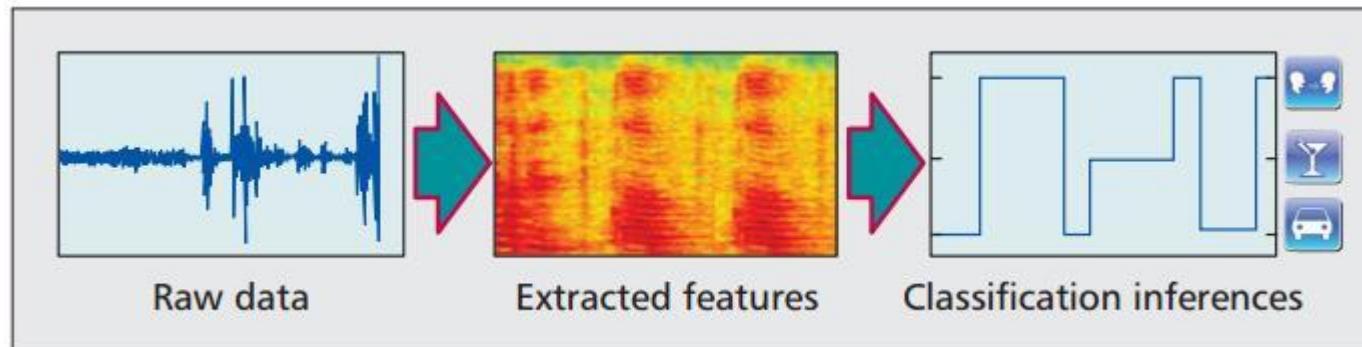
# Mobile Sensing

- Mobile devices can sense human, environment
- Example: Human activity sensing (e.g. walking, driving, climbing stairs, sitting, lying down)
- Example 2: Waze crowdsourced traffic

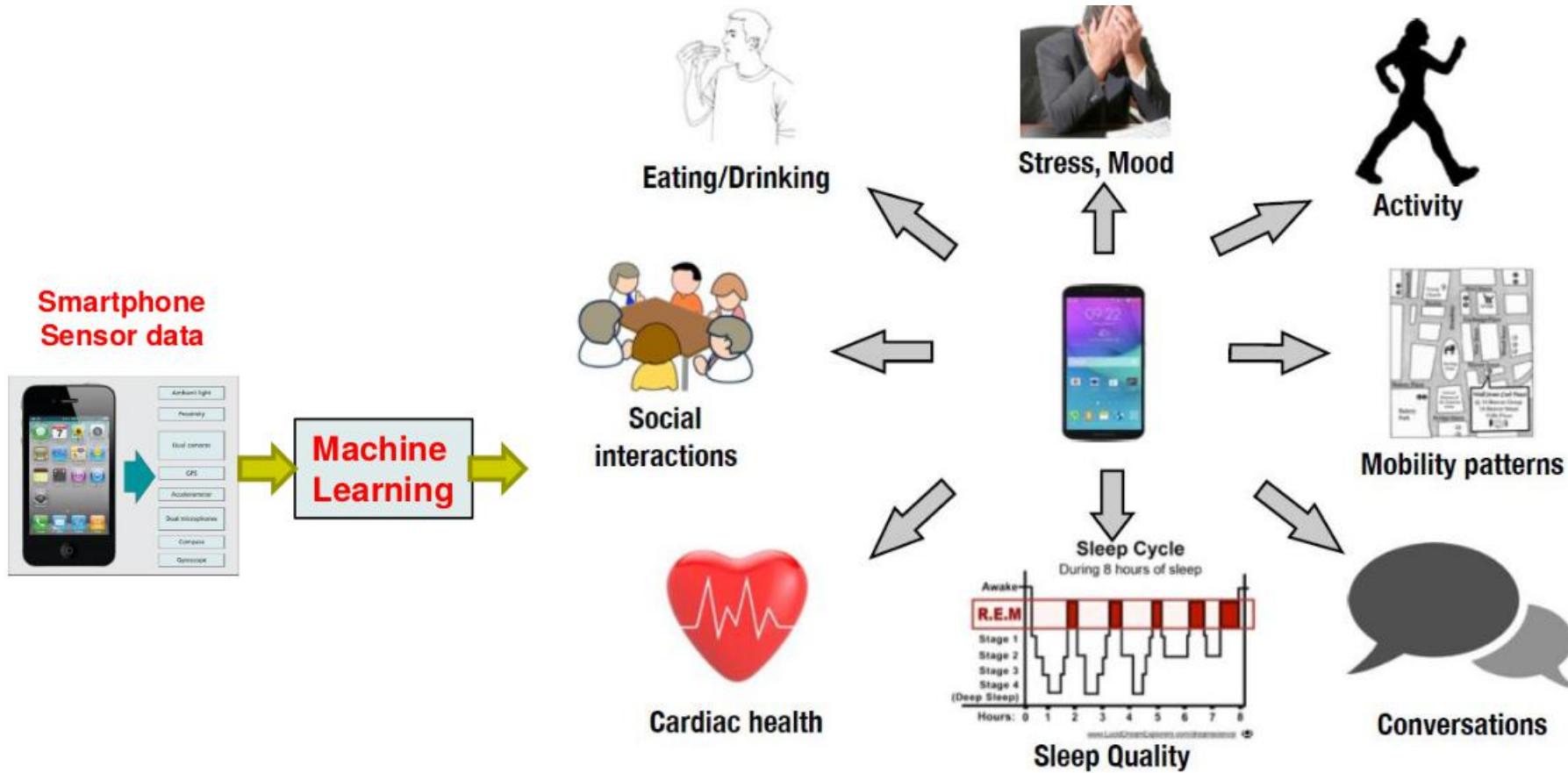


# Sensor Data Processing

- Machine learning commonly used to process sensor data
  - Action to be inferred is hand-labelled to generate training data
  - Sensor data is mined for combinations of sensor readings corresponding to action
- Example: Smartphone detects user's activity (e.g. walking, running, sitting,) by classifying accelerometer sensor data



# What can be detected by the phone?



# Networking

# Wireless Networks On a Phone

- Wi-Fi (802.11): (e.g. Starbucks Wi-Fi)
- Cellular networks: (e.g. Sprint network)
- Bluetooth: (e.g. car speaker)
- Near Field Communications (NFC)
  - e.g. Mobile pay: swipe phone at Dunkin Donuts



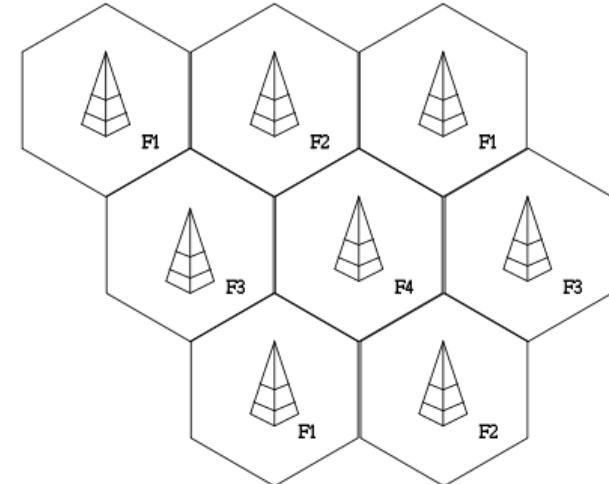
**WIFI**



**NFC**



**Bluetooth**



**Celluar Networks**

# Wireless Networks Comparison

<b>Network Type</b>	<b>Speed</b>	<b>Range</b>	<b>Power</b>	<b>Common Use</b>
WLAN	600 Mbps	45 m – 90 m	100 mW	Internet.
LTE (4G)	5-12 Mbps	35km	120 – 300 mW	Mobile Internet
3G	2 Mbps	35km	3 mW	Mobile Internet
Bluetooth	1 – 3 Mbps	100 m	1 W	Headsets, audio streaming.
Bluetooth LE	1 Mbps	100+ m	.01–.5 W	Wearables, fitness.
NFC	400 kbps	20 cm	200 mW	Mobile Payments