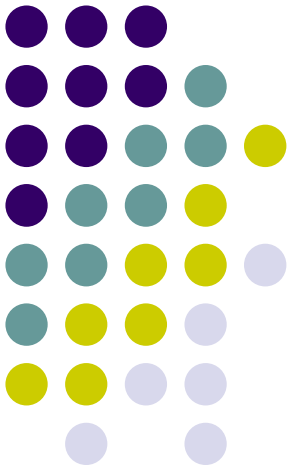
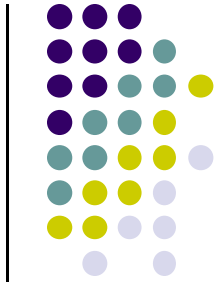
CS 528 Mobile and Ubiquitous Computing
Lecture 1: Introduction

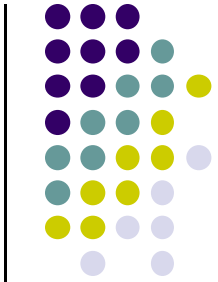
Emmanuel Agu



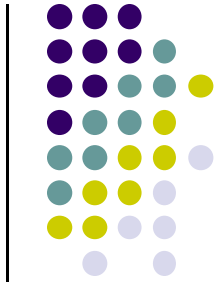


About Me

A Little about me

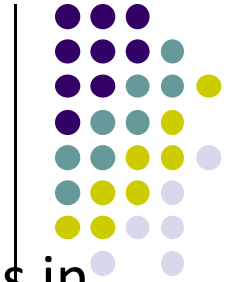


- WPI Computer Science Professor
- Research interests:
  - mobile computing especially mobile health, computer graphics
- How did I get into mobile and ubiquitous computing
  - 3 years in wireless research LAN lab
  - Group built working wireless LAN prototype (*pre 802.11*)
  - Designed, simulated, implemented wireless protocols
- Computer Systems/Electrical/Computer Science background
  - Hardware + software
- Current active research: Mobile health apps
  - E.g: AlcoGait app to detect how drunk Smartphone owner is



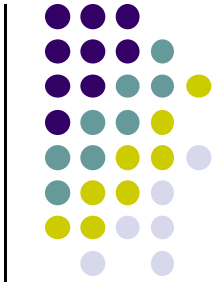
Administrivia

About this class



- **Class goal:** overview, insight into hot topics, ideas and issues in mobile and ubiquitous computing
- **Focus:** implement ideas on Android smartphone
- **Semester break:** March 9 (no class)
- **Website:** http://web.cs.wpi.edu/~emmanuel/courses/cs528/S16/
- **Projects:** 3 assigned, 1 big final project
- This class combines lots of other areas: (networking, OS, software, machine learning, programming, etc)
  - Most students don't have all the background!!
  - **Independent learning is crucial!**
  - **Final Projects:** Make sure your team has requisite skills

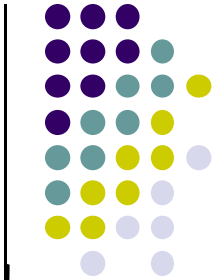
Administrivia: Schedule



- ***Week 1-6:*** I will present
  - Course introduction, Android programming
- ***Weeks 7 – 8:*** Students will present papers
  - **Goal:** understand cutting edge research ideas
  - Student talks short and direct (~15 minutes)
  - Discussions
- ***Week 9:*** Students propose final project
- ***Weeks 10-13:*** Students present more papers
- ***Week 14:*** Students present + submit final projects

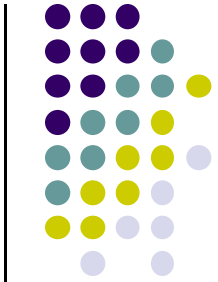
- **Breaks:** Each week, 15-min break halfway

Requirements to get a Grade



- **Seminar class:** Come to class + Discuss + Do good projects!!
  - Points for participation, discussion
- **Weeks 7-8, 10-13:** Student paper presentations
  - Each student will present 1 paper (maybe in groups?)
  - If not presenting, submit summaries for any 1 of week's papers
- **Projects**: 3 assigned and 1 final project(s)
- **Final project:** 5-phases (See website for deadlines)
  - Pick partner + decide project area
  - Brainstorm on ideas
  - Submit proposal intro + related work + proposed project plan
  - Build, evaluate, experiment, analyze results
  - Present results + submit final paper (in week 14)
- **Grading policy:** Presentation(s) 15%, Class participation 6%, Assigned Projects 24%, Final project: 40%, Summaries: 15%

Written Summaries

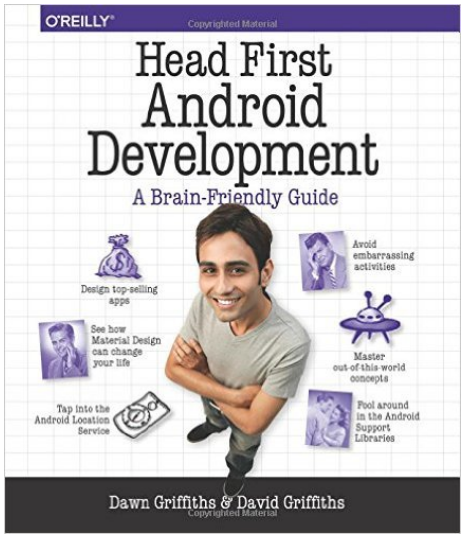


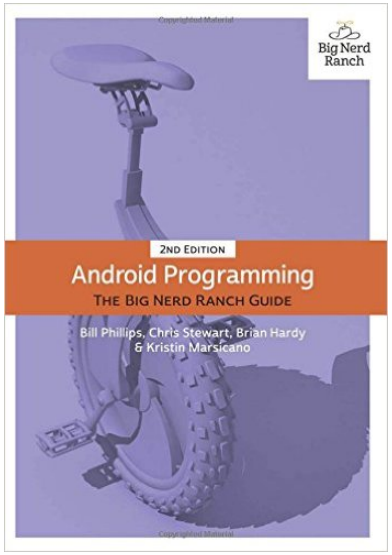
- Submit using turnin *before class*
- Summarize key points of any 1 of papers for week
  - Main ideas
  - Limitations of the work
  - What you like/not like about paper
  - Any project ideas?
- Half a page max per paper
- Summary should quickly refresh memory in even 1 year's time
  - Include main ideas/algorithms, results, etc.
- See handout for more details

Course Texts

- **Android Texts:**
  - *Head First Android Development,* Dawn and David Griffiths, O'Reilly, 2015
  - *Android Programming: The Big Nerd Ranch (Second edition)*, Bill Phillips and Brian Hardy, The Big Nerd Ranch, 2015

Gentle intro

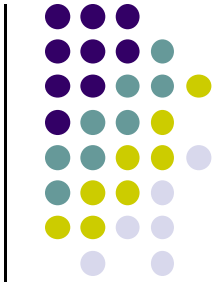




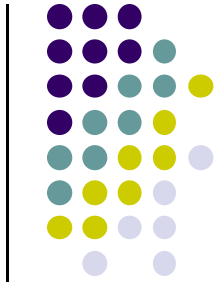
Bootcamp Tutorial

- Will also use official Google Android documentation
- Research papers: Why not text?

Poll Question



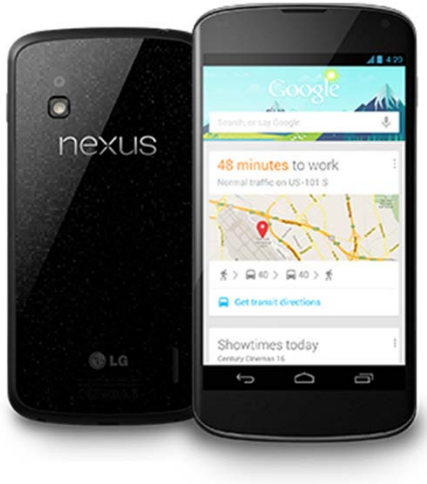
- How many students:
  1. Own recent Android phones (running Android 4.4, 5.0 or 6.0?)
  2. Can borrow Android phones for projects (e.g. from friend/spouse)?
  3. Do not own and cannot borrow Android phones for projects?



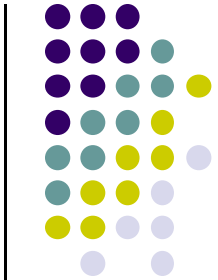
Mobile Devices

Mobile Devices

- Smart phones (Blackberry, iPhone, Android, etc)
- Tablets (iPad, etc)
- Laptops



SmartPhone Hardware



- **Communication:** Talk, text, Internet access, chat
- **Computing:** Java apps, JVM, apps
  - Powerful processors: Quad core CPUs, GPUs
- **Sensors:** Camera, video, accelerometer, etc
- Smartphone = Communication + Computing + Sensors
- Google Nexus 5 phone: Quad core 2.5 GHz CPU, Adreno 330 GPU

	Nexus 4	Galaxy S III	iPhone 5	Moto Droid
CPU	APQ8064	MSM8960	Apple A6	OMAP 3430
	1.7 GHz **Quad**-core	1.7 GHz **Dual**-core	1.3 GHz **Dual**-core	600 MHz
GPU	Adreno 320	Adreno 225	PowerVR SGX543MP3	PowerVR SGX 530
	OpenGL ES 3.0 **OpenCL 1.2** OpenVG 1.1	OpenGL ES 2.0 OpenVG 1.1	OpenGL ES 2.0 Shader Model 4.1	OpenGL ES 2.0 Shader Model 4.1
	NA **40-45 GFLOPS**	400 MHz **19.2 GFLOPS**	266 MHz (**Tri**-core) **25.5 GFLOPS**	200 MHz (1.6 GFLOPS)

GLOPS: floating-point operations per second

Comparison courtesy of Qian He (Steve)

Smartphone Sensors

- Typical smartphone sensors today
  - accelerometer, compass, GPS, microphone, camera, proximity

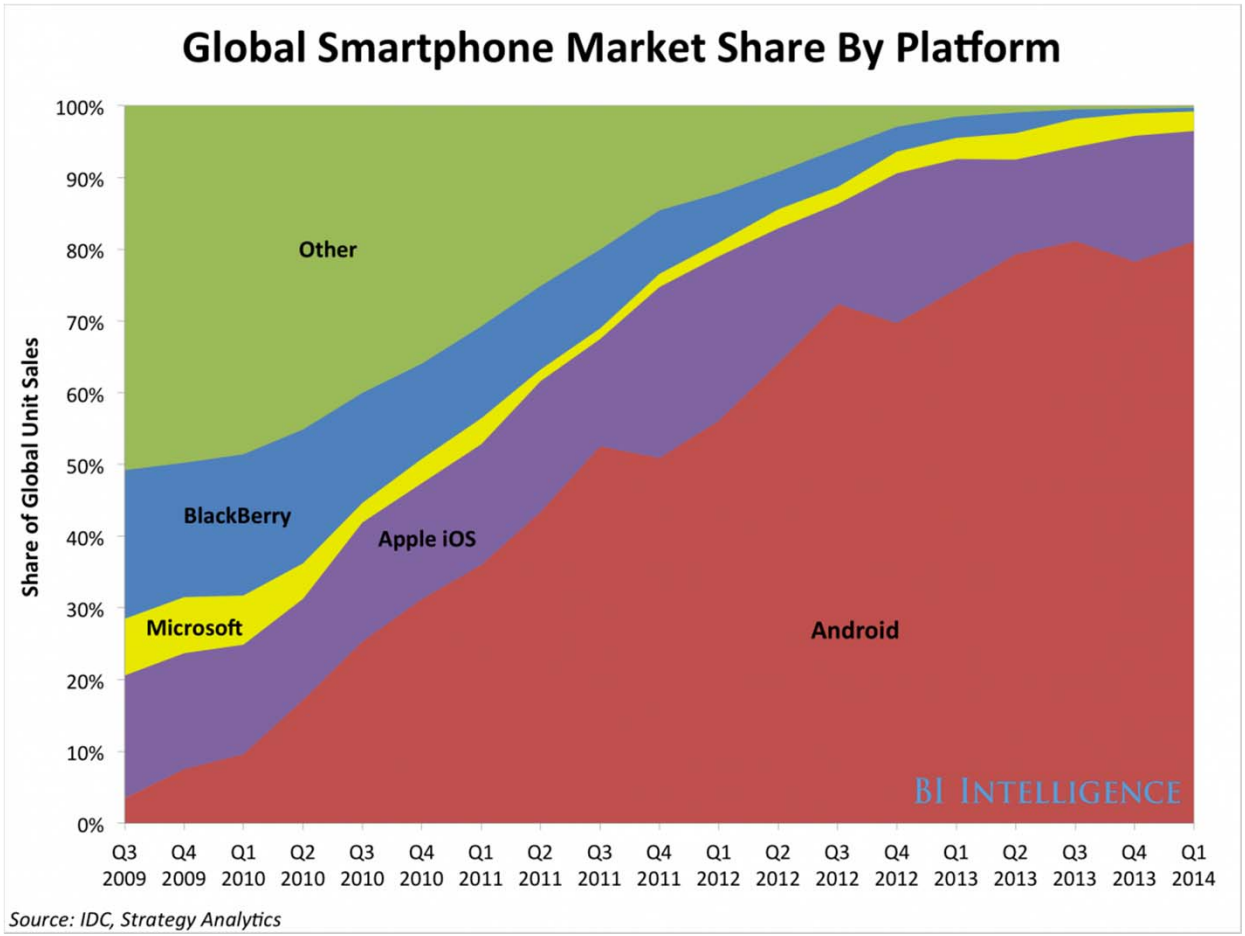
Future sensors?

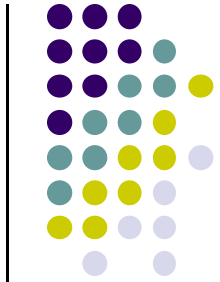
- Heart rate monitor,
- Activity sensor,
- Pollution sensor,
- etc

SmartPhone OS

- Over 80% of all phones sold are smartphones
- Android share 78% worldwide
- iOS 18%

Source: IDC, Strategy Analytics

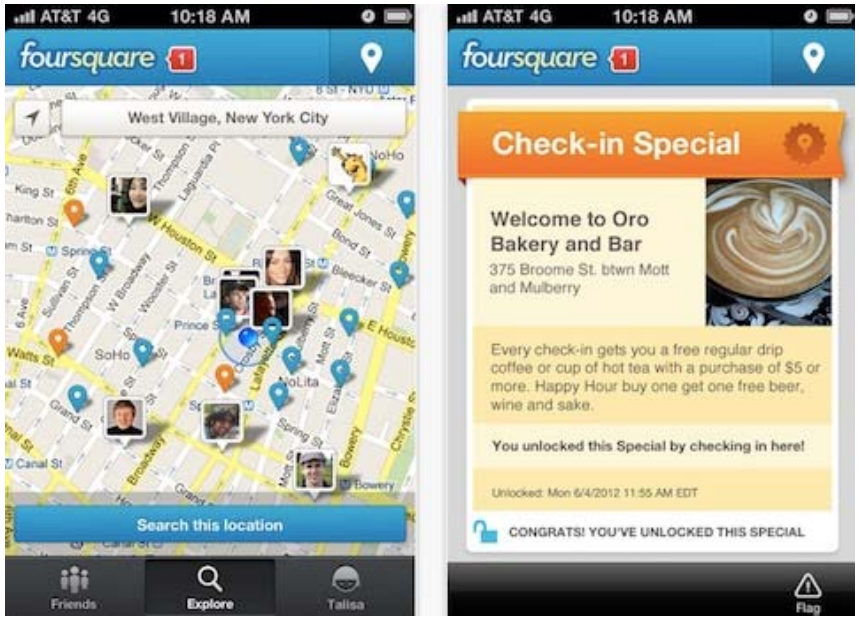




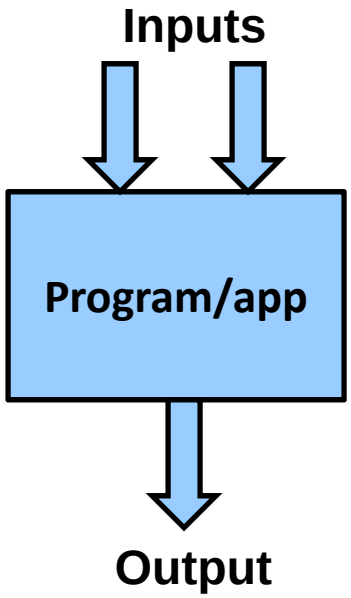
Mobile Computing

Mobile Computing

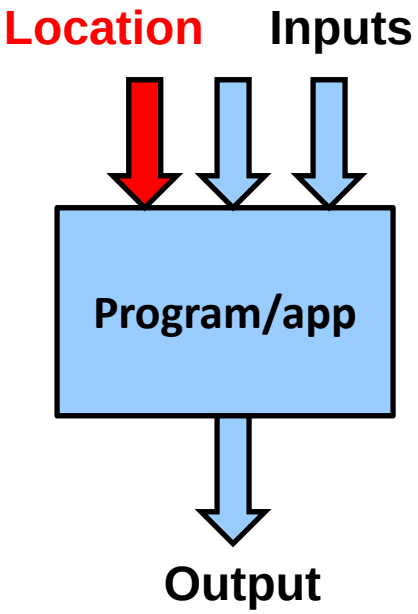
- Mobile? Human computes while moving, continuous network connectivity
- **Note:** Human initiates all activity, clicks on apps!!
- mostly *passive* network components
- **Example:** Using *foursquare.com* on smart phone



What does mobile mean?



Non-mobile app



Mobile app

- Mobile computing = computing while location changes
- Location (e.g) must be one of app/program's inputs
- Different user location = different output (e.g. maps)
- User in California gets different map from user in Boston

What does mobile mean?

- Truly mobile app must have different behavior/output for different locations
- Example: Mobile yelp
  - **Example search:** Find Indian restaurant
  - App checks user's location
  - Indian restaurants **close to user's location** are returned

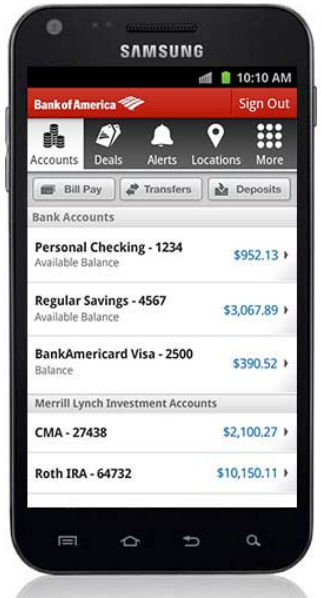
Example of Truly Mobile App: Word Lens

- Translates signs in foreign Language

- Location-dependent because sign location varies

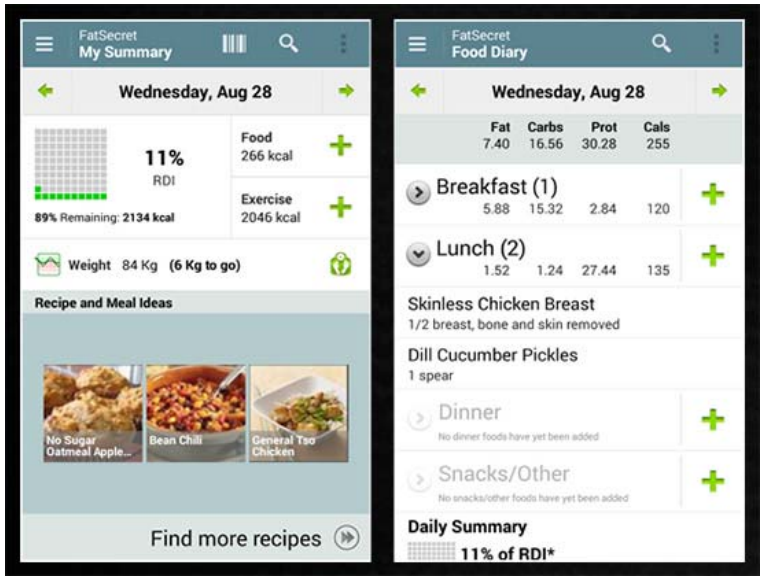
Some apps are not truly mobile?

- If output does not change as location changes, not truly mobile
- Apps run on mobile phone **just for convenience**
- Output does not change as location changes
- **Examples:**



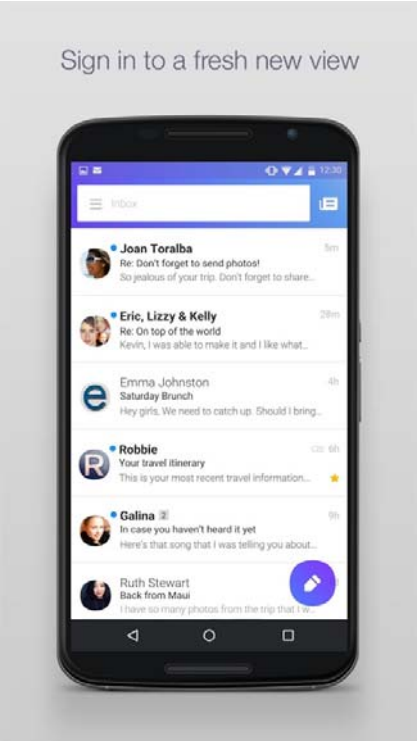
Mobile banking app

Internet Retailer app

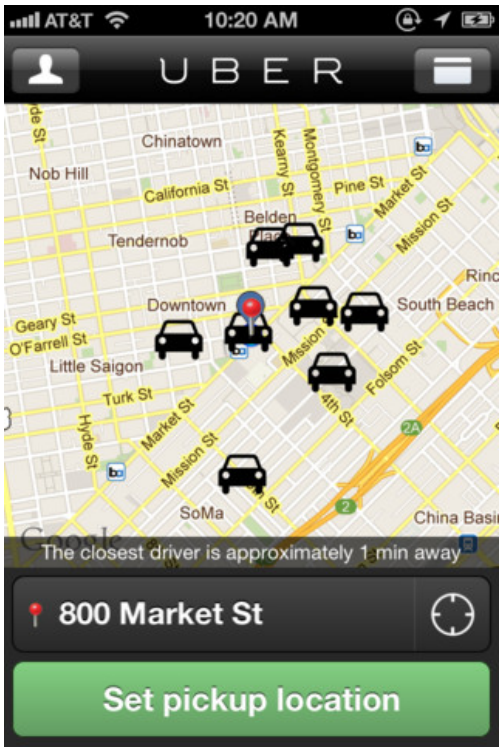


Diet recording app

Which of these apps are truly mobile?



a. Yahoo mail mobile



b. Uber app

Which of these apps are truly mobile?

c. Badoo dating app

Mobile Device Issue: Energy Efficiency

- Most resources increasing exponentially *except* battery energy (ref. Starner, IEEE Pervasive Computing, Dec 2003)

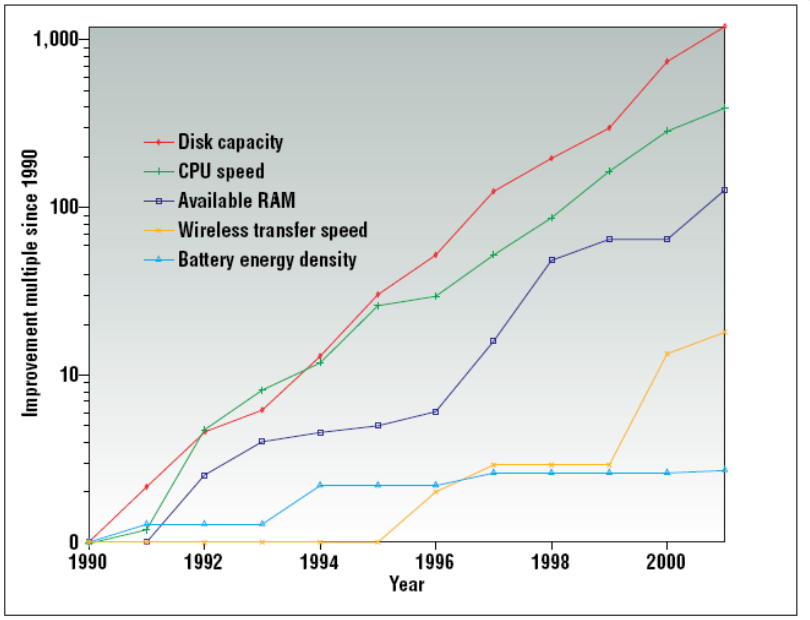
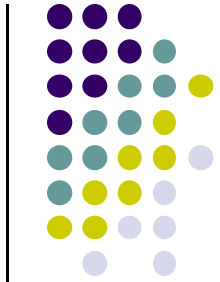


Figure 1. Improvements in laptop technology from 1990–2001.

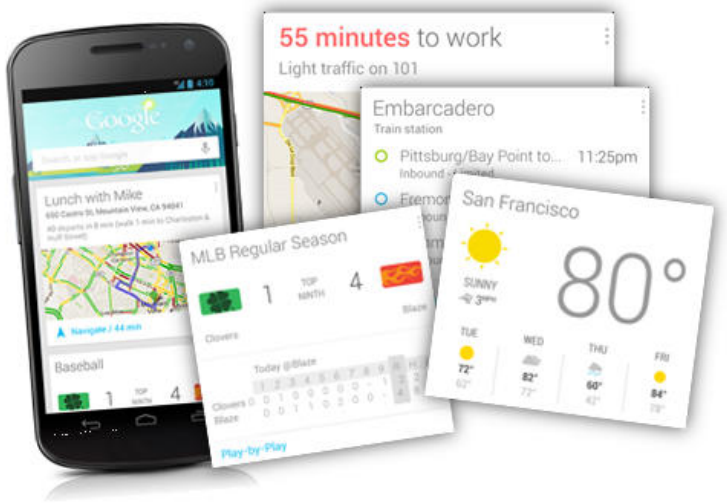
- Some Strategies:
  - **Energy harvesting:** Energy from vibrations, charging mats, moving humans
  - **Scale content:** Reduce image, video resolutions to save energy
  - **Better user interface:** Estimate and inform user how long each potential task will take
    - E.g: At current battery level, you can either type your paper for 45 mins, watch video for 20 mins, etc



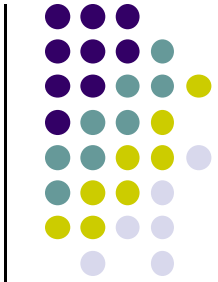
Ubiquitous Computing

Ubiquitous Computing

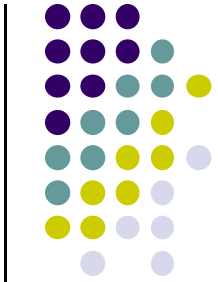
- Collection of specialized assistants to assist human in tasks (reminders, personal assistant, staying healthy, school, etc)
- Array of *active* elements, sensors, software, Artificial intelligence
- Extends *mobile computing* and *distributed systems* (more later)
- **Note:** System/app initiates activities, has intelligence
- **Example:** Google Now app



Ubicomp Senses User's Context



- Context?
  - *Human:* motion, mood, identity, gesture
  - *Environment:* temperature, sound, humidity, location
  - *Computing Resources:* Hard disk space, memory, bandwidth
  - *Ubicomp example:*
    - *Assistant senses:* Temperature outside is 10F (environment sensing) + Human plans to go work (schedule)
    - *Ubicomp assistant advise:* Dress warm!
- Sensed **environment + Human + Computer resources** = ***Context***
- *Context-Aware* applications adapt their behavior to context



Sensing the Human

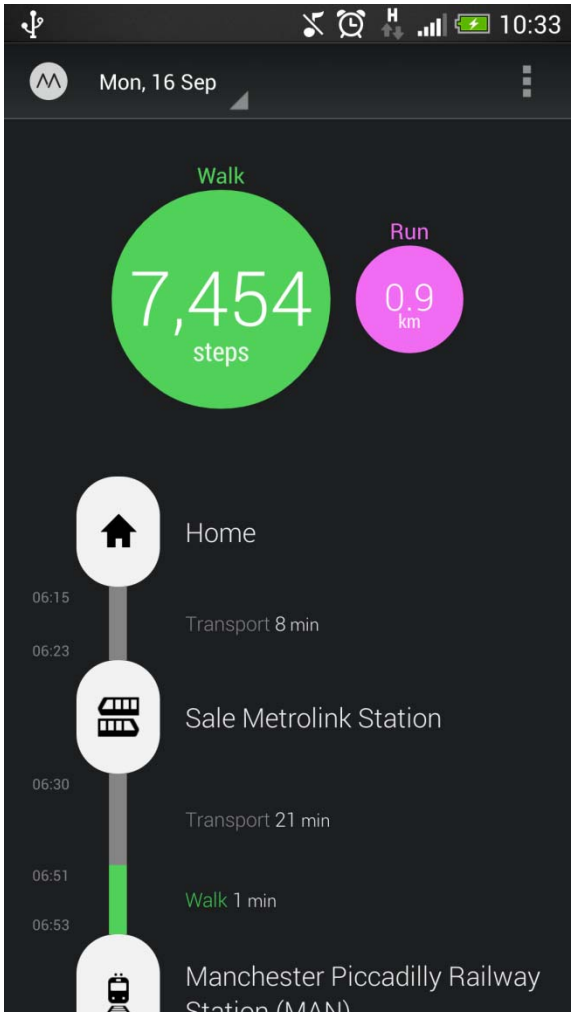
- Environmental sensing is relatively straight-forward
  - Use specialized sensors for temperature, humidity, pressure, etc
- Human sensing is a little harder (ranked easy to hard)
  - **When:** time (Easiest)
  - **Where:** location
  - **Who:** Identification
  - **How:** (Mood) happy, sad, bored (gesture recognition)
  - **What:** eating, cooking (meta task)
  - **Why:** reason for actions (extremely hard!)

5 W's + 1 H

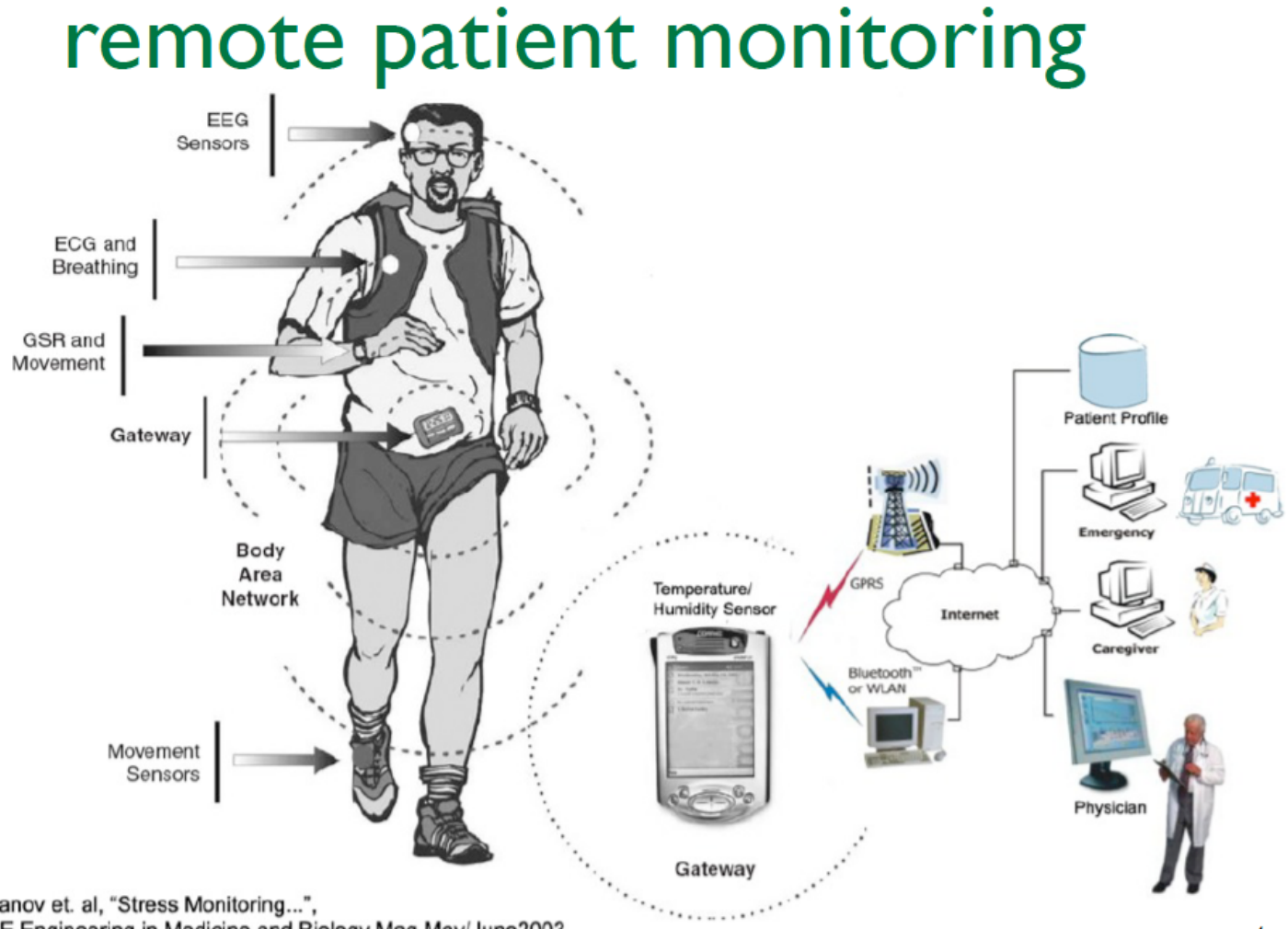
- Human sensing (gesture, mood, etc) easiest using cameras
- Research in ubiquitous computing integrates
  - location sensing, user identification, emotion sensing, gesture recognition, activity sensing, user intent

UbiComp Example: Moves App

- Counts Smartphone users steps through the day

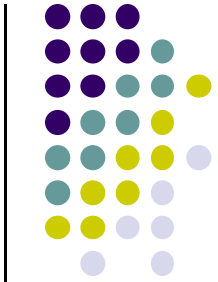


Ubiquitous Computing: Wearable sensors for Health



Jovanov et. al, "Stress Monitoring...",
IEEE Engineering in Medicine and Biology Mag.May/June2003

Ubicomp example
Context-Aware Search



- *[Hapori: Context-based Local Search for Mobile Phones using Community Behavioral Modeling and Similarity,* Nicholas D. Lane, Dartmouth College]

- Goal: Improves mobile search results using context, such as weather, age, profile of user, time, location and profile of other users to improve search.

- **Example:** a teenager gets a completely different set of recommendations from and elder.

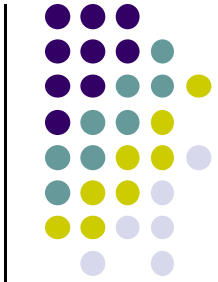
UbiComp: Wearables, BlueTooth Devices

Body Worn Activity Trackers

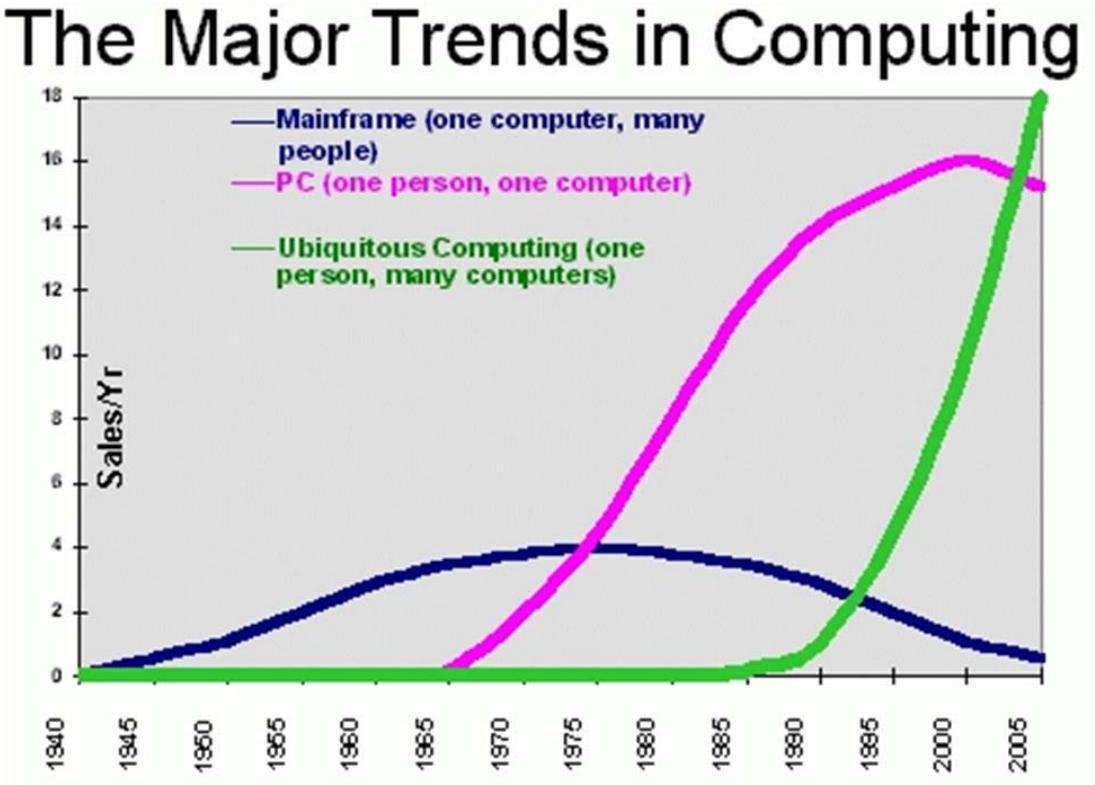
Bluetooth Wellness Devices

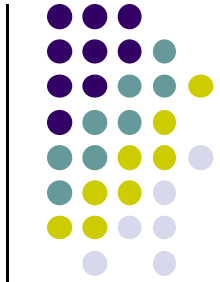
External sources of data for smartphone

A lot (Explosion) of Devices



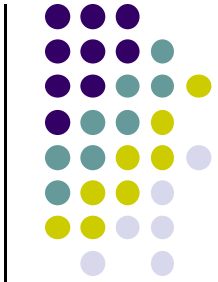
- *Recent Nokia quote:* More cell phones than tooth brushes
- Many more sensors envisaged
- ***Ubiquitous computing:*** Many computers per person





Definitions: Portable, mobile & ubiquitous computing

Definitions: Portable, mobile & ubiquitous computing



- ***Distributed computing:*** system is physically distributed. User can access system/network from various points. E.g. Unix, WWW. (huge 70's revolution)

- ***Portable (nomadic) computing:*** user intermittently changes point of attachment, disrupts or shuts down network activities

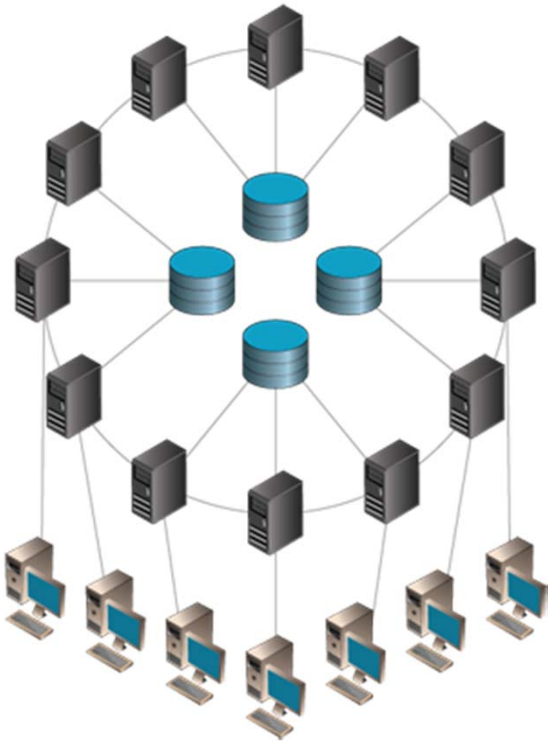
- ***Mobile computing:*** continuous access, automatic reconnection

- ***Ubiquitous (or pervasive) computing:*** computing environment including sensors, cameras and integrated active elements that cooperate to help user

- This class focus on **mobile and ubiquitous computing**

Distributed Computing

- ***Distributed computing example:*** You, logging in and web surfing from different terminals on campus (library, your dorm room, etc). Each web page consists of hypertext, pictures, movies anywhere on the internet.
- Note: network is fixed, Human moves



- Issues:
  - Remote communication (RPC),
  - Fault tolerance,
  - Availability (mirrored servers, etc)
  - Caching (for performance)
  - Distributed file systems (e.g. Network File System (NFS)
  - Security (Password control, authentication, encryption)

Portable (Nomadic) Computing

- ***Portable (nomadic) computing example:*** I own a laptop. Plugs into my home network, surf web while watching TV. In the morning, bring laptop to school, plug into WPI network, start up!
- Note: Network is fixed, device moves and changes point of attachment, no computing while moving

- Issues:
  - File/data pre-fetching
  - Caching (to simulate availability)
  - Update policies
  - Re-integration and consistency models
  - Operation queuing (e.g. emails while disconnected)
  - Resource discovery (closest printer while at home is not closest printer while at WPI)

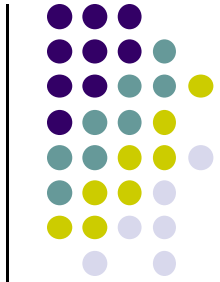
Mobile Computing Example

- ***Mobile computing:*** John owns SPRINT PCS phone with web access, voice, SMS messaging. He runs apps like facebook and foursquare and remains connected while walking around Boston

- Note: Network topology changes, because sarah and mobile users move. Network deals with changing node location
- Issues
  - Mobile networking (mobile IP, TCP performance)
  - Mobile information access (bandwidth adaptive)
  - System-level energy savings (variable CPU speed, hard disk spin-down, voltage scaling)
  - Adaptive applications: (transcoding proxies, adaptive resource resource management)
  - Location sensing
  - Resource discovery (e.g. print to closest printer)

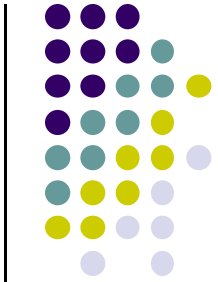
Ubiquitous Computing Example

- ***Ubiquitous computing:*** John is leaving home to go and meet his friends. While passing the fridge, the fridge sends a message to his shoe that milk is almost finished. When John is passing grocery store, shoe sends message to glasses which displays “BUY milk” message. John buys milk, goes home.
- **Core idea:** ubiquitous computing assistants **actively** help John
- Issues:
  - Sensor design (miniaturization, low cost)
  - Smart spaces
  - Invisibility (room million sensors, minimal user distraction)
  - Localized scalability (more distant, less communication)
  - Uneven conditioning
  - Context-awareness (assist user based on current situation)
  - Cyber-foraging (servers augment mobile device)
  - Self-configuring networks



More Examples of Topics/Ideas in this Class

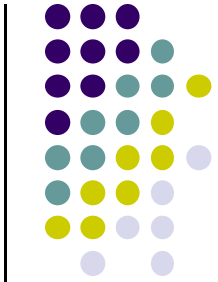
Mobile CrowdSensing



- **Internet of things:** Sensing data from consumer-centric devices including
  - Smartphones (iPhone, Google Nexus,)
  - Music players (iPods)
  - Sensor embedded gaming systems (Wii, Xbox, kinect)
  - In-vehicle sensors (GPS)
  - Body-worn sensors (e.g. fitbit, Nike+)

- **Mobile crowdsensing:** sense these devices
  - personal, community- and Internet-wide
- Sensing applications at community scale possible

Mobile CrowdSensing



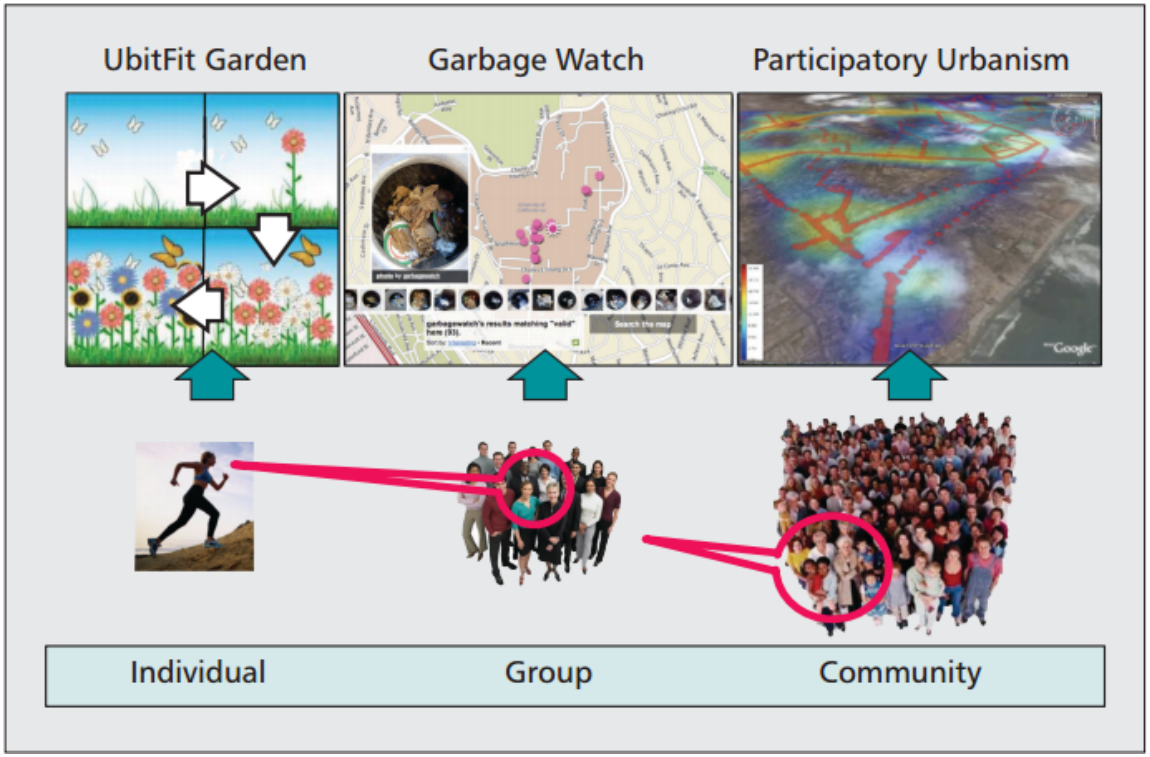
- **Personal sensing:** phenomena pertain to individual
  - E.g: activity detection and logging for health monitoring
- **Group:** friends, co-workers, neighborhood
  - GarbageWatch to improve recycling, neighborhood surveillance
- **Community sensing (mobile crowdsensing):**
  - Large-scale phenomena monitoring
  - Many people contribute their individual readings
  - **Examples:** Traffic congestion, air pollution, spread of disease, migration pattern of birds, city noise maps

Mobile CrowdSensing Types

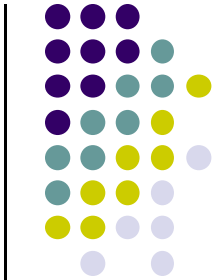
- **Participatory sensing: active** involvement of individuals (e.g taking a picture, reporting potholes)

- **Opportunistic sensing: passive** user involvement

(continuous location sampling without explicit user action)

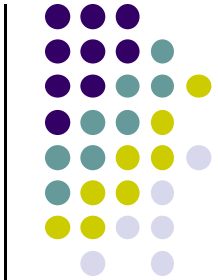


Mobile Crowd Sensing



- ***Classic example:*** Comparative shopping
- At CVS, ready to buy toothpaste. Is CVS price the best locally?
- Phone has software to query other members of my network
- People at other local stores (Walmart, Walgreens, etc) respond with prices

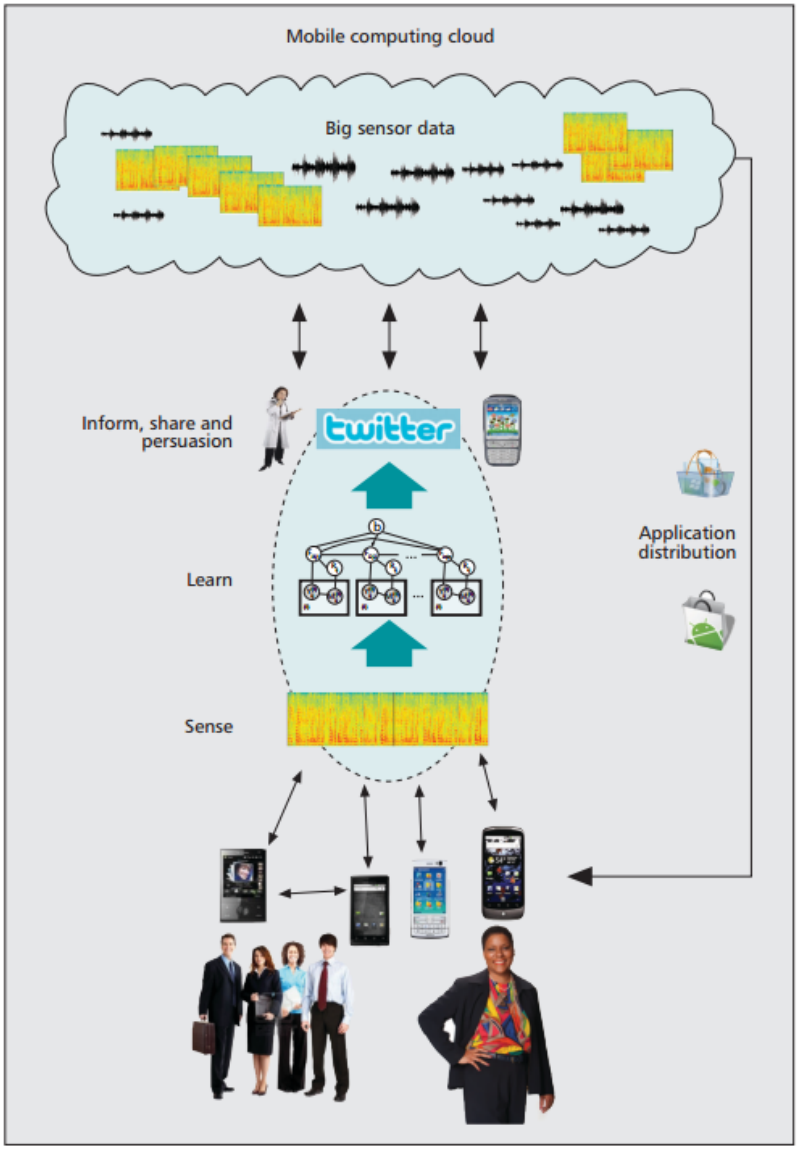
Sense What?



- **Environmental:** pollution, water levels in a creek
- **Transportation:** traffic/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

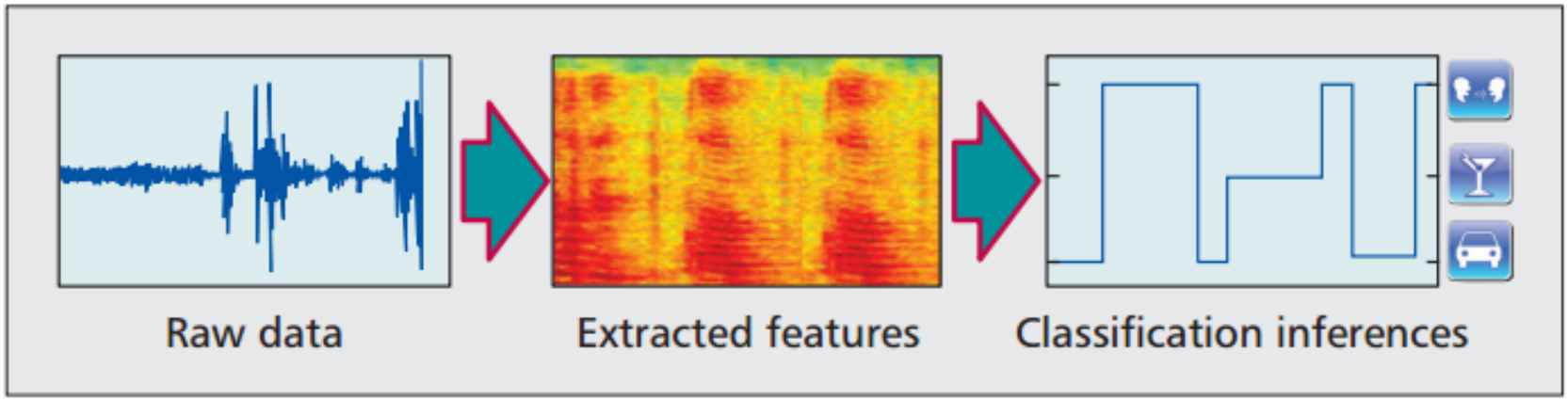
Mobile Phone Sensing Architecture

- **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

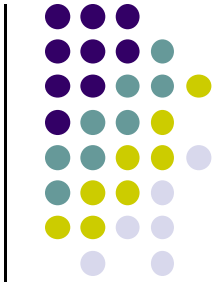


Sensor Processing

- **Machine learning** commonly used to process sensor data
  - Action to be inferred is hand-labelled to generate training data
  - Actual data is mined for combinations of sensor readings corresponding to action



Sensing Human Behavior



- *[Social Sensing for Epidimiological Behavior Change,* Anmol Madan et al, MIT Media Lab]

- **Goal:** infer how falling sick affects the [mobile/network] behaviors of human beings.
- **Examples:** Changes in call rates or visiting low entropy places more could mean person is sick
- Statistics of number of calls, co-location, proximity, WLAN and bluetooth entropy found to be good predictors of illness.
- Findings could be used as an early warning tool.
- If strong inference, then nurse could call the person
- **This work was basis for Venture funded company Ginger.io**

Mobile Computing: Measurement Studies

- How, when, where existing apps, mobile web, are being used
- Example: Where users engage in mobile commerce in UK