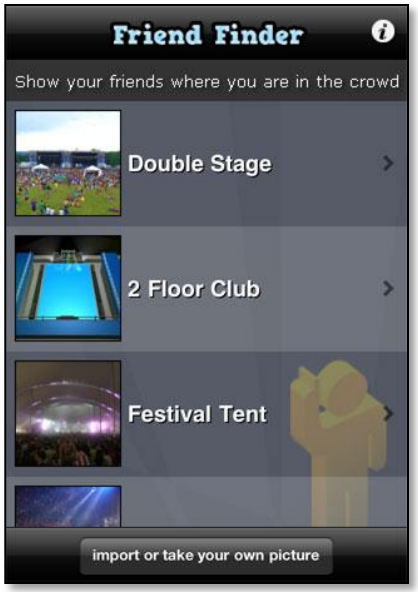
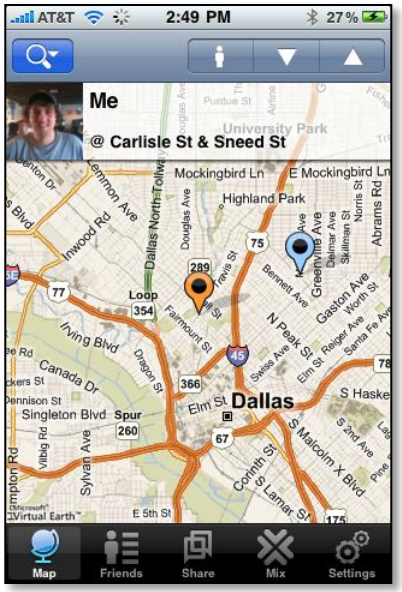
Leveraging Cloud for Distributed Mobile Apps

Suman Nath

Microsoft Research

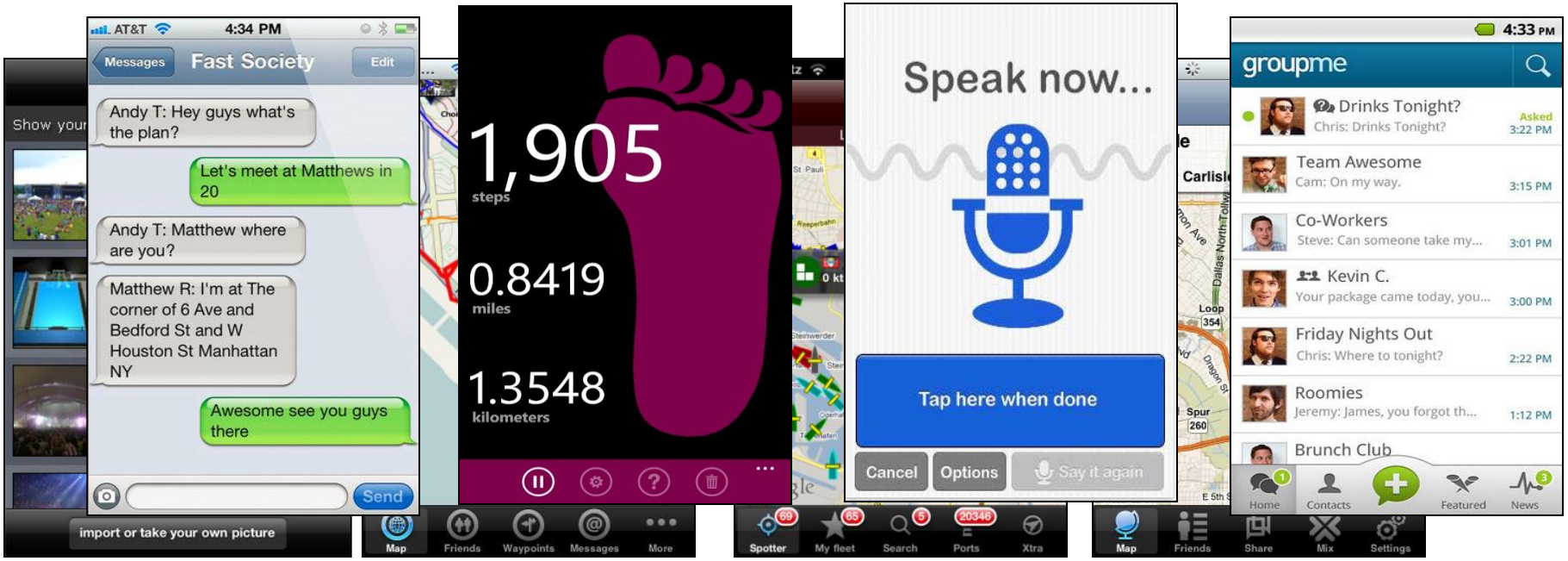
A new class of applications …





- Find where my friends are now
- Alert me whenever my friends are near me
- Send coupons when near Seattle Starbucks

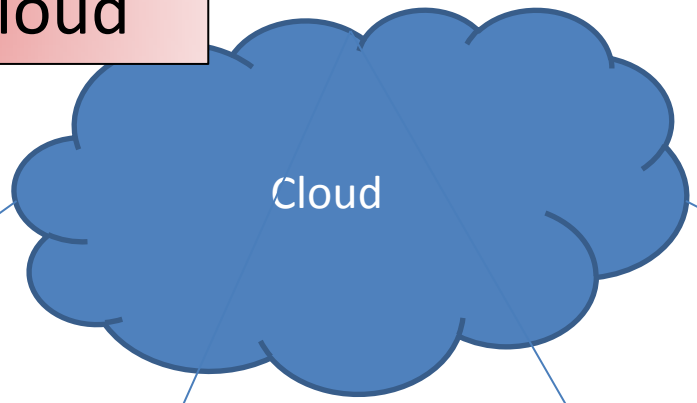
Examples



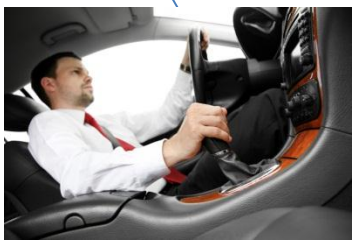
Many such apps in iOS, Android, and Windows Phone marketplaces

Context-aware Cloud-edge Apps

powerful cloud



continuous connectivity



Sensors-equipped edge-devices

Contexts: location, group, activity

CACE Platform

- Context-aware Cloud-edge app platform
- Goals:
  - Simplify programming
  - Rich, efficient runtime
  - Reliable, privacy-preserving context
  - …
- Current prototype: runs on Windows Azure and Windows Phone

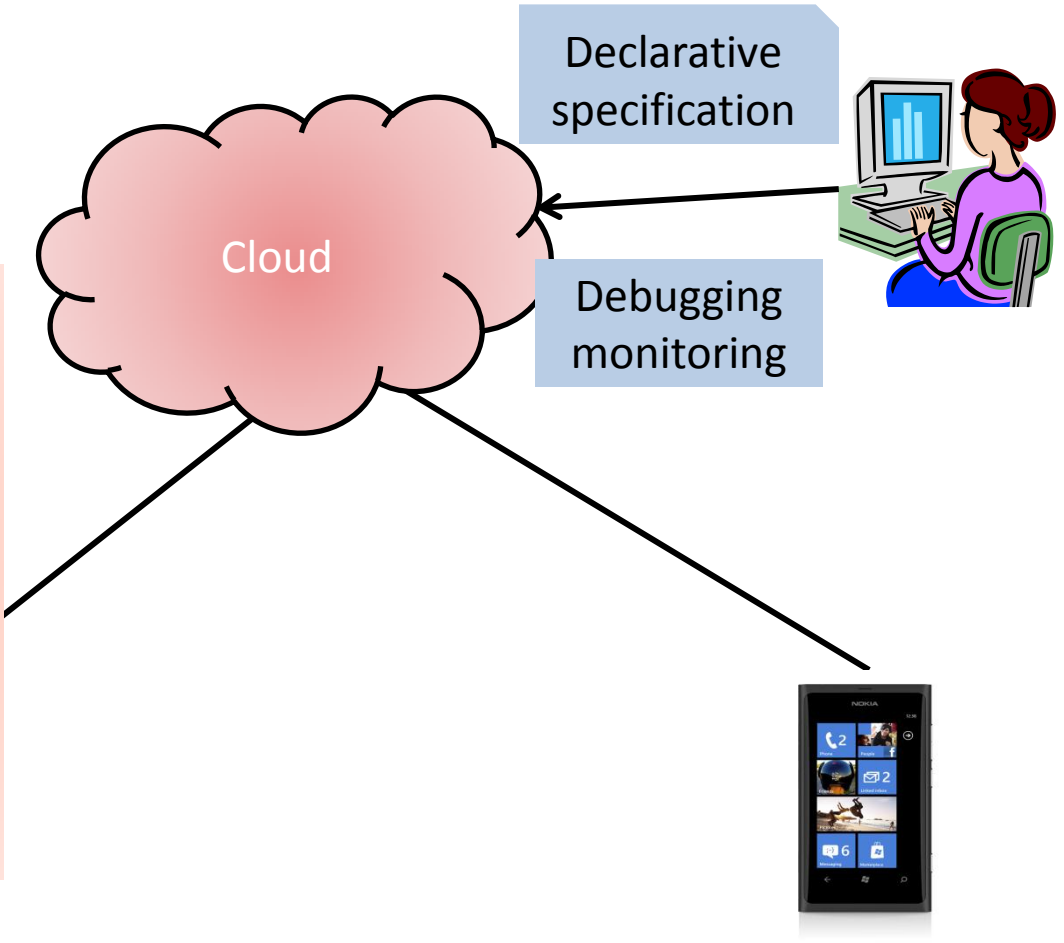
CACE Platform

Simple programming model

Abstracted as continuous queries on streaming data

Temporal SQL
(Microsoft StreamInsight LINQ)

15 lines of code for FriendFinder

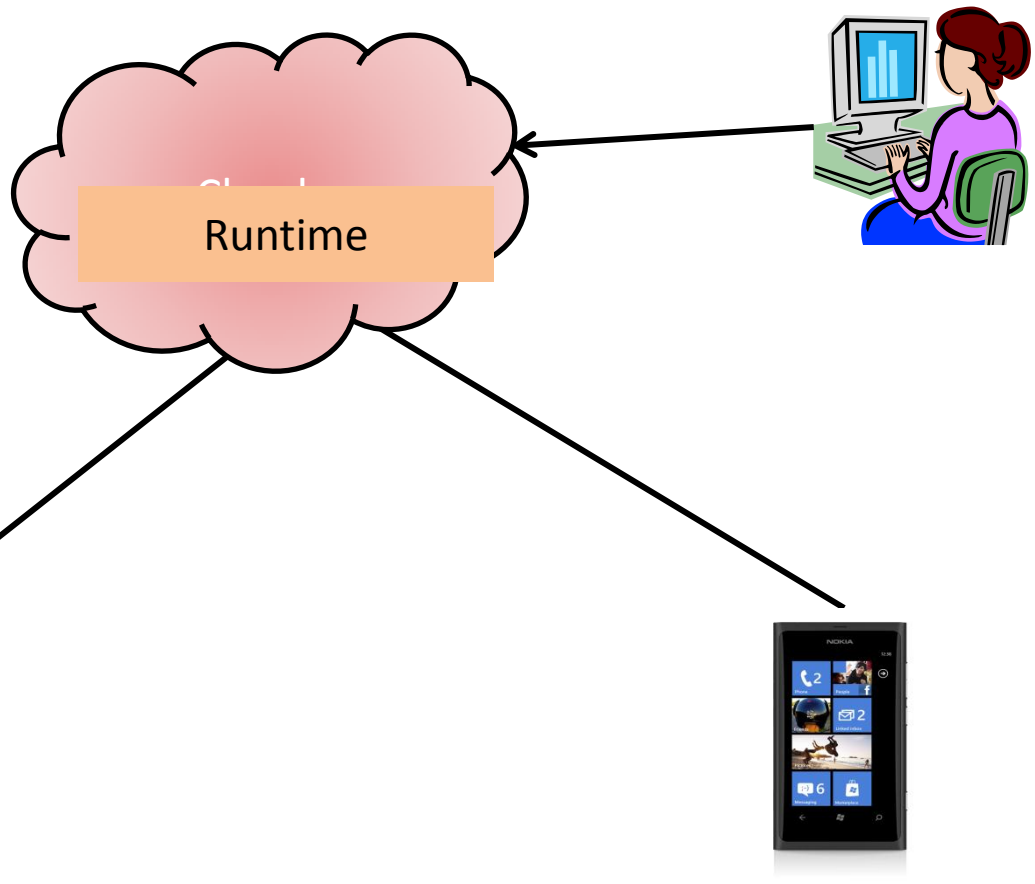


CACE Platform

Simple programming model

Runtime optimization for energy, bandwidth, etc.

Dynamically move tasks among phones and Cloud to minimize resource usage



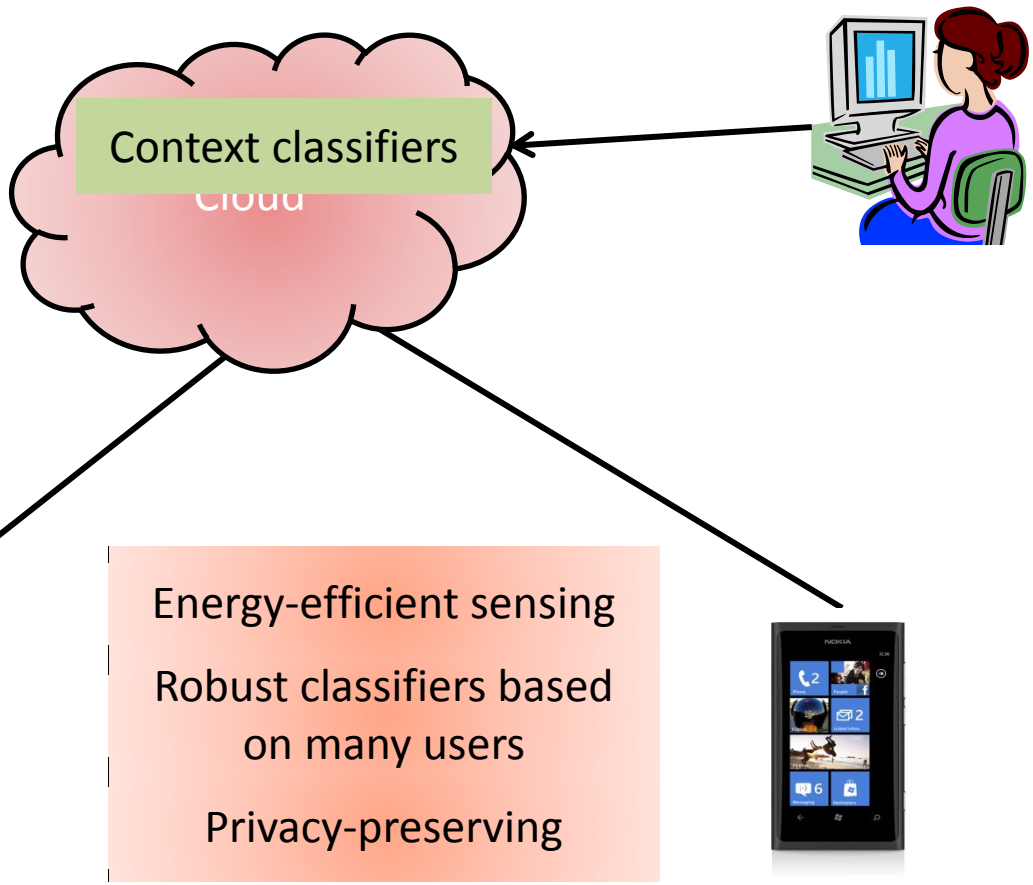
CACE Platform

Simple programming model

Runtime optimization for energy, bandwidth, etc.

Robust context inference

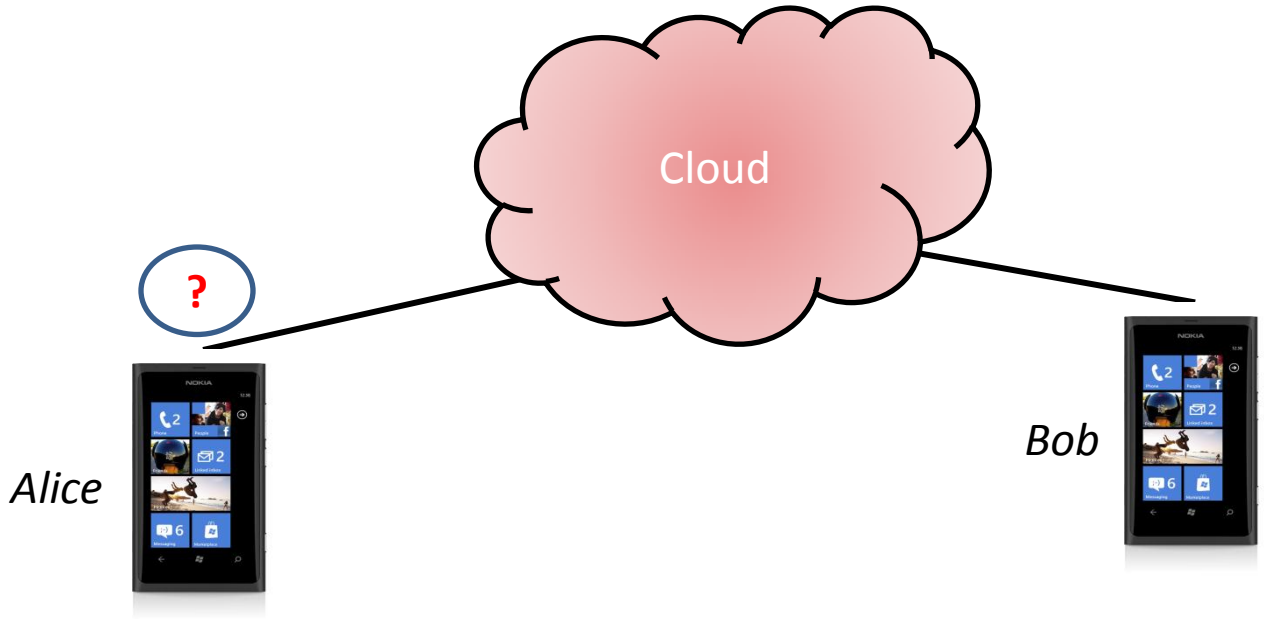
Privacy and security



Cloud is the key (*We are from Seattle!*)

- Rendezvous point for social apps
  - 'Emulate' phone-to-phone communication
- Data collection place
  - Collect data from many users to train robust classifiers
    - Train voice models to identify who with you
- Offloading expensive tasks
  - Static: e.g., speaker identification
  - Dynamic

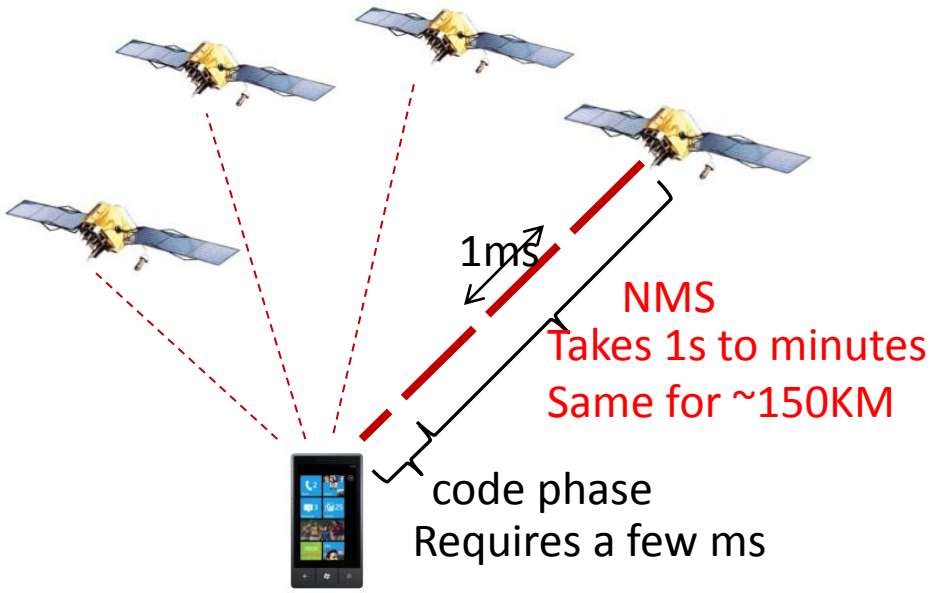
Offloading tasks



Where to run task? Cloud? Which phone?

Not trivial for large social graph

Offloading sensing: GPS



- Mobile phone sends to server:
  - Code phases
  - Cell tower ID
  - Time stamp
- Server:
  - Computes NMS
  - Computes mobile location

- Not regular GPS replacement

- Location-based services (e.g. mobile search)
- Batched location estimation (e.g. path prediction)
- Delay-tolerance positioning (e.g. geo-tagging photos)
- Crowdsourcing

LEAP: A Low Energy Assisted GPS for Trajectory-Based Services, Ramos et al. Ubicomp 2011

Cloud is helpful for …

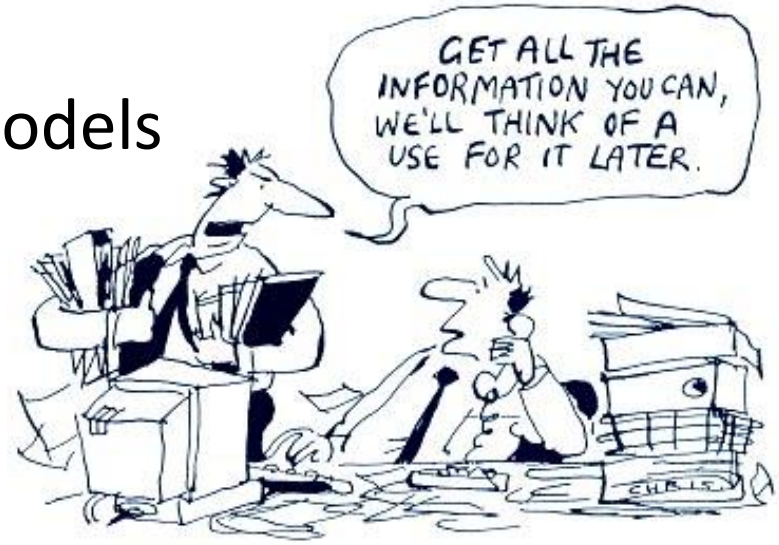
- Inter-device communication
- Data collection, training
- Offloading computation and sensing
- Inter-device synchronization
- Runtime visibility of apps
- Integration with existing cloud-based services
- Etc.

Challenge: Simple Programming

- Simple programming model
  - Today: C++, Java, C#
- Debugging tools
  - Limited for distributed apps
- Runtime monitoring
  - Today: Flurry in sourcecode, on single phone app

Challenge: Data

- Lack of data
  - to train context inference models
- Quality of data
- Lack of incentives
- Possible solution:
  - Monitory incentive (like many crowdsourcing platform)
  - Not clear how to price data



Challenge: Privacy Guarantee

- No intuitive definition of privacy
- Guarantee privacy without hurting utility much
  - 2/3 rule: privacy, utility, efficiency

"Sorry, I can't discuss my childhood. Before I could leave home I had to sign a non-disclosure agreement."

- Trickier for context-aware apps
- Give user control over her data

My wishlist

- Simplify programming, debugging, and runtime visibility
- Without affecting my phone battery much
- Train classifiers with massive, good-quality data
- Respect my privacy
  - Let me control my data, let me revoke if needed

Thank you