UCL

## GZ06 Adaptive and Mobile Systems
## Dr. Cecilia Mascolo

### Cartel: A Distributed Mobile Sensor Computing System

B. Hull, V. Bychkovsky, Y Zhang, K. Chen, M. Goraczko, A. Miu, H. Balakrishnan, S Madden

MIT

UCL

## Cartel Motivation

- Technology push: thousands of sensors available embedded in computers and phones
- There is a need for cheap monitoring applications
  - Wrt to instrument with static sensors, use mobile sensors
    - Bigger geographical areas covered
    - No need for static infrastructure put in place

## Applications

- Smart route planning and traffic congestion control
  - Cars with GPS gather fine-grained delay information
- Pre-emptive automobile maintenance
  - On-board + additional sensors for anomaly detection
  - Vibration, acoustic, chemical,…
- Civil infrastructure monitoring
  - Detect problems on roadways (e.g., potholes)
  - Vibration, image (camera), …
- Peer-to-peer car networks
- Cars as data delivery networks for “in the field” sensornets

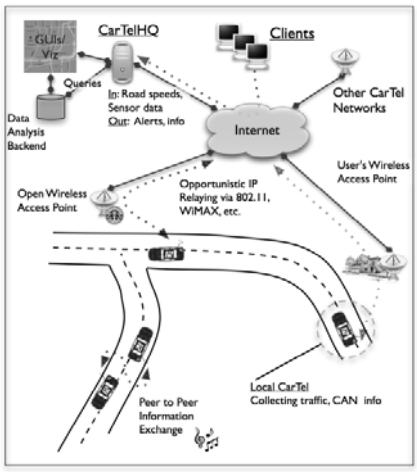
## What is Cartel?

- System for data collection, processing, delivery and visualization of remote, mobile and intermittently connected sensor nodes.
- Simple centralized programming interface
- Handles large volumes of heterogeneous sensor data
- Copes with intermittent network connectivity
- Application in automotive sensor networks
  - Energy is not an issue in this sensor network!!

## Contributions

- Simple programming interface
  - Cartel apps can be written as web apps (distribution and mobility are hidden)
- Large amounts of heterogeneous data
  - Any kind of sensors could be plugged
  - Media rich devices data (large amounts) should be buffered and processed on mobile nodes
- Intermittent connectivity
  - Opportunistic wireless: eg wifi AP passing by and therefore intermittent
  - Additional storage devices (USB) as data mules
  - P2P data exchange via ad hoc networks (Bluetooth or Wifi)

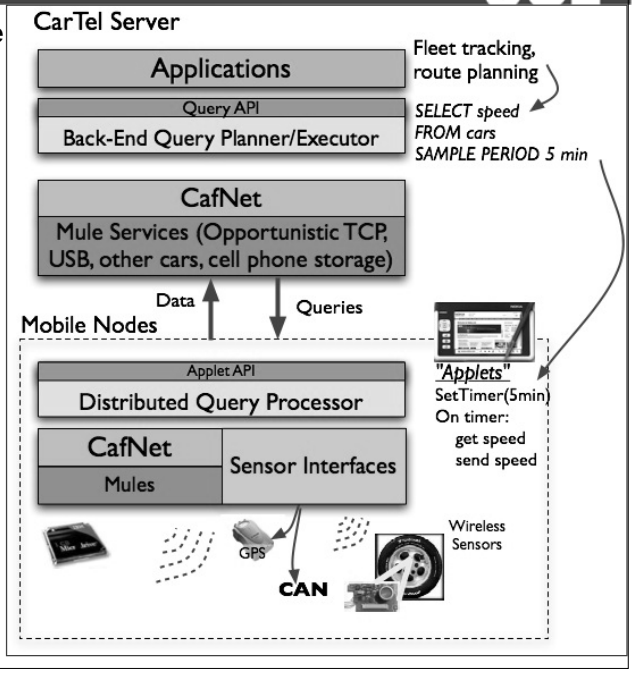
## The big picture



## Cartel Components

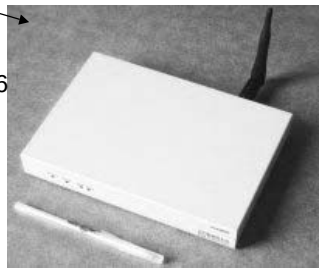
- ICEDB (intermittently connected database)
  - A delay-tolerant continuous query processor
- CafNet (carry-and-forward network)
  - a delay-tolerant network stack
- Portal
  - Sink of the data
  - Data visualization

## CarTel System Architecture

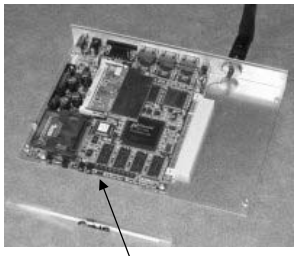


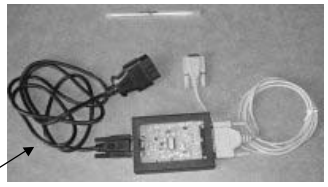
## Technology that makes CarTel possible

A 586-class processor running at 266 MHz with 128 MB of RAM and 1GBytes of flash memory



GPS





OBD-interface

miniPCI Wi-Fi Card

## Query-Based Distributed Data Management

```
SELECT cars.carid, cars.speed, cars.pos,
       history.events
FROM cars, history
WHERE cars.tire_pressure < 25 psi
AND cars.speed > 0
AND cars.carid = history.carid
SAMPLE PERIOD 1 min
```

User

Queries are
Easy to write
Easy to distribute
Easy to upgrade
Securable

Execution of query spread across many locations

Car n

Car 1

Results

Internet

CarTelHQ

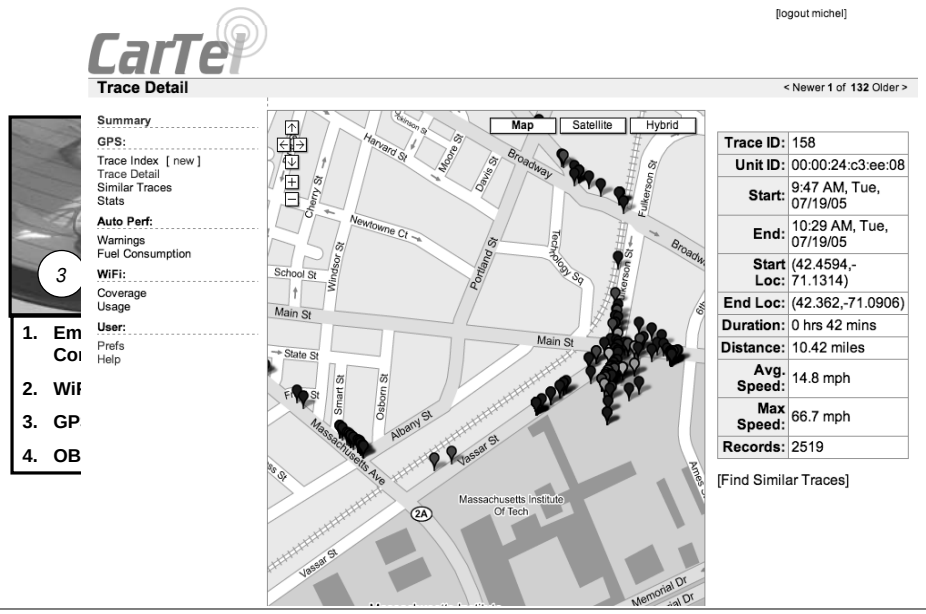
GUIs/Viz

Application Services "applets"

CarTel Server

Security Policy

## Current Status




## Querying

- Applications run on the portal
- They issue continuous queries on ICEDB (no synchronous wait)
- The queries specify
  - the sensors on which they need to run
  - priorities
  - Filtering
- Replies are streamed back to the intermittently connected network to the DB
- Snapshot queries run on data available on the DB

## Querying and Adapters

- Queries are delivered using CafNet to the remote nodes when nodes connect to portal
- Adapters allow the handling of details of specific sensors converting to normalized form
- Adapters contain:
  - Id
  - Type (push/pull)
  - Rate
  - Forwarding flag (if to forward raw data/ continous query)
  - Schema
  - Priority

## Continuous query model

- Continuous queries specify “rate”
- Batches of data delivered when car able to connect to portal
- Intermittently connected network: delivering the data in FIFO order is sub-optimal
- “Value” of data is application dependent (specified with priorities)
- ICEDB decides on the data order based on these
  - Assigns a score to replies (can change over time) locally
  - With global scoring the portal is involved in scoring (feedback)

## Local priorities

- Priority keyword is used: ICEDB transmits results in order of priority
- Delivery_order: allows dynamic, function dependent ordering of results to be shipped
- No feedback from portal needed

## Global Priorities

- Summarize_as: aggregates the results of the many query results in the buffer
  - The aggregation result is transmitted first upon connection with the portal
  - Aggregation based on user defined functions
  - Grouping of data can be performed
  - Useful to dynamically change the order of the upload with changing application needs

UCL

## CafNet

- Network stack for delay tolerant communication
- All nodes have an globally unique id
- Application Data Units (ADUs) have ids
- Network stack does no buffering
  - Applications buffer
- CafNet informs the application when connection available
  - Application can decide what data to send “at the last moment” instead of committing earlier to the data in advance by sending to the net layer

UCL

## CafNet

- 3 layers
  - Transport:
    - provides notification of network to application
    - Delivers incoming data
    - Registers data to transmit
    - Notifies of success
    - Request data
  - Network:
    - Routing
  - Mule Adaptation Layer
    - Uniform neighbour discovery (media independent interface)

## Buffering

- Buffering only happens in the application
- For optimization purposes a small amount of buffering happens in CafNet
  - This is to avoid delays due to data packing by the application when connection is available
  - Right tradeoffs of the size of this buffer need to be found

## Case Studies

- Road traffic analysis: GPS adapter and continuous queries to keep track of routes and delays
  - Commute time analysis
  - Traffic hot spot heuristics
- Wifi measurements
- Driving patterns analysis
- On board diagnostic data

UCL

## Related work

- Mobile sensor networks (zebranet, scar..)
- DTNs
- Query processing
- Road monitoring projects

UCL

## Research Challenges

- Programming model - How are simple apps built?
- Traffic-aware route planning
- Security and privacy
- Routing
  - GPS Errors
    - What road does a given GPS coord belong to?
  - Privacy-introduced errors