Sensornet 2.0: the new frontier

Feng Zhao
Microsoft Research
http://research.microsoft.com/zhao

IEEE RTSS Keynote, Rio de Janeiro, Brazil, November 6, 2006

Bell's Law of Computer Classes

- “Roughly every decade, technology advances in semiconductors, storage, *networks*, and *interfaces* enable a new, lower cost computer class aka platform to form to serve a new need that is enabled by smaller devices e.g. less transistors per chip, less expensive storage, displays, i/o, network, and unique interface to people or some other information processing sink or source.”

- In 2005 the computer classes include: mainframes (60's); minicomputers (70's); personal computers and workstations evolving into a network enabled by Local Area Networking or Ethernet (80's); web browser client-server structure that were enabled by the Internet (90's); web services e.g. Microsoft's .Net (2000's) or the Grid; cell phone sized devices c(2000); Wireless Sensor Networks aka motes (>c2005). Bell predicts home and body area networks will form by 2010.

Source: Wikipedia, http://en.wikipedia.org/wiki/Bell's_Law_of_Computer_Classes

Sensornet 0.1

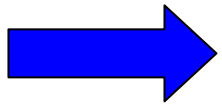
(pre-historic, circa 1980)

- DARPA DSN (PM: Bob Kahn)
  - Truck-sized sensors (aka radars)
  - Connected via Ethernet
  - With microwave radios

Mobile Node

Equipment Rack

Sensornet 1.0

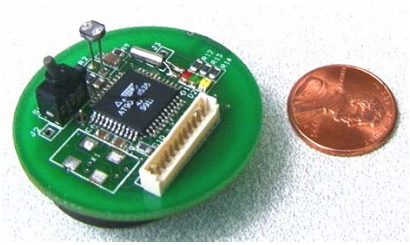


- Single purpose, one-time, dedicated instrumentation
- Homogeneous platforms (aka motes)
- Device centric (e.g. TinyOS)
- Closed architecture, centrally managed
- User interfaces
- Small, isolated
- Sensing
- It’s all about the system

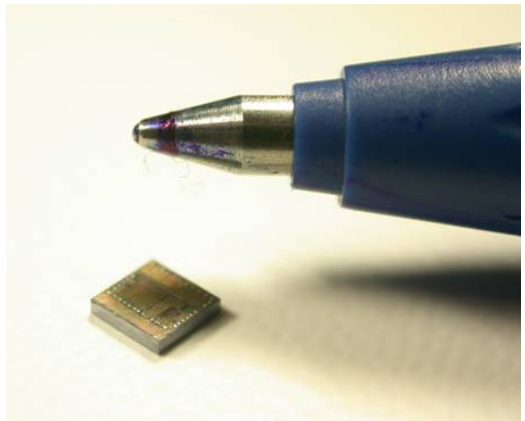
Sensornet 2.0

- Multi app/user, re-usable, shared infrastructure
- Heterogeneous, reconfig (motes, webcams, cell phones, watches)
- Network centric (e.g., web services, tasking)
- Open, extensible, inter-operable
- Mash-ups
- Network effects, participatory (e.g. blogging)
- Sensing and control
- Systems *and* data

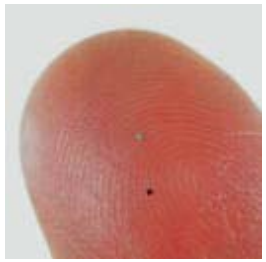
Evolution of sensornet platforms



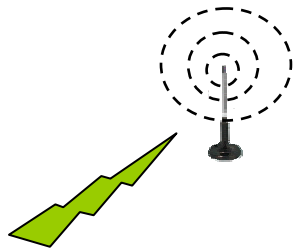
Berkeley WeC mote



Berkeley Spec mote



Hitachi
mu-chip RFID



SPOT watch

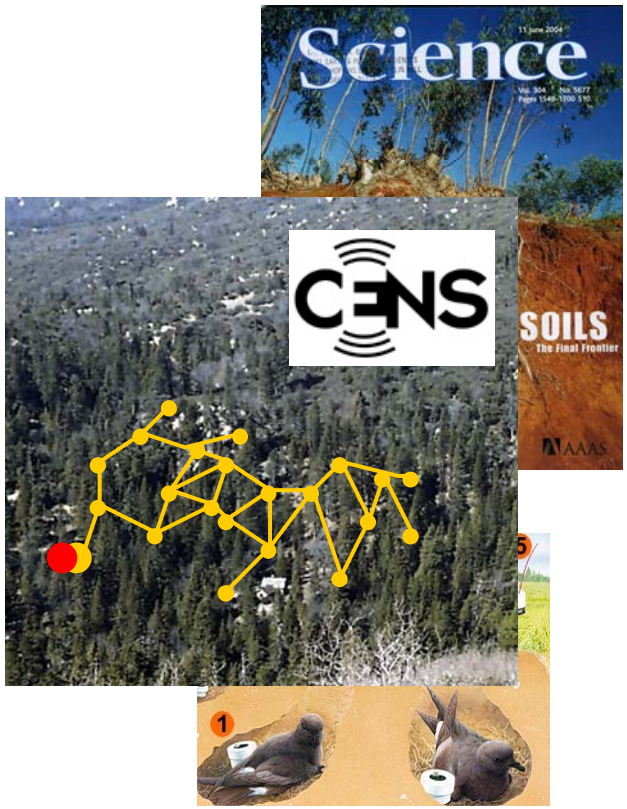
Sensoria WINS NG 2.0

iPAQ handheld

Cell phones

pedometer

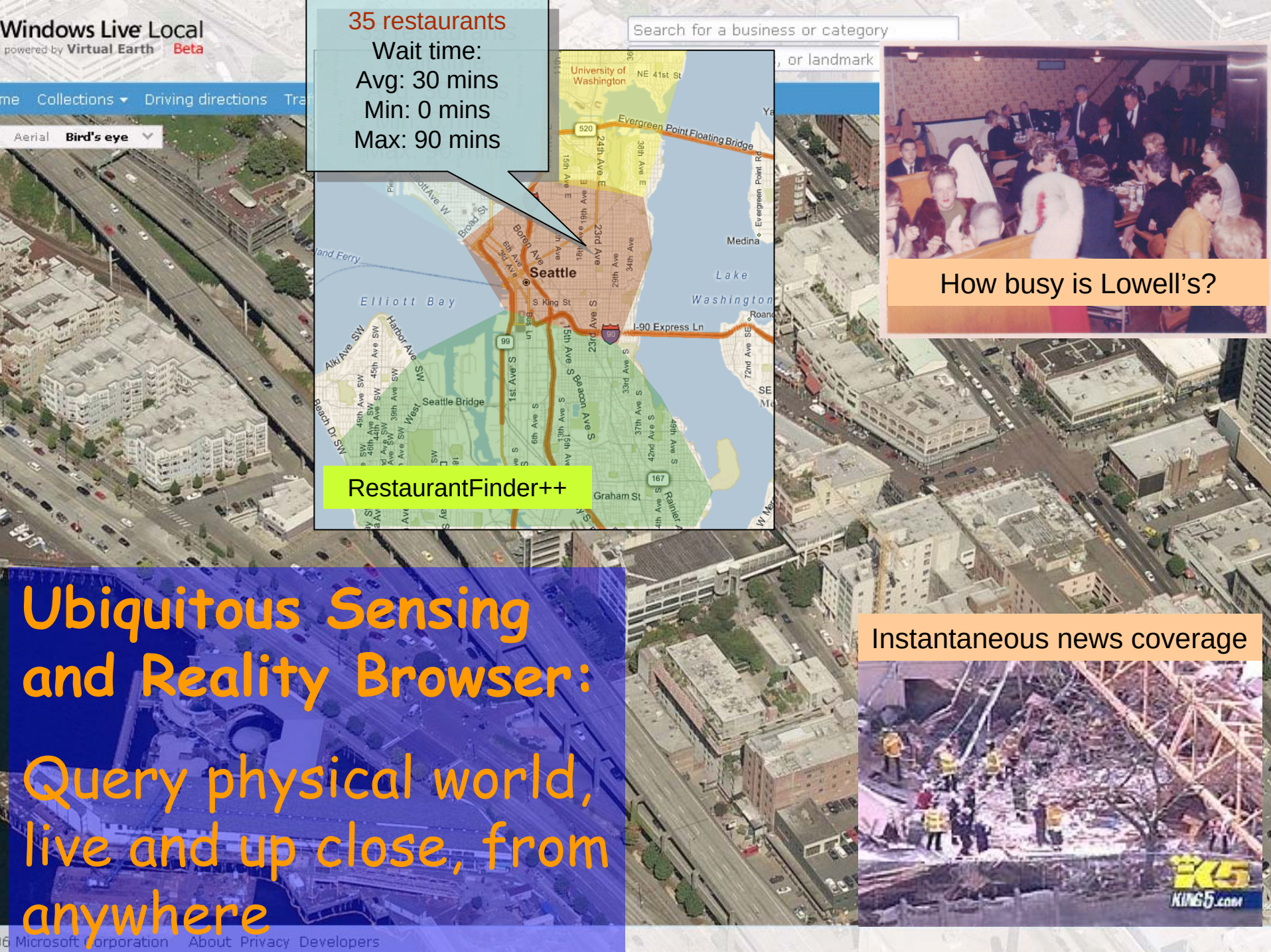
Evolution of sensornet applications



Environmental
- Monitoring space
- E.g., habitat, birds

Industrial:
- Monitoring objects
- E.g. machines, inventories

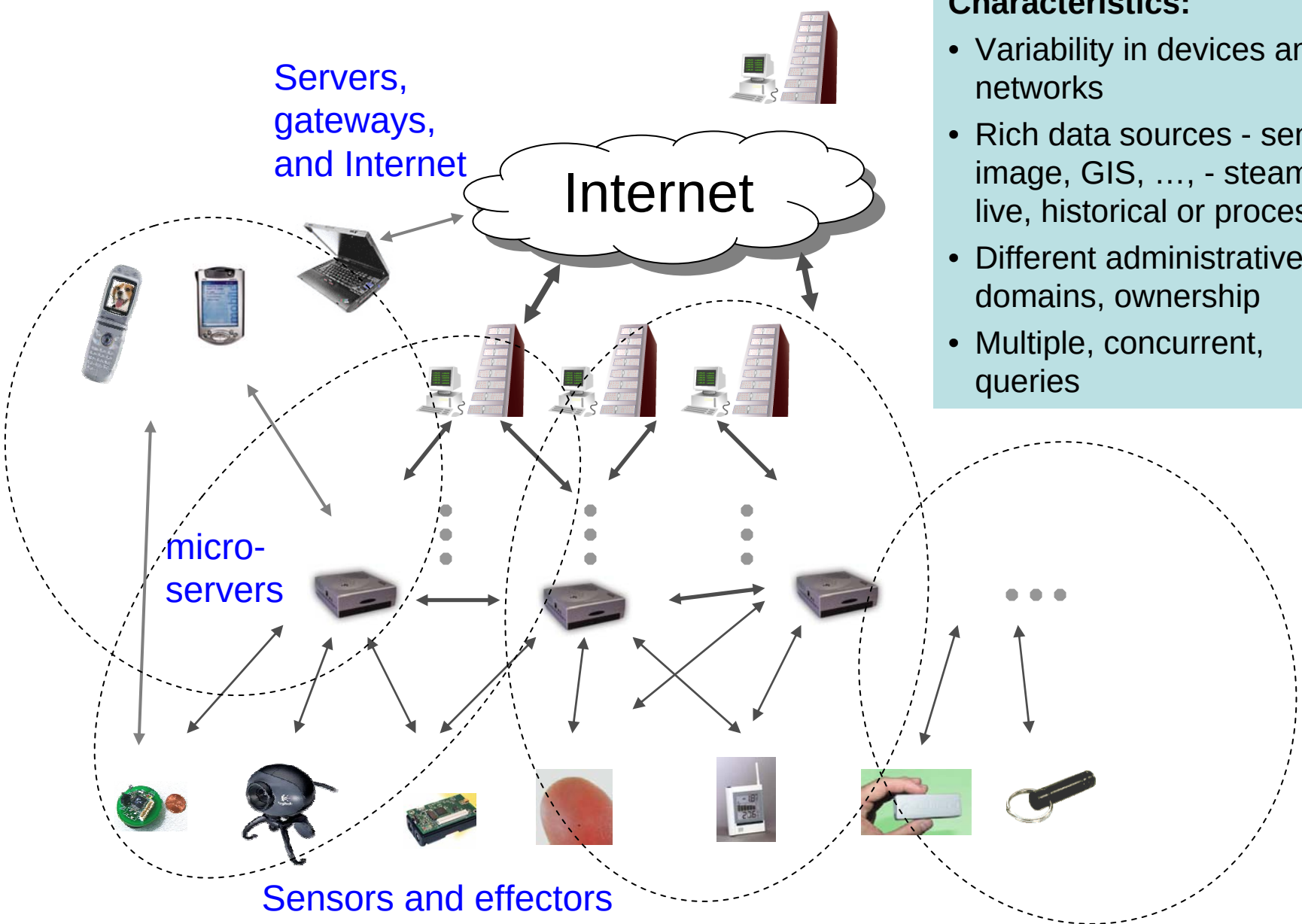
People and community:
- Monitoring activities
- E.g. heath, play, connect


Windows Live Local
powered by Virtual Earth Beta
Search for a business or category
or landmark
Collections
Driving directions
Aerial
Bird's eye
35 restaurants
Wait time:
Avg: 30 mins
Min: 0 mins
Max: 90 mins
University of Washington
NE 41st St
Evergreen Point Floating Bridge
520
Seattle
Medina
Lake Washington
Elliott Bay
S King St
I-90 Express Ln
Seattle Bridge
Graham St
RestaurantFinder++
How busy is Lowell's?
Instantaneous news coverage
KING5.com
Ubiquitous Sensing and Reality Browser:
Query physical world, live and up close, from anywhere
Microsoft Corporation
About
Privacy
Developers

Where is real time?

- Devices interact with env and user; must promptly process time-critical events
- Network system should adapt to load and env changes
- Closing the loop of sensing and actuation requires real-time tasking of devices

Sensornets and the Web



Characteristics:

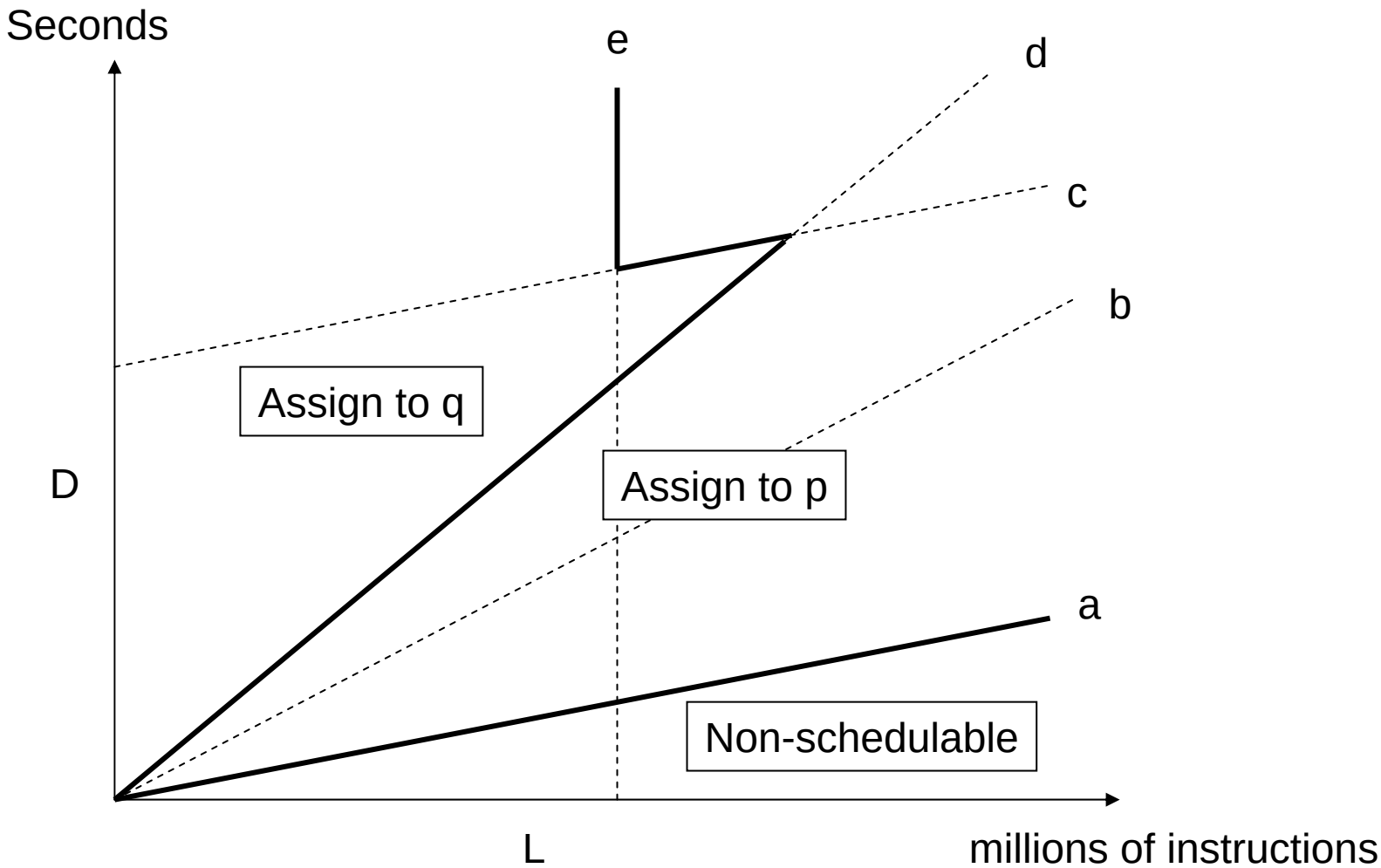
- Variability in devices and networks
- Rich data sources - sensor, image, GIS, …, - steaming, live, historical or processed
- Different administrative domains, ownership
- Multiple, concurrent, queries

Device platforms

Heterogeneous multi-processor energy efficiency

- Faster processor has higher MIPS and MIPJ, but consumes more power when active or during active/sleep transition
- Slower microcontroller has smaller current in active, sleep, or transition, but takes more time to do the same job
- Heterogeneous, multi-proc architecture allows more customizable aggregation of proc power
- When load changes, jobs can be migrated from slow to fast, or vice versa, to optimize for energy utilization
- The deadline, load, frequency of transition matter in scheduling (in general NP-hard)

Min-energy, deadline-aware scheduling: a two-processor case



mPlatform: A clean-slate approach to reconfigurable networked embedded computing

- What we want
  - Re-think how hw/sw for small devices are organized, using a flexible and yet constrained platform
  - Drive this with concrete apps

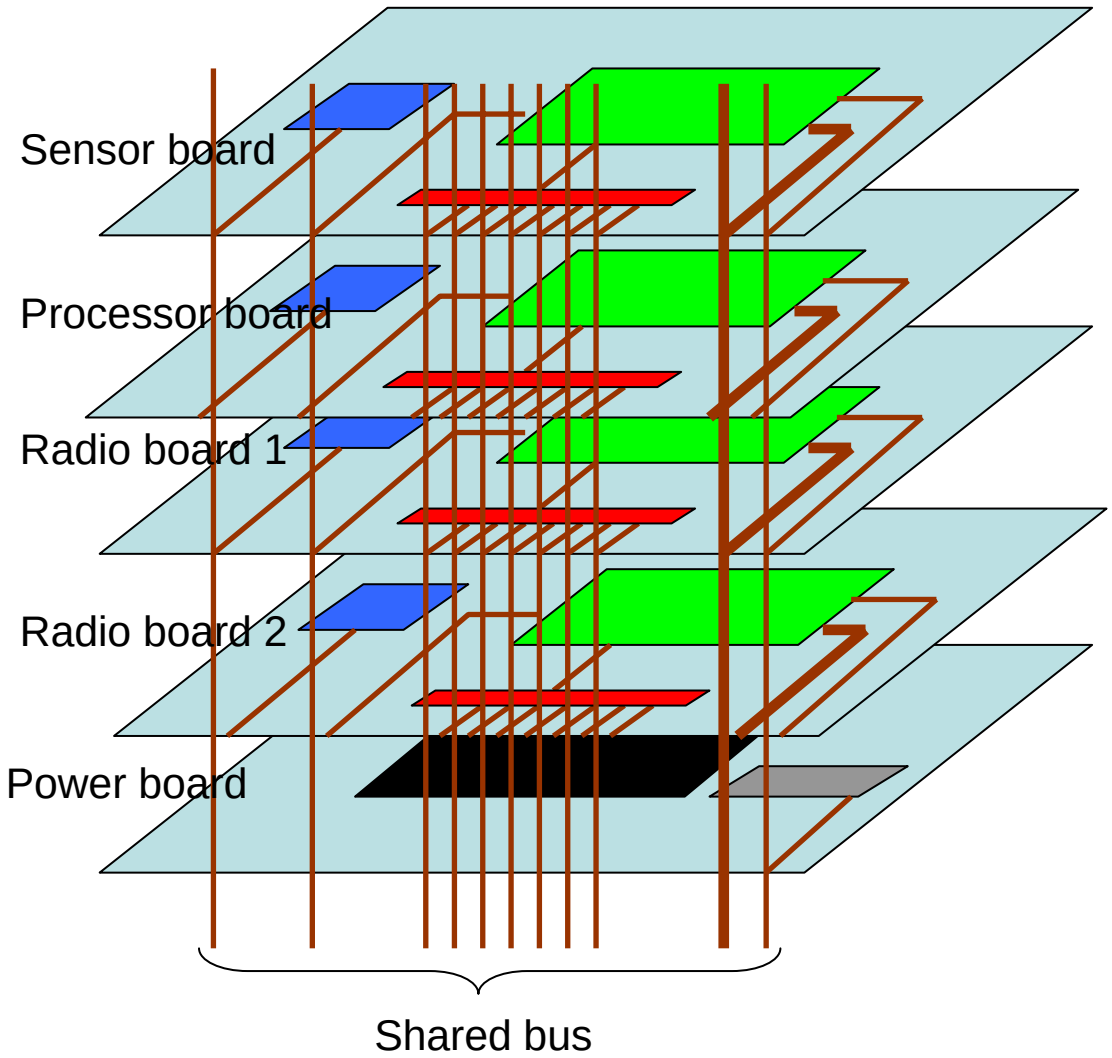
- Lego-like, extensible multi-board device
  - Multiple radios and processors, add-ons (sensors)
  - Real-time handling of time-critical events
  - Programmable inter-connects between boards

- Reconfigurable, modular software platform
  - Uniformity in on/off node messaging
    - Between boards (via bus) and between nodes (via radio)
    - Make app development processor-/transport-agnostic
  - Power-aware scheduling and dynamically loadable SW modules
  - Multi-board tasking and network data flow programming model

Issues to address

- Real time response
  - Handling of time-critical events
- Resource sharing and management
  - Resource (energy, bandwidth, processing, storage) models and monitoring
  - Task prioritization and scheduling
- Programming the networks as a whole
  - Programming model; event driven vs. threaded
  - Interfaces between various components
- System management
  - Tools for sys config/mgmt, for data collection and vis, for in-situ debugging

A reconfigurable, extensible multi-module design

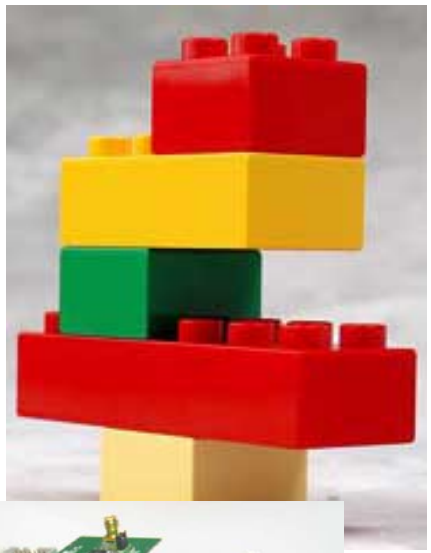


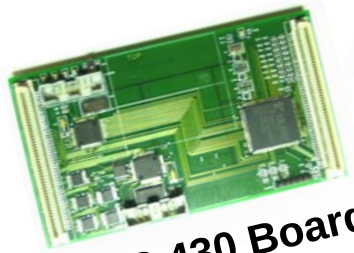
- Stackable heterogeneous modules
- Each board has processor/storage to handle time critical events
- A reconfigurable high-speed bus for cross-module data sharing
- A shared clock signal enables system-wide synchronization
- Advantages:
  - Can deign, build, and test independently
  - Flexible hardware configuration

Prototype boards for mPlatform

Lymberopoulos, Priyantha, Zhao, "mPlatform: A Reconfigurable Architecture and Efficient Data Sharing Mechanism for Modular Sensor Nodes"

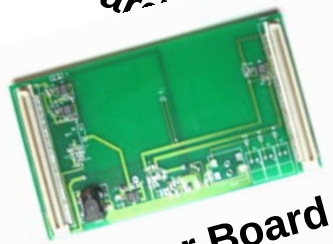
Lego-like, plug-and-play





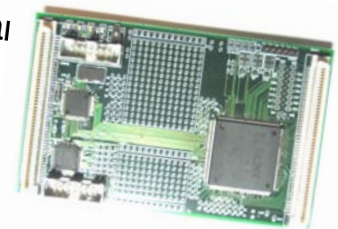
MSP 430 Board

ARM7 Board

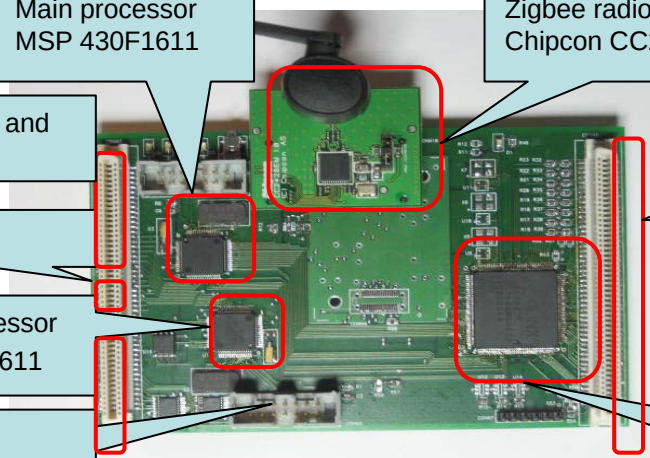


802.11 Radio Board

Power Board



Prototyping Board



Main processor
MSP 430F1611

Zigbee radio
Chipcon CC2420

Node config and
time synch

I2C bus

Config processor
MSP 430F1611

Switchable
multi-serial bus

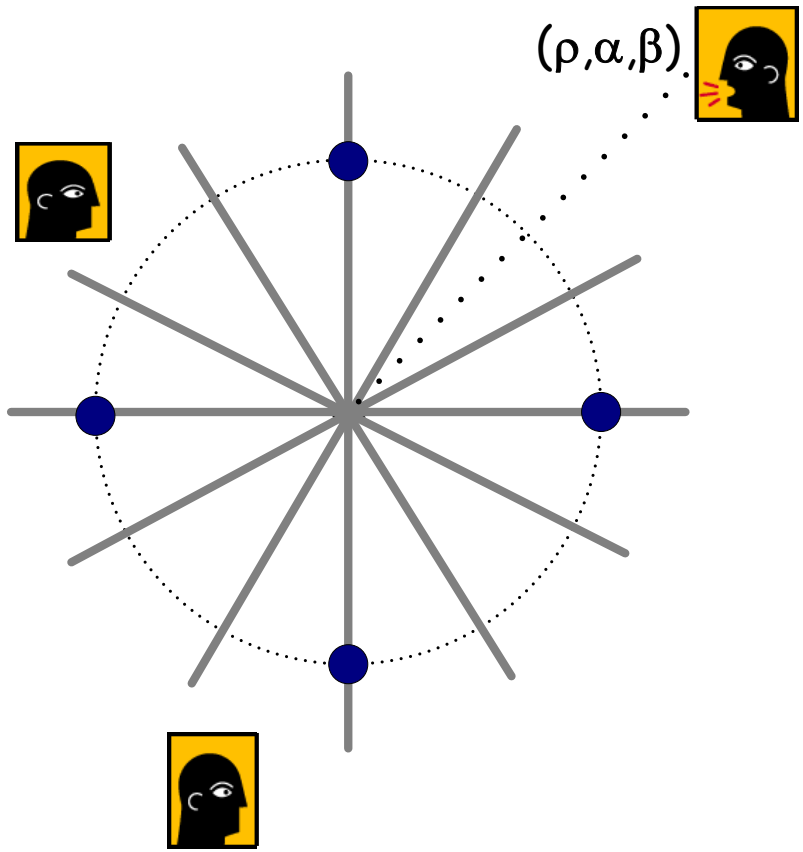
High-speed
parallel databus

CPLD - Xilinx
CoollRunner series

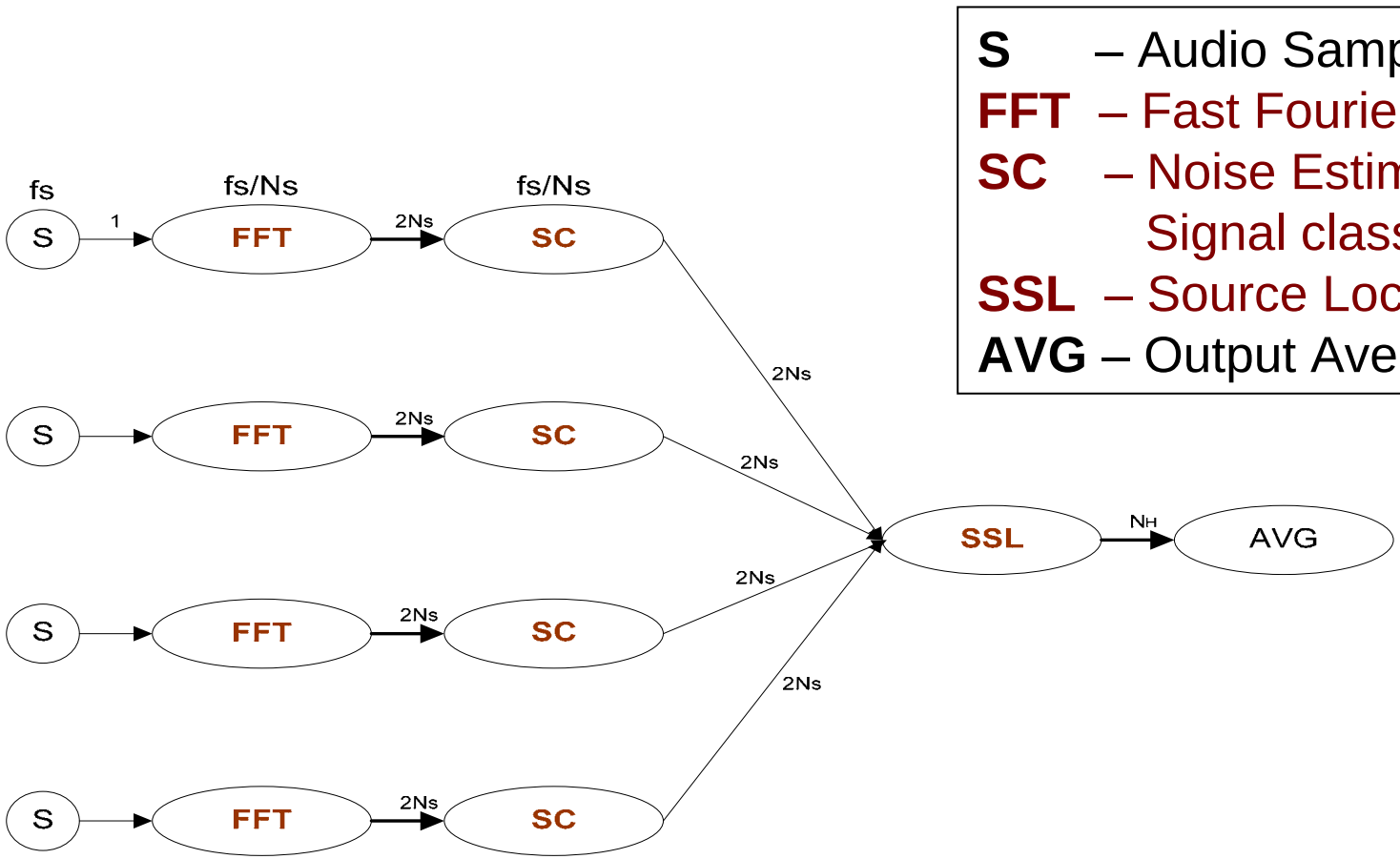
802.15.4 Radio Board

An Application: Sound source localization (SSL)

- Microphone array
  - Intelligent rooms
  - Beamforming, tracking
- Location (ρ,α,β)
  - Estimate time-difference of signal arrival via maximization of correlation
  - Hypothesis testing through discretization of the coordinate space



SSL Task Graph



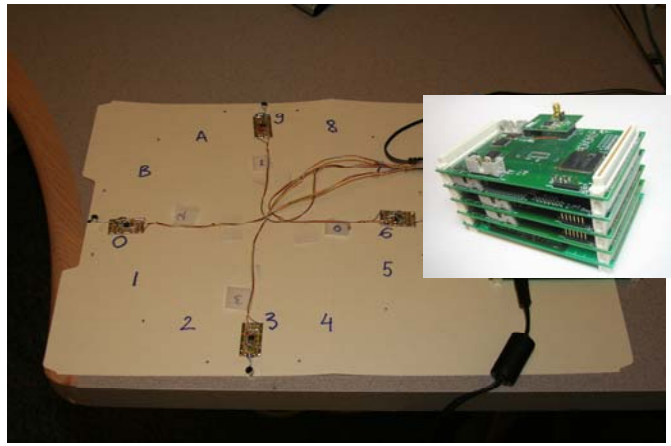
Ross Cutler, Yong Rui, Anoop Gupta, Jonathan J. Cadiz, Ivan Tashev, Li wei He, Alex Colburn, Zhengyou Zhang, Zicheng Liu, and Steve Silverberg. Distributed meetings: a meeting capture and broadcasting system. ACM Multimedia, 2002.

MSR RingCam

Two dual CPU 2.2GHz Pentium 4

Can we run SSL on distributed array of low-power procs?

mPlatform



4 TI **MSP430** each with mic for low power sensing. Spec: 6MHz, 10K RAM, 48K flash
1 **ARM7** TDMI for energy efficient comput. Spec: 60MHz 32K RAM, 512K ROM

- How many MSP/ARM boards do we need for real-time SSL?
- What is the energy optimal assignment of tasks to proc?
- Can the system respond to time-critical events such as fire if also used as an emergency alert system?
- Will the system still function despite failures of its parts?

Capabilities to Support

Desired Features:

- Real-time response to time-critical events
  - Meet deadlines, handle high-priority events
- Fine-grained power management
  - Conserve resources, maximize battery life
- Load balance and failure recovery
  - Move processing around to accommodate new tasks or recover from local failures
- Cross-board programming model
  - Simplify task naming, communication, and synchronization

Approaches:

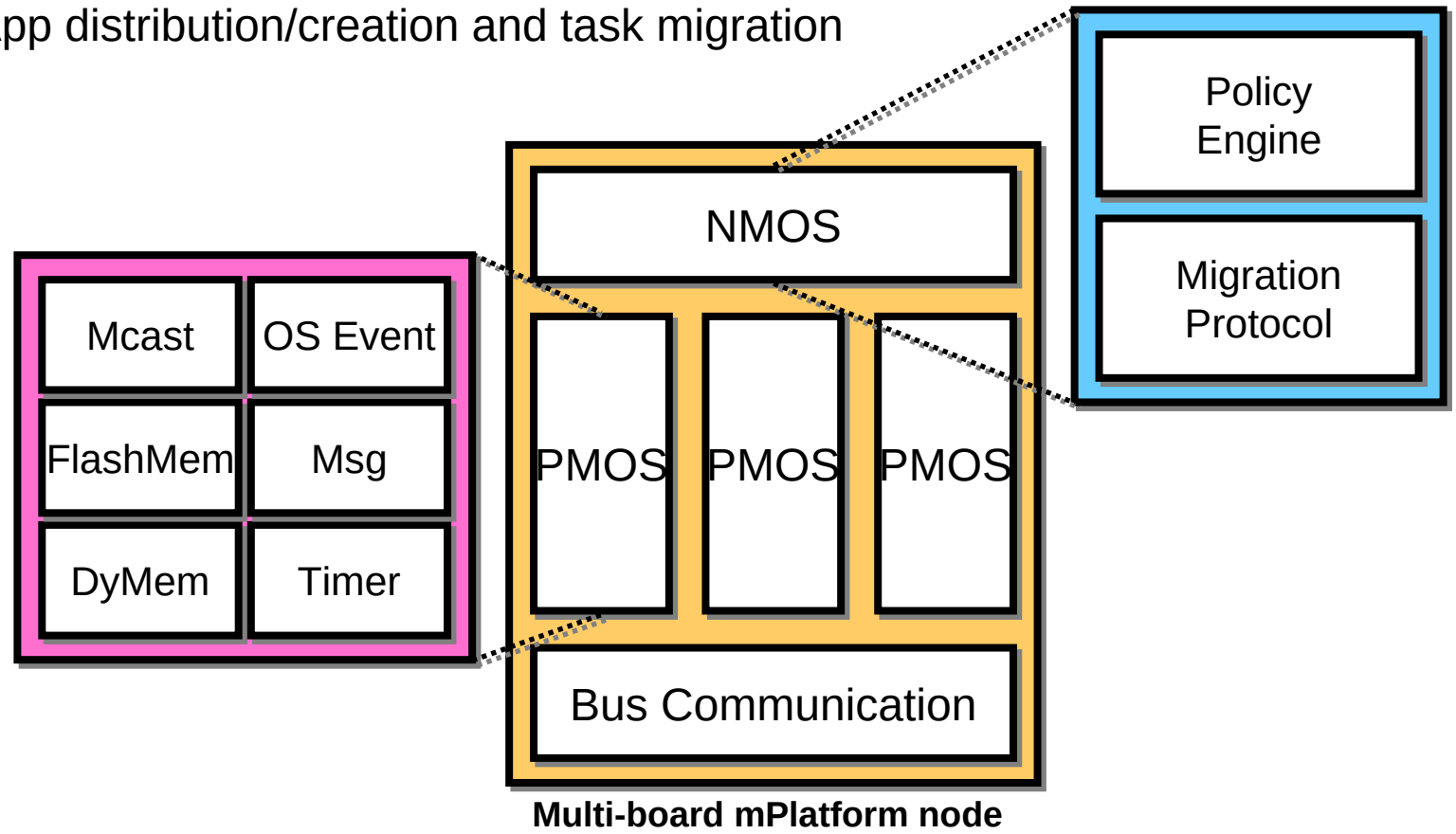
- Preemption with priorities
- Resource models and scheduling
- Dynamic task migration
- Message passing and data-flow abstraction

Design Choices

- Preemption with priorities
  - Asymmetric stack based preemption
  - Run-to-completion semantics
  - Suitable for simple processing tasks, with minimal memory overhead
- Dynamic task migration
  - Resource tracking, notification, and recovery
  - Serialization and deserialization of task state and timer
  - Migration policy
- Resource models/tasking
  - Energy models for compo such as proc, radio, bus
  - Power/frequency modes of proc/bus and task mapping to procs
  - Remote tasking over serial/wireless comm
- Component oriented message passing
  - Uniform messaging for local and remote communication
  - Task abstraction and late binding to resources

CoMOS: A light-weight, component messaging OS

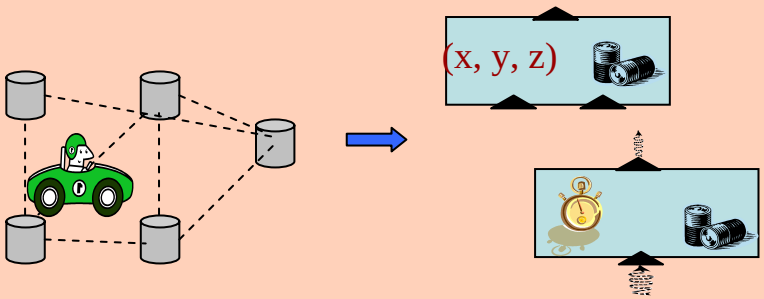
- PMOS (Priority Message OS)
  - Dynamic memory, timer, uni/multi-cast msgg, preemption, task priority, resource serialization
- NMOS (Network Message OS)
  - App distribution/creation and task migration



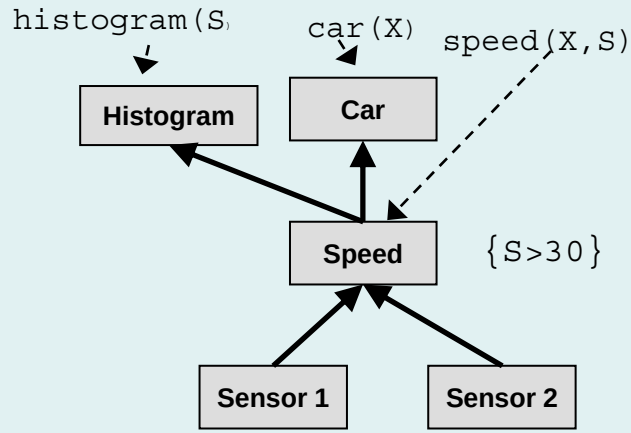
Han, Goraczko, Helander, Liu, Priyantha, Zhao, "CoMOS: An Operating System for Heterogeneous Multi-Processor Sensor Devices"

Task scheduling and execution

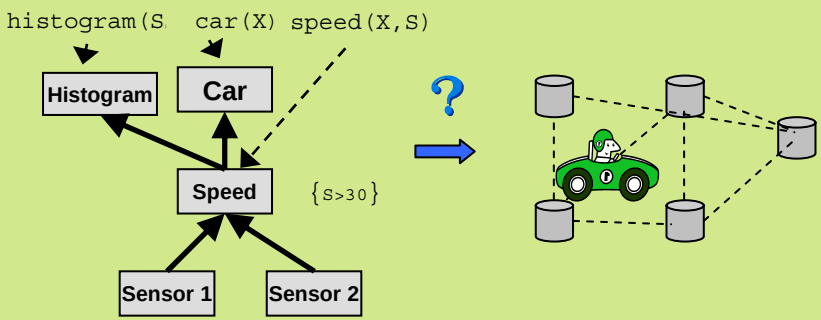
Task Abstraction and Interface



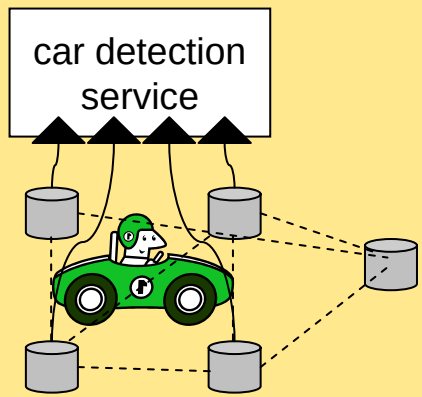
Task Planning



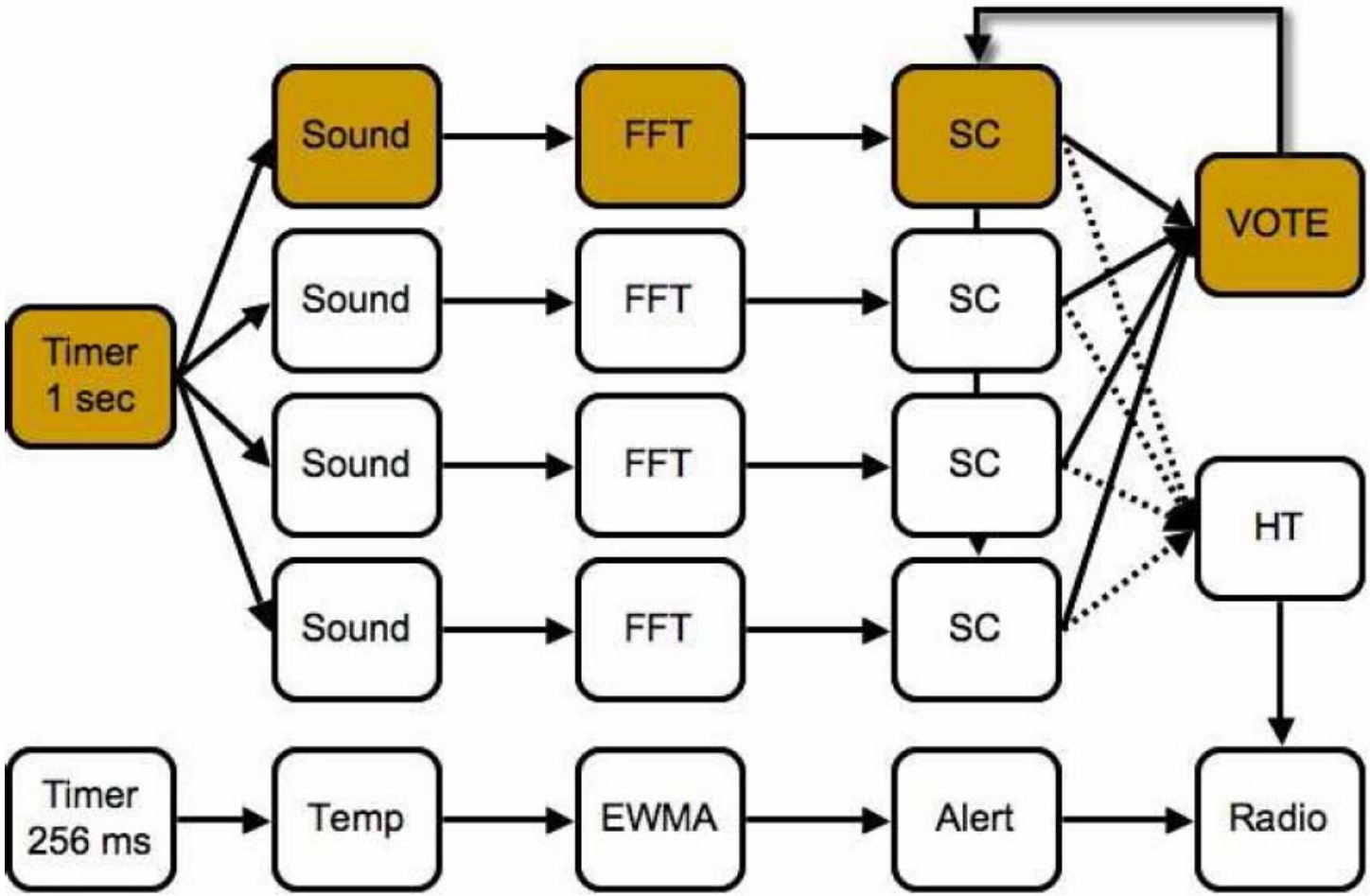
Task Mapping



Task Scheduling and Execution

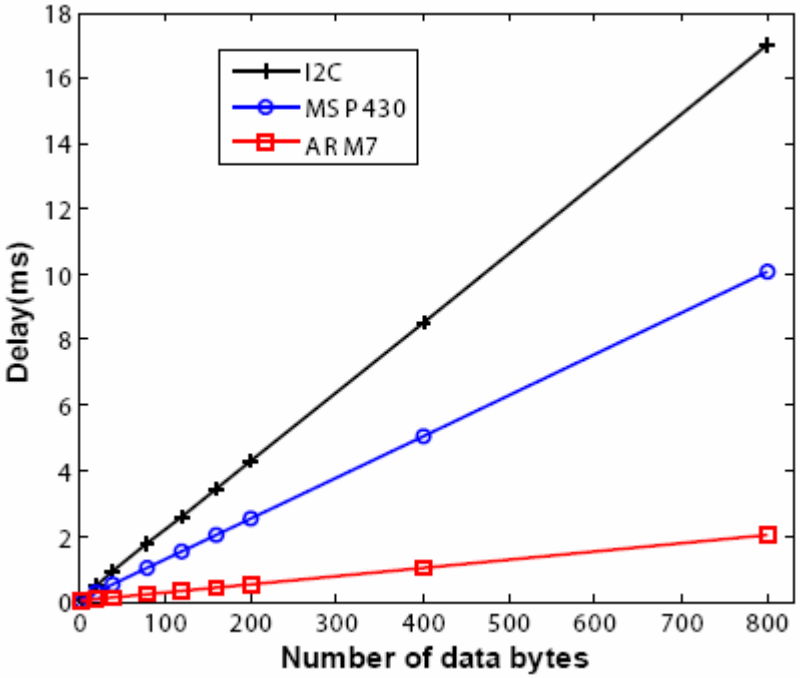


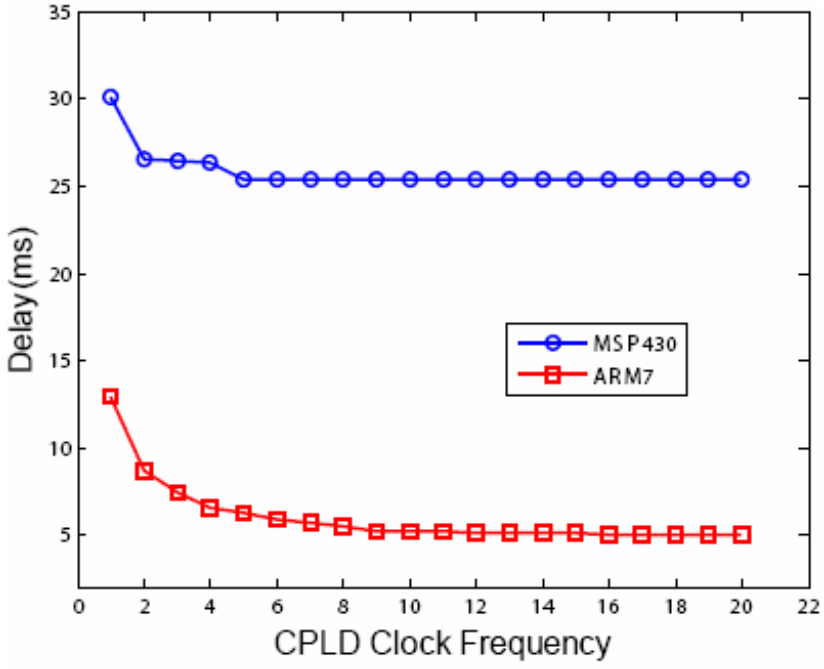
Application task graph



20 tasks, 21 uni-casts, 2 multi-casts

Scalable inter-processor communication





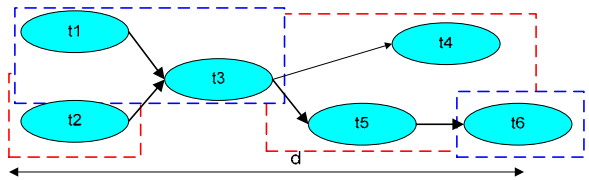
(End-to-end delay sending 2K bytes)

Performance of CPLD-based, TDMA bus

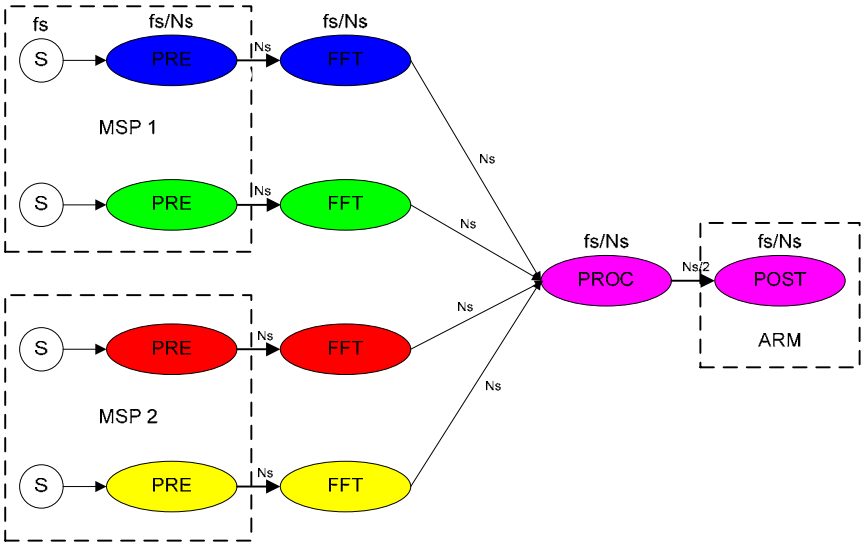
- Much faster than I2C, allowing processors to be decoupled from the bus
- Scalable: processor is the bottleneck, not the bus
- Performance gain at low power: at 5MHz, CPLD power is 5mW

Scheduling and Task Mapping

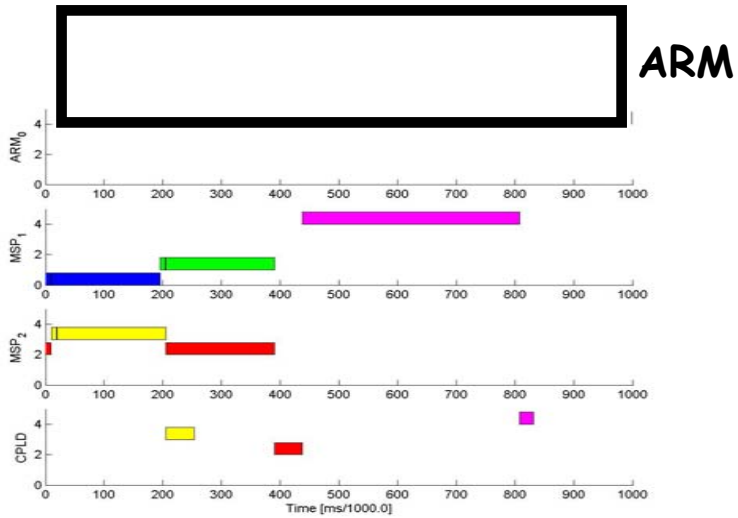
- Input:
  - CPU (board) and CPLD:
    - Power model (power for each mode),
    - Utilization bound
  - Task graph:
    - Number of cycles i.e. execution time for each task
      wcet(**t**,**p**,**m**) (for each CPU, mode)
    - Release and deadline times
    - Message data size i.e. communication time
      wcct(**t**,**m**) (for each CPLD mode)
  - Sensor:
    - Energy per reading
- Output:
  - CPU / CPLD to power mode mapping
  - Task to CPU mapping
  - List of task execution and communication start times



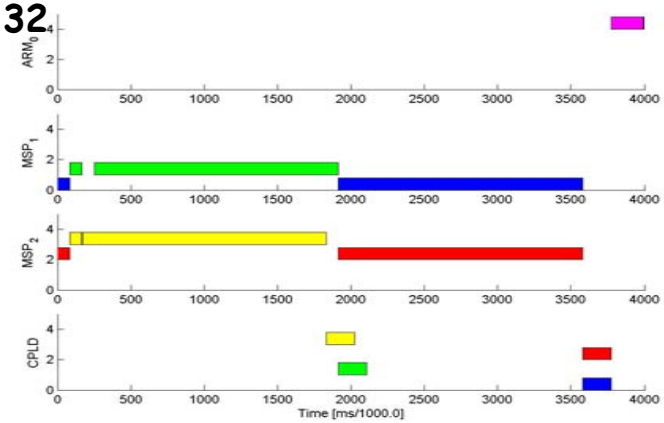
SSL mapping to mPlatform



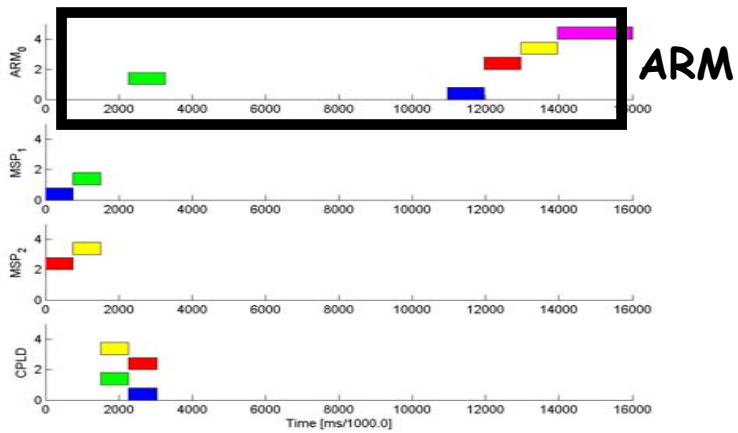
N=8



N=32



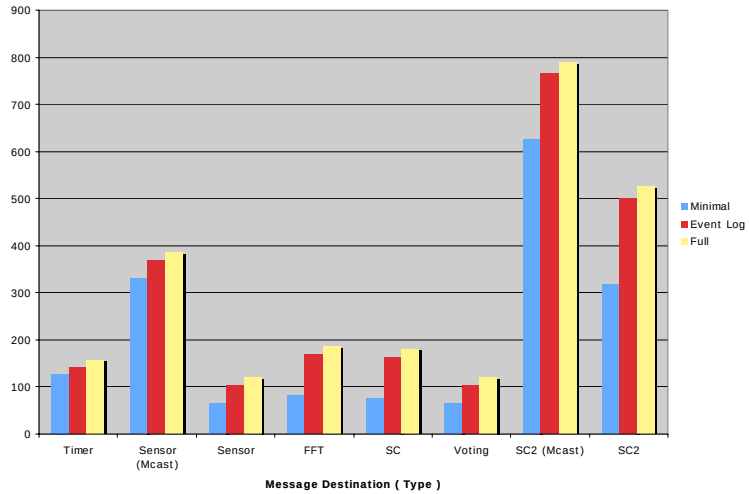
N=128



Matic, Goraczko, Liu, Lymberopoulos, Priyantha, Zhao, "Resource Modeling and Scheduling for Extensible Embedded Platforms"

Evaluation

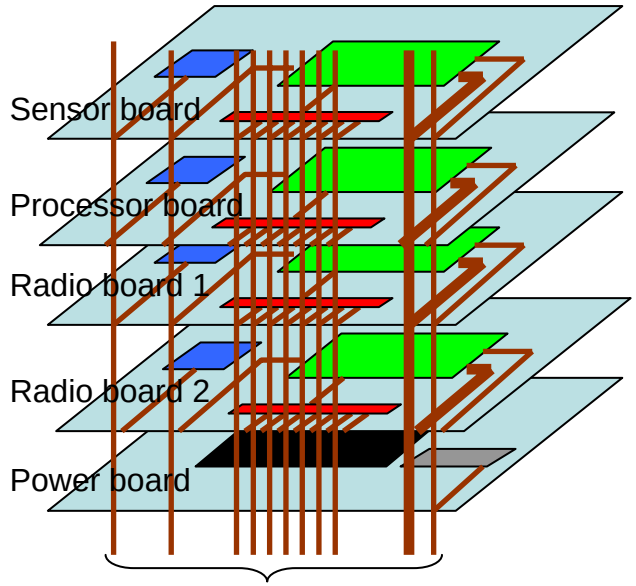
- Profiled SSL on MSP430 module
  - Compare CoMOS vs no-CoMOS (app hand optimized)
- Latency overhead
  - End-to-end SSL latency overhead, 253ms (CoMOS) vs. 250.6ms
  - Minimal CoMOS messaging overhead is 65.32 us
- Memory overhead
  - RAM: 164B; code:11.7KB



	RAM	ROM
Sched	135	5166
Timer	10	1732
Mcast	4	862
Msg Q	5	1162
Bus	6	1022
Mem	2	1278
Ports	2	506
Total	164 (81)	11728

mPlatform summary

- Designed and fabricated processing (ARM7/MSP430), radio (802.15.4/802.11), and power boards
- Developed a scalable, low-power CPLD based bus for cross-board communication, and drivers for the 802.15.4/Zigbee radio
- Developed a light-weight OS and ported to MSP430 and ARM7 boards, a power-aware scheduler with deadlines, and an overall tasking architecture
- Supported the development of SSL app on CoMOS

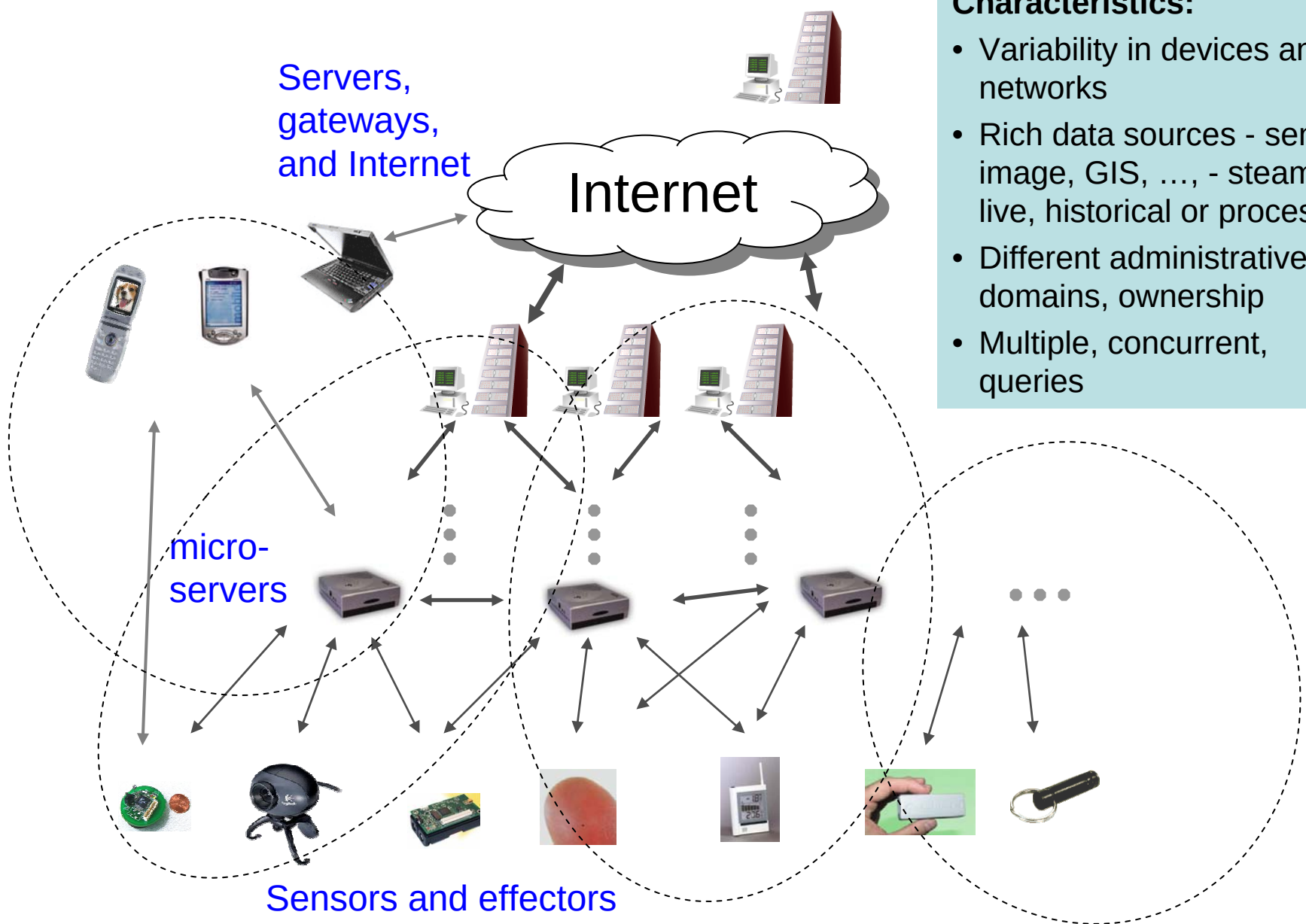


Shared bus

Prototype boards for mPlatform

World-Wide Sensor Web

Sensornets and the Web

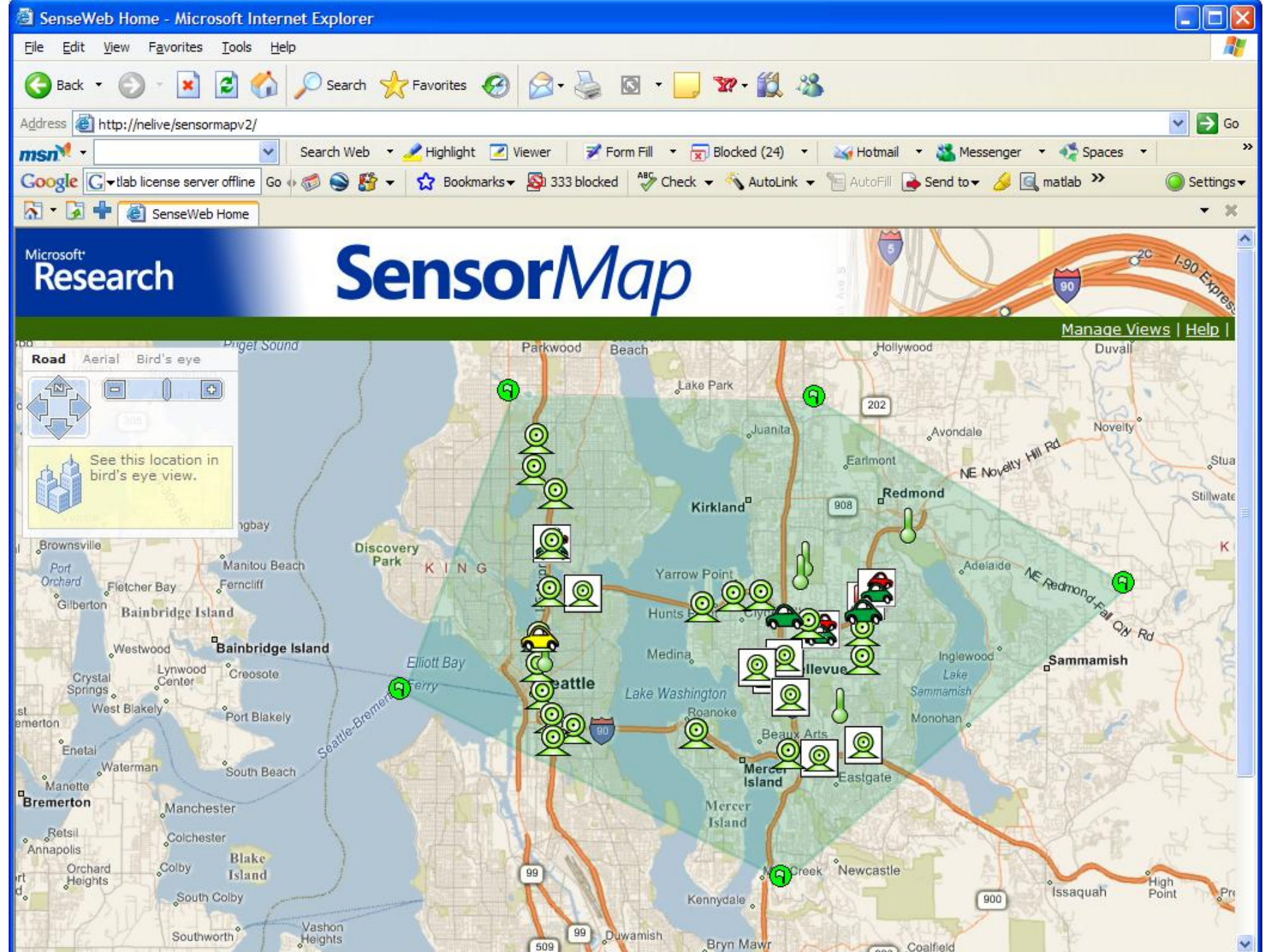


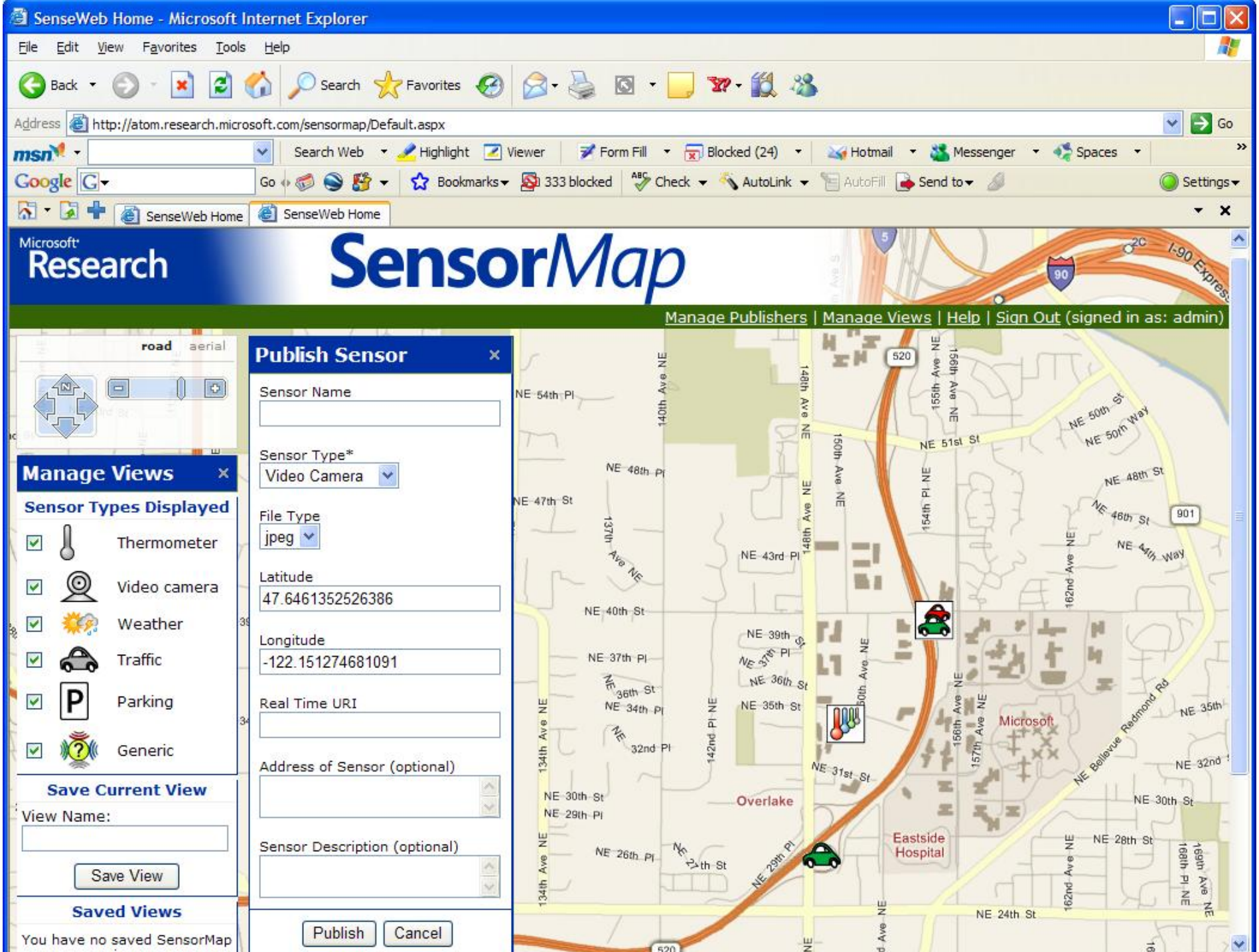
Characteristics:

- Variability in devices and networks
- Rich data sources - sensor, image, GIS, …, - steaming, live, historical or processed
- Different administrative domains, ownership
- Multiple, concurrent, queries

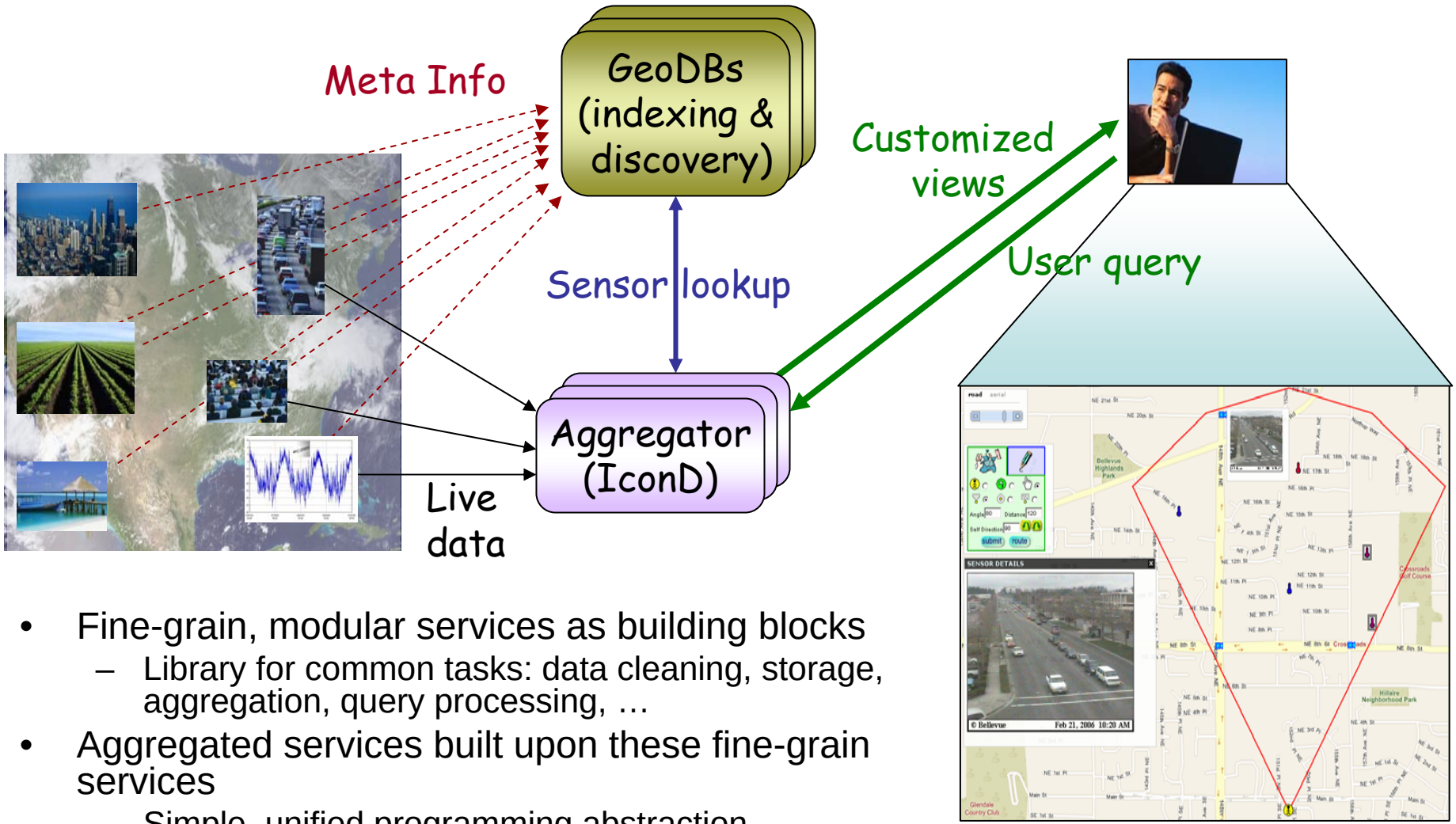
SensorMap as an example

http://atom.research.microsoft.com/sensormap


SenseWeb Home - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://nelive/sensormapv2/
Go
Search Web Highlight Viewer Form Fill Blocked (24) Hotmail Messenger Spaces
Google tlab license server offline Go Bookmarks 333 blocked Check AutoLink AutoFill Send to matlab Settings
SenseWeb Home
Microsoft Research
SensorMap
Manage Views | Help |
Road Aerial Bird's eye
See this location in bird's eye view.
Puget Sound
Parkwood
Beach
Hollywood
Duvall
Lake Park
Juanita
Avondale
Novelty
Earlmont
NE Novelty Hill Rd
Redmond
Kirkland
908
202
90
5
Discovery Park
K I N G
Yarrow Point
Adelaide
NE Redmond-Fall City Rd
Brownsville
Port Orchard
Fletcher Bay
Ferncliff
Manitou Beach
Gilberton
Bainbridge Island
Westwood
Bainbridge Island
Elliott Bay
Ferry
Seattle-Bremerton
Crystal Springs
Lynwood Center
Creosote
West Blakely
Port Blakely
Medina
Lake Washington
Roanoke
Beaux Arts
Mercer Island
Eastgate
Inglewood
Lake Sammamish
Monohan
Sammamish
Enetai
Waterman
South Beach
Manette
Bremerton
Manchester
Colchester
Retsil
Annapolis
Orchard Heights
Colby
Blake Island
South Colby
Southworth
Vashon Heights
Kennydale
Newcastle
Issaquah
High Point
99
900
509
Duwamish
Bryn Mawr
Coalfield


SenseWeb Home - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://atom.research.microsoft.com/sensormap/Default.aspx
Go
Search Web Highlight Viewer Form Fill Blocked (24) Hotmail Messenger Spaces
Google Go Bookmarks 333 blocked Check AutoLink AutoFill Send to Settings
SenseWeb Home
SenseWeb Home
Microsoft Research
SensorMap
Manage Publishers | Manage Views | Help | Sign Out (signed in as: admin)
road aerial
Manage Views
Sensor Types Displayed
Thermometer
Video camera
Weather
Traffic
Parking
Generic
Save Current View
View Name:
Save View
Saved Views
You have no saved SensorMap
Publish Sensor
Sensor Name
Sensor Type*
Video Camera
File Type
jpeg
Latitude
47.6461352526386
Longitude
-122.151274681091
Real Time URI
Address of Sensor (optional)
Sensor Description (optional)
Publish
Cancel
NE 54th Pl
140th Ave NE
148th Ave NE
155th Ave NE
156th Ave NE
520
NE 50th St
NE 50th Way
NE 51st St
150th Ave NE
154th Pl NE
NE 48th Pl
NE 47th St
137th Ave NE
NE 48th St
NE 46th St
901
NE 44th Way
NE 43rd Pl
162nd Ave NE
NE 40th St
NE 39th St
NE 37th Pl
NE 36th St
NE 35th St
NE 34th Pl
NE 32nd Pl
142nd Pl NE
150th Ave NE
134th Ave NE
NE 31st St
Microsoft
157th Ave NE
NE Bellevue Redmond Rd
NE 35th
NE 32nd
NE 30th St
NE 29th Pl
Overlake
NE 26th Pl
NE 27th St
NE 29th Pl
Eastside Hospital
NE 28th St
168th Pl NE
169th Ave NE
NE 24th St
162nd Ave NE
520
I-90 Express
90
5
2C

SensorMap Architecture



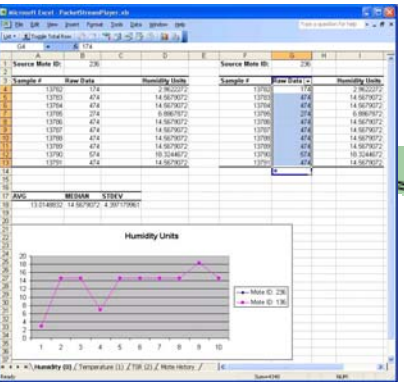
- Fine-grain, modular services as building blocks
  - Library for common tasks: data cleaning, storage, aggregation, query processing, …
- Aggregated services built upon these fine-grain services
  - Simple, unified programming abstraction
- Extensible, friendly to 3rd party services

Issues to address

- Publishing sensor data to the Internet
  - Standards, API, security, privacy
  - Data integrity, validation
  - Incentives
- Deliver live information to end users
  - Overlay infrastructure, resource sharing
  - Relevance metrics, economic models
  - Data fusion and sensemaking
- Architectural considerations
  - Service composition, tasking architecture
- Closing the loop of sensing and actuation
  - On-demand sensor control

MSRSense: A toolkit for data publishing, archiving, visualization

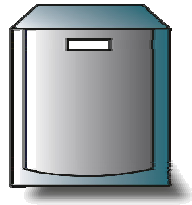
Visualize Events/ Process Data



User Interface / Data Processing
(Excel viz)

SQL Query / Report

Archiving Events

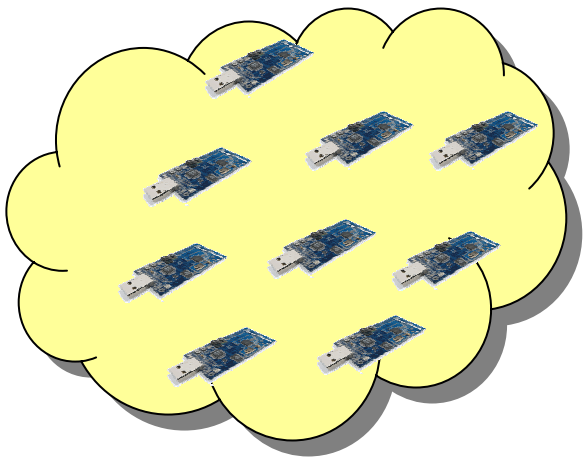


DataBase
(SQL Server 2005 or Express)

Transformed XML

Raw data
(XML packets)

Microserver Tasking



Sensor Net (Tmote Sky)

Status / Sensor Readings
(TinyOS Packets)

Publishing Data

Gateway (MicroServer data publishing web service)

Working with the research community

Microsoft® Research

Networked Embedded Computing & External Research and Programs

Sensor Networks Workshop 2005

Woodinville, Washington, USA
Willows Lodge

http://research.microsoft.com/workshops/sensorws2005

To probe further

- The **mPlatform** hardware platform, OS, scheduling work is published as MSR Tech Reports (available online)
  - http://research.microsoft.com
- **SensorMap** prototype demoed 5/06, with hundreds of blogs, and over 130,000 hits on Google search within two weeks
  - http://atom.research.microsoft.com/sensormap
- MSRSense **Toolkit** released 12/14/05, with 7000+ downloads since. Source code available for download at:
  - http://research.microsoft.com/nec/msrsense
- **Microsoft Robotics Studio** announced 7/06. An application development platform for the robotics community, designed for a wide variety of users, hardware, and scenarios. Downloads at:
  - www.msdn.microsoft.com/robotics
- MSR/UW **Summer Institute** on World Wide Sensor Web held in 8/06, with 50 people attending. Keynote by Astrophysicist Alex Szalay of JHU. All the presentation materials are online at:
  - http://www.cs.washington.edu/mssi/2006/schedule.html

Acknowledgment

- Joint work with Simon Han, Dimitrios Lymberopoulos, Slobodan Matic, Andre Santanche, Siddharth Seth, Michel Goraczko, Jie Liu, Suman Nath, Bodhi Priyantha, Tim Olson, Alec Woo

- Collaborations with Johannes Helander, Tom Blank, Mike Sinclair, Ray Bittner

- Engineering help from Jessica Miller, Darren Gehring

- Partnership with Stewart Tansley of MSR ERP

Sensornet 2.0: the new frontier

- Heterogeneous, reconfigurable platforms
  - Many forms, beyond motes
  - Better match with app needs
  - More efficient resource utilization
- Multi app/user, shared infrastructure
  - Open, extensible, inter-operable
  - Network and service centric
  - Closing the loop of sensing and control
- Community based sensing
  - Network effects, participatory (e.g. blogging)
  - Web-based mash-ups
  - Economic models, incentives and privacy