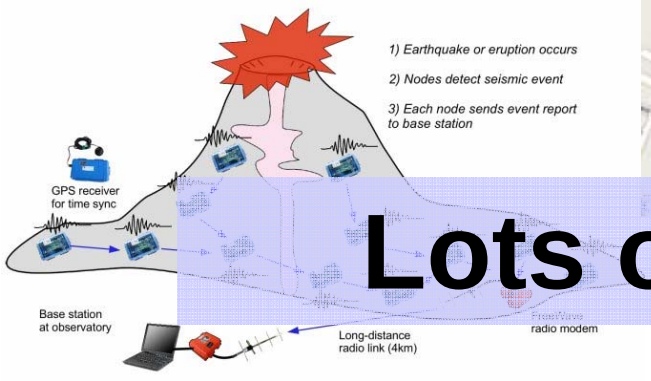
The Quest for a General-Purpose Sensing System

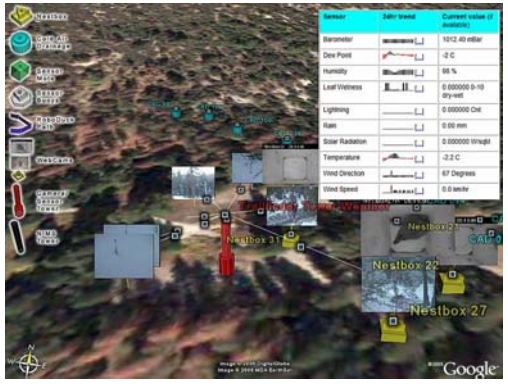
Ramesh Govindan
ramesh@usc.edu

University of Southern California

Wireless Sensing: Applications



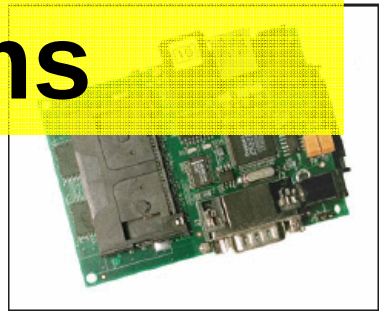
Lots of applications



Wireless Sensing: Platforms

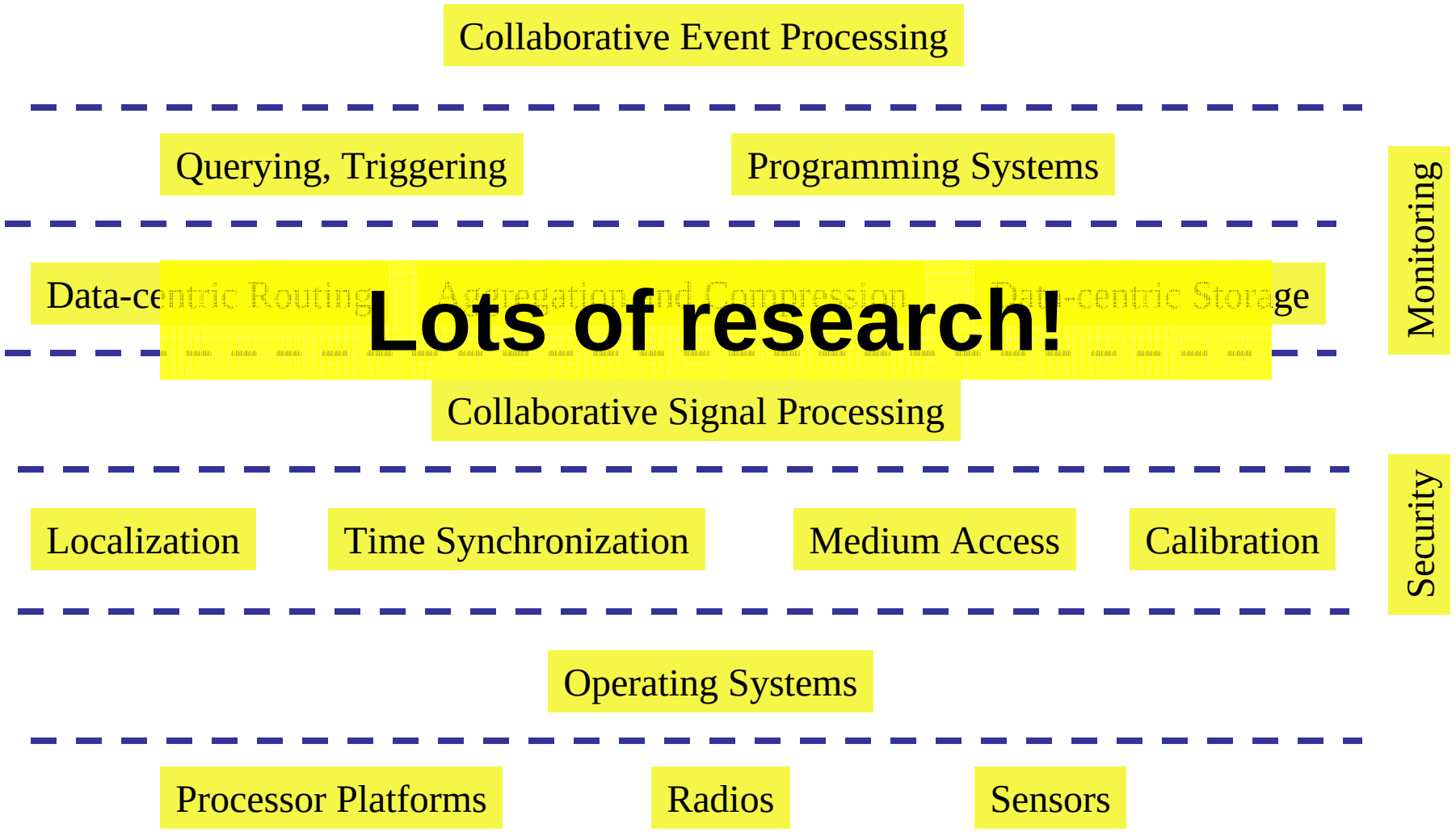
Motes: 8 or 16 bit sensor devices

Lots of platforms

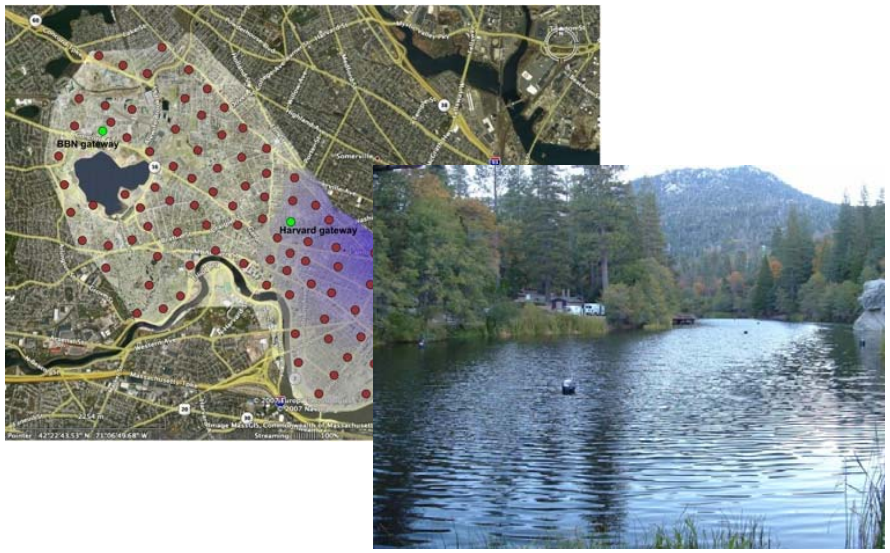


32-bit embedded single-board computers

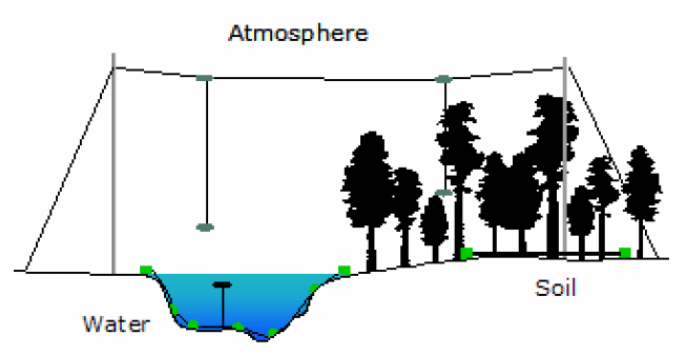
Wireless Sensing: Research



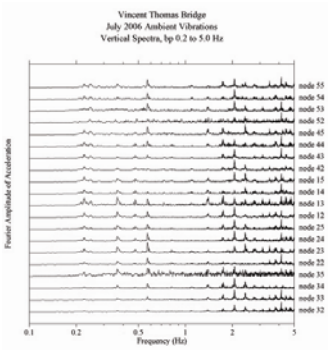
Wireless Sensing: The Future

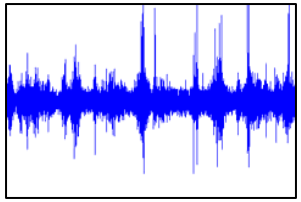


Larger spatial coverage



Reactive mobile elements





Complex sensing modalities

What's Missing?

Reuse as much code as possible

Leverage investment in large network to do many things

A General-Purpose, Multi-User, Reactive Sensing System

GPSS

System does more than just sense!

Outline

What is a GPSS?

Do we need GPSSs?

Hand-wavy

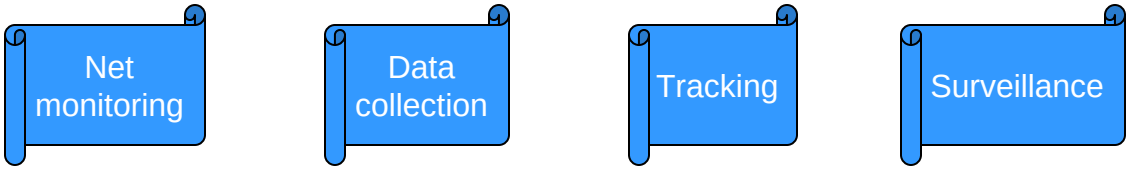
How have GPSSs evolved?

Personal

GPSS Definition

Applications

One-per-network, or concurrent



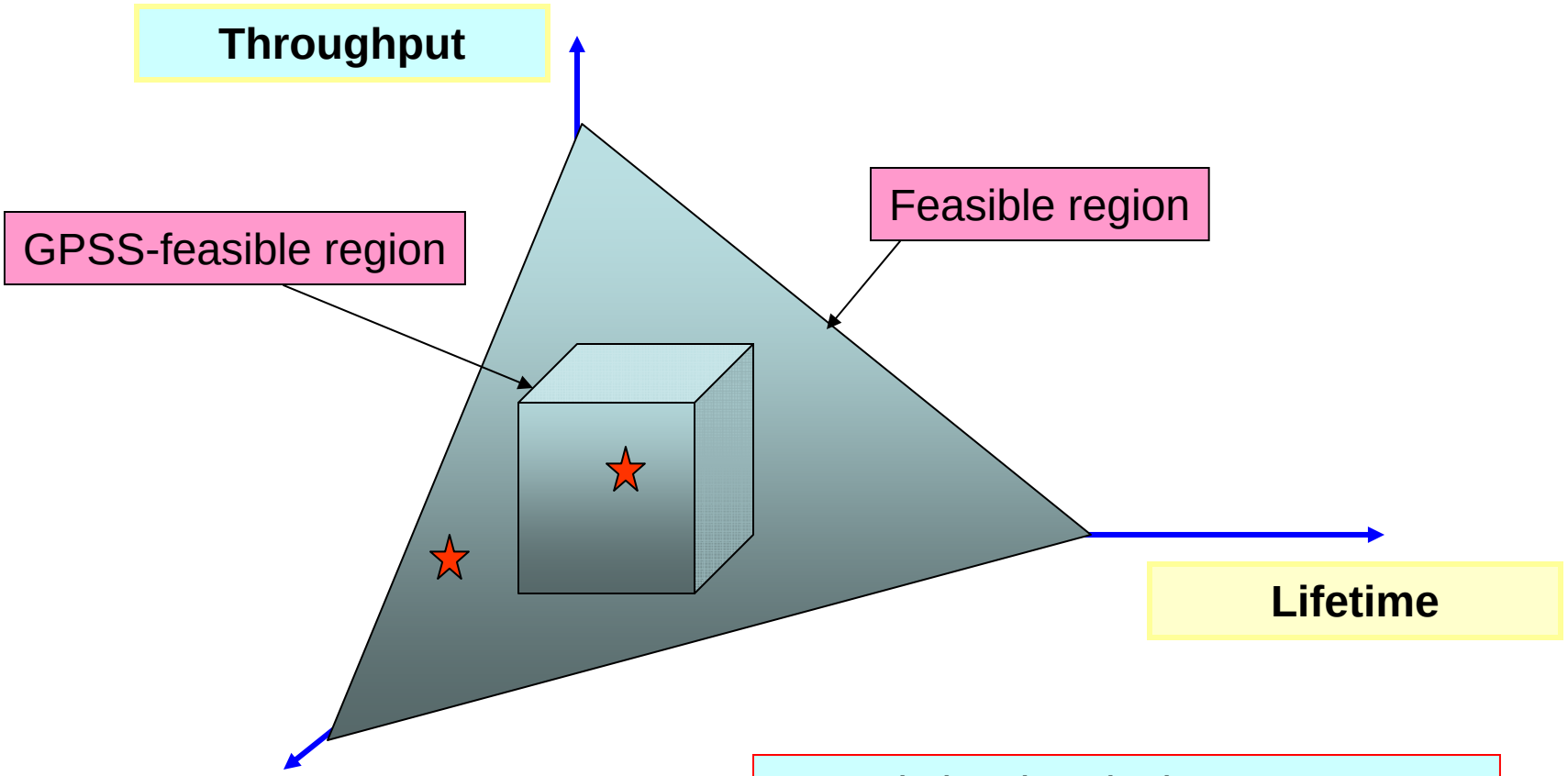
Definition is a bit fuzzy and subjective

A well-defined, compact abstraction for sensing and actuation

Isolation, security, energy-efficiency

An identifiable, and not easily changeable, core that implements the abstraction

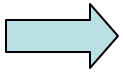
GPSS Limits



It is hard to design a GPSS
that covers the entire feasible region

Outline

What is a GPSS?



Do we need GPSSs?

How have GPSSs evolved?

If you don’t believe we do…

An Appeal to your Wallet

MONDAY JUL 10, 2006

The Rise of the General Purpose System

For the most part, customers love special purpose systems. There's only one part they don't love. Living with the economics.

By design, general purpose systems …
focus on horizontal segments of the market …
and allow us to amortize R&D investments over a far broader opportunity

An Appeal to Your Pride

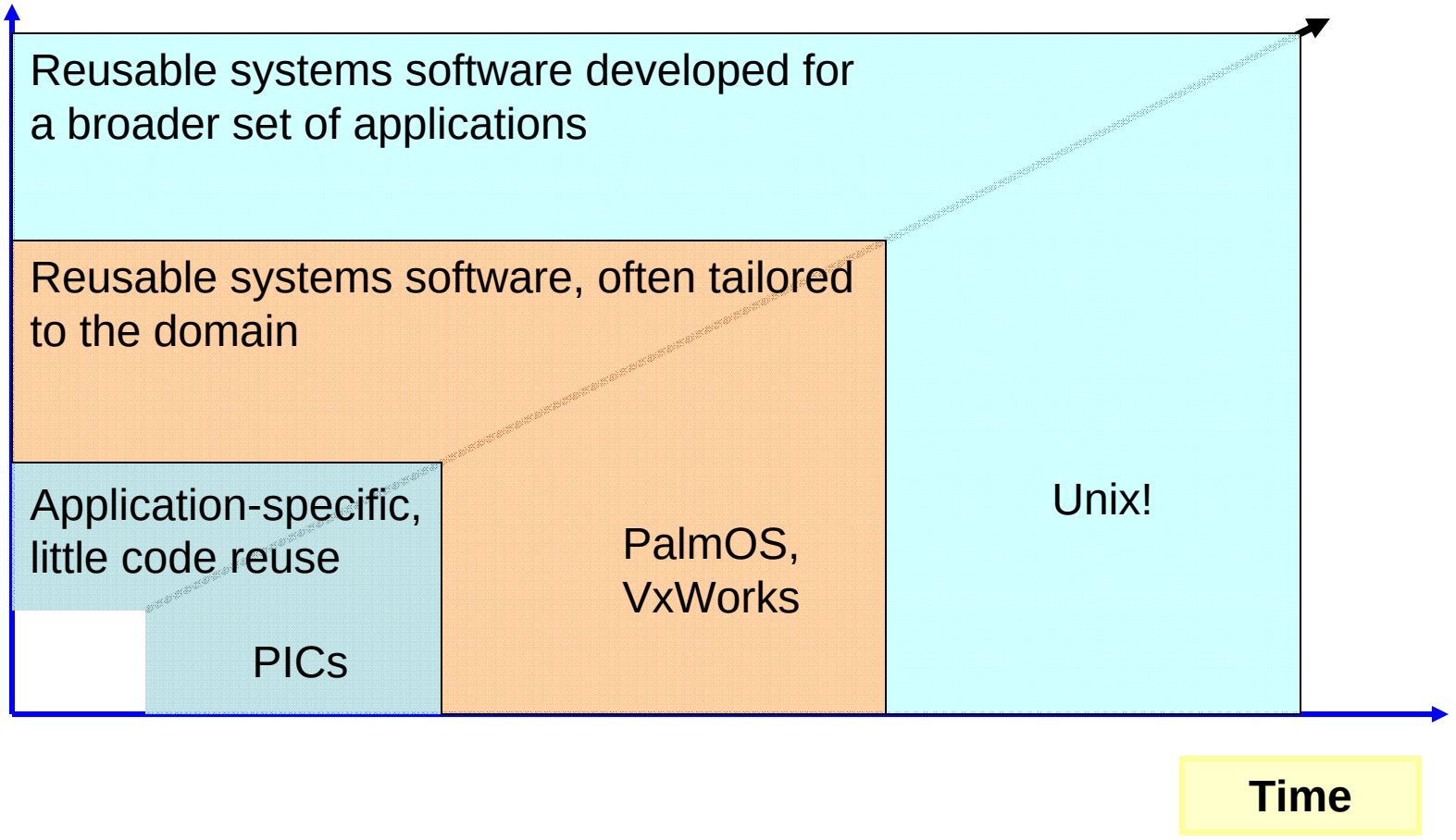
Remember this guy?

If we don’t build it, a physicist will

Not If, but When and How…

Processor Capability

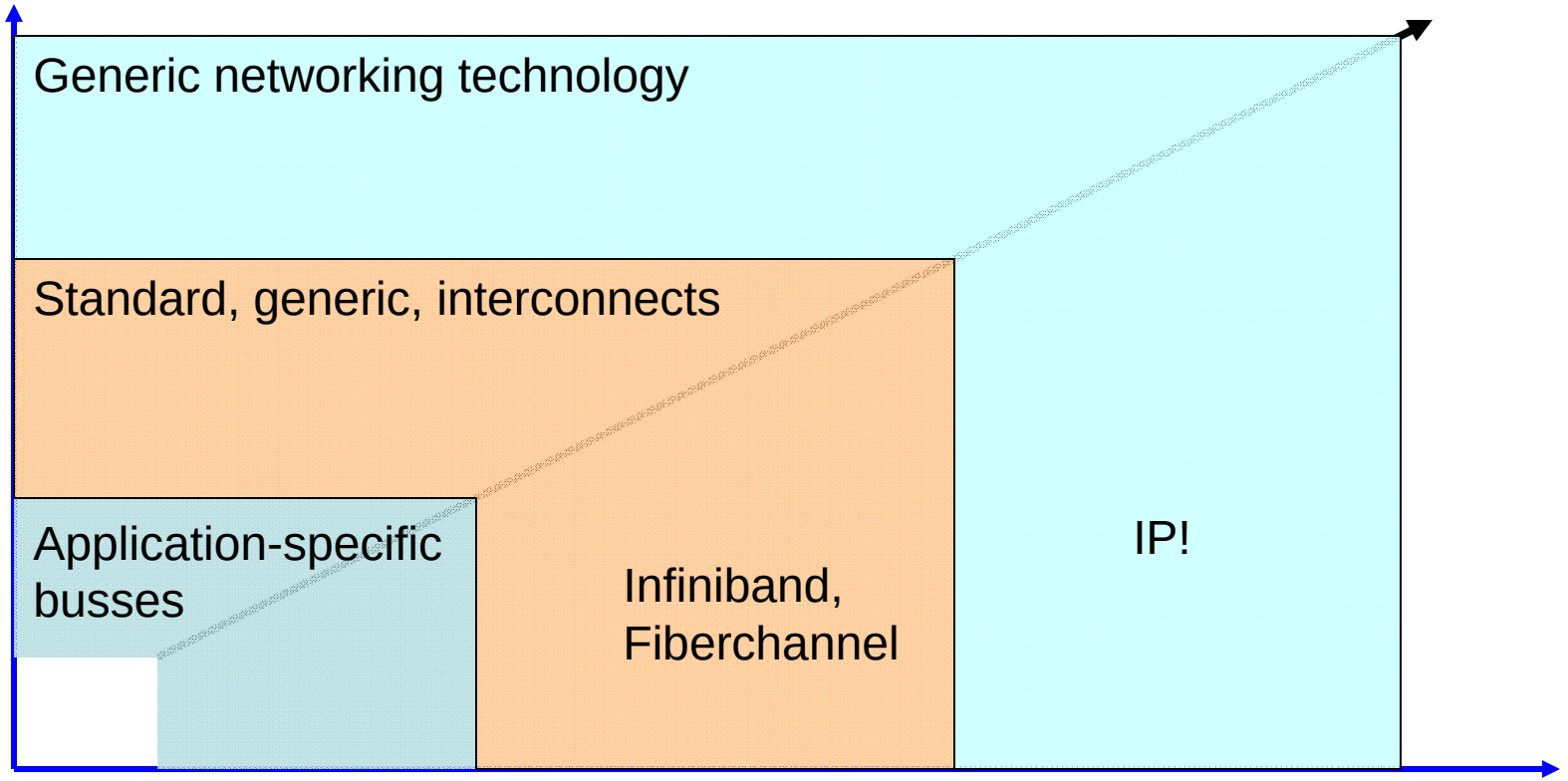
A lesson from embedded-system OSs



Not If, but When and How…

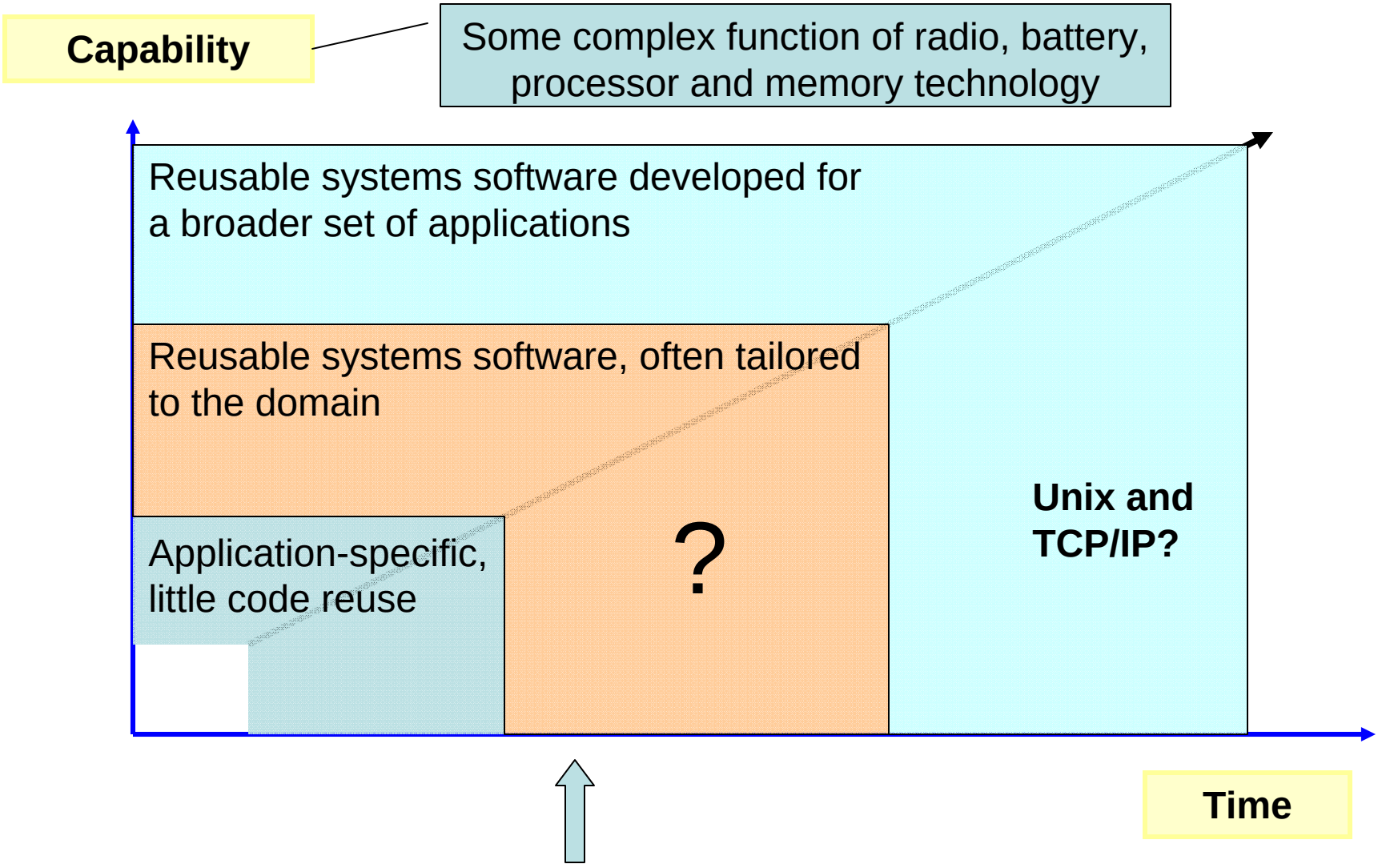
Transfer speed

A lesson from high-speed busses

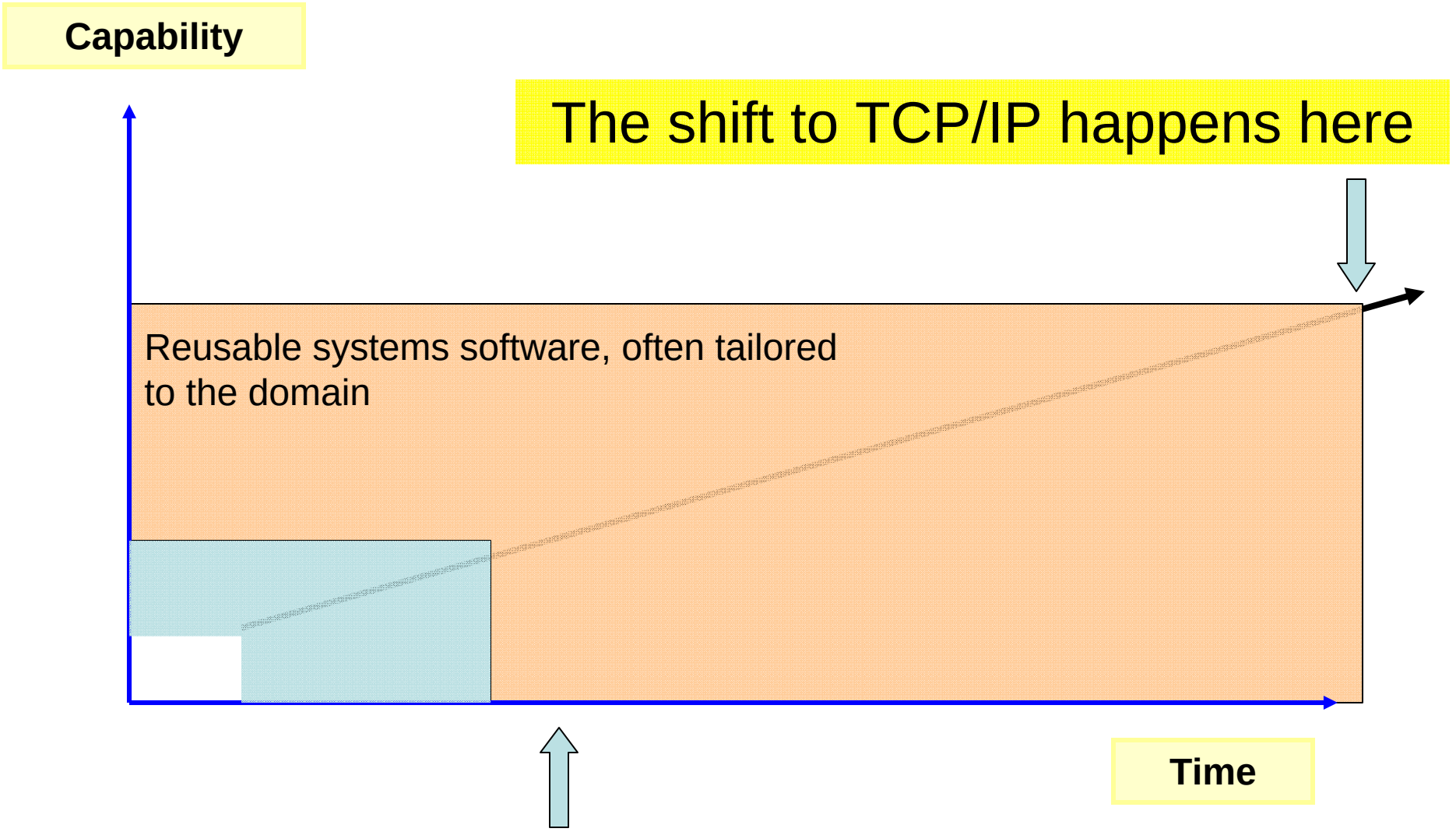


Time

Not If, but When and How…

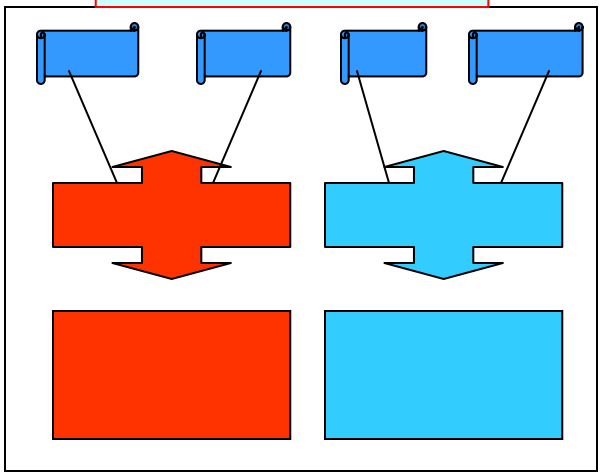


When?

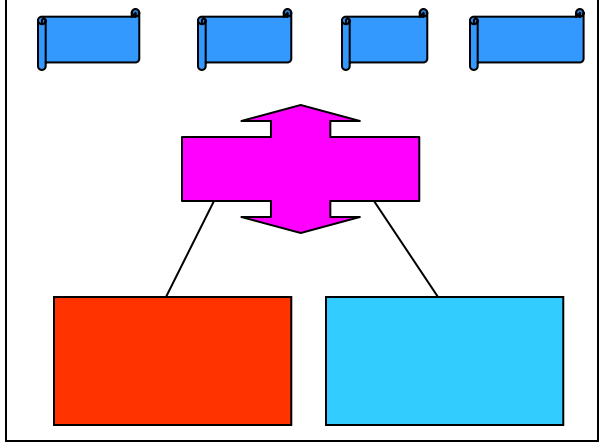

Capability
The shift to TCP/IP happens here
Reusable systems software, often tailored to the domain
Time

How?

One GPSS per application class

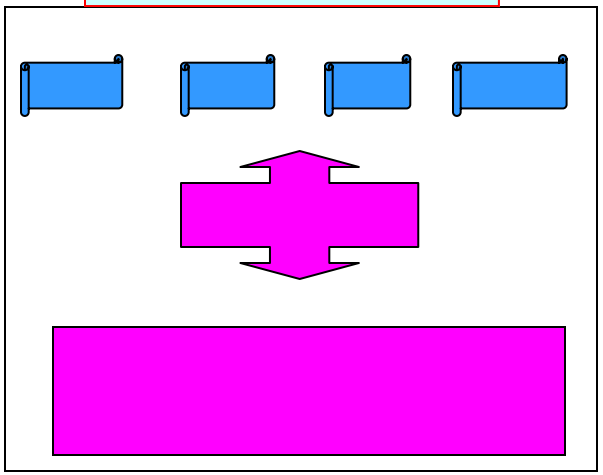


One GPSS abstraction, separate implementations



Most intellectually challenging?

One GPSS fits all



Higher efficiency

Greater reuse

Will we get to TCP/IP?

I mean true IP and TCP to the edges

- Not packet translation, not just IP headers

Answer

- Yes, but after a long time
- Long enough to seriously consider implementing a domain-tailored GPSS

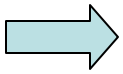
Why?

- Current Internet architecture doesn't handle intermittently connected wireless multi-hop nets well

Outline

What is a GPSS?

Do we need GPSSs?



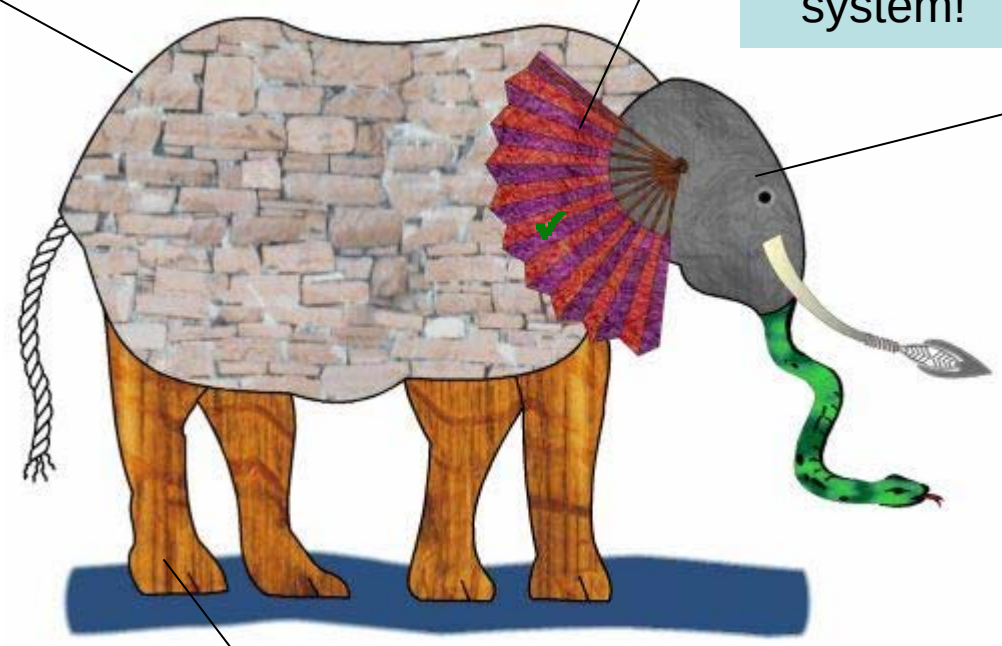
How have GPSSs evolved?

A Personal Odyssey

It's a shared-memory distributed system!

It's a publish-subscribe system!

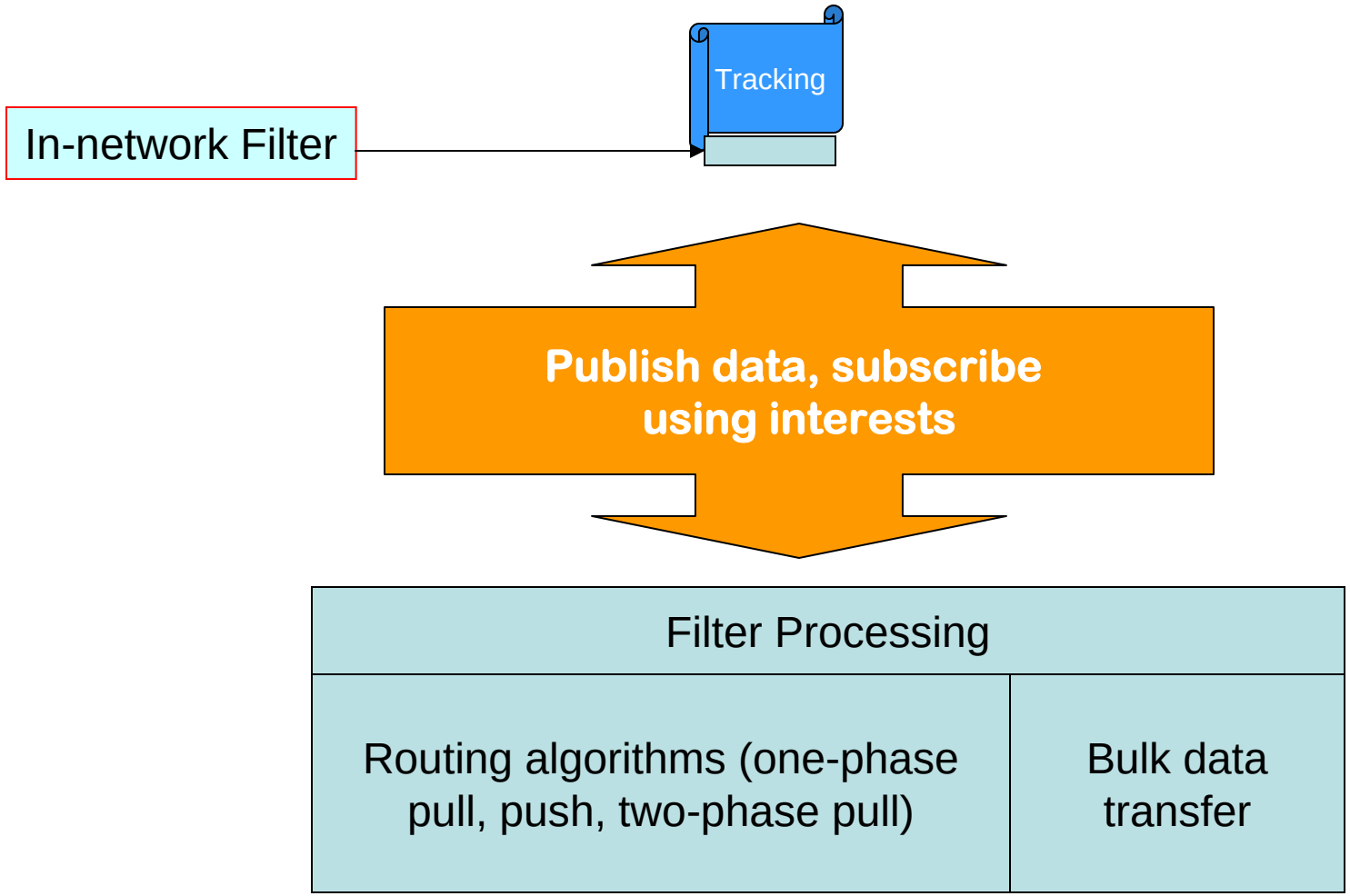
It's a database!



It's a tiered network!

The work of many, many collaborators

Directed Diffusion



Experience

Nice idea, novel at the time

- Directly attacked energy-efficiency

Implementation and significant experience with tracking

- PARC IDSQ

Why did it fail?

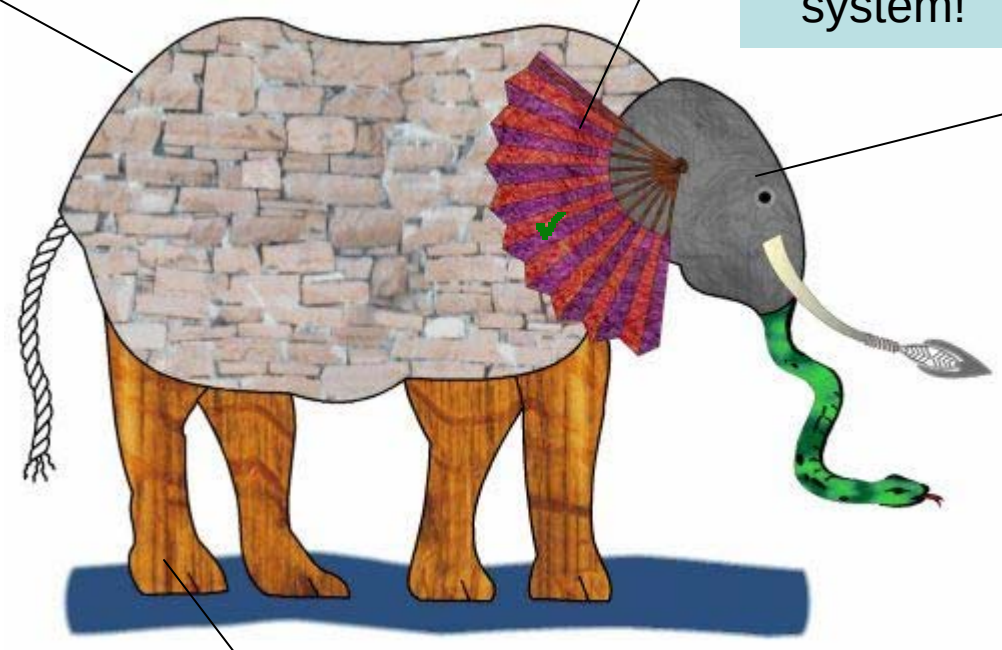
- Mote implementation was hard
- Complex routing algorithms
- No natural way to express actuation

A Personal Odyssey

It's a shared-memory distributed system!

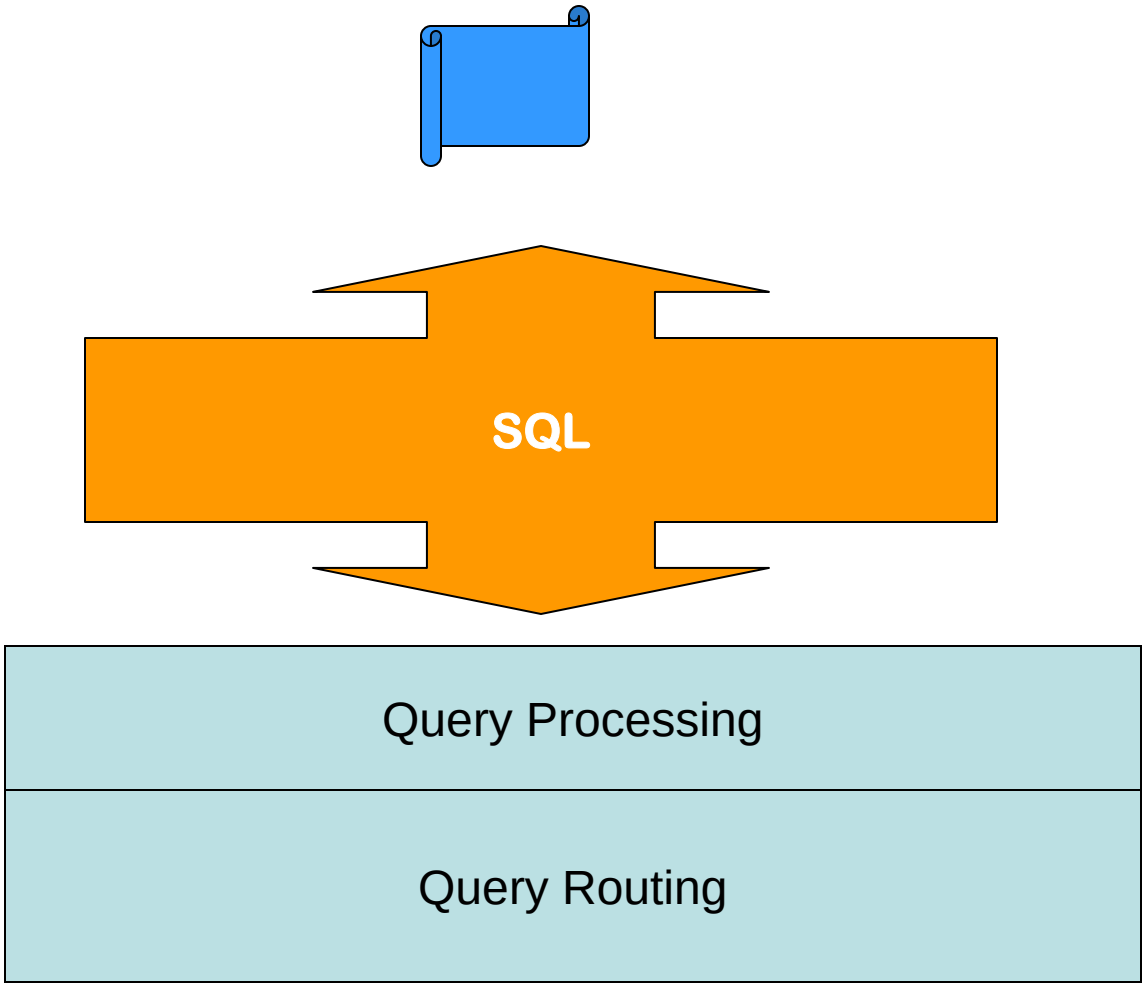
It's a publish-subscribe system!

It's a database!



It's a tiered network!

Sensornet as Database

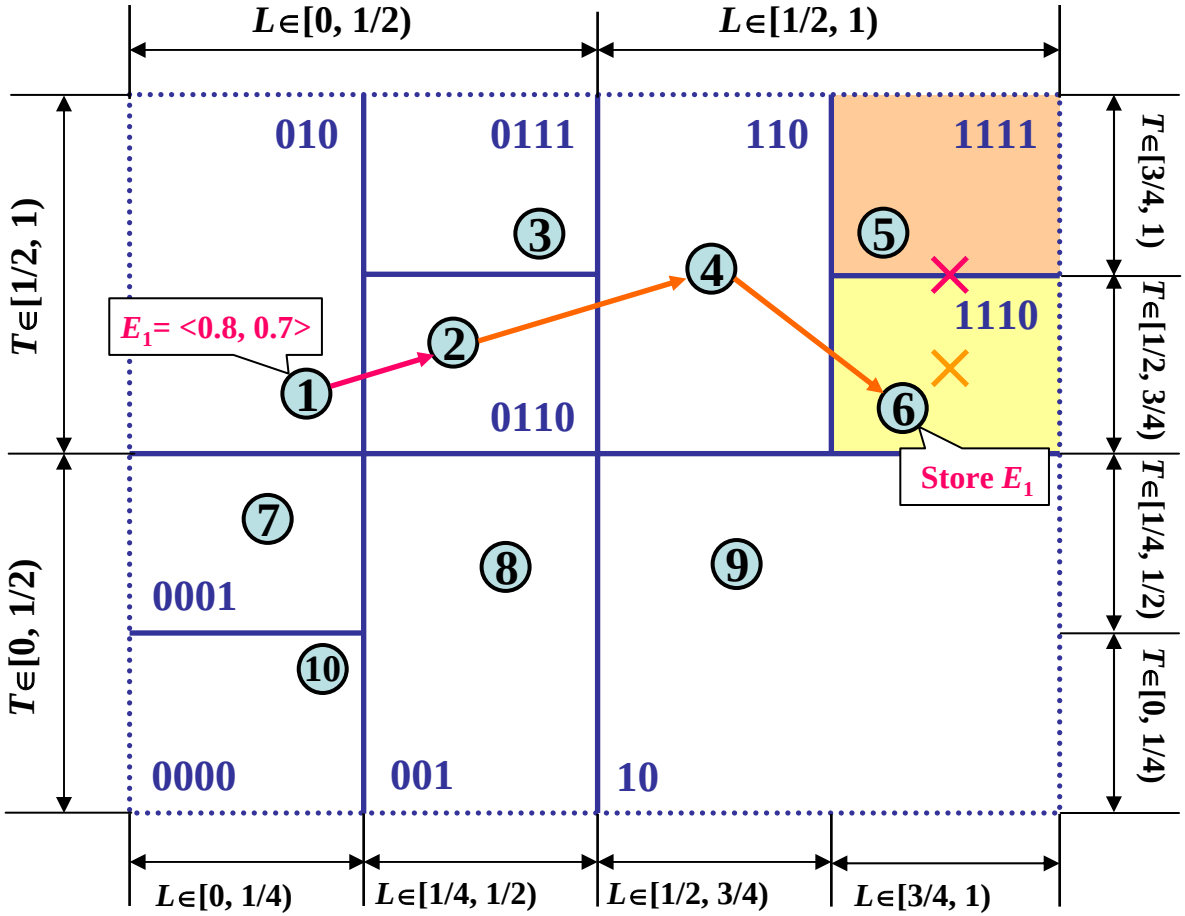


Supporting Multi-Dimensional Range Queries

Which sensors have light readings between (10,20), and temperature readings between (50,100)?

Supporting Multi-Dimensional Range Queries

- Encode events
- Compute geographic destination
- Use geo-routing
- Intermediate nodes can refine the destination estimate
- Queries work similarly



***L*: Light, *T*: Temperature**

Experiences

Some interesting research directions

- Moved away from energy-efficiency

Why did it fail?

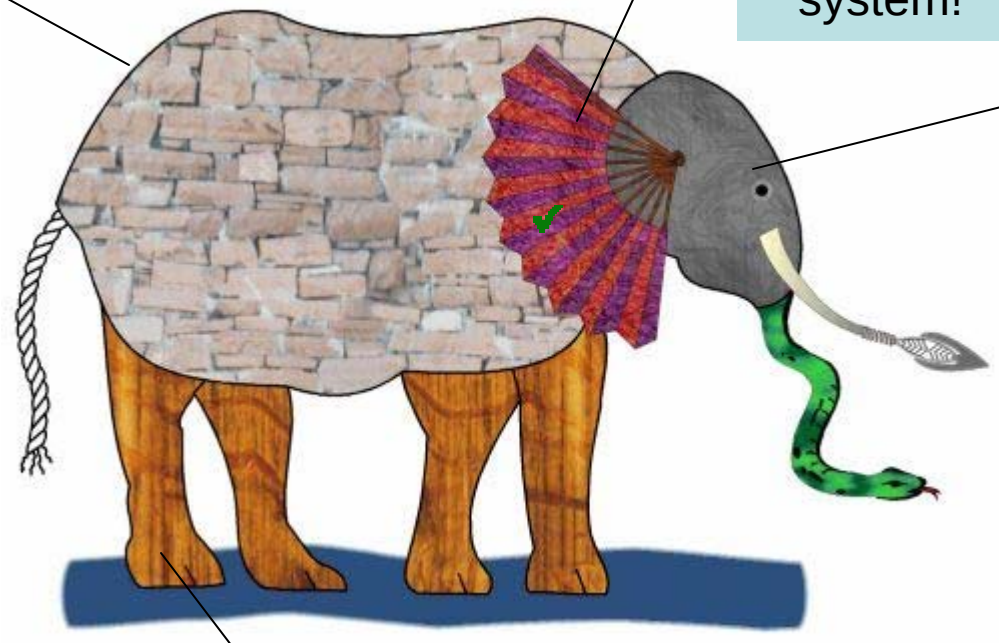
- We were left looking for an application
- Existing geo-routing protocols didn’t work
  - Had to invent our own, but that took time
- Again, no natural way to express actuation

A Personal Odyssey

It's a shared-memory distributed system!

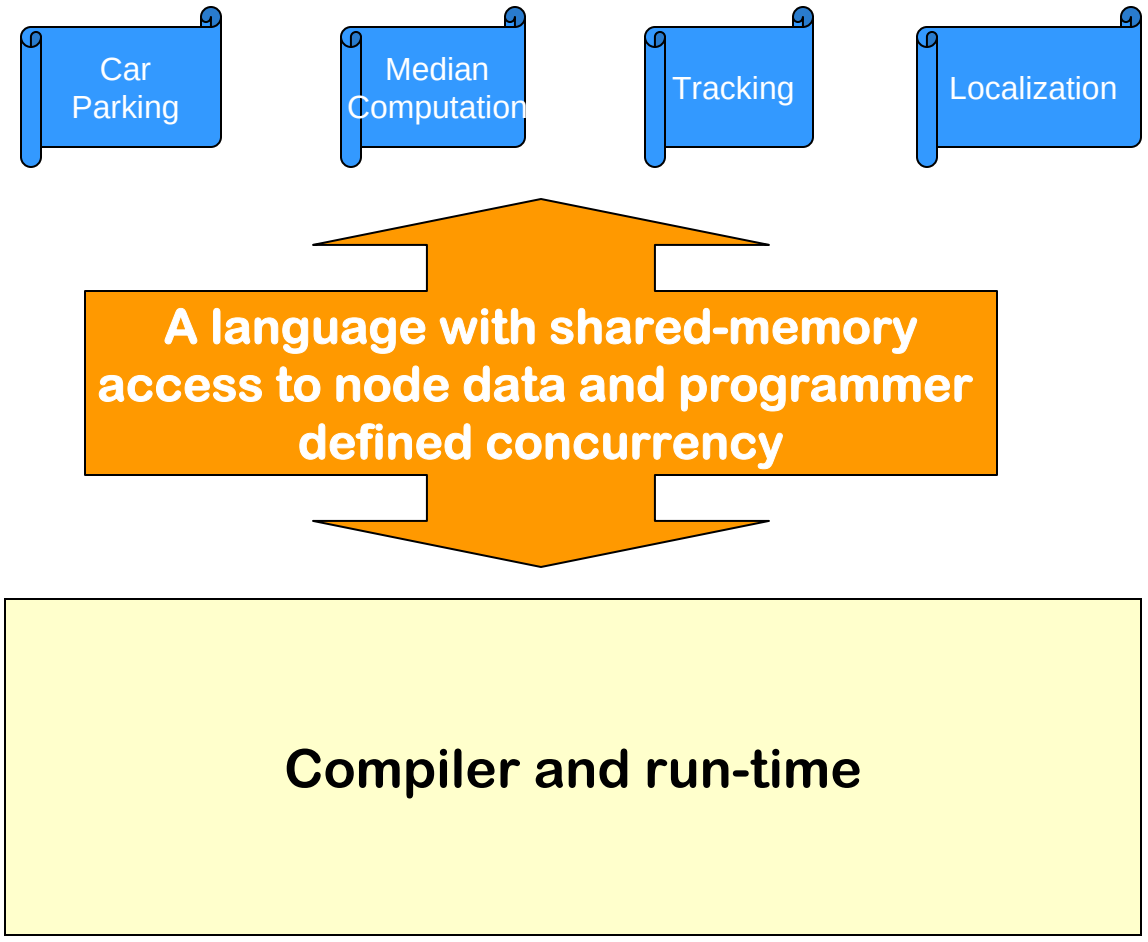
It's a publish-subscribe system!

It's a database!



It's a tiered network!

The Pleaides System



The Basic Idea

Central program that specifies application behavior

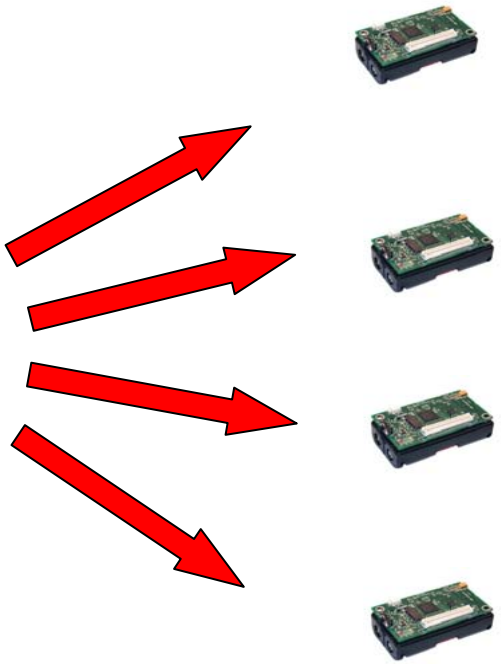
Compiler

Node-local program written in nesC

+

Runtime

Compiled to mote binary



Change of Perspective

```
int valLOCAL;

void main() {
        node_list     all = get_available_nodes();
        int           max = 0;

        for (int i = 0, node n = get_node(all, i);
                n != -1;
                                        max = val@n;
        }
}
```

Easily recognizable maximum computation loop

Pleaides Language

```
int val LOCAL;

void main() {
        node_list     all = get_available_nodes();
        int           max = 0;

        cfor (int i = 0, node n = get_node(all, i);
                n != -1;
                n = get_node(all, ++i)) {
                        if (val@n > max)
                                max = val@n;
        }
}
```

Node-local variable

Central variable

List of nodes in network

Network Node

Access node-local variable at node

Concurrency Primitive

Experience

Approach forces focus on *consistency*

- Lots of interesting compiler challenges

What's been hard to think about?

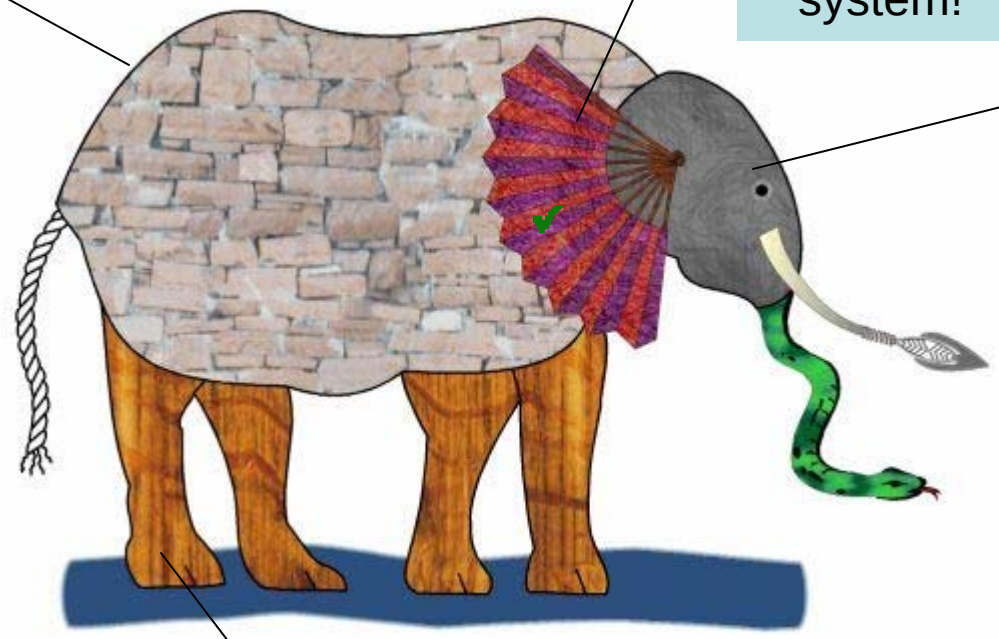
- Robustness vs. consistency tradeoff
- Energy-efficiency
- Perhaps more suited to a network of larger embedded systems?

A Personal Odyssey

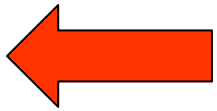
It's a shared-memory distributed system!

It's a publish-subscribe system!

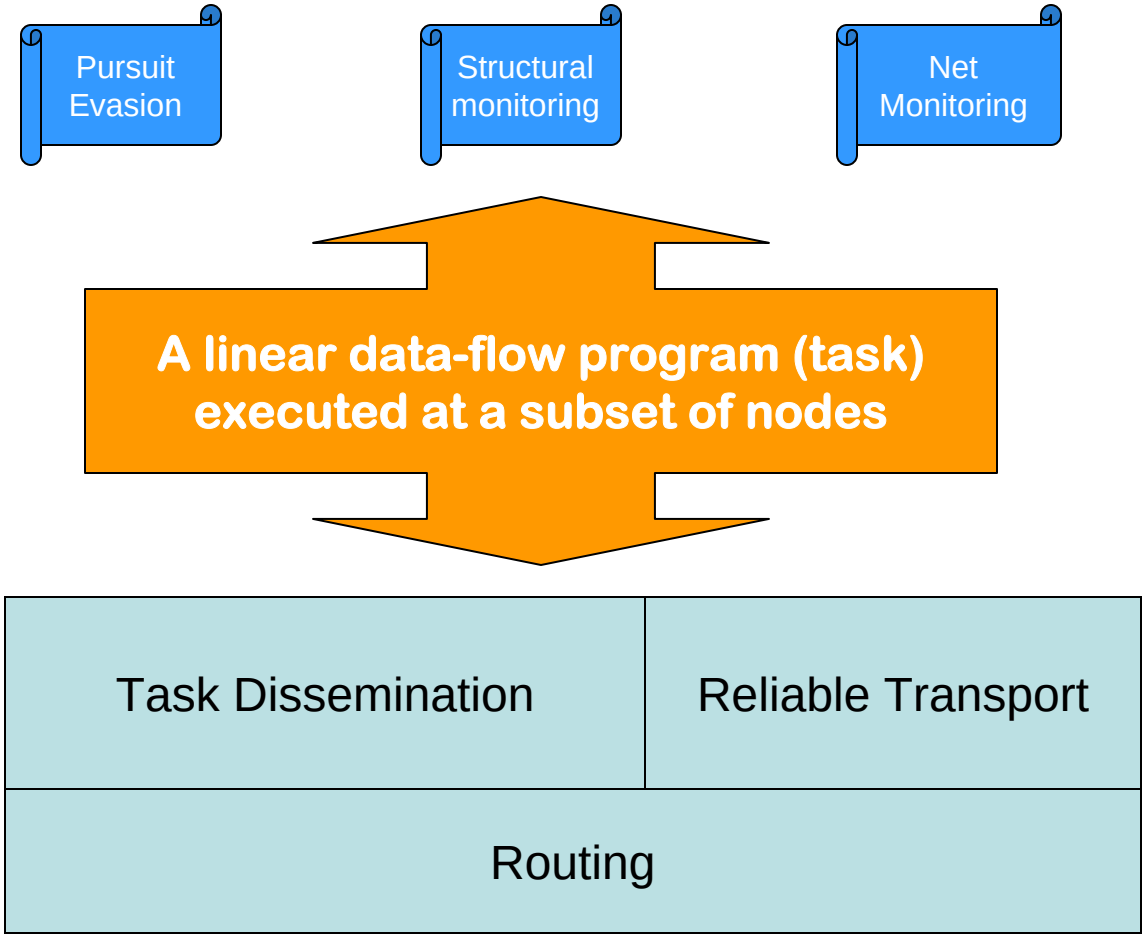
It's a database!



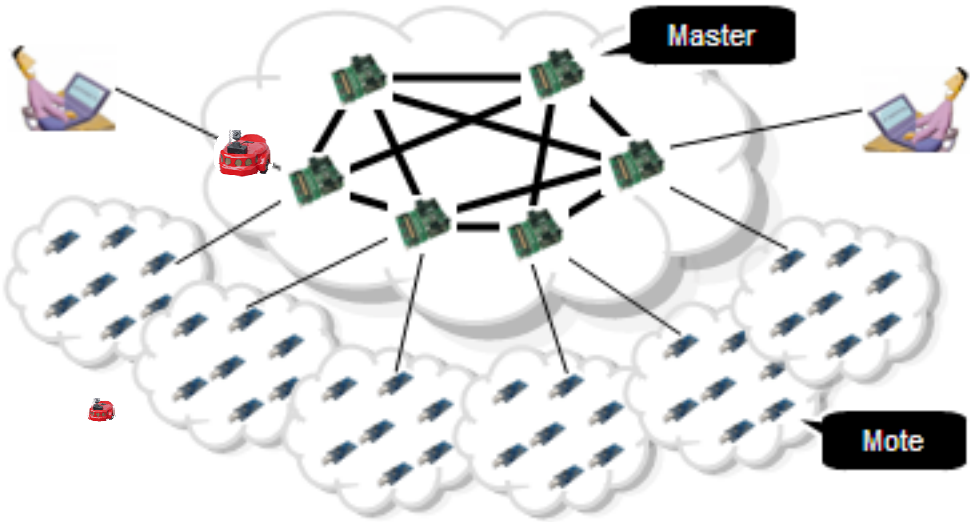
It's a tiered network!



The Tenet System



Tiered Networks



Masters

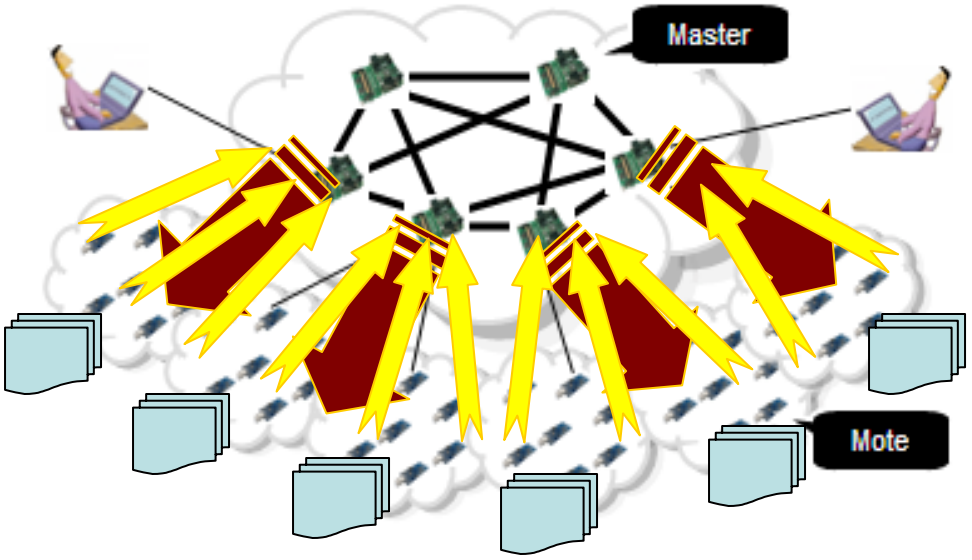
Provide greater network capacity, larger spatial reach

Motes

Enable flexible deployment of dense instrumentation

Some deployments will have mobile elements in one or both tiers

Tenet Architecture



Masters control motes

Applications run on masters, and masters task motes

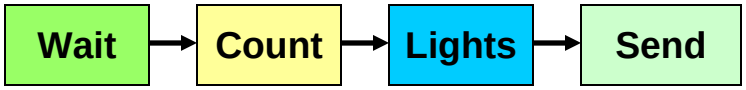
Motes process data,

and may return responses

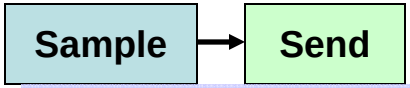
No collaborative processing in the mote tier

Tasks in Tenet

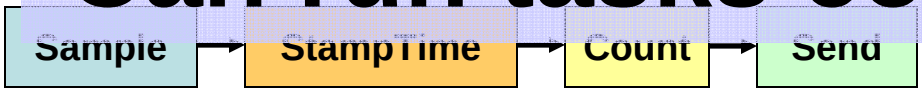
CntToLedsAndRfm



SenseToRfm

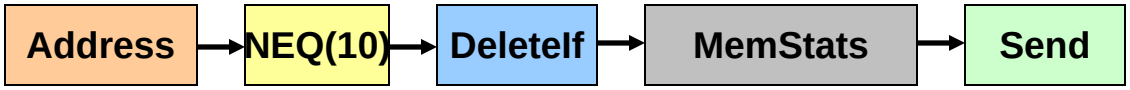


With time-stamped sensor data

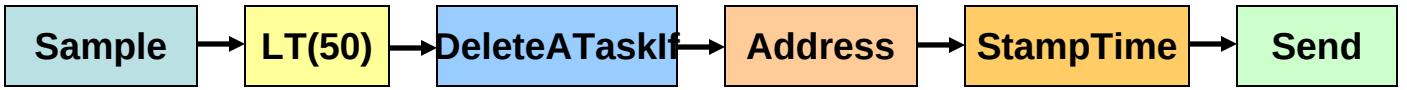


Can run tasks concurrently

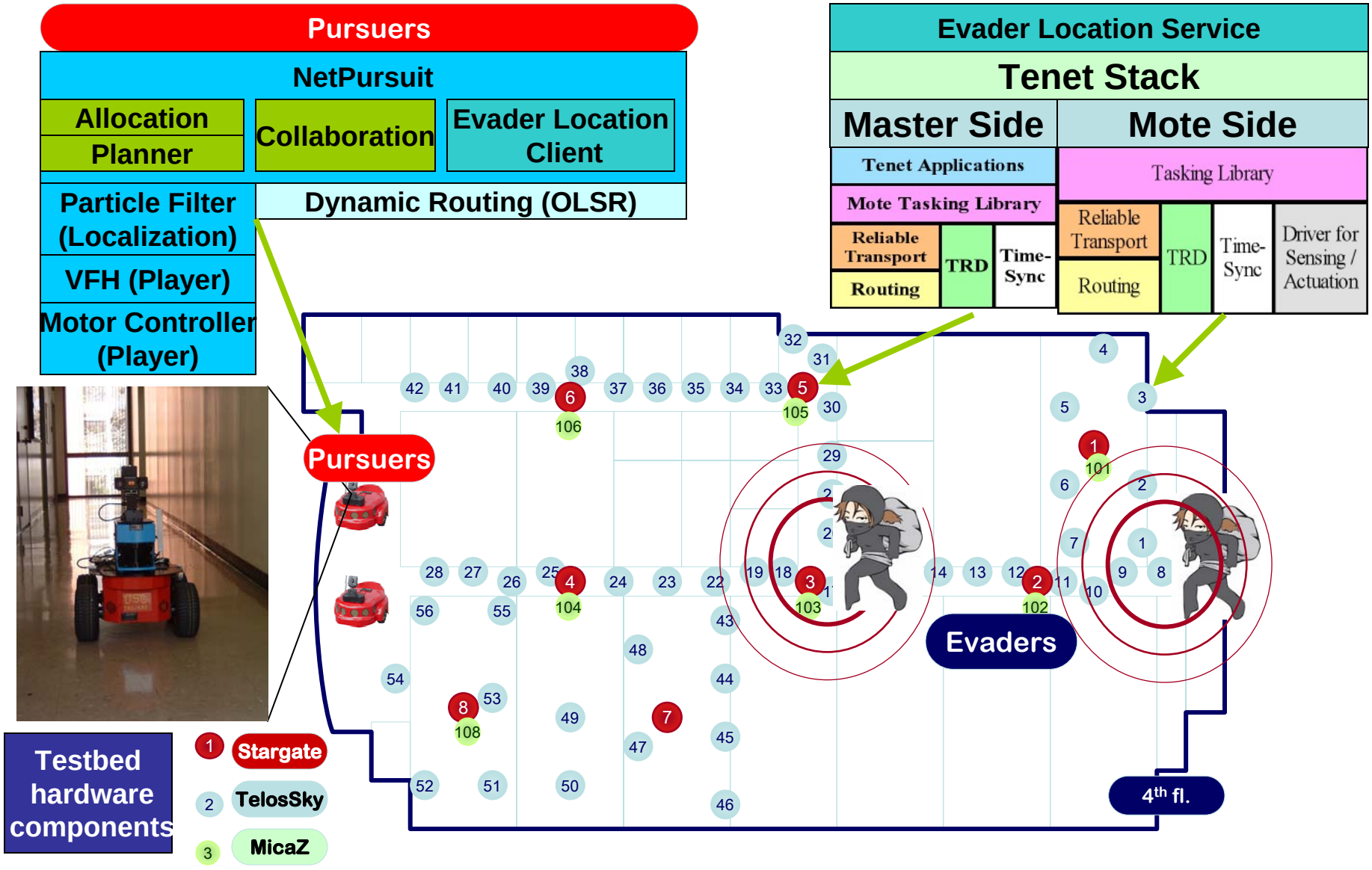
Get memory status for node 10



If sample value is above 50, send sample data, node-id and time-stamp



Pursuit-Evasion Games



Deployment on Vincent Thomas Bridge

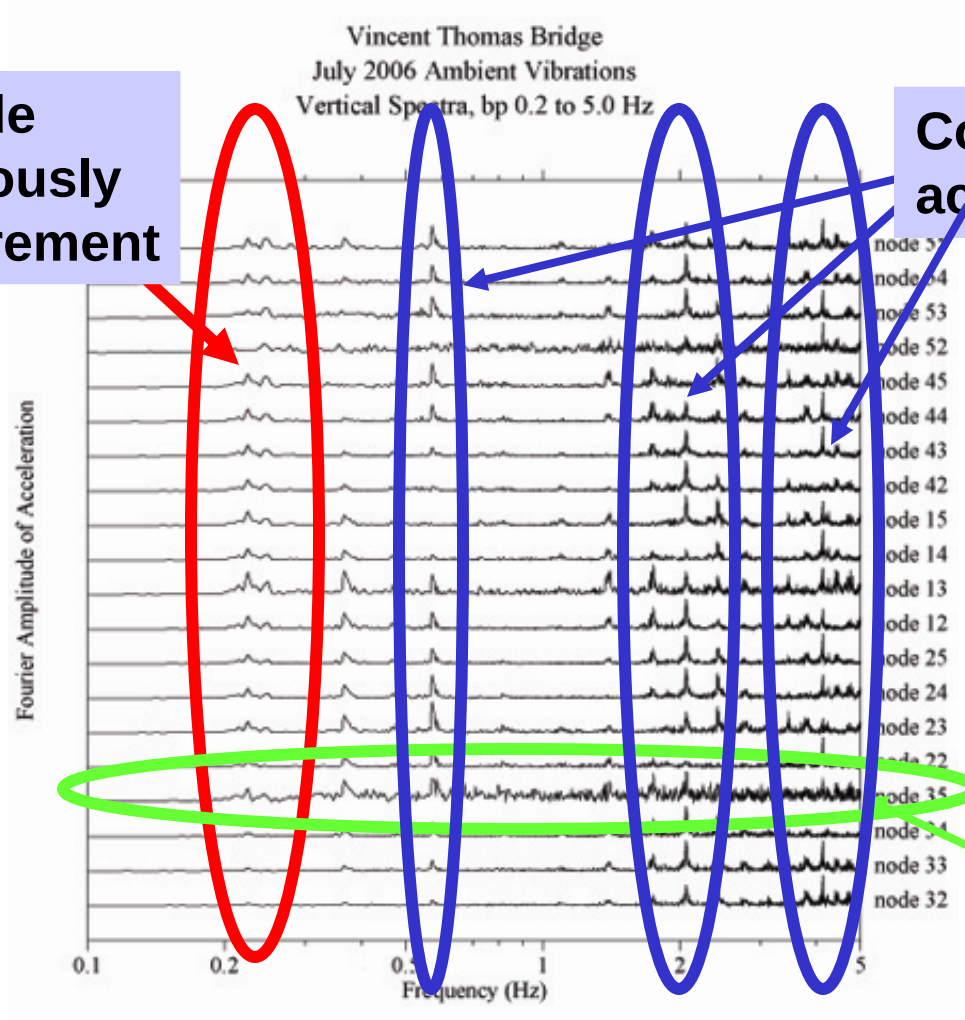
Master

Mote

Ran successfully for 24 hours
100% reliable data delivery
Deployment time: 2.5 hours
Total sensor data received: 860 MB

Interesting Observations

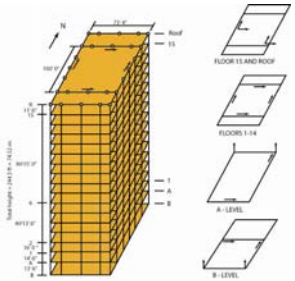
Fundamental mode agrees with previously published measurement

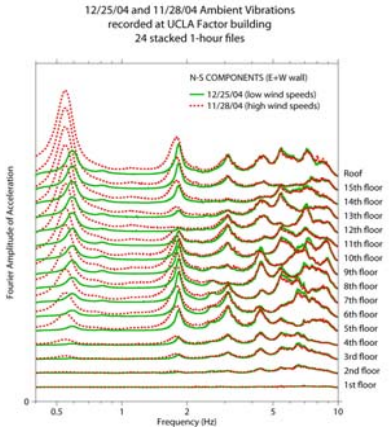


Consistent modes across sensors

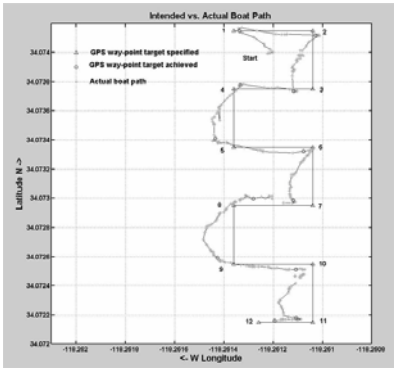
Faulty sensor!

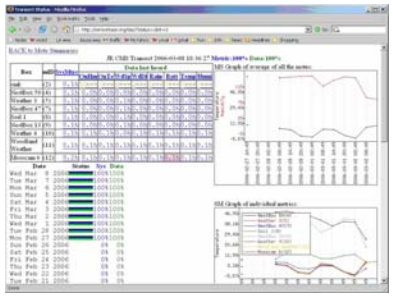
Planned Deployments





ShakeNet

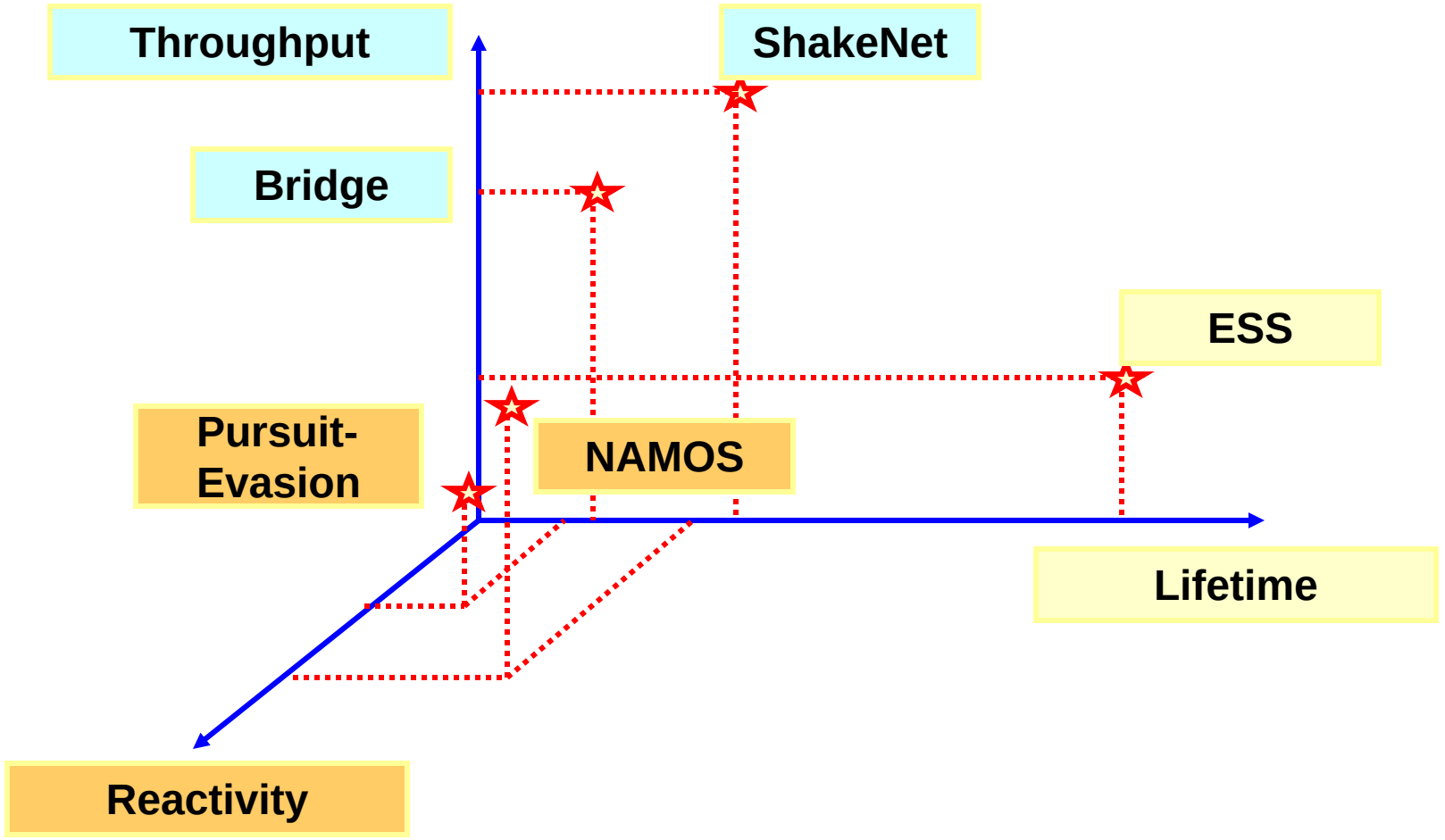




NAMOS

CENS ESS

Deployments in Context



Concluding Remarks

To me, a Tenet-like system…

- feels *right* as a GPSS

But…

- Need to show reasonable solutions for lifetime, security

Other Feasible GPSSs

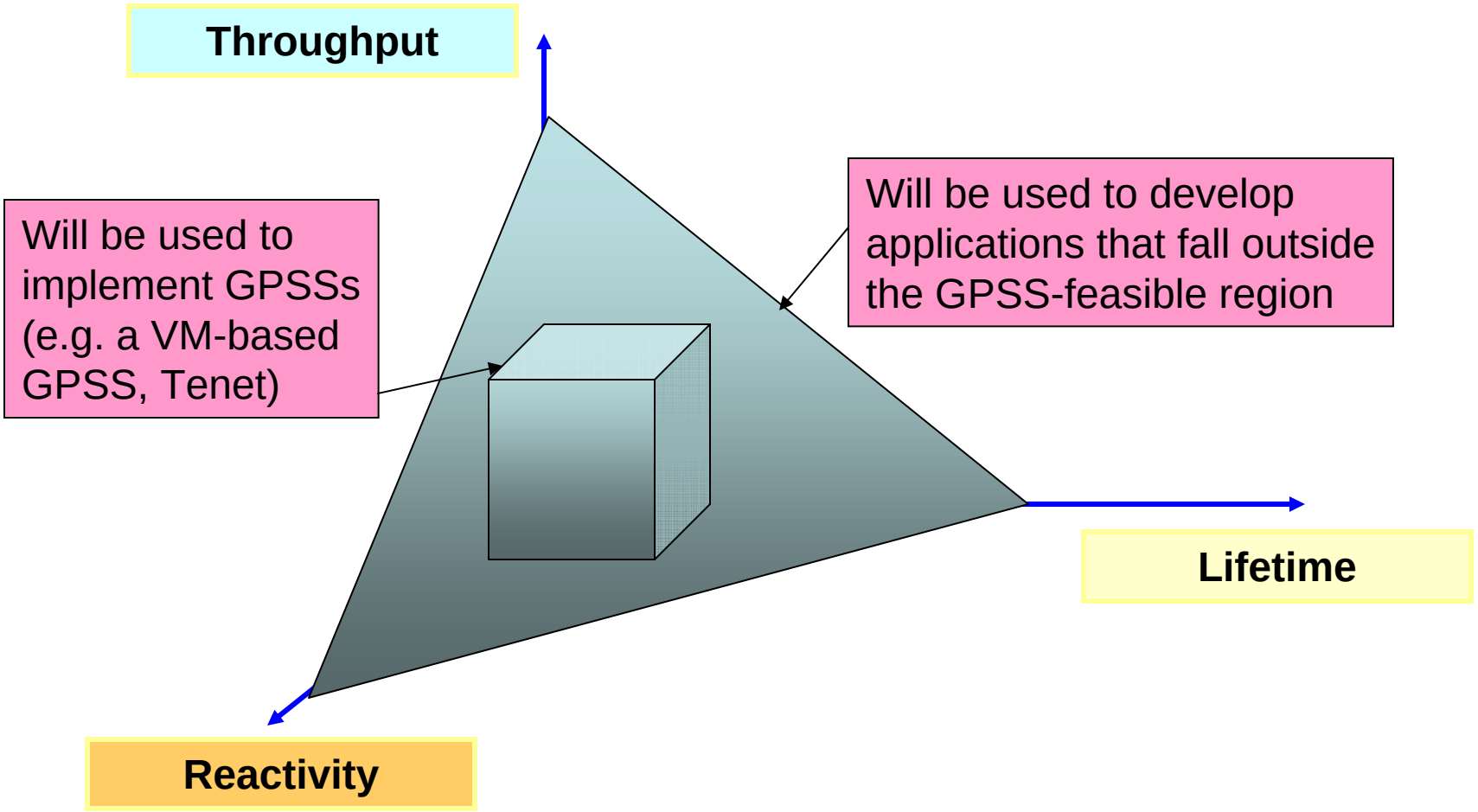
Other node programming approaches

- Mate, ASVMs, others
- State machines

Other macro-programming approaches

- Regiment
- Task graphs

What about TinyOS?



Concluding Remarks

It is time for a shift in thinking

- Concerted move towards building *most applications* on one/two general-purpose systems
- Put large deployments using these systems in place
  - Guaranteed to learn a lot

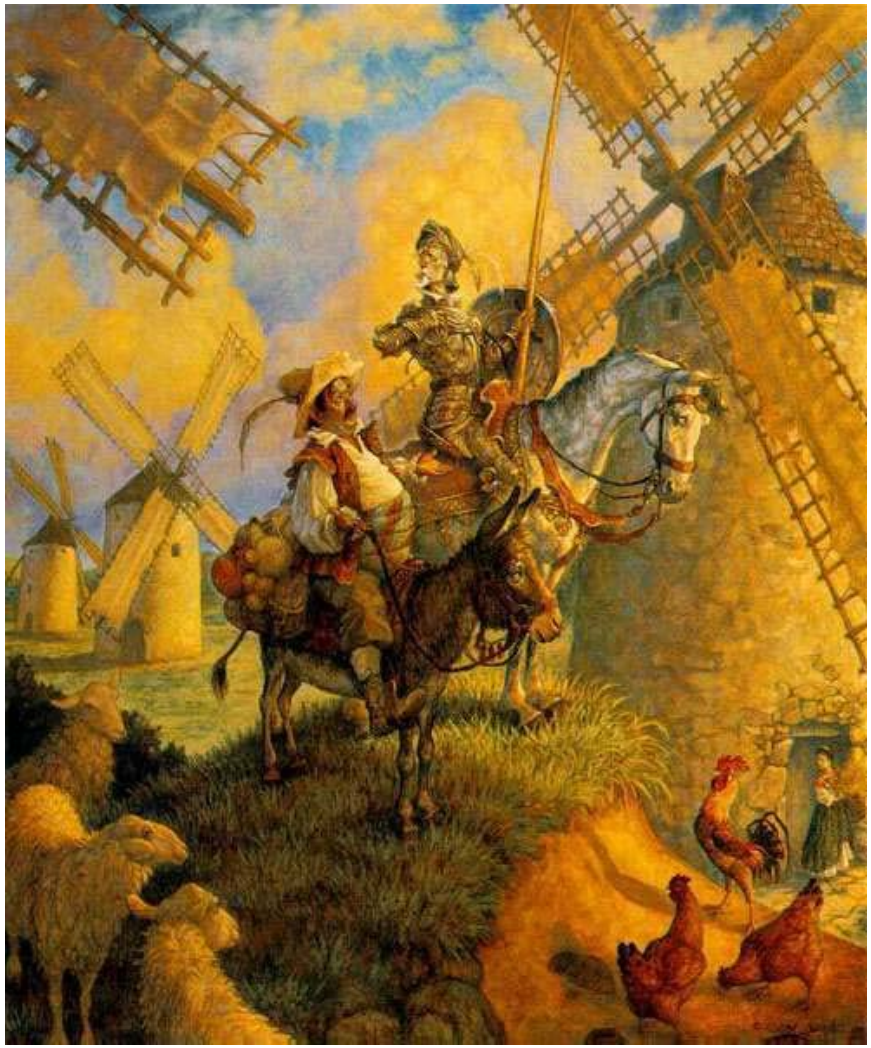
Lots of interesting times ahead!

Concluding Remarks

But…

- Given that I've been wrong at least 3 out of 4 times
- You may have wasted a perfectly good hour listening to me!

I'm very grateful that most of you chose to sit through the talk!



Brickbats?