

Network Edge Considered Harmful

... with apologies to Edsger Dijkstra

Bob Iannucci, Ph.D.

Distinguished Service Professor, ECE

Director, CyLab Mobility Research Center

<http://sv.cmu.edu/bob>

November, 2018

Agenda

- A bit of history
- A few thoughts

Iannucci's Law

*In each generation of computing, the emergence of a **standard platform** transforms the industry by shifting value from hardware to software and services.*

History: Computing Platforms, Past and Present

	Golden Age	Standard Platform	Compatibles	Value Shift to Software	Value Shift to Services
Mainframe			 Magnuson Computer		
Mini					
PC					
Mobile		Mobile Apps plus cloud services			

In each case, the platform made programming simple enough to attract many developers

Assumption: the next Platform will be Cyber-Physical

An *integration* of

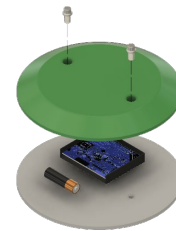
1. Flexible, powerful, programmable, open device families
2. Flexible, powerful, programmable, open wired and wireless networks
3. Easy enough to program that millions of programmers will invest their time and creativity

Observations: #1 mostly exists. #2 & #3 do not.

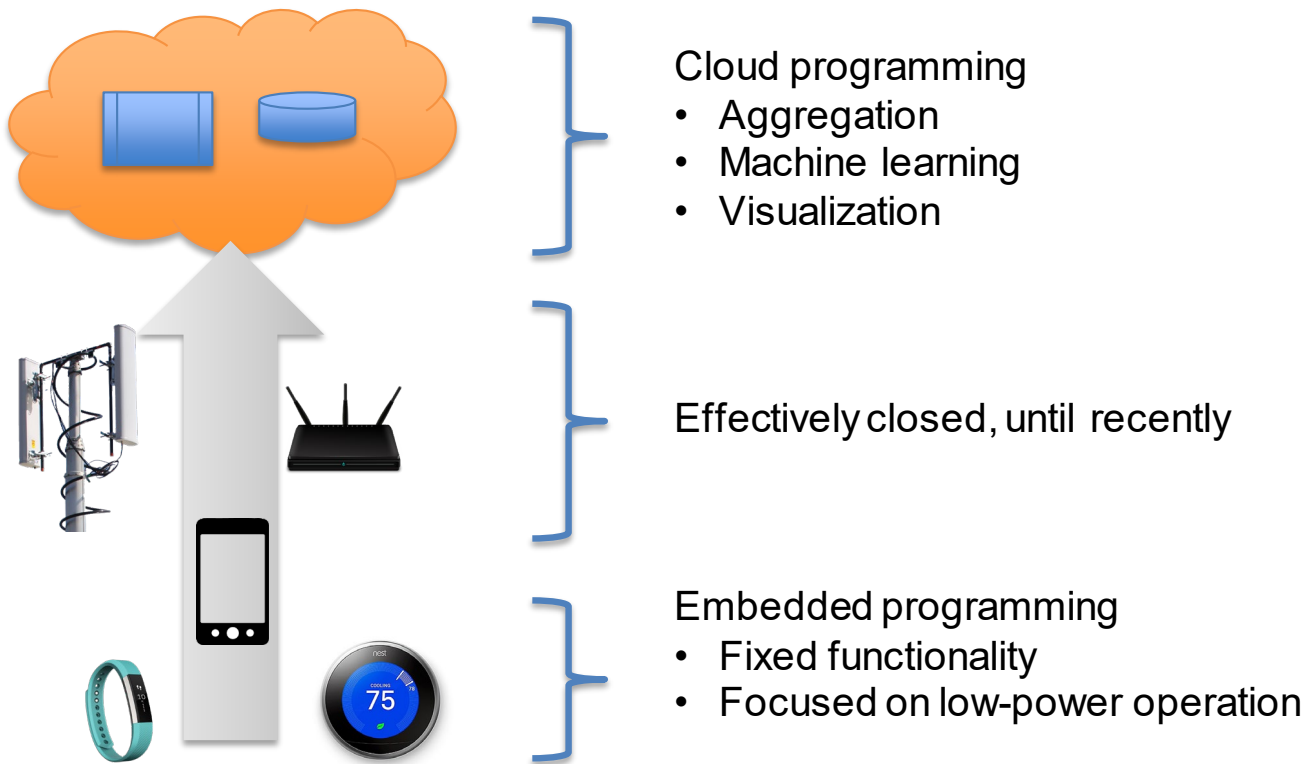
Why?

CPS Platform

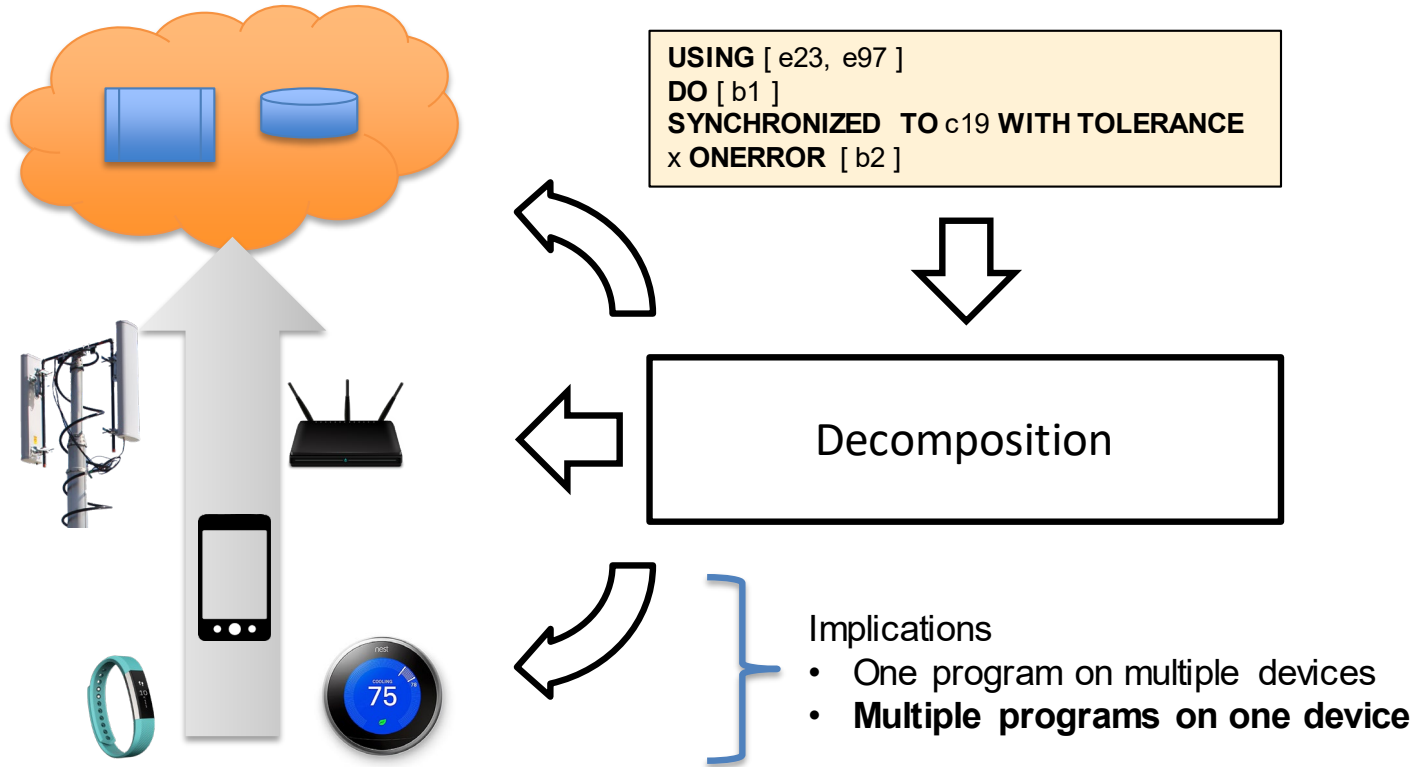
- Millions of developers
- Each can write a **single app** that uses deployed CPS sensors and actuators to sense-compute-actuate
 - Separating the questions of logical correctness from the practical matters of partitioning, migration, latency minimization
- CPS makes this hard
 - Managing time in a large distributed system
 - Scale + Wireless = Inevitable Failures
 - Managing **scarce energy** in OHIO, **multi-tenant** devices
- Does exposing the Edge make this easier or harder?



Programming Today's Distributed Systems



CPS Programming for the Millions

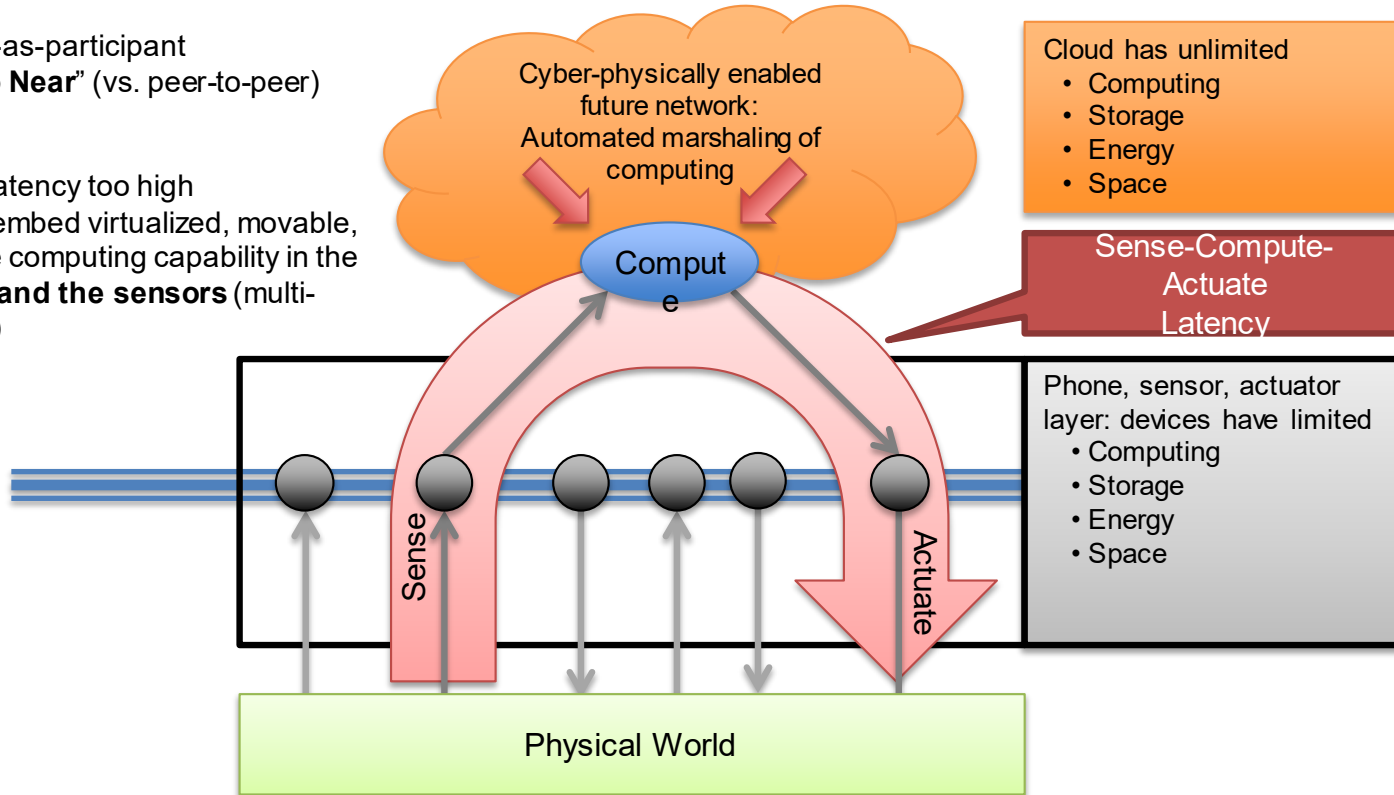


So, Should We Expose or *Hide* the Edge?

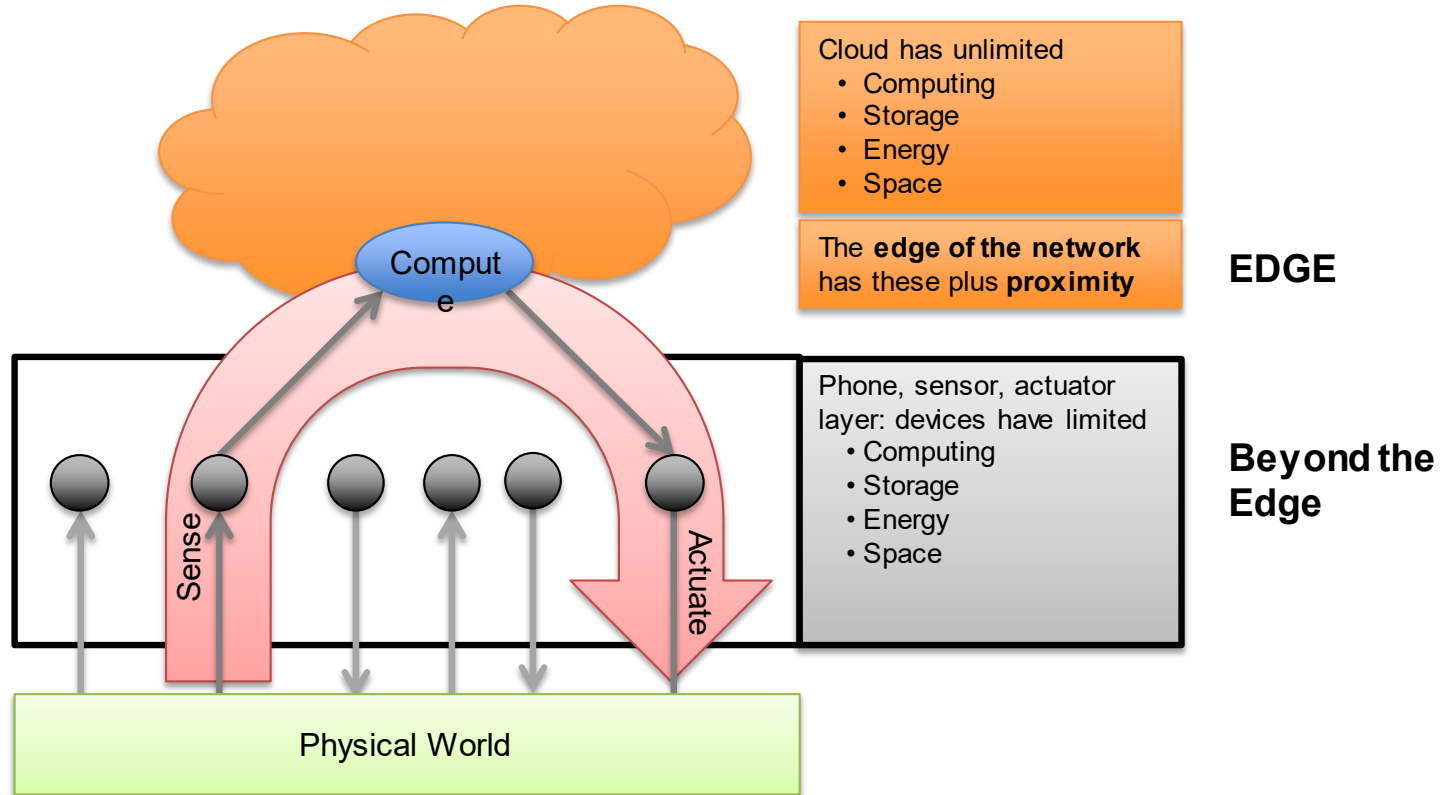
Network-as-participant
“Near to Near” (vs. peer-to-peer)

Today: latency too high
Future: embed virtualized, movable, real-time computing capability in the network **and the sensors** (multi-tenancy)

The
Time
Line

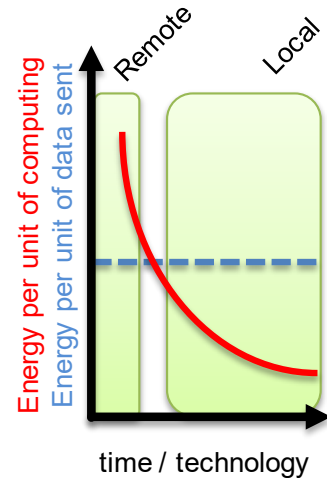
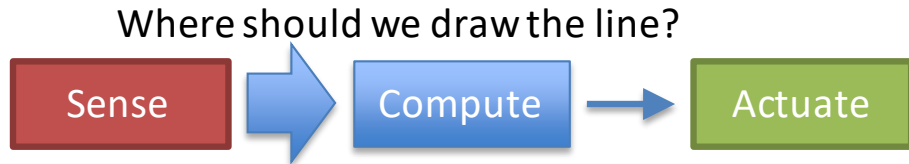


By the Way... Terminology Creep



Energy Cost of Computing On-Device Declining: Devices Becoming Virtualized, Multi-Tenant Hosts

- **Interplanetary travel** design approach
- **Pack your energy** for the mission
 - May harvest a little along the way
- **Distribute intelligently**
 - Local vs. remote computing (incl. machine learning)

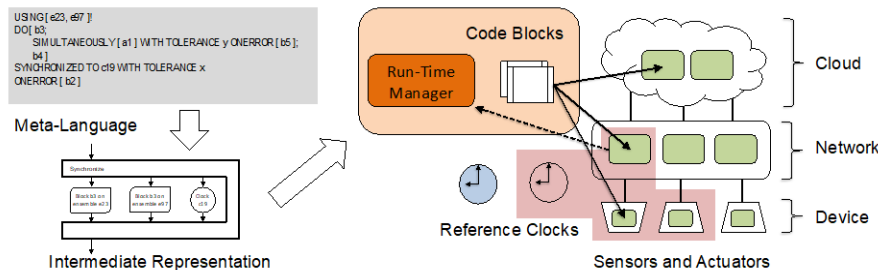


Energy concerns push us toward computing **LOCALLY** (in the sensor)
and only communicating when necessary

Our CPS Vision: Edgeless Computing

- Present: “apps” made smartphones a force of nature
- Future: single “apps” that distribute to the cloud, the **network and devices**
 - Abstract specifications of physical distribution and temporal constraints
 - Runtime that manages real distribution and real synchronization
 - Exposing the “or else...” cases to the programmer for inevitable failures
 - Separately address program correctness and program distribution
 - Integrated federation services (*e.g.*, search engines for sensed data)

The
TickTalk Language
Concept
(joint work with
Aviral Shrivastava, ASU)



Summary

- Profound economic transformation from each generation of computing is predicated on the emergence of a platform that makes programming easy
- The next platform is likely CPS
- Making programming easy requires us to hide complexity
- Hiding, rather than exposing, the Edge may accelerate arrival of the CPS platform
- Re-invigorate and refocus work in distributed programming and real-time computation from hard verification to graceful degradation
- **Visit Poster #34 to learn about TickTalk**

Schedules → Hints

Partitions → Containers

Golden clock → Multiclock, Timing islands

Verification → Programmer-managed failure

Space-Time Memory → Statistical Stream Algorithms

As predictable as possible → Adequate predictability subject to power/energy constraints

Carnegie Mellon University
CROSSMobile

Thank you!

<http://sv.cmu.edu/bob>