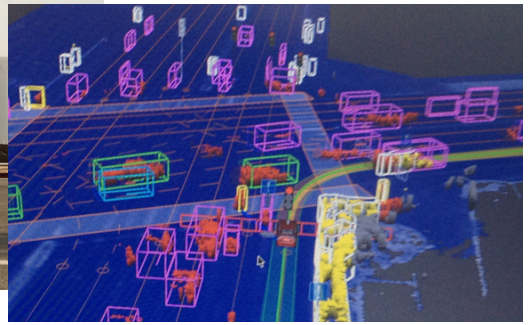
Mobile and Ubiquitous Computing

Today

- Mobile, pervasive and volatile systems
- Association and Interoperation
- Sensing context and adaptation



How is mobility different

- Mobile elements are resource-poor
  - Relative to static elements; for a given cost and level of tech
  - Better in the absolute, but still relatively poor
- Mobility is inherently hazardous
  - Loss, stolen, damage …
- Mobile connectivity is highly variable in performance and reliability
  - In and out of building, different floors, outdoors, …
- Mobile elements rely on a finite energy source
  - Even if battery keeps improving, still need to be carefuly about consumption

- Intrinsic features of mobility, not artifact of current tech

Adaptation

- A characteristic of all distributed systems – tension between autonomy and interdependence
  - Cache and consistency
- Mobility exacerbates the tension
  - Relative resource poverty of mobile elements, lower trust and robustness ➔ rely on static servers
  - Unreliable and low-performance network, concerns about energy usage ➔ self-reliance
- Find a balance, but the balance can't be static as conditions change ➔ adaptive mobile systems

Taxonomy of adaptation

- Range of adaptation

	Laissez-faire	App-transparent
System support	No need	All
Adoption	Hard to write apps	Backward compatible
Central resource arbitration & control	No	Yes
Effectiveness of adaptation	Customized	Sometimes inadequate/ counterproductive

- In between, application-aware adaptation
  - Application decide how, but system monitor resources and enforces allocation decisions

Extending the client-server model

- On a client-server model
  - Small number of trusted server host the data
  - Accessed from many untrusted client sites
  - Caching and read-ahead for performance
  - End-to-end authentication and encryption for security
  - Scalability in a core of servers that changes slowly and a much larger, dynamic set of clients
- Rethink the model in mobile settings
  - A blurred distinction – an extended model
  - Clients' resource limitations – certain operations done by client may need to be run at resource-rich servers
  - Uncertain connectivity – ask clients to act as servers

Some research implications*

- Caching metrics
  - A basic idea of cache misses is not enough – how much does it matter to the user? When does it happen?
- Coherent caches and weak connectivity – semantic callbacks and validators
  - Validating on demand reduces frequency (cheaper) but worsen consistency; maintaining coherence at multiple granularities and callbacks can help
- Algorithms for resource revocation
  - Classical work has focused on allocation
- Agile adaptation or instability
  - What's agile enough, are there metrics, guidelines?
- Global estimation from local observations
  - Are there systematic ways to do it? Can we bound the errors?

*M. Satyanarayanan, Fundamental challenges in mobile computing, In Proc. of PODC 1996

Pervasive computing

- *"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it."* – Mark Weiser, 1991
- Way ahead of its time, increasingly current
  - Mobile handheld and wearable
  - Wireless LANs
  - Sensing devices
  - Smart appliances

Problems and research fields*

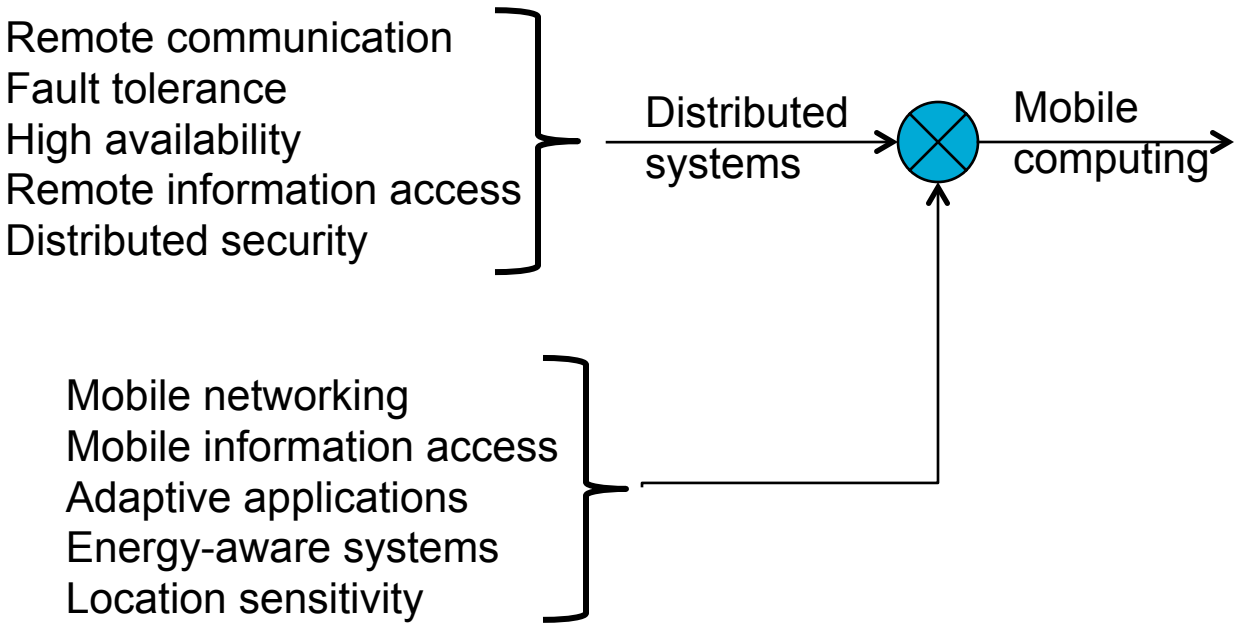
Remote communication
Fault tolerance
High availability
Remote information access
Distributed security

→ Distributed systems

- Mid-1970's through 1990's
- A conceptual framework and algorithmic base that has aged well
- Well codified in textbooks

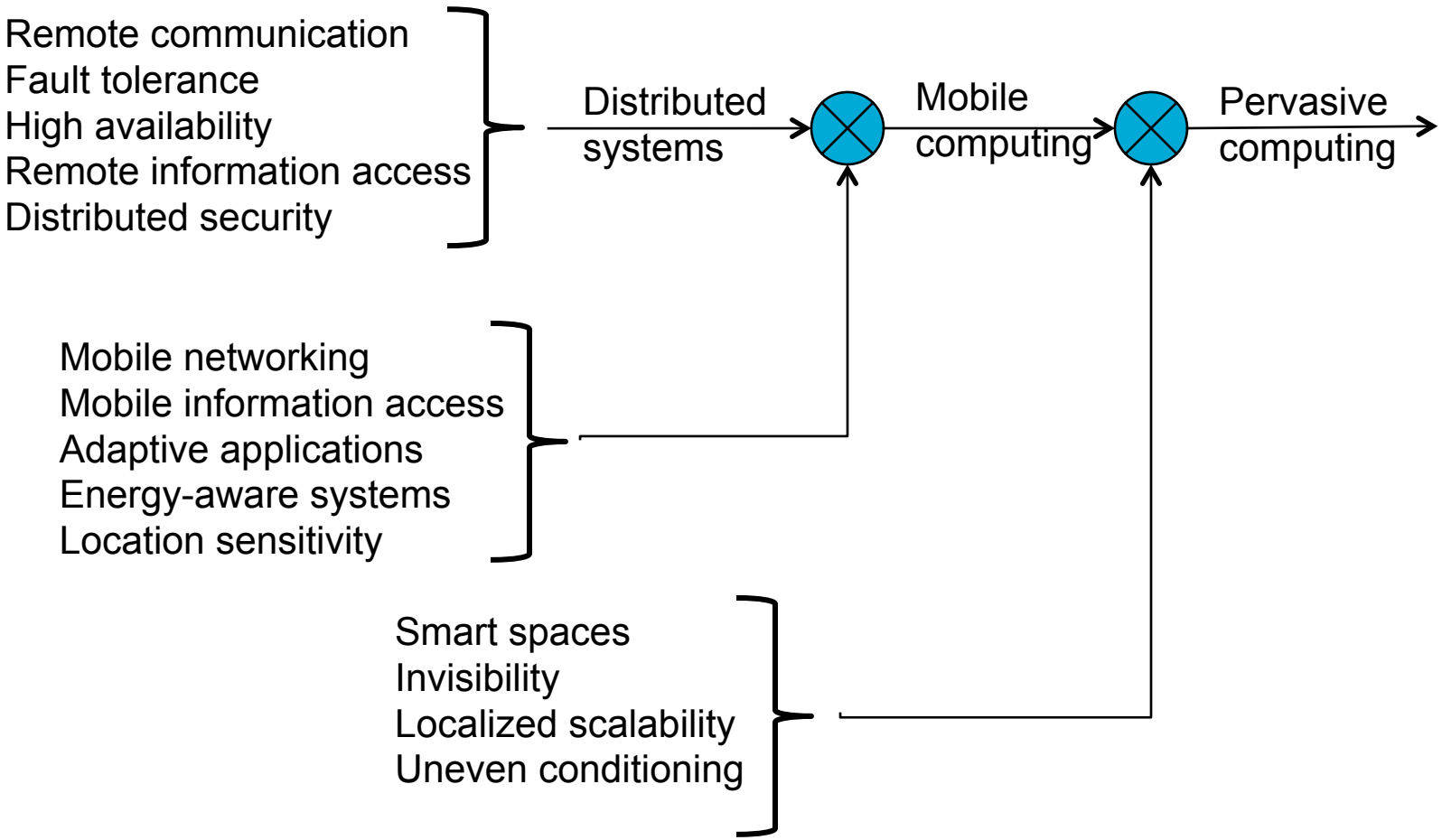
*M. Satyanarayanan, Pervasive computing: vision and challenges, IEEE Personal Communications, 2011

Problems and research fields



- Laptops and wireless LANS in early 1990s
- Known constraints that change the problem space
  - Unpredictable variation of network quality, << trust and robustness of mobile elements, limitations on local resources imposed by weight/size constrains, concerns about power usage

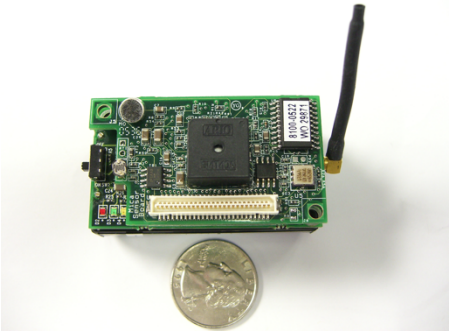
Problems and research fields



Mobile + pervasive: *volatile computing* - Rapid change is the norm rather than the exception

Smart spaces as an example

- Physical space provide the background to mobile and ubiquitous computing
- Smart space is a physical space with embedded devices (from a room to a train cart)
  - Algorithms (e.g., routing) and systems solution for more stable environments may fail
  - Devices have limited energy, typically are resource constrained and have a multitude of sensors
- Volatile connectivity and spontaneous interoperations
  - With the trust and privacy issues that come along



Association

- Volatile components need to interoperate
  - Network bootstrapping – the device must first get an address on the local network and may acquire/register a name
  - Association – once connected, components need to associate with services and/or provide services to others
- Network boostratpping
  - Some well-established solutions like DHCP, Apple's Bonjour
- Association
  - Considering scale – the number of devices and software components could be large, and
  - Scope – association with those in the smart space, no larger/ smaller
    - Boundary principle – smart spaces need to have system boundaries that correspond accurately to meaningful spaces

Discovery services for association

- Discovery service – a directory with services registered and looked up by attributes
  - Set of attributes against which queries are run may be determined at runtime
  - There may no be infrastructure to host a directory server
  - Registered services may spontaneously disappear
  - Protocols for accessing the discovery service must be careful of energy and bandwidth consumed
- Commonly, network discovery service – using a subnet for binding and the *boundary* issue
  - Space may expand beyond (WiFi reach) or be smaller than (hotel room) the subnet reach

Interoperation

- After association – how do components inter-operate?
  - From the communication protocol to the programming model
- Main difficulty – interface incompatibility
  - Allow heterogeneous interfaces and adapt interfaces
    - But semantic compatibilities are hard to overcome
  - Constraint interfaces to be identical across a class
    - Not as hard as a first take – consider Unix's pipes!
- Data-oriented programming for volatile systems
  - Event systems, tuple spaces, direct device interoperation

Sensing and context awareness

- Physical circumstances/context are relevant to determine volatile systems' behaviors
- To determine context – sensors
  - Location, velocity and orientation sensors
  - Ambient sensors
  - Presence sensors
- Working with sensors
  - Integrating idiosyncratic sensors
  - Abstracting sensors data (long/lat or the coffee shop?)
  - Combining sensors
  - Handling dynamics (has the user changed locations or is just an error on the sensor reading?)
  - Connecting multiple sensors – sensor networks

Security and privacy

- Trust in volatile systems with spontaneous interactions
  - Access to a printer, a screen or your heart rate monitor
- Mobile hardware simple “walks away”
- Devices may no have sufficient resources (including energy) for certain cryptographic approaches
- Some solutions out there
  - Out-of-band communication
  - Other side-channels – physical contact, infrared, audio, barcode/camera, …
- Privacy protection
  - Avoiding hard-wired identifiers (e.g., MAC or RIFD)
  - Mixing communication (statistically combining communication from many users)

Adaptation

- Wide heterogeneity of devices – consider producers and consumers of media content
- Adaptation of media data to the context of the consumer
  - Characteristics of the devices
  - … the task of the user
  - … user intent
  - … dynamic levels of energy and bandwidth
- Who should adapt the content and how?
  - Producer, consumer, proxies, …
- Ideally, adaptive software that accommodate to changing environments, resources, user demands
  - Cyber foraging and application composition

Summary

- Mobile and ubiquitous computing – a wide open field
  - Challenges of volatility
- Changing users, devices, services, connectivity, …
- Integration of devices with our physical world depends on sensing and context awareness
  - Many subtle complications
- Security, privacy and trust