Research Challenges in Project Aura

Distraction-free Ubiquitous Computing

M. Satyanarayanan

School of Computer Science

Carnegie Mellon University

Participants Span SCS

Duane Adams

Roger Dannenberg

David Garlan

Alex Hills

Takeo Kanade

Raj Reddy

Alex Rudnicky

M. Satyanarayanan

Jane Siegel

Dan Siewiorek

Peter Steenkiste

Alex Waibel

and many others . . . . .

Moore’s Law Reigns Supreme

Processor density

Processor speed

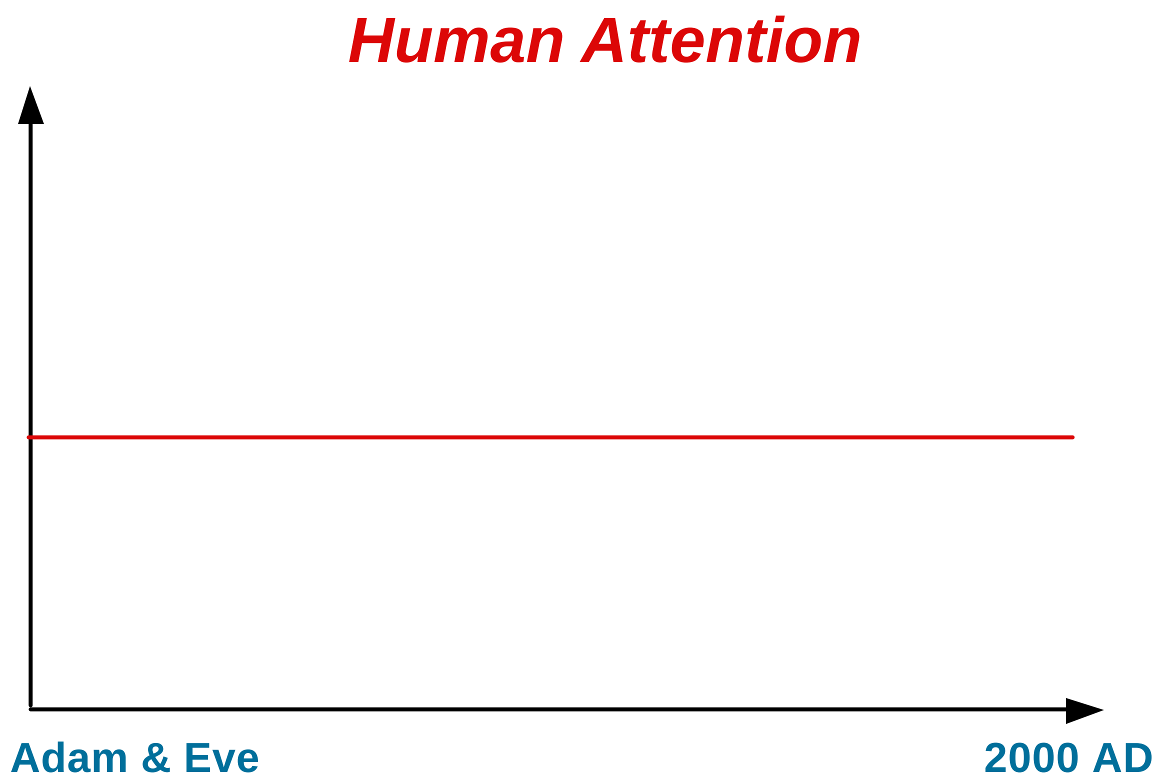
Memory capacity

Disk capacity

Memory cost

• • • • • •

Glaring Exception



Aura Thesis

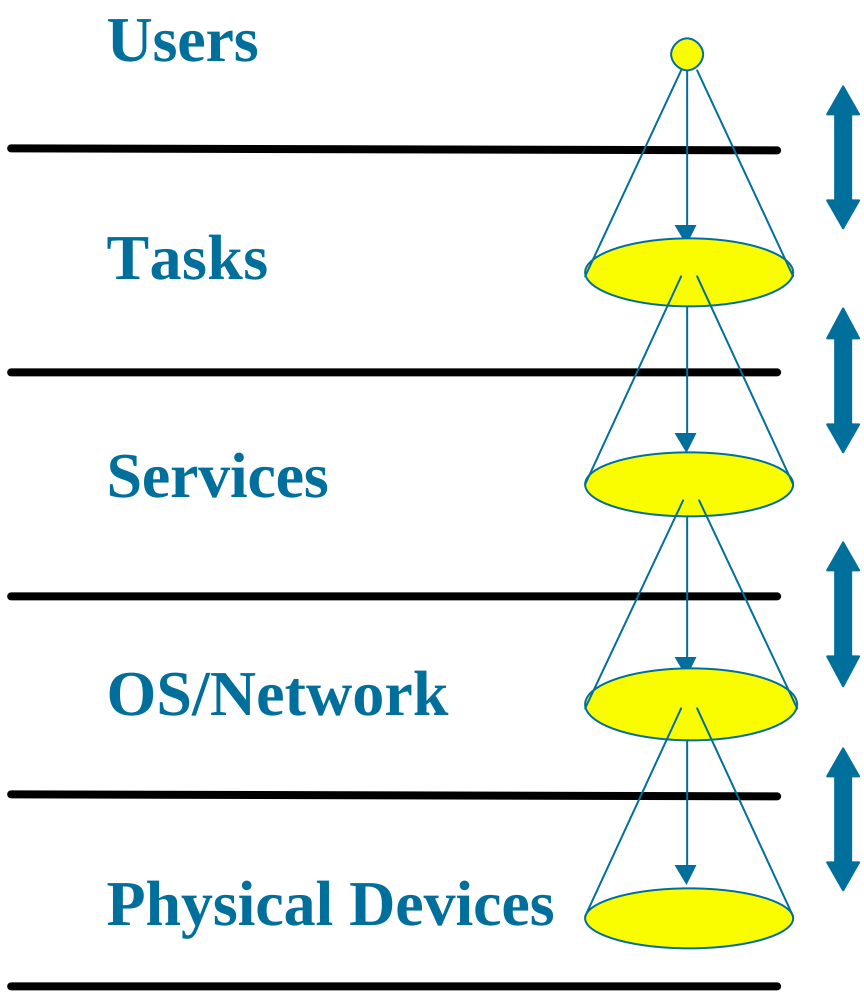
The most precious resource in computing

is human attention

Aura Goals

- **reduce user distraction**
- **trade-off plentiful resources of Moore's law for human attention**
- **achieve this *scalably* for *mobile users* in a**
  failure-prone, variable-resource environment

Aura Research Framework



Research Examples

HCI/Agents/Speech

Task-driven computing
Rapid service configuration

Energy-aware adaptation
Multi-fidelity computation
Nomadic data access

Intelligent networking
Power/bw-aware devices
Wearable computers

Technologies Being Explored

This Talk

Task-Driven Computing

Energy-Aware Adaptation

Multi-Fidelity Computation

Intelligent Networking

Resource Opportunism

Omitted for Brevity

Speech Recognition

Language Translation

Multimodal User Interfaces

User Interface Adaptability

Software Composition

Proxies/Agents

Collaboration

Robustness, Reliability

Rapid Failover

Security & Privacy

• • • • • • • • •

Task-Driven Computing

Problem

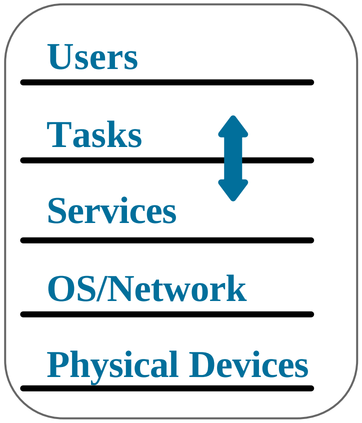
Humans interact at too low a level with computer

- **URLs, filenames, program names, etc.**
- **very explicit, step-by-step interactions**
- **like programming in machine language!**

Result

- **brittle behavior**
- **many details change with failures, platform changes, etc.**
- ***consume mobile user's attention***

Solution: Task-Driven Computing



Support user intentions

- capture high-level intent as ***tasks***
- raise level of abstraction of user interactions

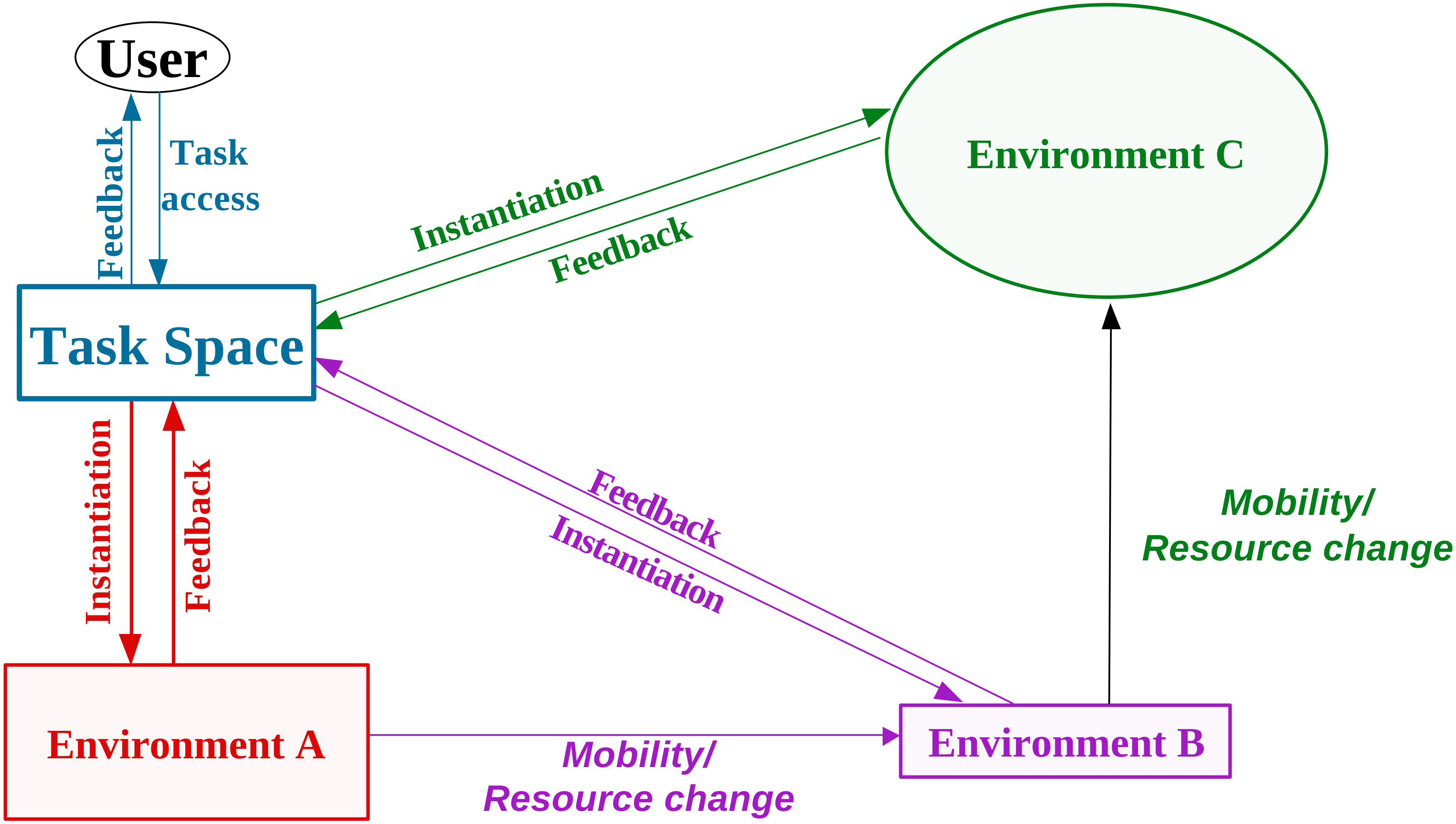
Support mobility

- suspend/resume on different platforms and locations
- dynamically reconfigure to match available resources

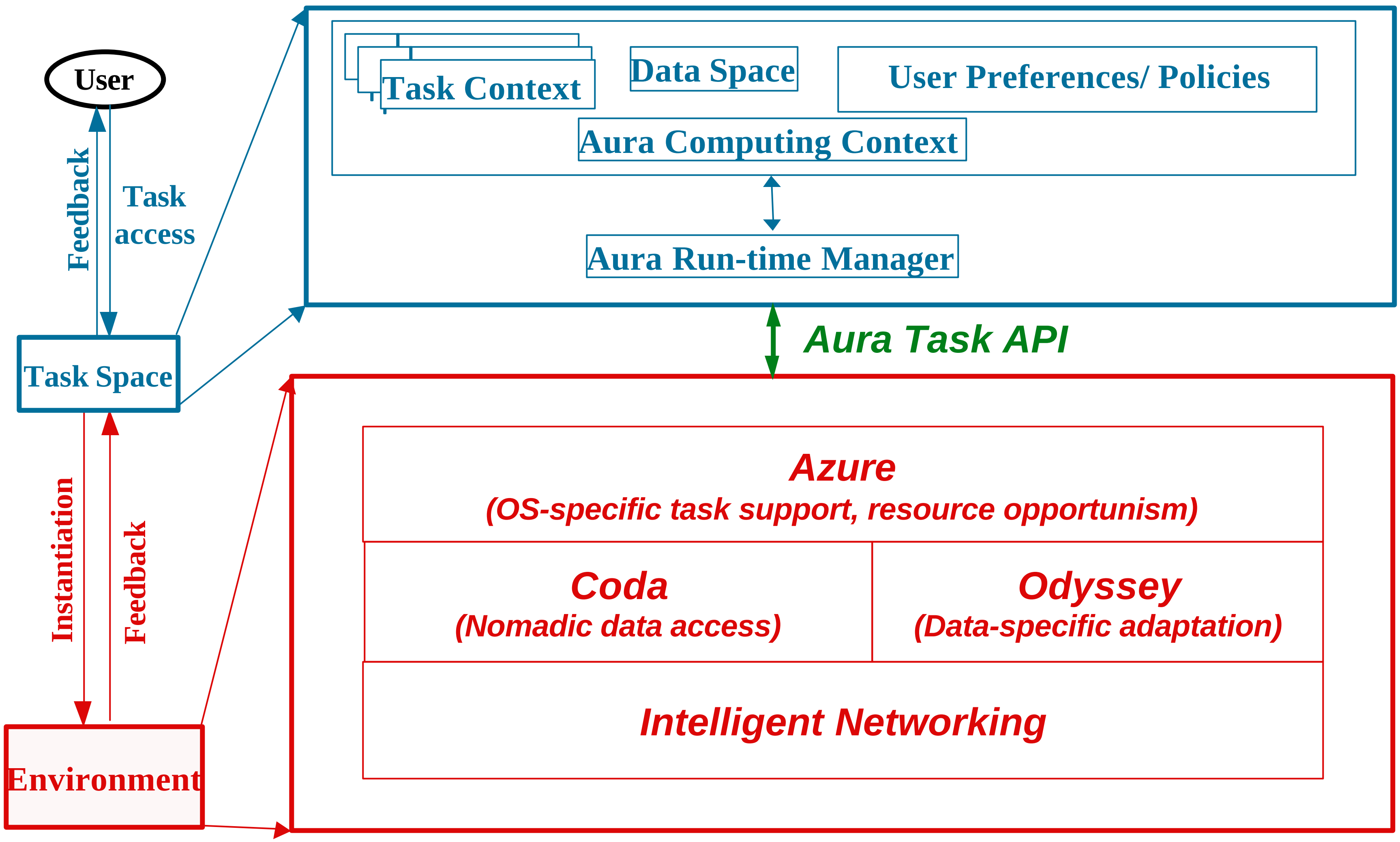
Support proactivity

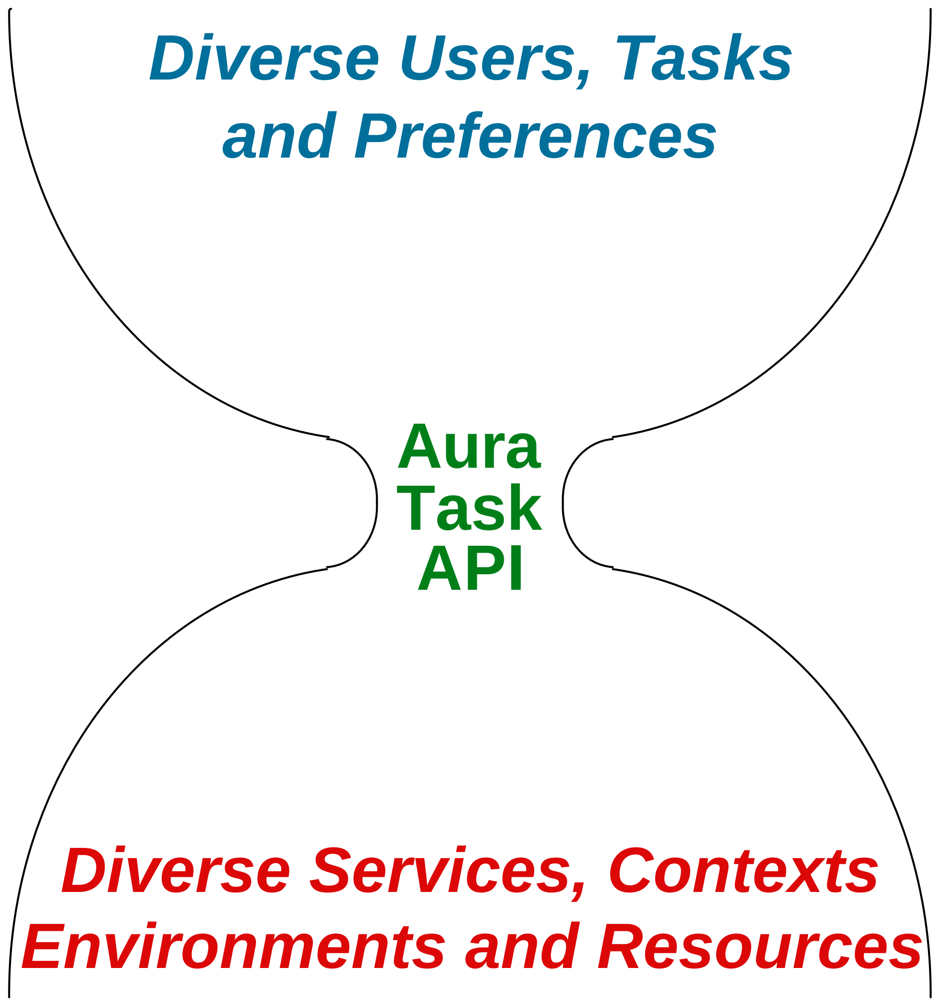
- active guidance from system
- corrections, alternatives, persistence

Adapting to Environment



Architecture




Diverse Users, Tasks and Preferences
Aura Task API
Diverse Services, Contexts Environments and Resources

Some Research Challenges

- Design of task definition language with rich expressive power
- Multi-modal interface for creating and modifying tasks
- Integration of task libraries and legacy workflow tools
- Mechanisms for tracking, suspending and restoring task state
- Design of platform-independent Task API
- Platform-specific implementations of Task API
- Effective exploitation of mechanisms like Coda & Odyssey
- Triggering mechanisms for pro-activity

Energy-Aware Adaptation

Problem

Battery power is a critical resource when mobile

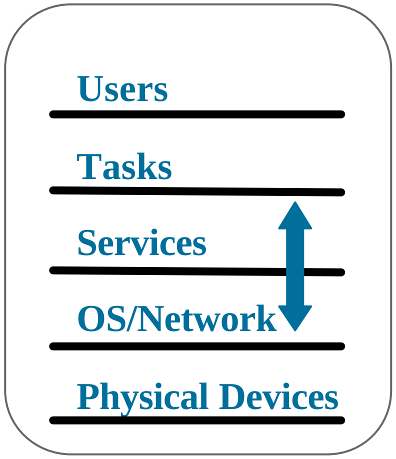
Current approaches only offer limited extensions of mission life

- **no dramatic battery improvements foreseen**
- **low-power hardware is important, but not enough**
- **wireless transmission is a major consumer of energy**

Explicit user management of energy consumes attention

Solution: Energy-Aware Adaptation

Applications change behavior as battery drains



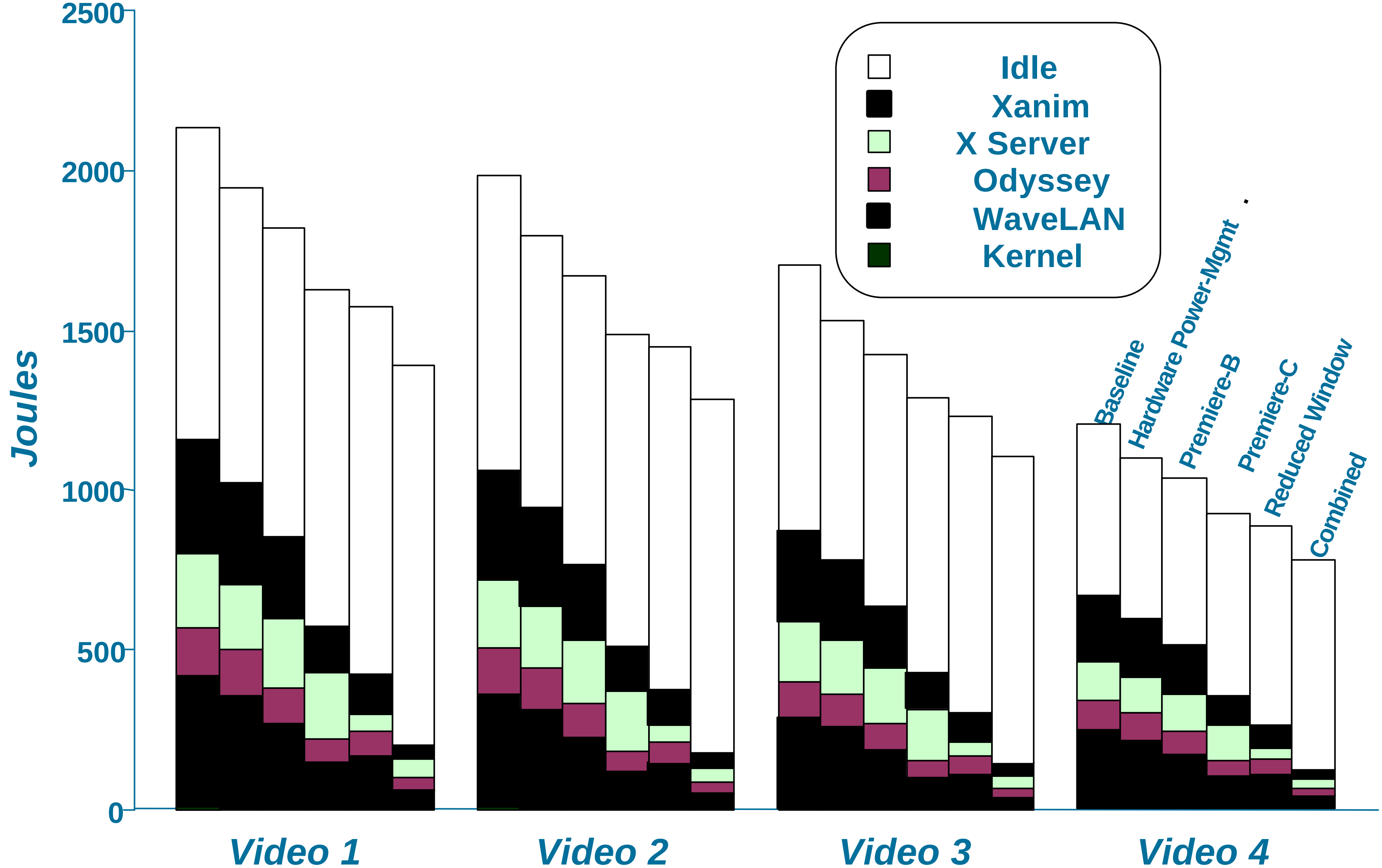
Collaborative relationship between OS & apps

- **OS monitors energy supply & demand**
- **notifies apps when to change fidelity**

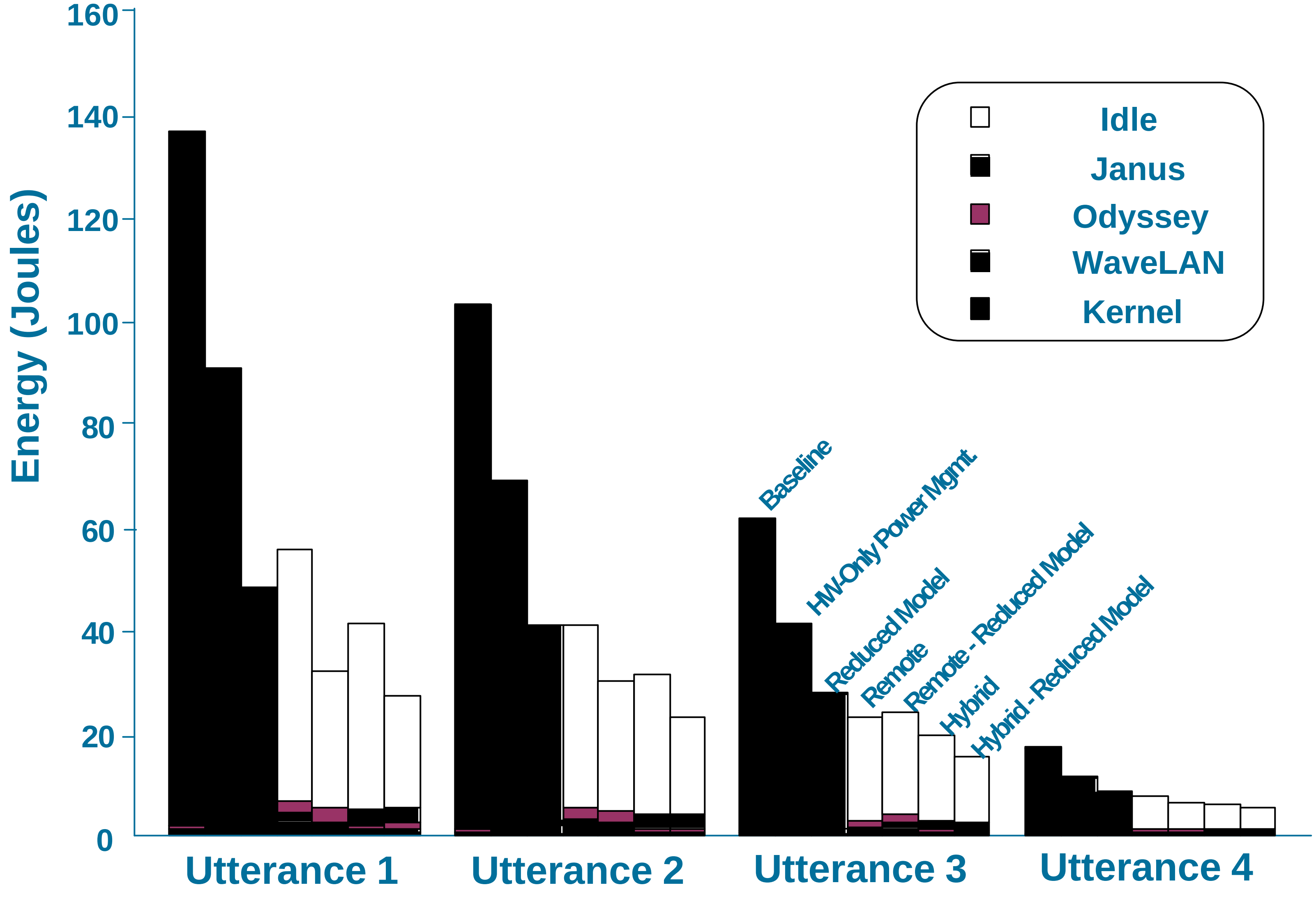
Extension: *goal-directed adaptation*

- **user provides time estimate**
- **OS ensures goal is met**
- **mid-course corrections accommodated**

Energy Impact: Video



Energy Impact: Speech



Some Research Challenges

- Validate approach for broader set of applications
- Use compact instrumentation for mobility
- Exploit emerging standards such as ACPI and Smart Battery Spec
- Design user interface that balances control with transparency
- Exploit history for agile adaptation
- Energy-sensitive remote execution mechanism
- Coupling of energy considerations to task layer

Multi-Fidelity Computation

Problem

Good performance on interactive mobile apps very difficult

- **resource-poor hardware**
  (size, weight, energy constraints)
- **demanding apps (e.g. augmented reality)**
- **wireless energy drain if computing shipped off-site**
- **serious user discomfort & distraction with poor performance**

Catch-22!

Solution: Multi-Fidelity Computation

Fundamentally rethink our model of computing

Classic notion of algorithm

- **fixed correctness criteria**
- **variable amount of resources consumed to meet this**

Adaptation for mobility suggests a different viewpoint

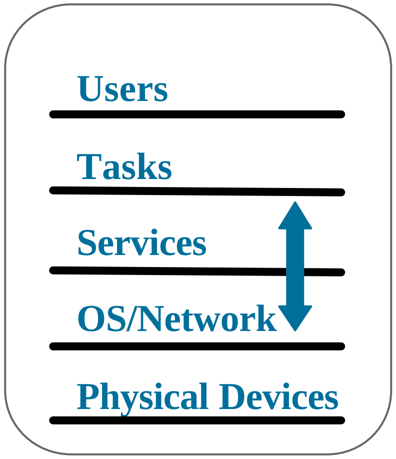
- ***"Do the best you can using no more than X units of resource"***
- **correctness criteria no longer fixed**
- **multiple notions of "correct"; each is a level of fidelity**

Why is this Concept Useful?

Resource consumption can now be independent variable

Special case: let system find *"sweet spot"*

- ***"Give the best result you can cheaply"***
- **knee in fidelity-resource usage curve**



Interactive mobile applications fit this model well

- **users are tolerant of imperfect results, if so labeled**
- **"sharp cliffs" in performance imply big wins**

Example: Architect's Aid

Augmented reality for rapid preview of design changes

- wearable computer with heads-up display
- used on-site for remodeling projects

Architect and customer can explore alternatives together

- initial "quick & dirty" rendering using local computation
- many iterations to shrink design space
- when a design looks promising, request high fidelity

Rendering algorithms span wide range of cost & fidelities

Example: On-Site Logistics Aid

On-site engineer solving unexpected problem

- handheld machine with spreadsheet-like tool
- wireless link to compute server

Many “quick and dirty” calculations

- done locally, at low fidelity
- results displayed in distinctive font or color

Full fidelity when promising solution identified

- include all design checks
- ship to remote compute server, close to databases

Concept of fidelity natural to numerical computation

- terms in series expansions, mesh coarseness, iteration count, ...

Some Research Challenges

- Validation in real mobile applications
- Design of API for multi-fidelity computation
- History-based resource-estimation mechanisms
- Integration with cache management
- Dynamic balance of tradeoffs across different resources
- Sweet spot discovery

Intelligent Networking

Problem

Today's systems assume network is dumb

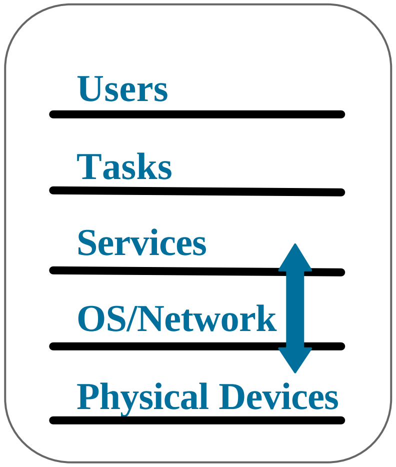
- **very restricted interfaces**
- **mismatch between network QoS and user-perceived QoS**
- **cannot take advantage of active networks**

Hard to incorporate network-triggered pro-activity

- **more generally, environment-triggered pro-activity**

Solution: Expressive System APIs

Extend APIs to allow rich two-way flow of QoS information



Strategy

- condition environment if possible
- resort to client adaptation otherwise
- network "traffic report" on multiple time scales
- "network weather map" for proactive feedback
- extend to non-network aspects of environment

Impact

- improve perceived quality of network service
- support mix of dumb & intelligent networks
- enable "intelligent workspaces" to be exploited

Some Research Challenges

- Design of API extensions that are flexible yet not “kitchen sink”
- Mechanisms to exploit active networks
- Dynamic integration of Aura clients into smart workspaces
- Support for wireless “network weather service”
- Algorithms for pro-active guidance in wireless environments

Resource Opportunism

Problem

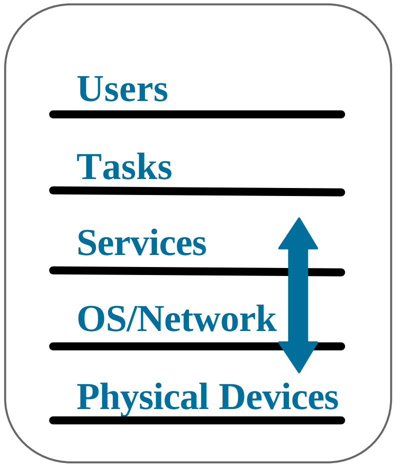
Mobile hardware optimized for weight, size and battery life

- reduced compute power relative to desktops & servers
- client often forced to low-fidelity behavior

Mobile users often forced to sacrifice creature comforts

Solution: Resource Opportunism

Exploit non-mobile hardware in local environment

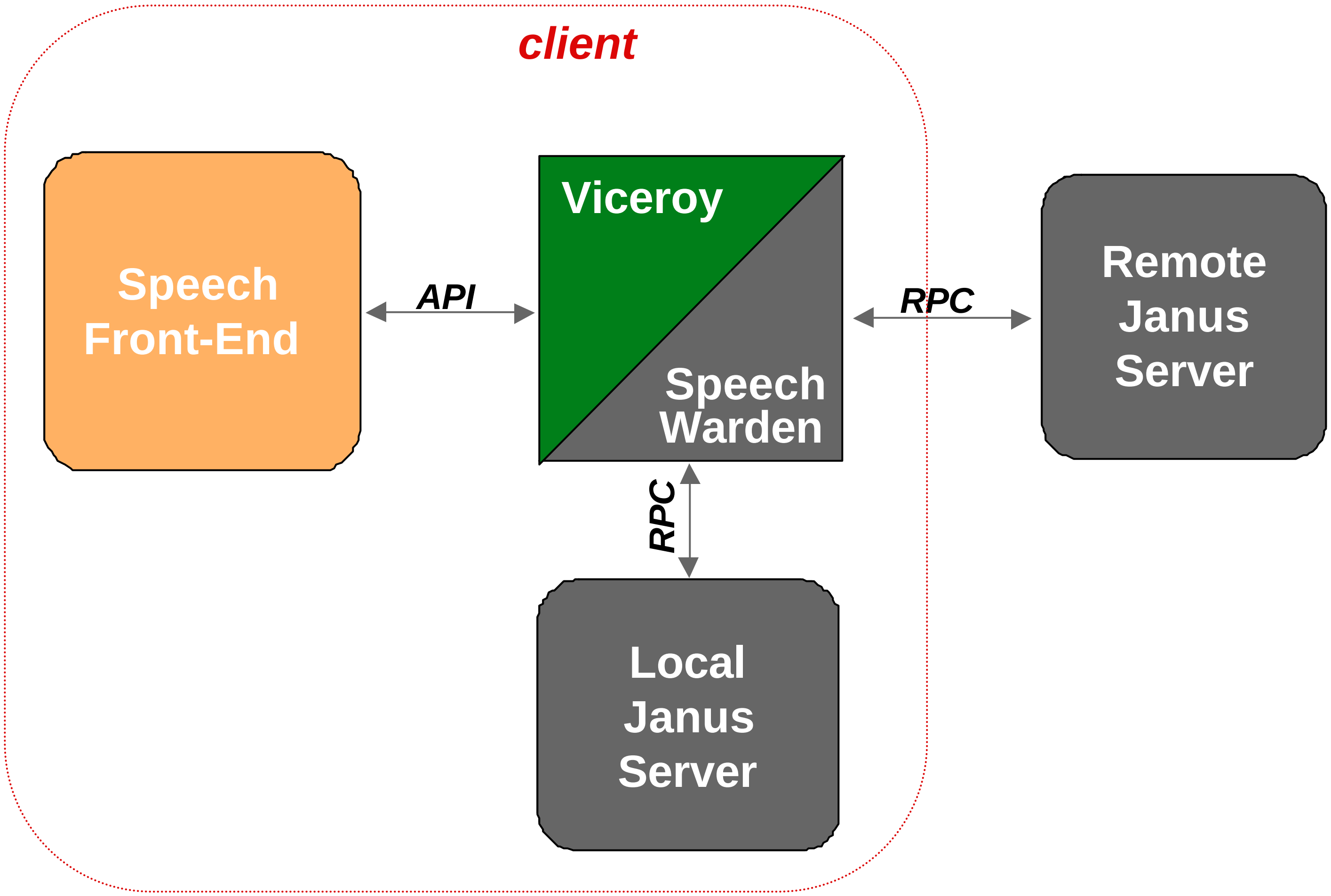


Dynamically detect presence of useful resources

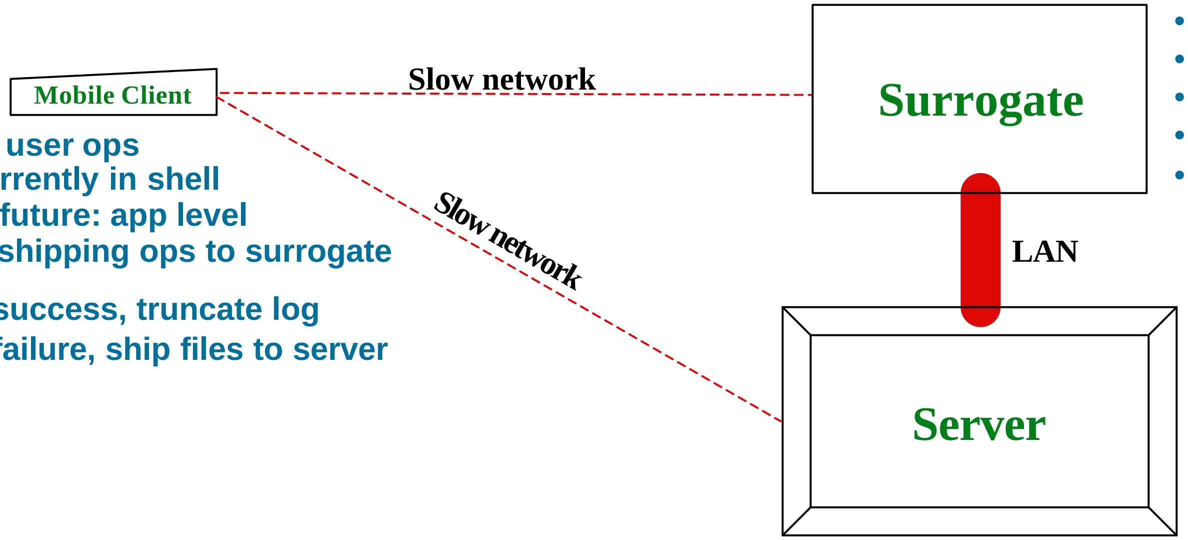
- **ship compute-intensive operations to intermediary**
- **use intermediaries to stage cache data**
- **smooth, seamless & transparent to user**

But retain ability to function without external resources

Odyssey: Speech Recognizer



Coda: Operation Shipping



- Log user ops
  - currently in shell
  - in future: app level
- Try shipping ops to surrogate
- On success, truncate log
- On failure, ship files to server

- Re-create client environment
- Re-execute user ops
- Validate with fingerprints
- On success, ship files to server
- On failure, return error to client

Typical Performance Benefit

Data volume reduction: 12X to 245X

Elapsed time improvement:

0.8X to 5X at 64Kb/s

3.4X to 26X at 9.6Kb/s

Coping with minor re-execution glitches

Differences in temp file names
- side effect of apps like ar
- handled through rename optimization

Timestamps in output files
- side effect of apps like latex, dvips, etc.
- treated like transmission errors
- corrected using Reed-Solomon FEC

Some Research Challenges

- Resource discovery, including predictive ability
- Preserving security in foreign environments
- Design of API for resource opportunism
- Fault-tolerance mechanisms to cope with disconnection
- Rapid re-creation of execution environments

Closing Thoughts

Aura as an Expedition

Well-defined goal: ***Conserve human attention through Moore's Law***

But getting there will be an adventure!

And a lot of valuable research will happen along the way