Middleware, Debuggging, Virtualization

Larry Rudolph
MIT 6.883
May 10, 2007

Massachusetts Institute of Technology

1

Publish & Subscribe

Larry Rudolph
May 3, 2006
SMA 5508 & MIT 6.883

Massachusetts Institute of Technology

2

Agents

- **An agent is an autonomous program.**
  - It executes code and can communicate with other agents.
- **All the components in a pervasive computing application (whatever that is) usually called agents**
  - An agent may be a "proxy" for a device
  - Devices, like camera or keyboards, are controlled by some proxy agent
- **Agents may appear or disappear at any time**
  - There is some issue in how to start them
  - There can be problems when they crash
    - *there may be replicates*

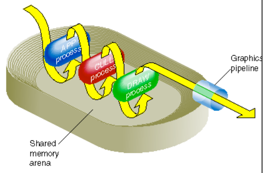
A collection of agents

- **Parallel or distributed programming**
  - a bunch of communicating agents working to solve a problem
  - faster
    - *two heads better than one*
  - geographically distributed
    - *everyone can't live together*

Agent communication

GEORGE W. BUSH ENGAGED IN THE AWESOMELY IMPORTANT BATTLE AGAINST TERRORISM.
MR.FISH

- **Two main choices:**
  - (which was best used to be "religious battle")
- **Shared memory (SM)**
  - agents load and store values
    - *start with a set of numbers*
    - *remove two numbers, insert their sum*
    - *done when only one value remains*
  - issues: synchronization, locks, etc.
- **Message-passing (MP)**

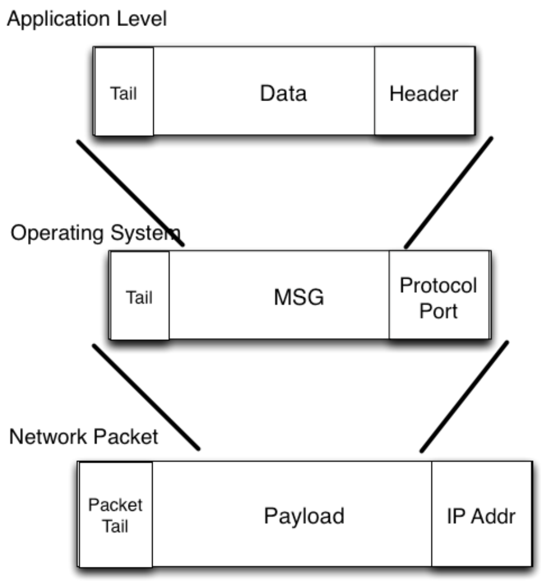


Agent communication

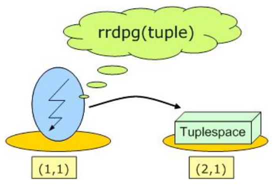
- **Message-passing (destination, data)**
  - Agent Bob: Send(Alice, "Do you want to go out?")
  - Agent Alice: Recv(from,msg)
    - *from = Bob; msg = "do you want to go out?"*
    - *send(Bob, "No")*
- **Issues:**
  - Sender must know destination, recv need not
  - blocking or non-blocking
  - low performance, lots of copying of data
- **MP can implement SM and vica-versa**
  - MP on clusters, SM on multiprocessors

Message Passing via Sockets

- **Sockets are general**
- **Application can specify**
  - port
  - protocol
  - other attributes
- **Message-Passing**
  - library does all the specification
  - may reformat data

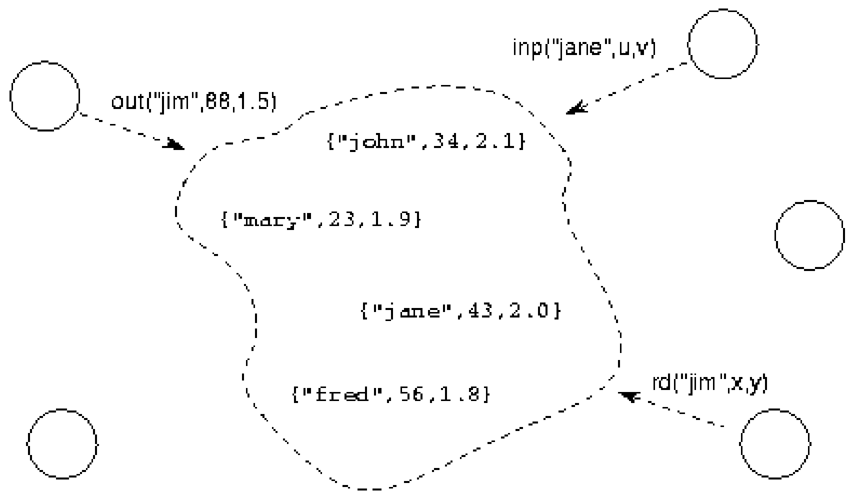


Tuple-space



- **A third communication mechanism!**
  - formed basis of Linda programming language
  - tuple: ordered collection of typed elements
- **Basic Operations**
  - **out**: inserts a tuple, whose fields are either
    - **actual**: *a static value* **formal**: *a program variable*
  - **in**: extracts tuple, argument is template to match
    - *actuals match fields of equal type and value*
    - *formals match fields of same type*
  - **rd**: same as in, but does not remove matched tuple

Tuple-space example



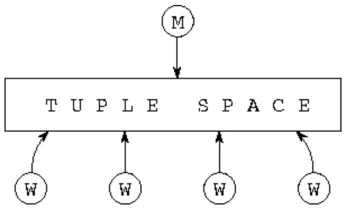
Linda programming example

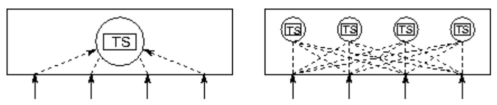
```
procedure manager
begin
  count = 0
  until end-of-file do
    read datum from file
    OUT("datum",datum)
    count = count+1
  enddo
  best = 0.0
  for i = 1 to count
    IN("score",value)
    if value > best then best = value
  endfor
  for i = 1 to numworkers
    OUT("datum","stop")
  endfor
end
```

```
procedure worker
begin
  IN("datum",datum)
  until datum = "stop" do
    value = compare(datum,target)
    OUT("score",value)
    IN("datum",datum)
  enddo
end
```

What is the big deal?

- **Virtual shared memory**
  - tuples with [address,value]
  - stores are inserts, loads are non-destructive reads
- **Virtual message passing**
  - tuples with [dest, data]
  - recv are destructive reads
- **Even more, when matching on multiple fields**
- **Allows many types of implementations**





Agent Interaction Choices

- **Direct communication model**
  - Jini
  - FIPA
- **Indirect, Shared Data-space models**
  - EventHeap (centalized)
  - MARS (fully distributed)
- **Event-based publish/subscribe models**
  - Siena
  - Jini Distributed Events
  - Selective subscription

Stanford's Event Heap

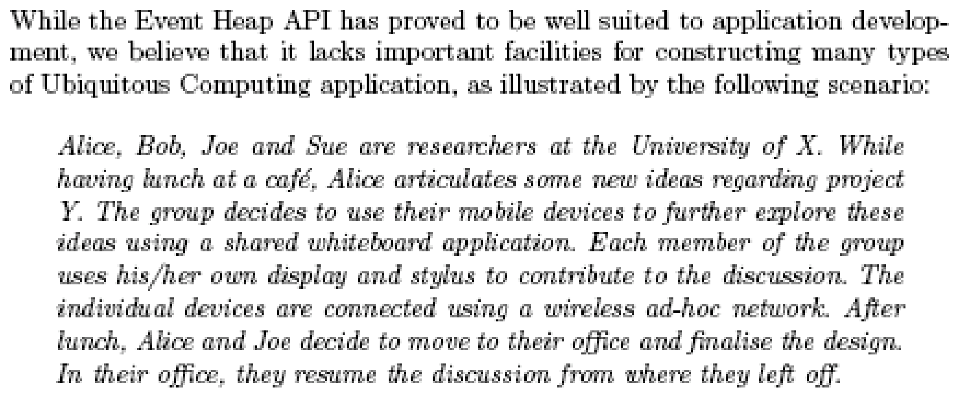
- Based on Tuple Space paradigm
  - tuple: arbitrary mix of typed fields
  - mechanism for passing data & events
- Extensions make it useful for agents
  - many projects exist based on different extensions

Event Heap Extensions

- **Extended Delivery Semantics:**
  - Per-source ordering, always see events in order they are generated by the source
  - Total order: if tuple space is centralized, get this even if multiple sources
- **Persistent Queries:**
  - non-destructive read of those matching
  - also matches tuples inserted in future
- **Event Notification:**
  - like PQ, get notified of future matches
  - at most once semantics

CSAIL

Need more than simple event heap



While the Event Heap API has proved to be well suited to application development, we believe that it lacks important facilities for constructing many types of Ubiquitous Computing application, as illustrated by the following scenario:

> *Alice, Bob, Joe and Sue are researchers at the University of X. While having lunch at a café, Alice articulates some new ideas regarding project Y. The group decides to use their mobile devices to further explore these ideas using a shared whiteboard application. Each member of the group uses his/her own display and stylus to contribute to the discussion. The individual devices are connected using a wireless ad-hoc network. After lunch, Alice and Joe decide to move to their office and finalise the design. In their office, they resume the discussion from where they left off.*

Suggested additions

- Need “distributed, replicated or federated local instances
  - (from paper by Storz, Friday, & Davies)
- Multiple event heap instances -- but not easy of implement
  - View: processes that share a view have consistent ordering

More general issues

- **Lots and lots of middleware systems**
  - no winner (may never happen)
- **What gets communicated?**
  - services, events, XML records
- **The shared space is often a: BROKER**
  - The broker stores the tuples and does the matching

CSAIL

Big Issues

- **Naming**
  - This is a big, big deal.
  - e.g. how do you name a camera:
    - *model brand, IP, DNS name, location, virtual space*
    - *via attributes (color, 740x1024), ownership?*
    - *Is there only one name for the agent?*
- **Matching**
  - A big deal
    - *Which attributes explicit, which implicit*
    - *Where to do the lookup?*

CSAIL

Issues

- **Addition information provided by broker**
  - for services: how to interface them
  - filtering events
  - higher level events implemented at broker
    - *based on multiple basic events*
- **Adaptivity**
  - When to discard services, events
    - *keep alive, heartbeats*
  - Invoke new instance of service automatically
  - Fault tolerance

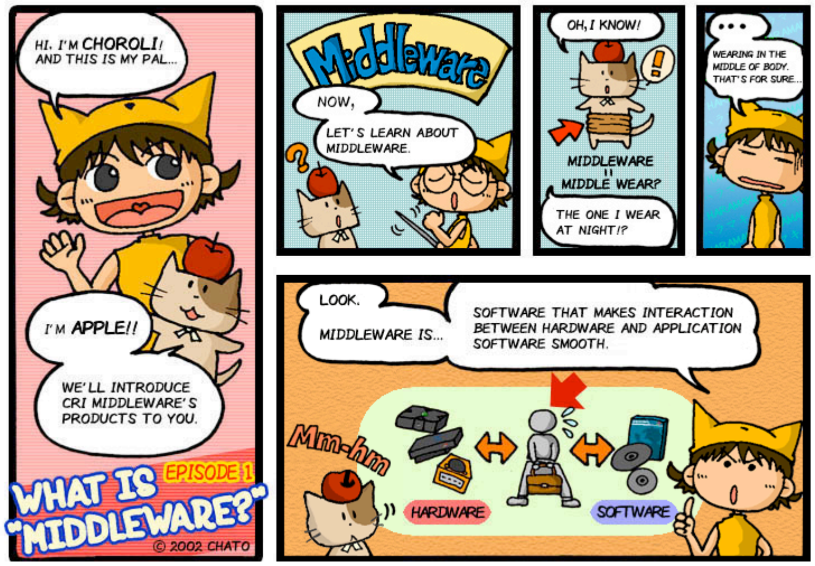
Issues

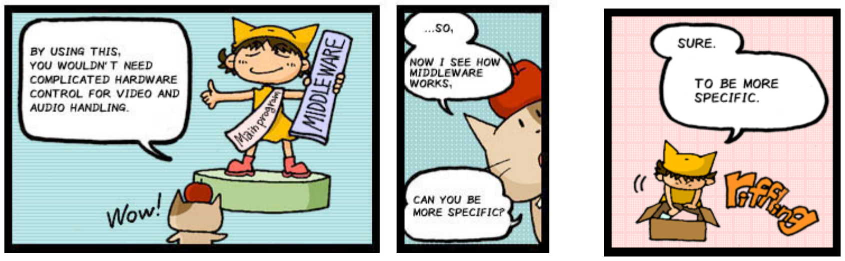
- **Standards**
  - XML, SOAP, WSDL
  - Proprietary Interfaces
- **Middleware may be new Operating System**
  - Whoever controls it will dominate
  - Not clear if there is or will be a winner
- **Integration with web-services**
  - Lightweight devices are different
  - May want stateful communication

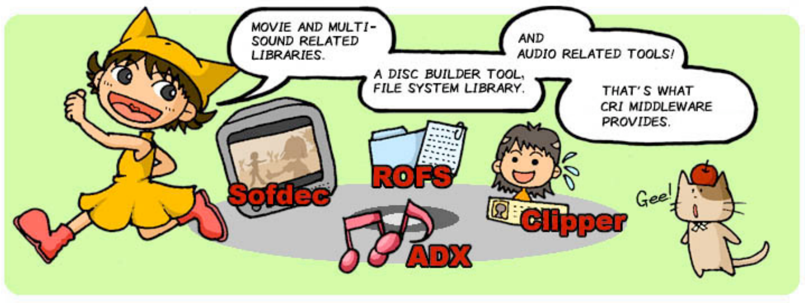
Discover the World of Interactive Television
with
IRT's iTV-Middleware
mhp
Institut für Rundfunktechnik IRT

Tell me about Middleware web comic EPISODE 1 WHAT IS MIDDLEWARE?

Page 1 2 3 4


HI, I'M CHOROLI! AND THIS IS MY PAL...
I'M APPLE!!
WE'LL INTRODUCE CRI MIDDLEWARE'S PRODUCTS TO YOU.
WHAT IS "MIDDLEWARE?" EPISODE 1
© 2002 CHATO
Middleware
NOW,
LET'S LEARN ABOUT MIDDLEWARE.
OH, I KNOW!
MIDDLEWARE
MIDDLE WEAR?
THE ONE I WEAR AT NIGHT!?
...
WEARING IN THE MIDDLE OF BODY. THAT'S FOR SURE...
LOOK.
MIDDLEWARE IS...
SOFTWARE THAT MAKES INTERACTION BETWEEN HARDWARE AND APPLICATION SOFTWARE SMOOTH.
Mm-hm
HARDWARE
SOFTWARE

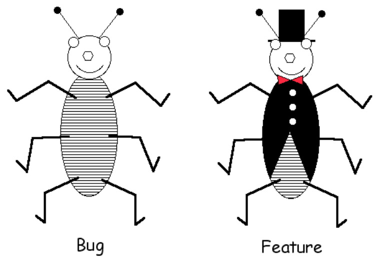

BY USING THIS, YOU WOULDN'T NEED COMPLICATED HARDWARE CONTROL FOR VIDEO AND AUDIO HANDLING.
Main program
MIDDLEWARE
Wow!
...SO,
NOW I SEE HOW MIDDLEWARE WORKS,
CAN YOU BE MORE SPECIFIC?
SURE.
TO BE MORE SPECIFIC.
Riffling


MOVIE AND MULTI-SOUND RELATED LIBRARIES.
A DISC BUILDER TOOL, FILE SYSTEM LIBRARY.
AND AUDIO RELATED TOOLS!
THAT'S WHAT CRI MIDDLEWARE PROVIDES.
Sofdec
ROFS
ADX
Clipper
Gee!

I GOT IT!
THESE TOOLS AND LIBRARIES HELP DEVELOP SOFTWARE SMOOTHLY IN A SHORT TIME.
BUT, ...
DON'T YOU NEED TO SWITCH BACK AND FORTH MIDDLEWARE OR LIBRARIES DEPENDING ON TARGET PLATFORM?
Hassle..
PS2
Xbox
GC
grin
GOOD POINT, MR.APPLE.
CRI MIDDLEWARE SUPPORTS ALL EXISTING MAIN CONSOLES.
SUCH AS PS2, GAMECUBE AND XBOX.
Common API
(API … APPLICATION PROGRAM INTERFACE)
IT TAKES CARE OF ADDITIONAL WORKS OTHER THAN GAME APPLICATION DEVELOPMENT!
EASY TO PORT, TOO!
NOT ONLY CONVENIENT!
HEAP!
THE NUMBER OF GAME TITLES USING THIS MIDDLEWARE IS MORE THAN 550!
IT'S SO HEARTY JUST LIKE MY "MIDDLE WEAR"!
SAYS YOU.

Debugging Applications in Pervasive Computing

Turning bugs into features



- Speech recognition is not 100%
  - Who likes it when its wrong?
  - Children
    - Example: story telling (easy reading)
    - Computer recognizes the words that the child is reading and animates it

Voice controlled animation

- **A very fertile domain: room for improvement**
  - mouse is very limited
    - *hard to specify parameters*
      - **choose from list -- awkward when long**
      - **one action and one parameter**
  - speech allows multiple parameters (and sub-parameters)
    - *objects are parameters; adjectives are params of params*
- **Unfortunately, no good models of children's voices**
  - so we have to act like children :)

Testbed for other ideas

- **Naming**
  - give basic object a name
  - give composite object a name (macro)
  - many parameters come from context (environment)
  - differentiate between base object and instantiated object

CSAIL

Controlling Errors

- **Two types of consequences to errors:**
  - something useful (or interesting)
  - something destructive (or boring)
- **Who gets to decide?**
  - tolerating some errors --> flexibility
  - avoiding all errors --> too rigid

CSAIL

Semantics

- Where does the semantics get checked?
  - no consensus (speech, vision, sketch)

Massachusetts Institute of Technology

Co-animation Approach

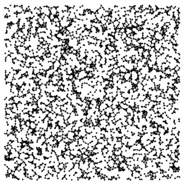
- **Command: action and parameters**
  - error: incompatible action and param
    - *dogs: sit, run, lick, beg, bark*
    - *cats: sit, run, lick, sleep, purr*
  - Consider the error: “dog purr”
    - *if cat is on stage, it purrs*
    - *if dog is on stage, do random action*
    - *random actor does random action*

CSAIL

Considerations

- **Really depends on the cost of error**
  - can action be "undone" easily?
  - is the user getting frustrated?
- **Rather than selecting at random**
  - choose the most likely action

For a civil engineer, there's no such thing as a "little mistake."



CSAIL

Informing the user

- **System consisted of lots of components on lots of machines**
  - flash (XP), galaxy (Linux), audio (iPaq)
  - how to find out about serious errors?
    - *cannot inform user; no output dev*
    - *not clear if other apps will forward*

CSAIL

Some Challenges of “traditional” debugging approaches

Stop/Inspect/Go

- **Stepping through the code (e.g. gdb)**
  - break; inspect memory & data structures
  - hard to get program to stop or break at correct point
- **Run backwards**
  - problem usually occurs just before death, so backup and check data-structures
  - Many ops are reversible (x = x + 1   x = x - 1)
  - push on stack control flow and non-reverse ops

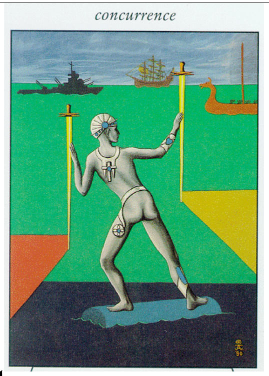
Stop/Inspect/Go

- **Logs**
  - Log all interesting events (I/O ?)
  - Need way to organize independent logs
  - Need way to see paths in the forest
    - *visualization tools are helpful*
    - *extensive log event tags*
- **Log control-flow history**
  - off-line playback or re-execution

Risk of Masking Bugs

- **Shared Memory (lots of experience)**
  - Many things look like share memory
    - *automatic synchronization; caching; distributed FS*
  - Low-level bugs due to strange timing bugs
    - *set flag; check flag; do operation*
- **Programmers think everything executes at same rate**
  - weird bugs when on process executes a little, pauses, executes a little more, pauses, etc.

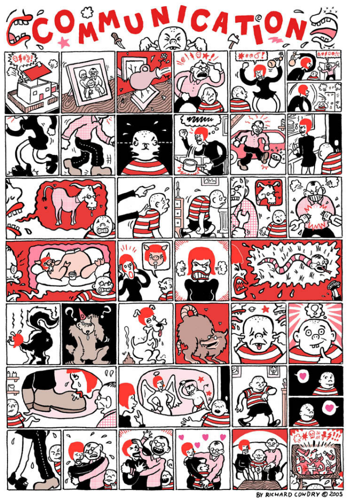
Concurrency



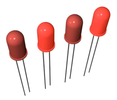
- **Debuggers don't deal well with threads.**
- **Conditional Breakpoints:**
  - Break when phone locks DB & camera locks mic
- **Need deterministic replay**
- **Need to understand all possible parallel executions**
  - race-condition detector
- **Software Transactions (memory & data-base)**
  - hand time-outs
  - heart-beat messages

CSAIL

Distributed Communication

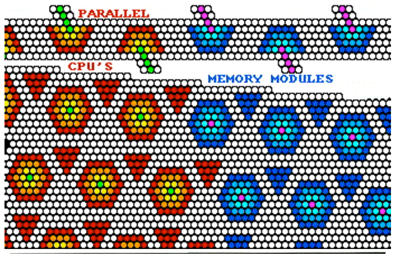


- **Central way to control system-wide parameters**
- **duplicate message detection; non-idempotent operations**
- **unified interface to debuggers on different systems & OS's**
  - start up; switch between debuggers
- **Distributed LEDs (one per process)**



CSAIL

Virtual Computer



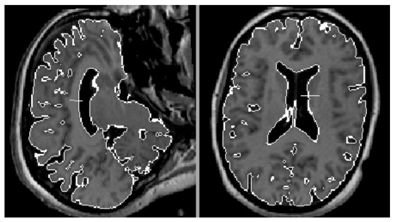
- **Start with a set of**
  - Emulators & Virtual Computers
- **Add**
  - Scheduler (various orderings)
  - Fault-Injection
  - Instrumentation
- **Debug under idealized world**
  - then move to real world

CSAIL

Yet another approach

CSAIL Massachusetts Institute of Technology

Change-point detections

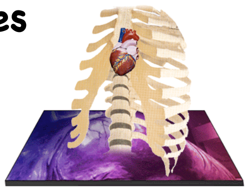


- **What do you do when things stop working?**
  - Seek out a friend. Their first question: "What did you change?"
  - Your first response: "Nothing"
  - Your second response: "Oh yea, thanks"
- **Too hard with pervasive computing environment**

CSAIL

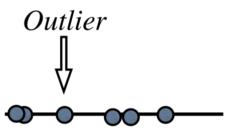
How to support this?

- **Too hard at the moment to automatically fix all problems.**
- **Worthwhile to point out potential sources**
- **Monitor everything, learn what's typical**
  - report what is atypical
  - monitoring must be on-line and cheap
- **Use human-level timing**
  - sec, min, hour, day, week, month, year



CSAIL

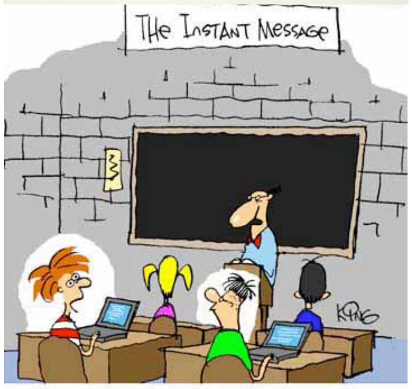
Isn't this like data-mining?



- **Data mining for failure indicators?**
  - No long log files; no labeled data
  - On-line and easier
- **Finding outliers is expensive**
- **Finding what recently changed is cheap**

Use out-of-band communication

- **If main application has problems**
  - error messages may not get forwarded
  - normal channels of communication might be the source of difficulties
  - want separate communication channel
- **Use IM & SMS for query**
  - ubiquitous, natural, usually works



"How did they get through boring lectures back in the days of paper? I can IM and it looks just like note-taking!"

Wrapping up

- **My conclusion is that**
  - physical world poses new challenges
  - user's must help in fixing problems
  - system must help the user in this task
  - we've only just begun ...

Virtualization

Helping to make Pervasive Computing Pervasive

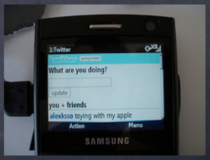
Larry Rudolph
MIT / CSAIL

Pervasive Computing

- Intelligent Environments
- Mobile Computing
- Natural Interfaces and Interactions with
  - digital media
  - other humans

The Good News: Success!

- Lots of interesting projects in
  - pervasive; ubiquitous; mobile computing
- Most successful ones
  - use special or uniform hardware
  - user-to-user via remote, third party server and simple data (text, picture, location, ...)



More than a curiosity?

- Most projects remain "in the lab"
  - system built
  - evaluated
  - published
  - forgotten

Example: In Education

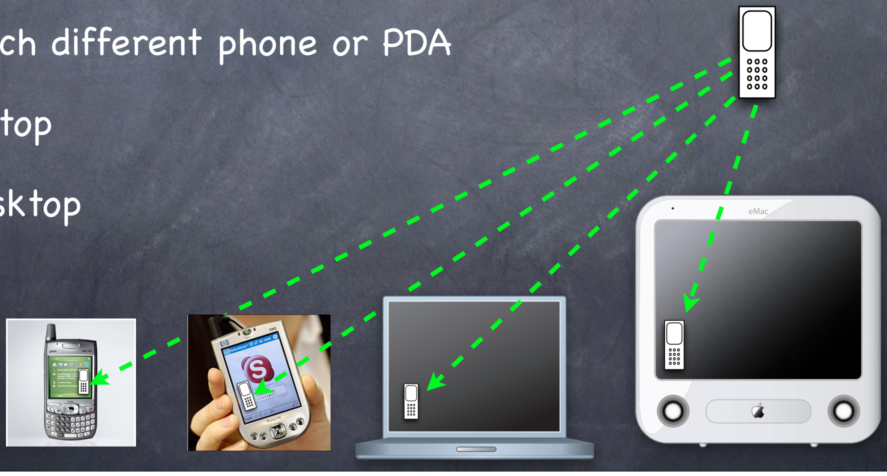
- Student has trouble with homework and needs a little help:
  - Social networking formulation: Friend of a friend, who took same course with same teacher and got a good grade
    - phone records classes attended, problem sets completed
    - when you walk around; phones silently interact and tell you if someone nearby can help
  - but its 3 AM -- is there someone who can help?
    - broadcast help request; after one answer delete others
- Require mobile phone integrated w/ PC to monitors (profile) everything
  - Nothing more than an experiment.

Solution: Virtualize

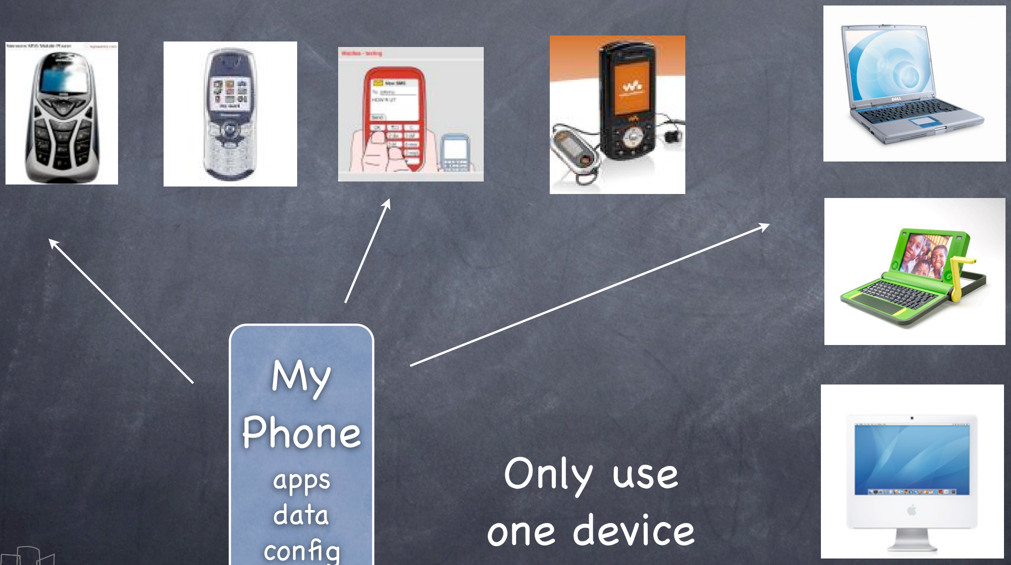
- Provide abstraction layer to overcome limitations of application, machines, and digital appliances
  - PC's have already been virtualized, I want Phone
- Why virtualize smartphone and handhelds?
  - it is powerful enough for many applications
  - but simple enough to be universal
  - phones are always with us

Solution: Virtualize

- Data, Applications, Communications can be moved to:
  - physical phone (symbian, windows, palm, linux, ...)
  - much different phone or PDA
  - laptop
  - desktop

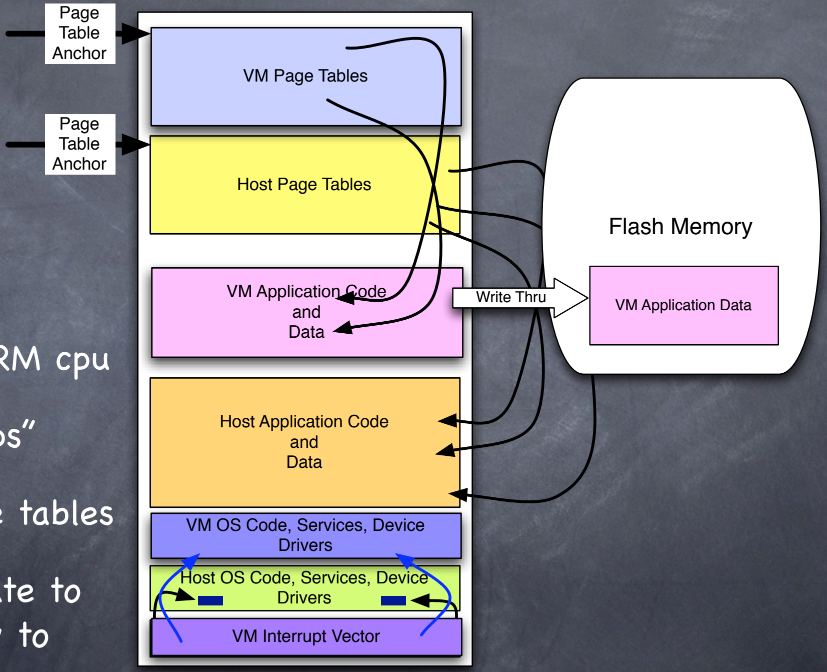


Virtual Phone



Only use one device at a time

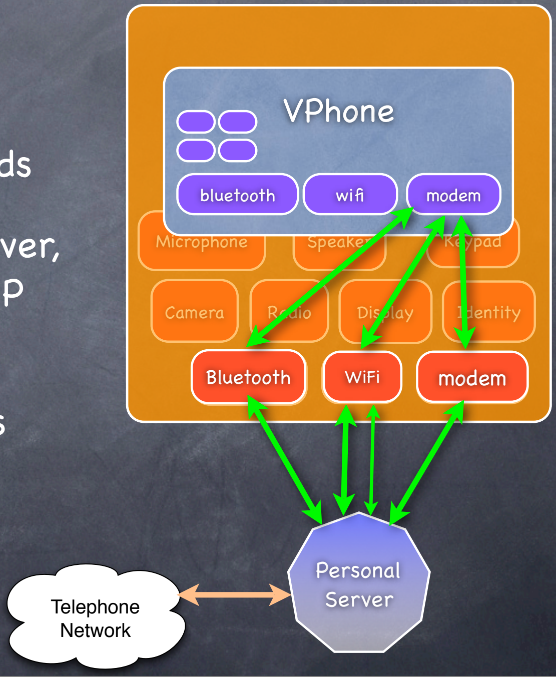
##


Page Table Anchor
Page Table Anchor
VM Page Tables
Host Page Tables
VM Application Code and Data
Host Application Code and Data
VM OS Code, Services, Device Drivers
Host OS Code, Services, Device Drivers
VM Interrupt Vector
Flash Memory
Write Thru
VM Application Data

- Virtualize ARM cpu
- Control "traps"
- Control page tables
- Snapshot state to make it easy to migrate

Virtual Telephone

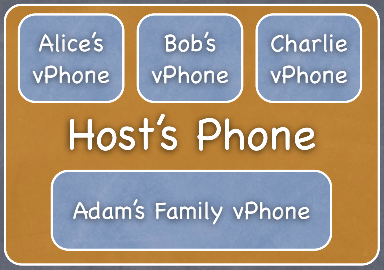
- Modem: call host modem, send dest. phone number, connect when call succeeds
- WiFi: send request to server, server calls, establish VoIP to server
- Bluetooth: VPhone acts as bluetooth headset



Virtualize Services

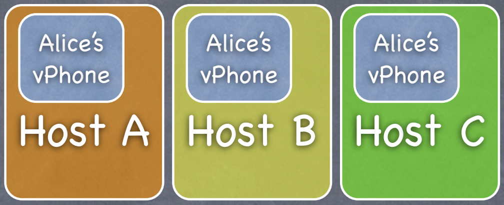
- Phone and PC Services
  - location
  - messages
  - browser
  - apps
  - .....

One Phone; Many vPhones



- User explicitly chooses vPhone to use
  - Protection (Isolation) mechanism
- Incoming call ==> correct vPhone accounting
- Transfer data between vPhones via Cut-n-paste
  - E.G. Take picture, copy, switch vPhones, Paste.

Group Phone one vPhone; many hosts



- Dial the group, all ring, anyone can answer
- Call out, anyone can join in.
- Anyone takes picture, appears in all group phones.
  - need to sync at virtual device layer
- Group Credit-card; anyone can charge and all see

Summary

- My phone is always with me and can monitor what I do with it
- But there is more to my life (beyond phone)
    - I spend time in front of a computer
- Infrastructure that enables application to run on large set of devices
- Make pervasive applications "real" and "private"

Phone Components

- Telephone & Keypad
- SMS Messaging
- MMS Messaging
- EMAIL
- Calendar
- Browser
- speaker & microphone
- camera (still & video)
- music player
- video player (also tv)
- WiFi
- Bluetooth
- Personal Identify Keys
    - e.g. credit cards

Also: Location Info & Applications

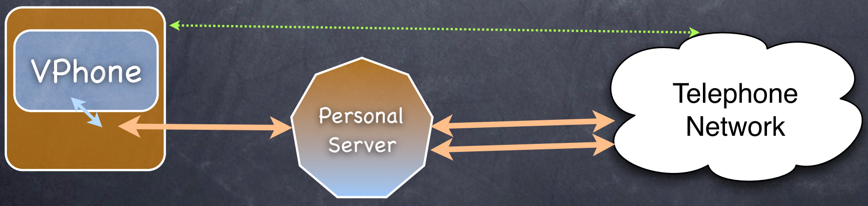
vPhone Components

- Telephone (in & out)
- Messaging (in & out)
- EMAIL
- Calendar (sync)
- Browser (record)
- speaker & microphone (virtualize)
- Cache & Memory
  - camera
  - music/video/tv
- WiFi (virtualize)
- Bluetooth (virtualize)
- Personal Identity (virtualize & trust)

Also: Location Info & Applications

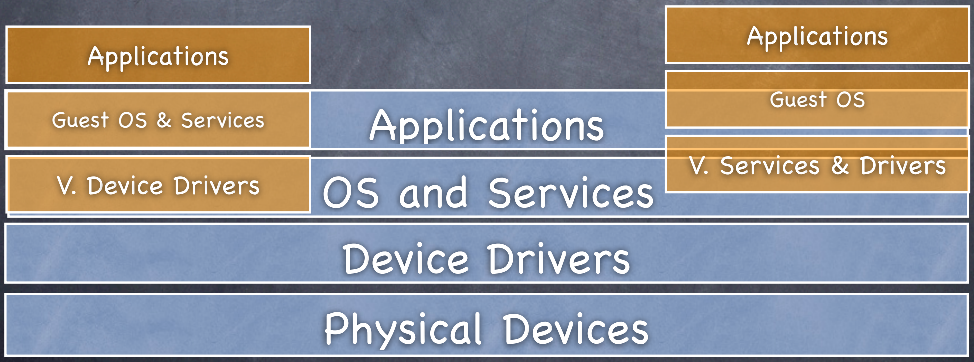
SMS Messages

- via Personal Server.
  - multiple messages in short time, encoded via phone call to server!
  - need to understand probability of next msg to be sent (or received). Can be learned
  - Simple compression ==> 2 messages in one



System Services

- Phone is more like appliance than general purpose dev.
- Applications on the phone make use of system services.
  - Like device drivers, these may be shared


Applications
Applications
Guest OS & Services
Applications
Guest OS
V. Device Drivers
OS and Services
V. Services & Drivers
Device Drivers
Physical Devices