

# Frameworks supporting pervasive environment applications

By

Manish Agarwal

## Need: New definition of application ...

An application is a means by which a user performs a task, not a software written to exploit a device's capabilities.

## Need : Computing Environment ...

Computing environment is an information enhanced physical space, not a virtual environment that exist to store & run software

# Challenges

- User focus on activity
- Devices are buried within the framework
- Task requirement changes all times
- Resources degrade frequently

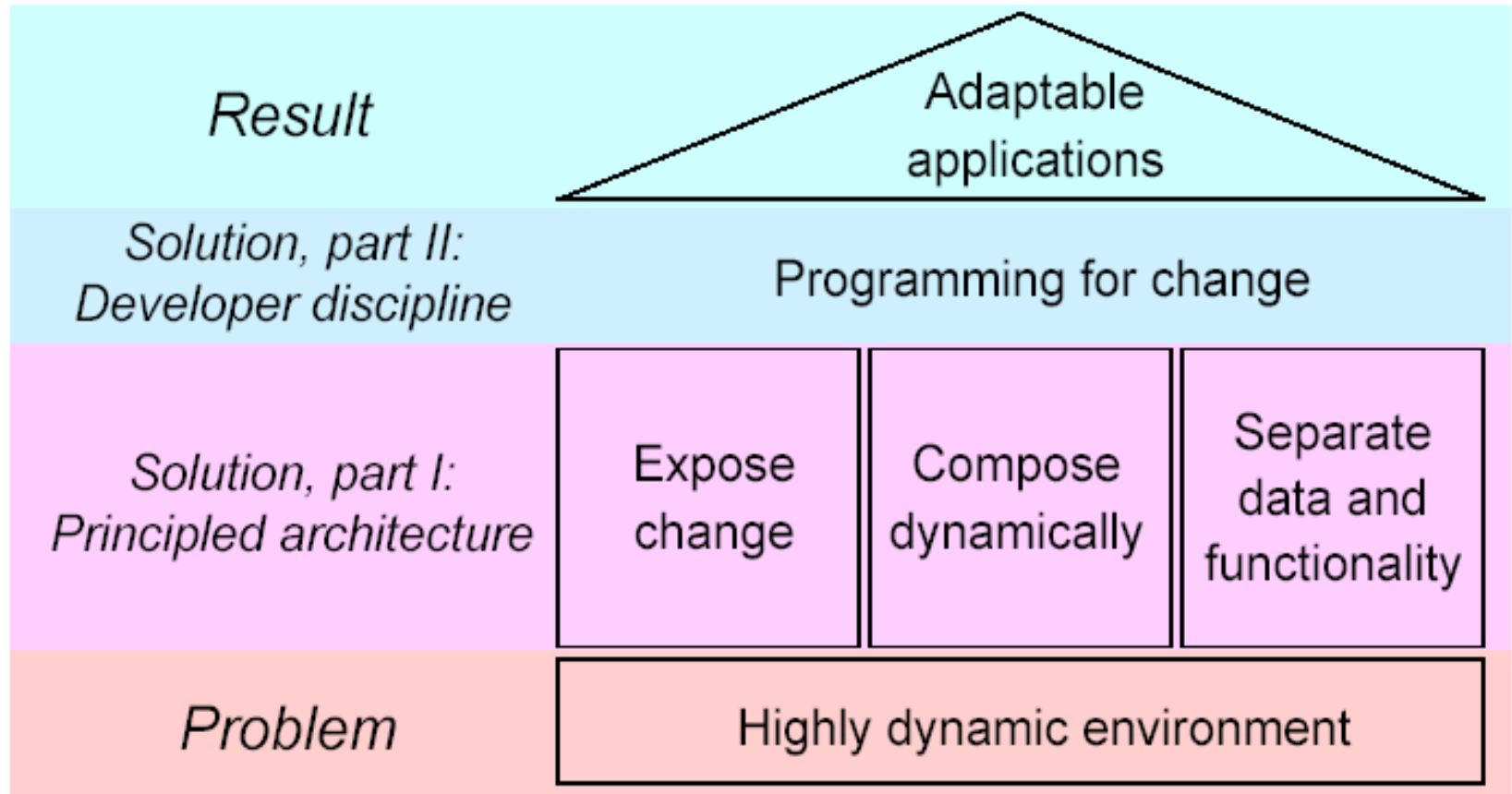
# OneWorld

- Goals
  - Develop a practical system that support building, deploying and using pervasive applications
  - Evaluate experiences and capture requirements

# Concept – One.World

- Embrace Contextual change
- Encourage Ad-Hoc Compositions
  - User expects applications and devices to just plug together (impractical to ask user to do it)
- Recognize sharing as the default
  - Dataflow and control flow managed by the framework with minimum support from end user or application developer

# OneWorld



# Expose Changes

- Let applications handle changes
- Do not hide distribution
- Provide primitives that assist in management i.e. Checkpoint, restore, move, find resources, data flow, etc...

# Composition

- Simplify interactions
- Modify and add behaviors
  - replication and migration logic
- Composition through interfaces
  - dynamic pool

# Separate data and functionality

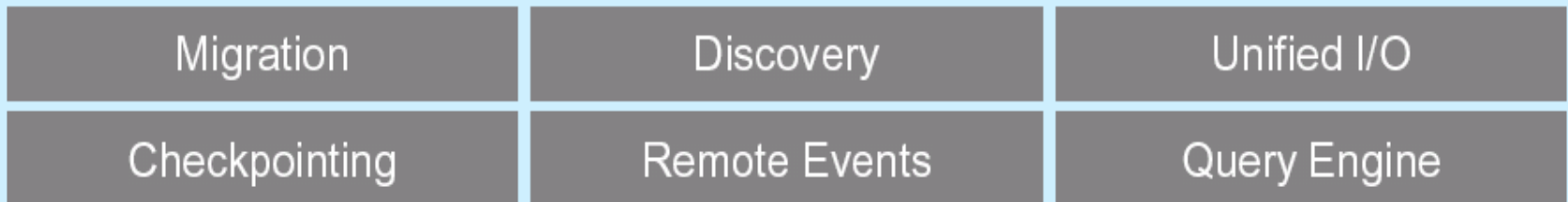
- Separate data simplify sharing, searching, translating...
- Specifying changes in terms of distribution rather than hiding information behind unifying interface
- Separate dataflow, control flow and data analysis.

# One.world System Architecture

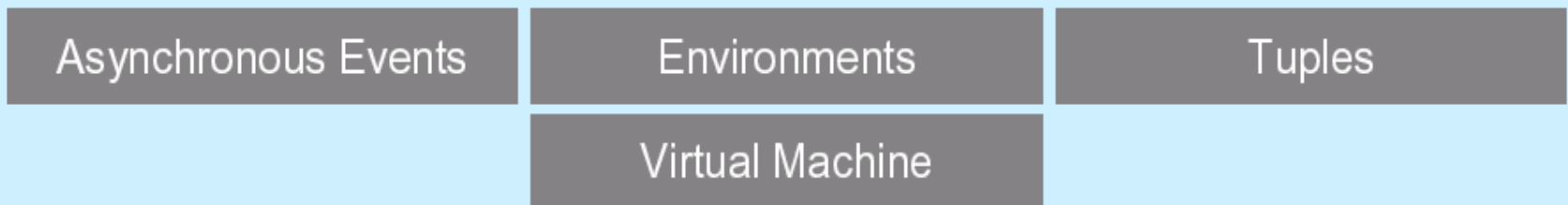
## USER SPACE



## SYSTEM SERVICES



## FOUNDATION SERVICES



*Change*

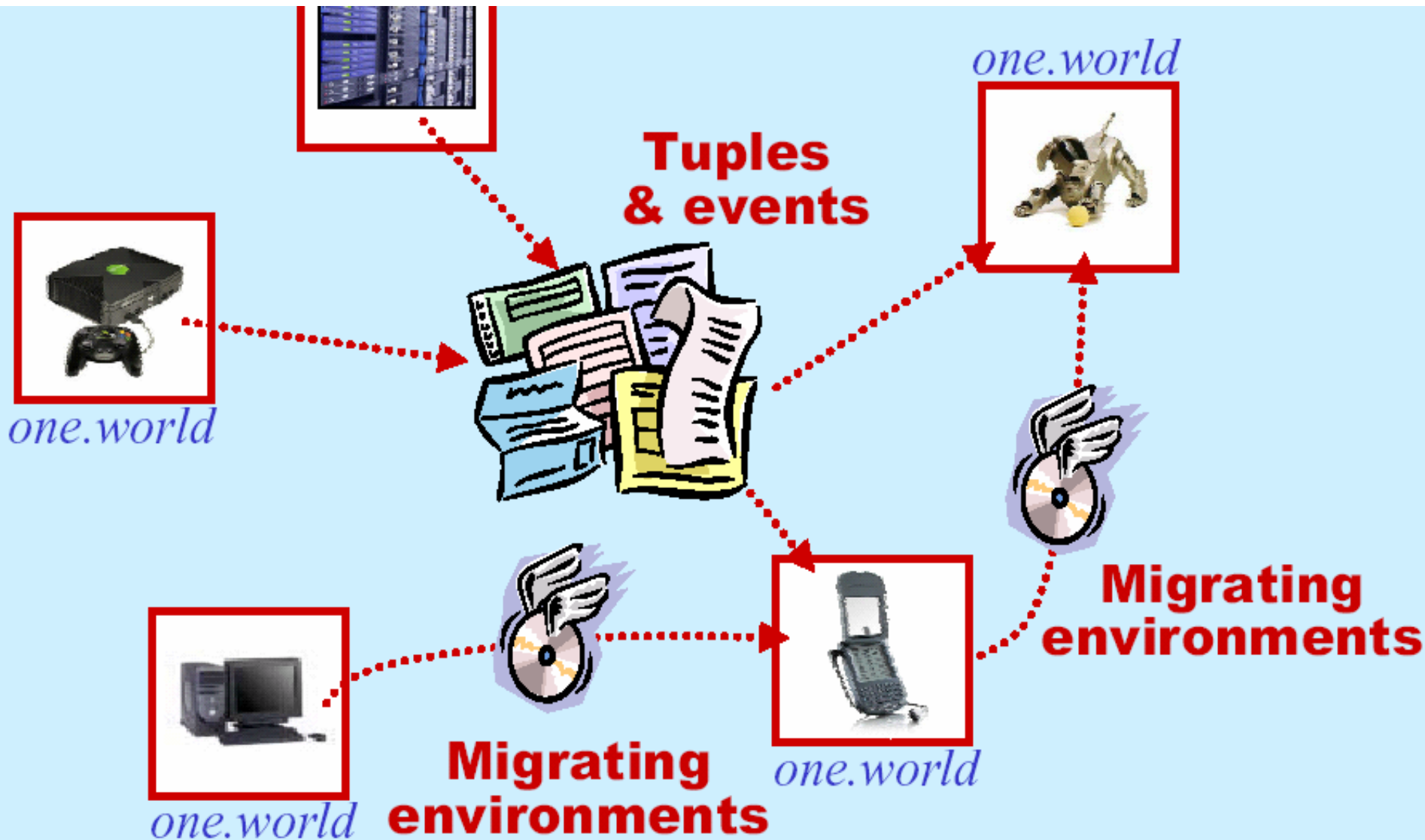
*Composition*

*Sharing*

# Foundation Services

- Virtual Machine: support ad hoc composition
- Tuples (self describing data): simplify sharing
- Events: make changes explicit to application
- Environment: Act like address spaces, including protection ; facilitate composition, check pointing, migration

# One World in Action



# System Services

Address the common requirements of pervasive applications

Application Requires

System Provides

Search

Query Engine

Locate

Discovery

Move

Migration

Fault-Protect

Checkpointing

Communicate in space

Remote Events

Communicate in time

Unified I/O

- **Discovery:**
  - find resources with unknown or changing location (locate handlers and route events)
  - Elected discovery server: self managing
- **Migration**
  - Moves and copies an application and its data (checkpoint – capturing execution change),
  - Works because applications already expects change
- **Unified I/O**
  - Provides unified interface to storage and networking and let applications share data between them

# Findings

- Asynchronous events scale better than threads and not only make state specification explicit but also support multiple scheduling policies
- Programming by specifying change is not hard and is ideal for pervasive environment
- Structured I/O , preserve structure of application data (atomic read and write) and is easy to express and manage

# Areas of improvements

- One.World has its own data model and network communications (require support to integrate legacy systems and applications)
- Events are harder to use (vs Swings)
- Require better support for managing asynchronous interactions
- (\*) Support swappable policies for supporting migrating (and compositions)
- Require support to map properties with behaviors and to enforce integrity constraints on data.

# Sentient Computing Systems

Systems which can change their behavior based on a model of the environment they construct using sensor data.

# Modeling the Environment

- Use data from sensors and from services
- Filter the incoming location data using knowledge of tracked objects' dynamics
- Update the world model using filtered data
- Increase fidelity of the model by determining if someone appears to be seated

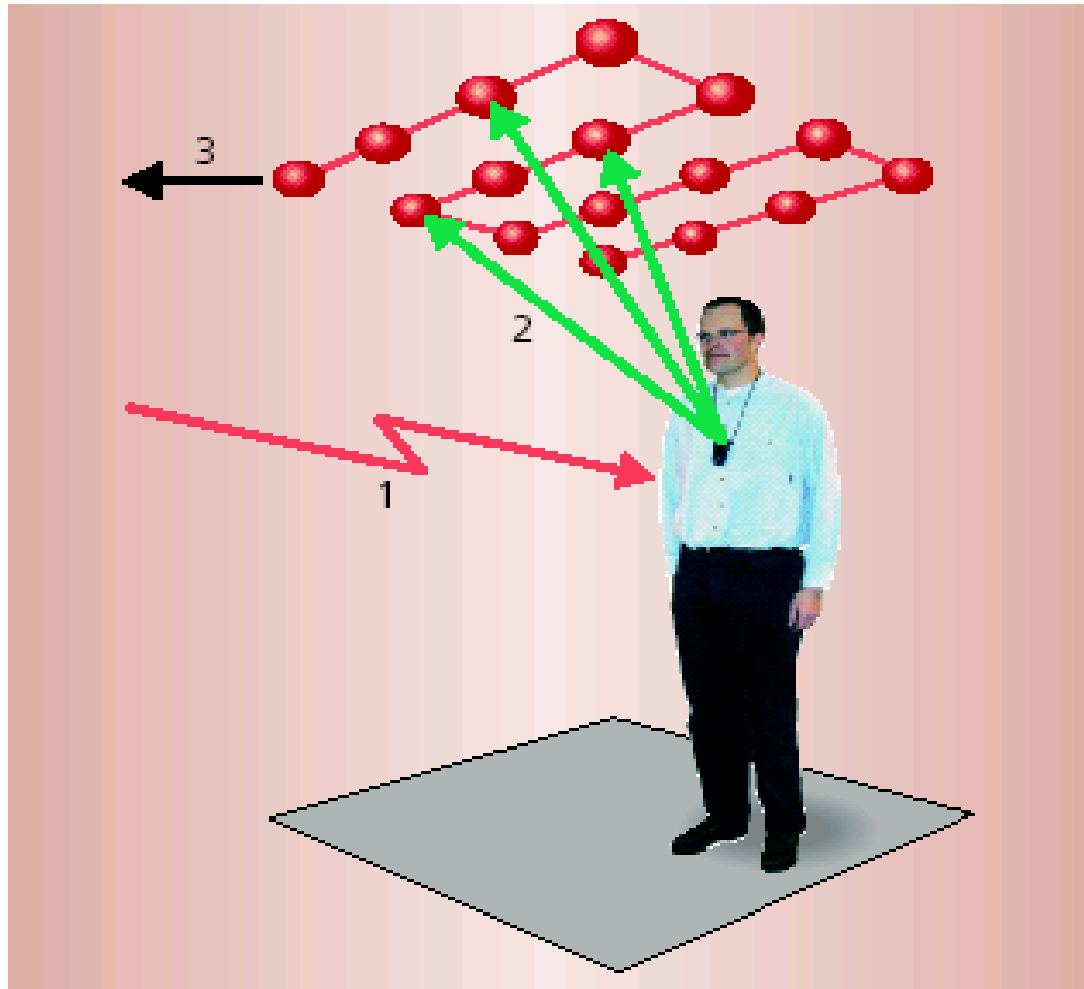
# Software support of sentient systems

- Spatial monitor
  - Formalizes imprecise spatial relationships in terms of containment and overlapping relationships between suitable 2D spaces
- Timeline-based data storage
  - Data generated by network appliances should be stored in a way that does not require the user to specify its destination.

# Sentient tools

- Browsing
  - Model browsers display the environment's current state, e.g., continuously updated map displaying personnel, furniture, telephones, workstations and other relevant information
- Follow-me Systems
  - Services are made ubiquitously available to “users” by moving their interfaces to the nearest appropriate input or output device

# Sentient System: Active Bat



*Figure 1. Operation of the Bat location sensor system. A Bat is triggered over a wireless link (1), which causes it to emit an ultrasonic pulse (2). Ceiling-mounted receivers measure the pulse's times of flight, and a controller retrieves the times of flight (3) over a wired network. The controller uses these measurements to calculate Bat-receiver distances and thus the Bat's 3D position.*

# Features of Active Bat Project ...

- Scheduling and power saving
  - The scheduling process allows the Bat to enter in a low-power sleep state.
  - A 10-bit local ID is used to achieve shorter addressing message and power saving.
  - Nonmoving Bats are placed into a low-power sleep state.
- Current Embodiment
  - Accuracy and Precision
    - 95% of 3D Bat position readings are accurate to within 3cm
  - Maximum Location update rate across each radio cell
    - 150 updates per second

# Limitations of Active Bat ...

- Active Bat requires a large fixed-sensor infrastructure throughout the ceiling.
- The system is sensitive to the precise placement of sensors.
- Therefore, there are disadvantages in scalability, ease of deployment, and cost.

# Future ..

Sentient computing is more than a solution to the problems of configuration and personalization. The environment itself will become the user interface.