

SARANA: A Space and Resource Aware Programming Infrastructure for Dynamic Networks

Pradip Hari

Dept. of Computer Science
Rutgers University

Research Team

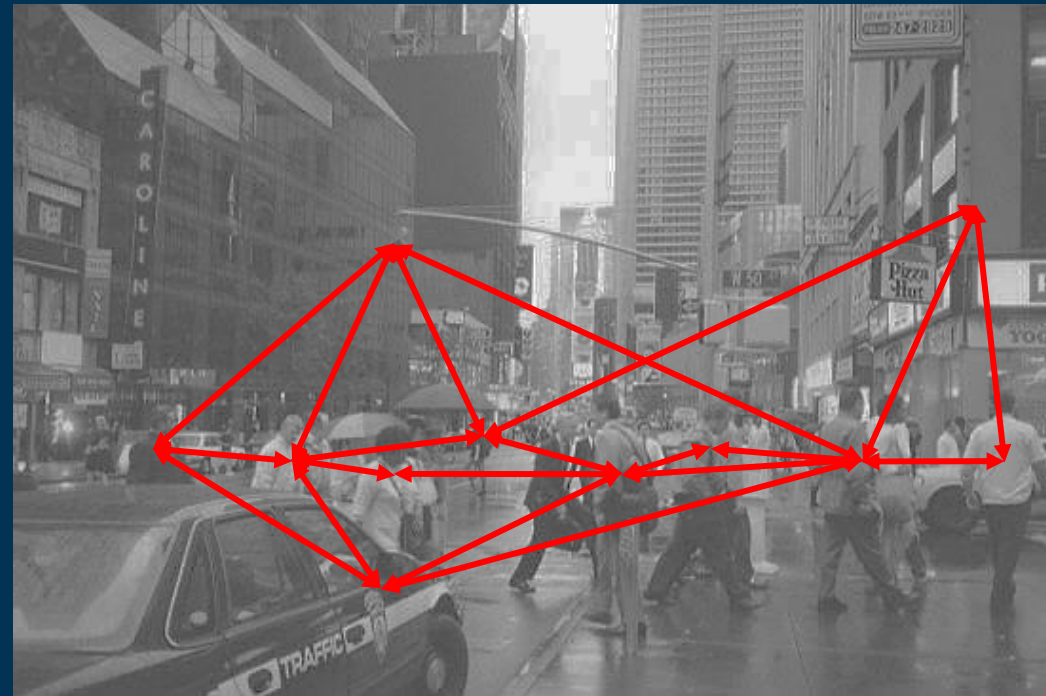
- Rutgers: Uli Kremer, John McCabe, Jonathan Banafato, Marcus Henry, Pradip Hari
- Princeton: Margaret Martonosi, Li-Shiuan Peh, Kevin Ko, Manos Koukoumidis
- Collaborators: Vinod Ganapathy, Amruta Gokhale (security)
- Funding: NSF, Nokia

Networked Computing: The Next Generation...



- All computing is mobile
- All computing is distributed
- Lots of devices are "computers"
- "Dynamic networks" (a.k.a. "grid networks")

Networked Computing: The Next Generation...

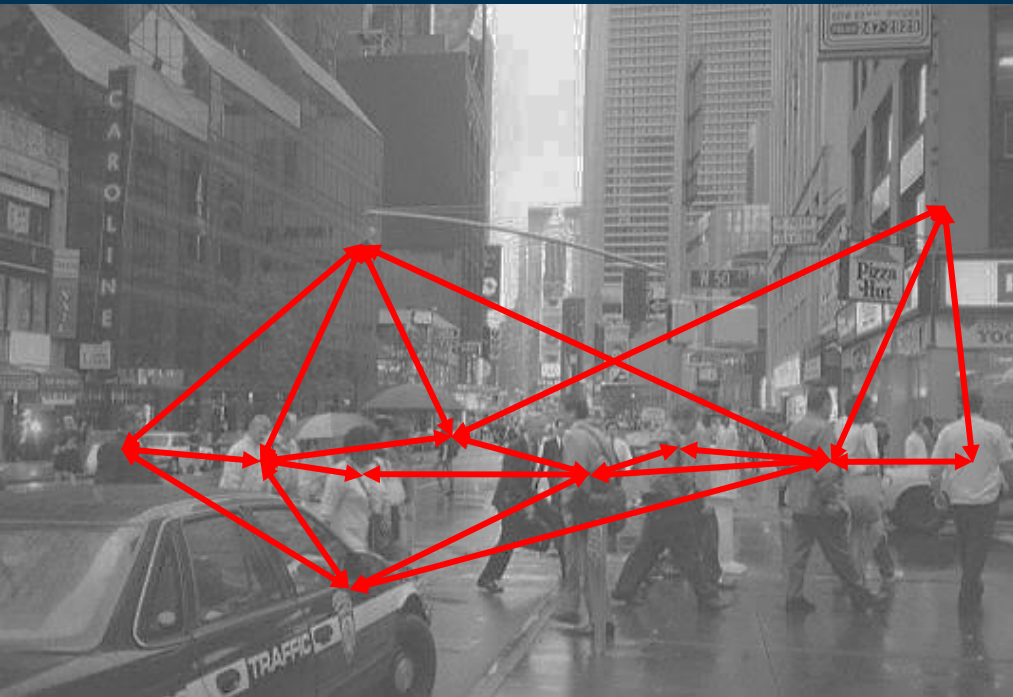


Opportunities:

- Collaborative, opportunistic data gathering and sharing
- Integrative data-mining and collaboration
- “Cloud computing”

Geographic location
is important!

Networked Computing: The Next Generation...



- Technical Challenges:
- Heterogeneous Computing:
 - Low & high-capability devices
- Heterogeneous Communication:
 - Wired, Wi-fi, Cellular
- Resource management
- Trust, Privacy, Security

How to program?
How to manage?

SARANA: System Support for Collaborative Low-Infrastructure Computation

SA → spatially aware

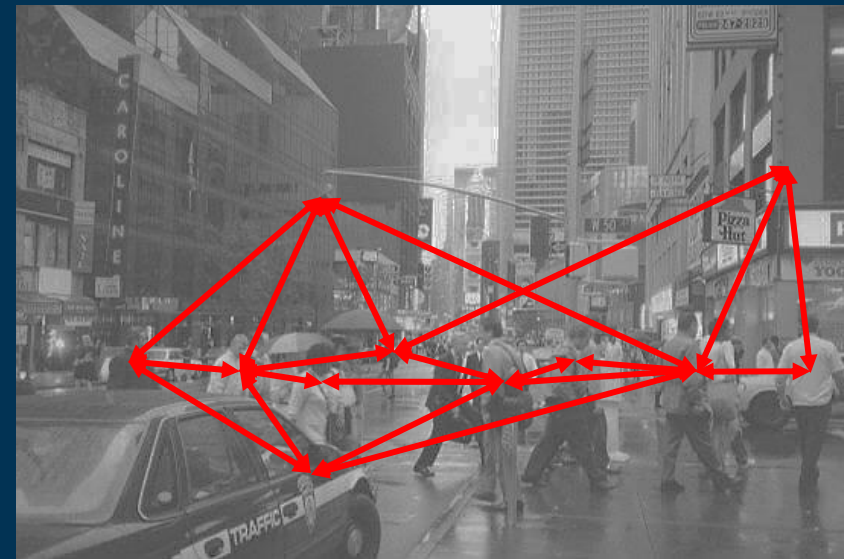
- Where are services located?

RA → resource aware

- How to use and share resources efficiently?

NA → network architecture

- How to leverage a variety of networked devices, from cell phones to servers?



SARANA: System Support for Collaborative Low-Infrastructure Computation

SA → spatially aware

- Where are services located?

RA → resource aware

- How to use and share resources efficiently?

NA → network architecture

- How to leverage a variety of networked devices?

Languages

- How do we express geographic relationships when we program?
- How do we optimize for them at compile time and run time?

SARANA: System Support for Collaborative Low-Infrastructure Computation

SA → spatially aware

- Where are services located?

RA → resource aware

- How to use and share resources efficiently?

NA → network architecture

- How to leverage a variety of networked devices?

Economic Models

- Dynamic adjustment of prices based on demand or energy

Optimization

- Cost management
- Resource selection

SARANA: System Support for Collaborative Low-Infrastructure Computation

SA → spatially aware

- Where are services located?

RA → resource aware

- How to use and share resources efficiently?

NA → network architecture

- How to leverage a variety of networked devices?

Directories

- How do we distribute and maintain information about what other nodes are nearby?
- Consistency
- Availability

Launch

- How to launch code on other nodes?
- Security?
- Efficiency?

Key Ingredients

- Macroprogramming
 - programmer sees region abstractions
- Quality of result (QoR)
 - programmer describes desired results
 - transparent resource cost management
- Dynamic code deployment
 - code chunks migrate where needed
 - services can be installed dynamically
- In-network computation

Sample Application: Amber Alert

- Finding kidnapped children
- Camera nodes: take photos of potentially interesting targets
- Analysis nodes: apply image-understanding to evaluate photos
- Display nodes: show photos to *nearby* individuals (e.g., police officers)

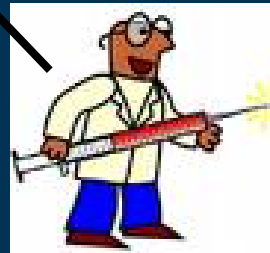
Event Chronology



Event Chronology



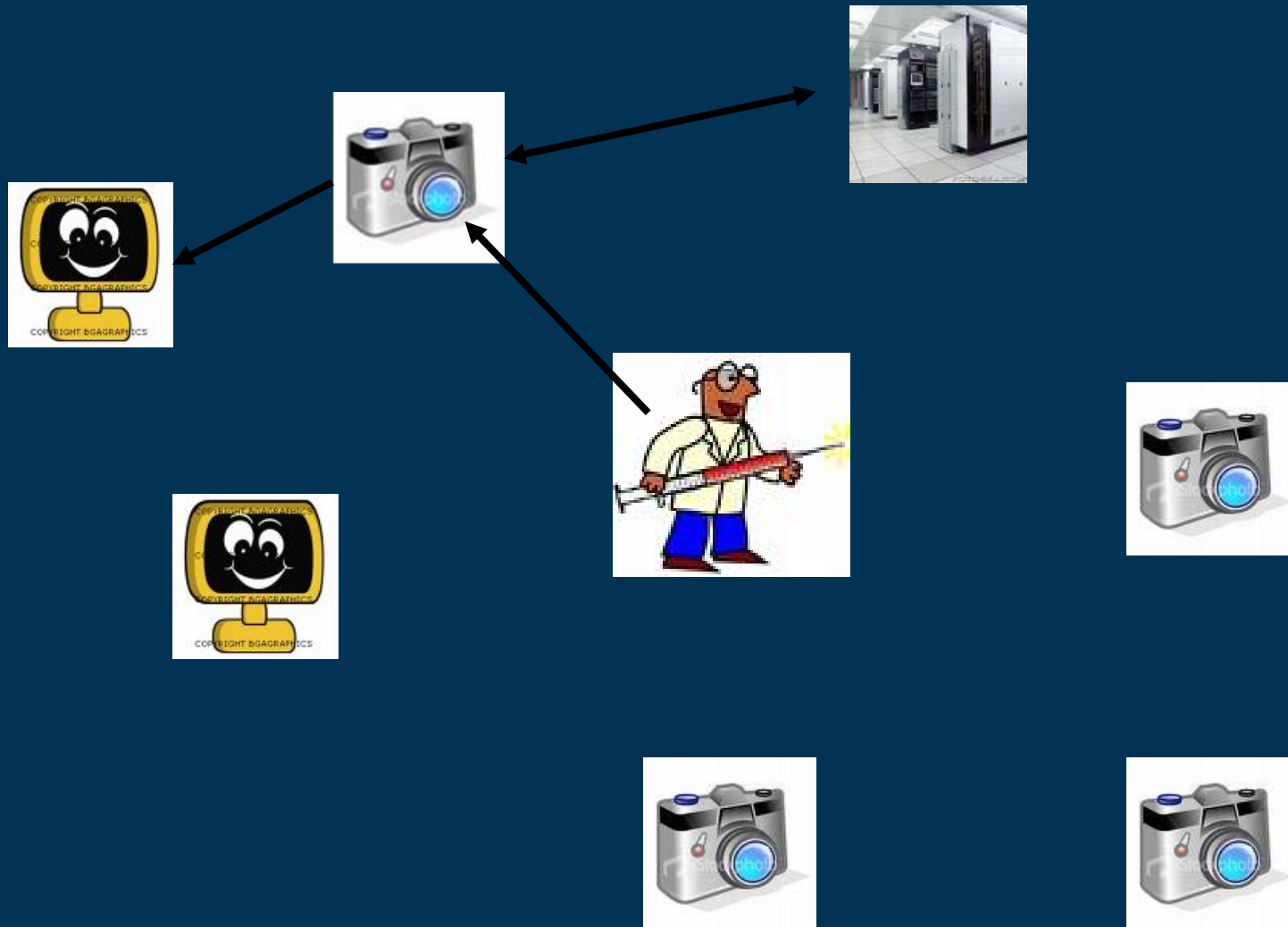
Event Chronology



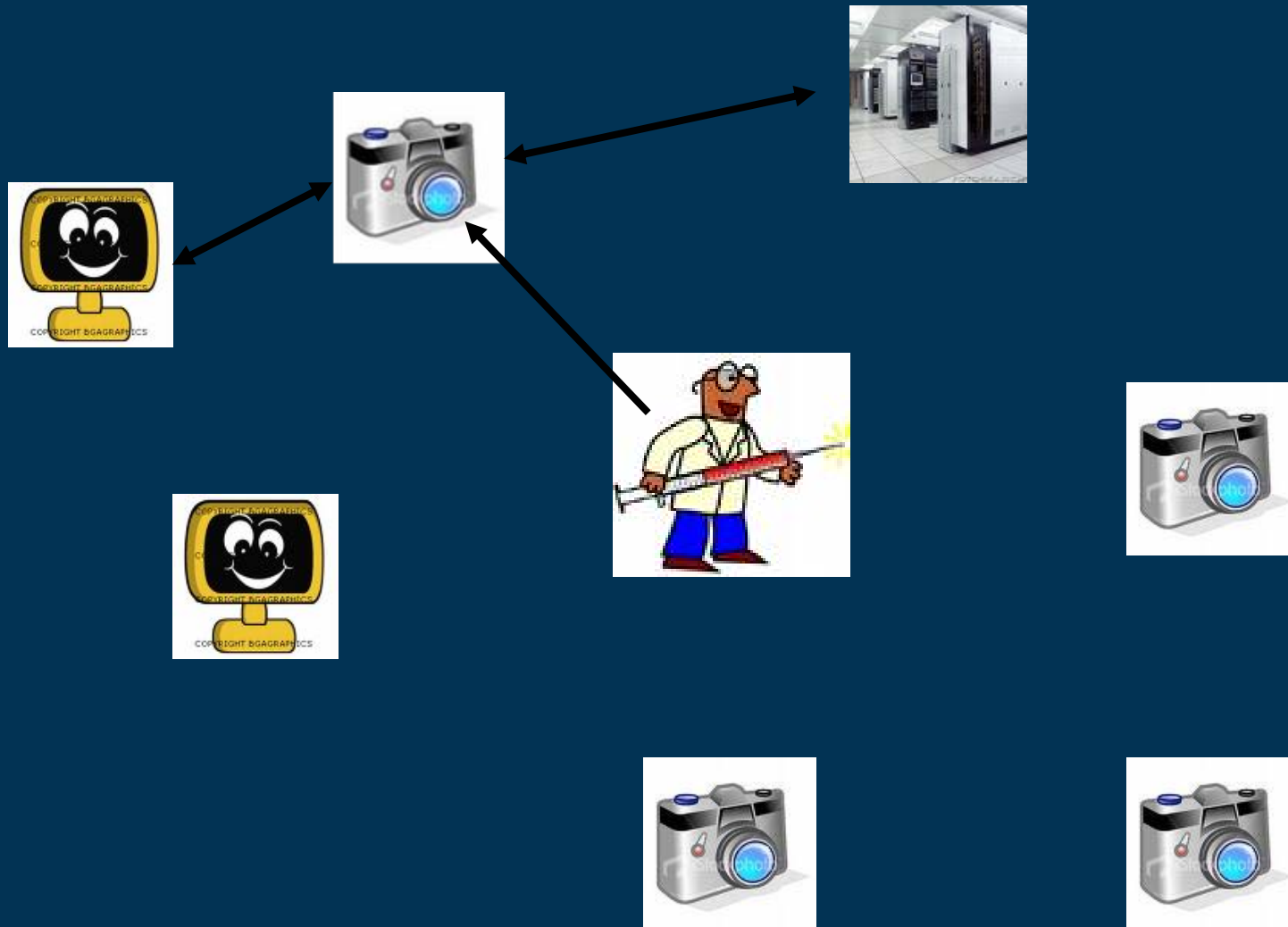
Event Chronology



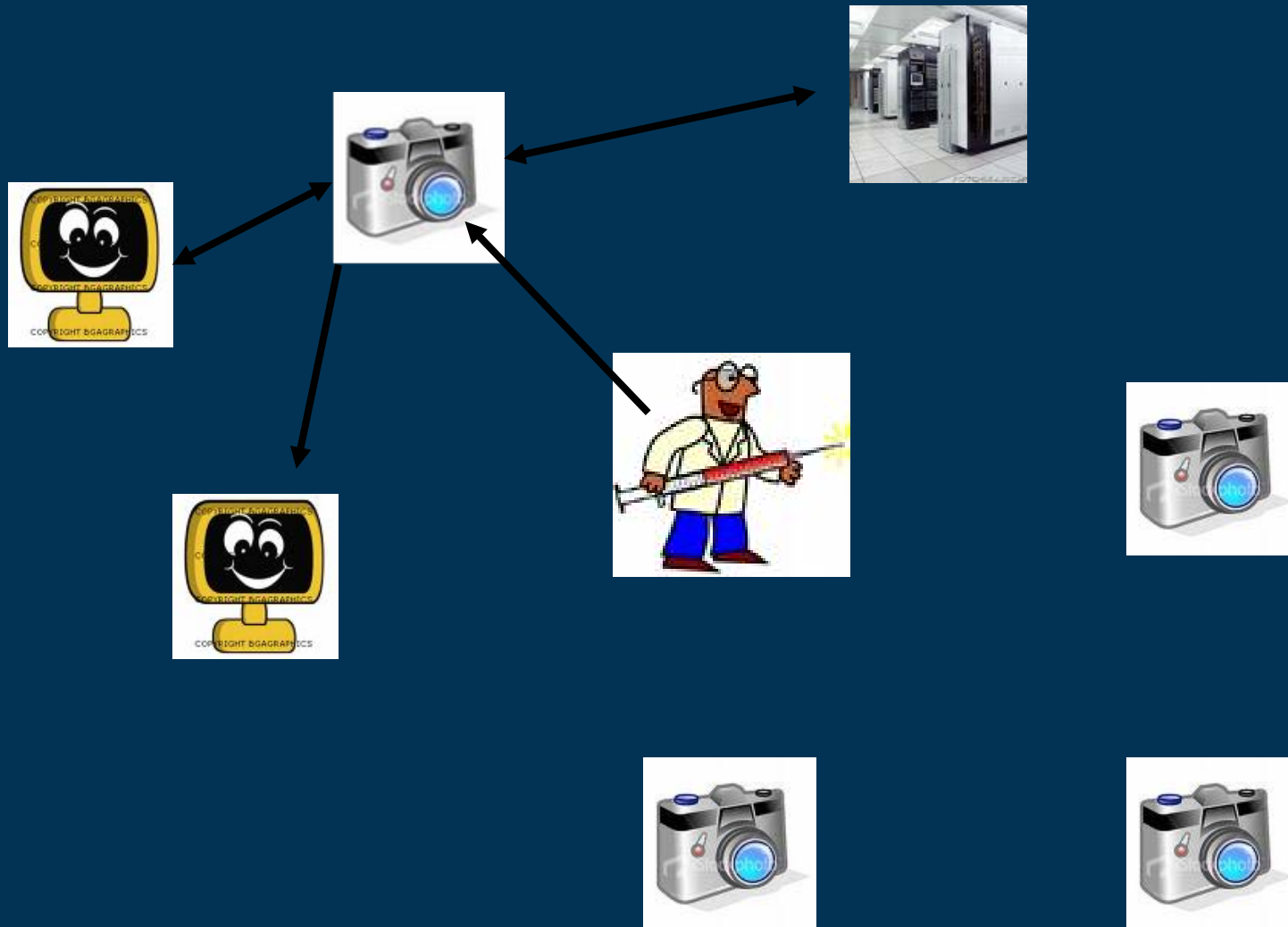
Event Chronology



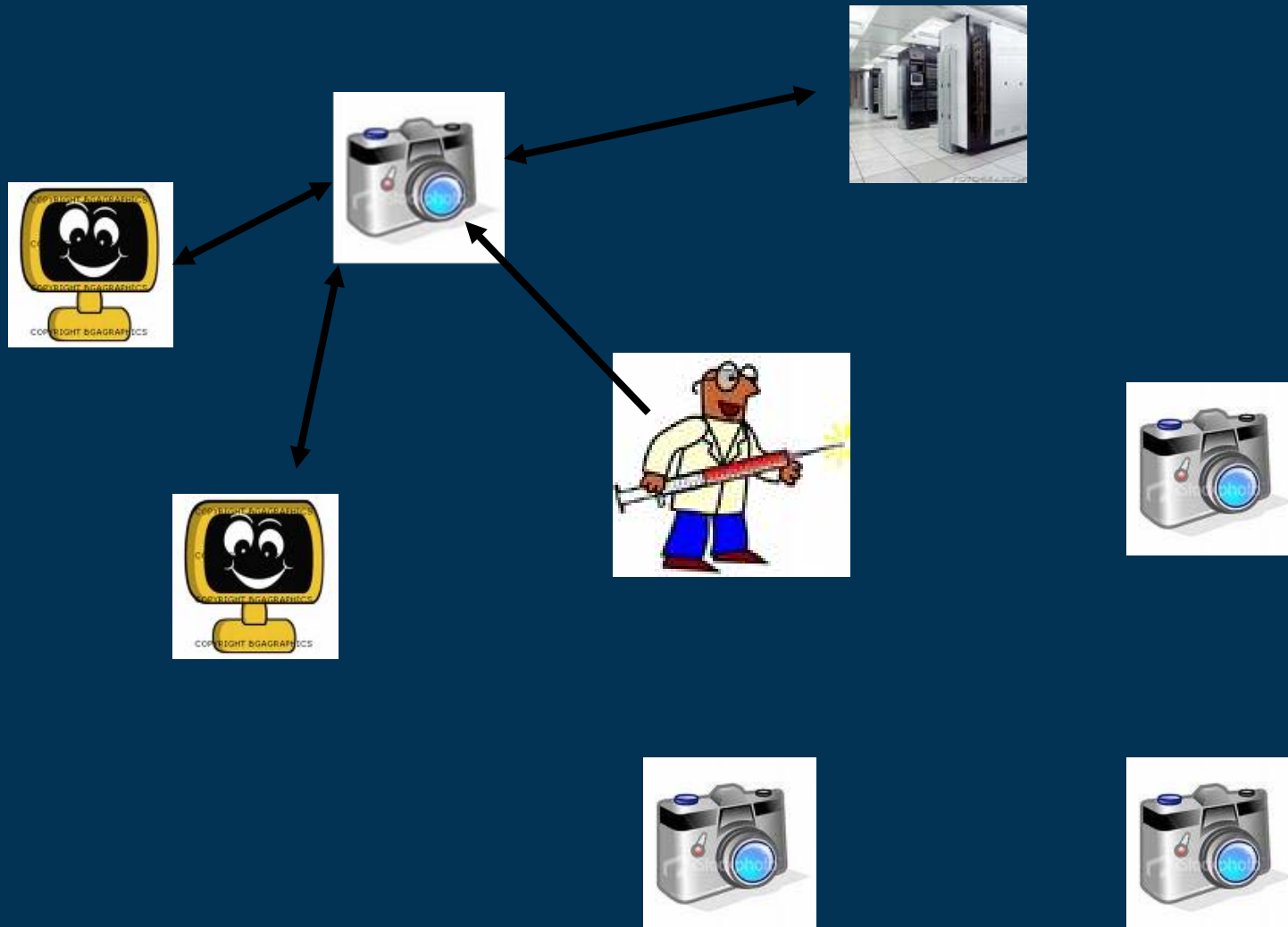
Event Chronology



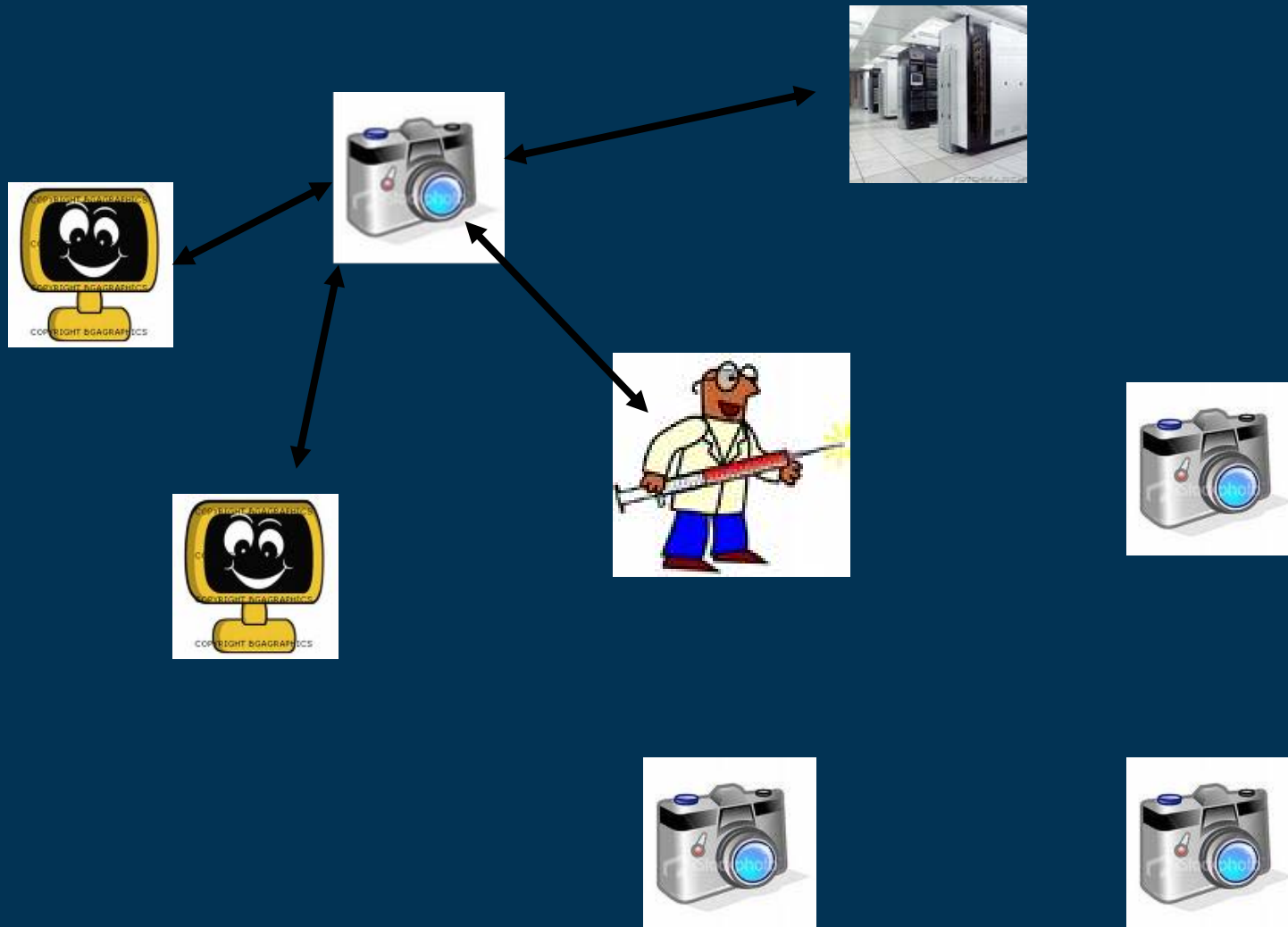
Event Chronology



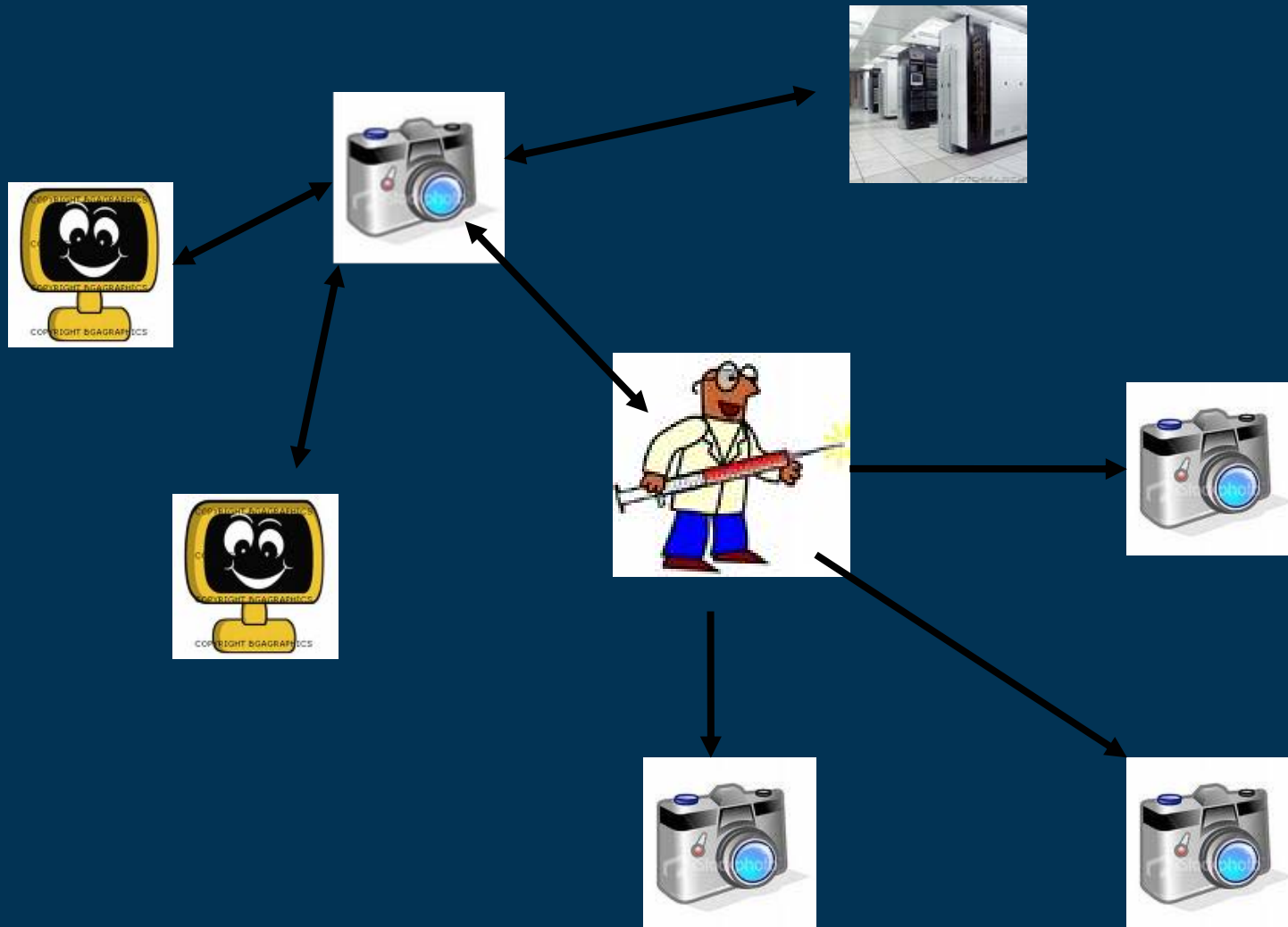
Event Chronology



Event Chronology



Event Chronology



Sample Code

```
collection_reduction ArrayList lst = new
ArrayList<ImageBlob>();
sum_reduction int displayedCount = 0;
visiteach 10-100 camera : cameraSpace : cluster(GOODPIC, 20){
    ImageBlob image = camera.takePhoto();
    or_reduction boolean success = false;
    visitone analyzer : analysisSpace {
        success |= analyzer.accepts(p);
    }
    if (success) {
        report GOODPIC;
        lst.add(image);
        visiteach monitor : displaySpace {
            monitor.display(image);
            displayedCount++;
        }}
    }}}
```

SARANA System Components

- Compiler
 - Task decomposition
 - Space/time scheduling
- Runtime System
 - Directory / Yellow Pages
 - Resource Monitor / Policy Controller
 - Process Manager
 - Code Distribution Manager
- Iteration Runtime Library
 - QoR spec satisfaction (incremental execution)

Implementation Status

- Platforms supported now:
 - Nokia N810 (Maemo)
 - Neo FreeRunner (OpenMoko)
 - Simulator (generic UNIX)
- Compiler status:
 - Initial version complete
- Near completion:
 - Peer-to-peer directory

Future Work

- Scheduling optimizations
- More expressive QoR specification
- Cost caching & prediction
- Wider set of platforms

Summary

- Dynamic networks are the future
- Robust programming infrastructure needed
- Geography important
- Resource management essential