



# Developing Adaptive Groupware Applications Using a Mobile Component Framework ( DACIA )

Radu Litiu and Atul Prakash  
University of Michigan, EECS



---

# Outline

- Motivation and Goals
- DACIA Architecture
- Building Adaptive Groupware Applications
- Related Work
- Summary

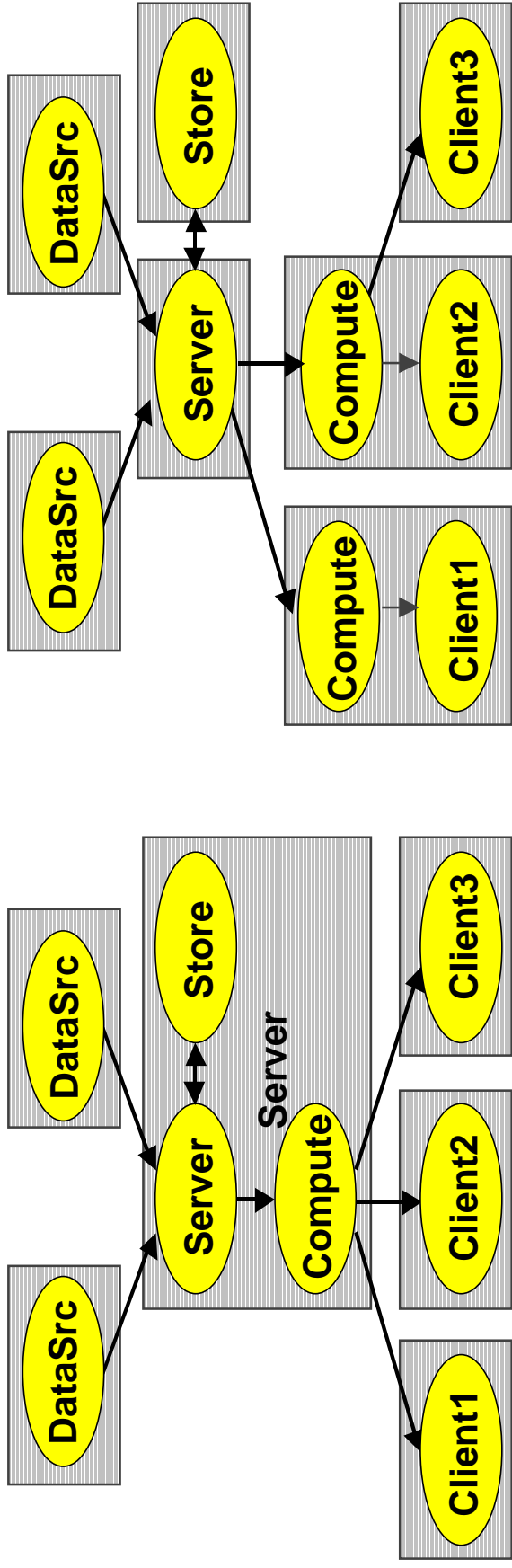


---

# Problems

- Flexibility and adaptability in CSCW systems (Roseman93, Bentley95, Shen92)
- Variability and heterogeneity
  - User and application demands
  - Hardware and network variability
- Application and user mobility (Bellotti96, Chung96 – X client migration)
  - Support for offline users

# Adaptation Need - SPARC Collaboratory

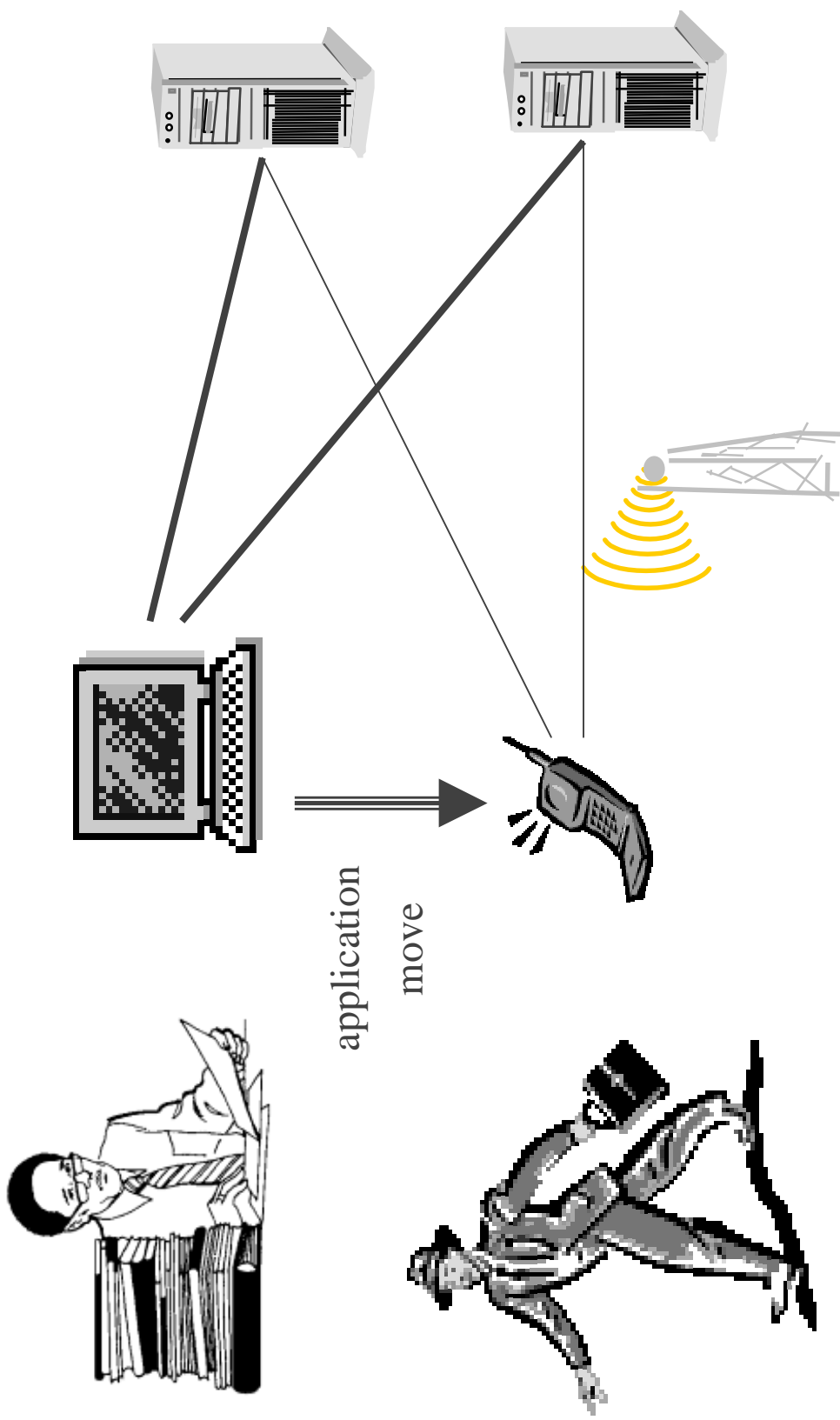


An application is a graph of connected components.

Possible changes:

- Execute the computation on the client machine
- Store computed images instead of raw data
- Add/remove modules

# Mobility





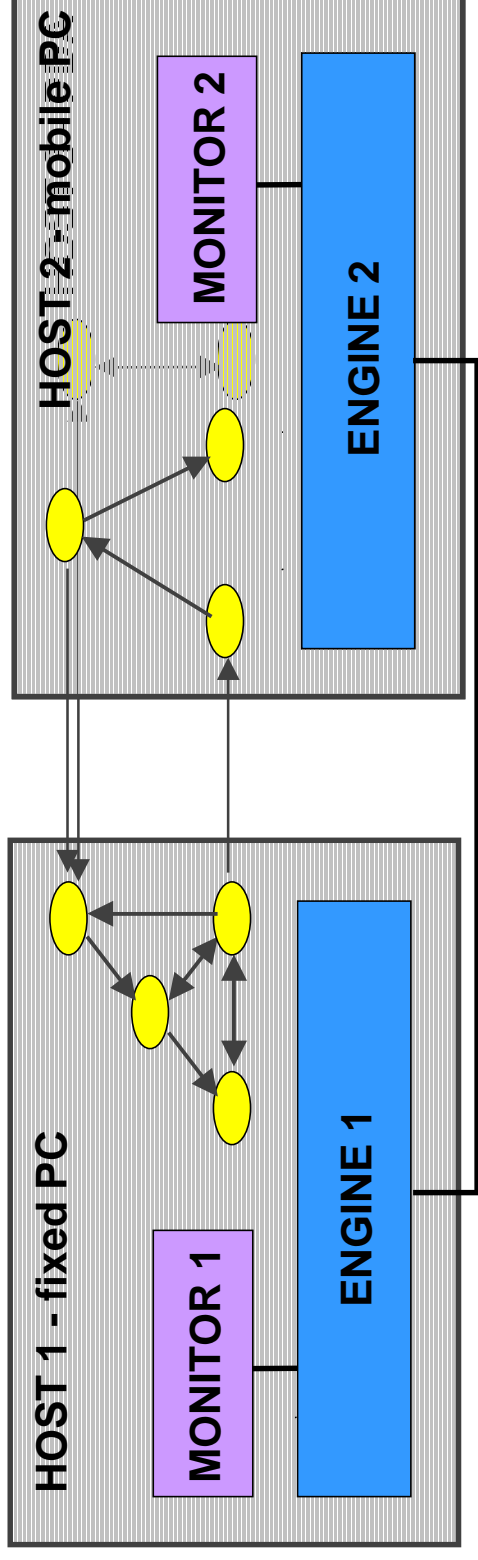
---

# DACIA\* Features

- Dynamic application reconfiguration
- Component mobility
- Persistent connectivity between components
- Support for offline users – application parking

\*Dynamic Adjustment of Component InterActions

# DACIA Architecture



## Engine (mechanism)

- Communicate between hosts
- Manage connections between components
- Relocate components
- Reconfigure the application

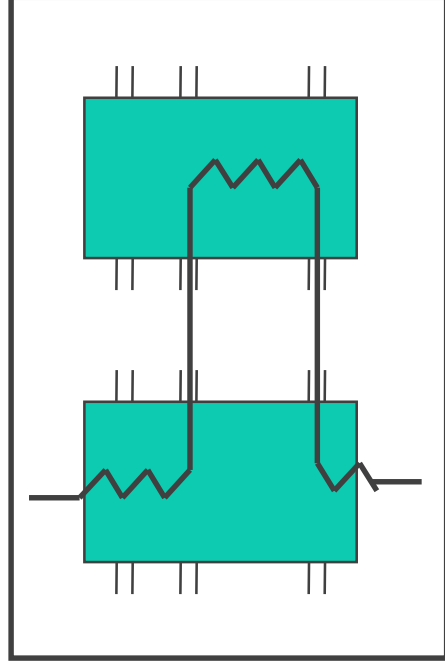
## Monitor (policy)

- Monitor performance
- Make reconfiguration decisions
- Implement application-specific reconfiguration policies

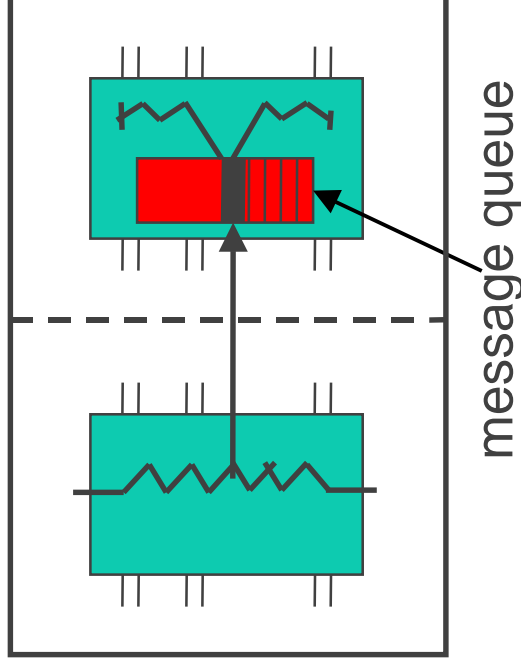
# PROCS\*

- Communication through ports
- Low communication costs
- Component mobility – state transfer
- Persistent connectivity

synchronous communication



asynchronous communication



\*Processing and Routing Component



# Component Mobility (I)

```

TERMI2
Engine > Connection
301]
connectProcs 402016
Engine: invalid com
Engine > connectProc
Engine > print
Connections :
SocketLaddr=sec
SocketLaddr=sar
Local PROCs :
402016 Chat:1,
0 -> 449016
Remote PROCs :
449016 Chat:1,
0 -> 402016
Engine > Can't send
print
Connections :
SocketLaddr=seoul,eecs.umich.edu/141.213.10.147,port=34192,localport=6301
SocketLaddr=sanjuan,eecs.umich.edu/141.213.10.55,port=36074,localport=6301
Local PROCs :
402016 Chat:1, 2 ports, host: saturn
0 -> 449016:1
1 -> 449016:0
Remote PROCs :
449016 Chat:1, 2 ports, host: seoul
0 -> 402016:1
1 -> 402016:0
Engine >

```

```

TERMI1
Connection closed: Socket[addr=saturn.eecs.umich.edu/141.213.10.1
00,port=1185,localport=6301]
Can't re-establish connection to host saturn
print
Connections :
Local PROCs :
Remote PROCs :
Engine > print
Connections :
Local PROCs :
Remote PROCs :
Engine > connect saturn 6301
Connection established: Socket[addr=saturn/14
01,localport=36074]
Engine > print
Connections :
Socket[addr=saturn/141.213.10.100,port=630
449016 Chat:1, 2 ports, host: seoul
0 -> 402016:1
1 -> 402016:0
402016 Chat:1, 2 ports, host: saturn
0 -> 449016:1
1 -> 449016:0
Engine >

```

```

TERMI2
Engine > Connection
301]
connectProcs 402016
Engine: invalid com
Engine > connectProc
Engine > print
Connections :
SocketLaddr=sec
SocketLaddr=sar
Local PROCs :
402016 Chat:1,
0 -> 449016
Remote PROCs :
449016 Chat:1,
0 -> 402016
Engine > Can't send
print
Connections :
SocketLaddr=seoul,eecs.umich.edu/141.213.10.147,port=34192,localport=6301
SocketLaddr=sanjuan,eecs.umich.edu/141.213.10.55,port=36074,localport=6301
Local PROCs :
402016 Chat:1, 2 ports, host: saturn
0 -> 449016:1
1 -> 449016:0
Remote PROCs :
449016 Chat:1, 2 ports, host: seoul
0 -> 402016:1
1 -> 402016:0
Engine >

```

# Component Mobility (II)

**DACIA: application layout for host saturn**

File View Engines Procs Help

saturn

seoul

449016 Chat

sanjuan

402016 Chat

Connections :  
Socket[addr=seoul.eecs.umich.edu/141.213.10.]

Local PROCs :  
Remote PROCs :  
402016 Chat:1, 2 ports, host: sanjuan  
0 -> 449016:1  
1 -> 449016:0  
449016 Chat:1, 2 ports, host: seoul  
0 -> 402016:1  
1 -> 402016:0

```

X TERM2
Remote PROCs :
449016 Chat:1,
0 -> 402016
Engine > Can't send
print
Connections :
SocketLaddr=seoul
SocketLaddr=sanjuan
Local PROCs :
402016 Chat:1,
0 -> 449016
1 -> 449016
Remote PROCs :
449016 Chat:1,
0 -> 402016
1 -> 402016
Engine > Warning: *
print
Connections :
SocketLaddr=seoul
SocketLaddr=sanjuan
Local PROCs :
Remote PROCs :
402016 Chat:1, 2 ports, host: sanjuan
0 -> 449016:1
1 -> 449016:0
449016 Chat:1, 2 ports, host: seoul
0 -> 402016:1
1 -> 402016:0
Engine >

```

**DACIA: application layout for host sanjuan**

File View Engines Procs Help

sanjuan

402016 Chat

seoul

449016 Chat

Connections :  
Socket[addr=saturn/141.213.10.100,port=6301.]

Local PROCs :  
Remote PROCs :  
449016 Chat:1, 2 ports, host: seoul  
0 -> 402016:1  
1 -> 402016:0

```

41.213.10.100.port=63
3
Engine > print
Connections :
Socket[addr=saturn/141.213.10.100,port=6301.]
Local PROCs :
402016 Chat:1, 2 ports, host: sanjuan
0 -> 449016:1
1 -> 449016:0
Remote PROCs :
449016 Chat:1, 2 ports, host: seoul
0 -> 402016:1
1 -> 402016:0
Engine >

```

**Chat**

Menu

Enter a new message:

Send

Latest message:

OUT: Hello, chat user from seoul  
IN : Where are you located ?  
OUT: I am on saturn.  
OUT: I am moving to sanjuan right now.  
IN : OK, I am waiting...

```

1 -> 449016:0
Engine > print
Connections :
Socket[addr=saturn/141.213.10.100,port=6301.]
Local PROCs :
402016 Chat:1, 2 ports, host: sanjuan
0 -> 449016:1
1 -> 449016:0
Remote PROCs :
449016 Chat:1, 2 ports, host: seoul
0 -> 402016:1
1 -> 402016:0
Engine >

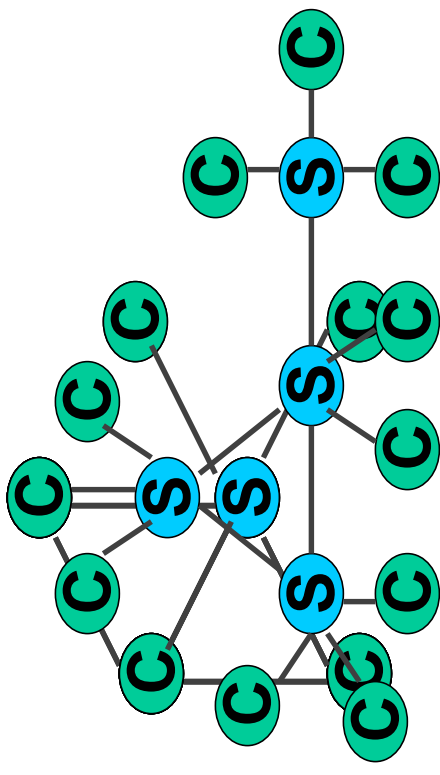
```



# Dynamic Application Reconfiguration

- Change connections between components
- Change components' location
- Load new components

An adaptive application:  
multi-party communication





---

# Reconfiguration Mechanisms

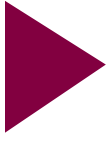
- Specialized monitors
- Programming API :
  - *connect [hostname] [portnumber]*
  - *connectProcs [sourceProcID] [sourcePortNo] [destProcID] [destPortNo]*
  - *disconnectProcs [sourceProcID] [sourcePortNo]*
  - *move [procID] [hostname]*
  - *start [procID]*
  - *startMonitor*
- Command-line interface
- Graphical interface



---

## Related Work


- Extensible architectures: Fitzpatrick96, Lee96
- Context-aware applications: Harter99
- Cooperative buildings: Streit98
- Code mobility & mobile agents: Telescript, Obliq, Sumatra, Tacoma, Aglets
- Distributed component architectures: CORBA, Scout, FarGo, Rover



---

# Summary

- DACIA - a framework for building adaptive groupware applications
- Dynamic reconfiguration
  - Improve the performance
  - Customized and extensible configurations
- Application and user mobility
- Persistent connectivity
- Application parking



---

# Current and Future Work

- PDA porting
- Policies and algorithms for application reconfiguration
- Formalism for specifying components and composition rules
- Deployment and experimentation
- Security infrastructure