

Resource Discovery with Evolving Tuples

Drew Stovall and Christine Julien
The University of Texas at Austin
{dstovall, c.julien}@mail.utexas.edu

Presented at:

Engineering of Software Services for Pervasive Environments (ESSPE '07)
(at ESEC/FSE 2007 - Dubrovnik, Croatia, September 4, 2007)

Overview

- Target environment
- Framework support for pervasive services
- Existing tuple models
- Evolving tuples
- Discovery - a sample pervasive service
- Discovery with evolving tuples
- Open Questions

Target Environment

- Heterogeneous connections
 - 802.11, Zigbee, BlueTooth, proprietary
- Heterogeneous capabilities
 - Processor(s), Memory, Non-volatile storage, Sensors
- Heterogeneous OS and software
 - Linux, TinyOS, proprietary
- Power provided by batteries

Target Environment

- We target the lowest capability nodes and networks
 - Sensor class hosts
 - Slow processors
 - Small memory
 - Small nonvolatile storage (flash)
 - Ad-hoc networks
 - Datagram based
 - With or without ACK
 - Max payload ~10's of bytes
 - Frequent connectivity changes

Frameworks for Pervasive Services

- Software engineering and services rely on frameworks to provide or support certain characteristics

▪Autonomous Action	▪Best Effort
▪Fully Distributed	▪Context-Aware
▪Open	▪Compatible
▪Localized	▪Implementable

- Tuples can provide a framework for these services

Tuples

- Original Tuple Model (Gelernter & Bernstein – 1982)
 - **Tuple** = name + ordered list of values
 - < name, value, value, ... >*
 - <"ping", 10, 5, 3>*
 - **Pattern** = name + ordered list of predicates
 - <"ping", 10, i:integer, j:integer>*
 - **Tuple Space** = bag of tuples
 - Add / Peek / Remove operations
- Tight design-time coupling between producers and consumers

Tuples

- Dictionary-like tuples (ELights / Lights)
 - **Tuple** = unordered set of name/value pairs
< (name, value), (name, value), ... >
< (msg_type="ping"), (destination=10), ..., (ttl=3) >
 - **Pattern** = unordered set of name/predicates
< (msg_type,"ping"), (ttl, "?integer")>
- Values are static in transit
- Producers rely on consumer behavior

Tuples

- Want general mechanism to process context data
 - Send
 - Collect
 - Aggregate
 - Combine
- Want behavior that can be redeployed or updated

Evolving Tuples

- Embed some of the service behavior into the network messages
- Builds on dictionary-based approach
- Adds the *formula* element to each field
- Adds the *evolution context*

Evolving Tuples

- The *formula* element
< (name, value, formula), (name, value, formula), ... >
< (oneHourAgo, 12, context[hour] - 1) >
- Used to automatically update the values of a tuple field

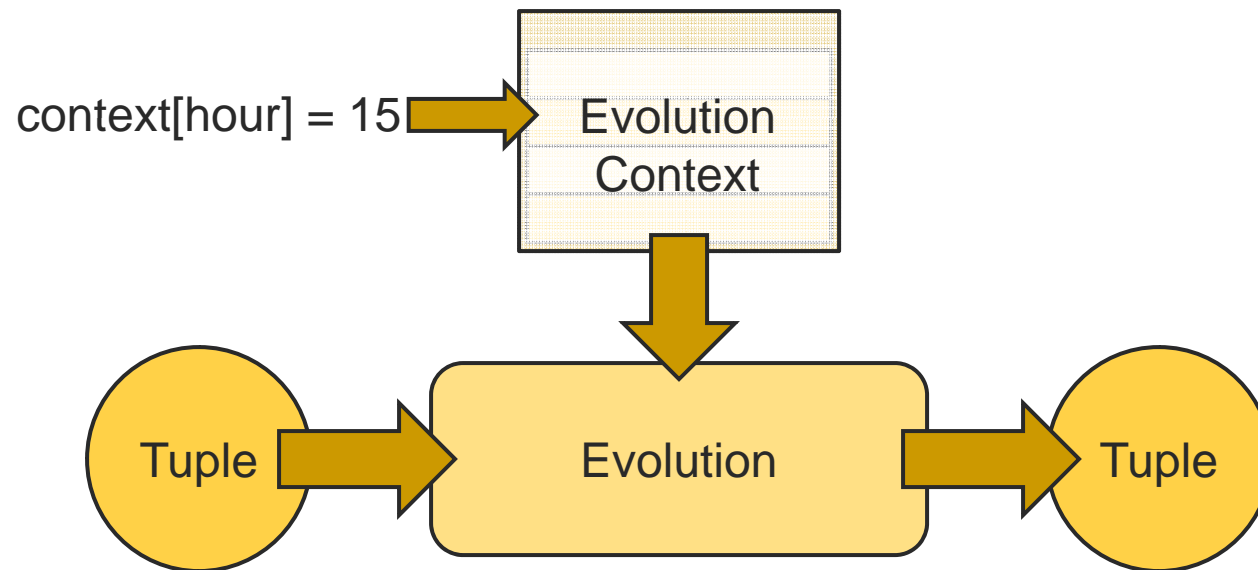
Operators	Function
+ - * /	Arithmetic
< ≤ > ≥ = !=	Comparison
! &&	Logic
if (x,y,z)	Conditional
<i>name</i> context[<i>name</i>]	Lookup

Evolving Tuples

- The evolution context is a dictionary of environmental values
- Provides access to host and application provided values

Key	Value
hour	15
month	september
year	2007
...	...

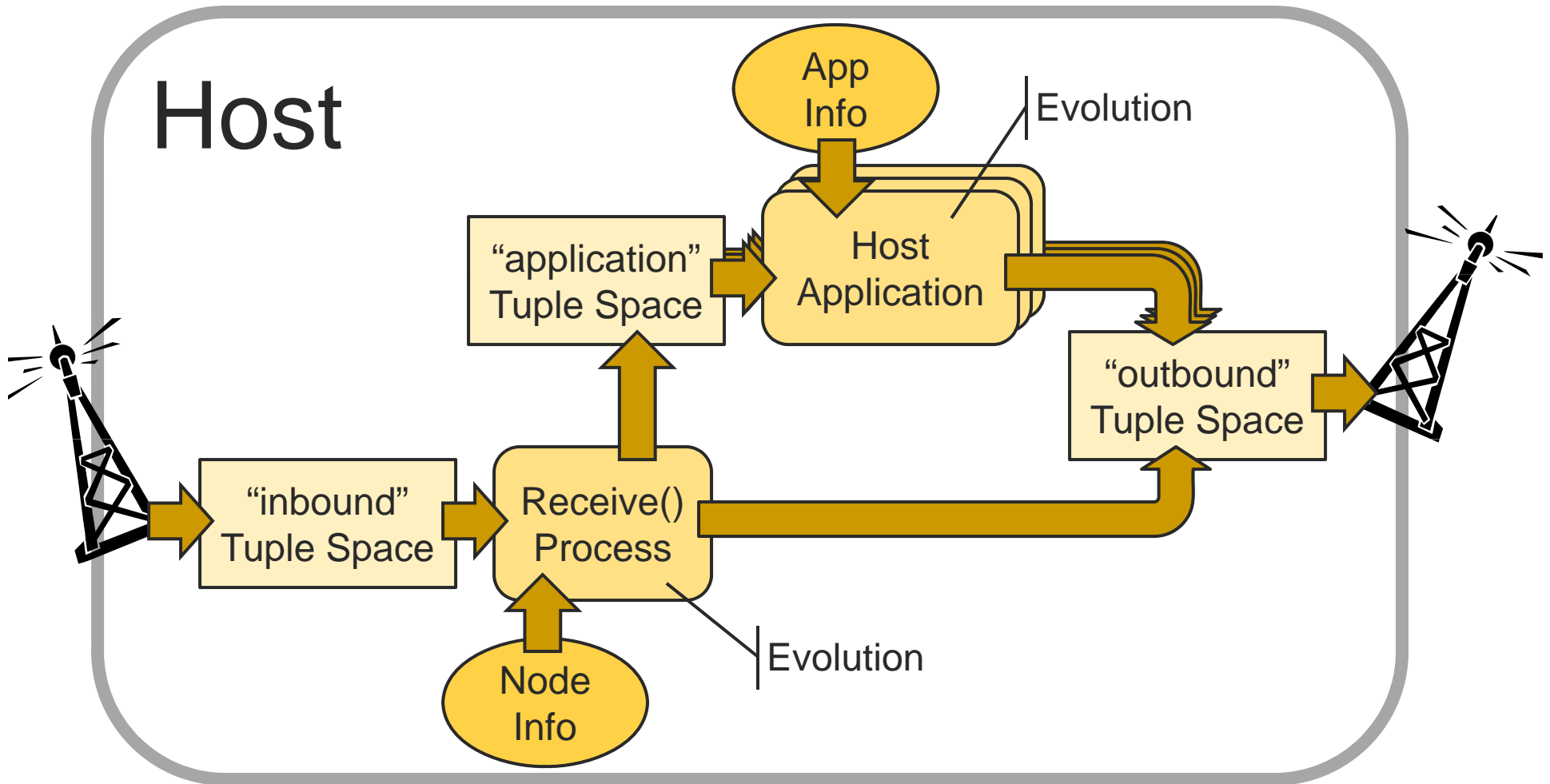
Evolving Tuples



$\langle \text{oneHourAgo}, 12, \text{context}[\text{hour}] - 1 \rangle$

$\langle \text{oneHourAgo}, 14, \text{context}[\text{hour}] - 1 \rangle$

Standard Deployment

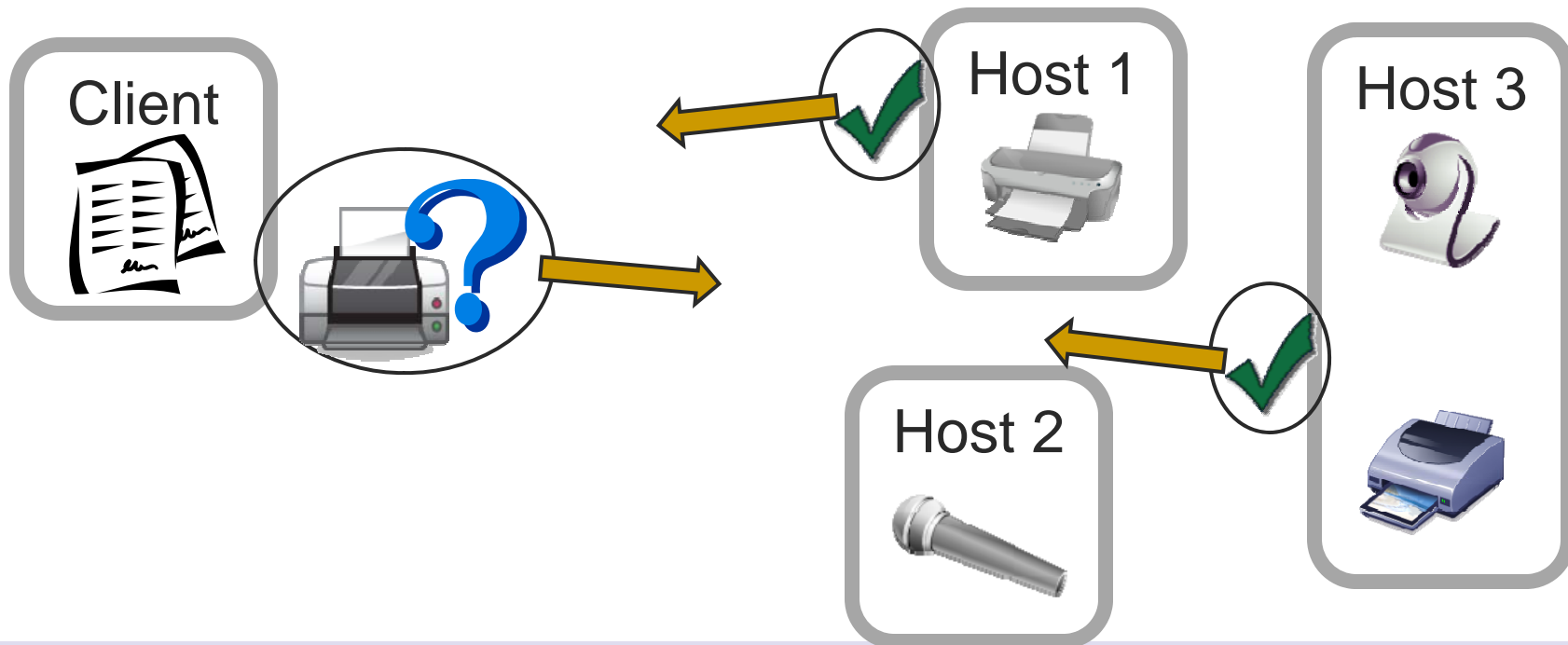


Evolving Tuples

- Dynamic values and behavior enables evolving tuples to be used for a variety of pervasive services
 - Discovery
 - Routing
 - Data dissemination
 - Data collection and aggregation
 - Remote Procedure Call
 - Instant messaging

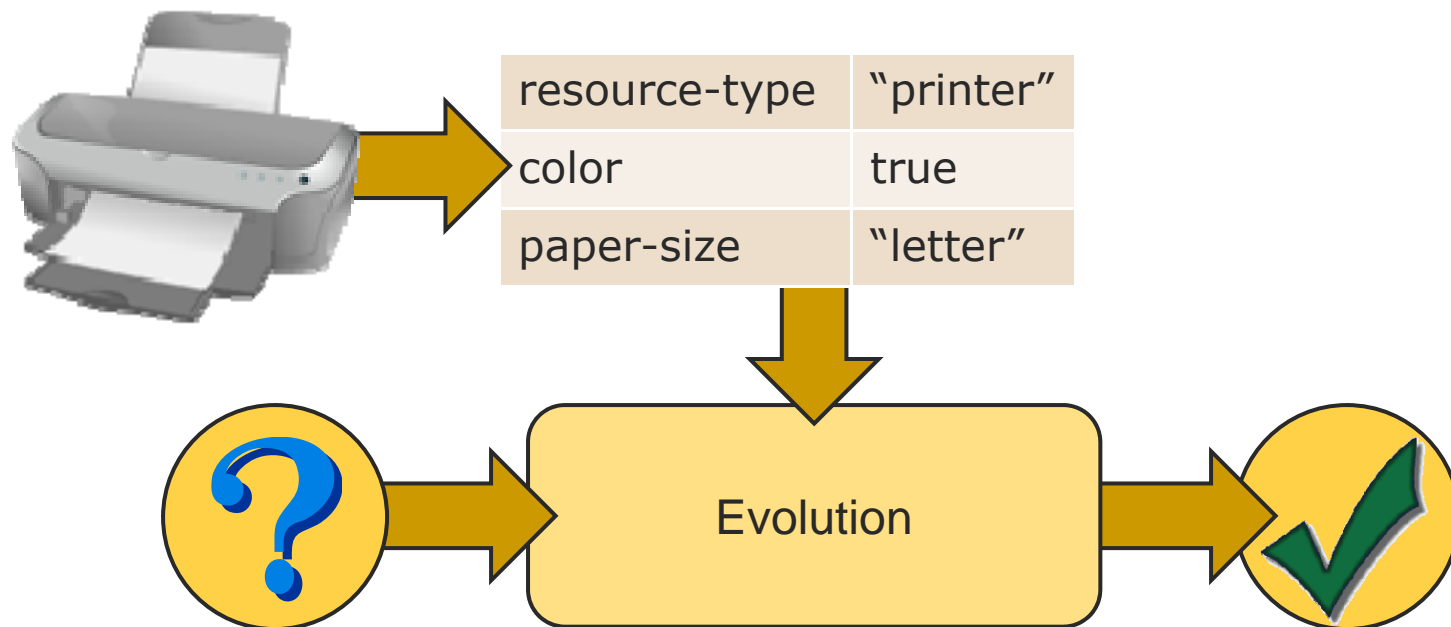
Discovery

- Finding implementations of interfaces
- One of the initial services required in pervasive environments



Discovery by Evolution

- Resource description as evolution context



Discovery by Evolution: An Example

- Looking for a “printer”
 - context[resource-type]=“printer”

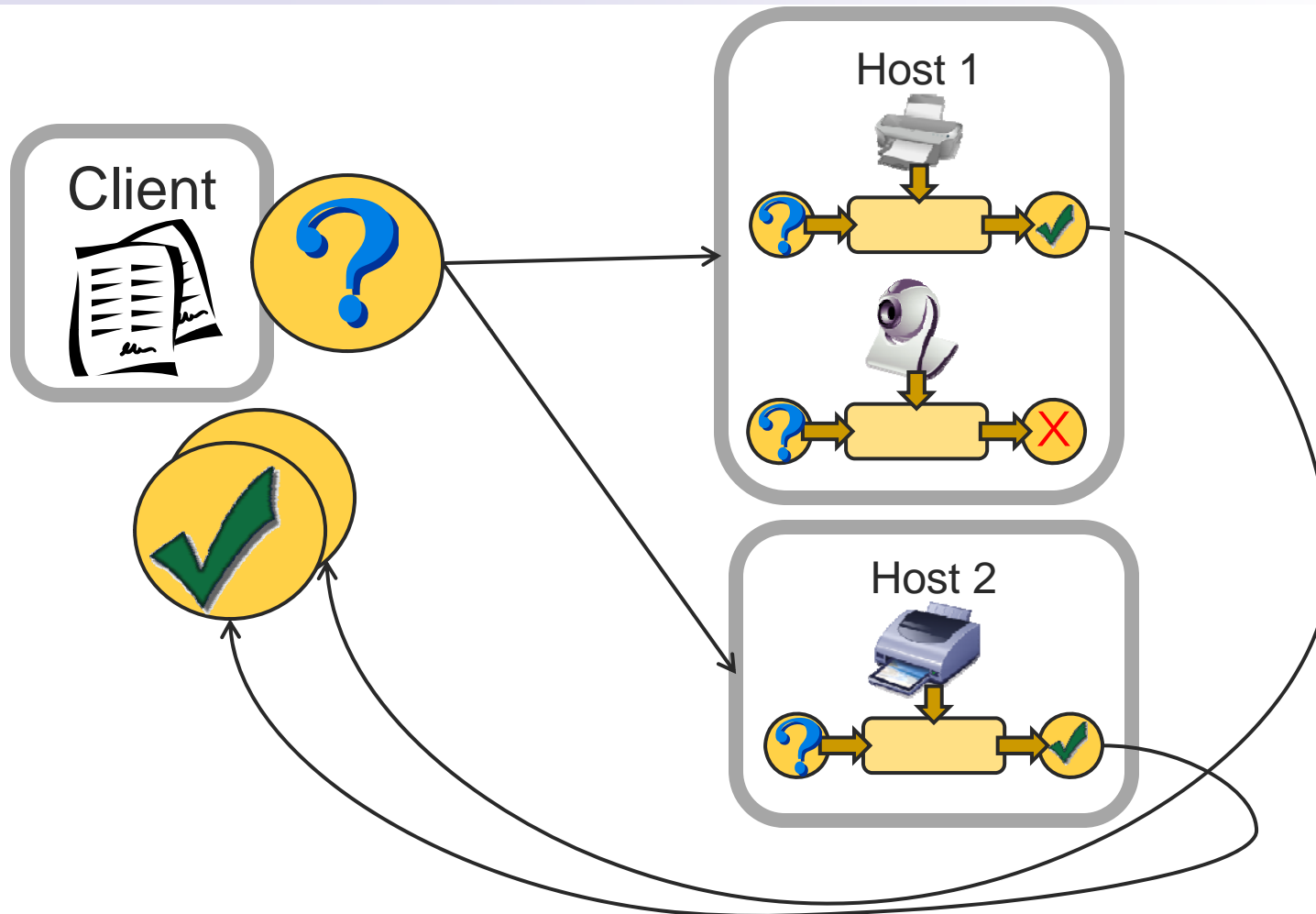
- Addresses
 - -1 = Broadcast
 - Null = No address, drop

Discovery by Evolution: An Example

source	5	
latency	0	...
match	null	... context[resource-type]="printer" ...
destination	-1	<pre>if (match != null, if(match = true, source, null), if(latency < 0.1, -1, null))</pre>

- Full tuple specification in paper

Discovery Example



Open Questions

- Performance vs. Capability trade-off
- Prototyping protocols and applications with evolving tuples?
- What is the “right” coordination model for Pervasive Services?
 - What is the Lowest Common Denominator amongst services and hosts?
 - Including sensor-class nodes
- Can (and how) do we collaborate between administrative domains?

Resource Discovery with Evolving Tuples

Drew Stovall and Christine Julien

The University of Texas at Austin

{dstovall, c.julien}@mail.utexas.edu

<http://mpc.ece.utexas.edu/evolving-tuples>