Ambient References:
Object Designation in Mobile Ad Hoc Networks

Tom Van Cutsem

Promotors:
Prof. Dr. Wolfgang De Meuter
Prof. Dr. Theo D'Hondt

Programming Technology Lab
Vrije Universiteit Brussel
Brussels, Belgium

Private PhD Defense, April 30th 2008, Brussels

Roadmap

Roadmap

Motivation

Mobile ad hoc networks

Roadmap

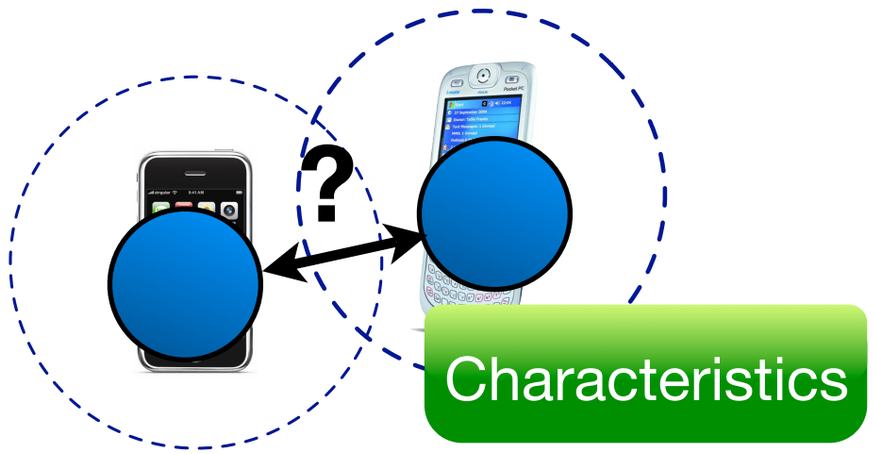
Motivation

Mobile ad hoc networks

Roadmap

Motivation

Coordination

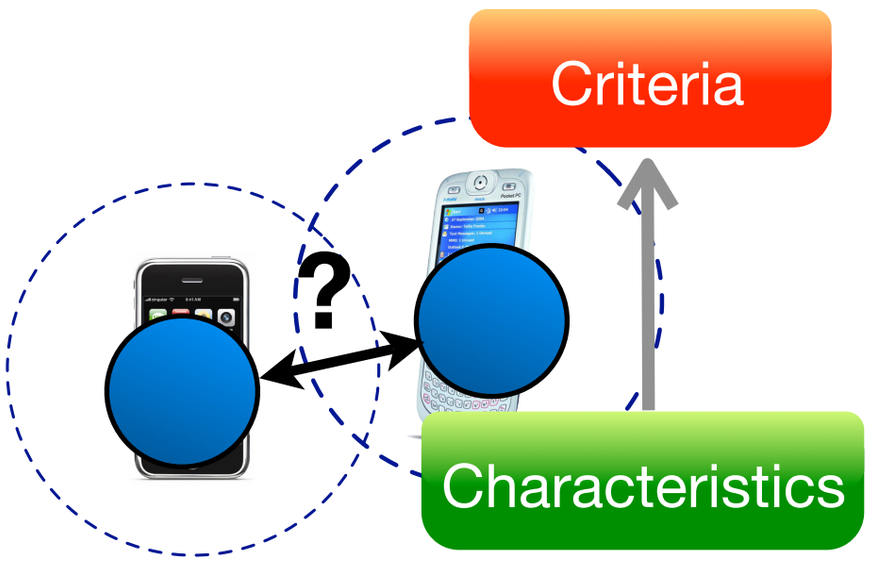


Mobile ad hoc networks

Roadmap

Motivation

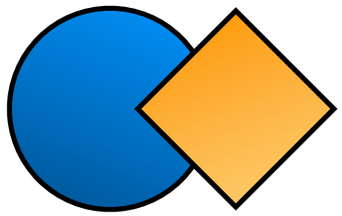
Coordination



Mobile ad hoc networks

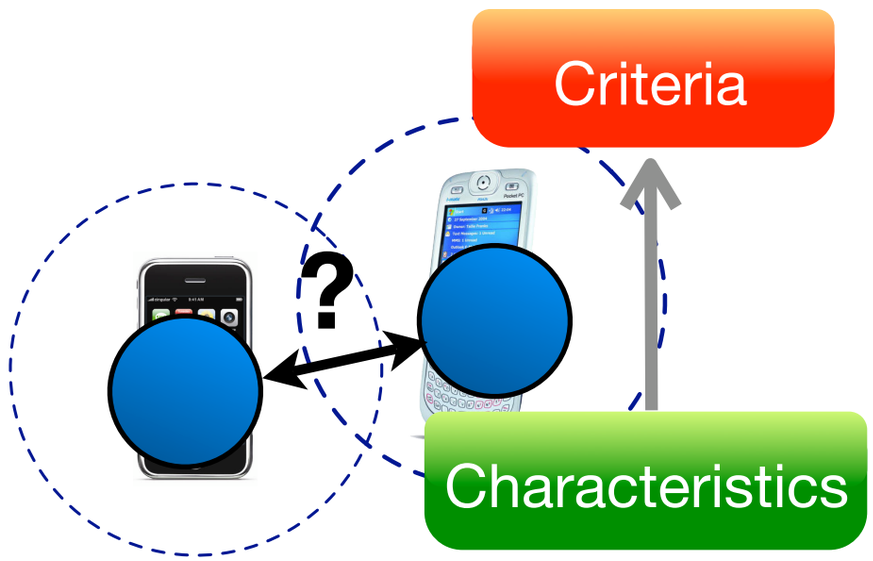
Roadmap

Motivation



Object-event impedance mismatch

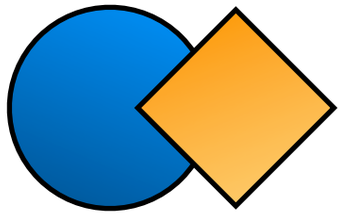
Coordination



Mobile ad hoc networks

Roadmap

Motivation

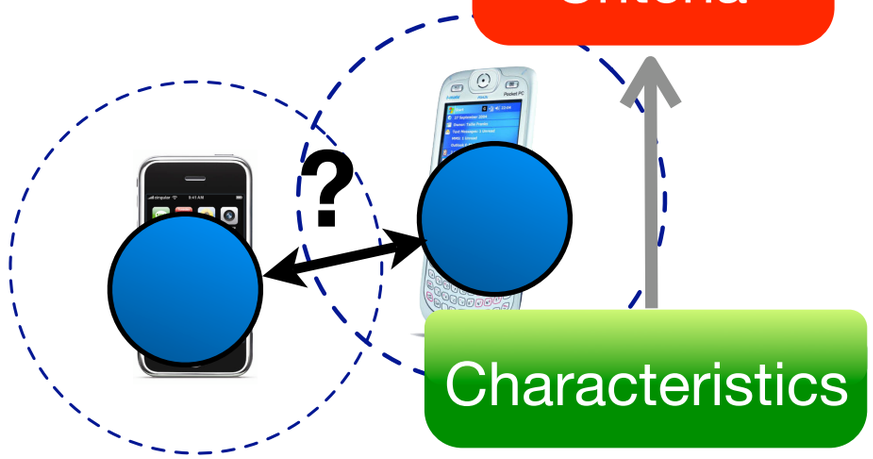


Object-event impedance mismatch

Issues

Coordination

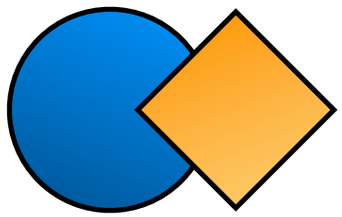
Criteria



Mobile ad hoc networks

Roadmap

Motivation

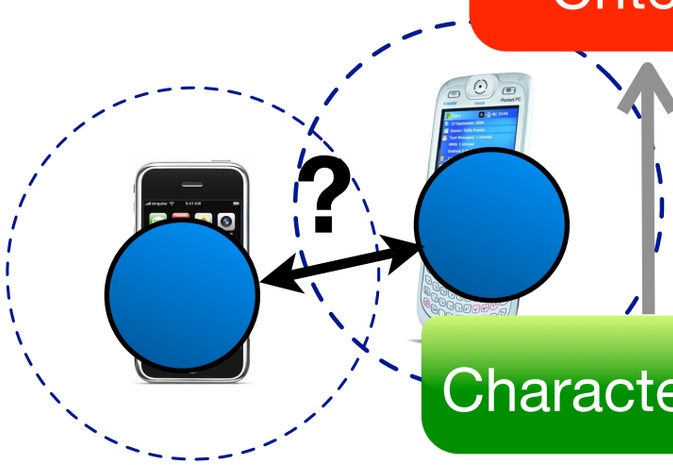


Object-event impedance mismatch

Issues

Coordination

Criteria



Characteristics

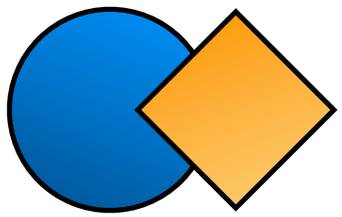
Mobile ad hoc networks

Contribution

AMBIENTTALK

Roadmap

Motivation

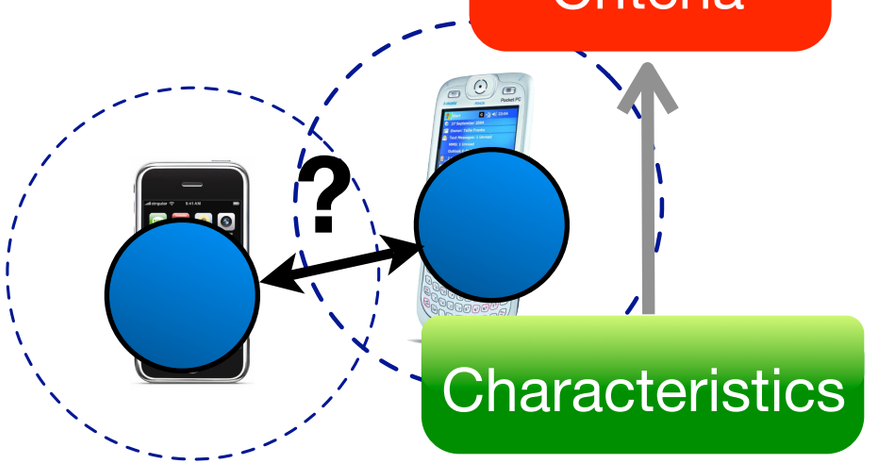


Object-event impedance mismatch

Issues

Coordination

Criteria

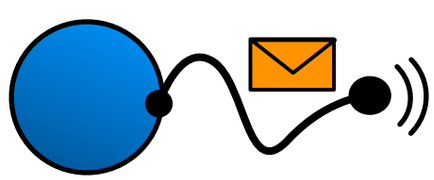


Characteristics

Mobile ad hoc networks

Contribution

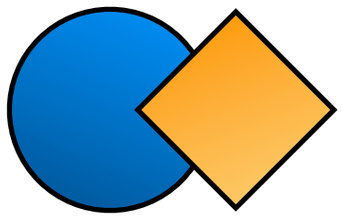
AMBIENTTALK



Ambient References

Roadmap

Motivation

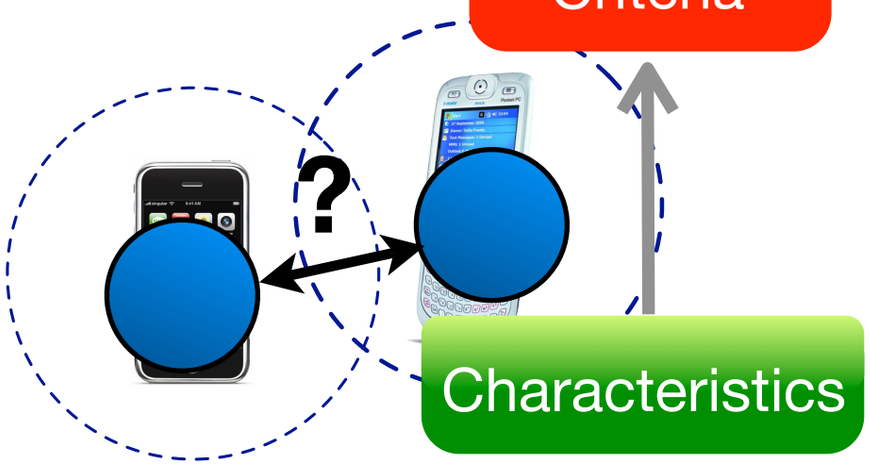


Object-event impedance mismatch

Issues

Coordination

Criteria



Characteristics

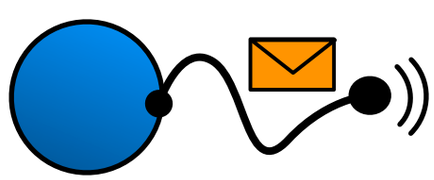
Mobile ad hoc networks

Contribution

AMBIENTTALK

Language

Language Feature



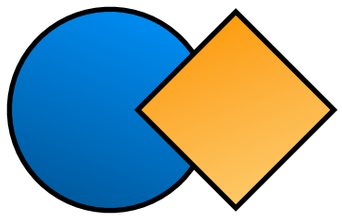
Ambient References

Validation

Evaluation

Roadmap

Motivation

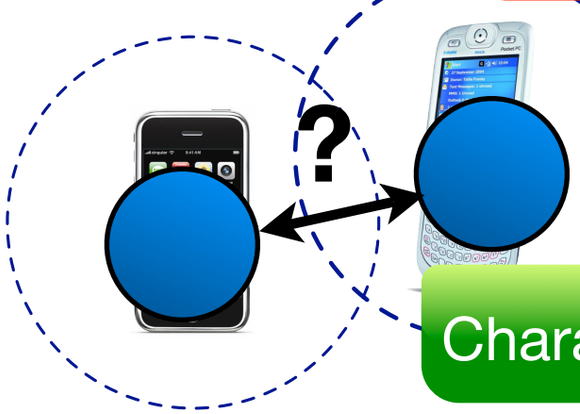


Object-event impedance mismatch

Issues

Coordination

Criteria



Characteristics

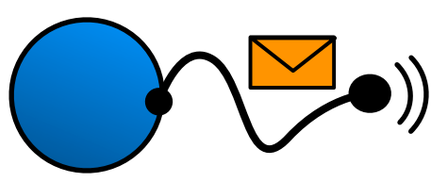
Mobile ad hoc networks

Contribution

AMBIENTTALK

Language

Language Feature



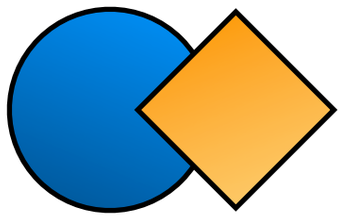
Ambient References

Validation

Evaluation

Roadmap

Motivation

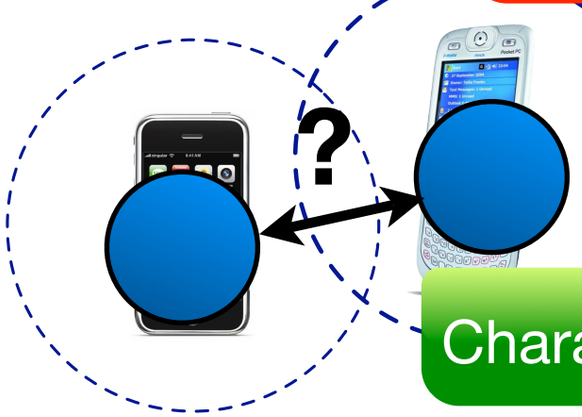


Object-event impedance mismatch

Issues

Coordination

Criteria



Characteristics

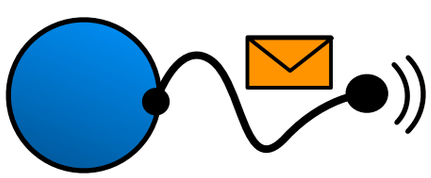
Mobile ad hoc networks

Contribution

AMBIENTTALK

Language

Language Feature



Ambient References

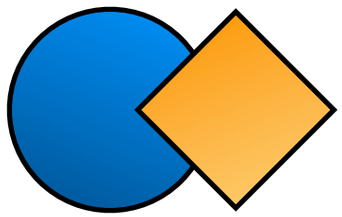
Validation

Evaluation

Implementation

Roadmap

Motivation



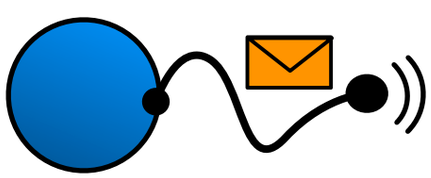
Object-event impedance mismatch

Coordination

Mobile ad hoc networks

Contribution

AMBIENTTALK



Ambient References

Validation

Evaluation

Implementation

Context & Problem Statement

Motivation

Object-event impedance mismatch

Coordination

Mobile ad hoc networks

Contribution

AMBIENTTALK

Ambient References

Validation

Evaluation

Implementation

Mobile Ad Hoc Networks

Networks composed of **mobile** devices which communicate **wirelessly**

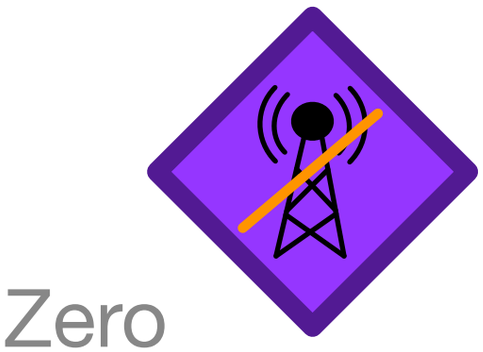
Ubiquitous Computing

Mobile Ad Hoc Networks

Networks composed of **mobile** devices which communicate **wirelessly**

Mobile Ad Hoc Networks

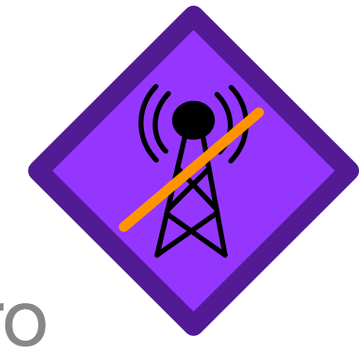
Networks composed of **mobile** devices which communicate **wirelessly**



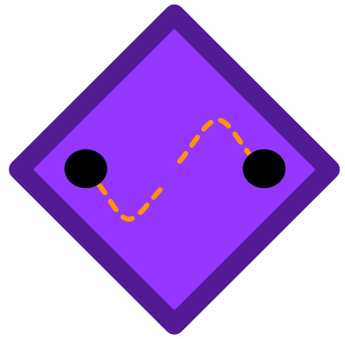
Zero
Infrastructure

Mobile Ad Hoc Networks

Networks composed of **mobile** devices which communicate **wirelessly**



Zero Infrastructure



Volatile Connections

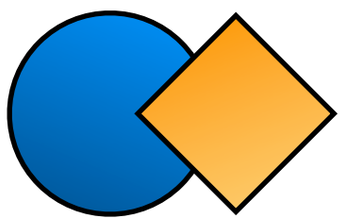
Problem Statement

- Ambient-oriented Programming [Dedecker06]: traditional OOP does not scale in MANETs. How can we:
  - **designate** objects, with zero infrastructure?
  - **communicate** with objects, across volatile connections?
- Answer: event-based communication
- But: such communication is **fundamentally** not object-oriented

Problem Statement

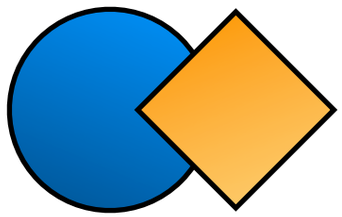
- Ambient-oriented Programming [Dedecker06]: traditional OOP does not scale in MANETs. How can we:
  - **designate** objects, with zero infrastructure?
  - **communicate** with objects, across volatile connections?
- Answer: event-based communication
- But: such communication is **fundamentally** not object-oriented

object-event impedance mismatch



Roadmap

Motivation



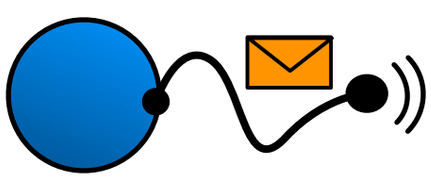
Object-event impedance mismatch

Coordination

Mobile ad hoc networks

Contribution

AMBIENTTALK



Ambient References

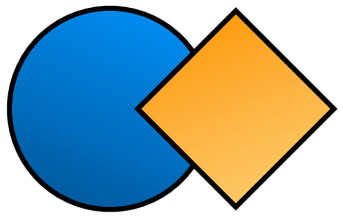
Validation

Evaluation

Implementation

Coordination in Mobile ad hoc Networks

Motivation



Object-event impedance mismatch

Issues

Coordination

Criteria

Characteristics

Mobile ad hoc networks

Contribution

AMBIENTTALK

Ambient References

Validation

Evaluation

Implementation

Coordination in Mobile Ad Hoc Networks

uMaMa

Coordination in Mobile Ad Hoc Networks

uMaMa

Discovery

Coordination in Mobile Ad Hoc Networks

uMaMa

Discovery

Communication

Coordination in Mobile Ad Hoc Networks

uMaMa

Discovery Communication Synchronisation

Coordination in Mobile Ad Hoc Networks

uMaMa

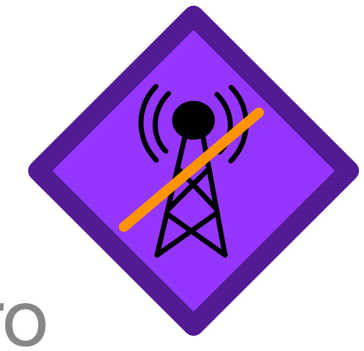
Discovery Communication Synchronisation

Coordination in Mobile Ad Hoc Networks

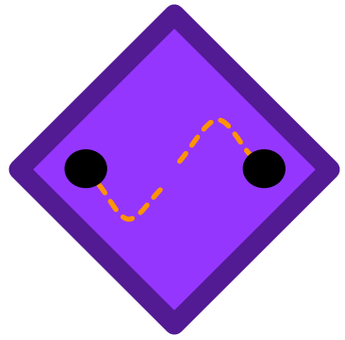
uMaMa

Discovery Communication Synchronisation Failure handling

Criteria for Coordination in MANETs

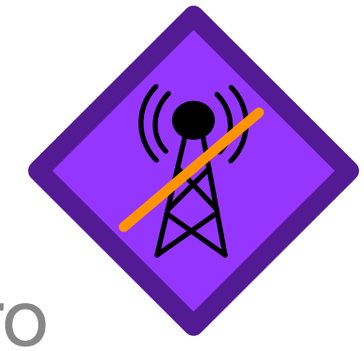


Zero Infrastructure

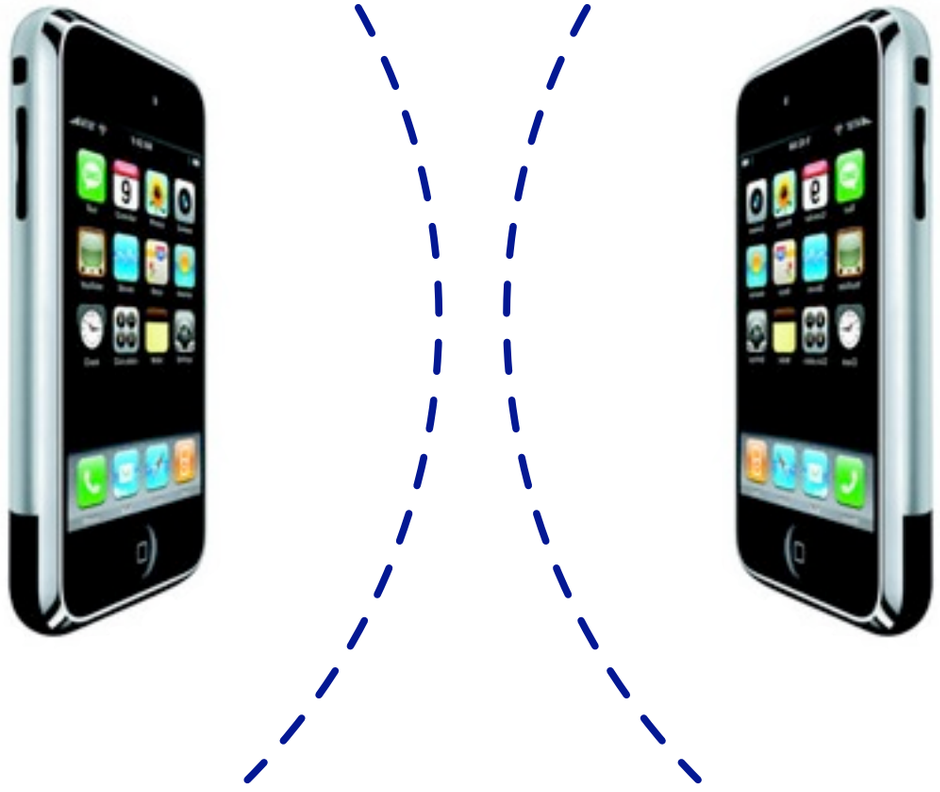


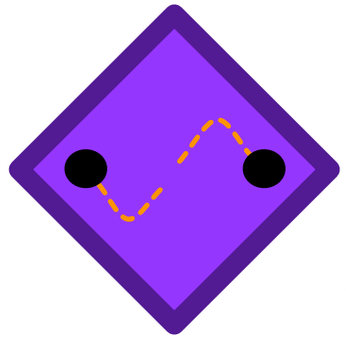
Volatile Connections

Criteria for Coordination in MANETs



Zero
Infrastructure

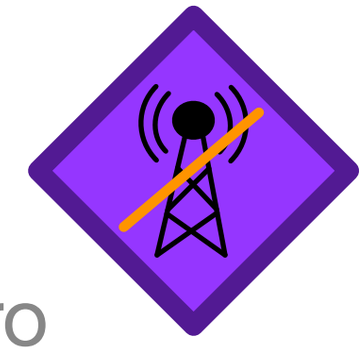




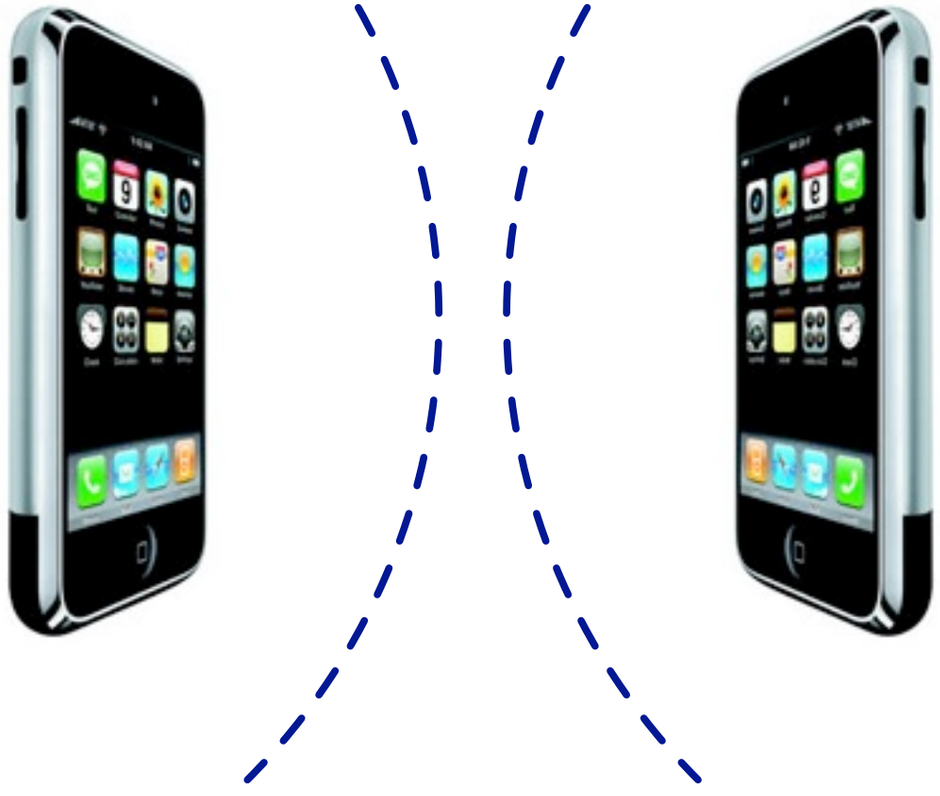
Volatile
Connections

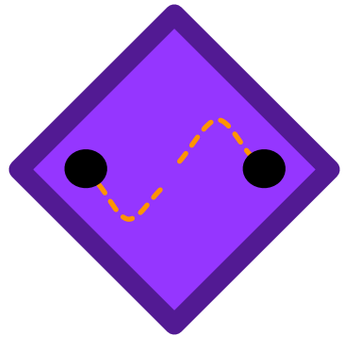
Decentralised Discovery

Criteria for Coordination in MANETs



Zero
Infrastructure



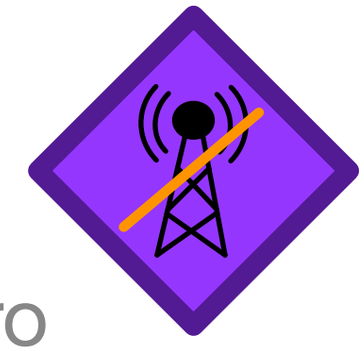


Volatile
Connections

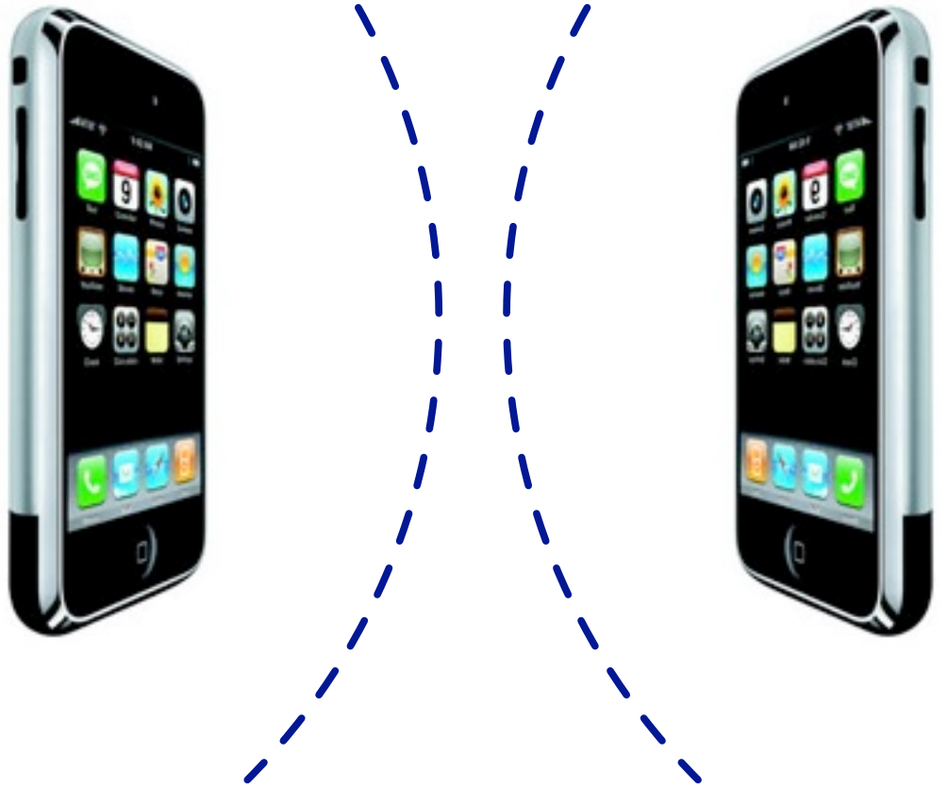
Decentralised Discovery

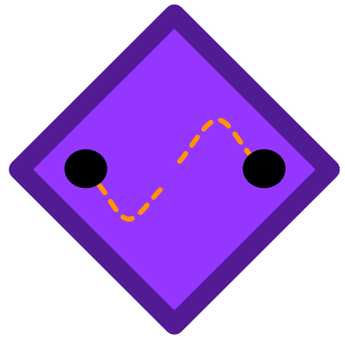
Space & arity decoupled
communication

Criteria for Coordination in MANETs



Zero Infrastructure



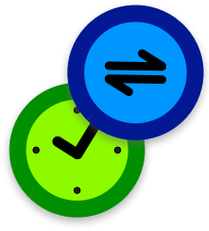


Volatile Connections

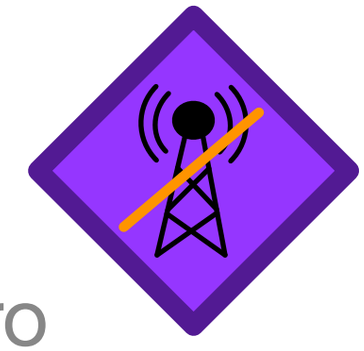
Decentralised Discovery

Space & arity decoupled communication

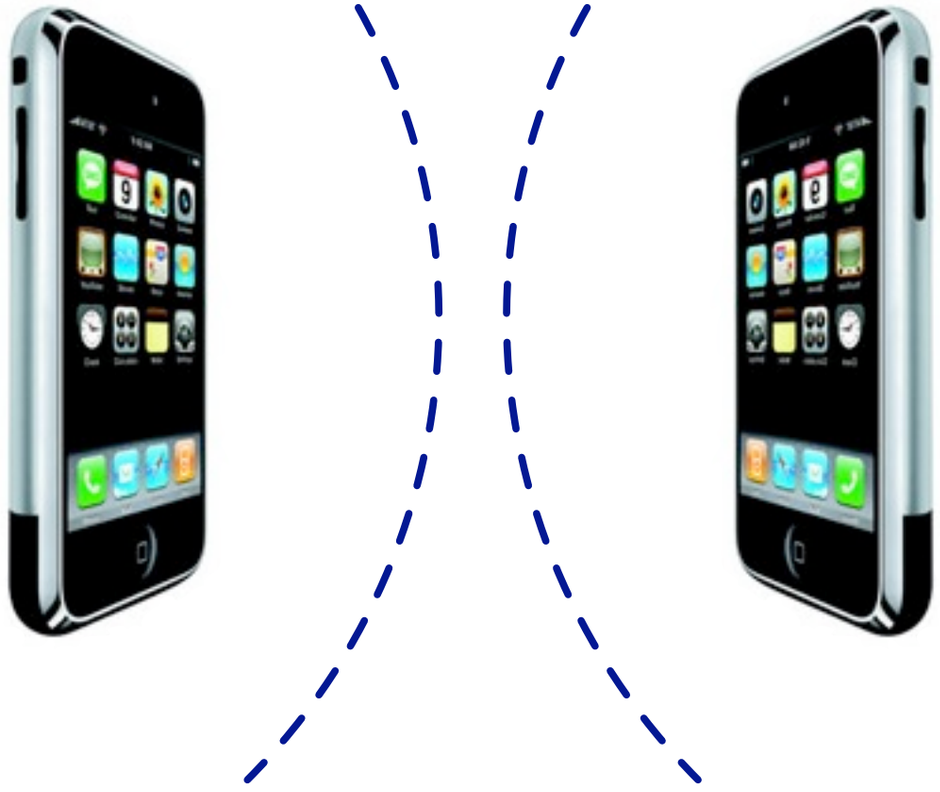
Time & synchronisation decoupled communication

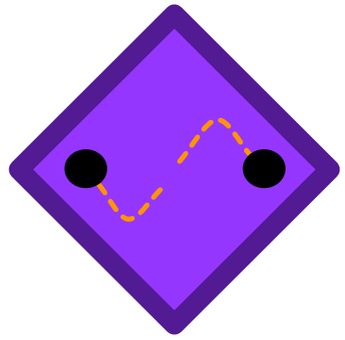


Criteria for Coordination in MANETs



Zero Infrastructure





Volatile Connections

Decentralised Discovery

Time & synchronisation decoupled communication

Space & arity decoupled communication

Connection-independent Failure handling

Survey of Related Work

	Decoupling in				Failure	Decentr.	Message
	Time	Space	Synchronisation	Arity	Handling	Discovery	Passing
Languages for Local Area Networks							
Emerald	N (Error)	N (Address)	N (RPC)	N	Exc.	N	Y
Obliq	N (Error)	N (Address)	N (RPC)	N	Exc.	N	Y
ABCL	Y (Buffer)	N (Address)	Sender only	N	N	N	Y
Languages for Wide Area Networks							
Erlang	Y (Buffer)	N (Address)	Sender only	N	React	N	Y
Argus	N (Error)	N (Address)	N(RPC)	N	Exc.	N	Y
Janus	Y (Blocks)	N (Address)	N (Logic Var)	N	N	N	N
Salsa	Y (Buffer)	N (Address)	Y (Async Msg)	N	N	N	Y
E	N (Error)	N (Address)	Y (Async Msg)	N	React	N	Y
Distributed Oz	Y (Blocks)	N (Address)	N (Logic Var)	N	React	N	Y
Languages for Wireless Sensor Networks							
ActorNet	Y (Buffer)	N (Address)	Y (Async Msg)	Y	N	N	Y
SpatialViews	Y (Mediator)	Y (Discovery)	Y (Mediator)	Y	N	N	N
Models and Calculi for Wide Area Networks							
Actors	Y (Buffer)	N (URI)	Y (Async Msg)	N	N	N	Y
ActorSpace	Y (Buffer)	Y (Mediator)	Y (Async Msg)	Y	N	N	Y
Mobile Ambients	Y (Mediator)	Y (Mediator)	Sender only	N	N	N	N
Tuple Space Middleware for Ad Hoc Networks							
LIME	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	React	Y	N
TOTA	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	React	Y	N
MARS	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	Lease	N	N
Middleware for Nomadic Networks							
Rover	Y (Buffer)	N (Address)	Y (Async Msg)	N	React	N	Y
JINI	N (Error)	Y (Discovery)	N (RPC)	N	Lease	Y	Y
Publish-Subscribe Middleware for Ad Hoc Networks							
EMMA	N (Lost)	Y (Mediator)	Y (Mediator)	Y	Lease	Y	N
LPS	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	N	N	N
STEAM	N (Lost)	Y (Mediator)	Y (Mediator)	Y	N	N	N
M2MI	N (Lost)	Y (Broadcast)	Y (Async Msg)	Y	N	N	Y
one.world	N (Lost)	Y (Discovery)	Y (Mediator)	Y	Lease	Y	N

Survey of Related Work

	Decoupling in				Failure Handling	Decentr. Discovery	Message Passing
	Time	Space	Synchronisation	Arity			
Languages for Local Area Networks							
Emerald	N (Error)	N (Address)	N (RPC)	N	Exc.	N	Y
Obliq	N (Error)	N (Address)	N (RPC)	N	Exc.	N	Y
ABCL	Y (Buffer)	N (Address)	Sender only	N	N	N	Y
Languages for Wide Area Networks							
Erlang	Y (Buffer)	N (Address)	Sender only	N	React	N	Y
Argus	N (Error)	N (Address)	N(RPC)	N	Exc.	N	Y
Janus	Y (Blocks)	N (Address)	N (Logic Var)	N	N	N	N
Salsa	Y (Buffer)	N (Address)	Y (Async Msg)	N	N	N	Y
E	N (Error)	N (Address)	Y (Async Msg)	N	React	N	Y
Distributed Oz	Y (Blocks)	N (Address)	N (Logic Var)	N	React	N	Y
Languages for Wireless Sensor Networks							
ActorNet	Y (Buffer)	N (Address)	Y (Async Msg)	Y	N	N	Y
SpatialViews	Y (Mediator)	Y (Discovery)	Y (Mediator)	Y	N	N	N
Models and Calculi for Wide Area Networks							
Actors	Y (Buffer)	N (URI)	Y (Async Msg)	N	N	N	Y
ActorSpace	Y (Buffer)	Y (Mediator)	Y (Async Msg)	Y	N	N	Y
Mobile Ambients	Y (Mediator)	Y (Mediator)	Sender only	N	N	N	N
Tuple Space Middleware for Ad Hoc Networks							
LIME	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	React	Y	N
TOTA	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	React	Y	N
MARS	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	Lease	N	N
Middleware for Nomadic Networks							
Rover	Y (Buffer)	N (Address)	Y (Async Msg)	N	React	N	Y
JINI	N (Error)	Y (Discovery)	N (RPC)	N	Lease	Y	Y
Publish-Subscribe Middleware for Ad Hoc Networks							
EMMA	N (Lost)	Y (Mediator)	Y (Mediator)	Y	Lease	Y	N
LPS	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	N	N	N
STEAM	N (Lost)	Y (Mediator)	Y (Mediator)	Y	N	N	N
M2MI	N (Lost)	Y (Broadcast)	Y (Async Msg)	Y	N	N	Y
one.world	N (Lost)	Y (Discovery)	Y (Mediator)	Y	Lease	Y	N

Survey of Related Work

	Decoupling in				Failure Handling	Decentr. Discovery	Message Passing
	Time	Space	Synchronisation	Arity			
Languages for Local Area Networks							
Emerald	N (Error)	N (Address)	N (RPC)	N	Exc.	N	Y
Obliq	N (Error)	N (Address)	N (RPC)	N	Exc.	N	Y
ABCL	Y (Buffer)	N (Address)	Sender only	N	N	N	Y
Languages for Wide Area Networks							
Erlang	Y (Buffer)	N (Address)	Sender only	N	React	N	Y
Argus	N (Error)	N (Address)	N(RPC)	N	Exc.	N	Y
Janus	Y (Blocks)	N (Address)	N (Logic Var)	N	N	N	N
Salsa	Y (Buffer)	N (Address)	Y (Async Msg)	N	N	N	Y
E	N (Error)	N (Address)	Y (Async Msg)	N	React	N	Y
Distributed Oz	Y (Blocks)	N (Address)	N (Logic Var)	N	React	N	Y
Languages for Wireless Sensor Networks							
ActorNet	Y (Buffer)	N (Address)	Y (Async Msg)	Y	N	N	Y
SpatialViews	Y (Mediator)	Y (Discovery)	Y (Mediator)	Y	N	N	N
Models and Calculi for Wide Area Networks							
Actors	Y (Buffer)	N (URI)	Y (Async Msg)	N	N	N	Y
ActorSpace	Y (Buffer)	Y (Mediator)	Y (Async Msg)	Y	N	N	Y
Mobile Ambients	Y (Mediator)	Y (Mediator)	Sender only	N	N	N	N
Tuple Space Middleware for Ad Hoc Networks							
LIME	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	React	Y	N
TOTA	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	React	Y	N
MARS	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	Lease	N	N
Middleware for Nomadic Networks							
Rover	Y (Buffer)	N (Address)	Y (Async Msg)	N	React	N	Y
JINI	N (Error)	Y (Discovery)	N (RPC)	N	Lease	Y	Y
Publish-Subscribe Middleware for Ad Hoc Networks							
EMMA	N (Lost)	Y (Mediator)	Y (Mediator)	Y	Lease	Y	N
LPS	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	N	N	N
STEAM	N (Lost)	Y (Mediator)	Y (Mediator)	Y	N	N	N
M2MI	N (Lost)	Y (Broadcast)	Y (Async Msg)	Y	N	N	Y
one.world	N (Lost)	Y (Discovery)	Y (Mediator)	Y	Lease	Y	N

Survey of Related Work

	Decoupling in				Failure	Decentr.	Message
	Time	Space	Synchronisation	Arity	Handling	Discovery	Passing
Languages for Local Area Networks							
Emerald	N (Error)	N (Address)	N (RPC)	N	Exc.	N	Y
Obliq	N (Error)	N (Address)	N (RPC)	N	Exc.	N	Y
ABCL	Y (Buffer)	N (Address)	Sender only	N	N	N	Y
Languages for Wide Area Networks							
Erlang	Y (Buffer)	N (Address)	Sender only	N	React	N	Y
Argus	N (Error)	N (Address)	N(RPC)	N	Exc.	N	Y
Janus	Y (Blocks)	N (Address)	N (Logic Var)	N	N	N	N
Salsa	Y (Buffer)	N (Address)	Y (Async Msg)	N	N	N	Y
E	N (Error)	N (Address)	Y (Async Msg)	N	React	N	Y
Distributed Oz	Y (Blocks)	N (Address)	N (Logic Var)	N	React	N	Y
Languages for Wireless Sensor Networks							
ActorNet	Y (Buffer)	N (Address)	Y (Async Msg)	Y	N	N	Y
SpatialViews	Y (Mediator)	Y (Discovery)	Y (Mediator)	Y	N	N	N
Models and Calculi for Wide Area Networks							
Actors	Y (Buffer)	N (URI)	Y (Async Msg)	N	N	N	Y
ActorSpace	Y (Buffer)	Y (Mediator)	Y (Async Msg)	Y	N	N	Y
Mobile Ambients	Y (Mediator)	Y (Mediator)	Sender only	N	N	N	N
Tuple Space Middleware for Ad Hoc Networks							
LIME	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	React	Y	N
TOTA	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	React	Y	N
MARS	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	Lease	N	N
Middleware for Nomadic Networks							
Rover	Y (Buffer)	N (Address)	Y (Async Msg)	N	React	N	Y
JINI	N (Error)	Y (Discovery)	N (RPC)	N	Lease	Y	Y
Publish-Subscribe Middleware for Ad Hoc Networks							
EMMA	N (Lost)	Y (Mediator)	Y (Mediator)	Y	Lease	Y	N
LPS	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	N	N	N
STEAM	N (Lost)	Y (Mediator)	Y (Mediator)	Y	N	N	N
M2MI	N (Lost)	Y (Broadcast)	Y (Async Msg)	Y	N	N	Y
one.world	N (Lost)	Y (Discovery)	Y (Mediator)	Y	Lease	Y	N

OO Message Passing metaphor => bad decoupling

Survey of Related Work

	Decoupling in				Failure Handling	Decentr. Discovery	Message Passing
	Time	Space	Synchronisation	Arity			
Languages for Local Area Networks							
Emerald	N (Error)	N (Address)	N (RPC)	N	Exc.	N	Y
Obliq	N (Error)	N (Address)	N (RPC)	N	Exc.	N	Y
ABCL	Y (Buffer)	N (Address)	Sender only	N	N	N	Y
Languages for Wide Area Networks							
Erlang	Y (Buffer)	N (Address)	Sender only	N	React	N	Y
Argus	N (Error)	N (Address)	N(RPC)	N	Exc.	N	Y
Janus	Y (Blocks)	N (Address)	N (Logic Var)	N	N	N	N
Salsa	Y (Buffer)	N (Address)	Y (Async Msg)	N	N	N	Y
E	N (Error)	N (Address)	Y (Async Msg)	N	React	N	Y
Distributed Oz	Y (Blocks)	N (Address)	N (Logic Var)	N	React	N	Y
Languages for Wireless Sensor Networks							
ActorNet	Y (Buffer)	N (Address)	Y (Async Msg)	Y	N	N	Y
SpatialViews	Y (Mediator)	Y (Discovery)	Y (Mediator)	Y	N	N	N
Models and Calculi for Wide Area Networks							
Actors	Y (Buffer)	N (URI)	Y (Async Msg)	N	N	N	Y
ActorSpace	Y (Buffer)	Y (Mediator)	Y (Async Msg)	Y	N	N	Y
Mobile Ambients	Y (Mediator)	Y (Mediator)	Sender only	N	N	N	N
Tuple Space Middleware for Ad Hoc Networks							
LIME	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	React	Y	N
TOTA	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	React	Y	N
MARS	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	Lease	N	N
Middleware for Nomadic Networks							
Rover	Y (Buffer)	N (Address)	Y (Async Msg)	N	React	N	Y
JINI	N (Error)	Y (Discovery)	N (RPC)	N	Lease	Y	Y
Publish-Subscribe Middleware for Ad Hoc Networks							
EMMA	N (Lost)	Y (Mediator)	Y (Mediator)	Y	Lease	Y	N
LPS	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	N	N	N
STEAM	N (Lost)	Y (Mediator)	Y (Mediator)	Y	N	N	N
M2MI	N (Lost)	Y (Broadcast)	Y (Async Msg)	Y	N	N	Y
one.world	N (Lost)	Y (Discovery)	Y (Mediator)	Y	Lease	Y	N

OO Message Passing metaphor => bad decoupling

Survey of Related Work

	Decoupling in				Failure Handling	Decentr. Discovery	Message Passing
	Time	Space	Synchronisation	Arity			
Languages for Local Area Networks							
Emerald	N (Error)	N (Address)	N (RPC)	N	Exc.	N	Y
Obliq	N (Error)	N (Address)	N (RPC)	N	Exc.	N	Y
ABCL	Y (Buffer)	N (Address)	Sender only	N	N	N	Y
Languages for Wide Area Networks							
Erlang	Y (Buffer)	N (Address)	Sender only	N	React	N	Y
Argus	N (Error)	N (Address)	N(RPC)	N	Exc.	N	Y
Janus	Y (Blocks)	N (Address)	N (Logic Var)	N	N	N	N
Salsa	Y (Buffer)	N (Address)	Y (Async Msg)	N	N	N	Y
E	N (Error)	N (Address)	Y (Async Msg)	N	React	N	Y
Distributed Oz	Y (Blocks)	N (Address)	N (Logic Var)	N	React	N	Y
Languages for Wireless Sensor Networks							
ActorNet	Y (Buffer)	N (Address)	Y (Async Msg)	Y	N	N	Y
SpatialViews	Y (Mediator)	Y (Discovery)	Y (Mediator)	Y	N	N	N
Models and Calculi for Wide Area Networks							
Actors	Y (Buffer)	N (URI)	Y (Async Msg)	N	N	N	Y
ActorSpace	Y (Buffer)	Y (Mediator)	Y (Async Msg)	Y	N	N	Y
Mobile Ambients	Y (Mediator)	Y (Mediator)	Sender only	N	N	N	N
Tuple Space Middleware for Ad Hoc Networks							
LIME	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	React	Y	N
TOTA	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	React	Y	N
MARS	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	Lease	N	N
Middleware for Nomadic Networks							
Rover	Y (Buffer)	N (Address)	Y (Async Msg)	N	React	N	Y
JINI	N (Error)	Y (Discovery)	N (RPC)	N	Lease	Y	Y
Publish-Subscribe Middleware for Ad Hoc Networks							
EMMA	N (Lost)	Y (Mediator)	Y (Mediator)	Y	Lease	Y	N
LPS	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	N	N	N
STEAM	N (Lost)	Y (Mediator)	Y (Mediator)	Y	N	N	N
M2MI	N (Lost)	Y (Broadcast)	Y (Async Msg)	Y	N	N	Y
one.world	N (Lost)	Y (Discovery)	Y (Mediator)	Y	Lease	Y	N

OO Message Passing metaphor => bad decoupling

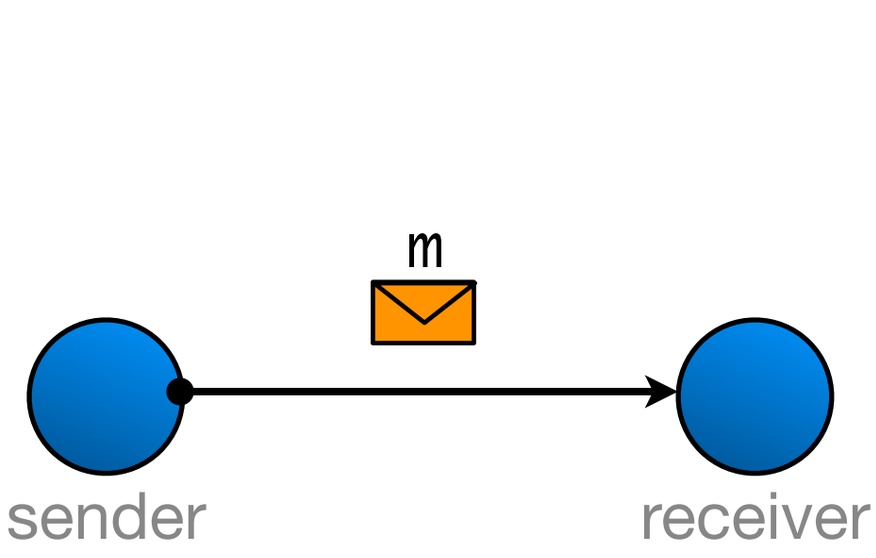
Survey of Related Work

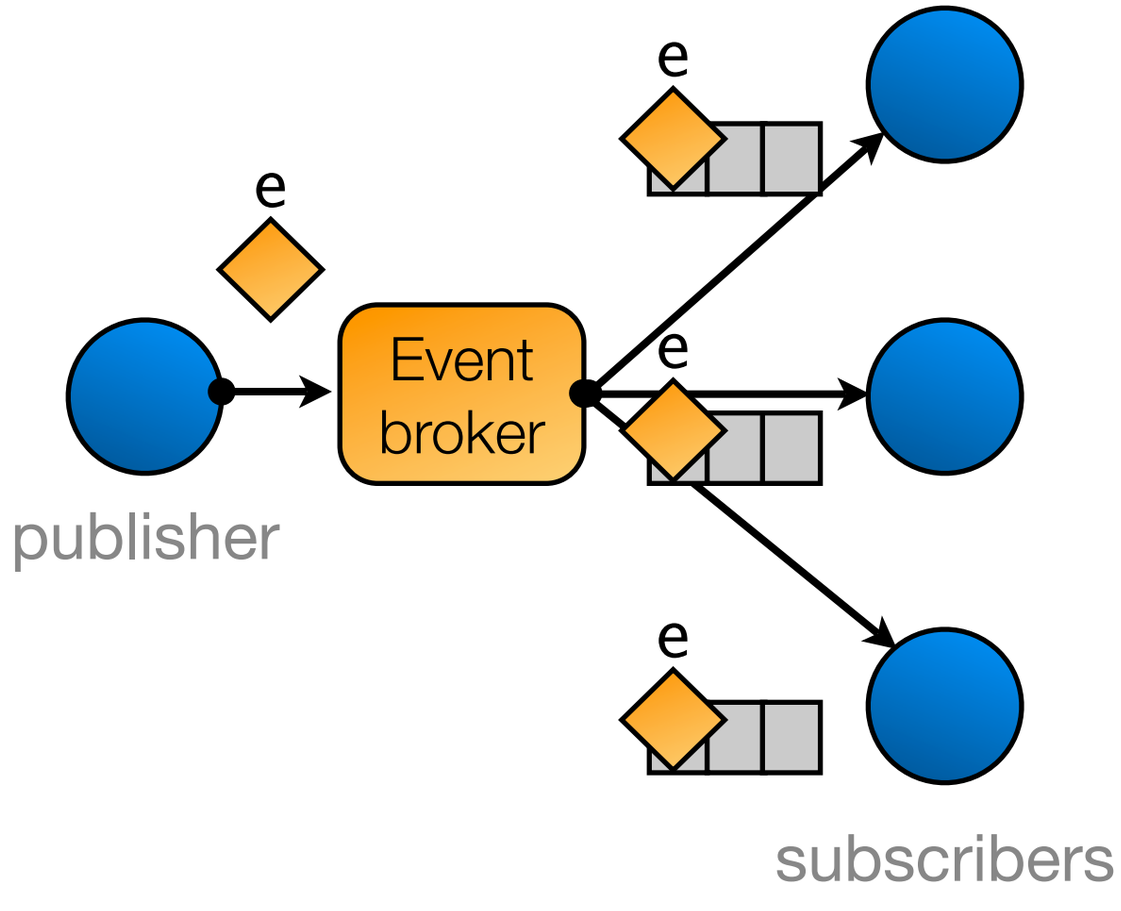
	Decoupling in				Failure Handling	Decentr. Discovery	Message Passing
	Time	Space	Synchronisation	Arity			
Languages for Local Area Networks							
Emerald	N (Error)	N (Address)	N (RPC)	N	Exc.	N	Y
Obliq	N (Error)	N (Address)	N (RPC)	N	Exc.	N	Y
ABCL	Y (Buffer)	N (Address)	Sender only	N	N	N	Y
Languages for Wide Area Networks							
Erlang	Y (Buffer)	N (Address)	Sender only	N	React	N	Y
Argus	N (Error)	N (Address)	N(RPC)	N	Exc.	N	Y
Janus	Y (Blocks)	N (Address)	N (Logic Var)	N	N	N	N
Salsa	Y (Buffer)	N (Address)	Y (Async Msg)	N	N	N	Y
E	N (Error)	N (Address)	Y (Async Msg)	N	React	N	Y
Distributed Oz	Y (Blocks)	N (Address)	N (Logic Var)	N	React	N	Y
Languages for Wireless Sensor Networks							
ActorNet	Y (Buffer)	N (Address)	Y (Async Msg)	Y	N	N	Y
SpatialViews	Y (Mediator)	Y (Discovery)	Y (Mediator)	Y	N	N	N
Models and Calculi for Wide Area Networks							
Actors	Y (Buffer)	N (URI)	Y (Async Msg)	N	N	N	Y
ActorSpace	Y (Buffer)	Y (Mediator)	Y (Async Msg)	Y	N	N	Y
Mobile Ambients	Y (Mediator)	Y (Mediator)	Sender only	N	N	N	N
Tuple Space Middleware for Ad Hoc Networks							
LIME	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	React	Y	N
TOTA	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	React	Y	N
MARS	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	Lease	N	N
Middleware for Nomadic Networks							
Rover	Y (Buffer)	N (Address)	Y (Async Msg)	N	React	N	Y
JINI	N (Error)	Y (Discovery)	N (RPC)	N	Lease	Y	Y
Publish-Subscribe Middleware for Ad Hoc Networks							
EMMA	N (Lost)	Y (Mediator)	Y (Mediator)	Y	Lease	Y	N
LPS	Y (Mediator)	Y (Mediator)	Y (Mediator)	Y	N	N	N
STEAM	N (Lost)	Y (Mediator)	Y (Mediator)	Y	N	N	N
M2MI	N (Lost)	Y (Broadcast)	Y (Async Msg)	Y	N	N	Y
one.world	N (Lost)	Y (Discovery)	Y (Mediator)	Y	Lease	Y	N

OO Message Passing metaphor => bad decoupling

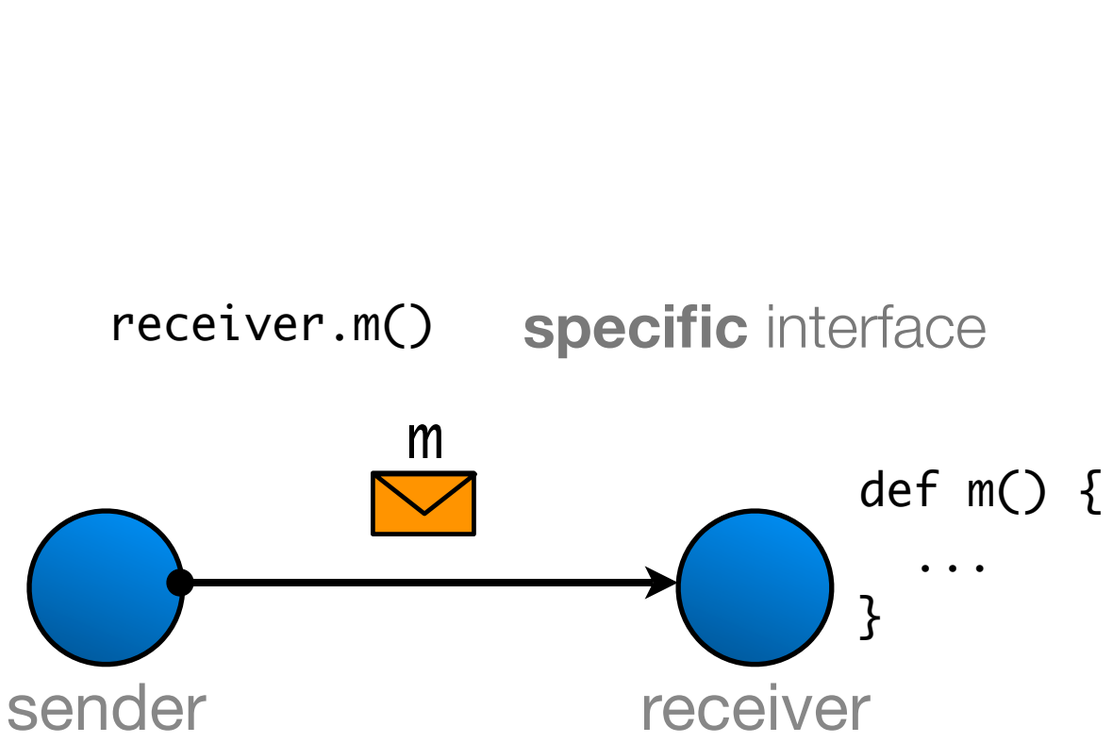
Pub/sub & Tuple spaces => better decoupling

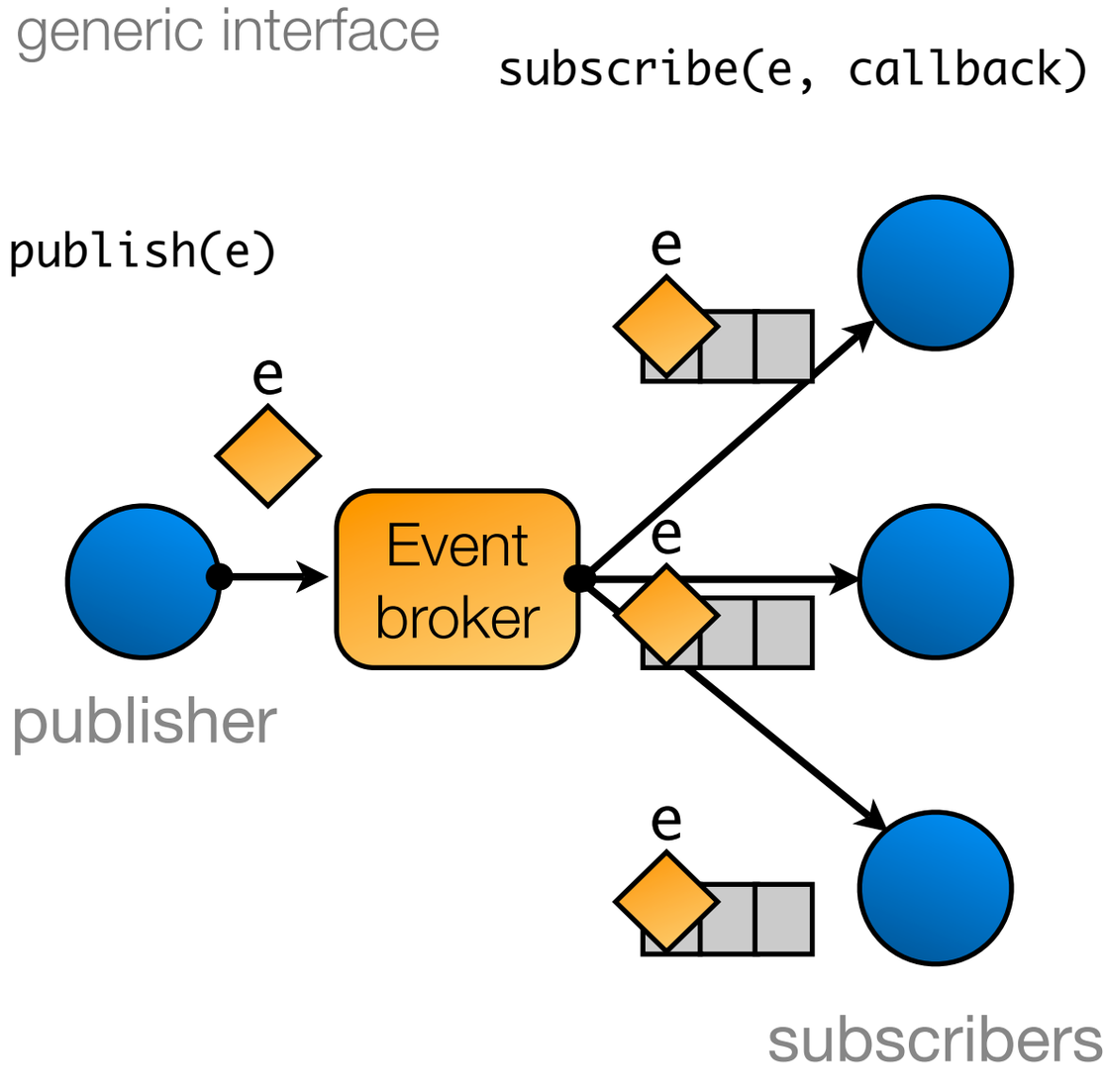
Object-event Impedance Mismatch



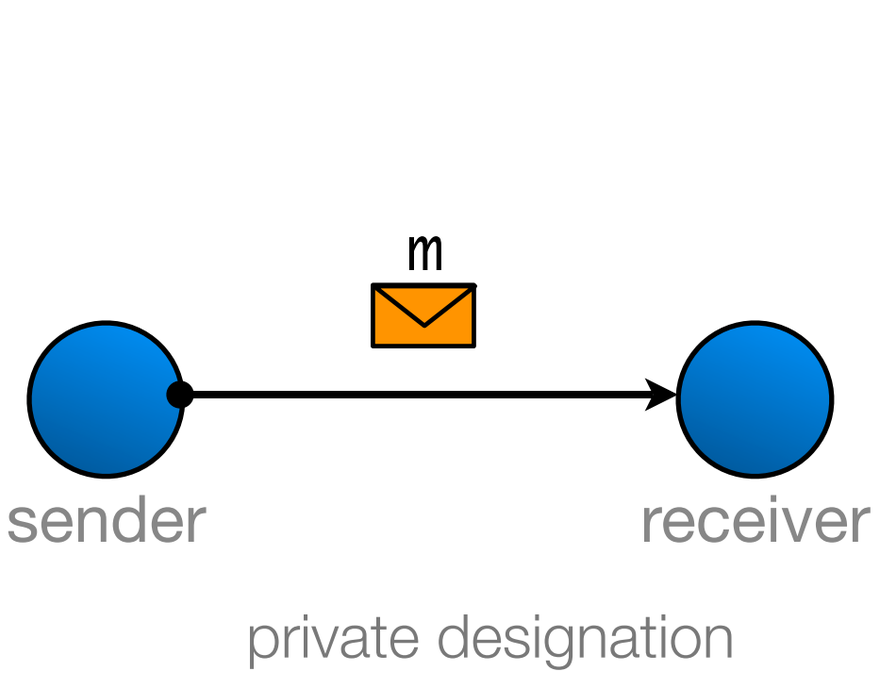


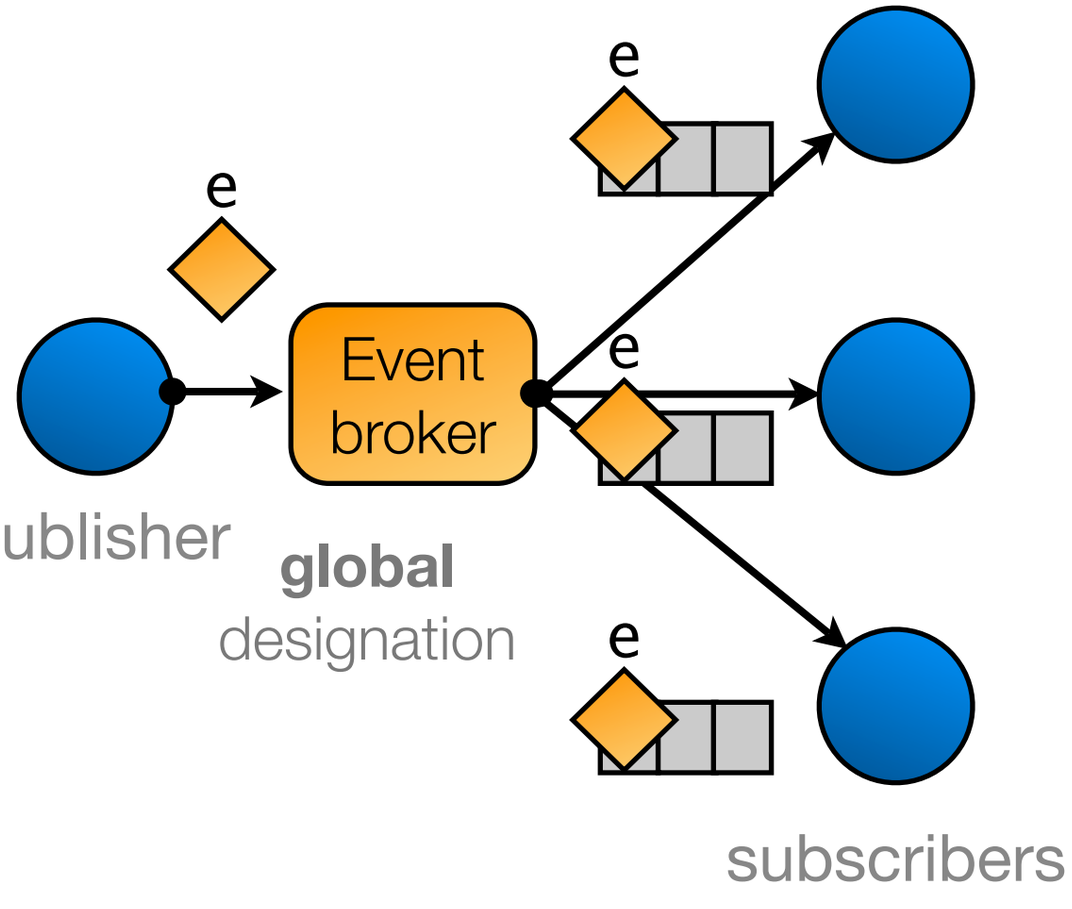
Object-event Impedance Mismatch



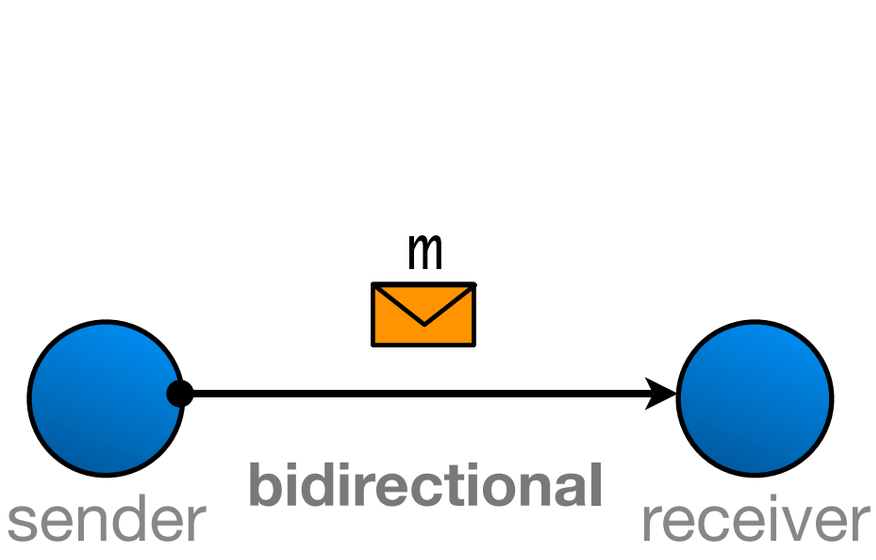


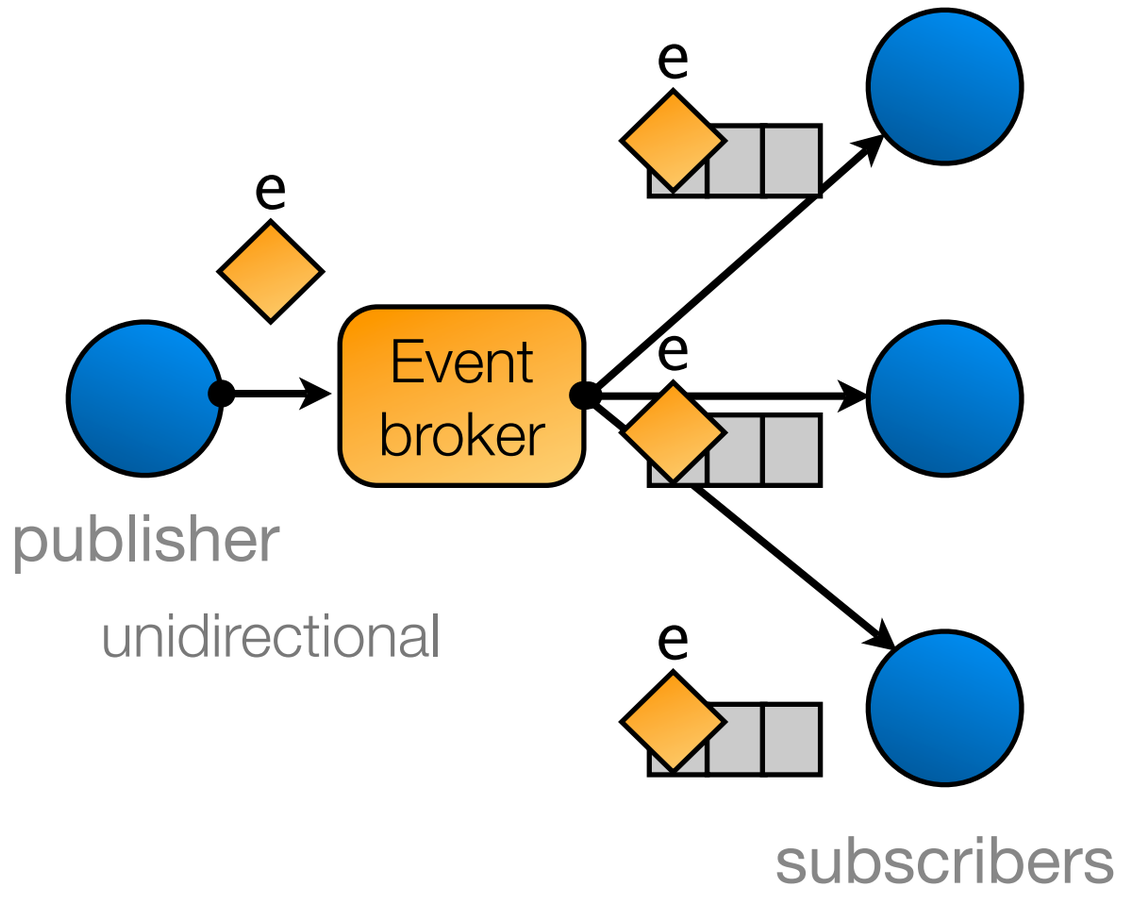
Object-event Impedance Mismatch



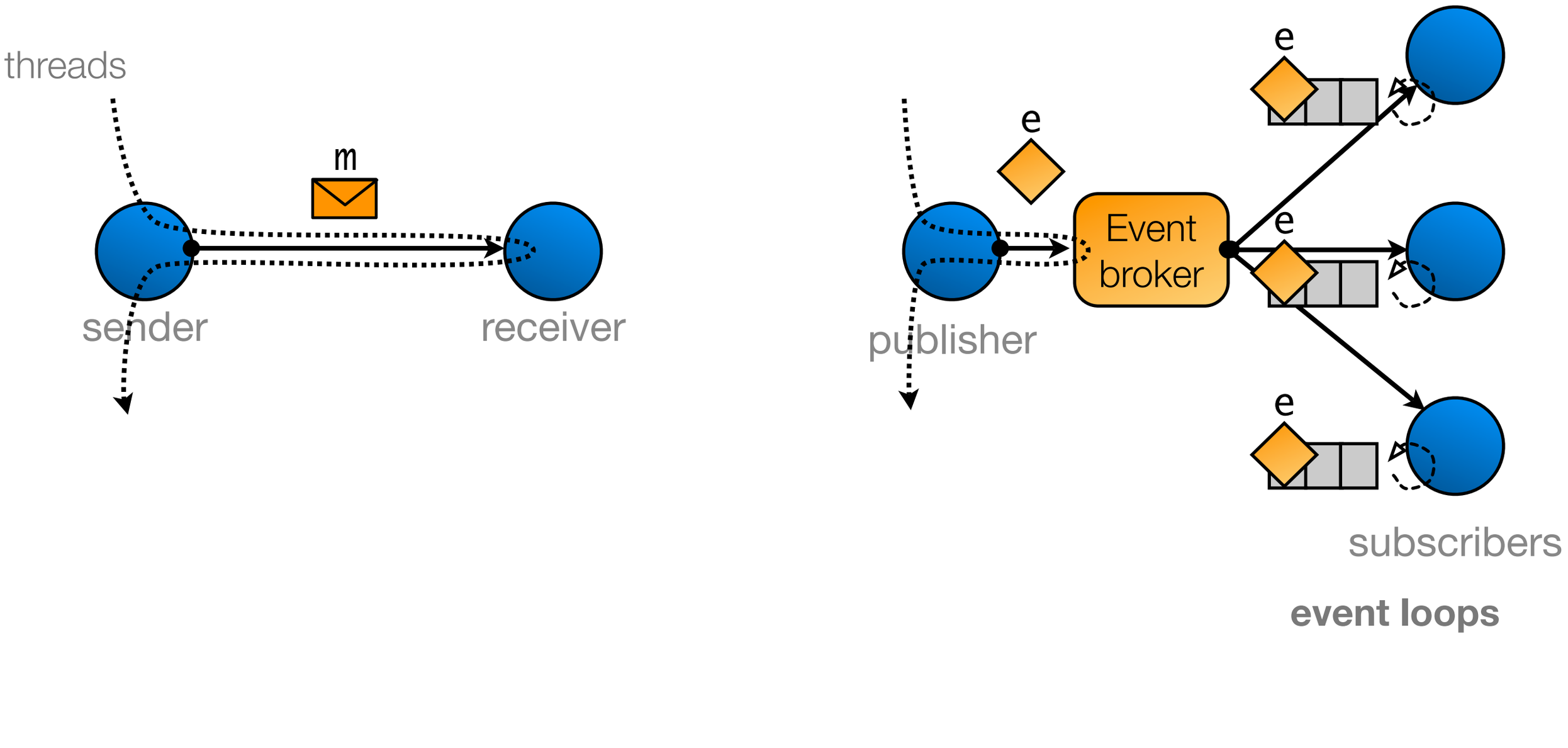


Object-event Impedance Mismatch



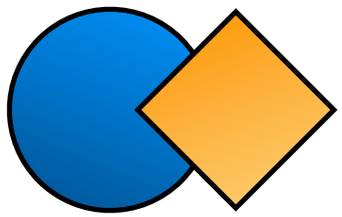


Object-event Impedance Mismatch



Roadmap

Motivation

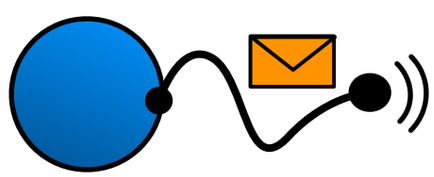


Object-event impedance mismatch

Coordination

Mobile ad hoc networks

Contribution



Ambient References

Validation

Evaluation

Implementation

Coordination in Mobile ad hoc Networks

Motivation

Object-event impedance mismatch

Coordination

Mobile ad hoc networks

Contribution

AMBIENTTALK

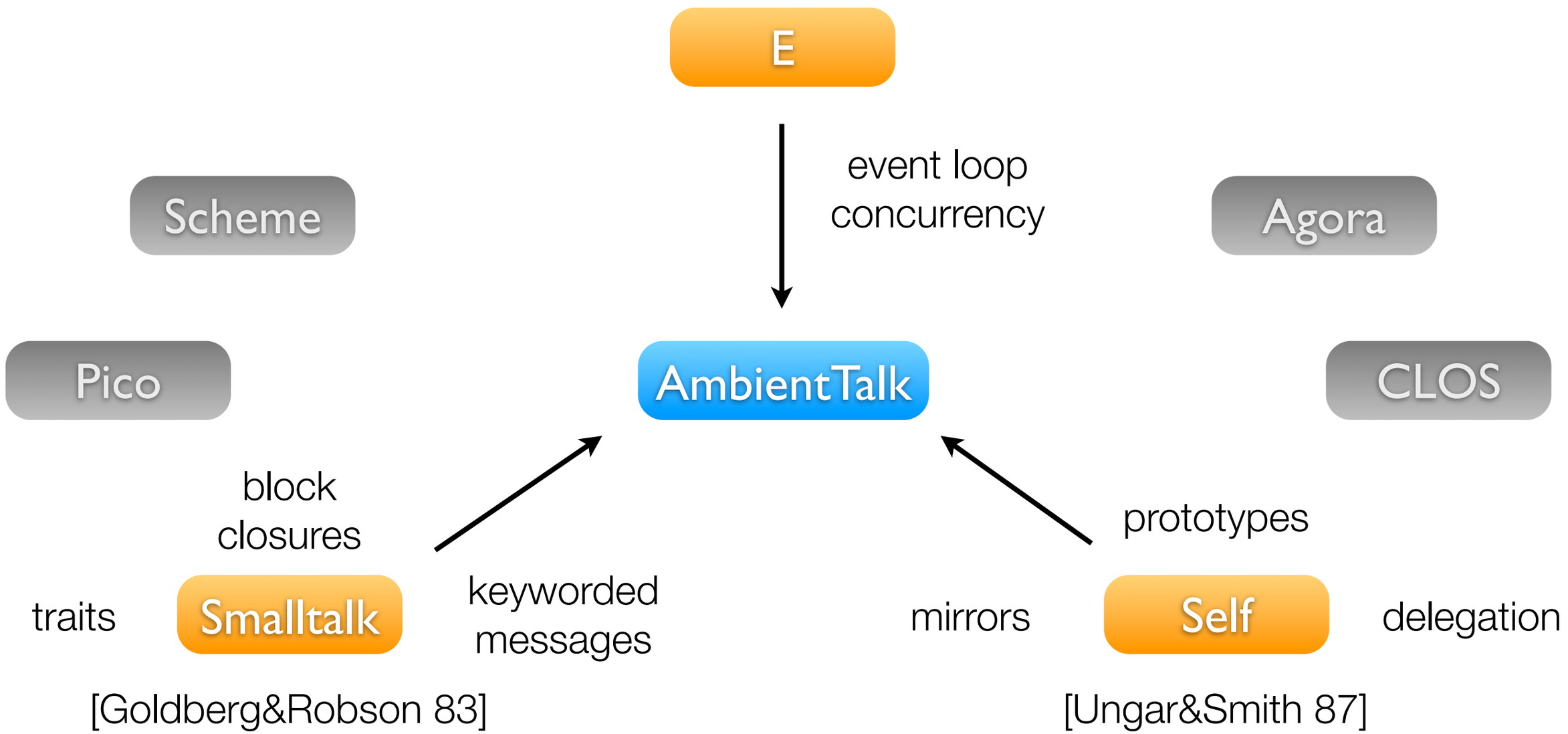
Ambient References

Validation

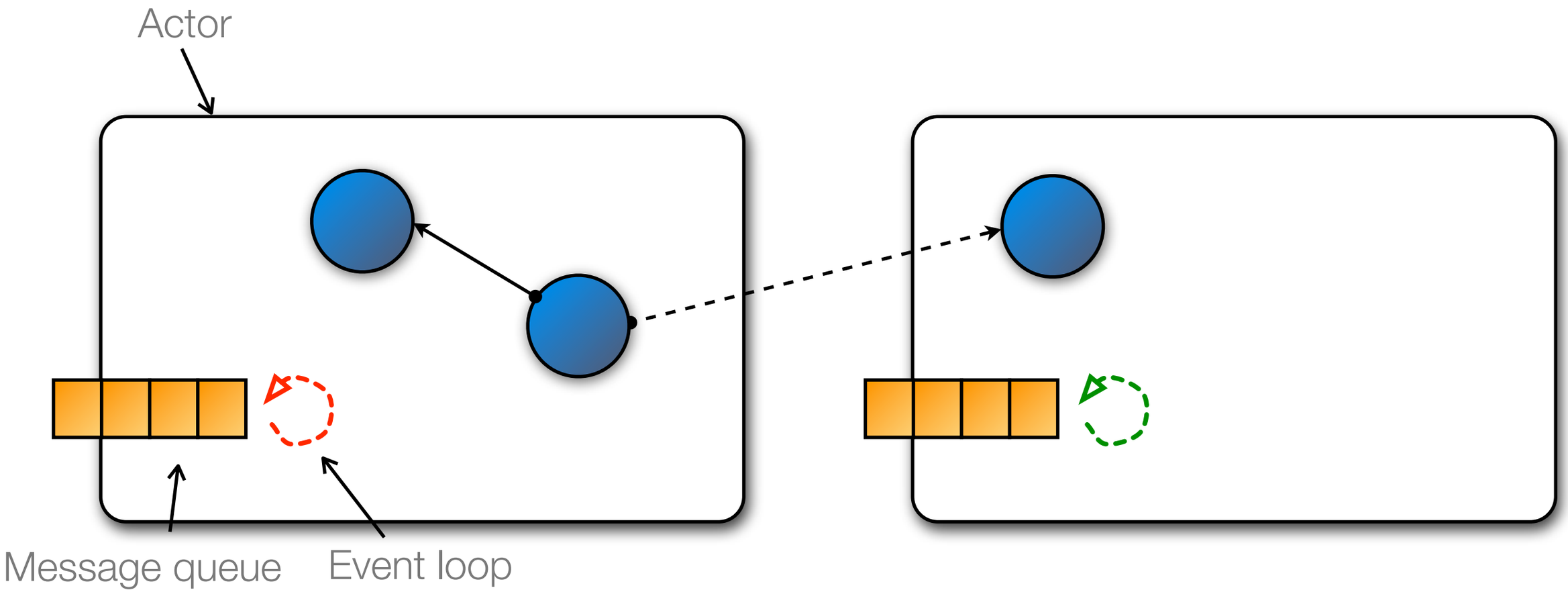
Evaluation

Implementation

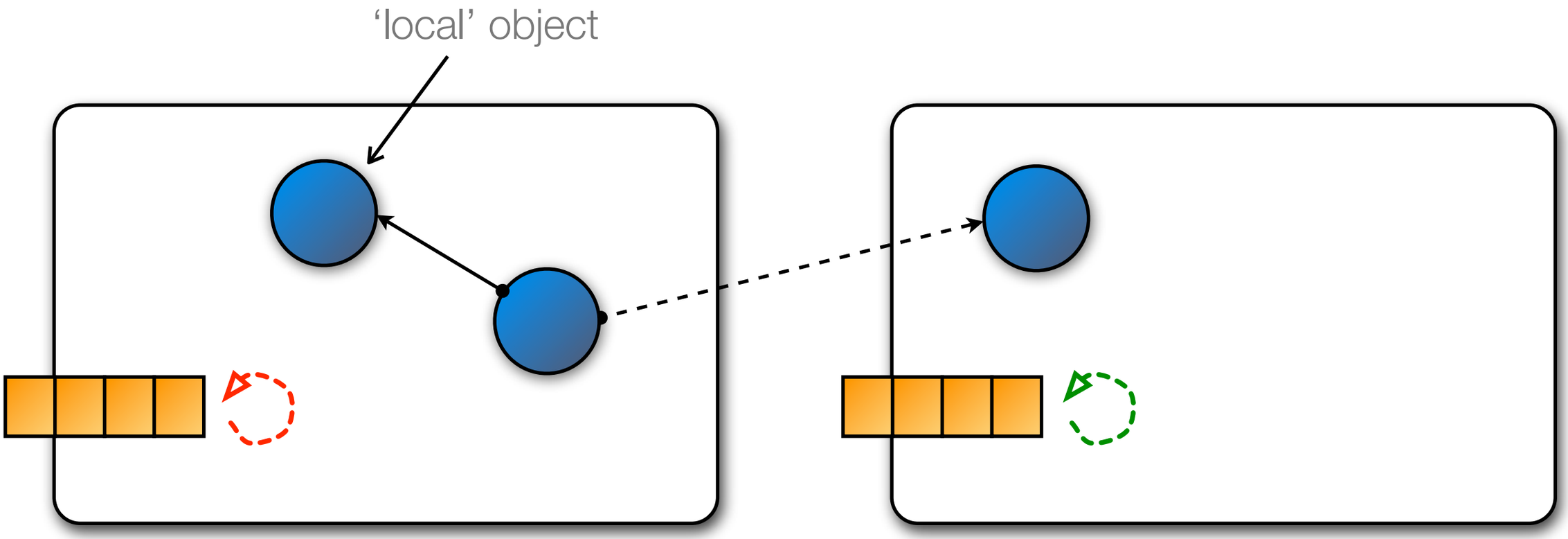
Language Features



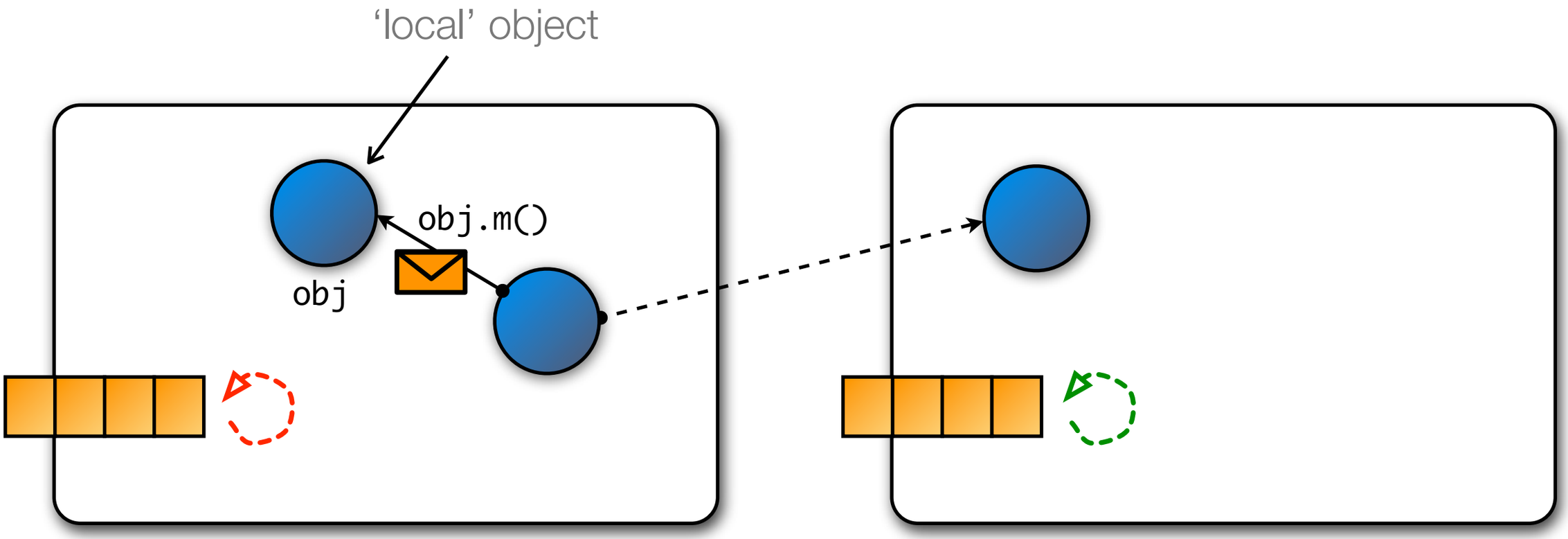
Event Loop Concurrency in AmbientTalk



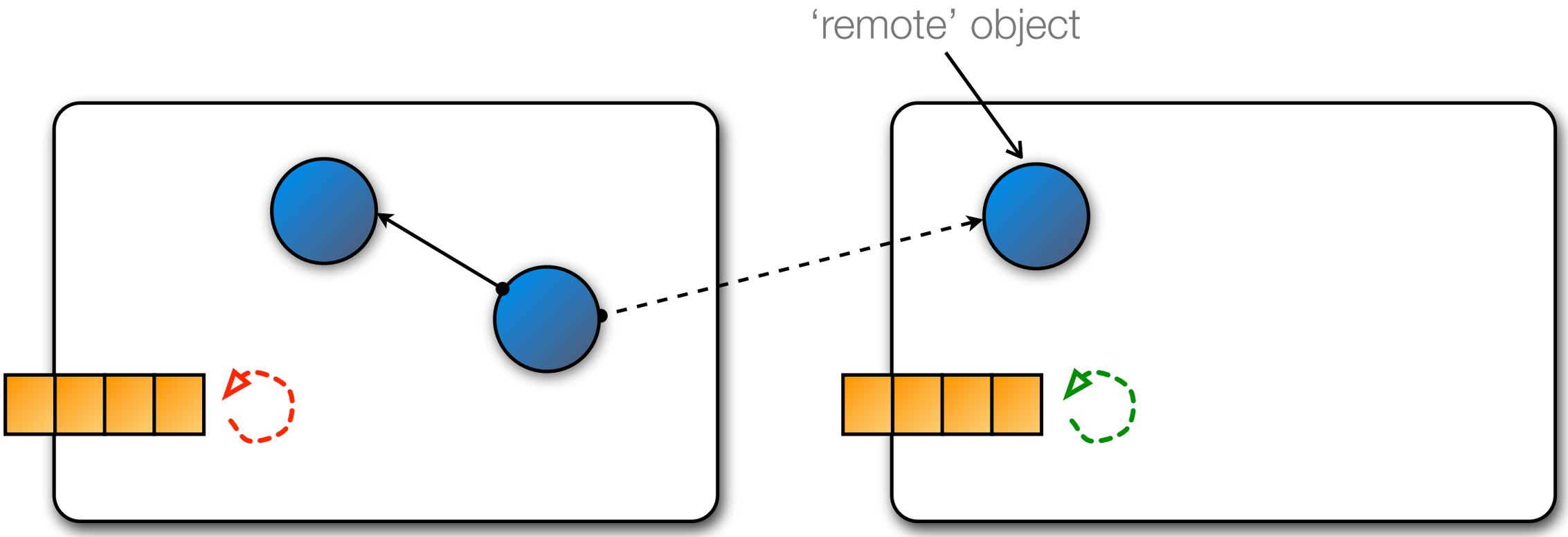
Event Loop Concurrency in AmbientTalk



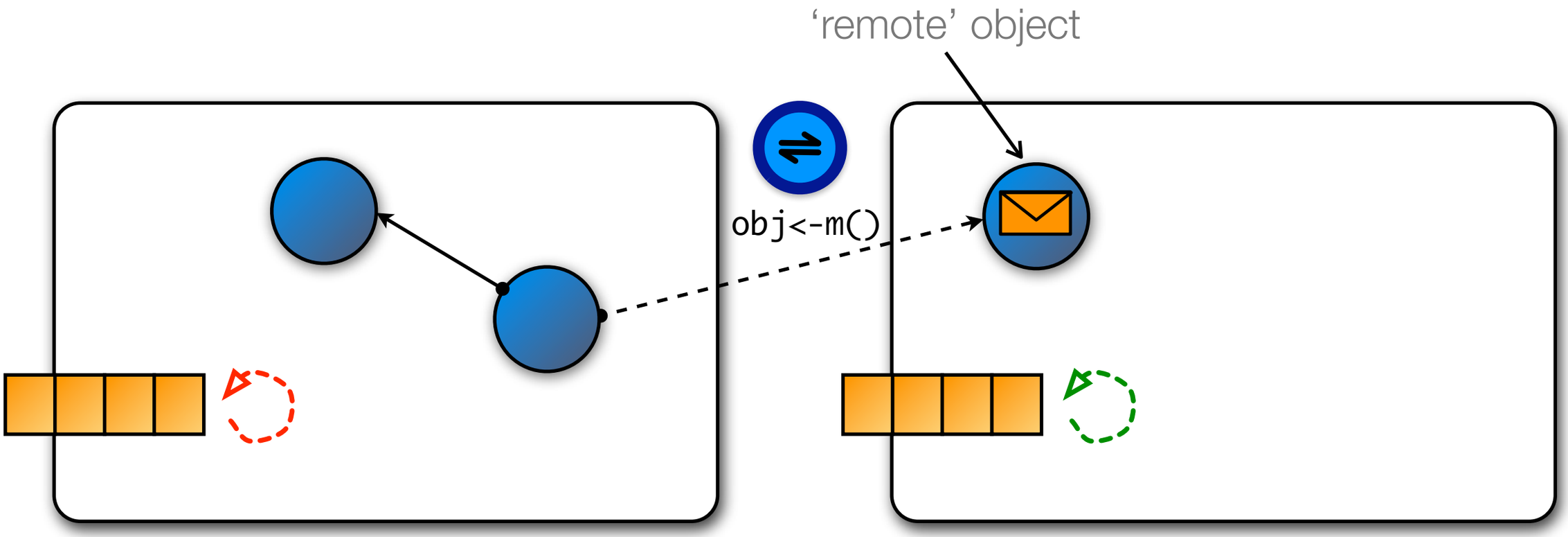
Event Loop Concurrency in AmbientTalk



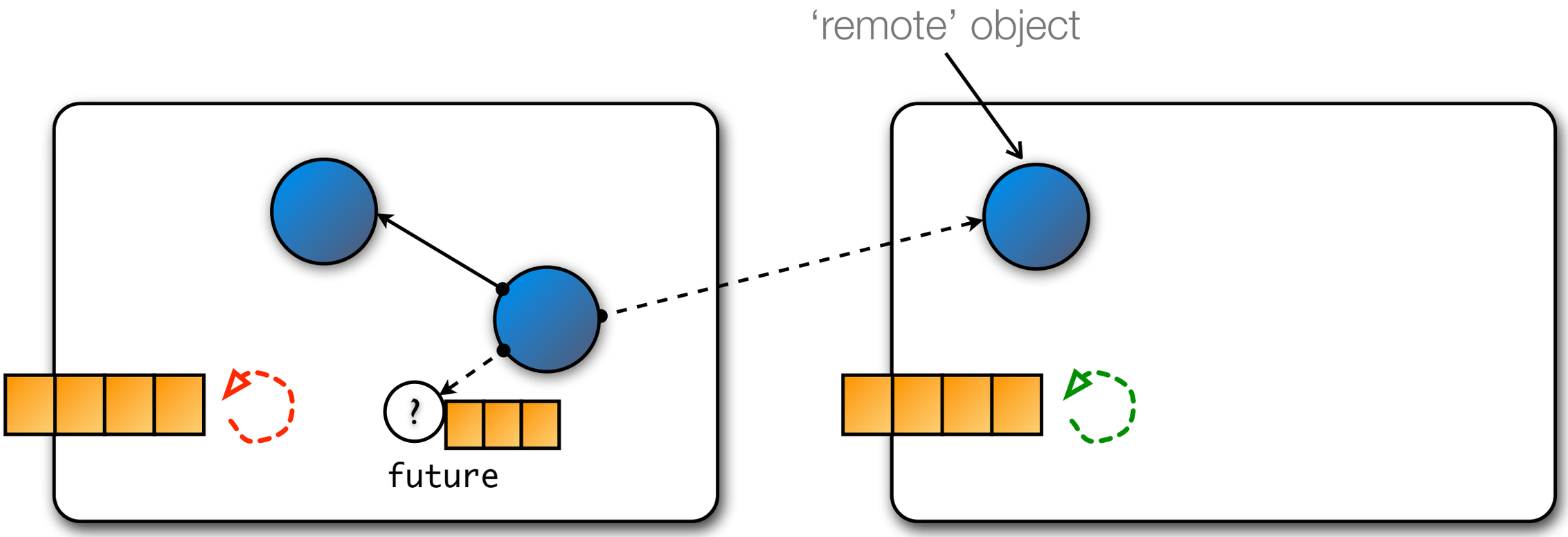
Event Loop Concurrency in AmbientTalk



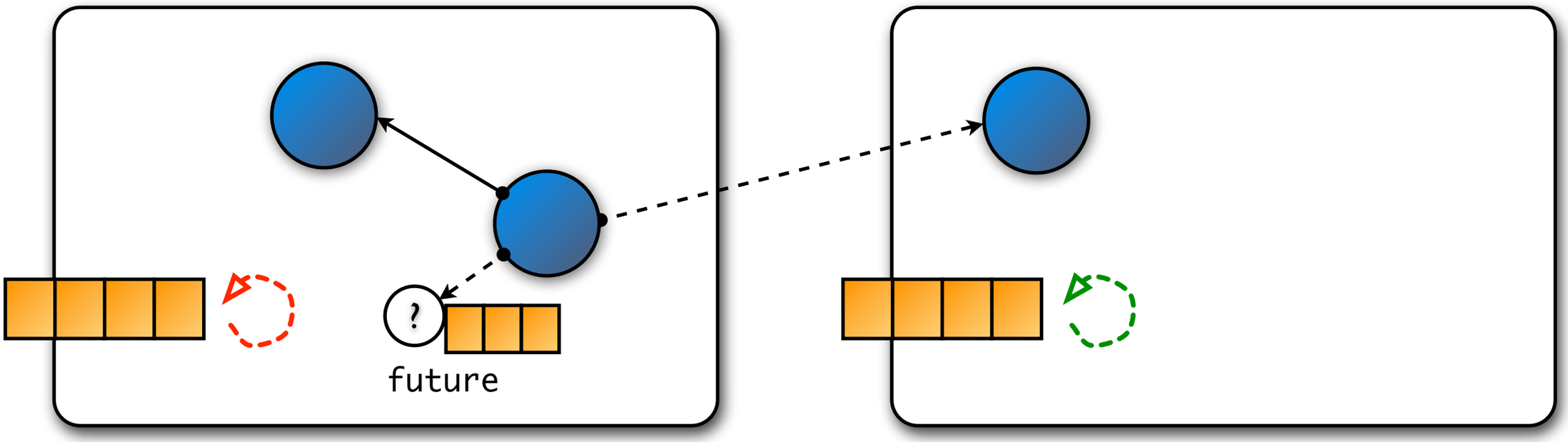
Event Loop Concurrency in AmbientTalk



Event Loop Concurrency in AmbientTalk

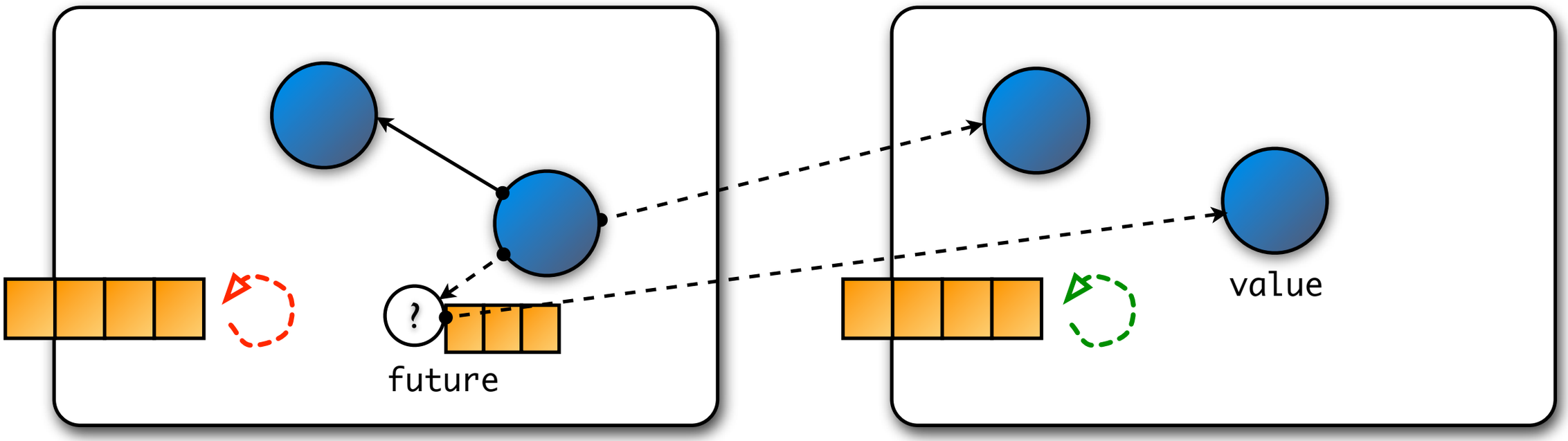


Event Loop Concurrency in AmbientTalk



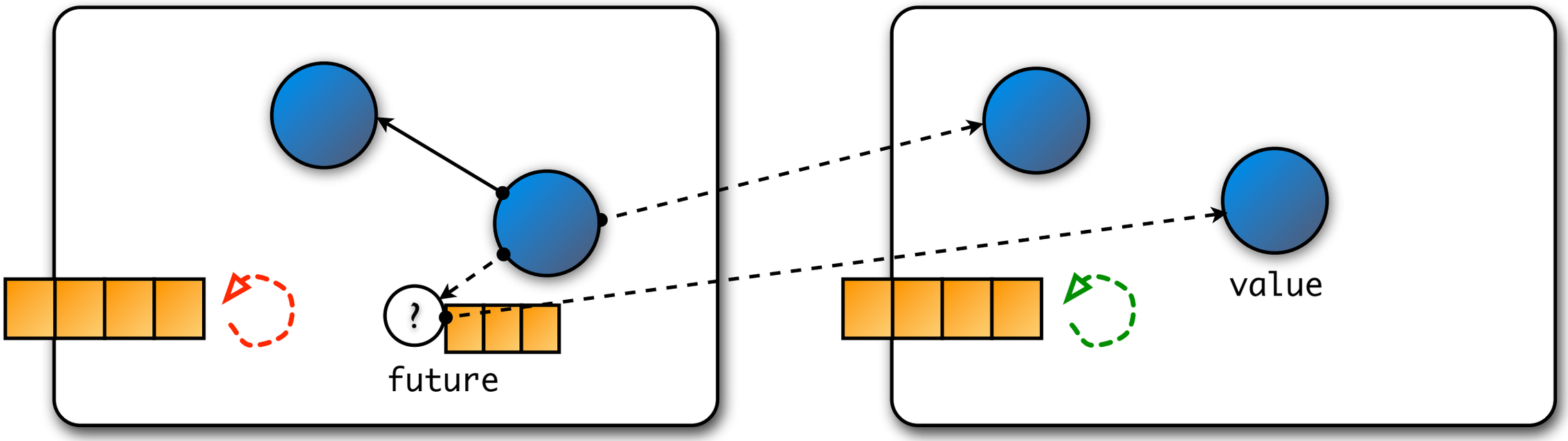
```
when: future becomes: { |value|
  // process reply
}
```

Event Loop Concurrency in AmbientTalk



```
when: future becomes: { |value|
  // process reply
}
```

Event Loop Concurrency in AmbientTalk

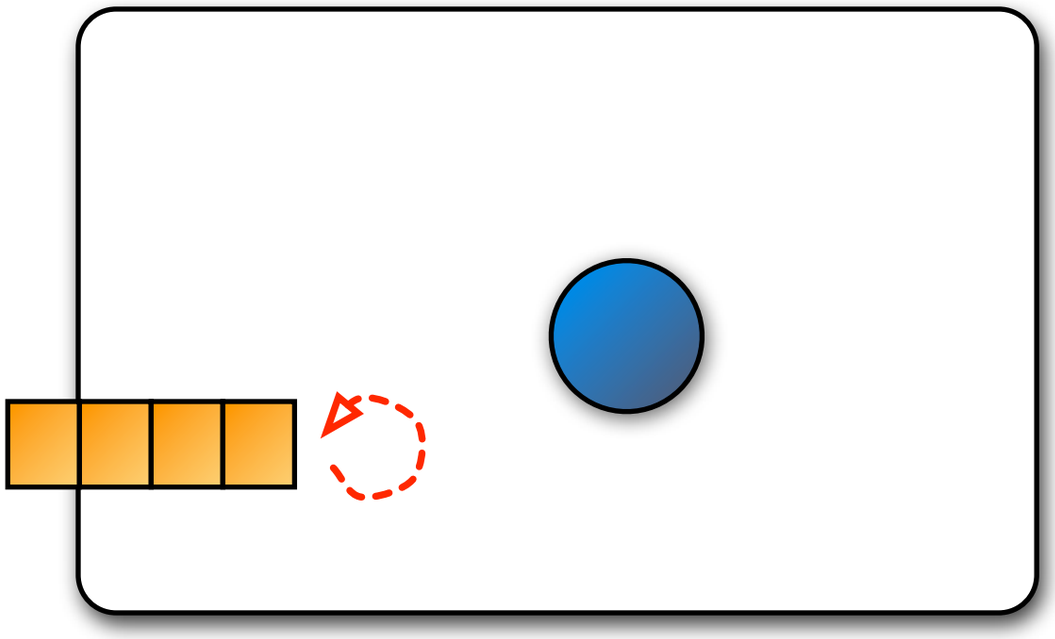


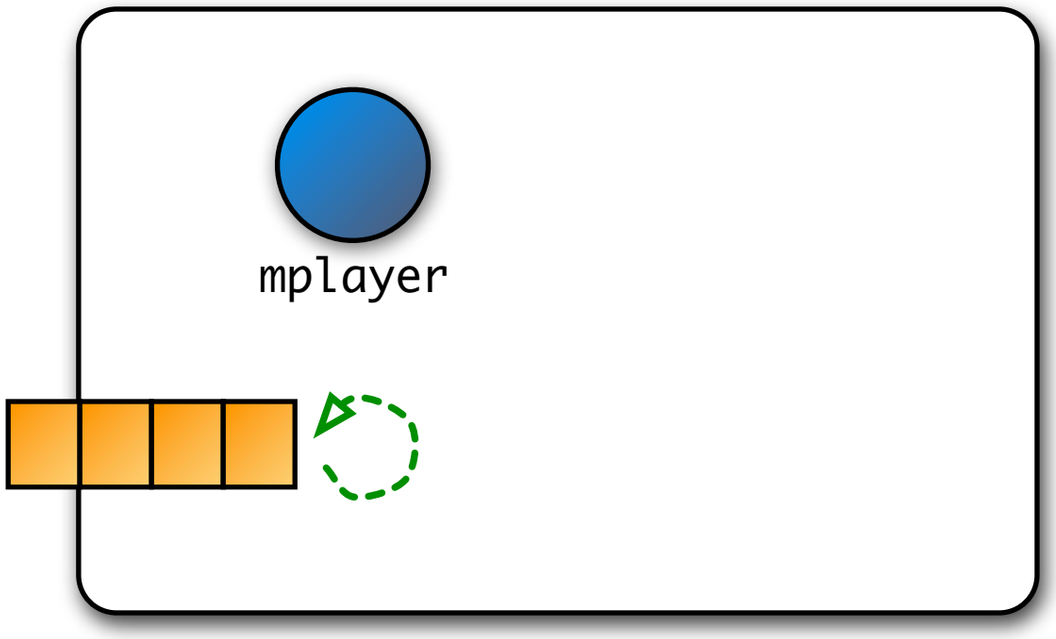
```
when: future becomes: { |value|
  // process reply
}
```

Actors cannot deadlock

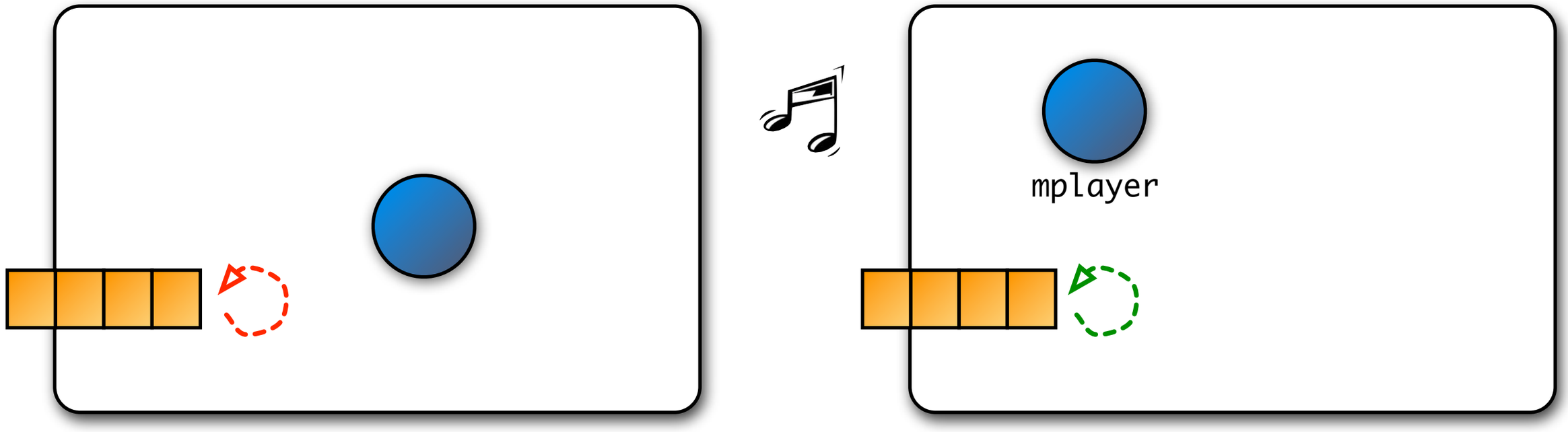
No race conditions on objects

Exporting & discovering objects





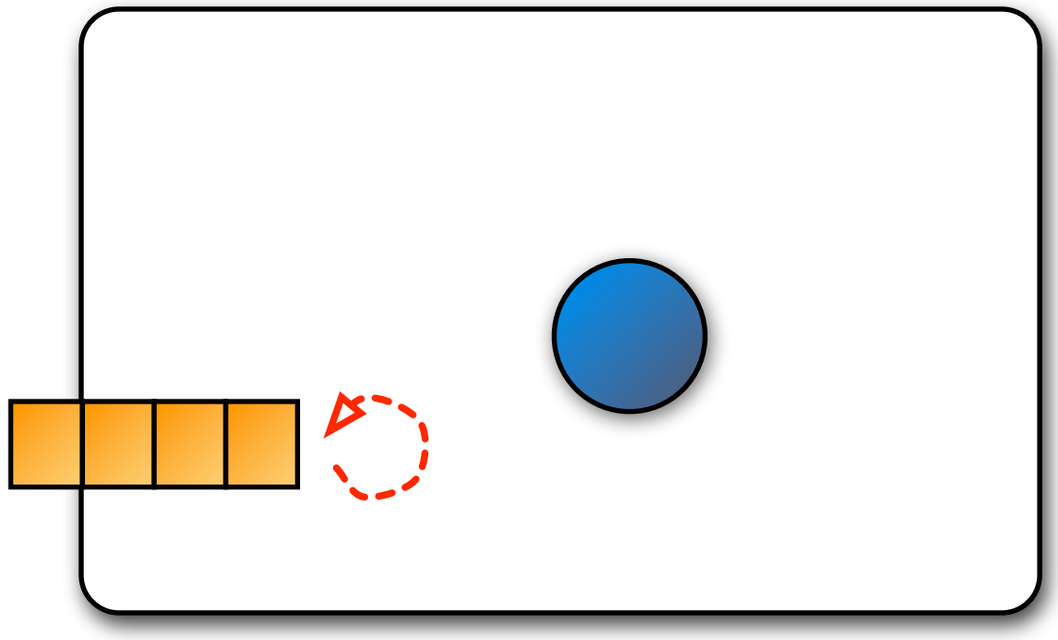
Exporting & discovering objects



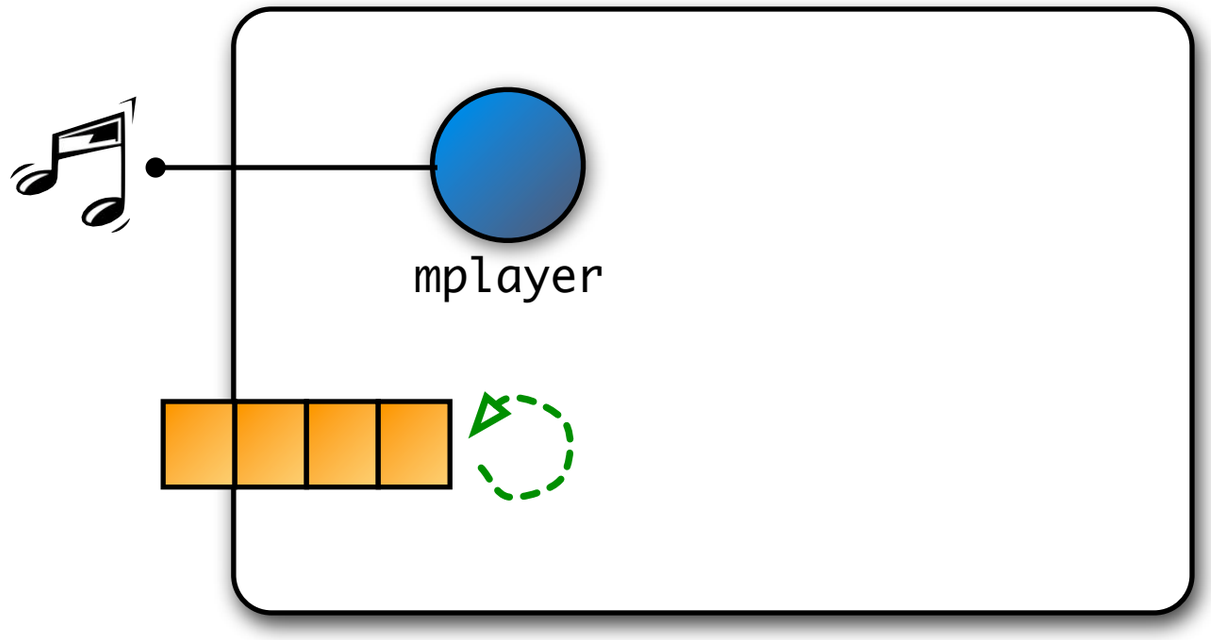
deftype MusicPlayer

deftype MusicPlayer

Exporting & discovering objects

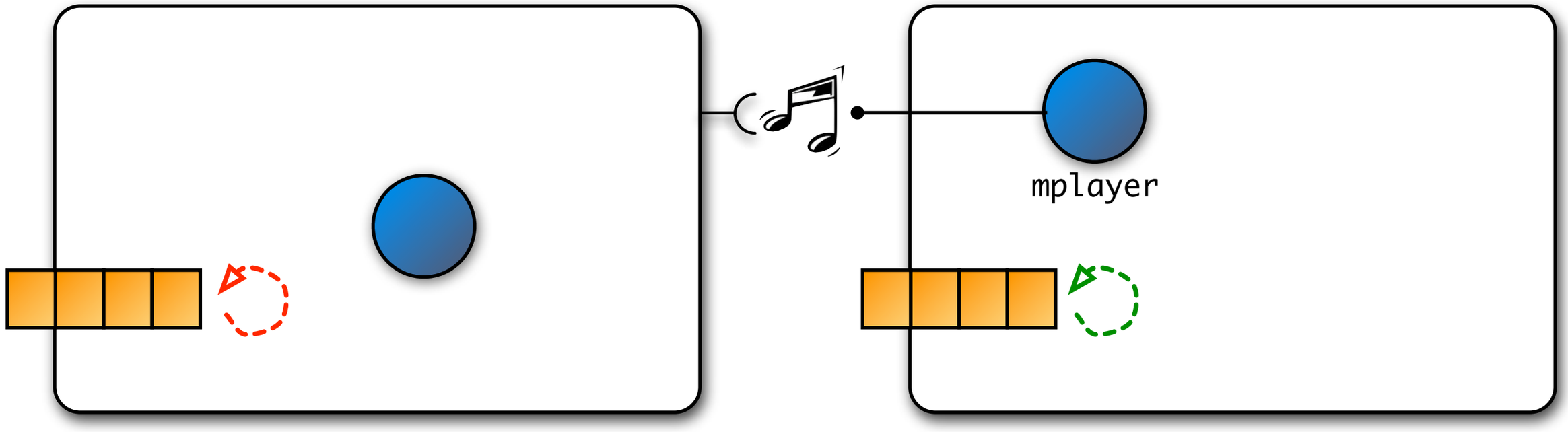


```
deftype MusicPlayer
```




```
deftype MusicPlayer

export: mplayer as: MusicPlayer
```

Exporting & discovering objects

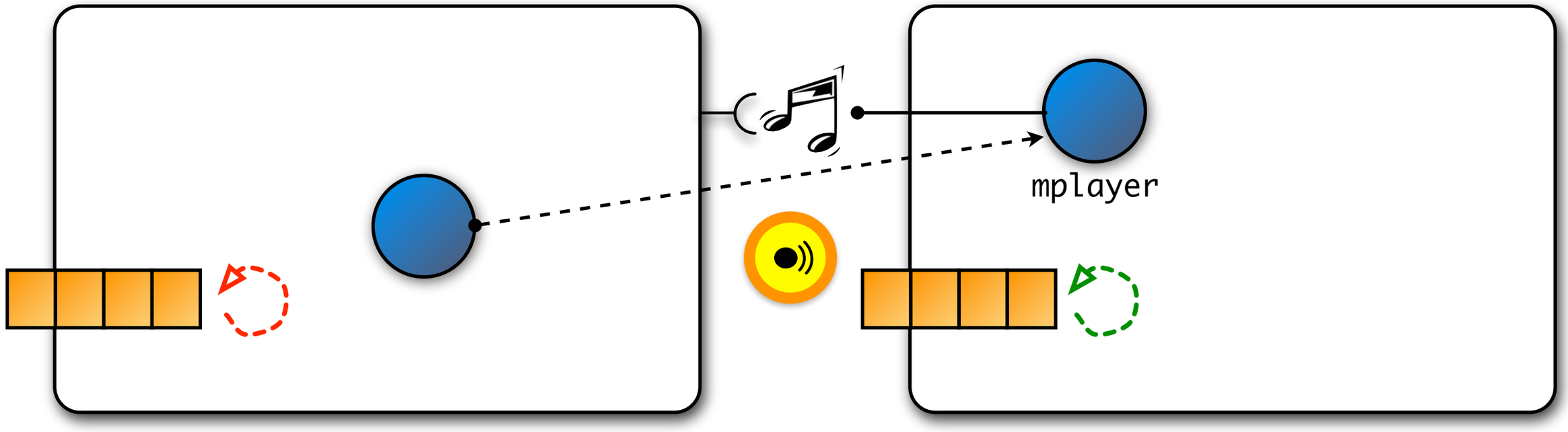



```
deftype MusicPlayer
```

```
deftype MusicPlayer

export: mplayer as: MusicPlayer
```

```
whenever: MusicPlayer discovered: { |mplayer|
  // open a session
}
```

Exporting & discovering objects

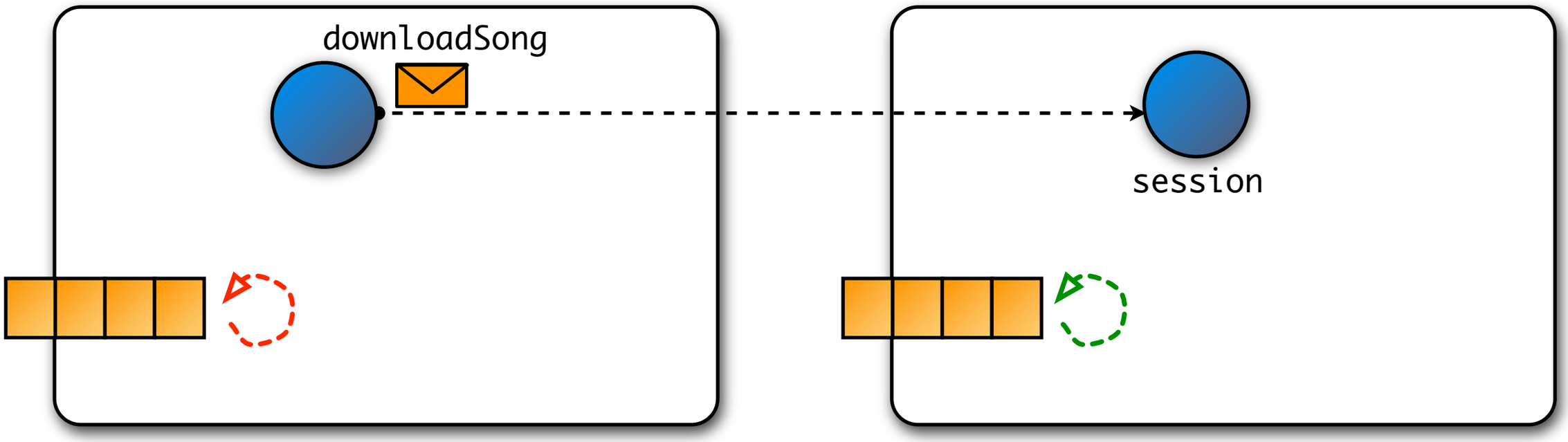



```
deftype MusicPlayer
```

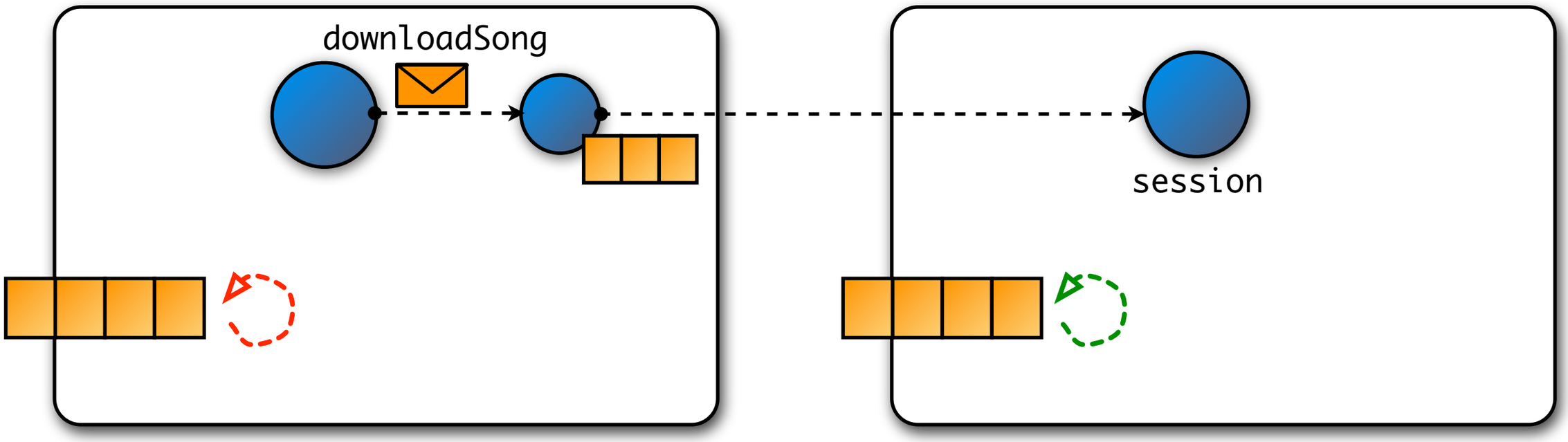
```
deftype MusicPlayer

export: mplayer as: MusicPlayer
```

```
whenever: MusicPlayer discovered: { |mplayer|
  // open a session
}
```

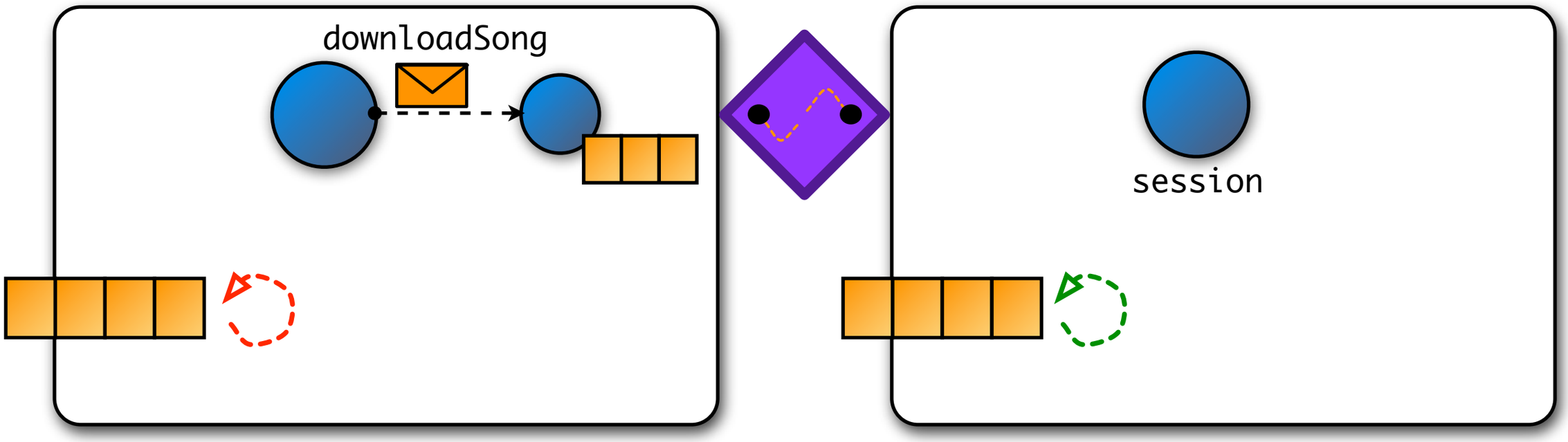
Far References



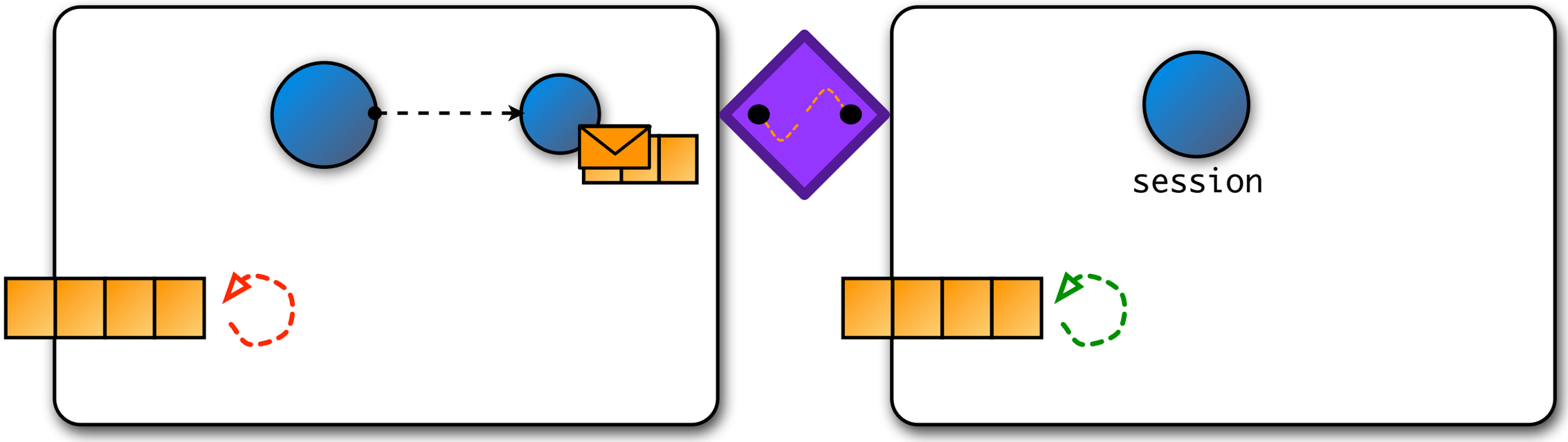
Far References



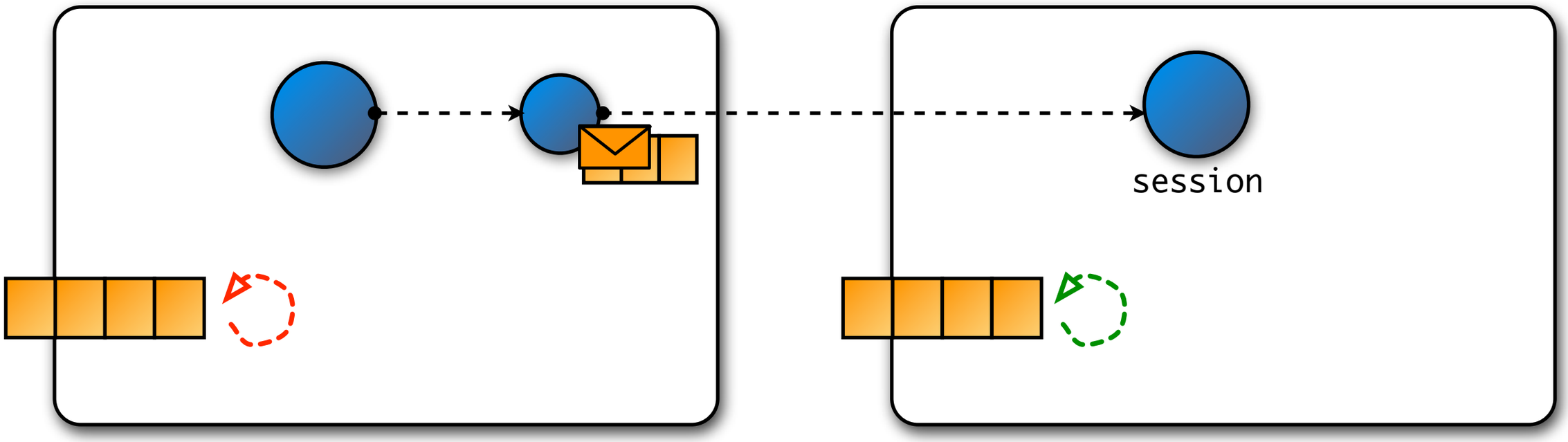
Far References



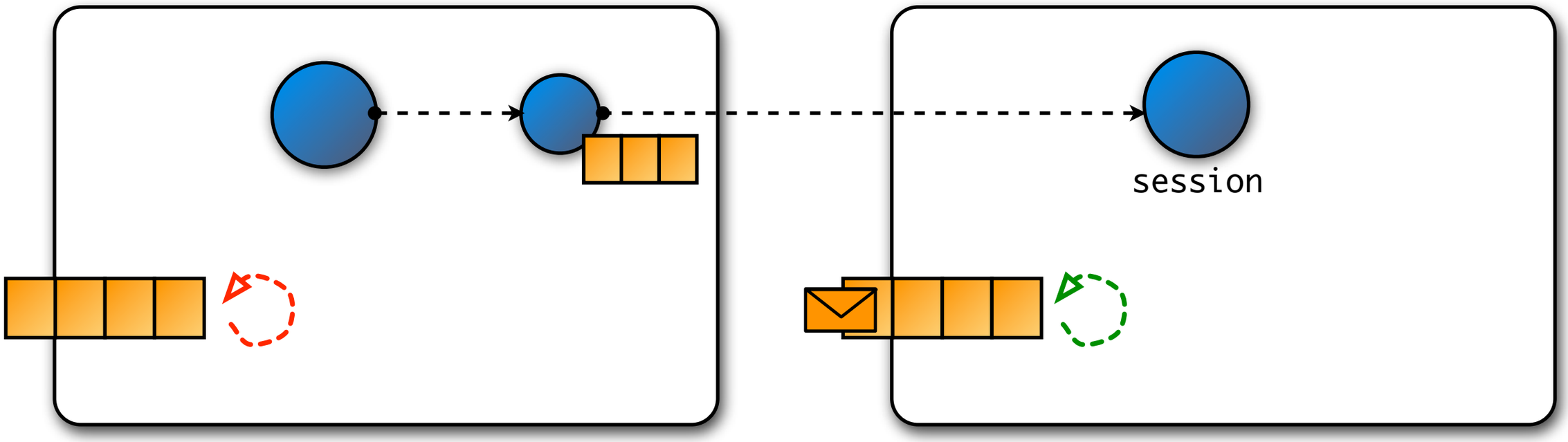
Far References



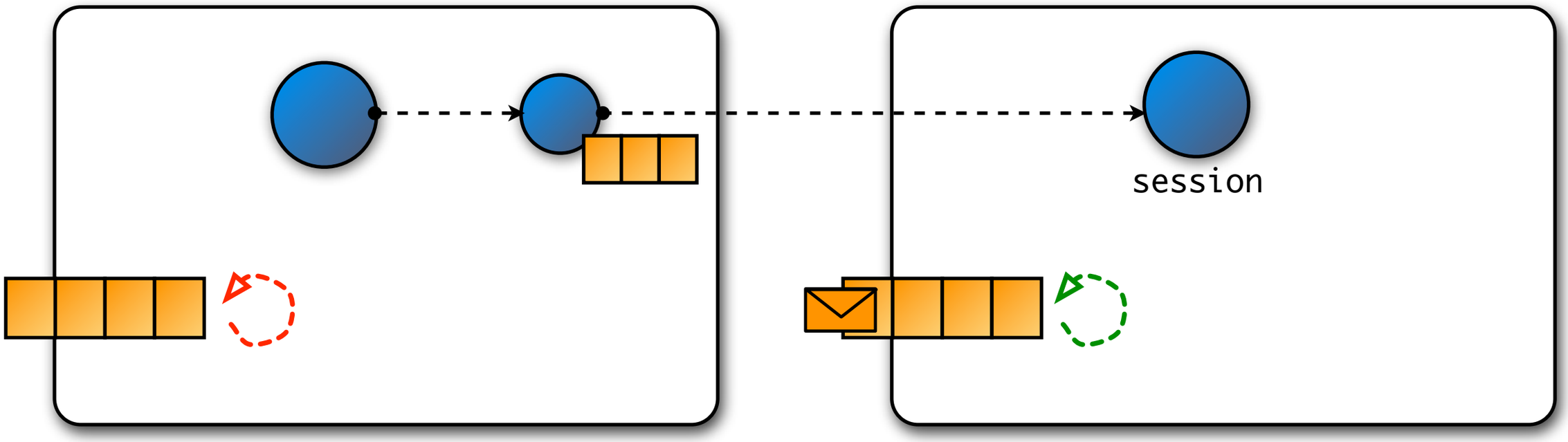
Far References



Far References



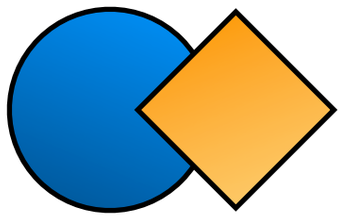
Far References




```
when: session<-downloadSong(s)@Due(timeout) becomes: { |ack|
  // continue exchange
} catch: TimeoutException using: { |e|
  // stop exchange
}
```

Roadmap

Motivation



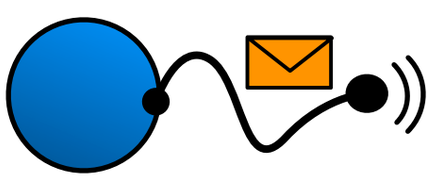
Object-event impedance mismatch

Coordination

Mobile ad hoc networks

Contribution

AMBIENTTALK



Ambient References

Validation

Evaluation

Implementation

Ambient References

Motivation

Contribution

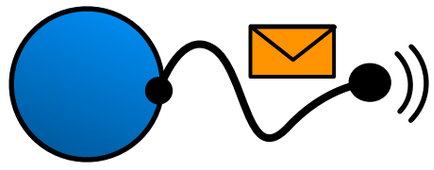
Validation

Object-event impedance mismatch

Coordination

Evaluation

Mobile ad hoc networks

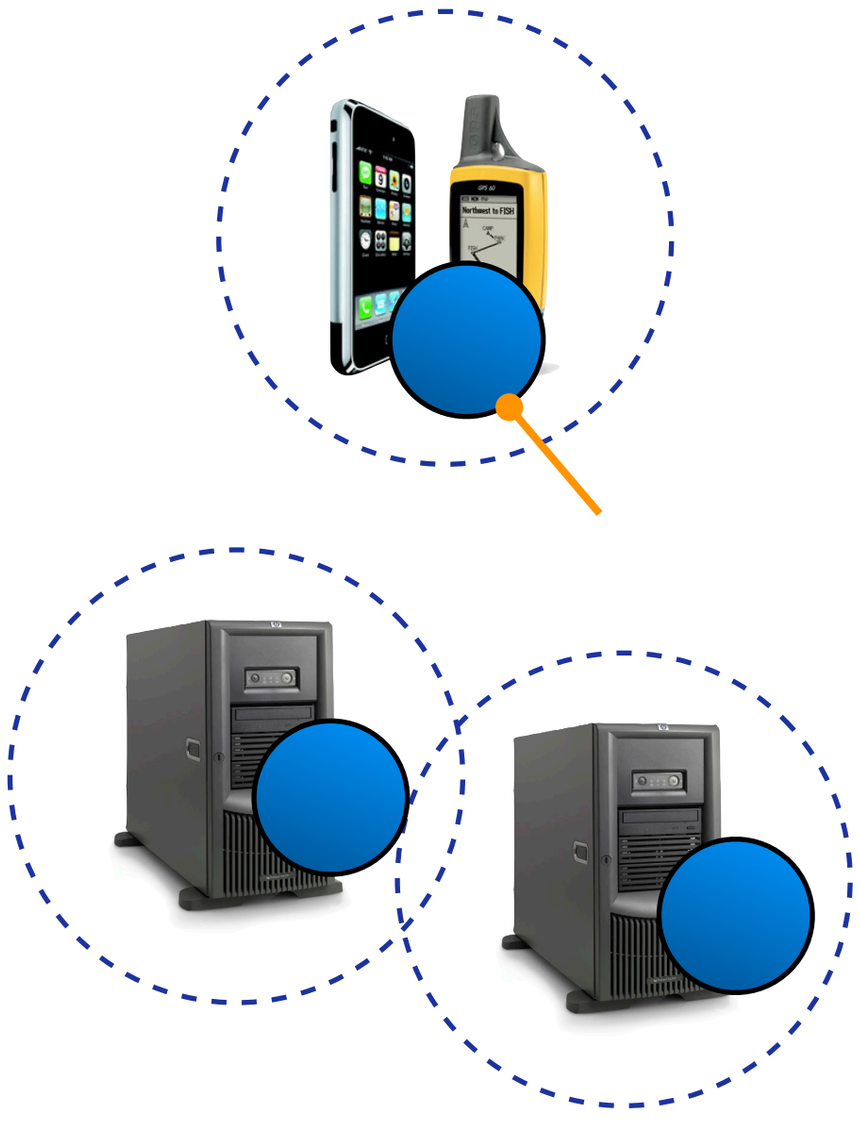


Ambient References

Implementation

Motivation

Second-class communication patterns



Roaming

Motivation

Second-class communication patterns

Roaming

Motivation

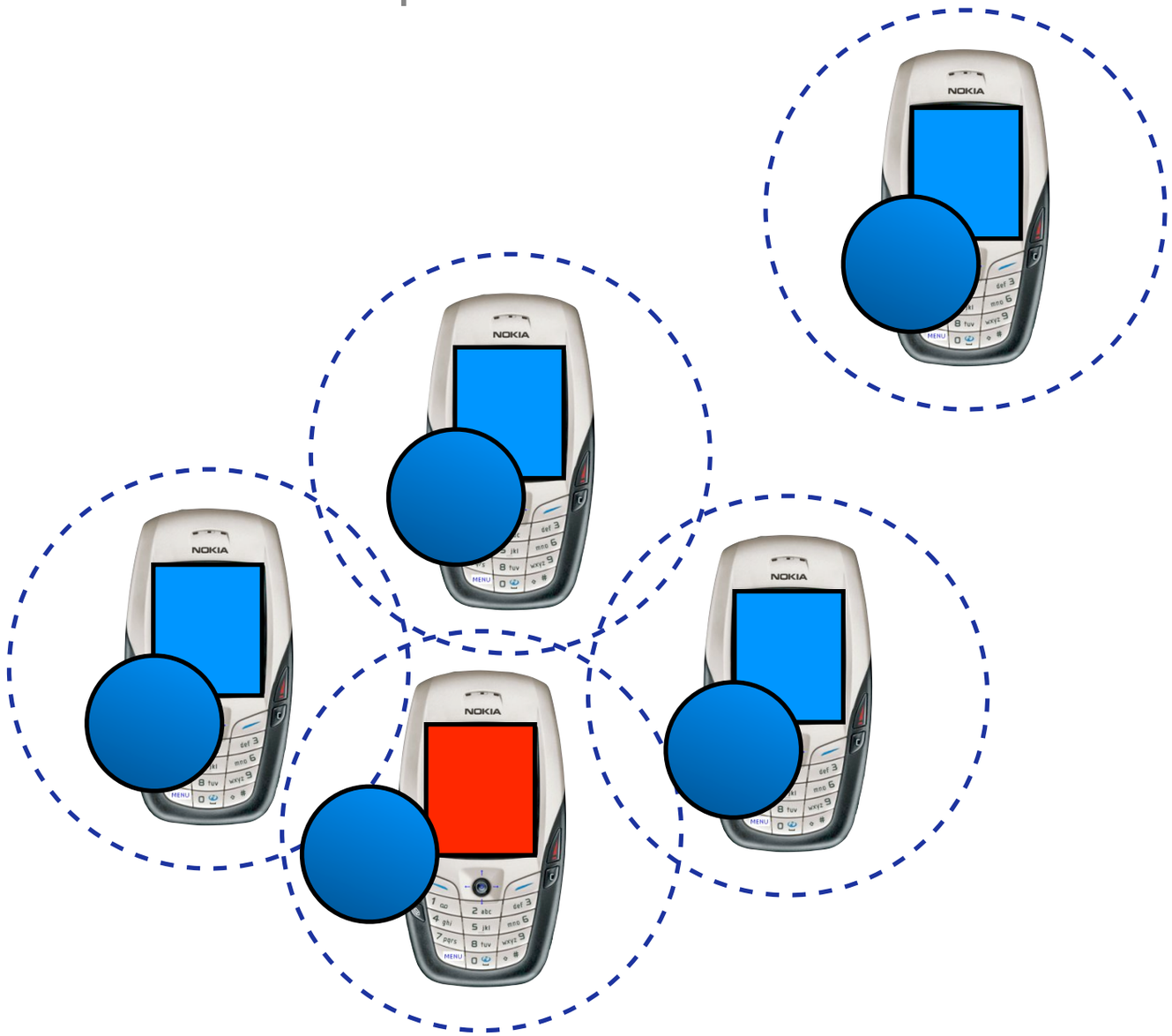
Second-class communication patterns

Roaming

Motivation

Second-class communication patterns

Roaming

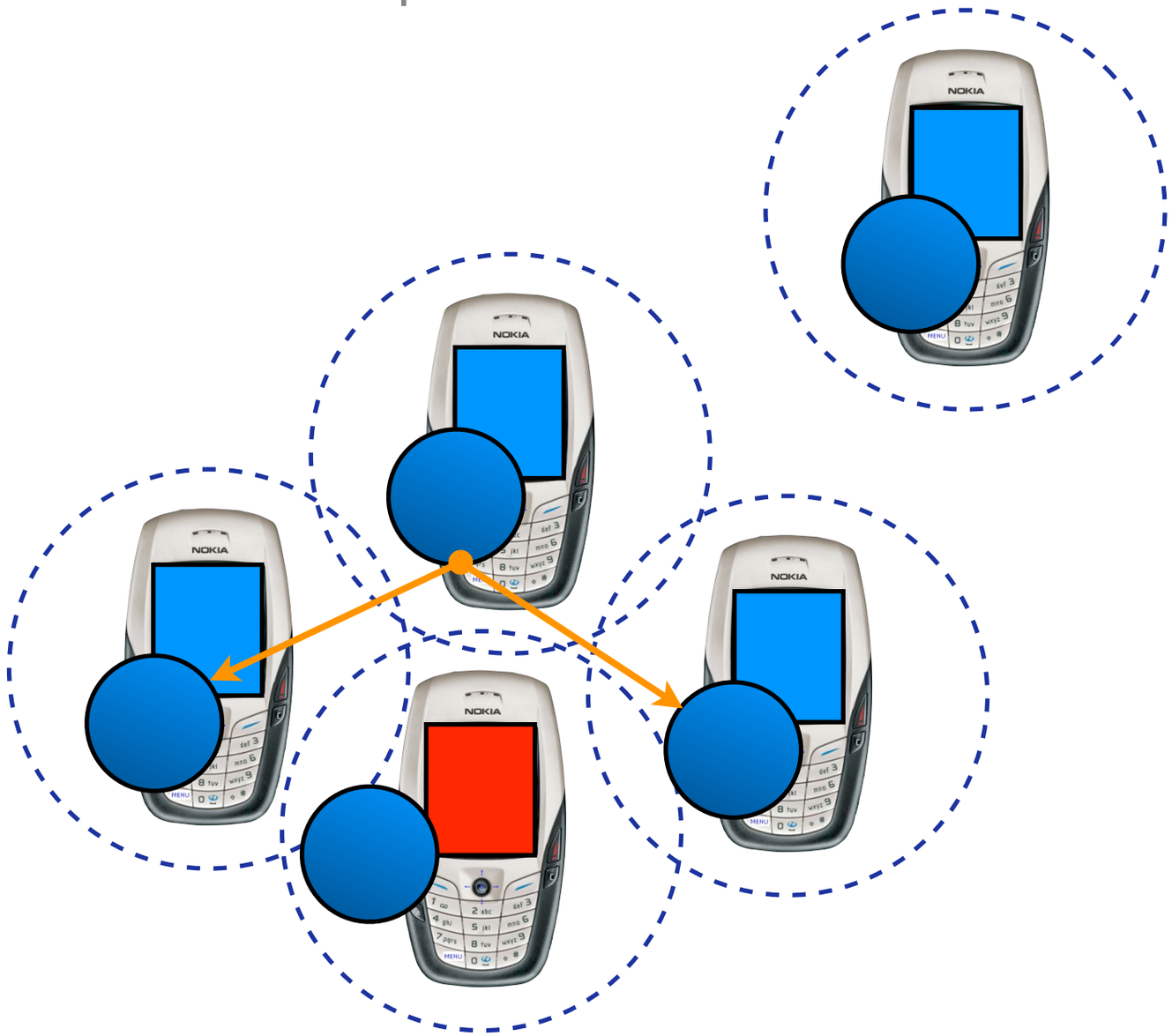


One-to-many communication

Motivation

Second-class communication patterns

Roaming

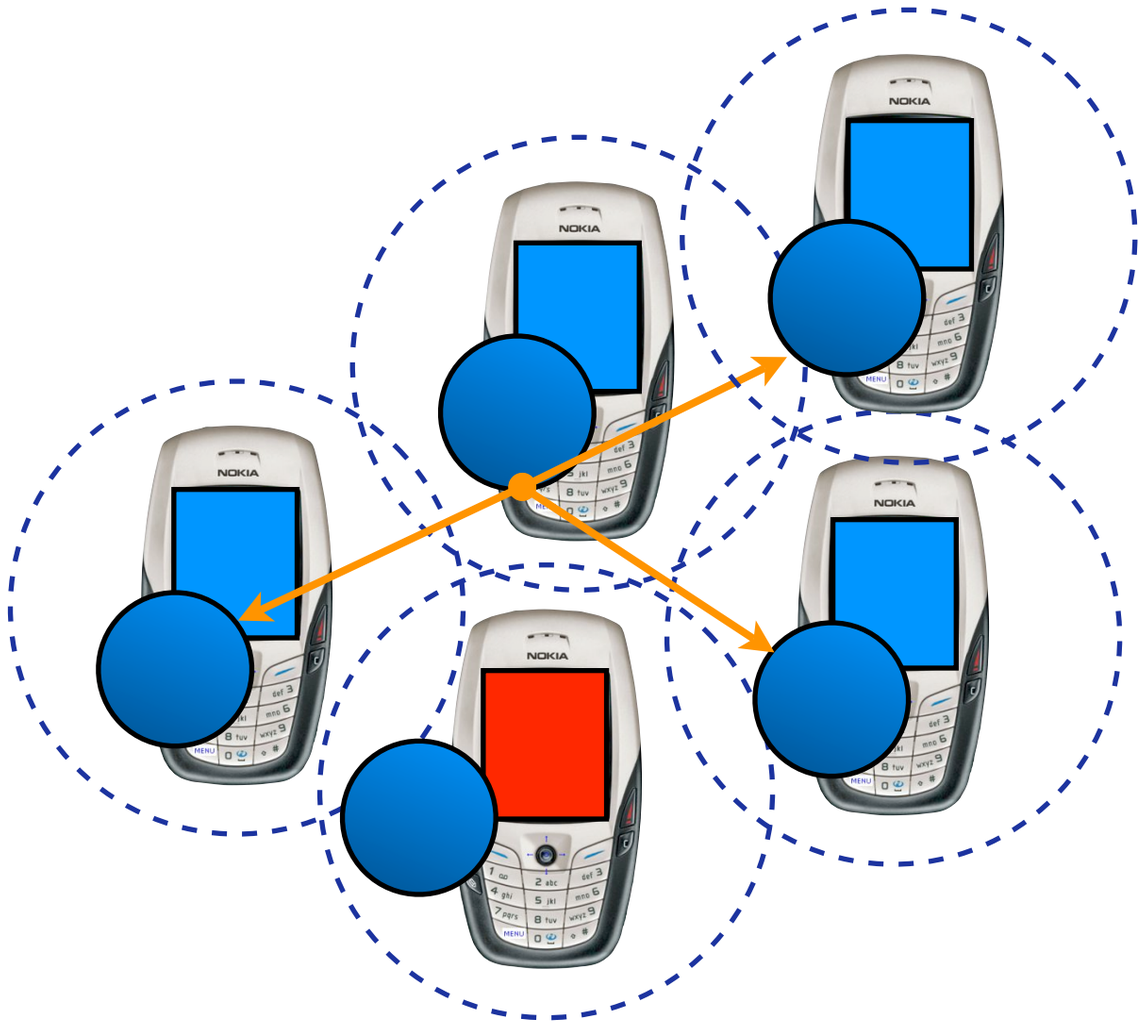


One-to-many communication

Motivation

Second-class communication patterns

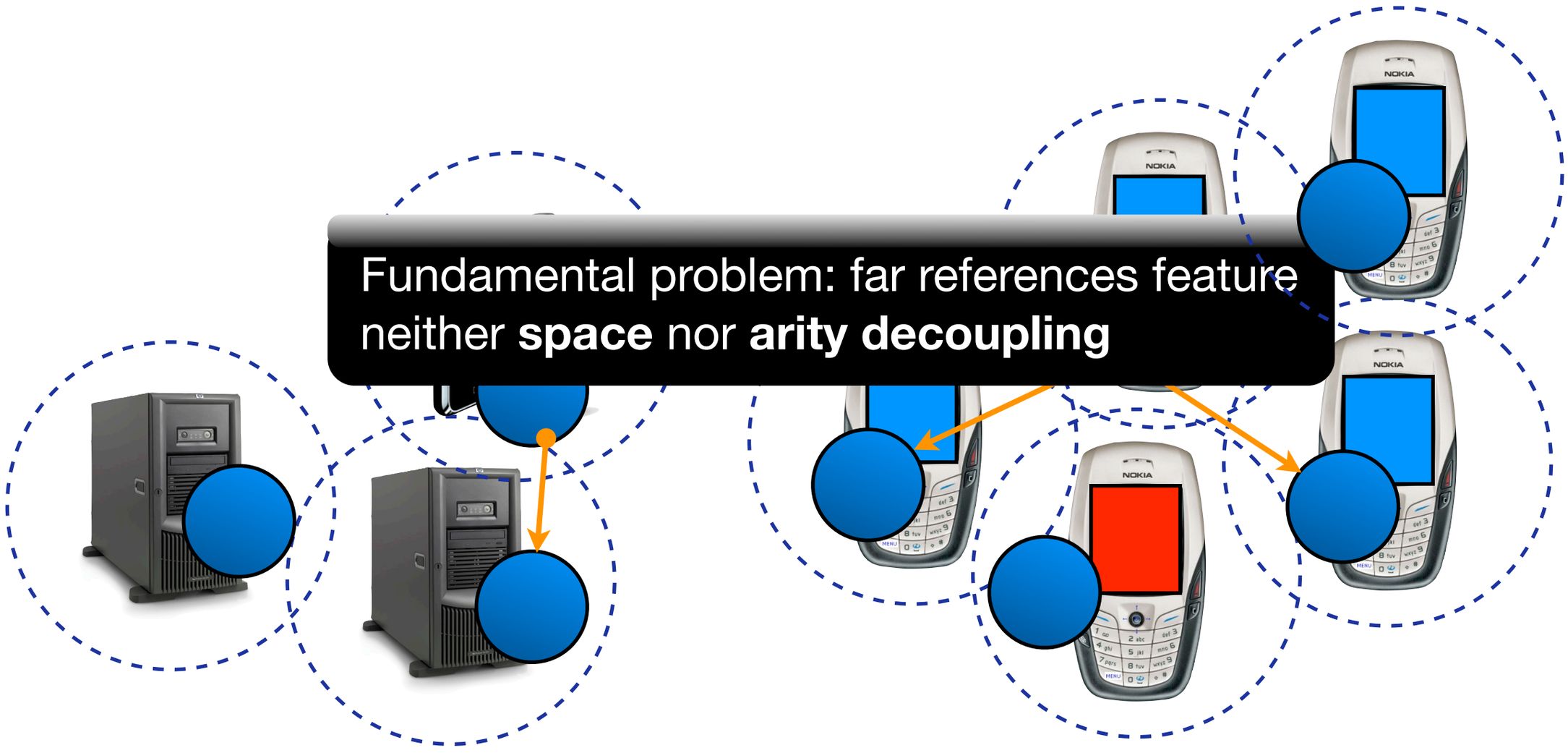
Roaming



One-to-many communication

Motivation

Second-class communication patterns

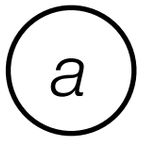


Roaming

One-to-many communication

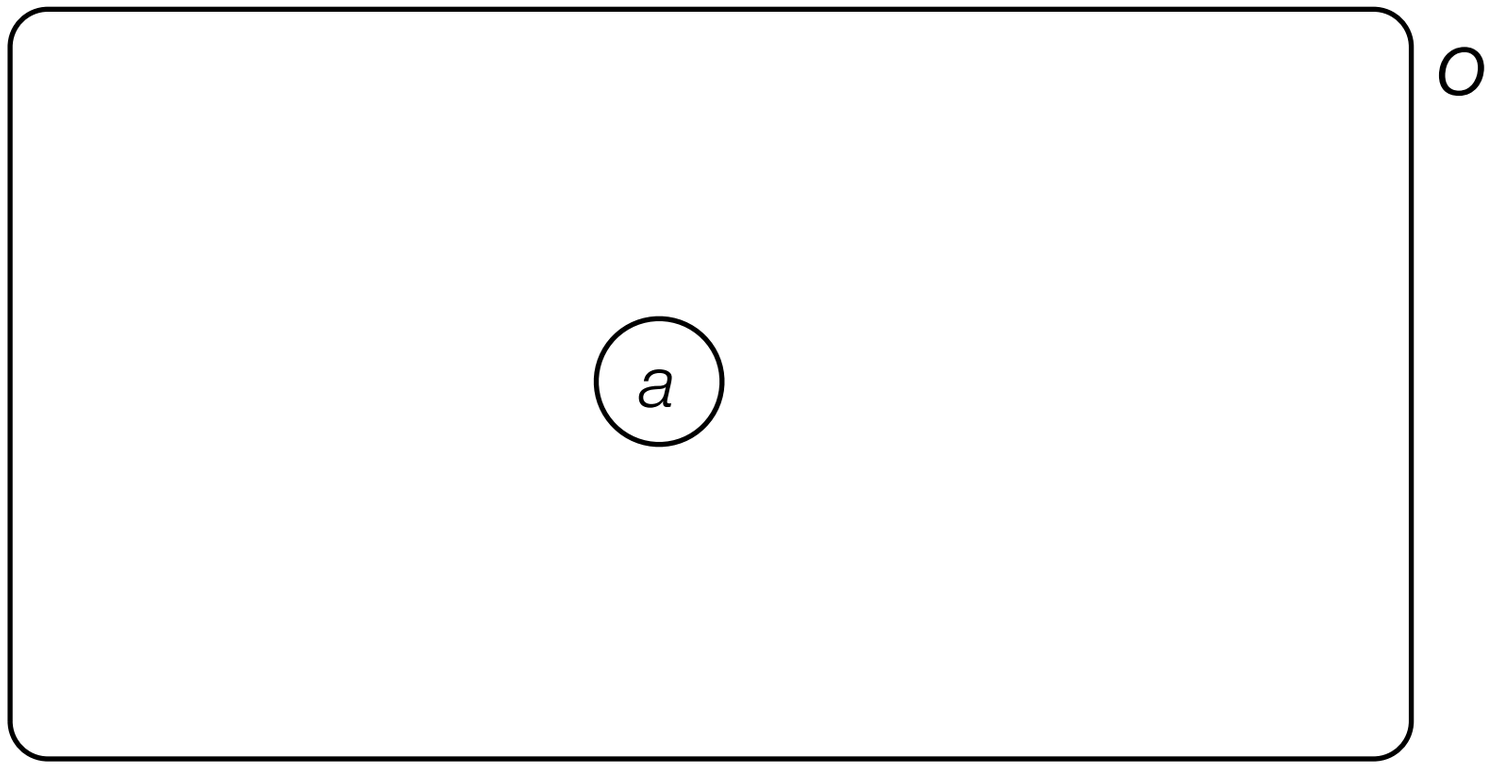
Abstractly spoken

Ambient reference $a = \langle f, M \rangle$



Abstractly spoken

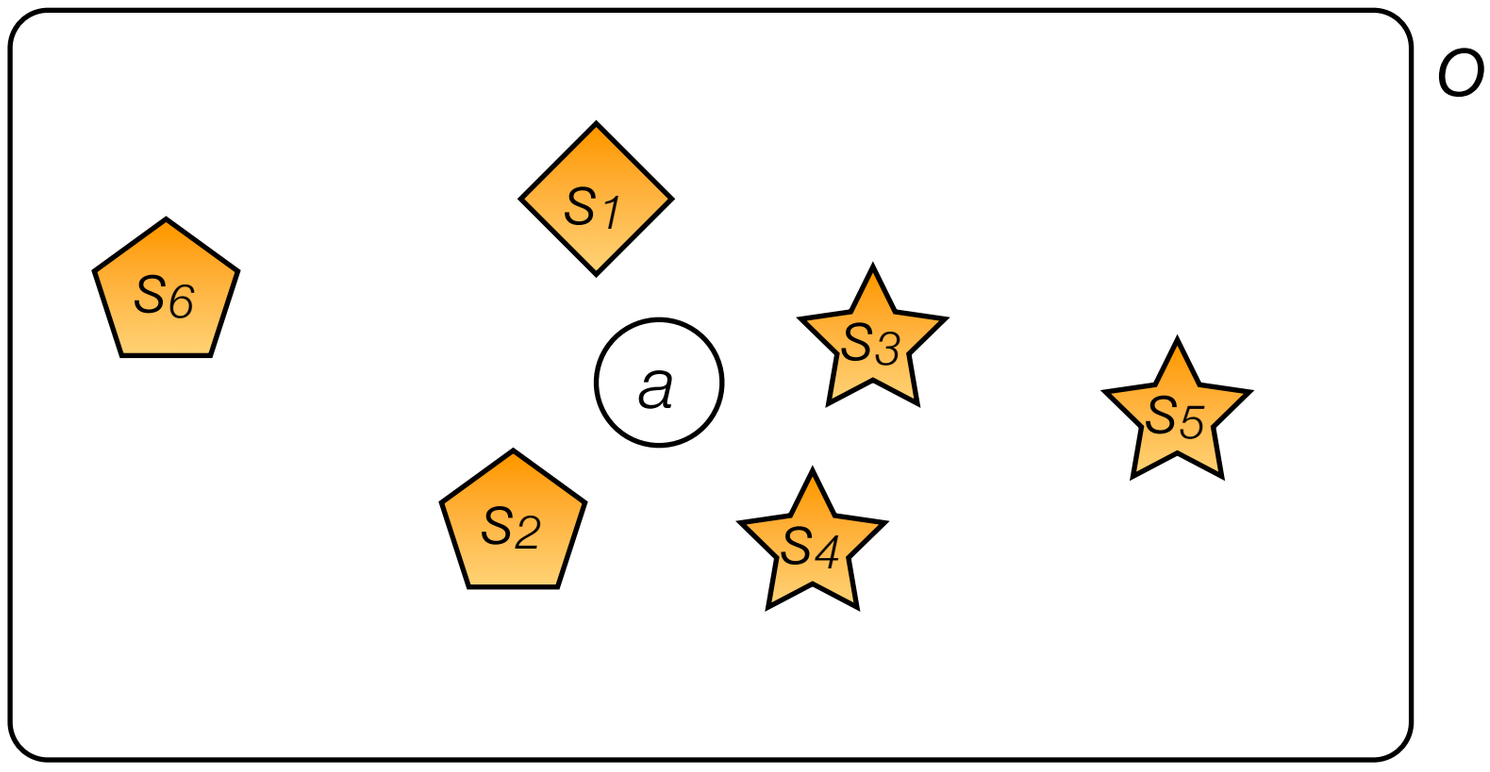
Ambient reference $a = \langle f, M \rangle$



Universe *O*

Abstractly spoken

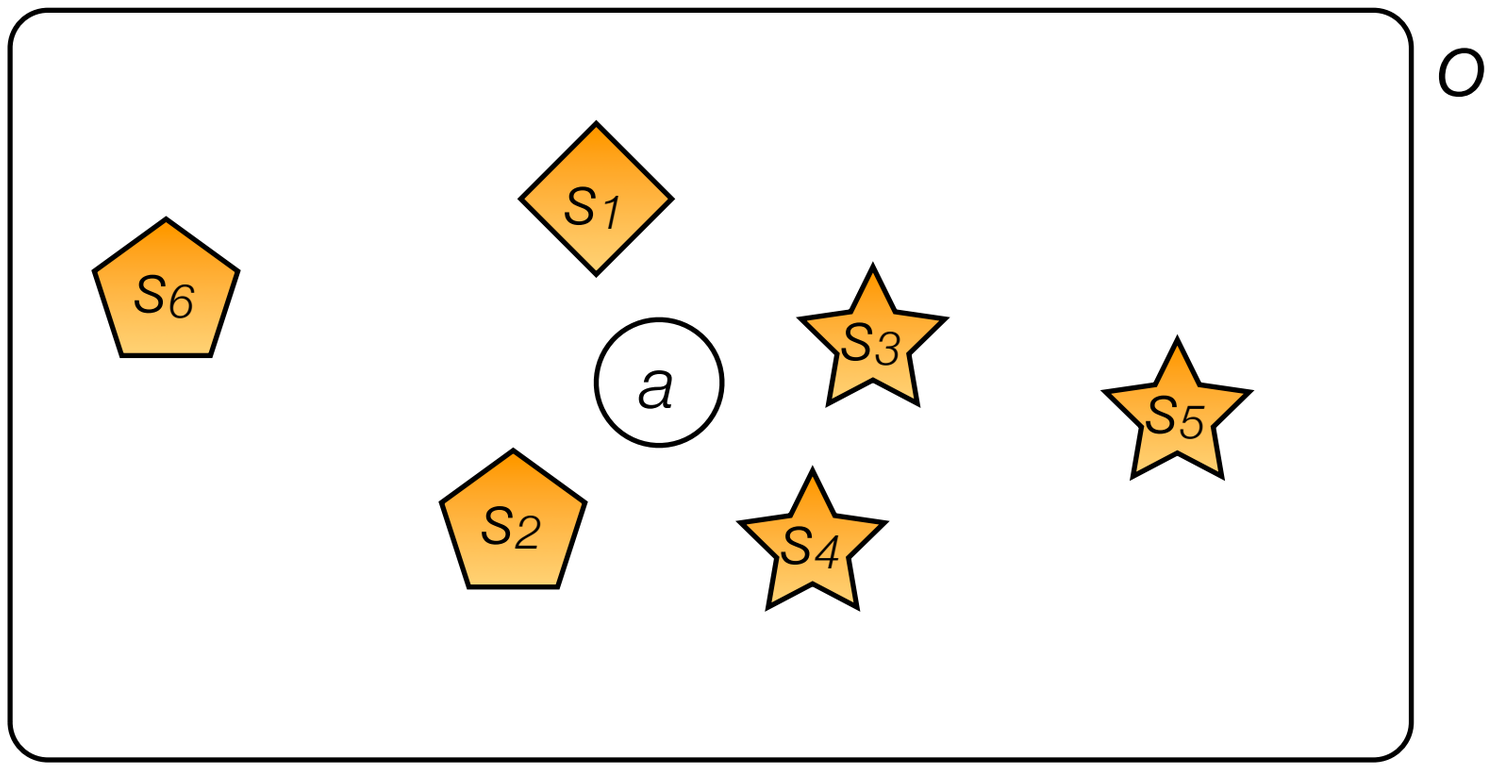
Ambient reference $a = \langle f, M \rangle$



Universe *O*

Abstractly spoken

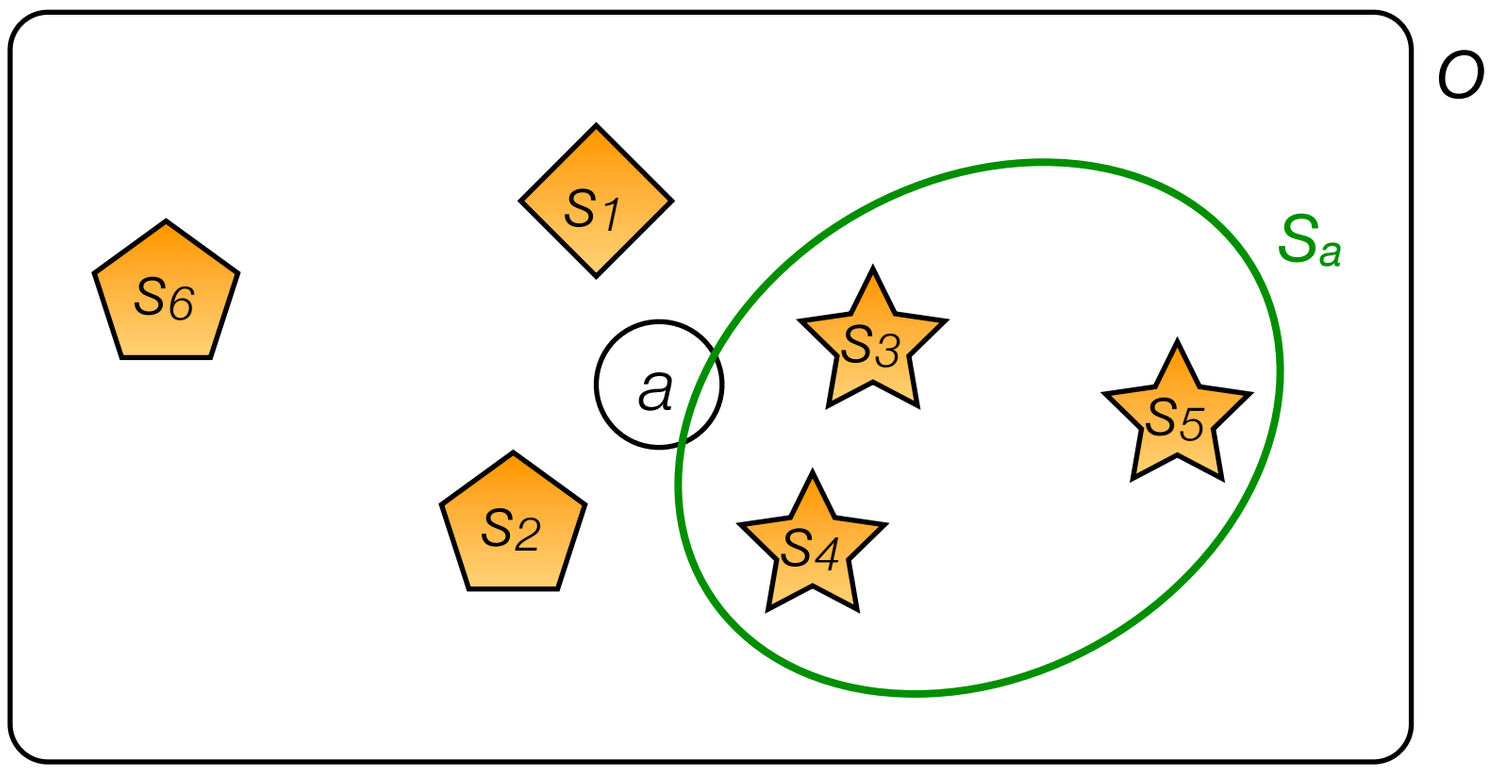
Ambient reference $a = \langle f, M \rangle$ $f_a(s) = s \equiv$ 



Universe *O*

Abstractly spoken

Ambient reference $a = \langle f, M \rangle$ $f_a(s) = s \equiv$ 

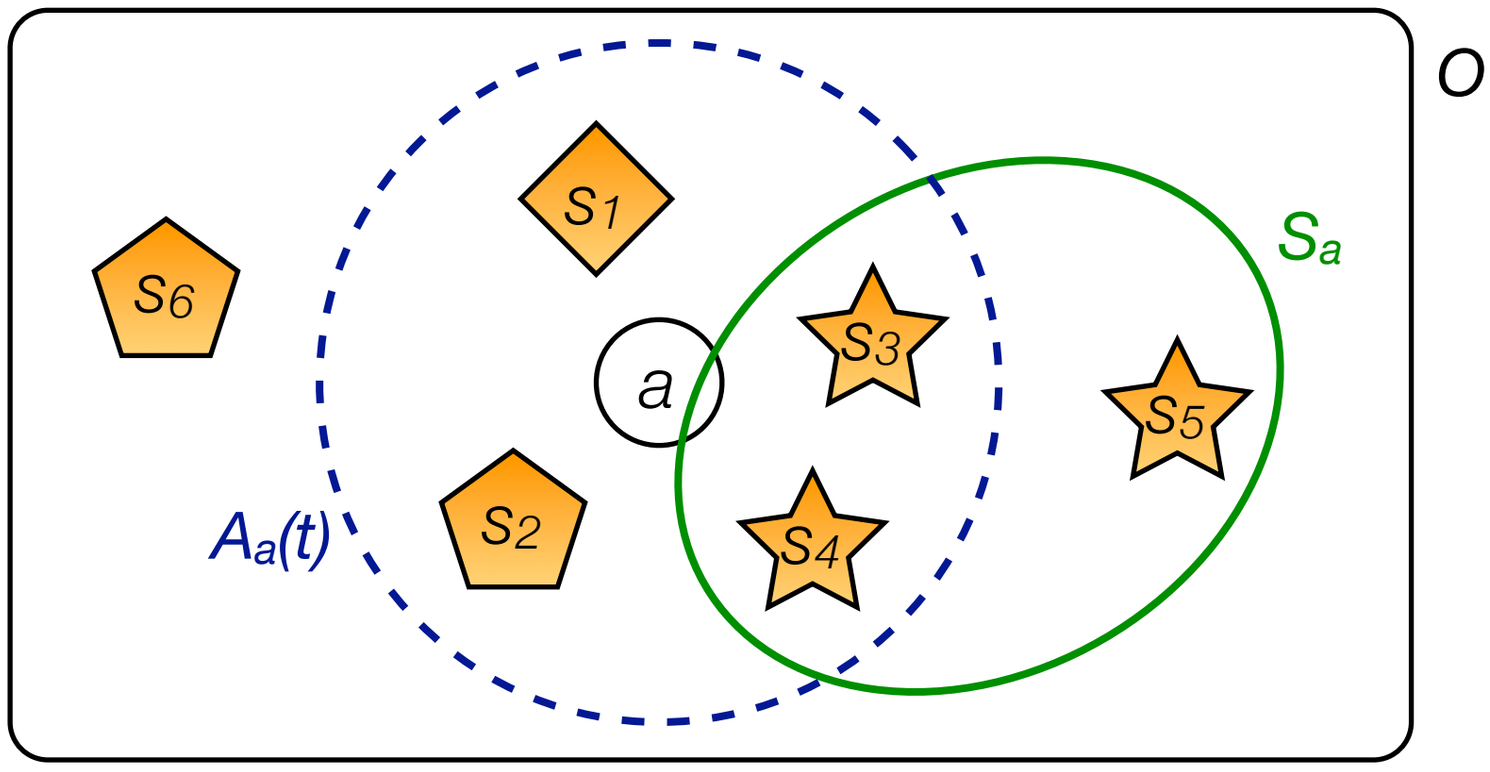


Universe O

Scope $S_a = \{ s \in O \mid f_a(s) \}$

Abstractly spoken

Ambient reference $a = \langle f, M \rangle$ $f_a(s) = s \equiv$ ★



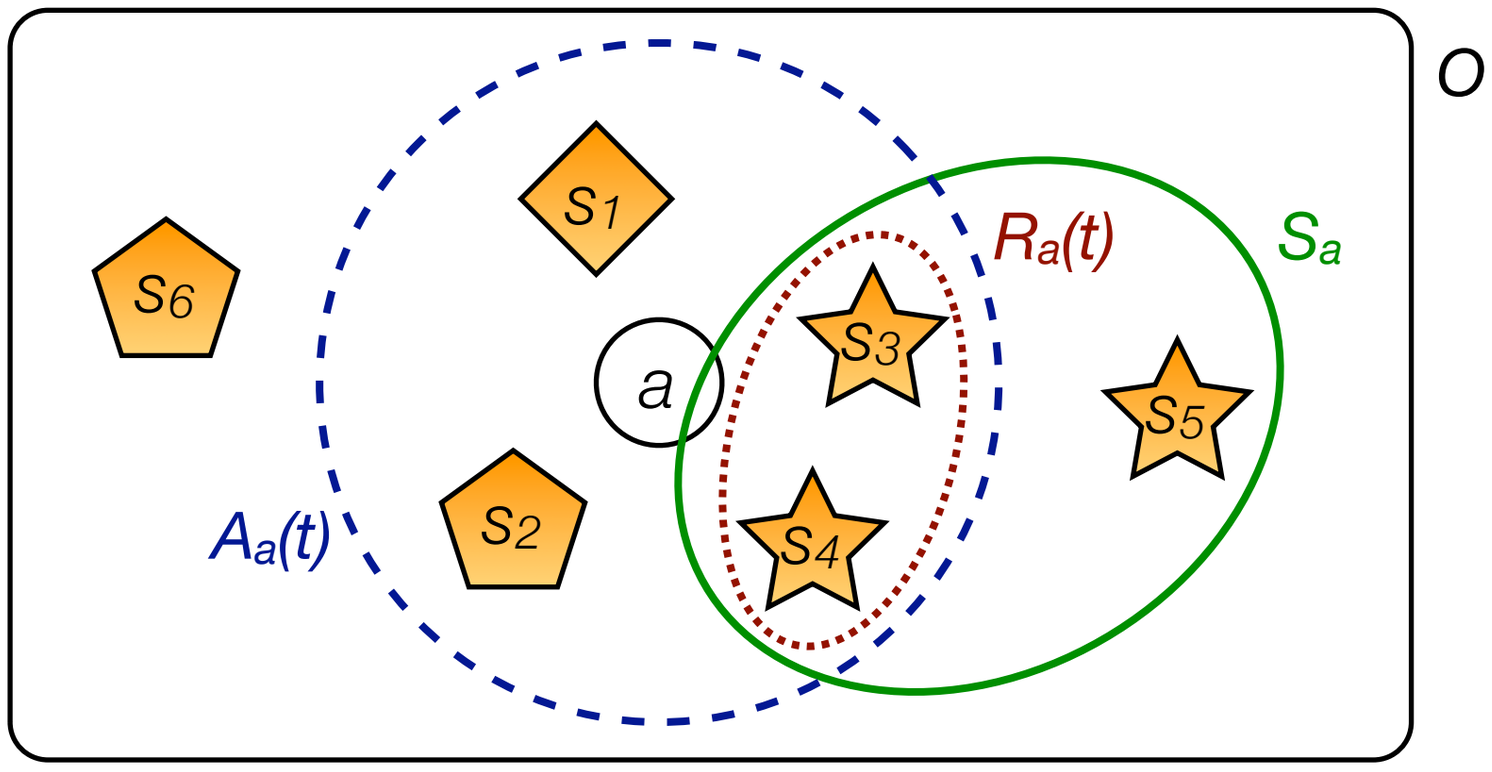
Universe O

Scope $S_a = \{ s \in O \mid f_a(s) \}$

Range $A_a(t)$ = all objects in communication range at time t

Abstractly spoken

Ambient reference $a = \langle f, M \rangle$ $f_a(s) = s \equiv$ ★



Universe O

Scope $S_a = \{ s \in O \mid f_a(s) \}$

Range $A_a(t)$ = all objects in communication range at time t

Reach $R_a(t) = S_a \cap A_a(t)$

Concretely spoken


```
def players := ambient: Player where: { |p| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
```

Concretely spoken

Scope

```
def players := ambient: Player where: { |pl| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
```

Concretely spoken

Type tags

```
deftype Player;
```

Scope

```
def players := ambient: Player where: { |pl| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
```

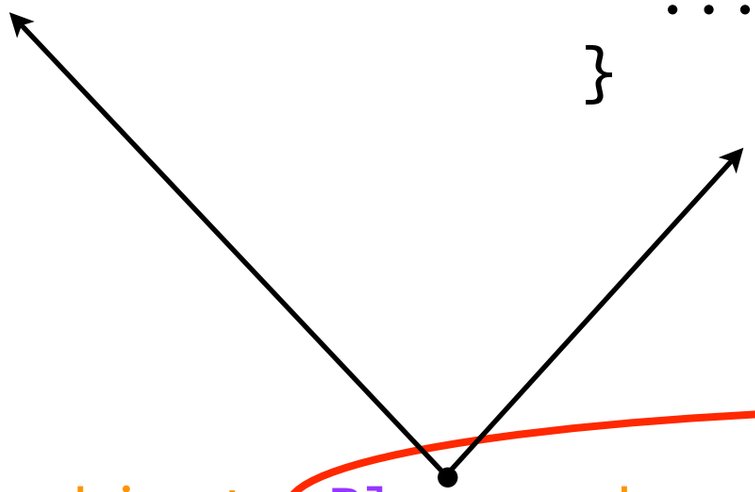
Concretely spoken

Type tags

```
deftype Player;
```

Protocols

```
def Player := protocol: {
  def askToVote(question);
  ...
}
```



Scope

```
def players := ambient: Player where: { |pl| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
```

Concretely spoken


```
def players := ambient: Player where: { |pl| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
```

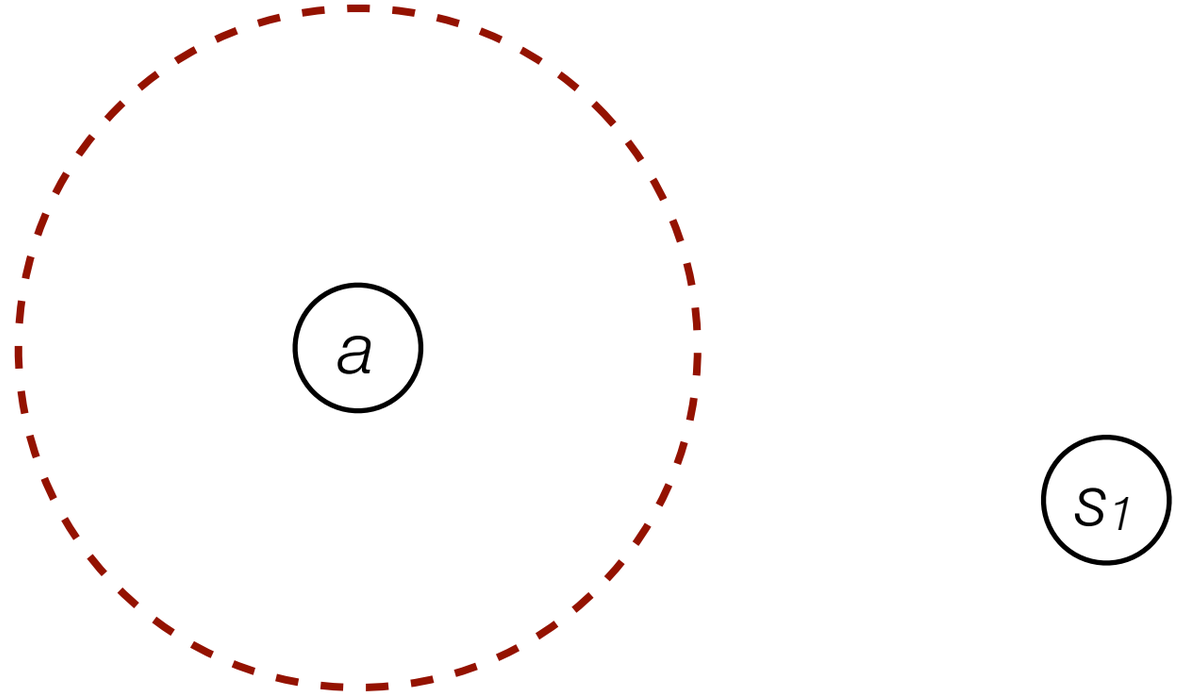
Eventual reference

Concretely spoken


```
def players := ambient: Player where: { |p| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
```

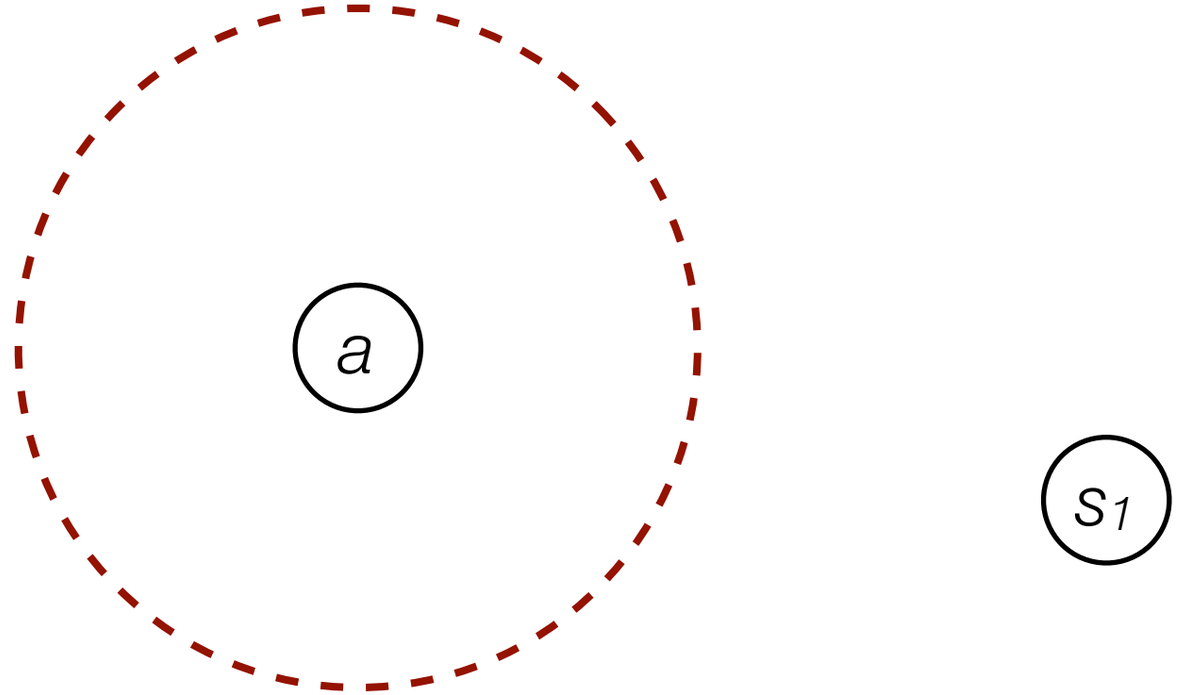
Delivery policies

Example: Location Tracker



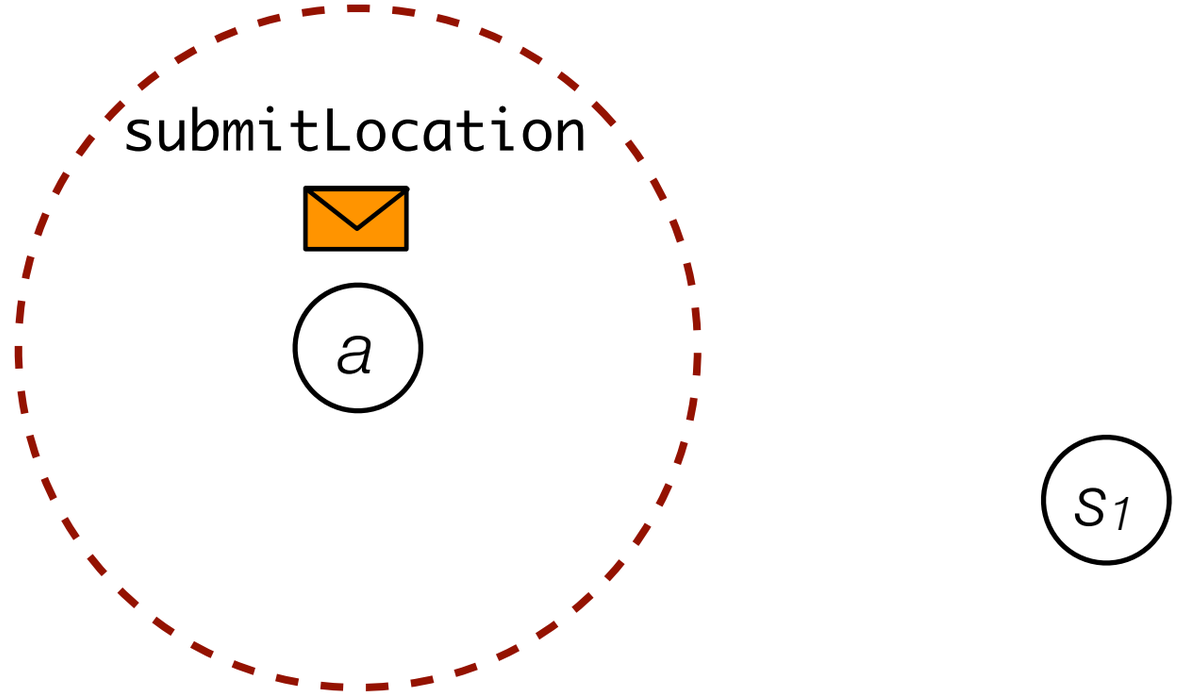

```
def ap := ambient: LocationService;
def updateLocation(newLoc) {
   ap<-submitLocation(id,newLoc)@[One,Instant,Oneway];
}
```

Example: Location Tracker



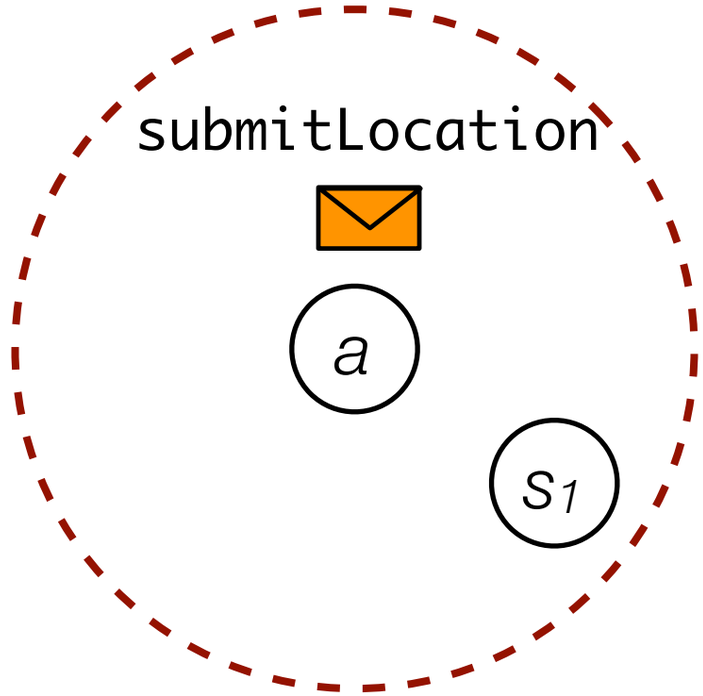

```
def ap := ambient: LocationService;
def updateLocation(newLoc) {
   ap<-submitLocation(id,newLoc)@[One,Instant,Oneway];
}
```

Example: Location Tracker



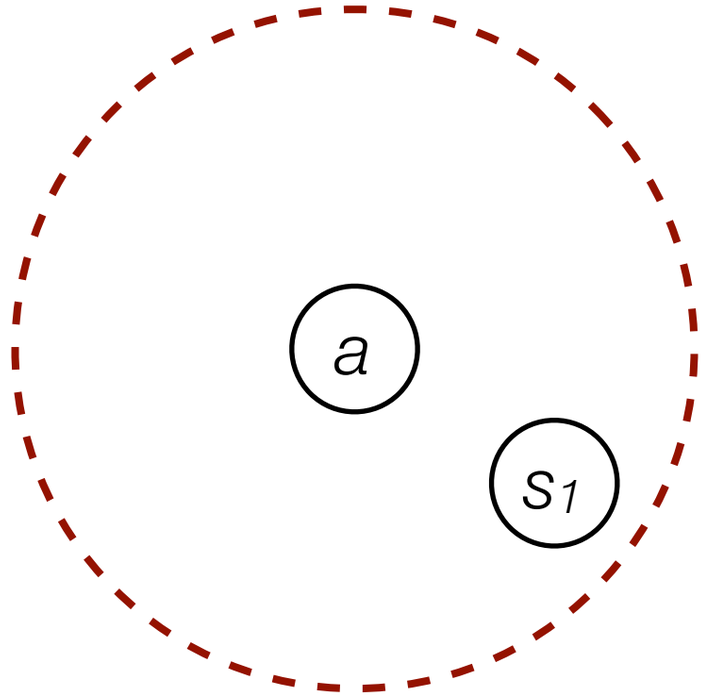

```
def ap := ambient: LocationService;
def updateLocation(newLoc) {
   ap<-submitLocation(id,newLoc)@[One,Sustain,Oneway];
}
```

Example: Location Tracker



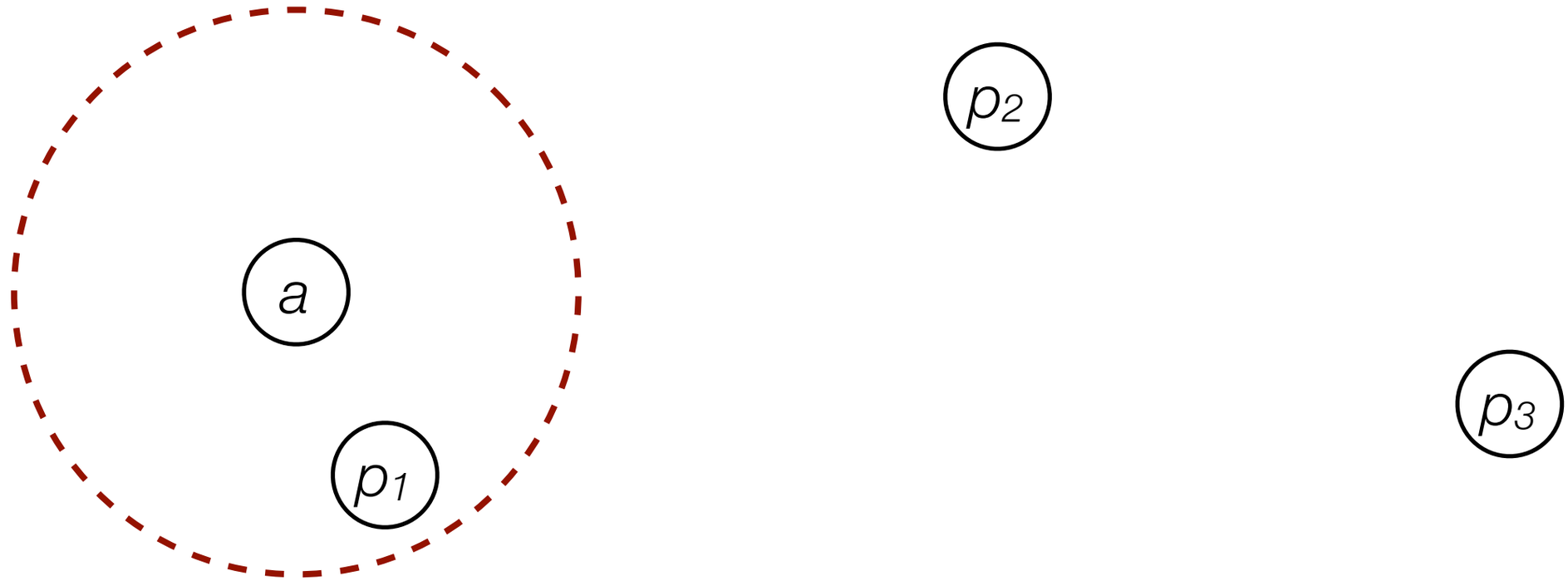

```
def ap := ambient: LocationService;
def updateLocation(newLoc) {
   ap<-submitLocation(id,newLoc)@[One,Sustain,Oneway];
}
```

Example: Location Tracker



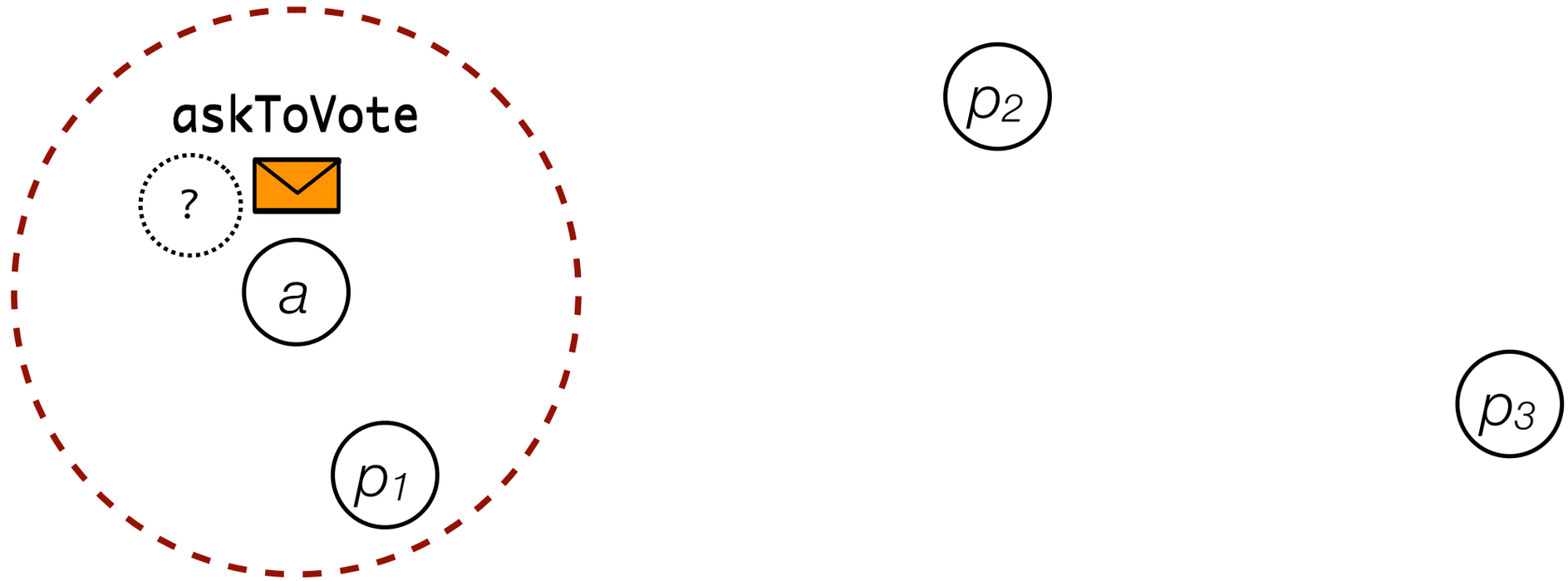

```
def ap := ambient: LocationService;
def updateLocation(newLoc) {
   ap<-submitLocation(id,newLoc)@[One,Sustain,Oneway];
}
```

Example: Voting



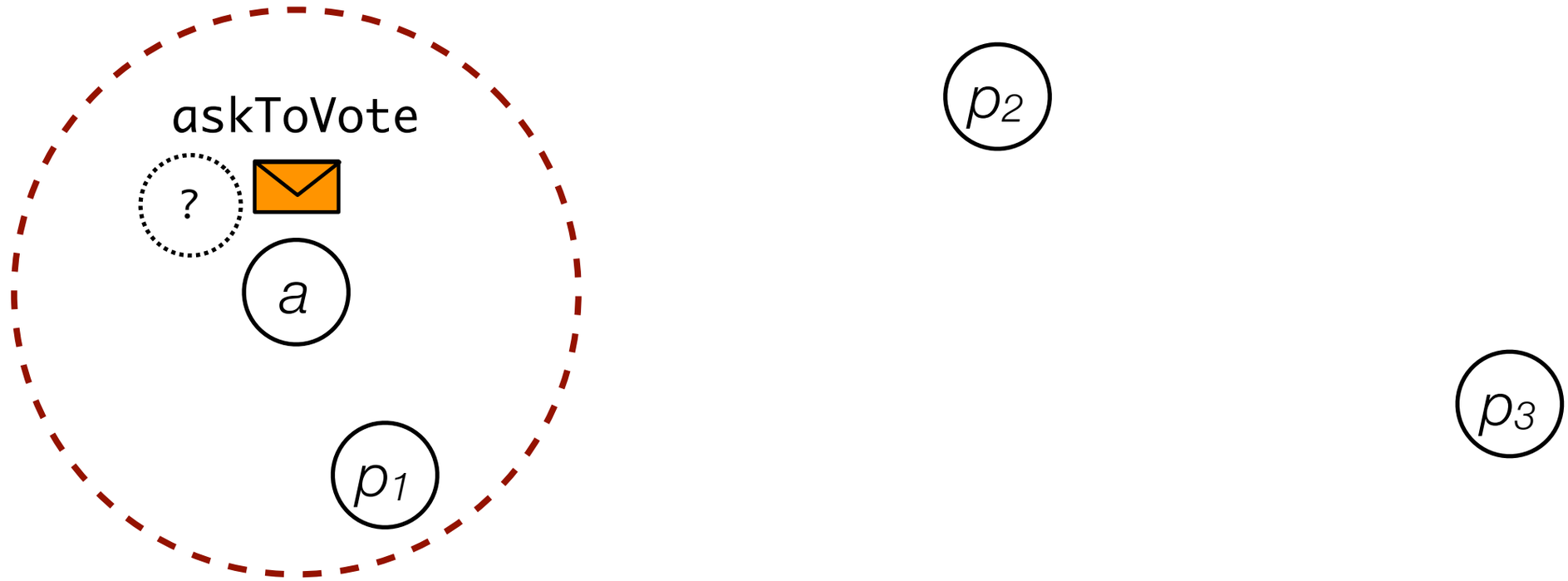

```
def players := ambient: Player where: { |p| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
whenAll: handle.future resolved: { |votes|
  // process the votes
} ruined: { |exceptions|
  // ignore faulty players
}
```

Example: Voting



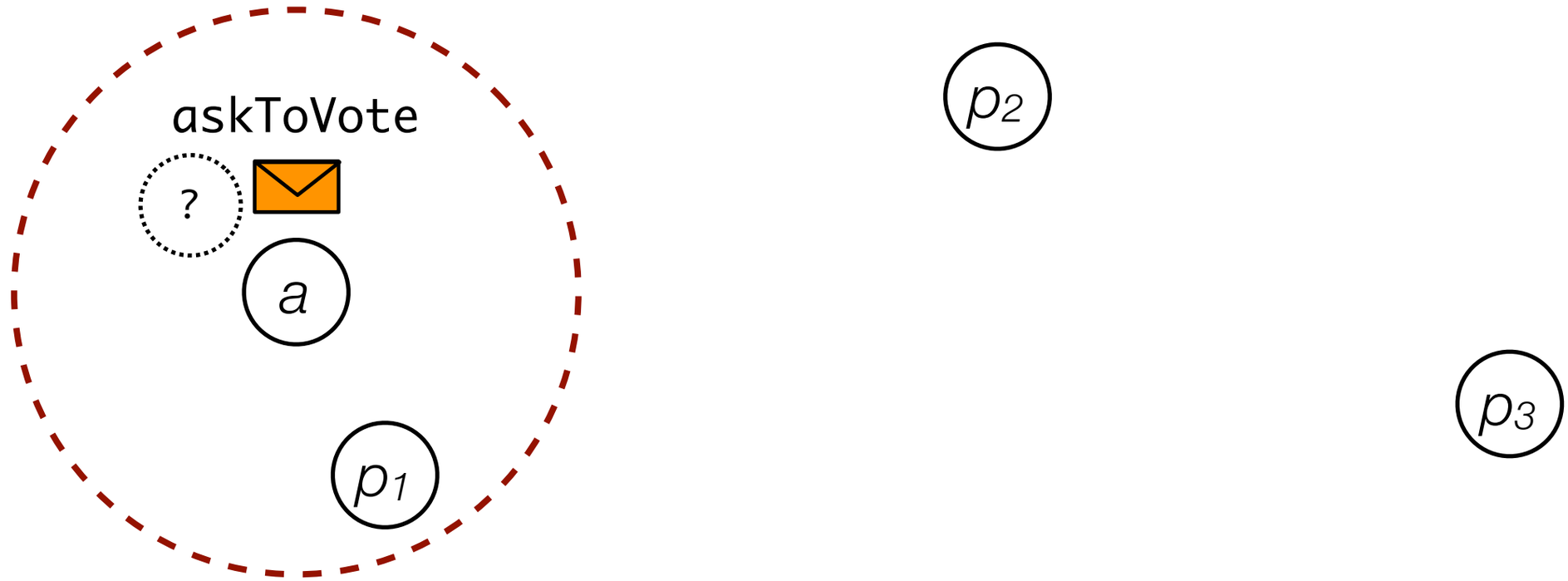

```
def players := ambient: Player where: { |p| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
whenAll: handle.future resolved: { |votes|
  // process the votes
} ruined: { |exceptions|
  // ignore faulty players
}
```

Example: Voting



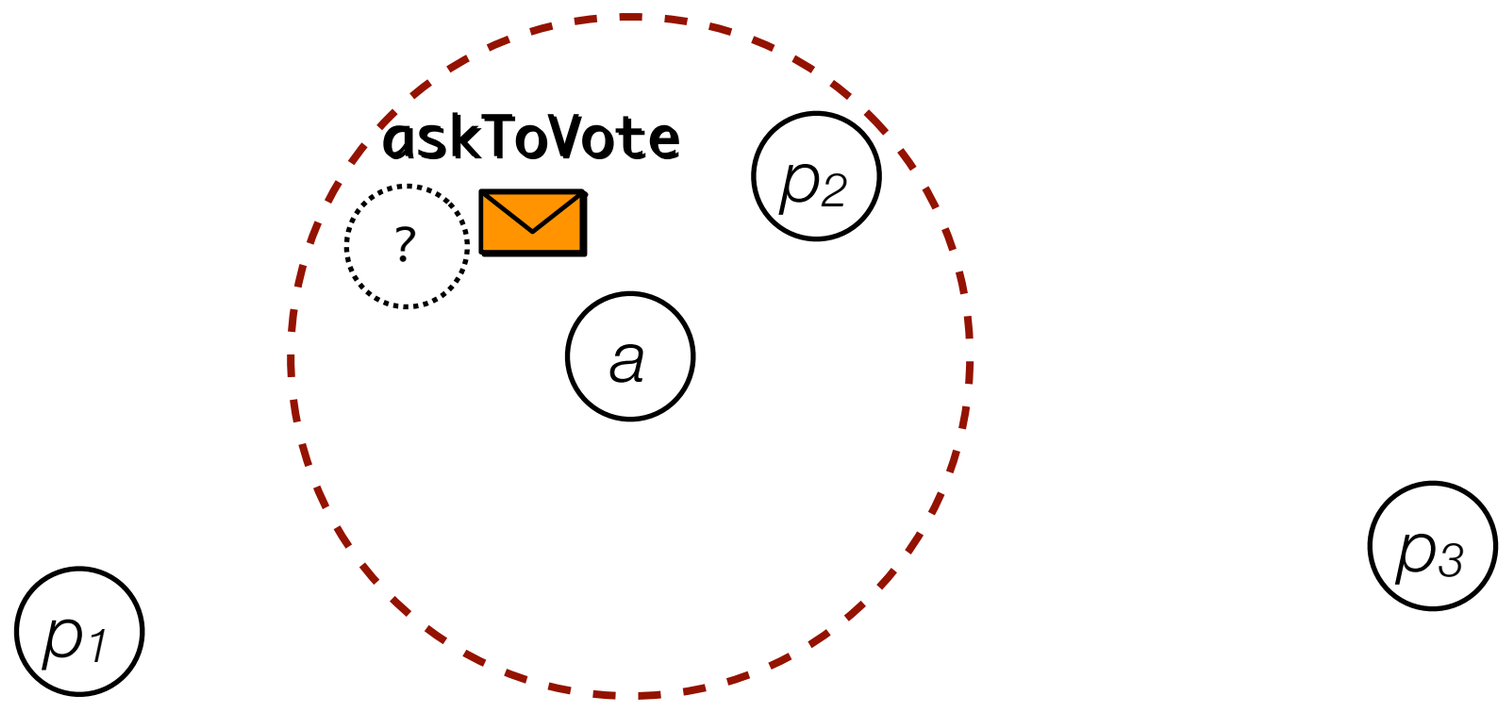

```
def players := ambient: Player where: { |p| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
whenAll: handle.future resolved: { |votes|
  // process the votes
} ruined: { |exceptions|
  // ignore faulty players
}
```

Example: Voting



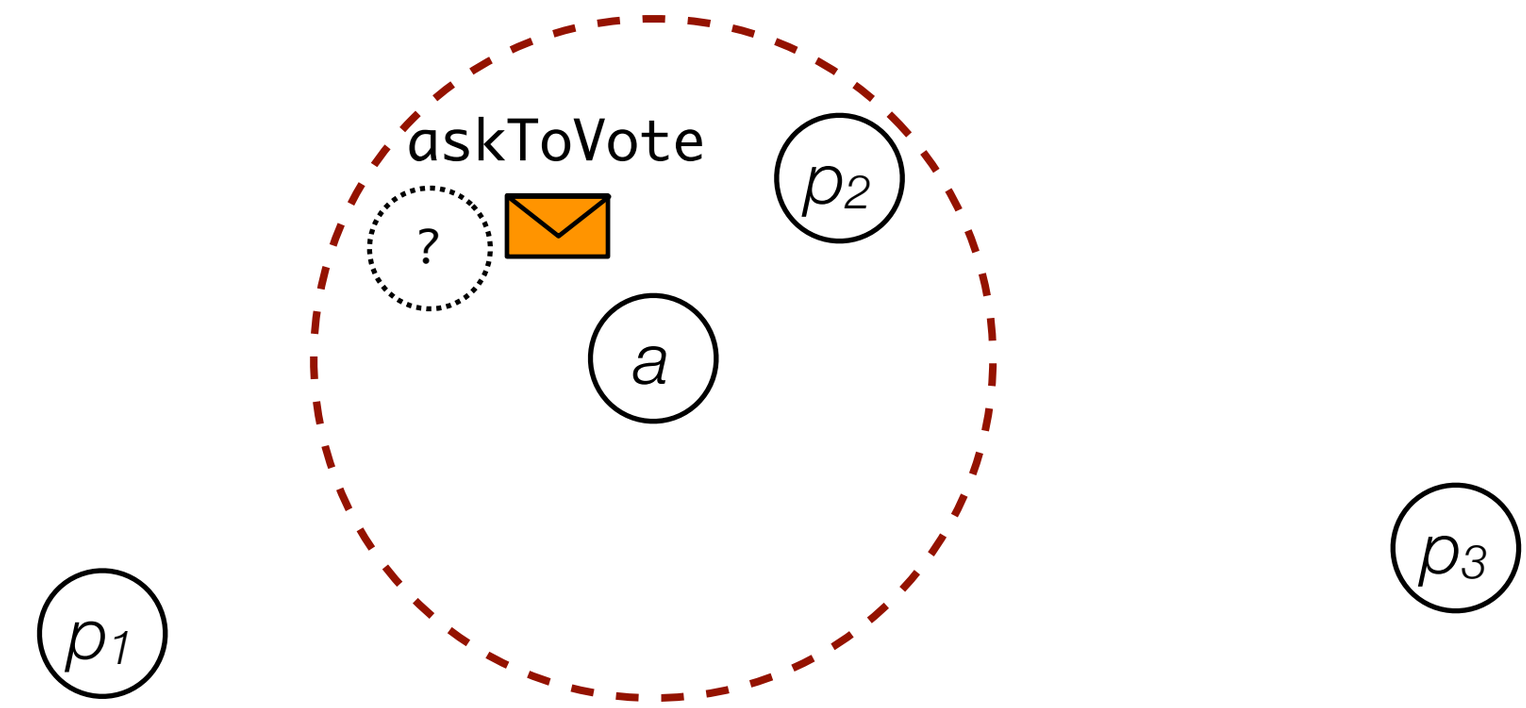

```
def players := ambient: Player where: { |p| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
whenAll: handle.future resolved: { |votes|
  // process the votes
} ruined: { |exceptions|
  // ignore faulty players
}
```

Example: Voting



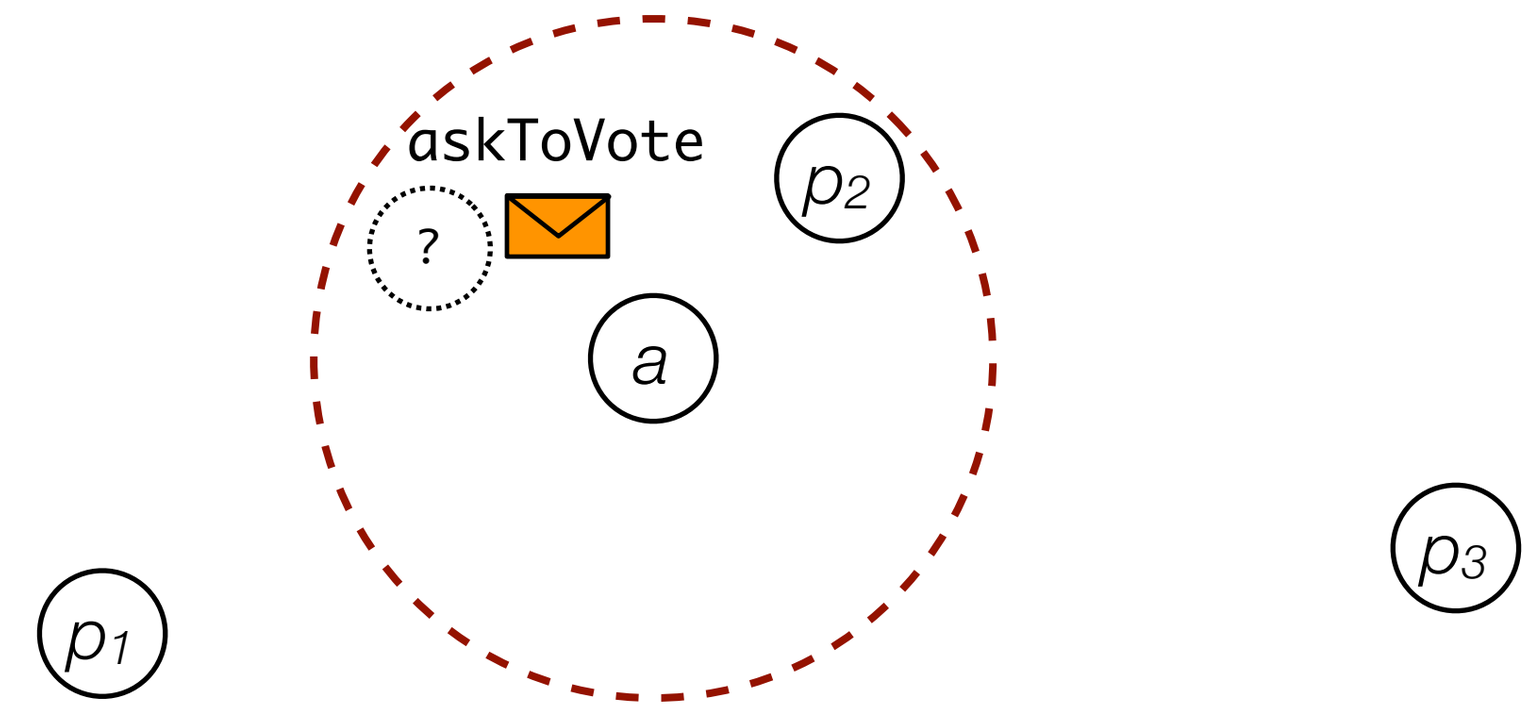

```
def players := ambient: Player where: { |p| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
whenAll: handle.future resolved: { |votes|
  // process the votes
} ruined: { |exceptions|
  // ignore faulty players
}
```

Example: Voting



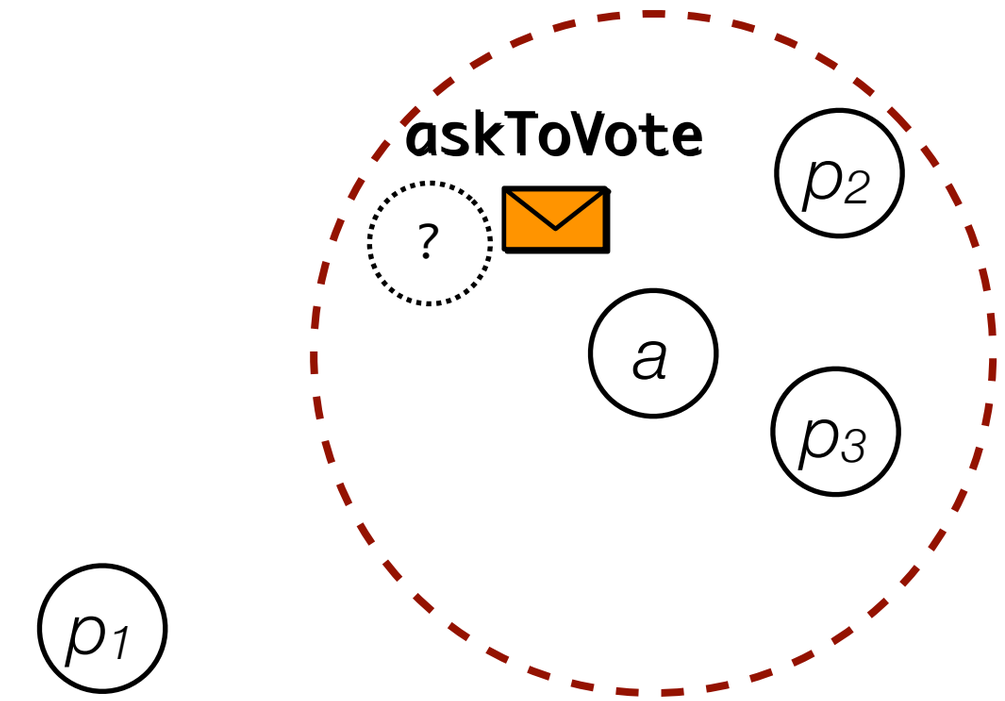

```
def players := ambient: Player where: { |p| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
whenAll: handle.future resolved: { |votes|
  // process the votes
} ruined: { |exceptions|
  // ignore faulty players
}
```

Example: Voting



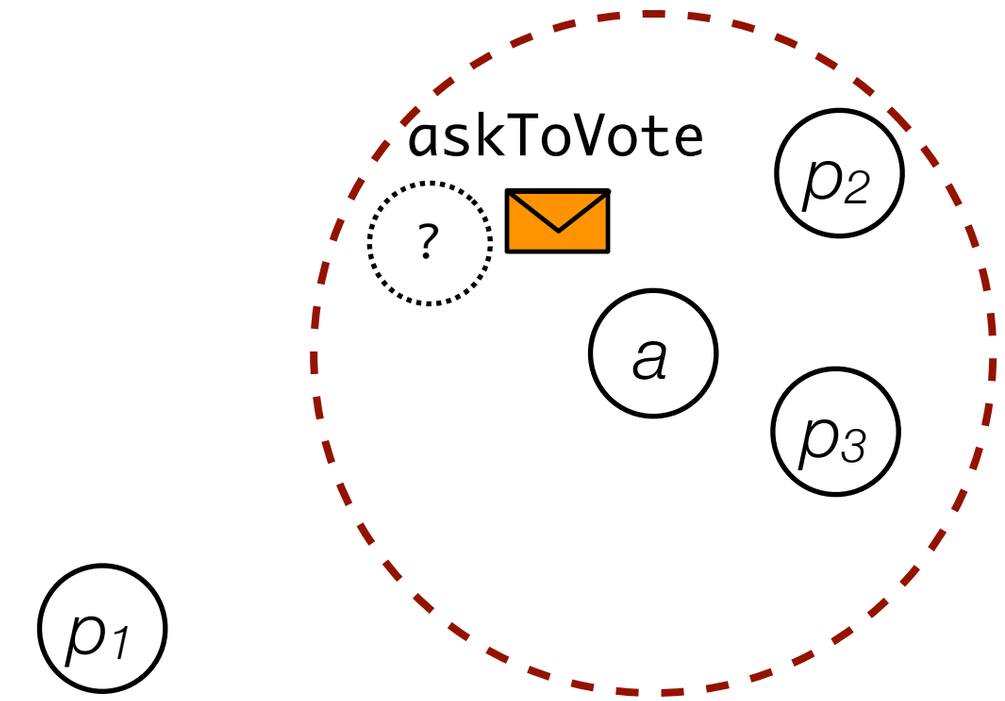

```
def players := ambient: Player where: { |p| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
whenAll: handle.future resolved: { |votes|
  // process the votes
} ruined: { |exceptions|
  // ignore faulty players
}
```

Example: Voting



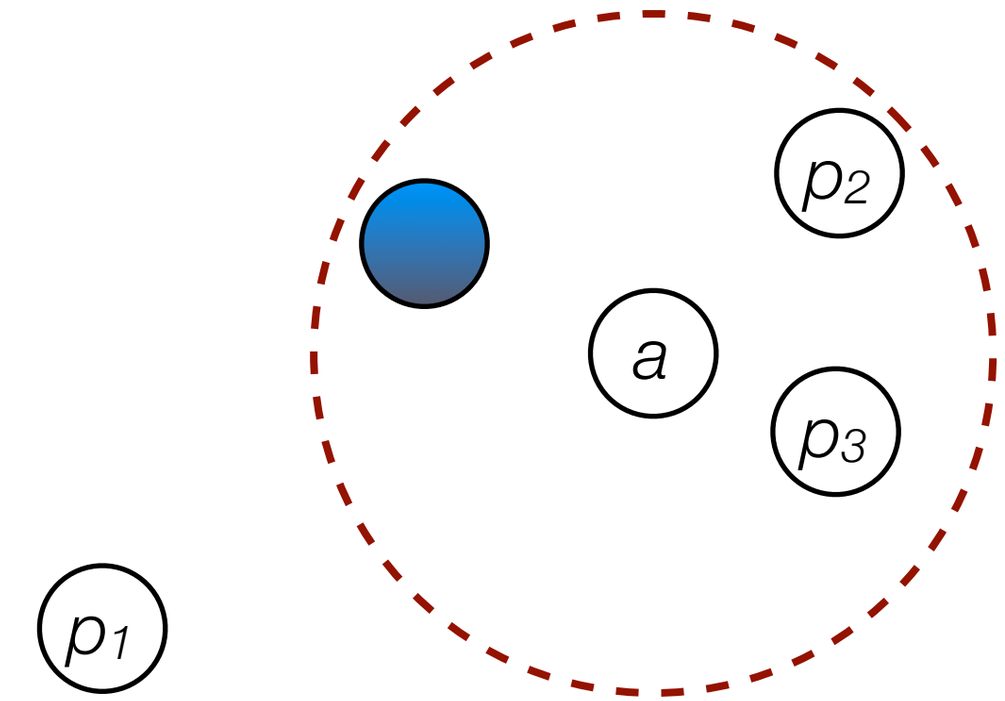

```
def players := ambient: Player where: { |p| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
whenAll: handle.future resolved: { |votes|
  // process the votes
} ruined: { |exceptions|
  // ignore faulty players
}
```

Example: Voting



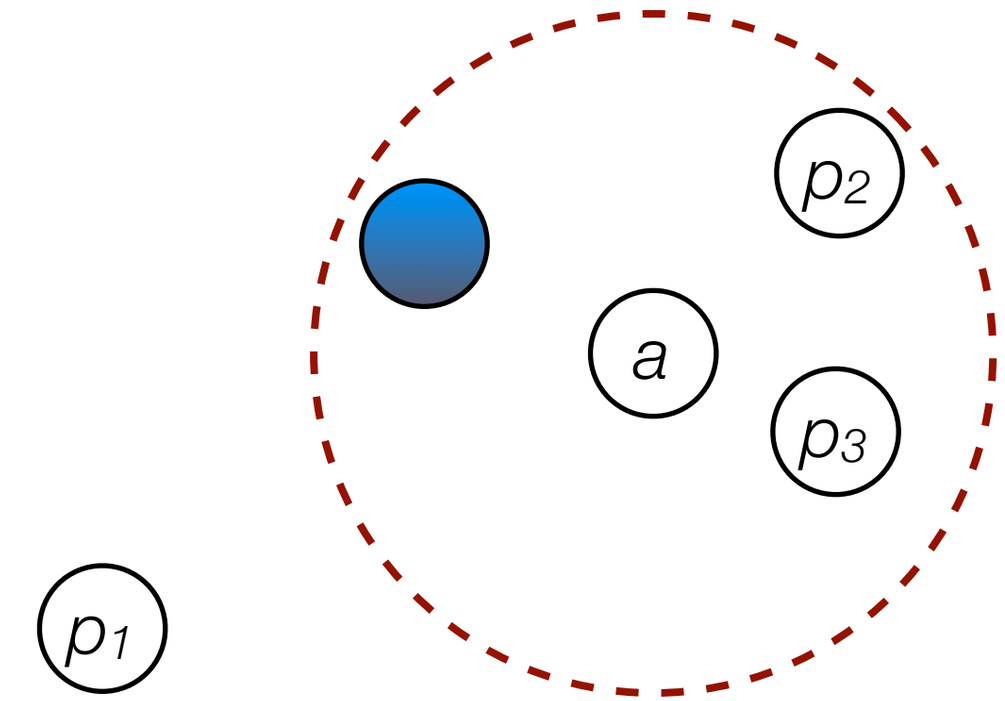

```
def players := ambient: Player where: { |p| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
whenAll: handle.future resolved: { |votes|
  // process the votes
} ruined: { |exceptions|
  // ignore faulty players
}
```

Example: Voting



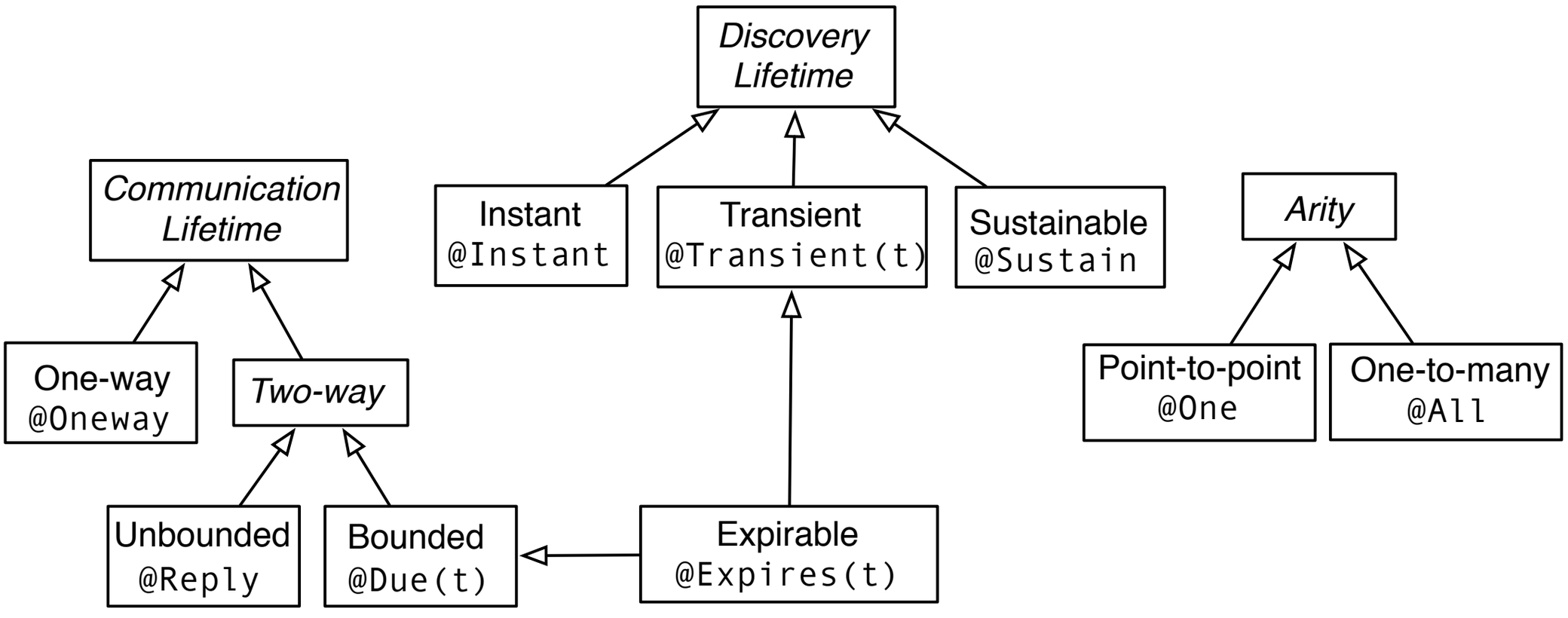

```
def players := ambient: Player where: { |p| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
whenAll: handle.future resolved: { |votes|
  // process the votes
} ruined: { |exceptions|
  // ignore faulty players
}
```

Example: Voting



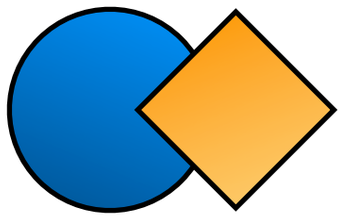

```
def players := ambient: Player where: { |p| p.team == “blue” }
def handle := players<-askToVote(q)@[All,Expires(minutes(1))];
whenAll: handle.future resolved: { |votes|
  // process the votes
} ruined: { |exceptions|
  // ignore faulty players
}
```

Delivery Policies



Roadmap

Motivation



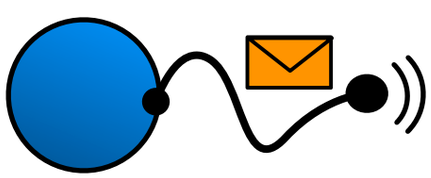
Object-event impedance mismatch

Coordination

Mobile ad hoc networks

Contribution

AMBIENTTALK



Ambient References

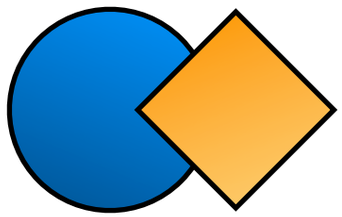
Validation

Evaluation

Implementation

Evaluation

Motivation



Object-event impedance mismatch

Issues

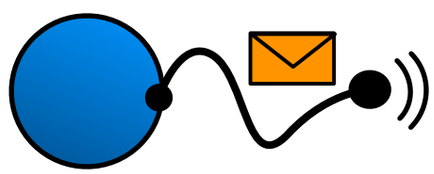
Coordination

Criteria

Contribution

Language

Language Feature



Ambient References

Validation

Evaluation

Implementation

Mobile ad hoc networks

Criteria Revisited

	AmbientTalk	Ambient References

Criteria Revisited

	AmbientTalk	Ambient References
Decentralised Discovery	`whenever:discovered:`	Can encode discovery

Criteria Revisited

	AmbientTalk	Ambient References
Decentralised Discovery	`whenever:discovered:`	Can encode discovery
Time-decoupled communication	far references + futures	`@Transient`, `@Sustain`

Criteria Revisited

	AmbientTalk	Ambient References
 **Decentralised** Discovery	`whenever:discovered:`	Can encode discovery
 **Time-decoupled** communication	far references + futures	`@Transient`, `@Sustain`
 **Synchronisation-decoupled** communication	`<-` `when:becomes:`	`<-` `whenAll:resolved:`

Criteria Revisited

	AmbientTalk	Ambient References
Decentralised Discovery	`whenever:discovered:`	Can encode discovery
Time-decoupled communication	far references + futures	`@Transient`, `@Sustain`
Synchronisation-decoupled communication	`<- when:becomes:`	`<- whenAll:resolved:`
Space-decoupled communication		`ambient:` description

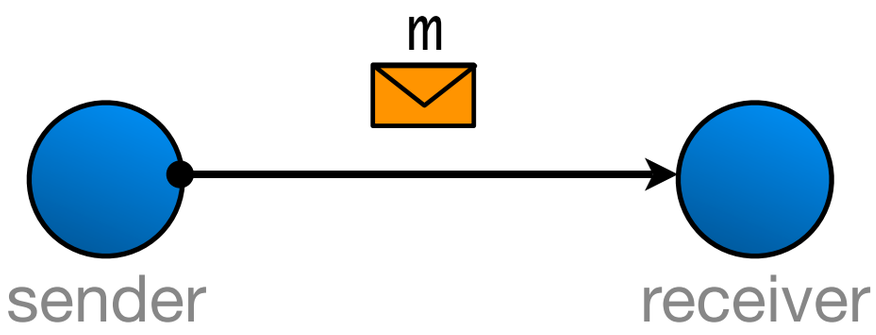
Criteria Revisited

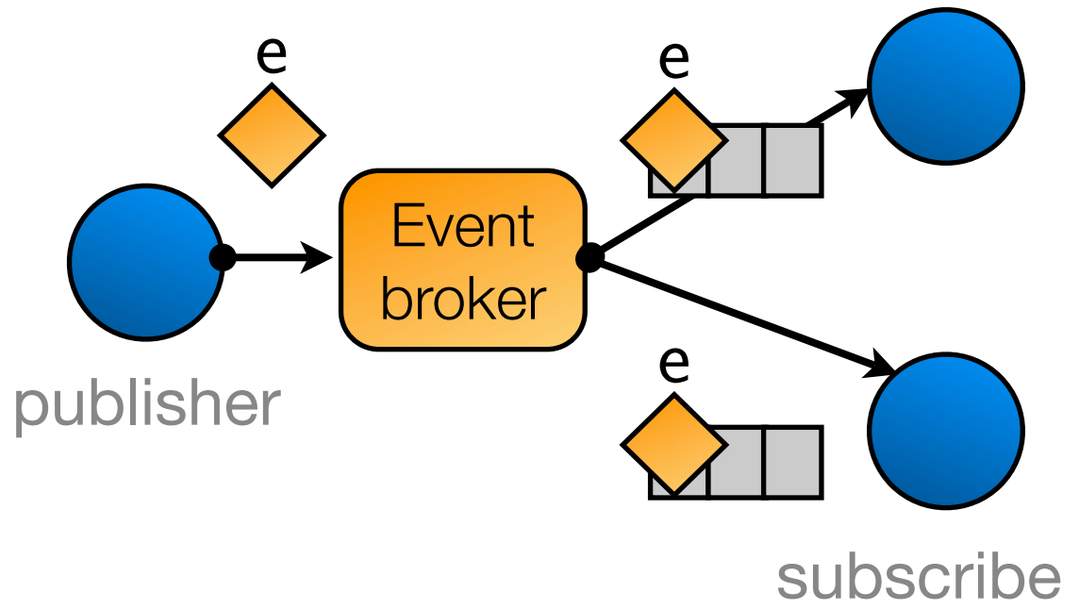
	AmbientTalk	Ambient References
Decentralised Discovery	`whenever:discovered:`	Can encode discovery
Time-decoupled communication	far references + futures	`@Transient`, `@Sustain`
Synchronisation-decoupled communication	`<- when:becomes:`	`<- whenAll:resolved:`
Space-decoupled communication		`ambient:` description
Arity-decoupled communication		`@All`

Criteria Revisited

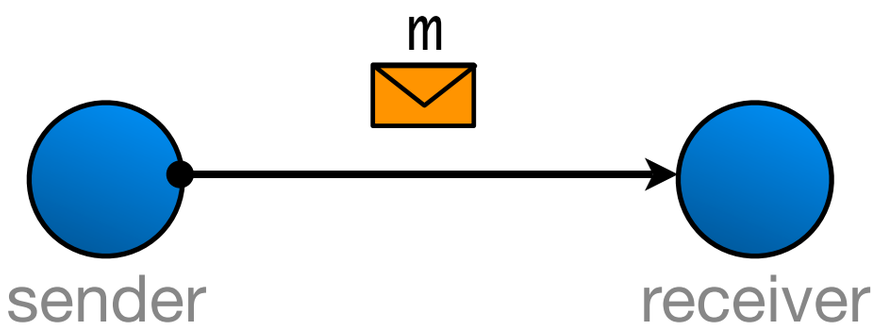
	AmbientTalk	Ambient References
Decentralised Discovery	`whenever:discovered:`	Can encode discovery
Time-decoupled communication	far references + futures	`@Transient`, `@Sustain`
Synchronisation-decoupled communication	`<-` `when:becomes:`	`<-` `whenAll:resolved:`
Space-decoupled communication		`ambient:` description
Arity-decoupled communication		`@All`
Connection-independent Failure handling	`@Due`, Leasing	`@Expires`

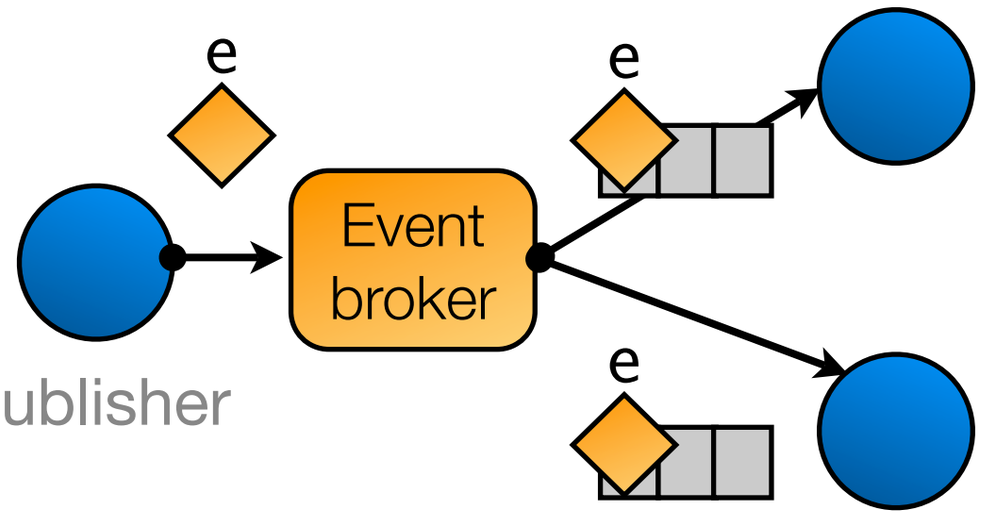
O/E Impedance Mismatch Revisited

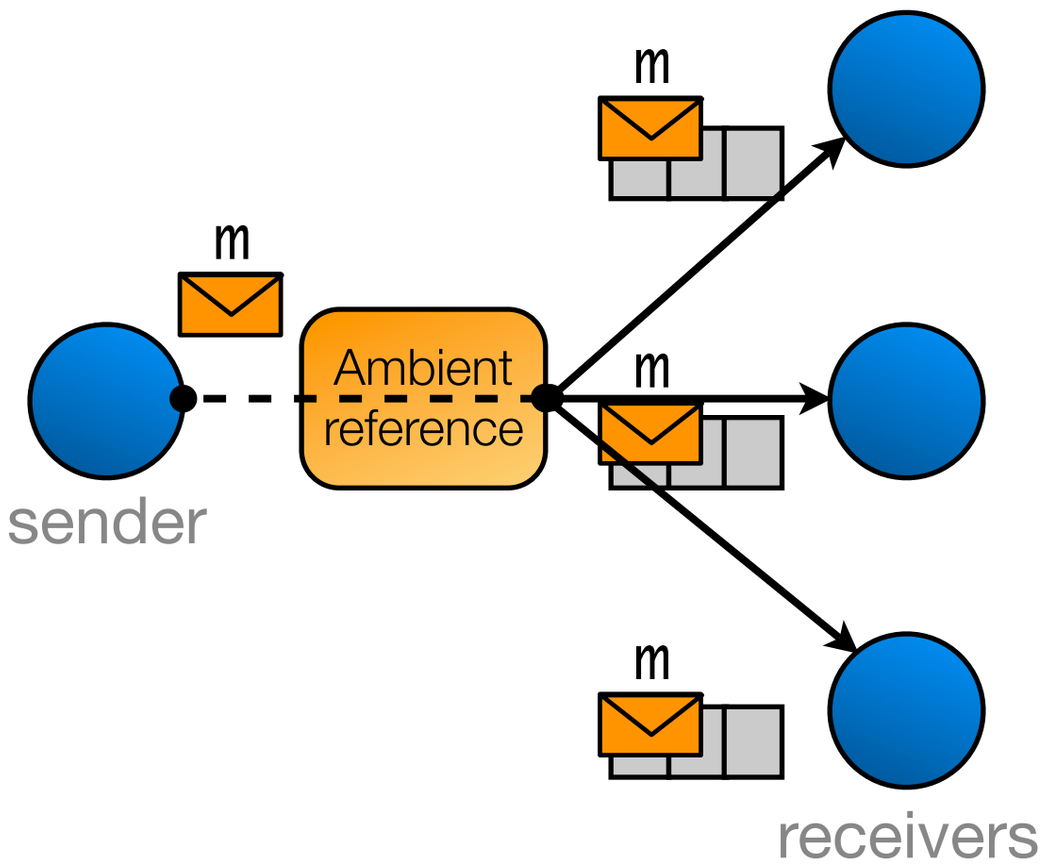




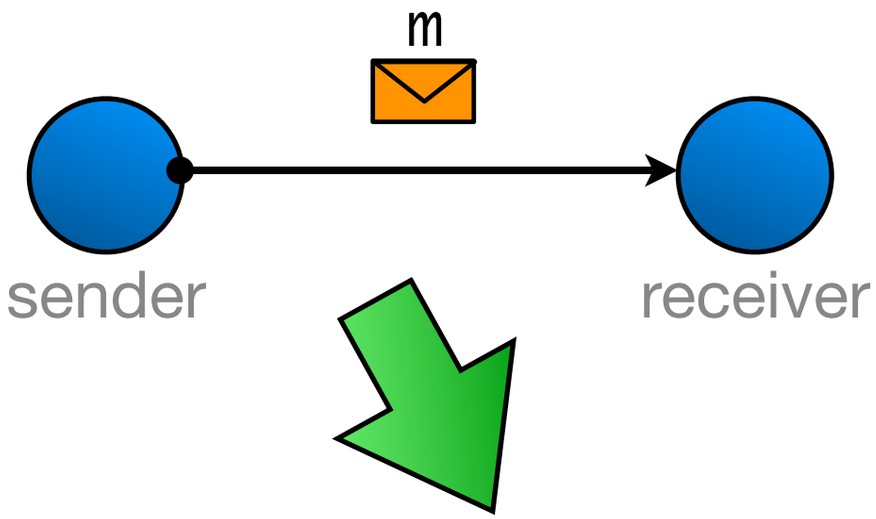
O/E Impedance Mismatch Revisited

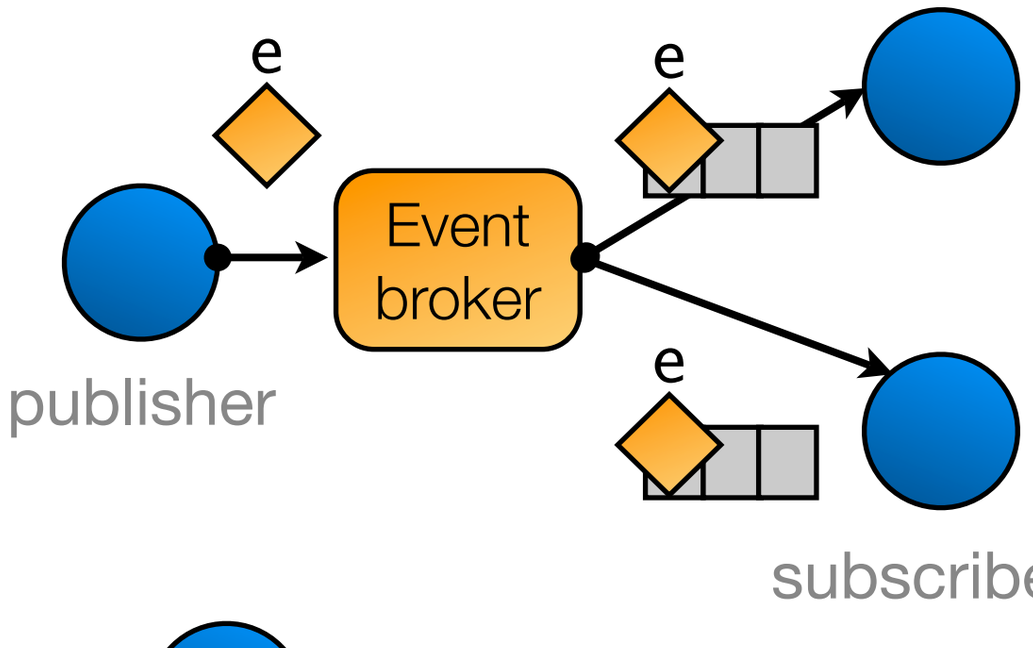


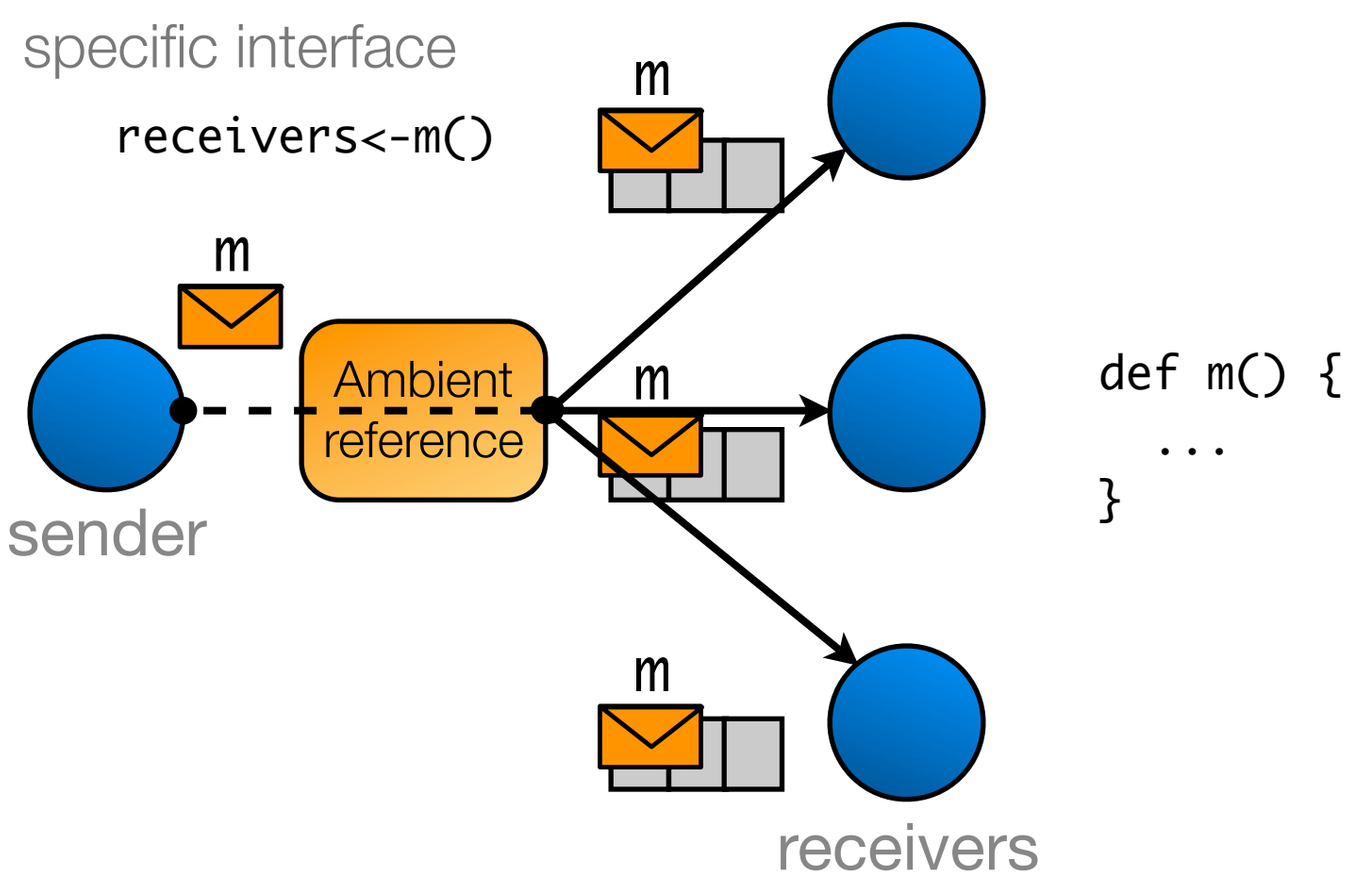




O/E Impedance Mismatch Revisited

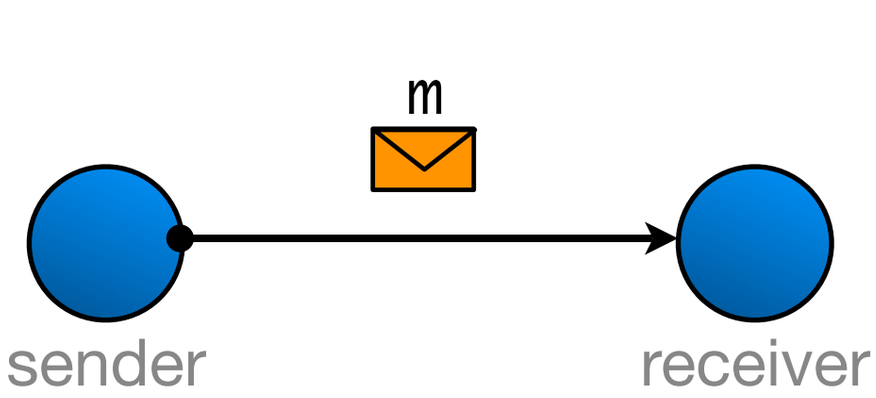


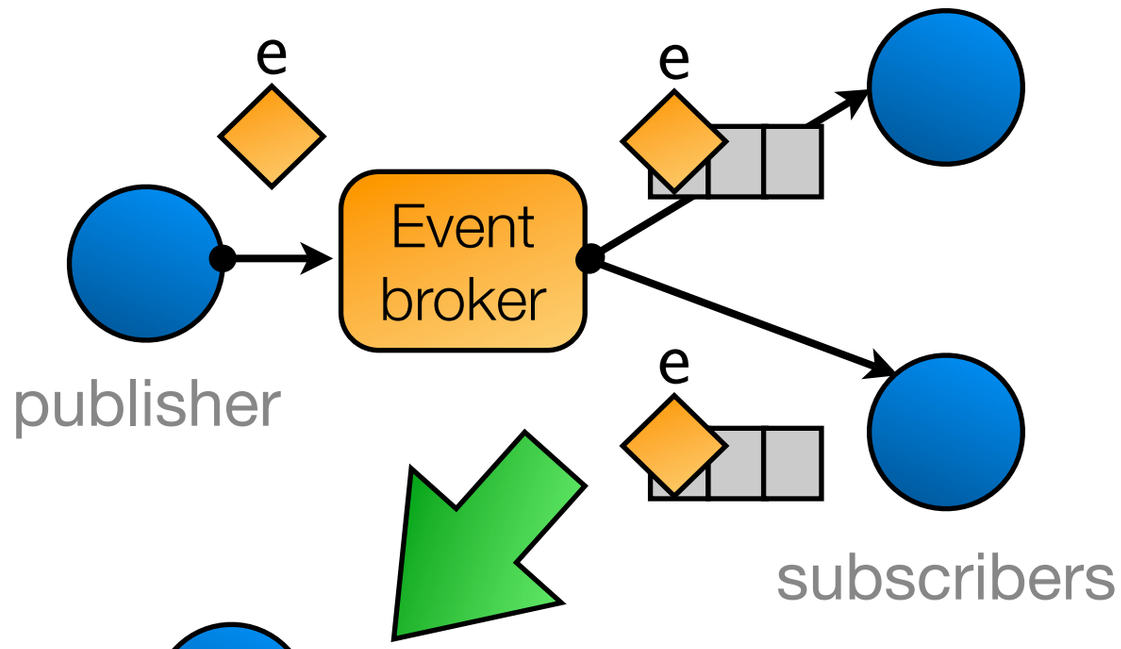


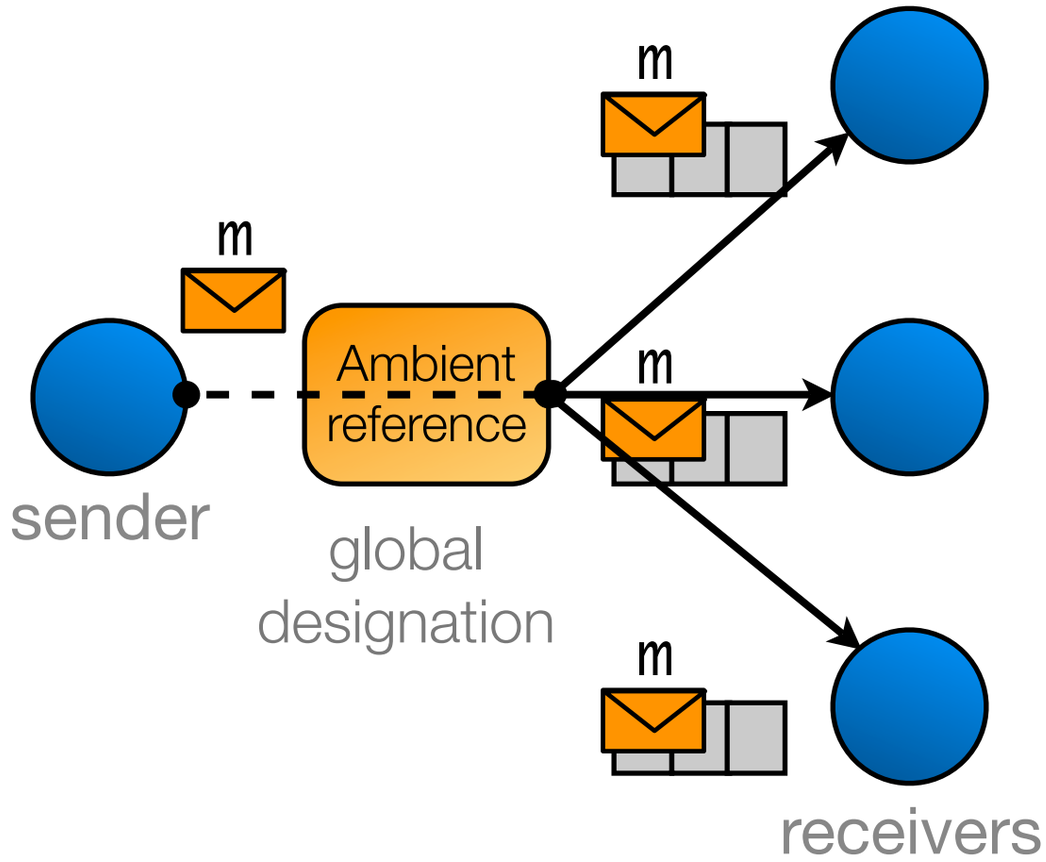



```
def m() {
   ...
}
```

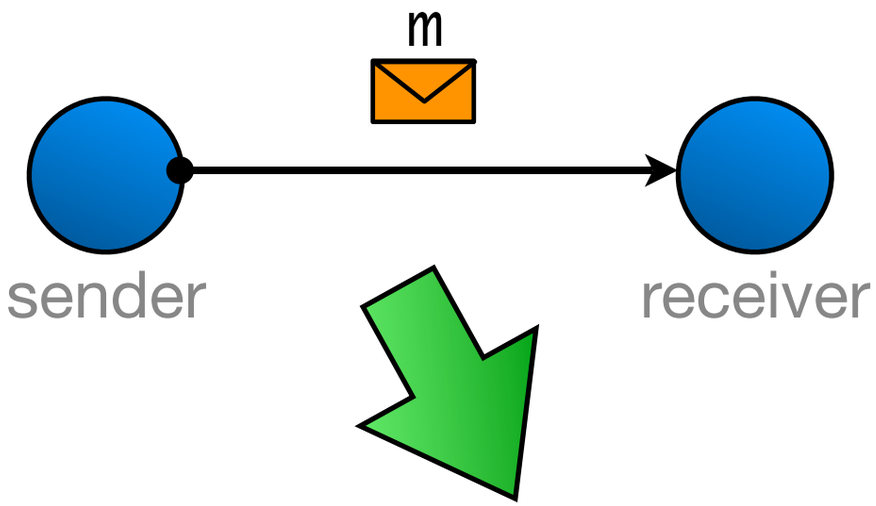
O/E Impedance Mismatch Revisited

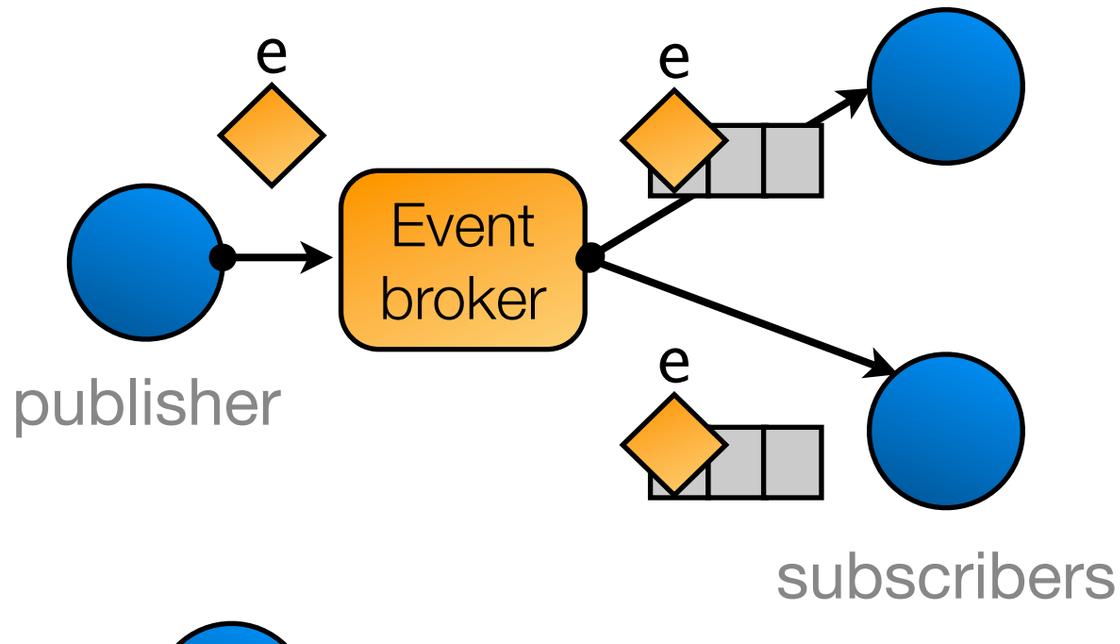


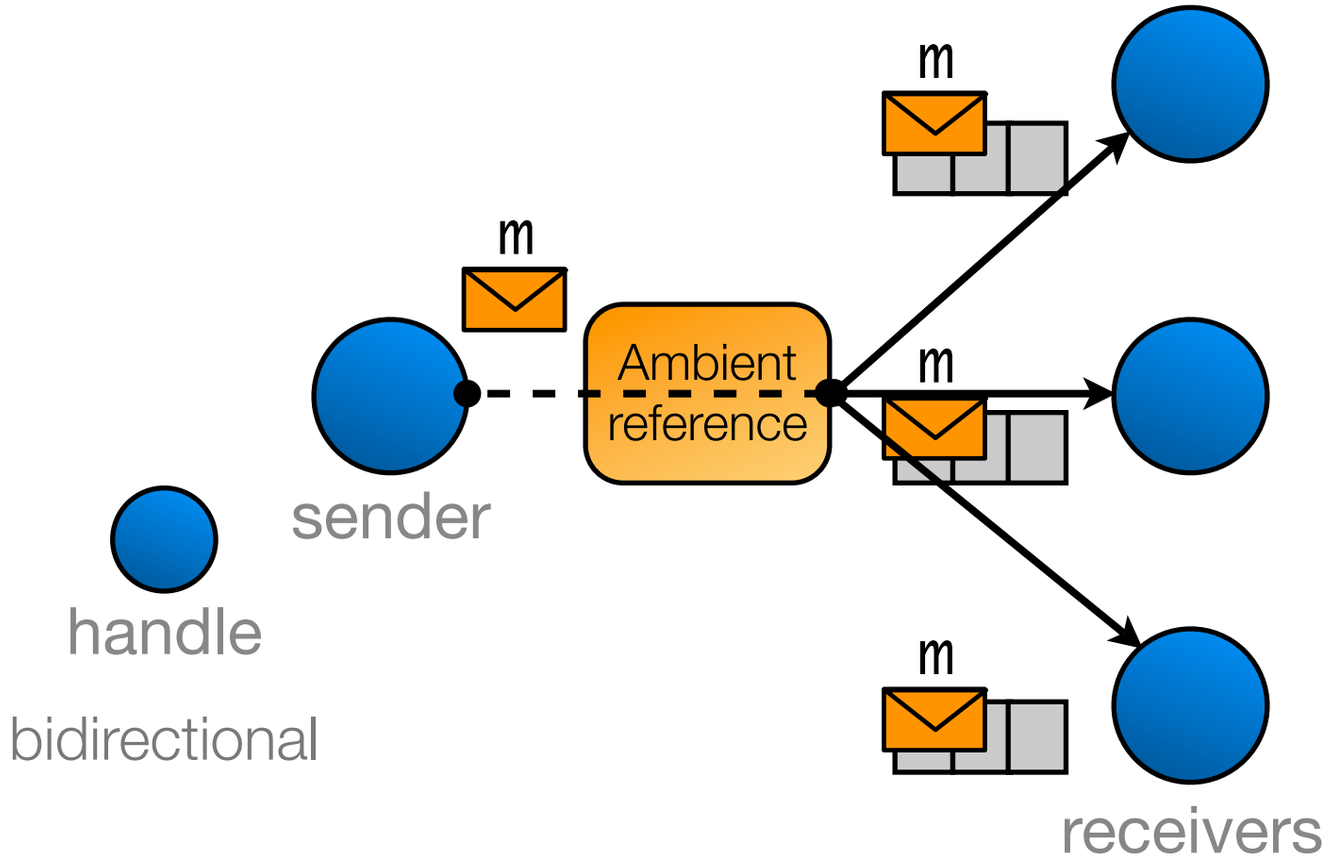




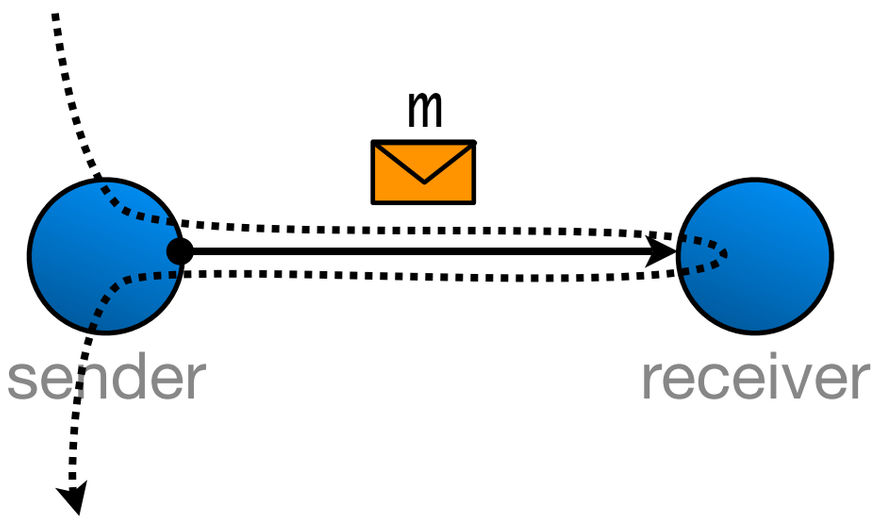
O/E Impedance Mismatch Revisited

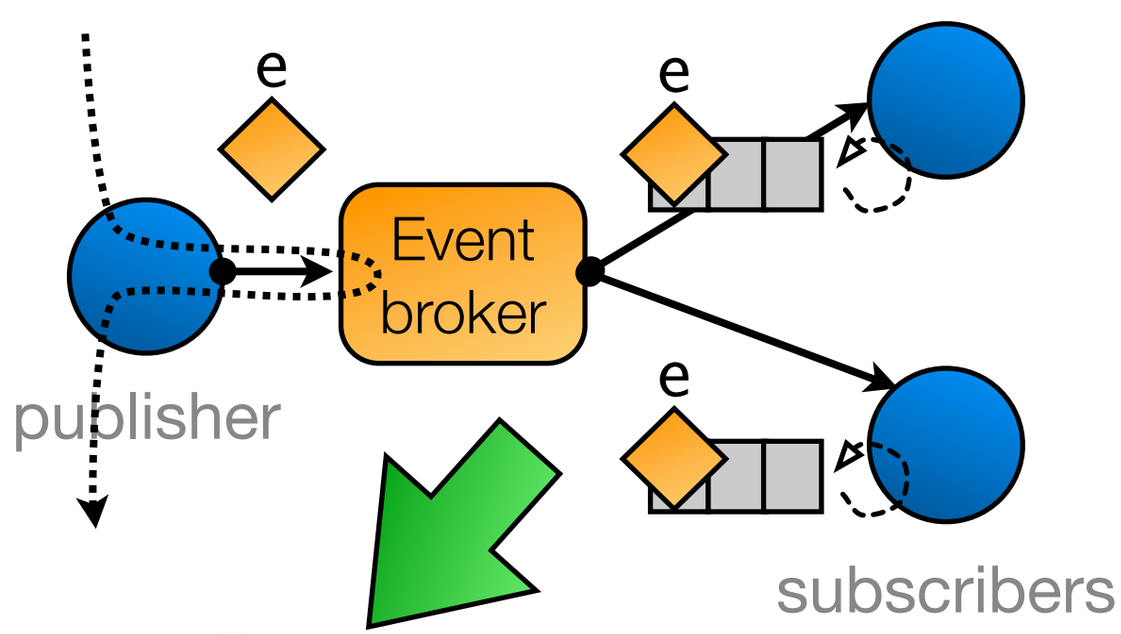


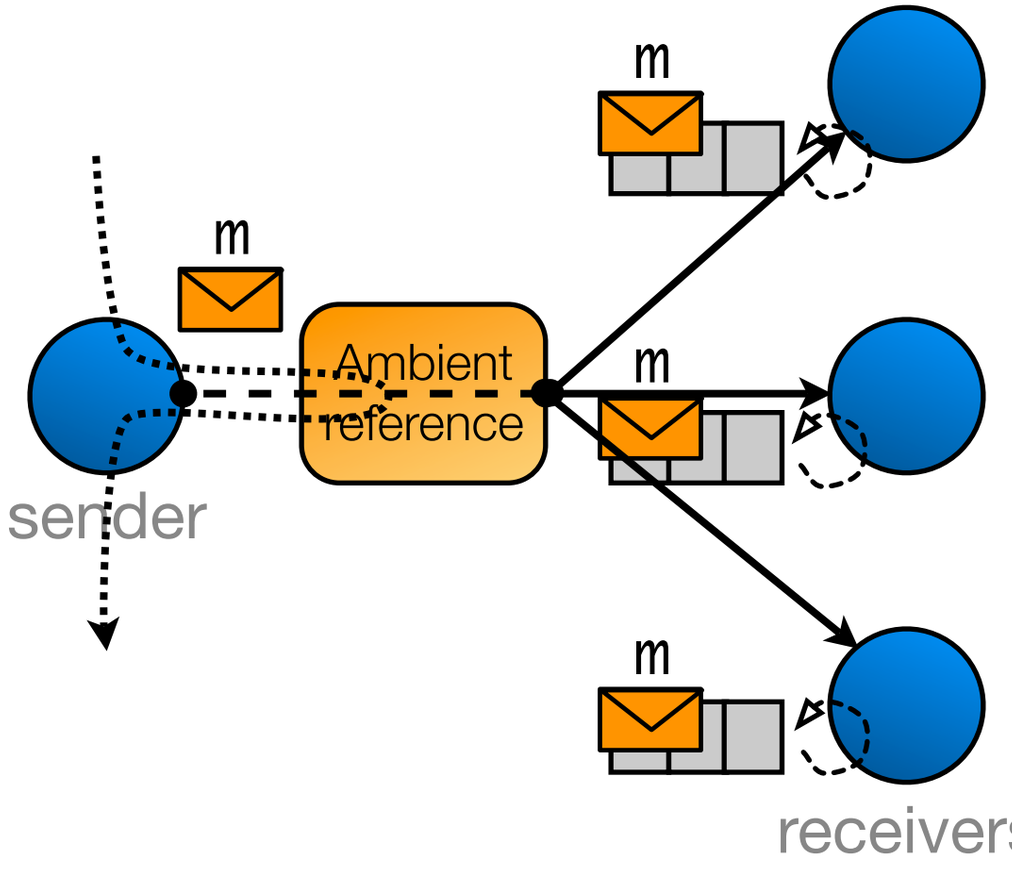




O/E Impedance Mismatch Revisited



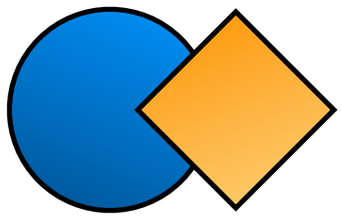




event loops

Roadmap

Motivation



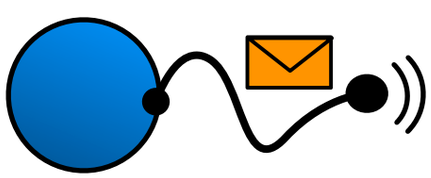
Object-event impedance mismatch

Coordination

Mobile ad hoc networks

Contribution

AMBIENTTALK



Ambient References

Validation

Evaluation

Implementation

Implementation & Validation

Motivation

Object-event impedance mismatch

Coordination

Mobile ad hoc networks

Contribution

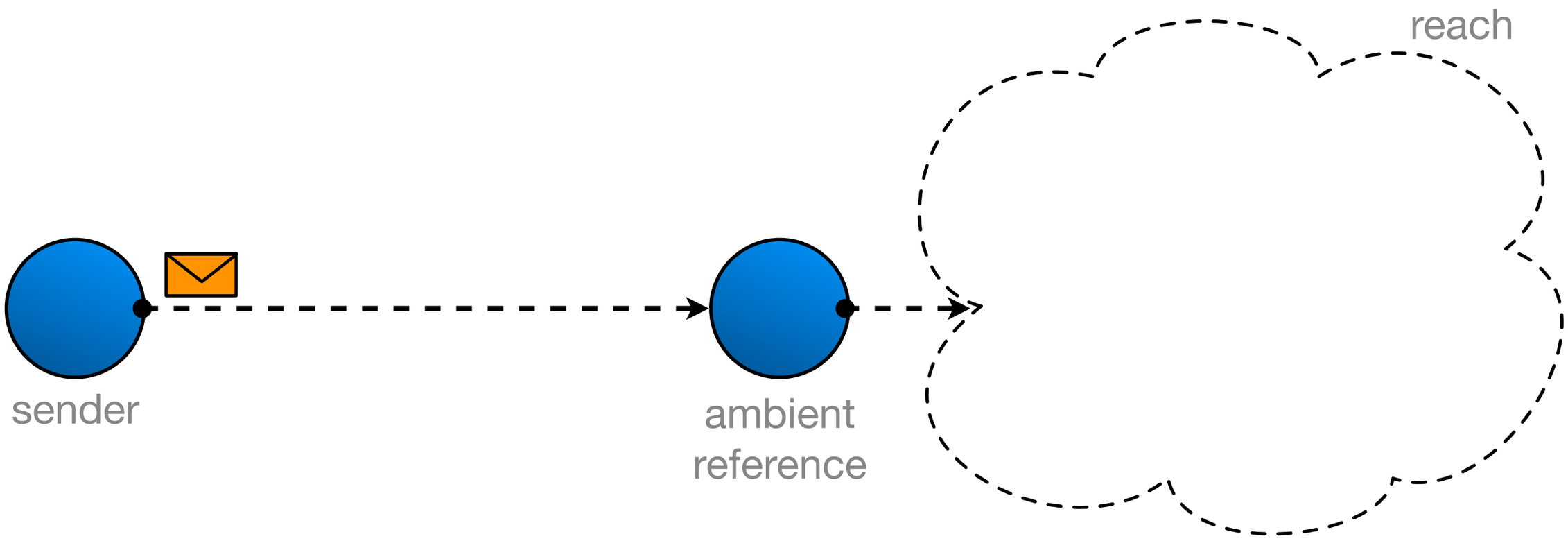
Ambient References

Validation

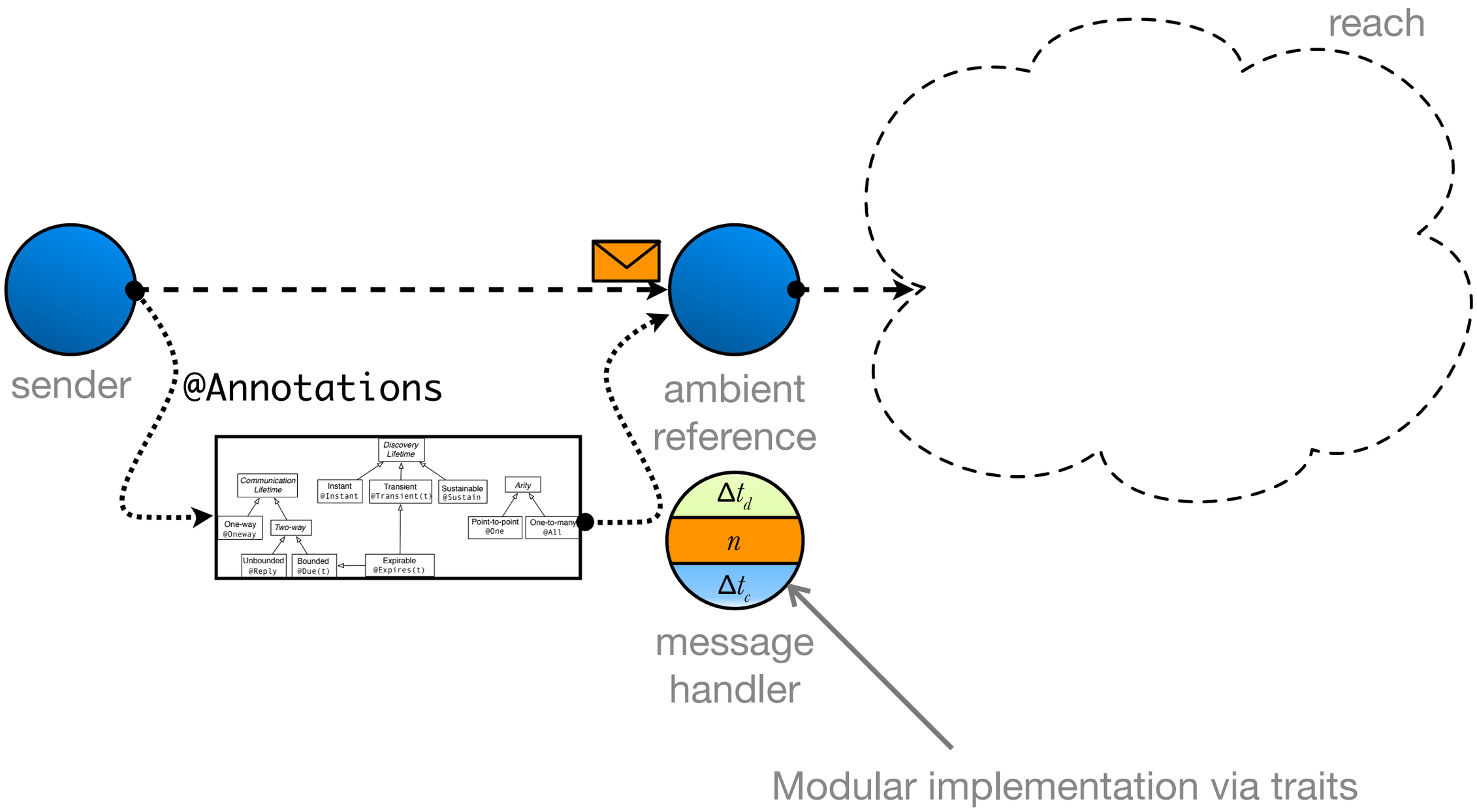
Evaluation

Implementation

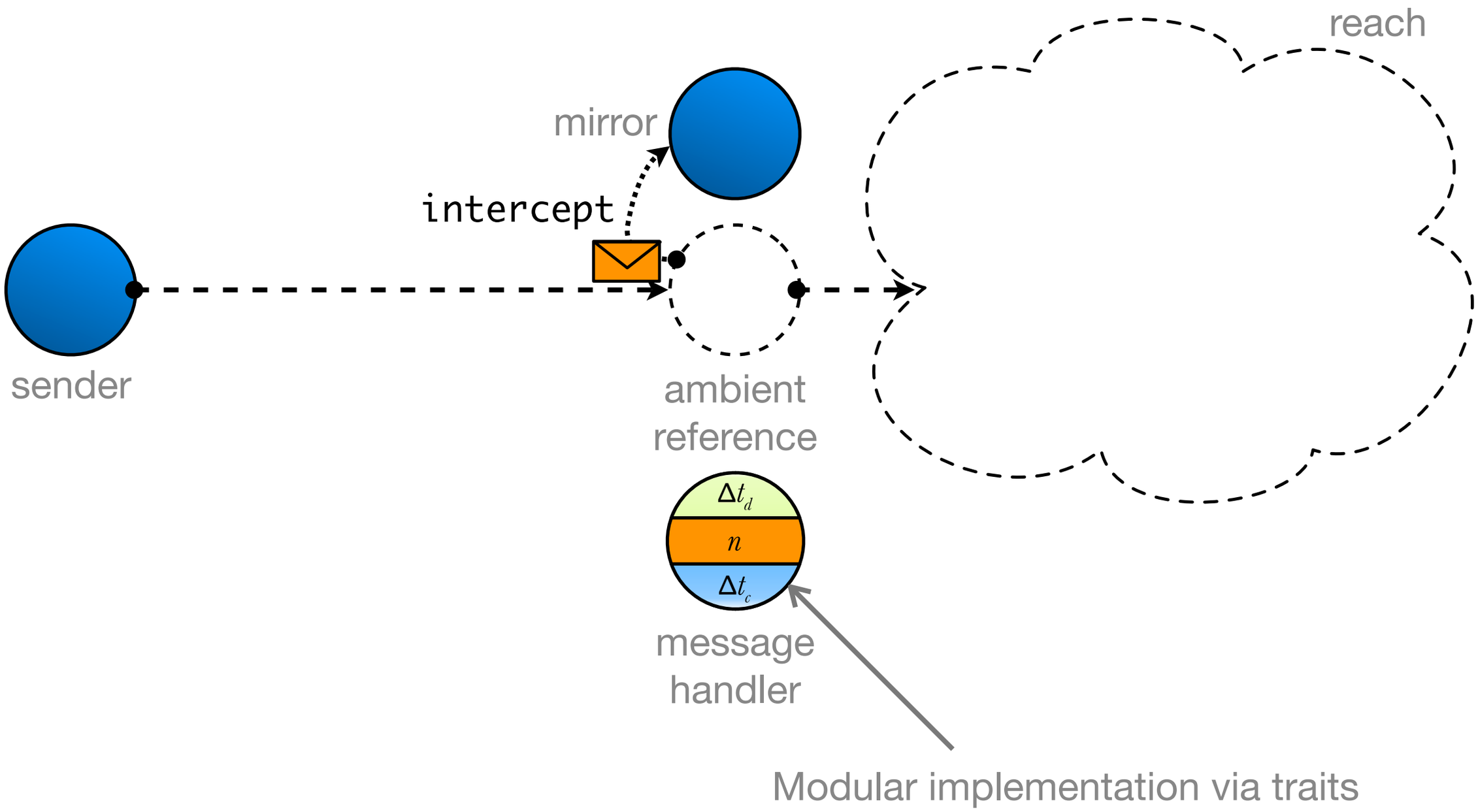
Implementation



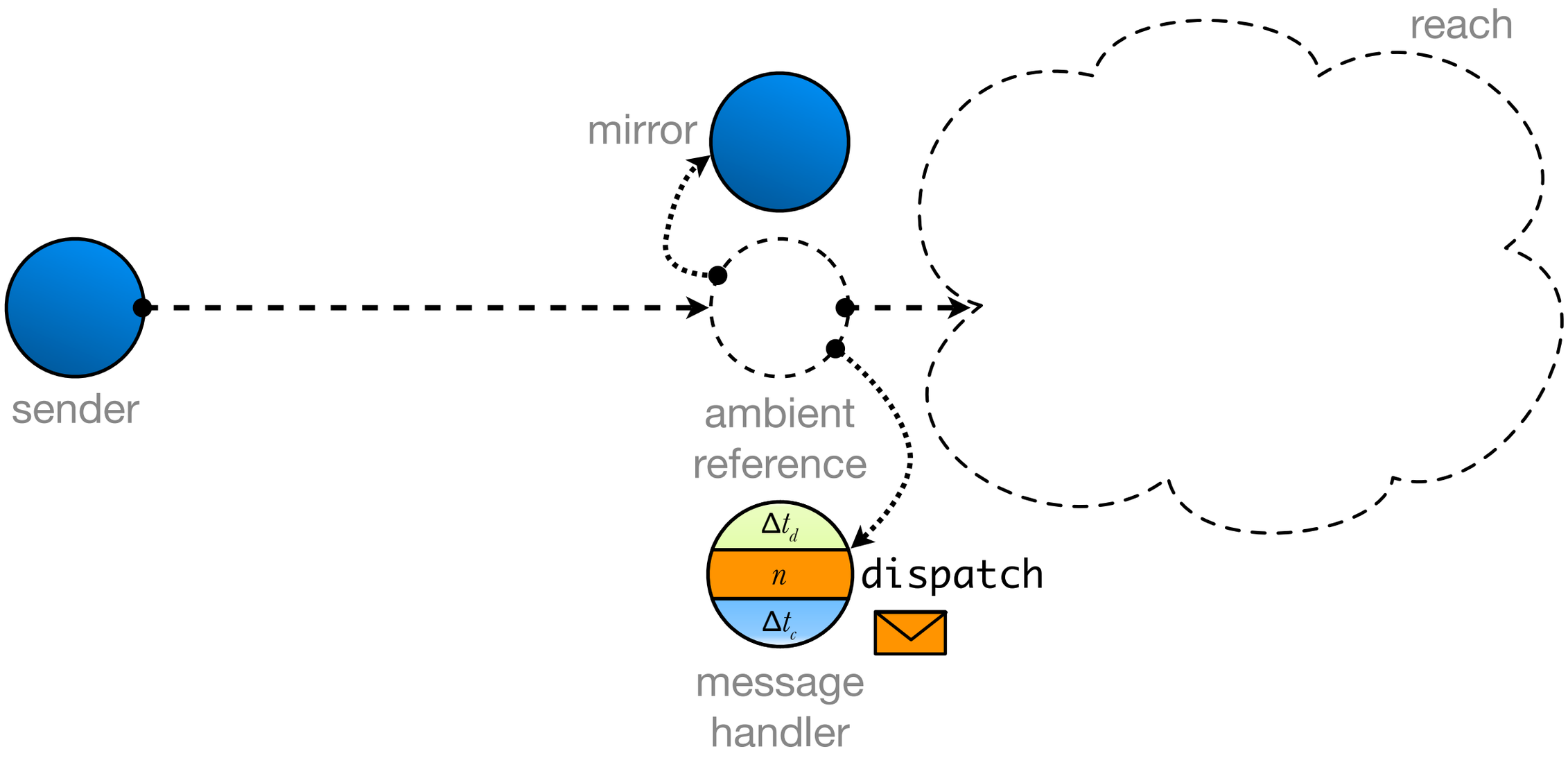
Implementation



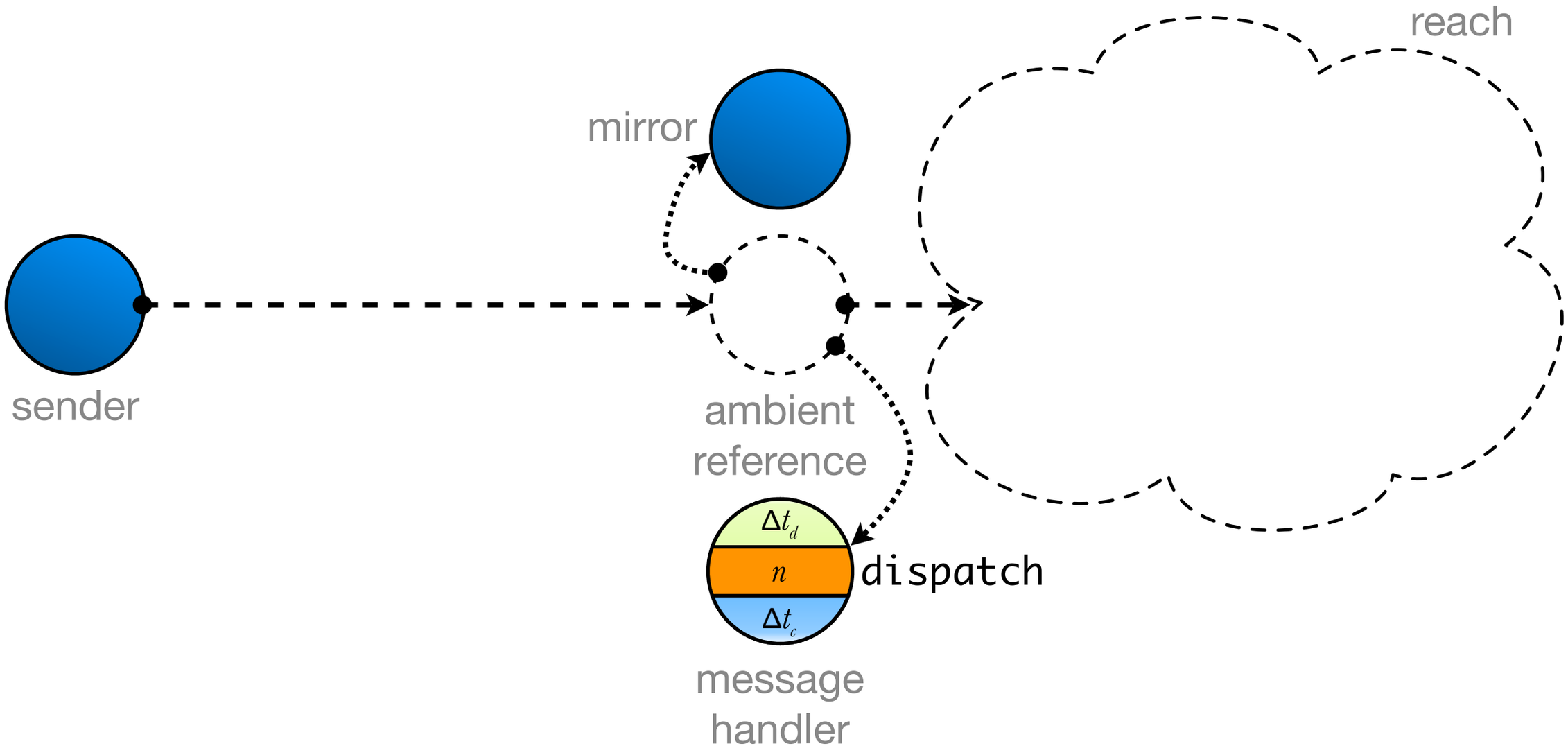
Implementation



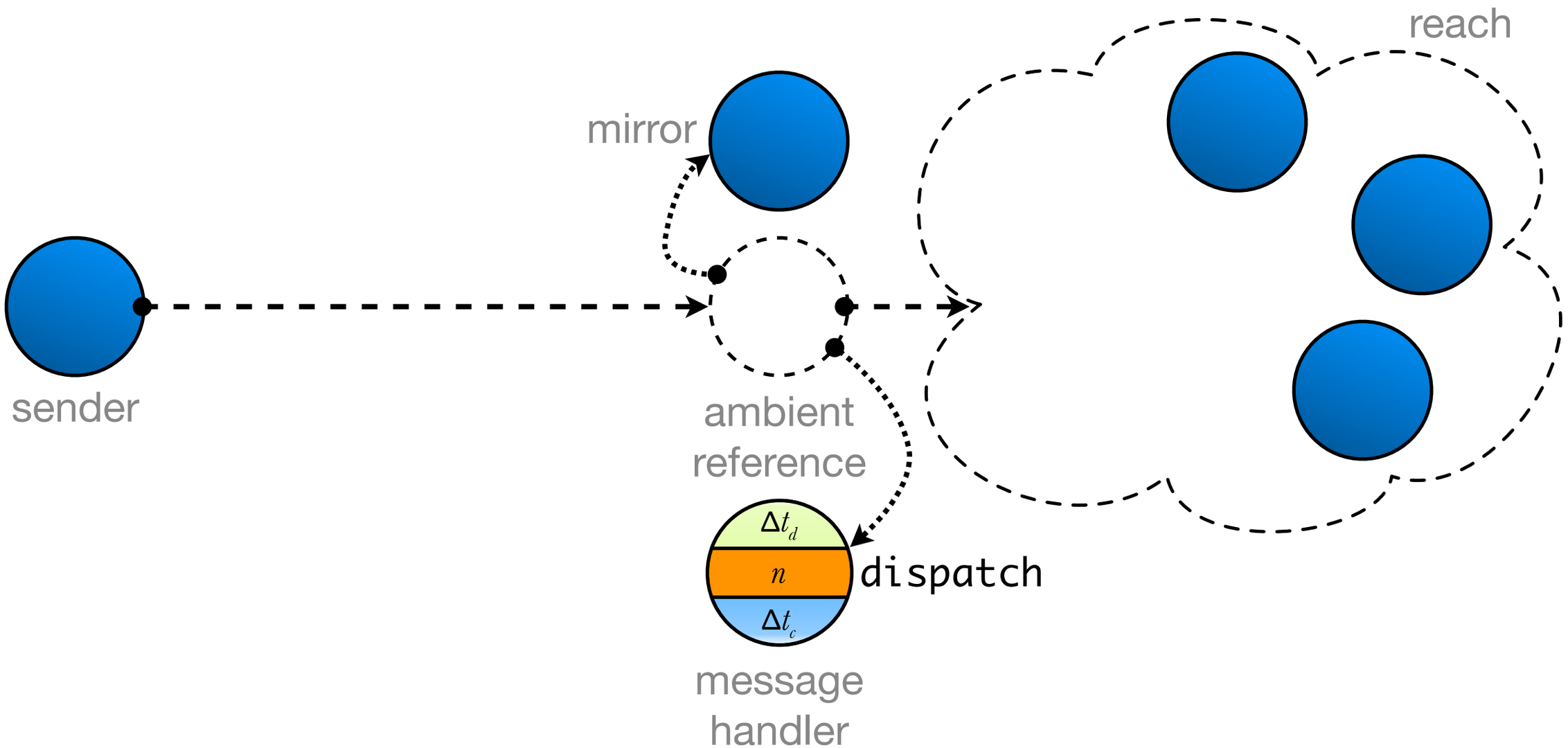
Implementation



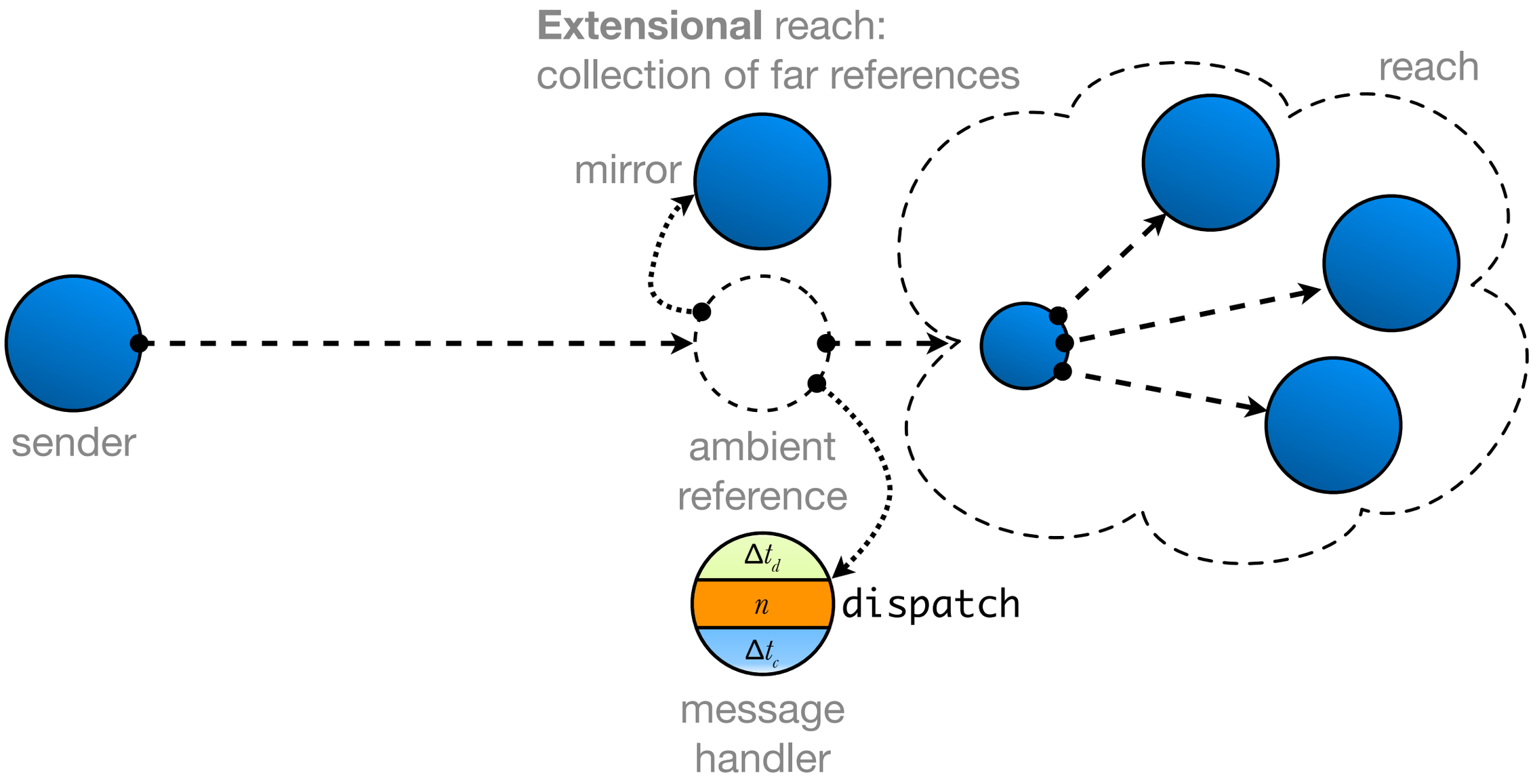
Implementation



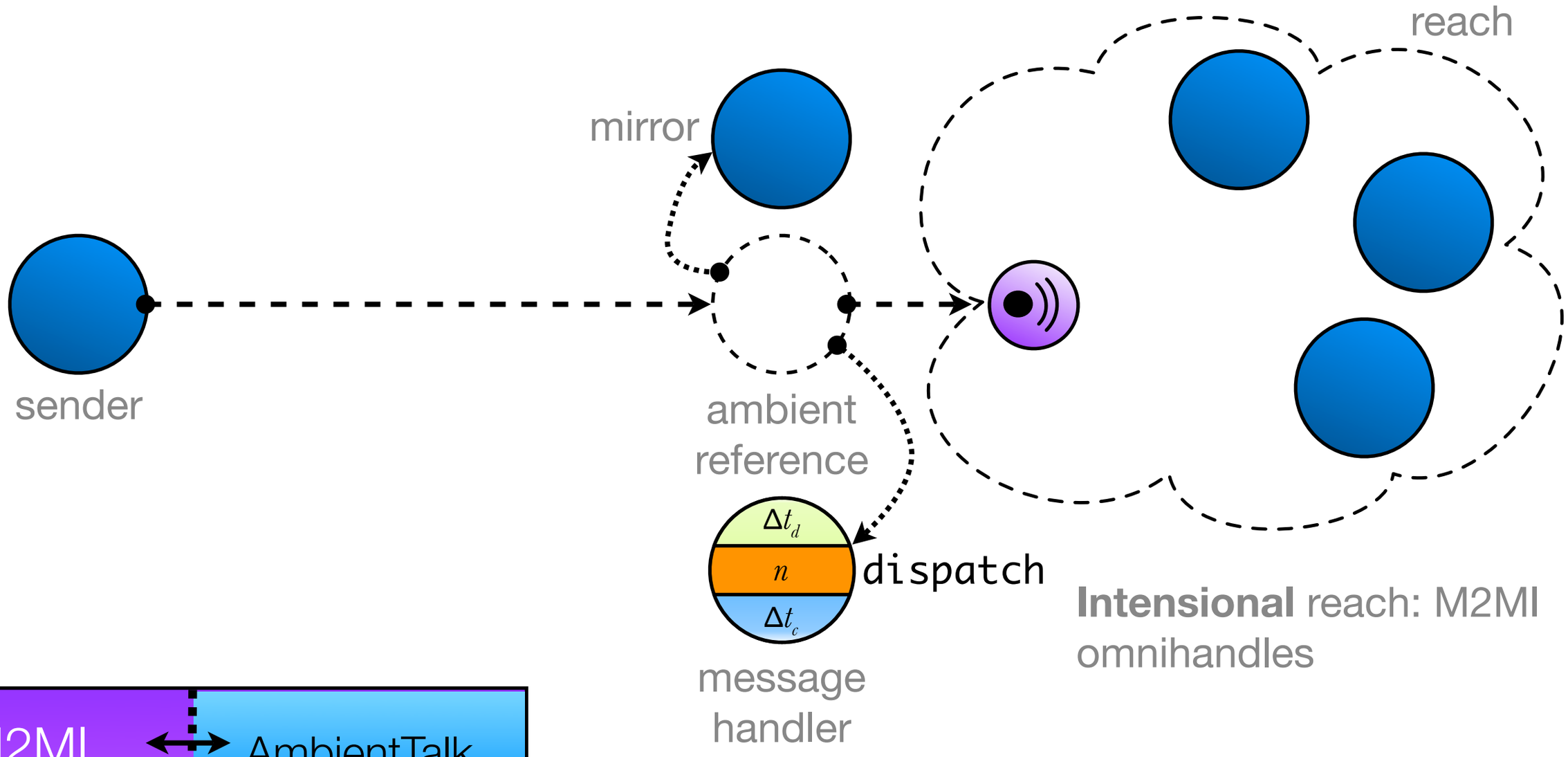
Implementation

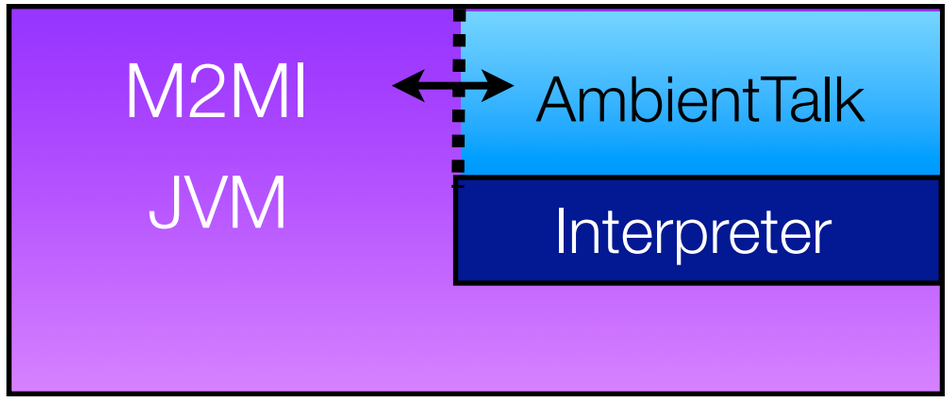


Implementation



Implementation





Linguistic Symbiosis

Validation

Ambient References

M2MI [Kaminsky & Bischof 02]

Validation

Ambient References

M2MI [Kaminsky & Bischof 02]

← Collaborative Chat

← Collaborative Slideshow

Validation

Ambient References

M2MI [Kaminsky & Bischof 02]

Collaborative Chat

Collaborative Slideshow

Location Tracker

Voting

Validation

Ambient References

M2MI [Kaminsky & Bischof 02]

Collaborative Chat

Collaborative Slideshow

Location Tracker

Voting

- Thread Synchronisation
- Explicit callbacks
- Explicit scheduling code
- No dynamic attributes

Concluding Remarks

Future Work

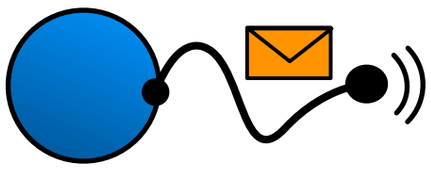
Service Discovery: ontologies, describing services, ...

Security: controlling **visibility** of exported objects & ambient references

More **efficient** implementation strategies

Conclusion

- Ambient-oriented Programming = “OOP in MANETs”
  - Object communication: decoupled in time & synchronisation [Dedecker 06]
  - Object **designation**: decoupled in space & arity
- Ambient references: **intensional** remote object references



- Unification of OO and event-based publish/subscribe interaction