

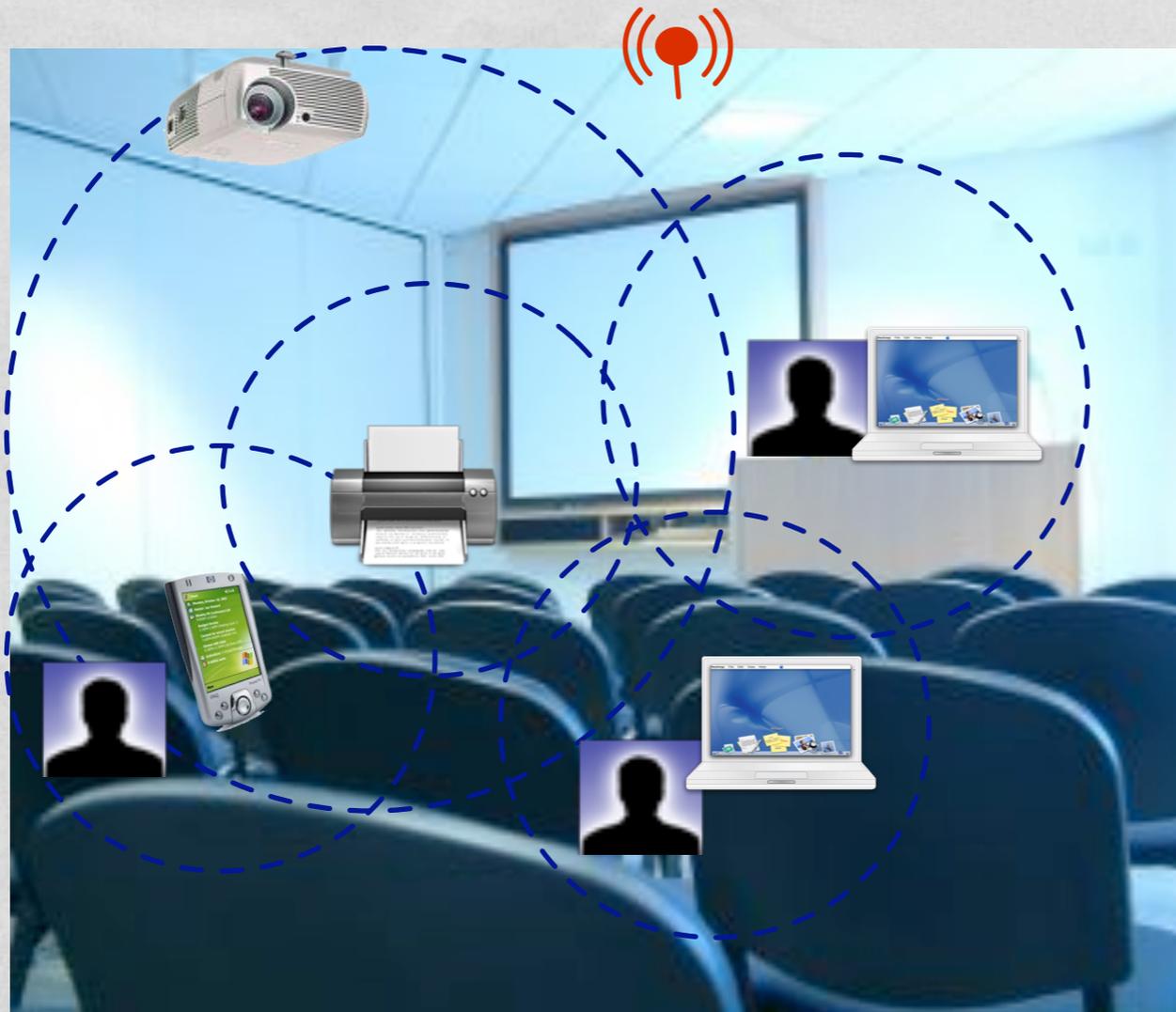
Language Support for Leasing in Mobile Ad hoc Networks

Elisa Gonzalez Boix



Vrije Universiteit Brussel

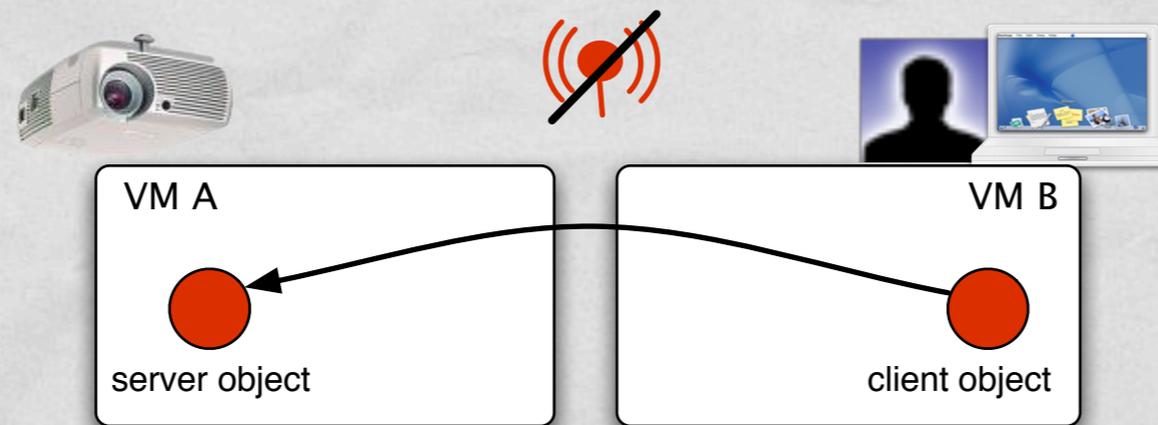
Mobile Networks



Volatile
Connections

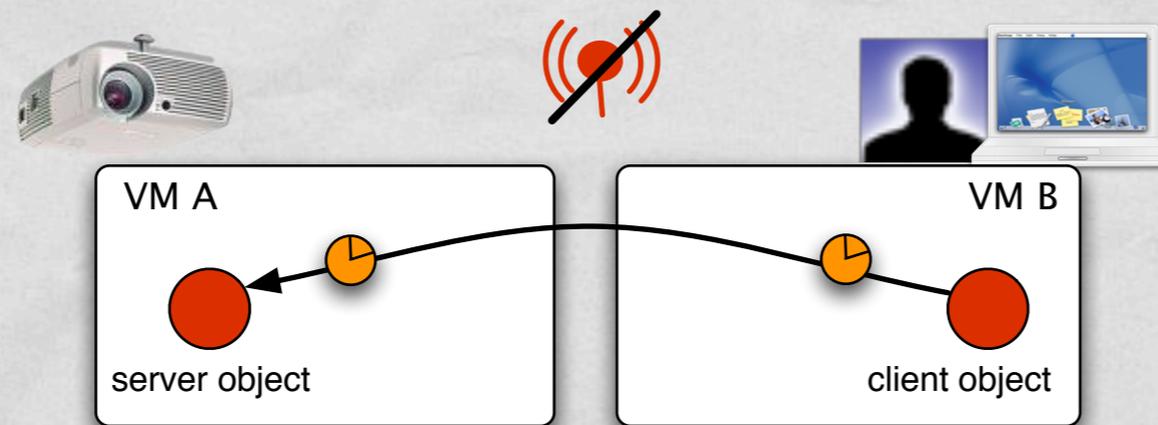
Scarce
infrastructure

Reclaiming Objects in MANETs



- What to do upon disconnection?
 - Time-decoupled remote references.
- But, transient and permanent disconnections!
 - Limiting lifetime.

Reclaiming Objects in MANETs



- What to do upon disconnection?
 - Time-decoupled remote references.
- But, transient and permanent disconnections!
 - Limiting lifetime.

Leased Object References

- Leases: proposed in distributed file cache consistency [gray89].
- .NET and JavaRMI: Leasing to reclaim unused connected references.
- Jini: Leasing to manage resources to tolerate network failures.

Leasing for first-class time-decoupled
remote refs

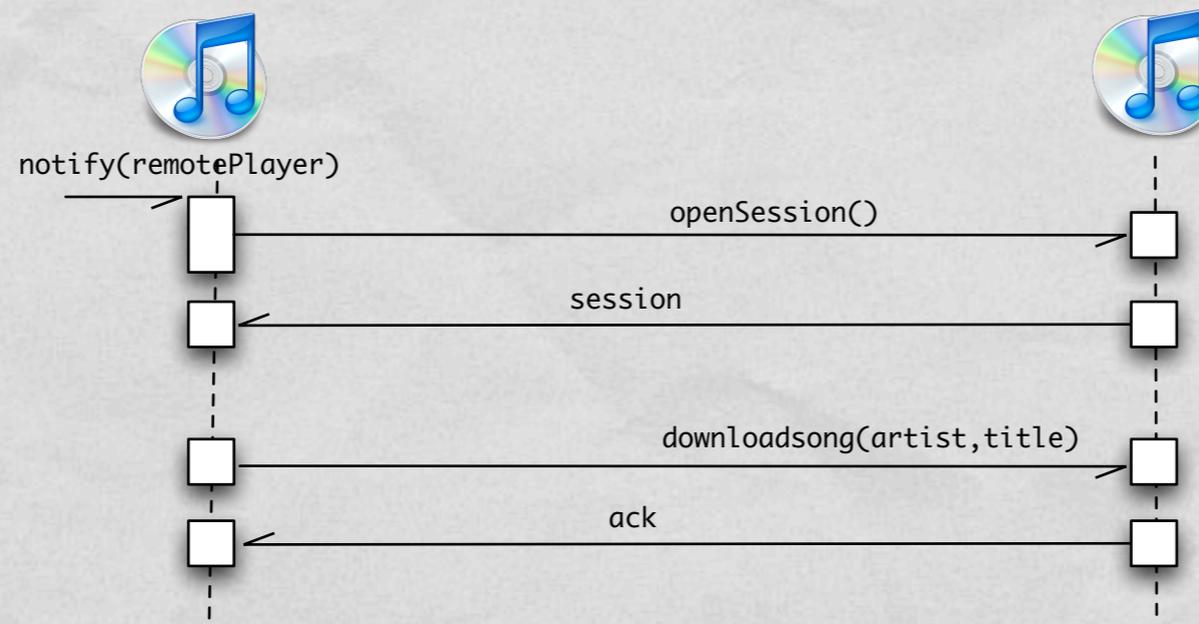
Language Support for Leasing

- Abstract low-level leasing management details.
- Expressiveness to determine lease periods.
- Leasing groups of objects.

What features to expose to the programmer
vs.
what to integrate in the language

Language Support in AmbientTalk

- Basic leased object references + 2 variants that adapt lease period.



Language Support in AmbientTalk

- Basic leased object references + 2 variants that adapt lease period.



```
def openSession() {
  def session := lease: minutes(10) for: (
    object: {
      def downloadSong(artist, title) {...}
      def endExchange() {...}
    }
  );
  when: session expired: {
    system.println("session with " + remoteUser + " timed out.");
    //cleanup code for session
  };
  session;
};
```

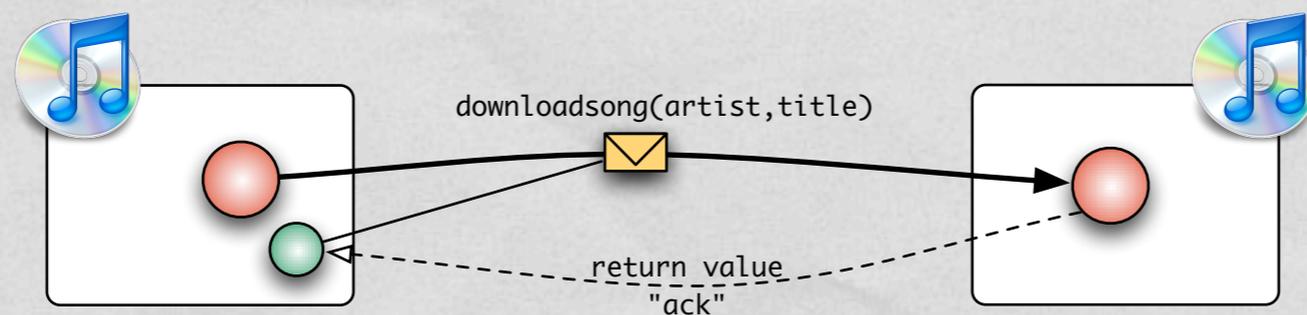
Language Support in AmbientTalk

- Renew-on-call leases: automatic prolongation of a lease upon method call.

```
def session := renewalOnCallLease: minutes(10) for: (  
  object: {  
    def downloadSong(artist, title) {...}  
    def endExchange() {  
      revoke: session;  
      ..  
      //calculate match percentage with my library  
    }  
  }
```

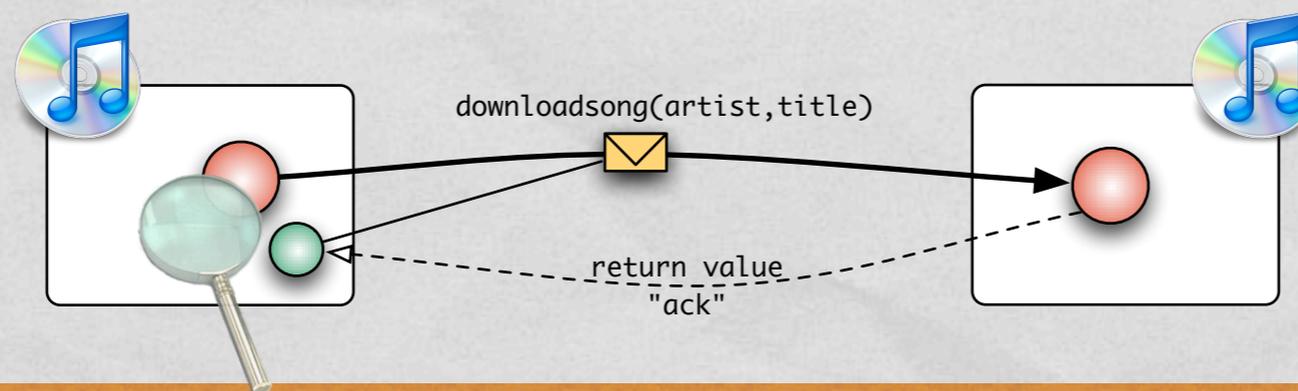
Language Support in AmbientTalk

- Single-call leases: automatic revocation of a lease upon method call.



Language Support in AmbientTalk

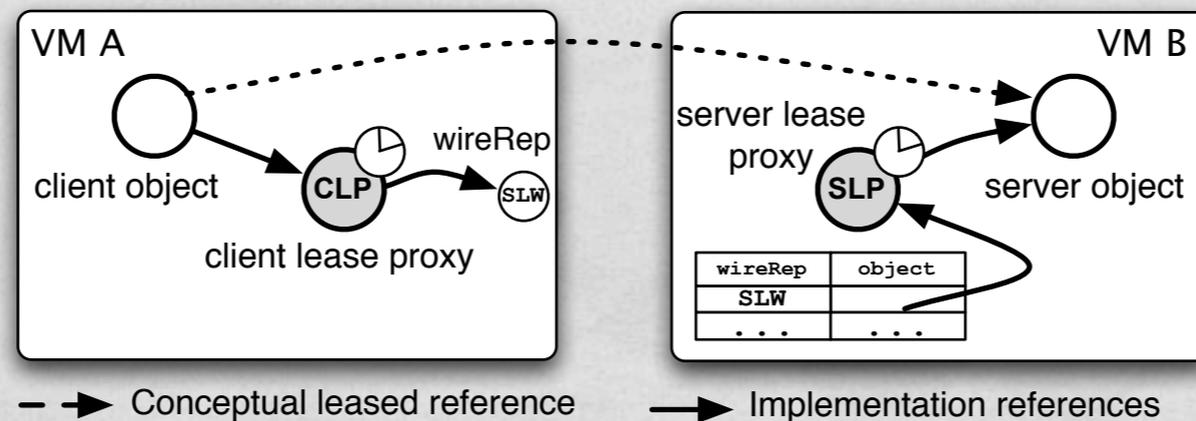
- Single-call leases: automatic revocation of a lease upon method call.



```
def songFut := session<-downloadsong(song.artist,song.title)@Due( leaseTimeLeft: session);  
  
when: songFut becomes: { |ack|  
  system.println("sent song " + song.artist + " - " + song.title);  
  sendSongs(); // recursive call to send the next song  
}  
catch: TimeoutException using: { |e|  
  system.println("stopping exchange: session timed out");  
}
```

Implementation

- Implemented Reflectively on AmbientTalk.



Determining proper lifetime..

- .. may depend on hardware events:

- Disconnection

```
lease: minutes(10) revokedOn: { networkDisconnected }
```

- Device resources

```
renewOnCallLease: minutes(10) renewalConditions: { if: (batteryLevel > 10%) }
```

- .. may depend on state of object graph:

```
session<-downloadSong(song.artist,song.title)@Due( leaseTimeLeft: session);
```

Strategies based on context info

- Built-in renewal of leases based on context parameters.
- Context info permeates to behaviour of leases!



Ongoing Directions

- Integration of leased references with other language constructs.
- New strategies to automatically adapt lease periods.
 - Strategies based on context information.
- Delimiting a scope of action for leasing.