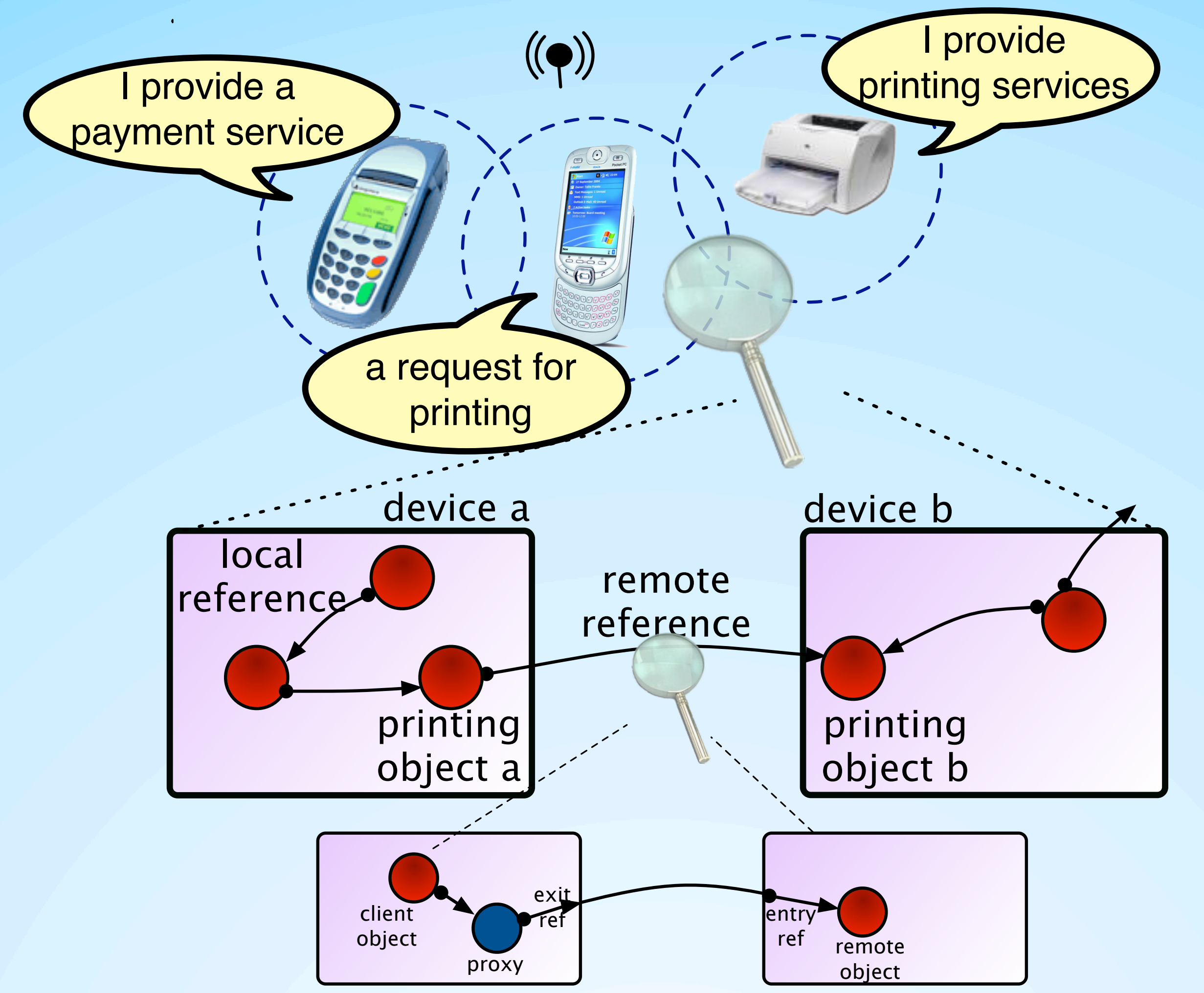
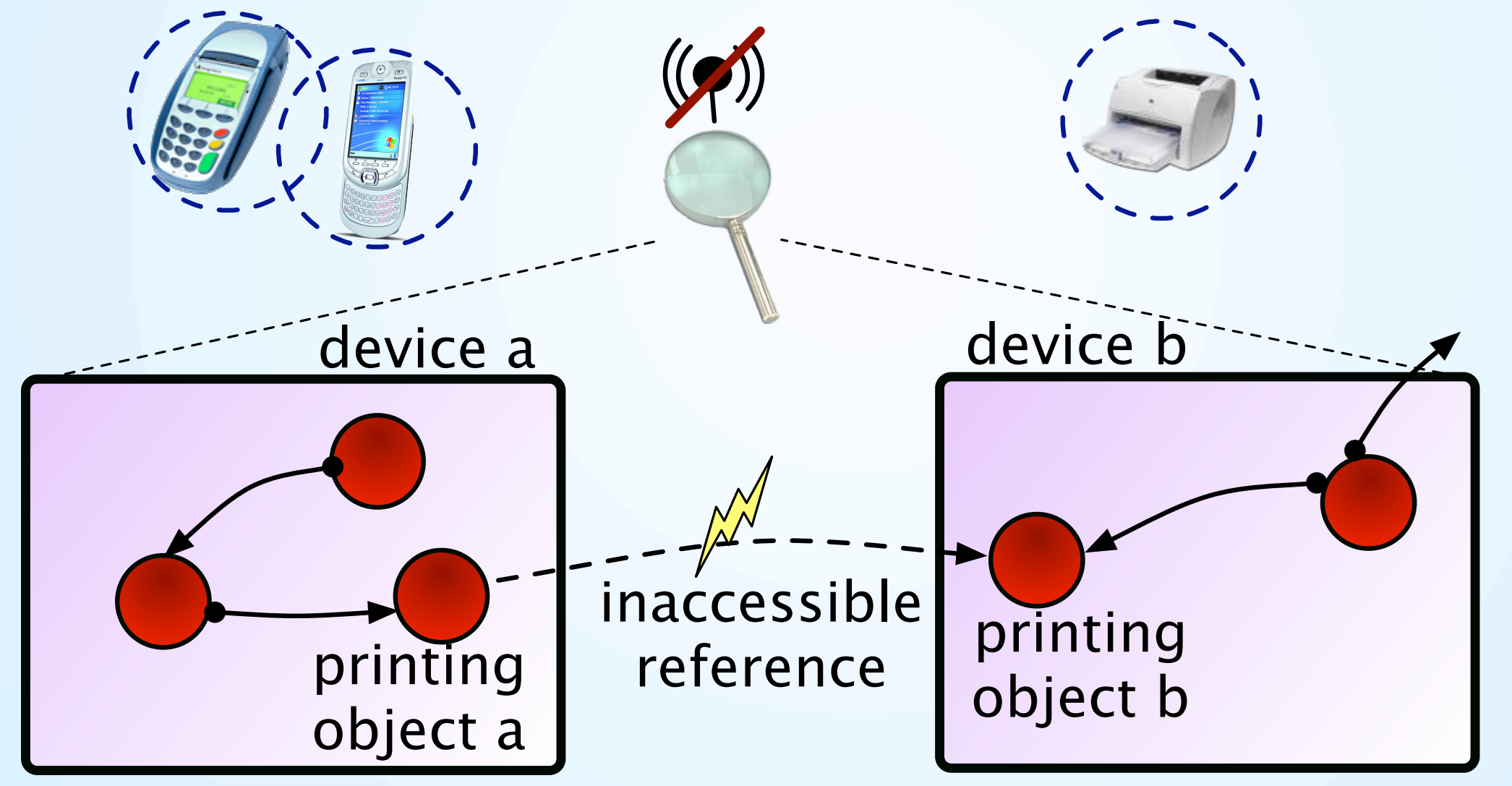
Semi-Automatic Garbage Collection for Mobile Networks

Elisa Gonzalez Boix   Tom Van Cutsem   Stijn Mostinckx
Jessie Dedecker   Wolfgang De Meuter   Theo D'Hondt
Programming Technology Laboratory - Vrije Universiteit Brussel - Belgium

Motivation: DGC in Mobile Networks

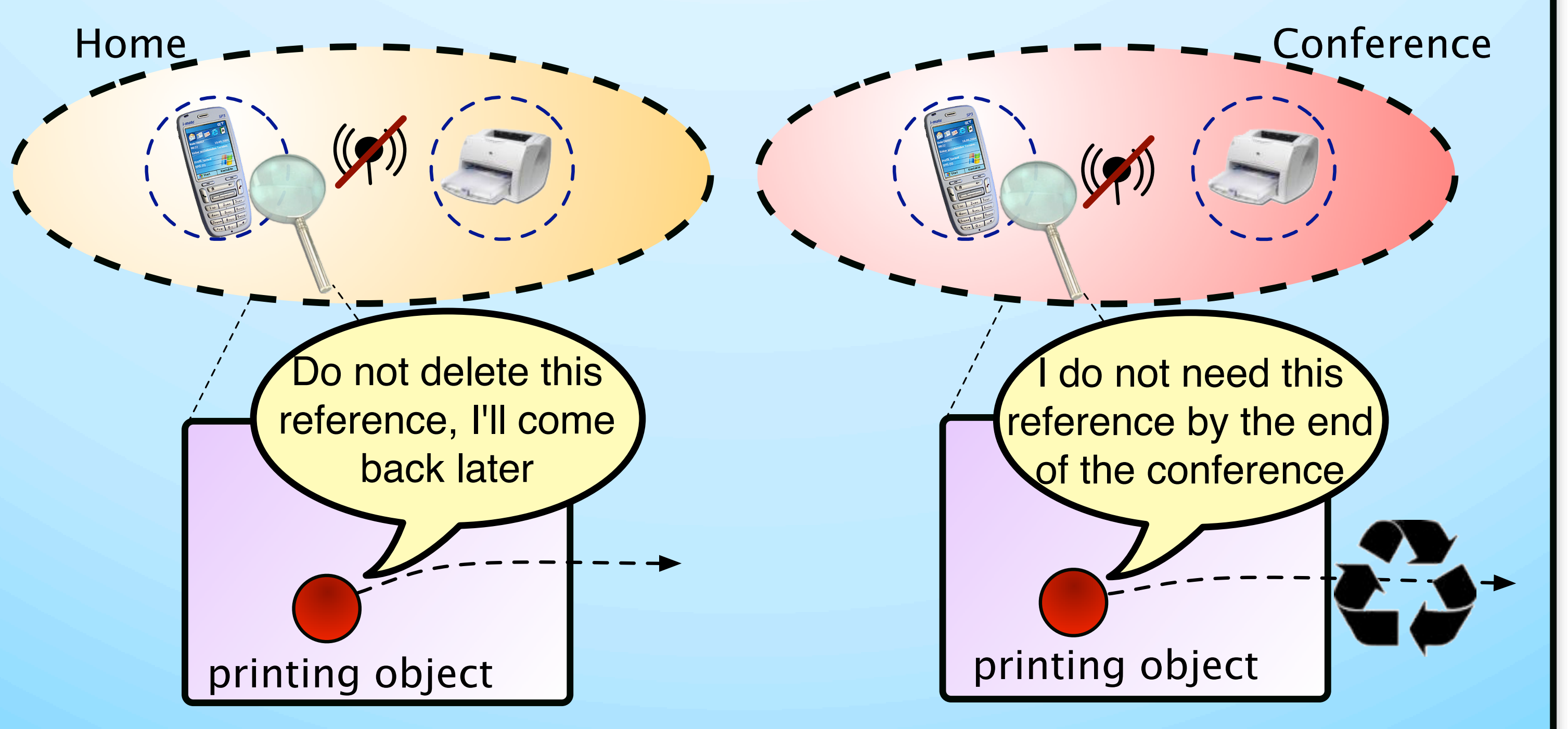


Volatile connections



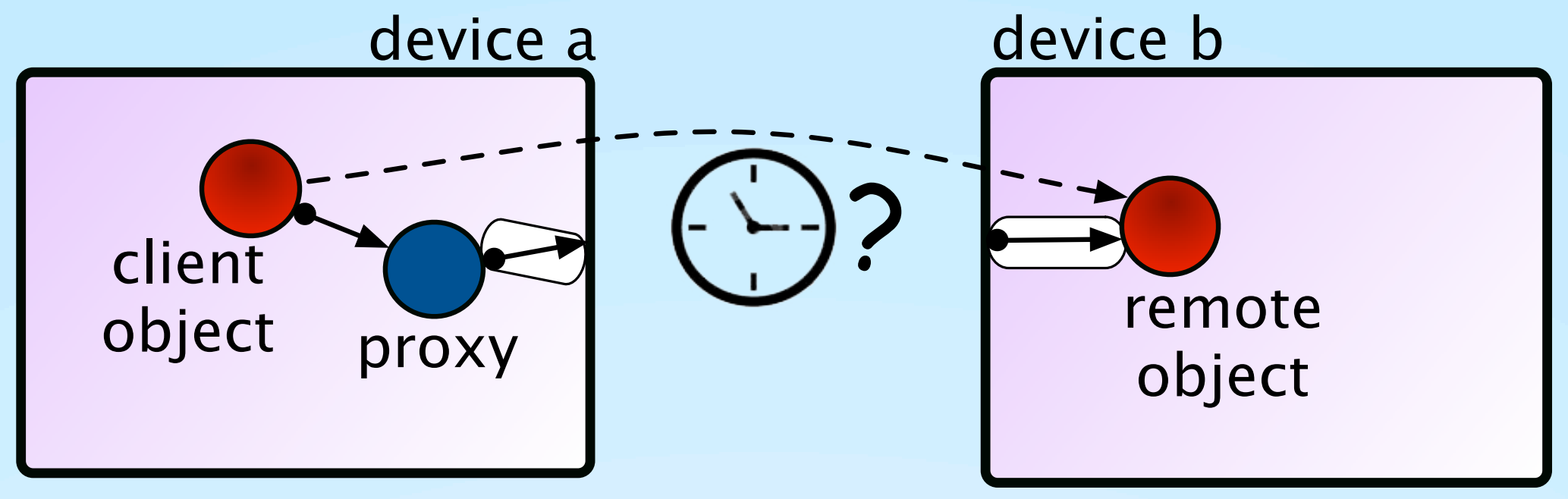
- Limited connectivity of the nodes.
- Inaccessible reference ≠ Broken reference.

Context-dependent information


Home
Conference
Do not delete this reference, I'll come back later
I do not need this reference by the end of the conference
printing object
printing object

- Semantics of the application are required to clear objects.

Problem Statement



How long should the system wait for the connectivity of the devices?

- Application-dependent!
- Even context-dependent!

Who is responsible for garbage collection?

Developer ? ▶ ~~Manual approach~~

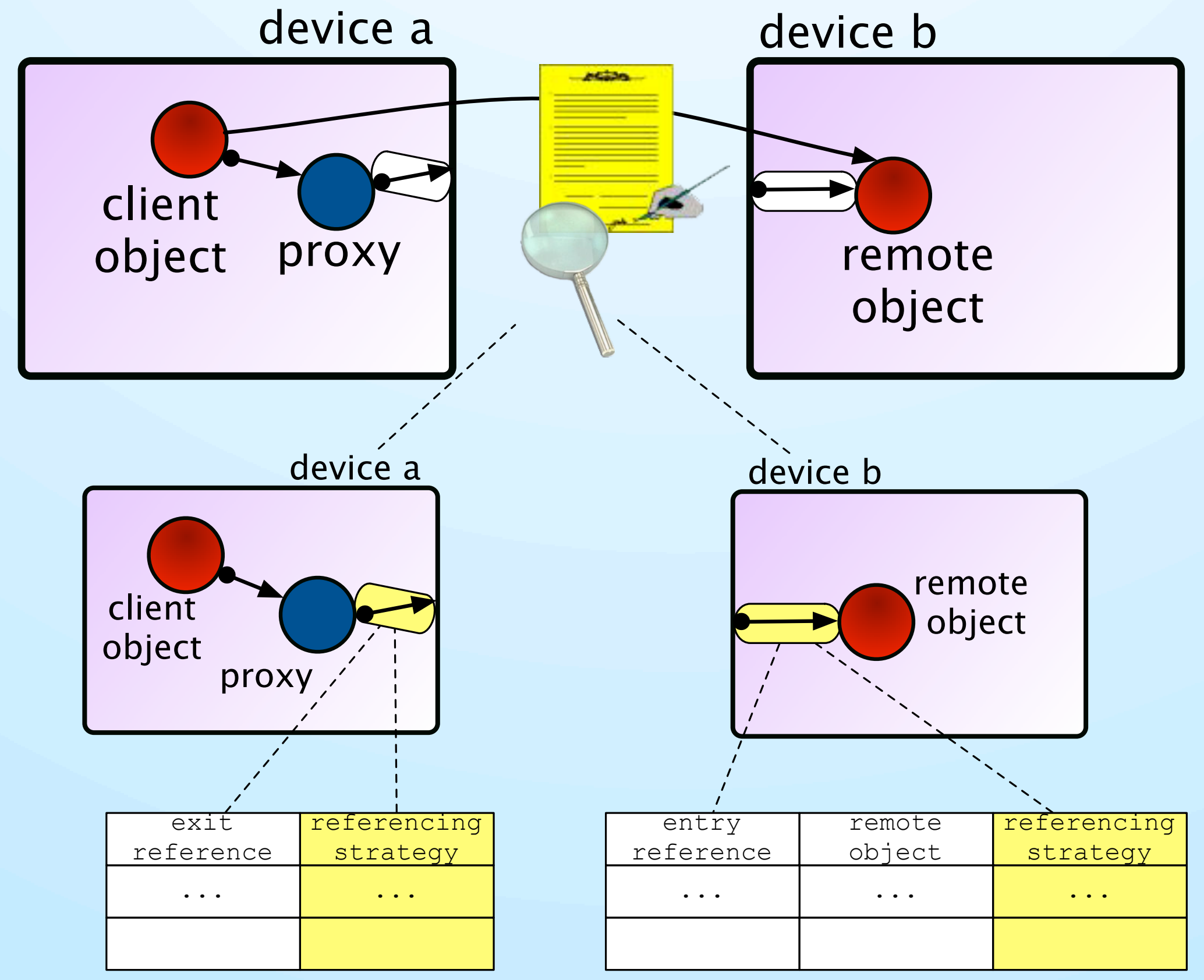
System ? ▶ ~~Automatic Transparent Garbage Collection~~

▼

A combination of both:

Semi-Automatic Garbage Collection

- Remote-references as a two-party contract.



- Referencing strategies express collection policies (remote object) + kind of rebinding (client object) upon disconnection.

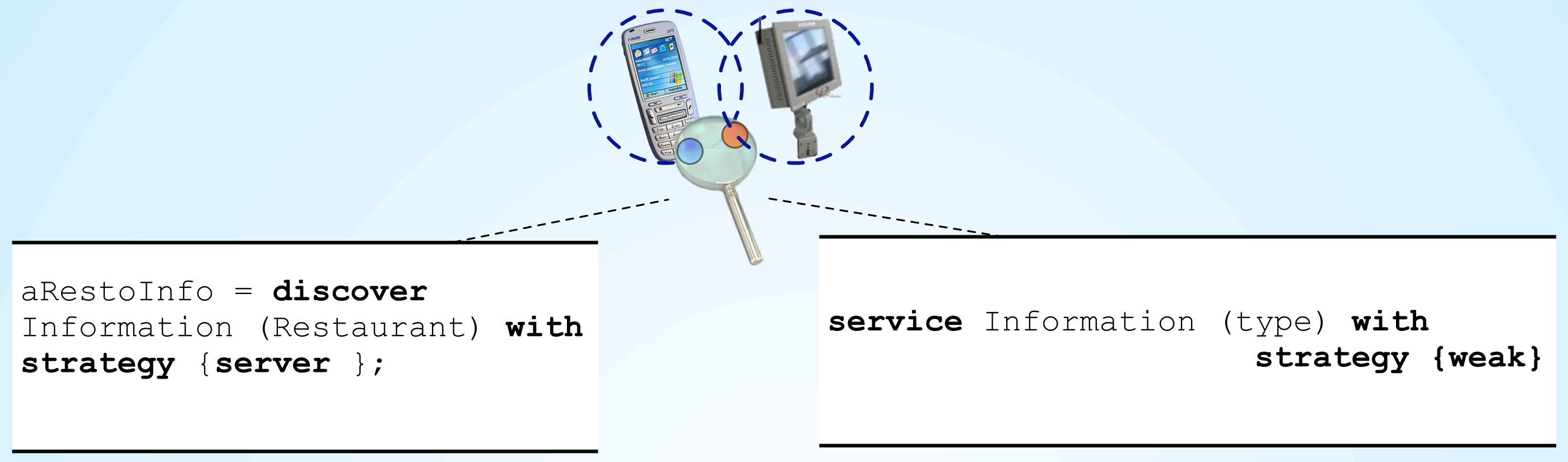
Referencing Strategies

- Language support to apply a collection strategy to the remote reference.
- Express the disposability of a reference upon disconnection:

Temporal disposability

... expressing disposability based on time constraints.

- Weak References: always reclaim.
- Temporal References: reclaim after time period.
- Strong References: never reclaim.

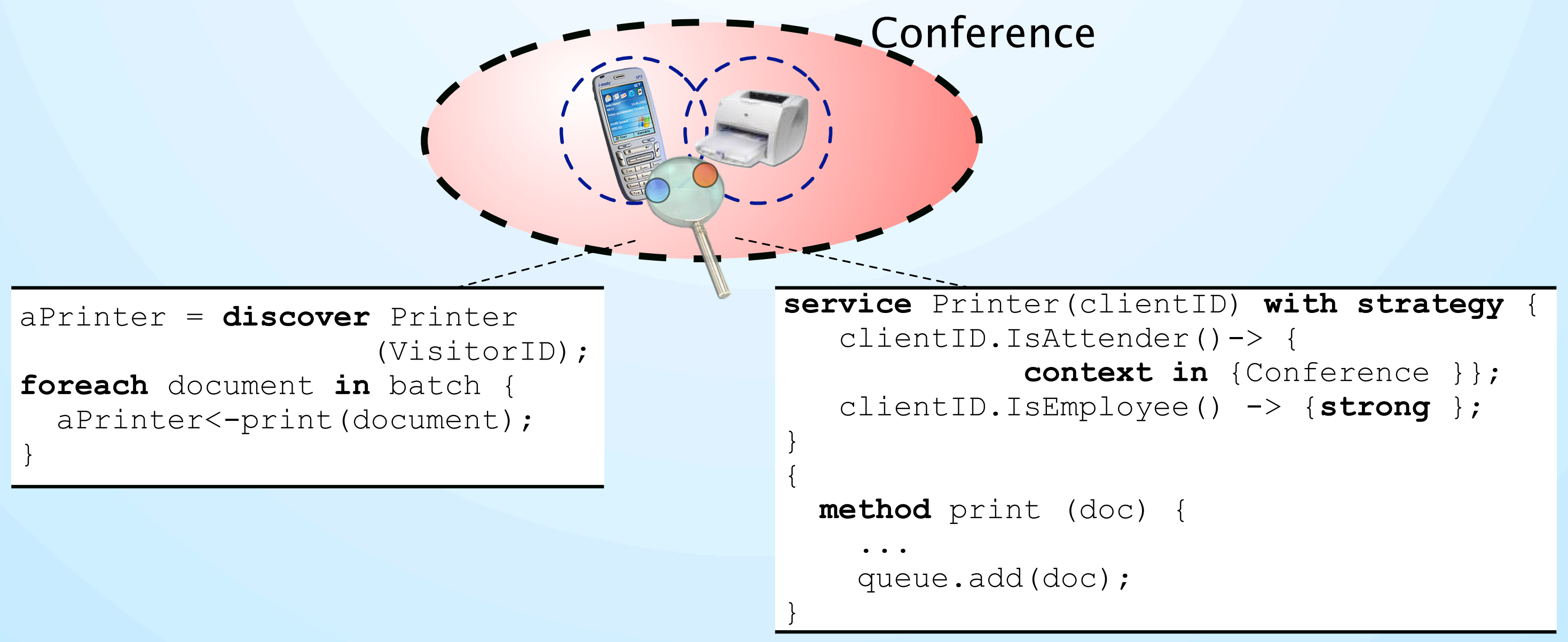


```
aRestoInfo = discover
Information (Restaurant) with
strategy {server };
```

```
service Information (type) with
                  strategy {weak}
```

Domain-specific disposability

... expressing disposability based on context information.



```
aPrinter = discover Printer
                     (VisitorID);
foreach document in batch {
  aPrinter<-print(document);
}
```

```
service Printer(clientID) with strategy {
   clientID.IsAttender()-> {
                context in {Conference }};
   clientID.IsEmployee() -> {strong };
}
{
  method print (doc) {
     ...
     queue.add(doc);
}
```

... some challenges

- Resolving conflicts.

	weak	temporal	strong
weak	✔	✔	✔
temporal	✘	✔ (t1<t2)	✔
strong	✘	✘	✔

- Indirect References.
- Language support to annotate groups of remote references.

Read more?

Gonzalez Boix E., Van Cutsem T., Mostinckx S., Dedecker J., De Meuter W., D'Hondt T. *Semi-Automatic Garbage Collection for Mobile Networks.* In workshop on Object Technology for Ambient Intelligence and Pervasive Computing (OT4AmI) co-located at the 20th European Conference on Object-Oriented Programming (ECOOP), Nantes, France. 2006.

http://prog.vub.ac.be/amop