

Garbage Dump

Esame di
Sistemi Multi Agente LS
prof. Andrea Omicini

Maria Teresa Roccia

Scenario

Obiettivo: pulire la discarica dai rifiuti, raccogliendo le diverse tipologie di rifiuti presenti in appropriati container per il riciclaggio.

Chi: cleaner agent addetti alla raccolta di plastica, carta o metallo, capaci di individuare i rifiuti mediante un apposito rilevatore.

Come: ciascuno ha il compito di raccogliere un certo numero massimo di rifiuti della stessa tipologia alla volta, raggiunto il quale dovranno essere scaricati nell'apposito container, quindi la raccolta potrà proseguire.

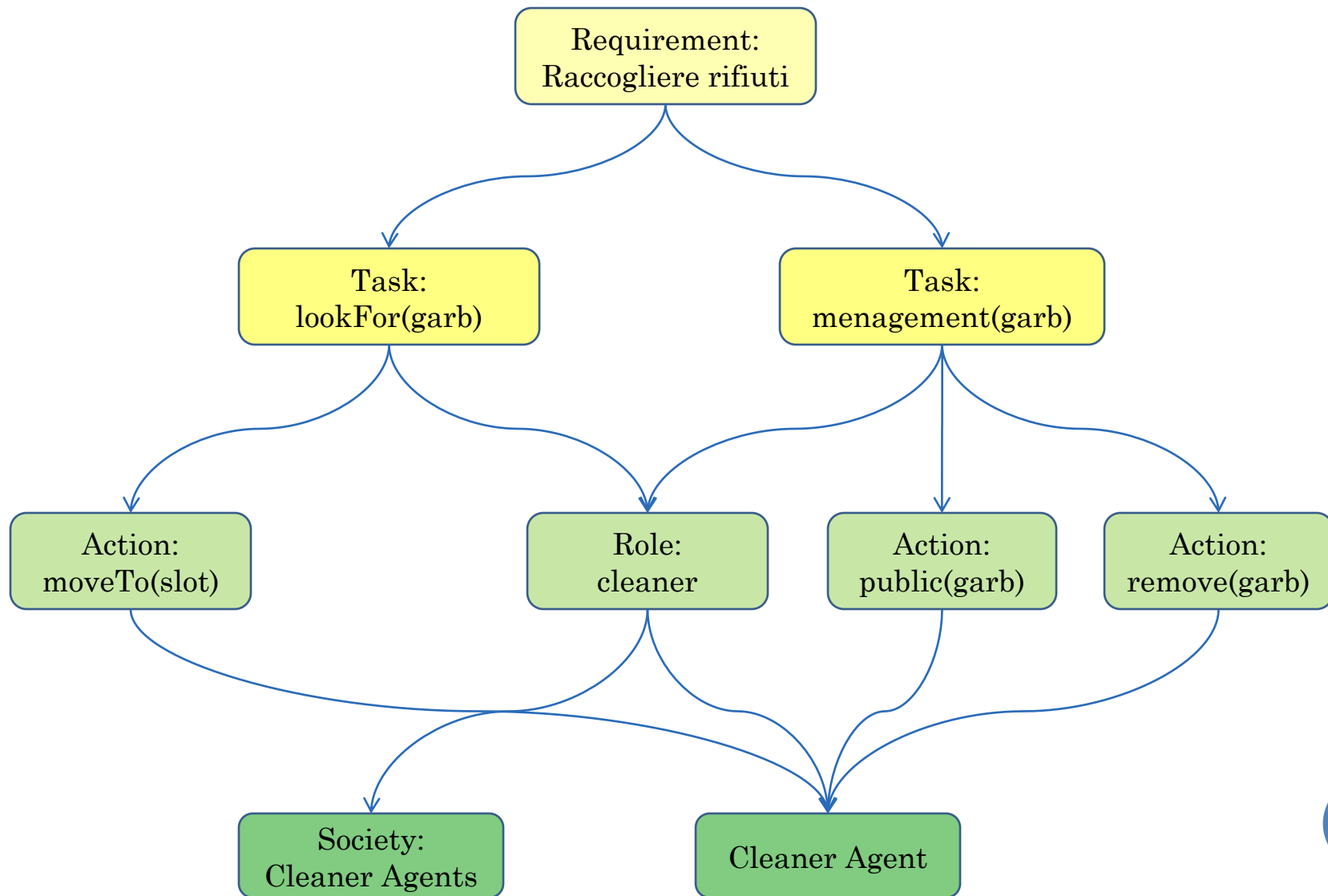
Quando viene individuato un rifiuto:

- Se è di interesse e l'agente non è pieno, viene rimosso dalla discarica.
- Se è di interesse, ma l'agente è pieno, oppure non è di interesse, viene pubblicato per la raccolta da parte dello stesso agente o di altri.

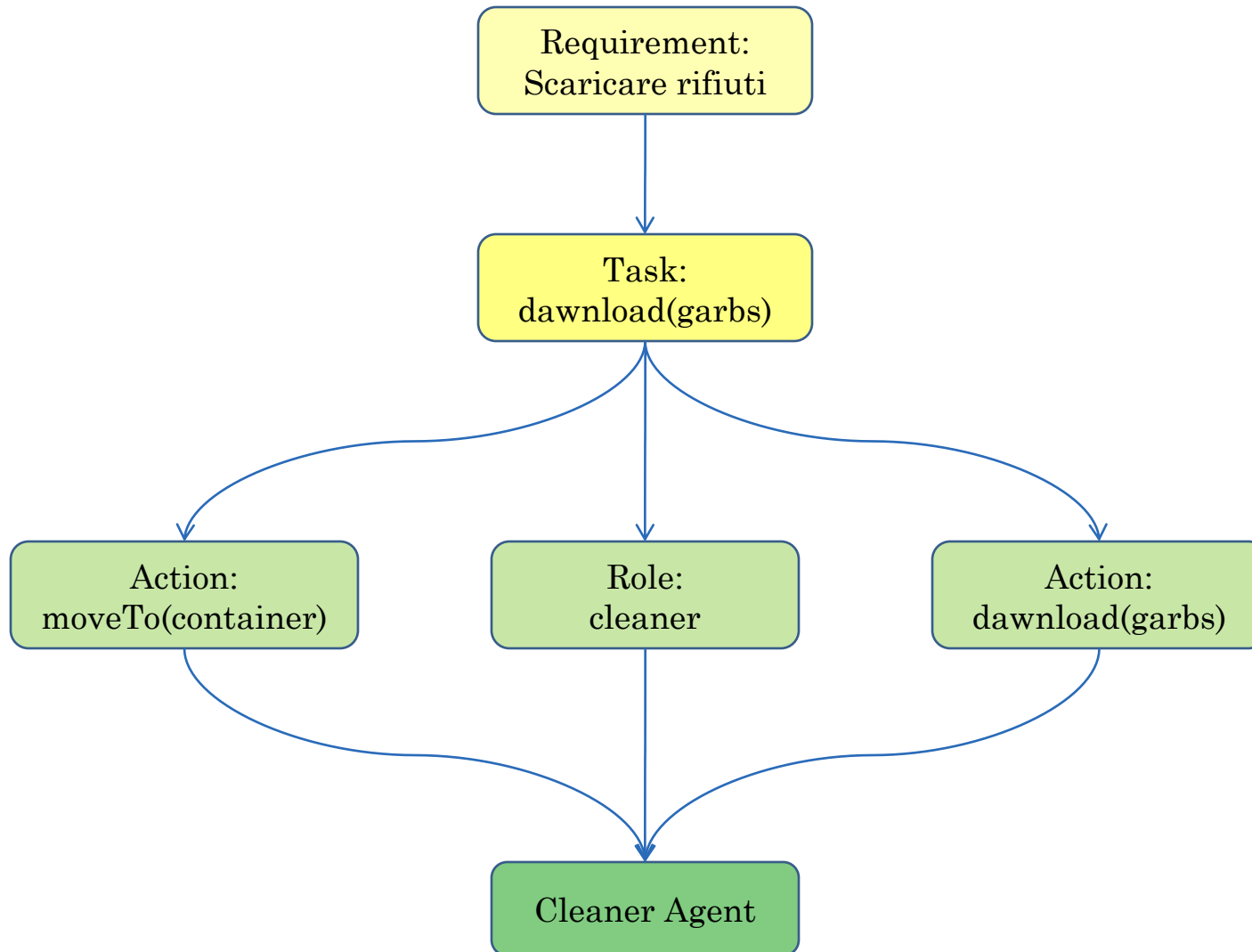
Un agente può accedere ad un container solo se questo è disponibile, cioè non occupato da un'altra operazione di download, non pieno e presente.



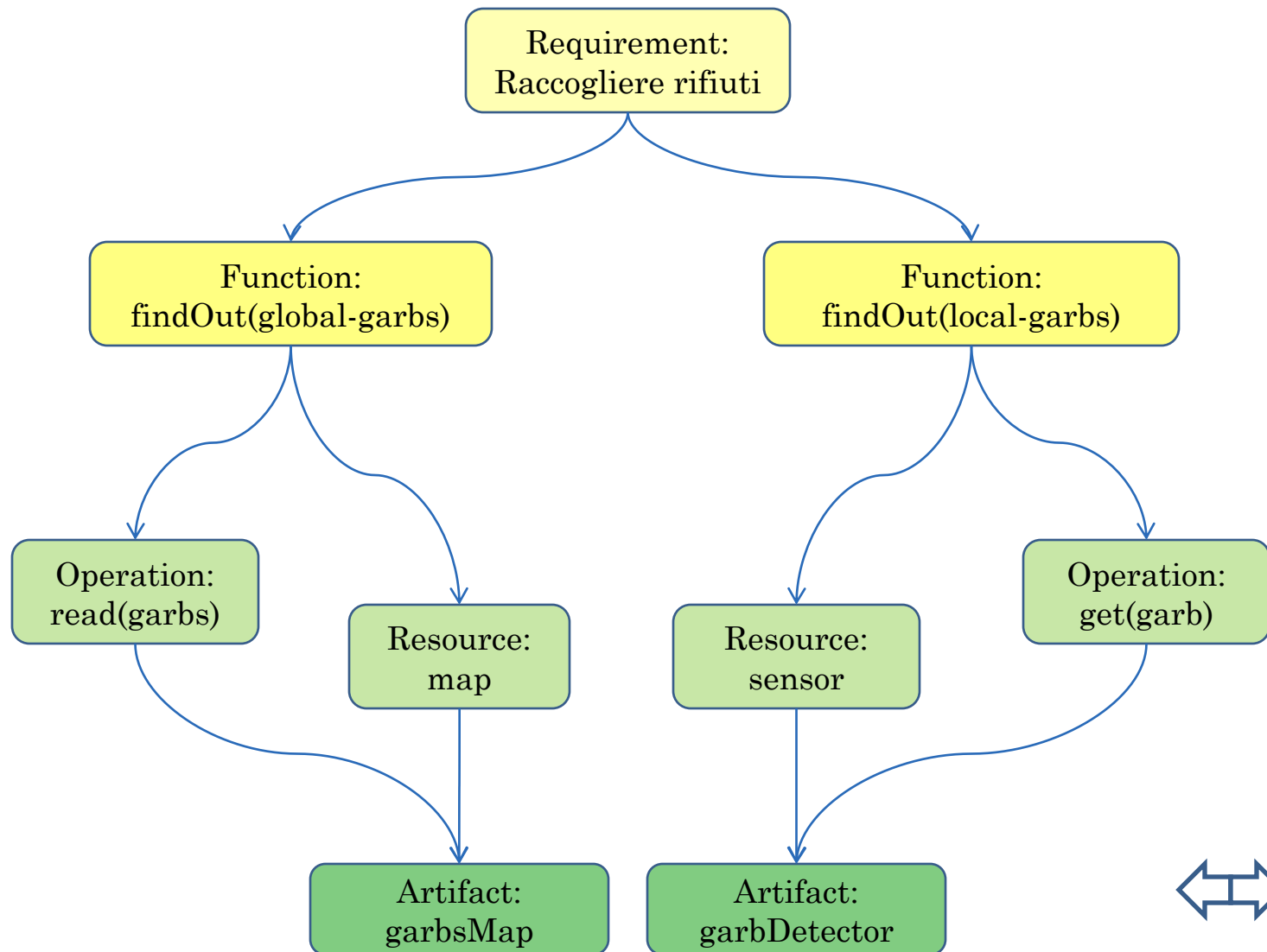
SODA Active Abstractions



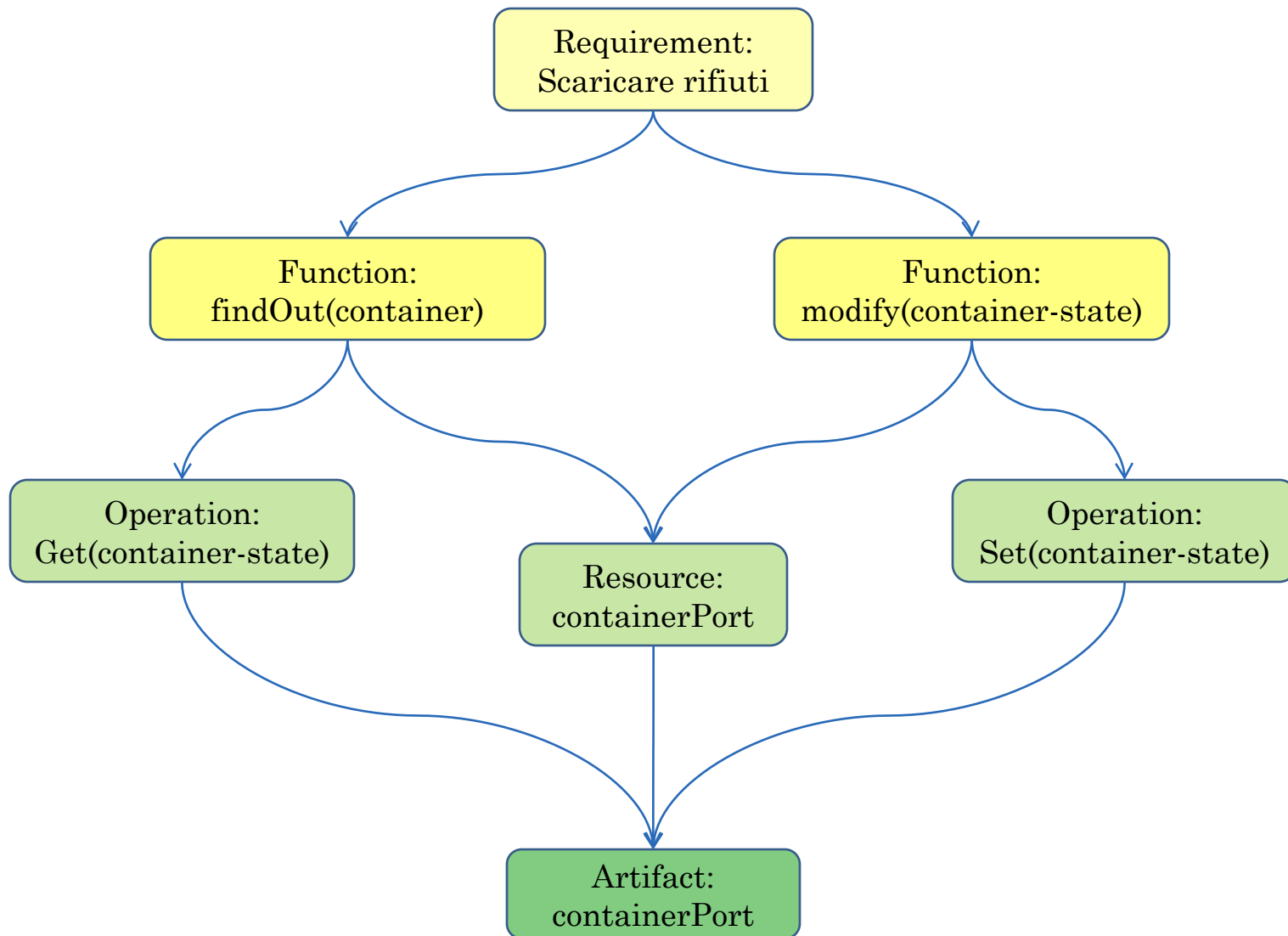
SODA Active Abstractions



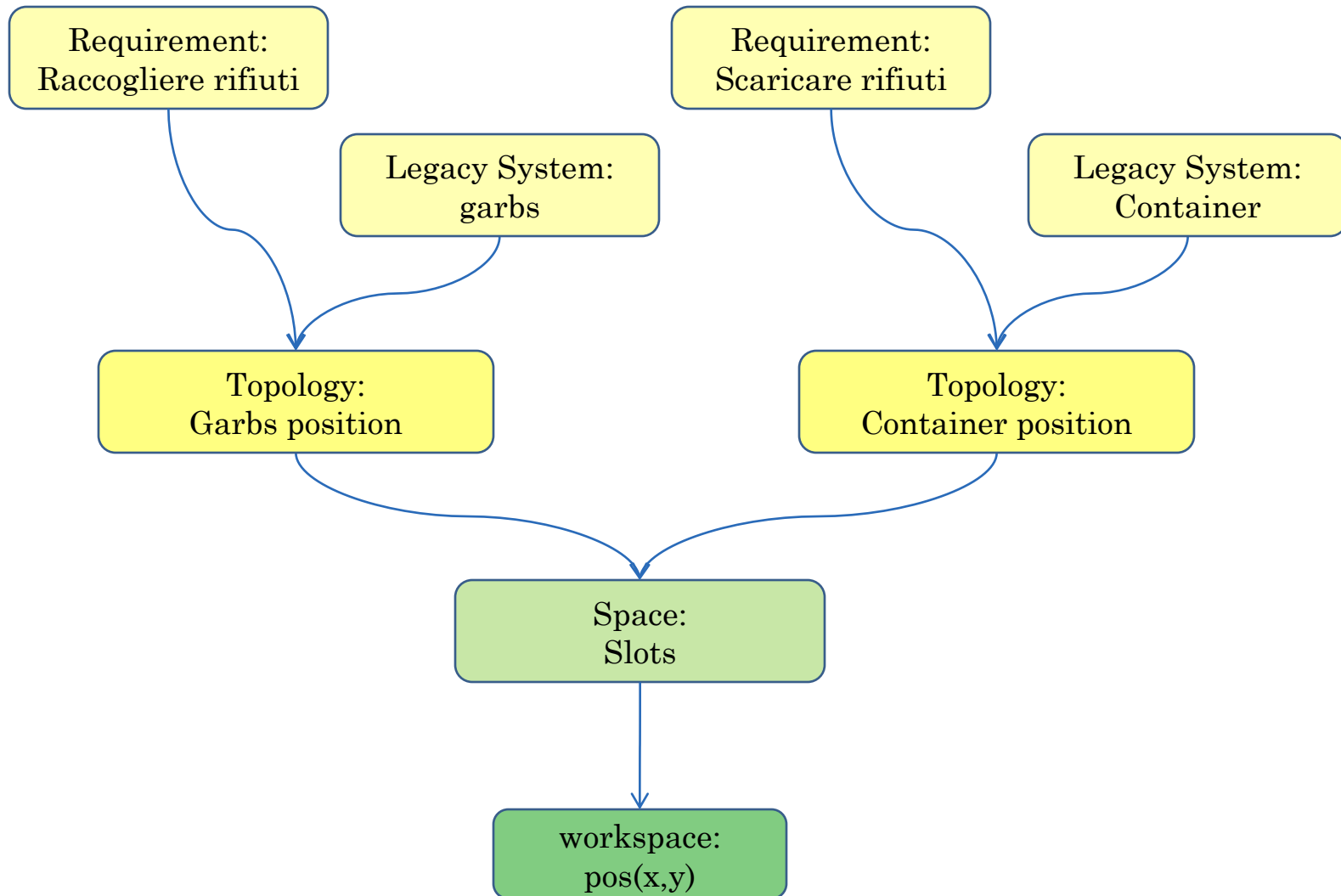
SODA Reactive Abstractions



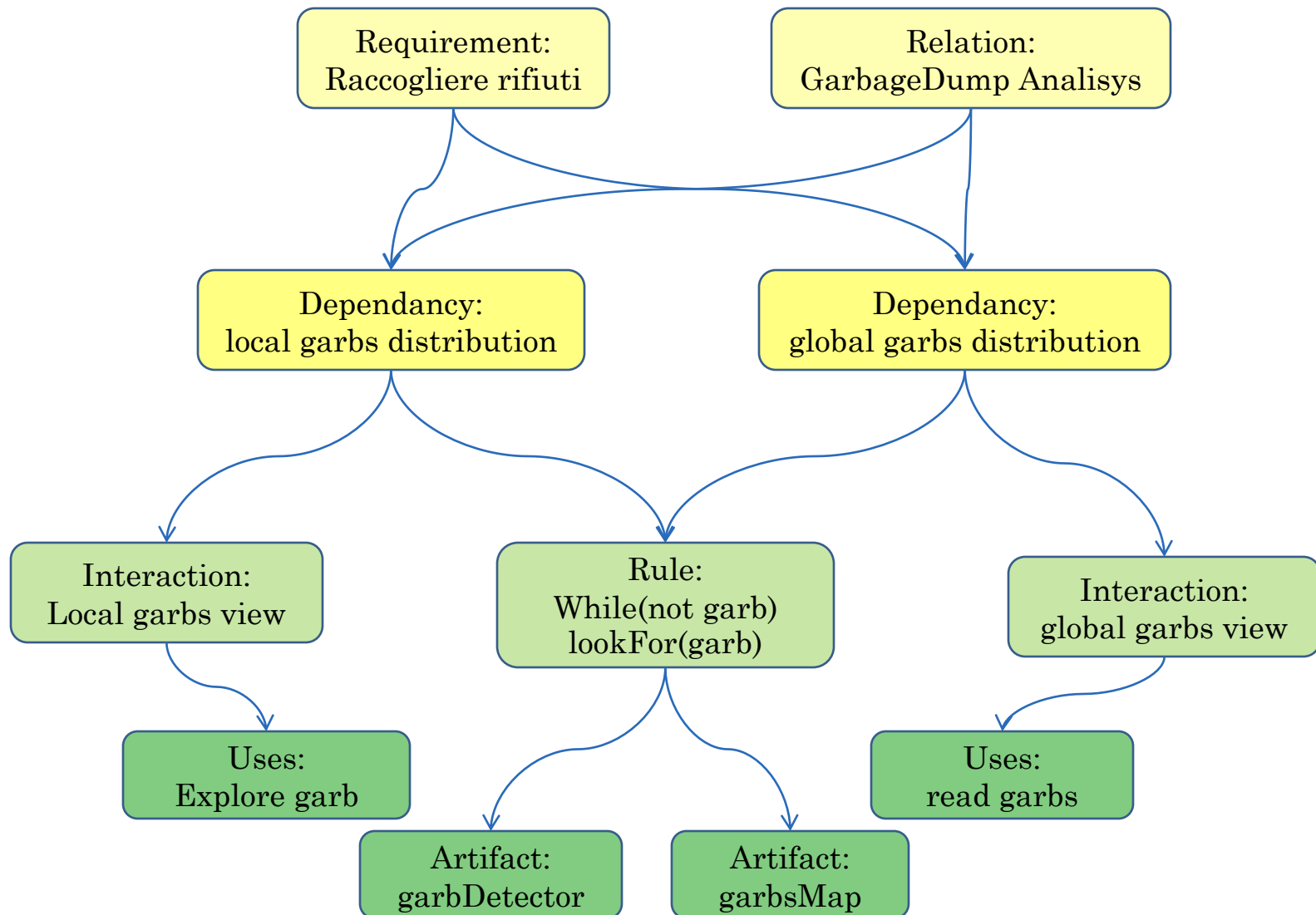
SODA Reactive Abstractions



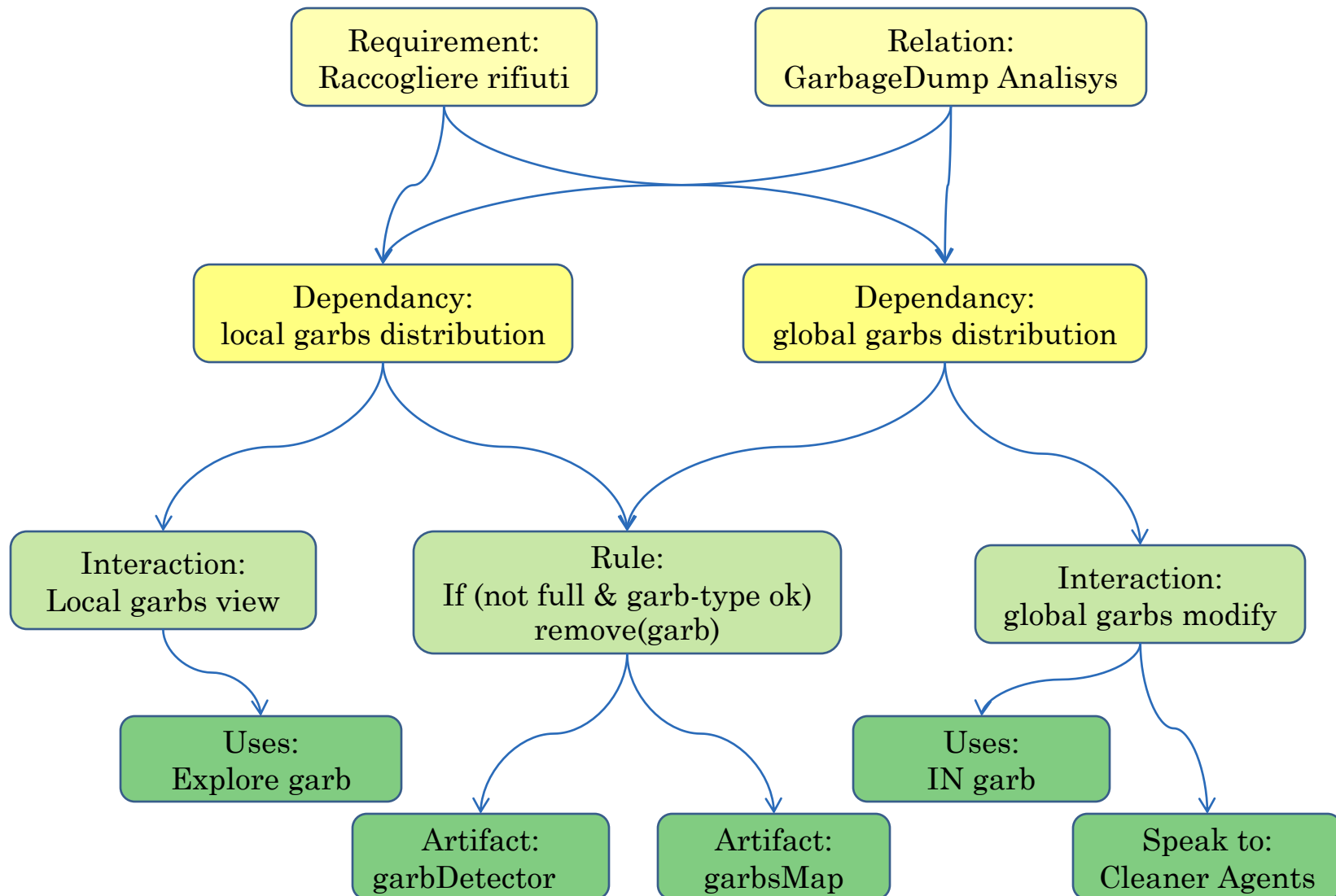
SODA Reactive Abstractions



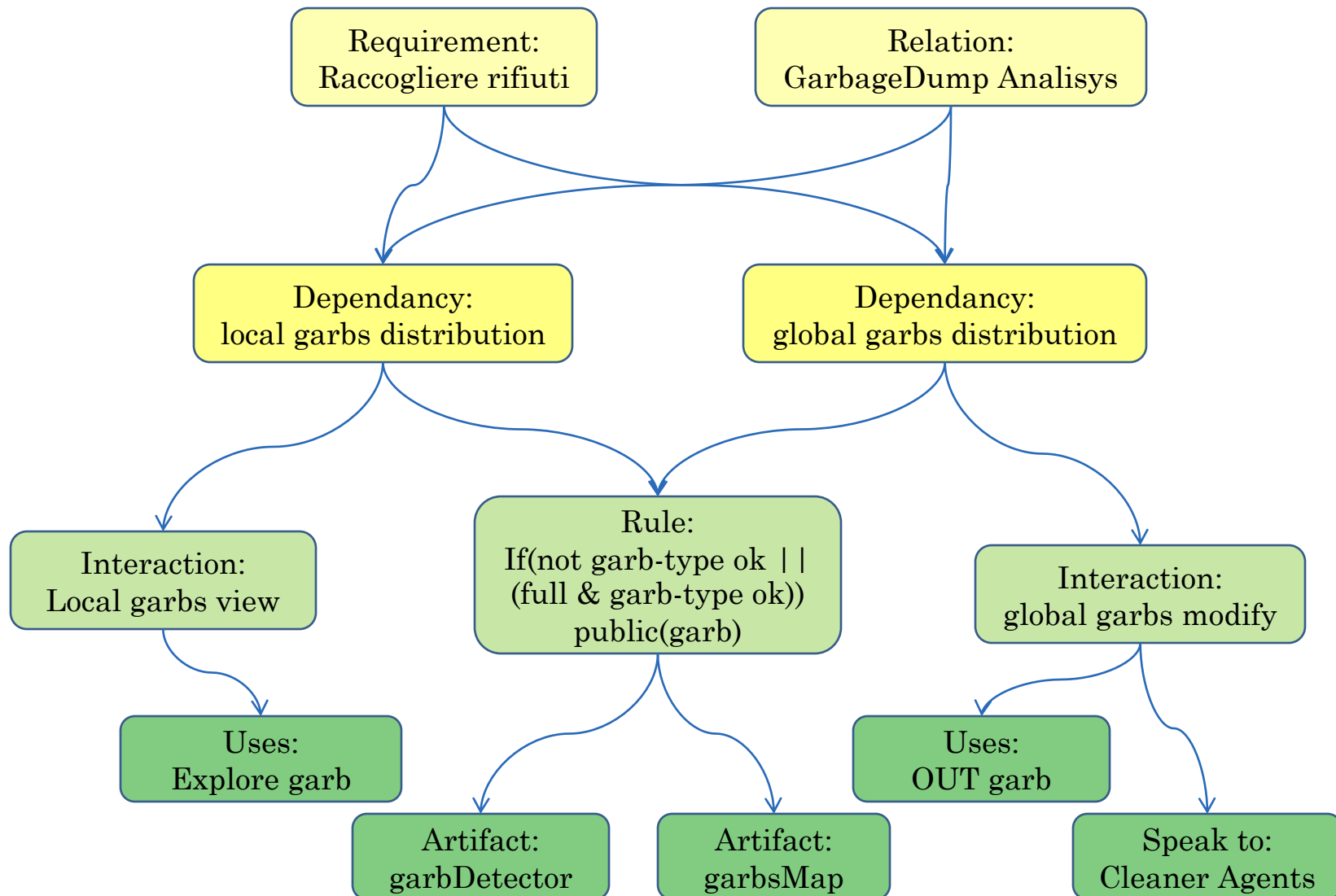
SODA Interaction Abstractions



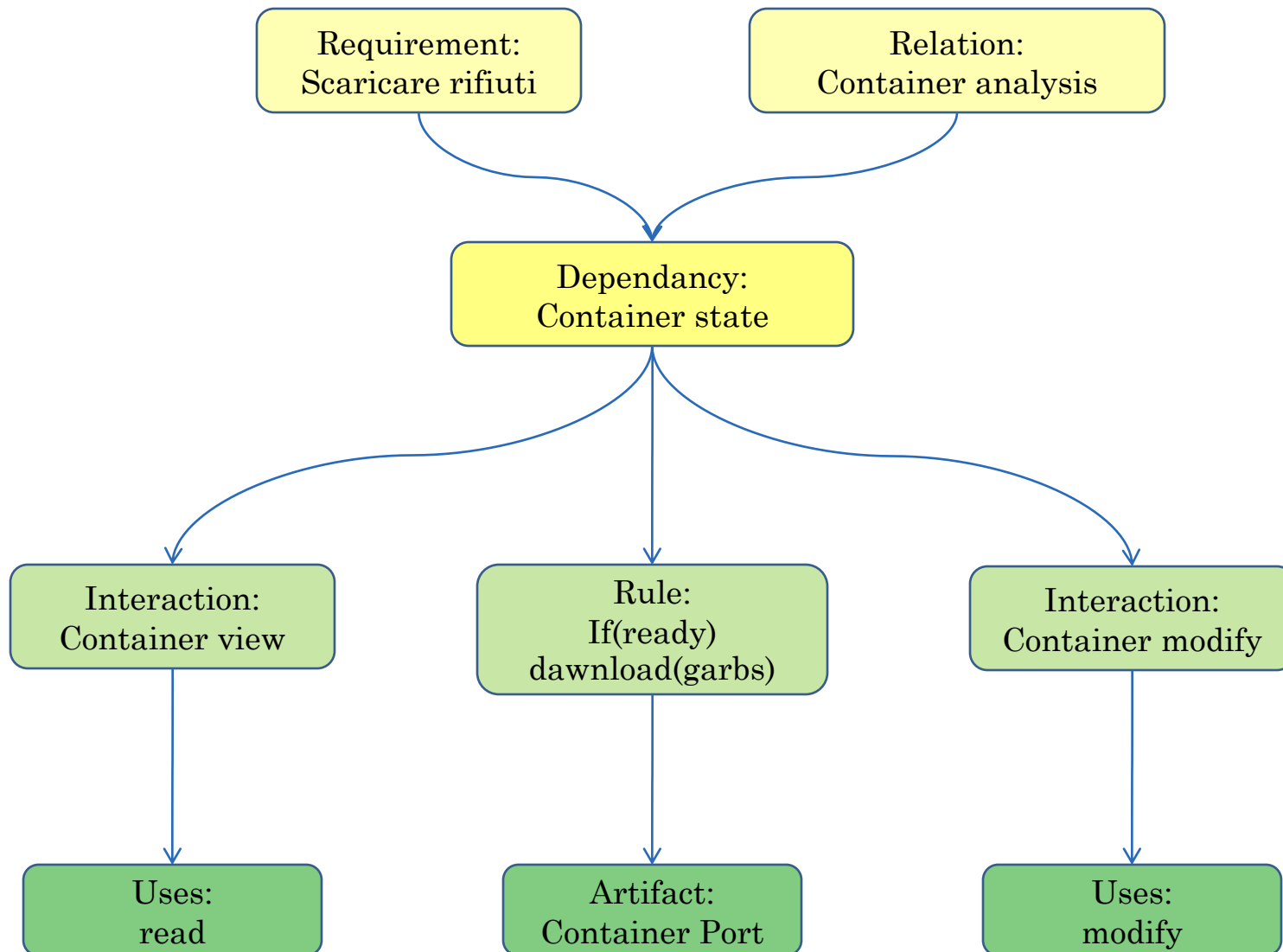
SODA Interaction Abstractions



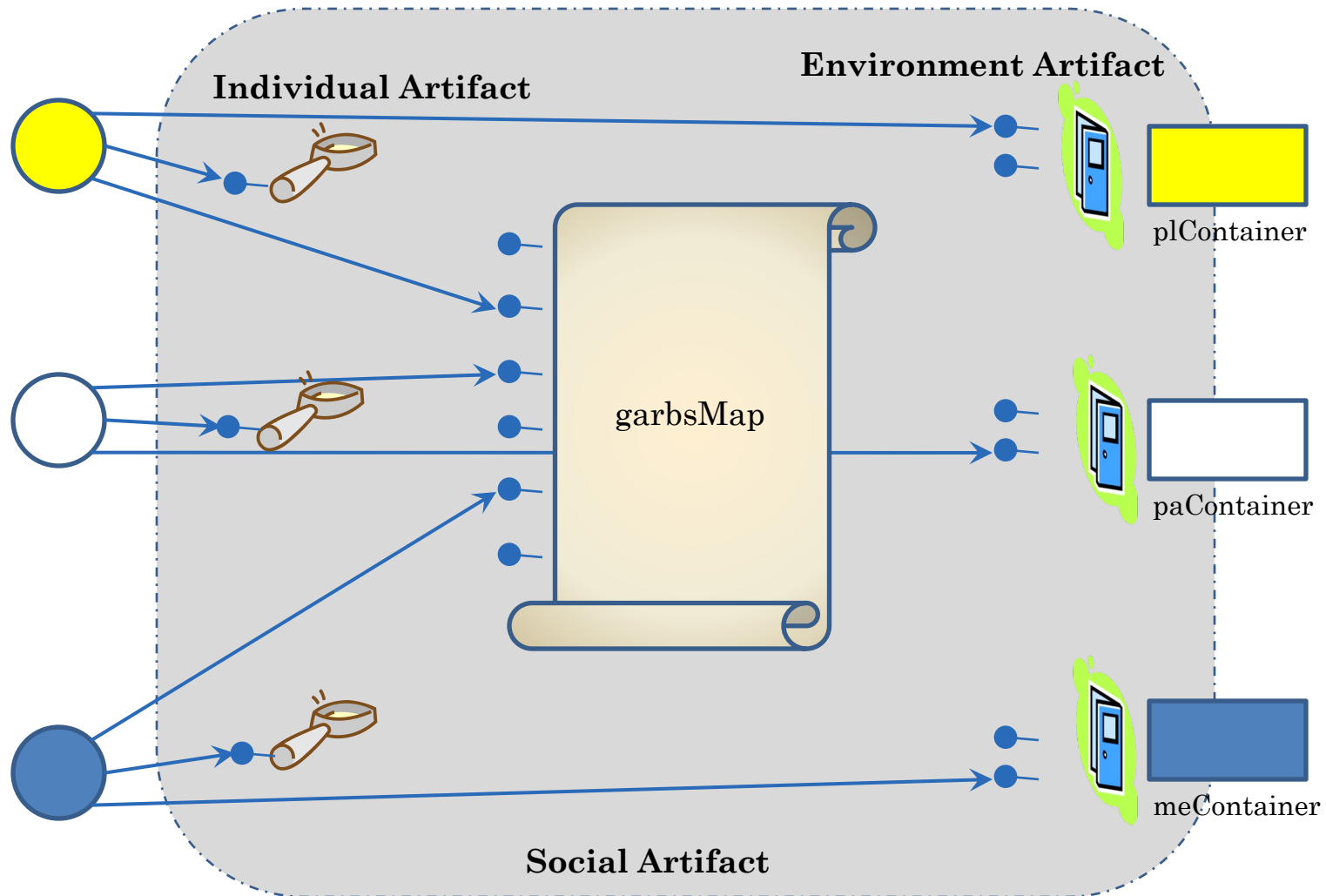
SODA Interaction Abstractions



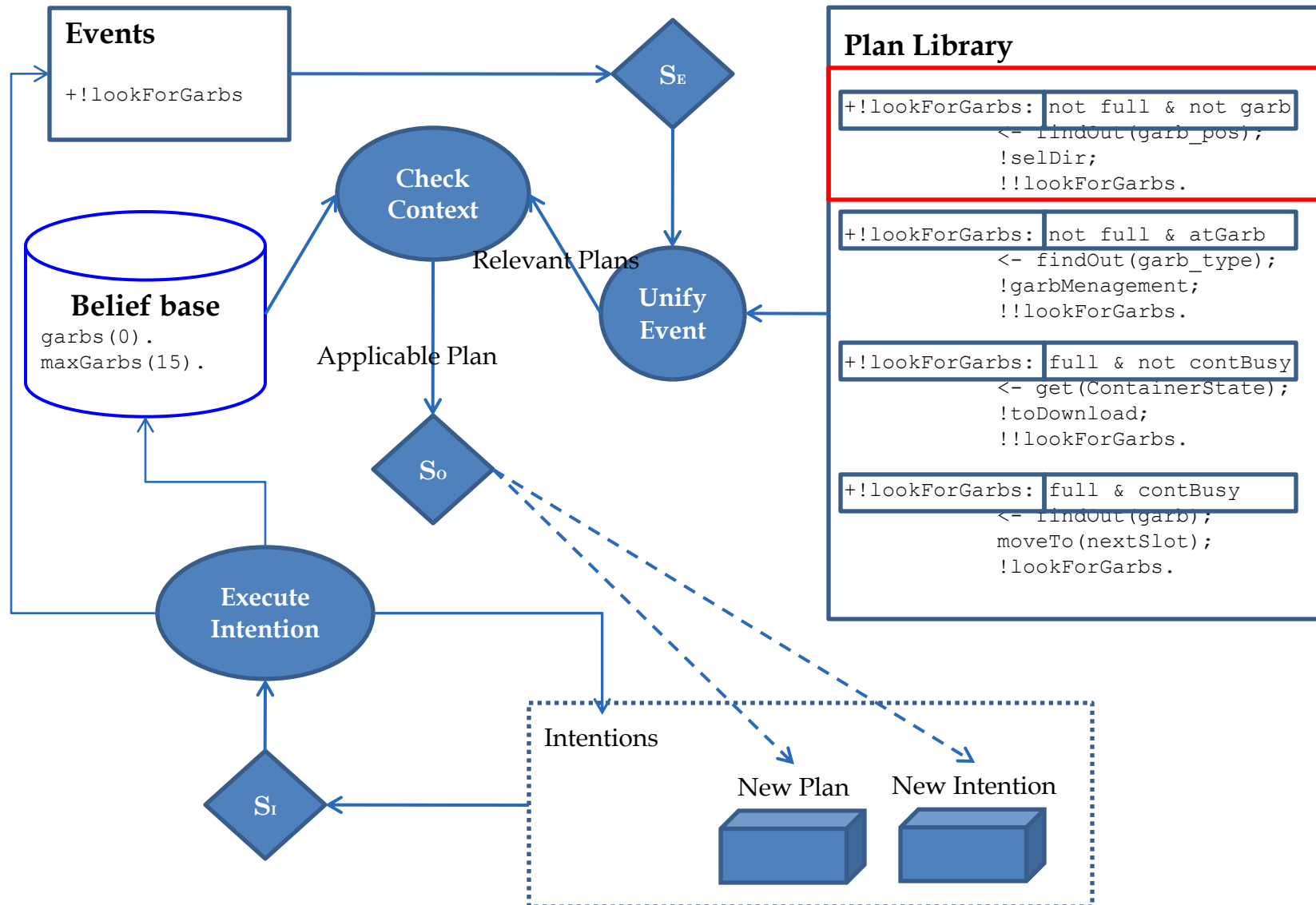
SODA Interaction Abstractions



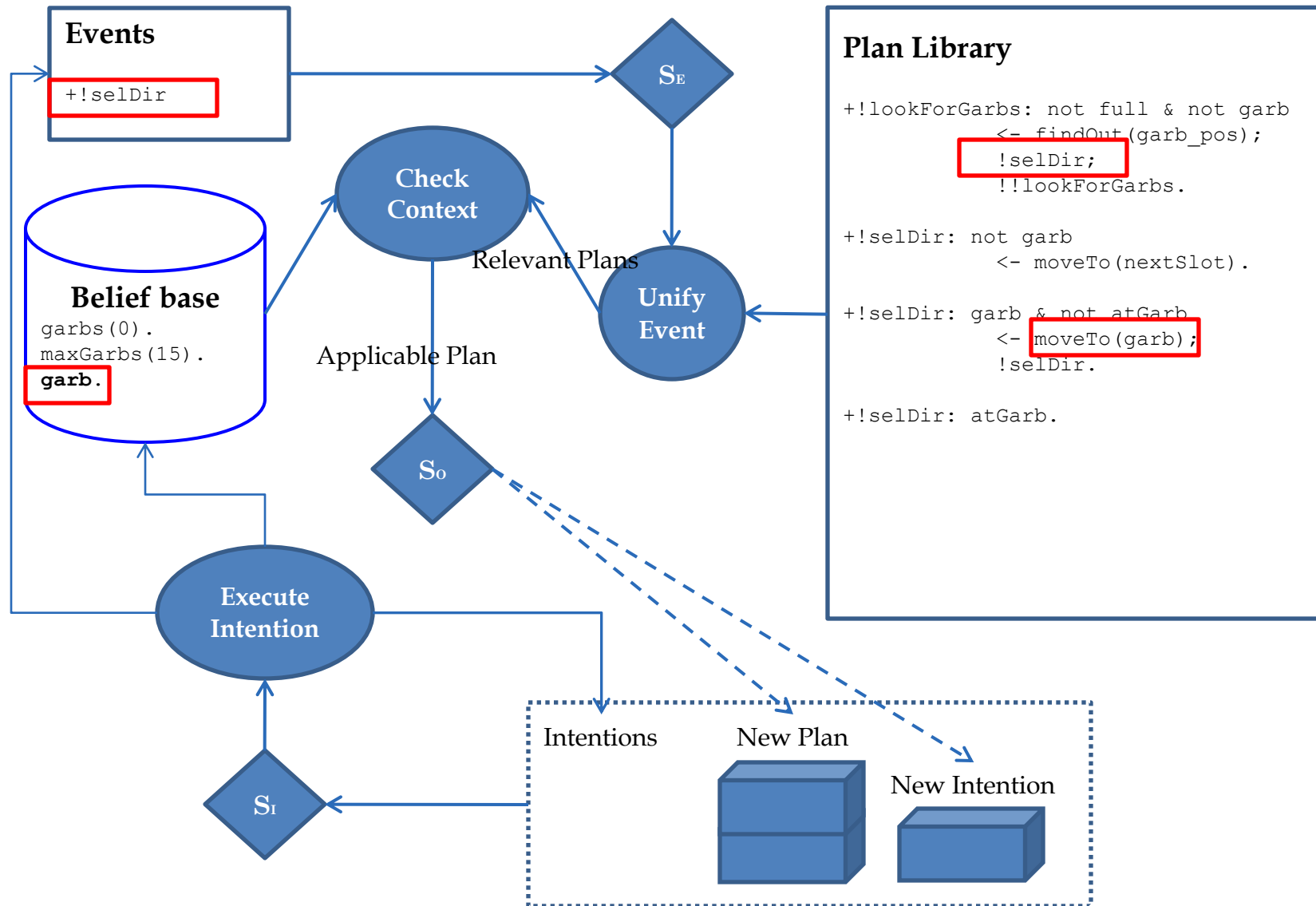
Agent Workspace



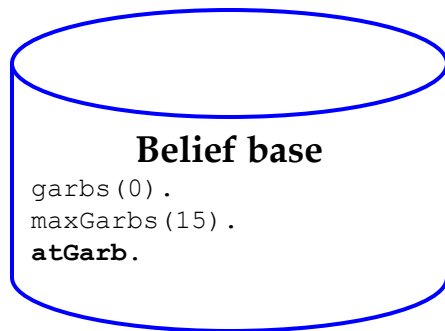
Jason for BDI Agents



Jason for BDI Agents



Jason for BDI Agents



Plan Library

```
+!lookForGarbs: not full & atGarb  
  <- findOut(garb_type)  
  !garbMenagement;  
  !!lookForGarbs.  
  
+!garbMenagement: garbType_OK  
  <- remove(garb);  
  ?garbs(X);  
  ?maxGarbs(Y);  
  !setState(X,Y).  
  
+!garbMenagement: ~garbType_OK  
  <- public(garb).  
  
+!lookForGarbs: full & not contBusy  
  <- get(ContainerState);  
  !toDownload;  
  !!lookForGarbs.  
  
+!toDownload: not atCont & not contBusy  
  <- moveTo(container);  
  !toDownload.  
  
+!toDownload: atCont & not contBusy  
  <- download(garbs);  
  ?garbs(X);  
  .print("garbs removed:",X);  
  -+garbs(0);  
  -full.  
  
...
```

The Plan Library contains several plans. The first plan, `+!lookForGarbs: not full & atGarb`, is highlighted with a red box. A blue arrow points from this plan to the **Belief base** cylinder, indicating that the plan's conditions are satisfied by the beliefs in the belief base.

Jason for BDI Agents

Belief base

garbs(0).
maxGarbs(15).
garbType_OK.

garbType_OK 0
~garbType_OK ?

Come decide un
CleanerAgent?

Plan Library

```
+!lookForGarbs: not full & atGarb
    <- findOut(garb_type);
    !garbMenagement;
    !!lookForGarbs.

+!garbMenagement: garbType_OK
    <- remove(garb);
    ?garbs(X);
    ?maxGarbs(Y);
    !setState(X,Y).

+!garbMenagement: ~garbType_OK
    <- public(garb).

+!lookForGarbs: full & not contBusy
    <- get(ContainerState);
    !toDownload;
    !!lookForGarbs.

+!toDownload: not atCont & not contBusy
    <- moveTo(container);
    !toDownload.

+!toDownload: atCont & not contBusy
    <- download(garbs);
    ?garbs(X);
    .print("garbs removed:",X);
    -+garbs(0);
    -full.
```

...

Individual Artifact: garbDetector

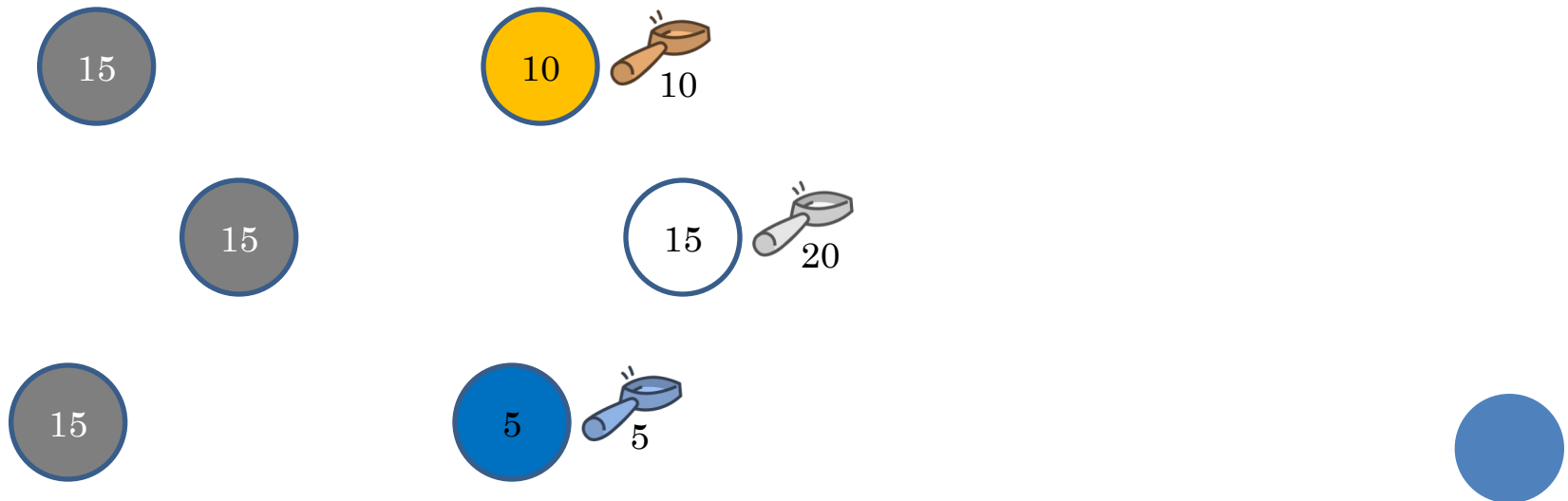


Autonomia dell'agente:

- Cerca e scarica
- Si muove
- Raccoglie massimo 15 rifiuti alla volta

Vincoli imposti dal Sistema:

- Raccogliere al più 10 plGarbs, 20 paGarb e 5 meGarb alla volta
- Raccogliere rifiuti dello stesso tipo

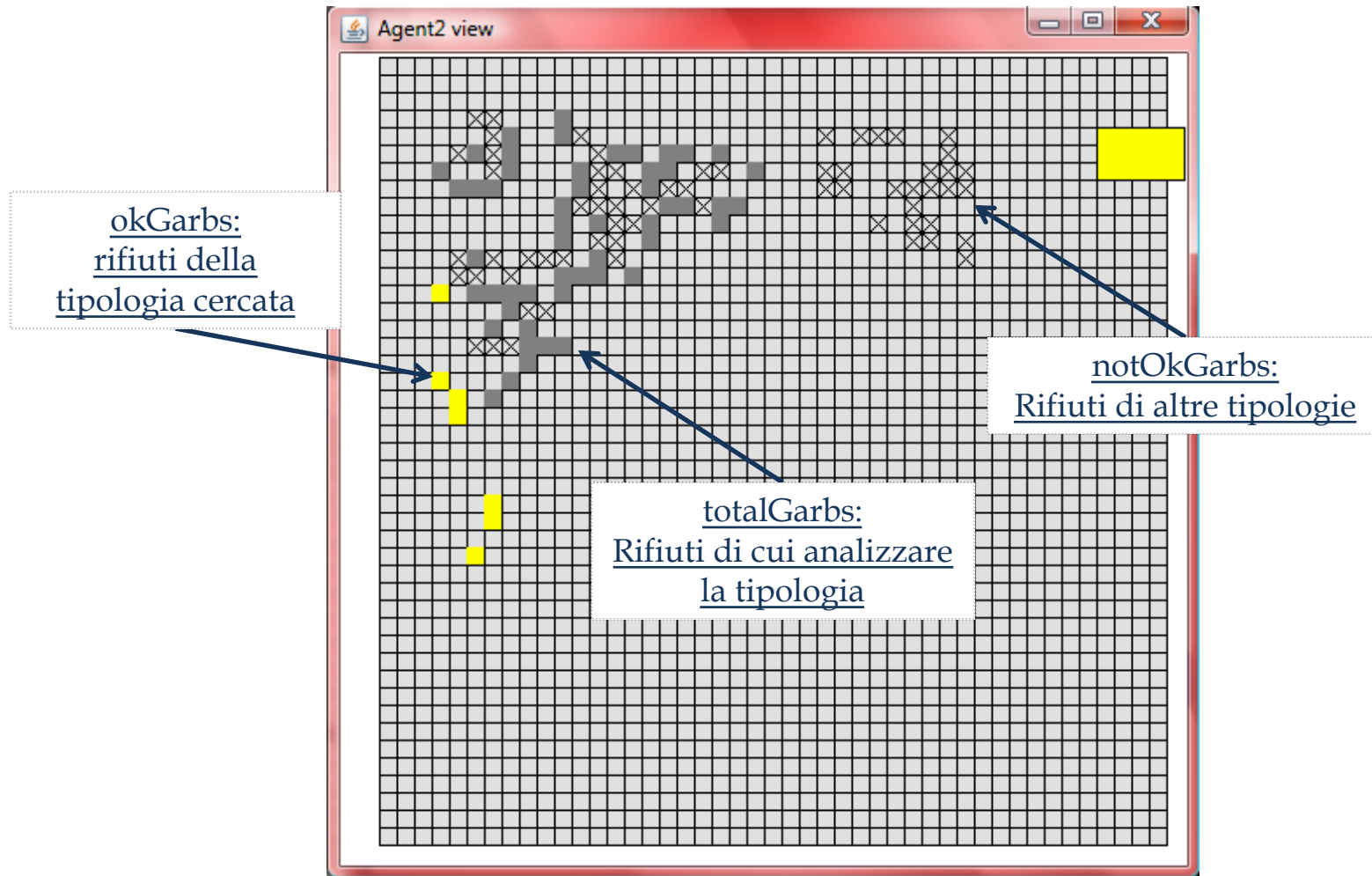


Individual Artifact: garbDetector



Mediante il garbDetector ogni agente si costruisce una visione individuale dell'ambiente;

Ad esempio un plAgent potrebbe vedere in un certo istante:



Social Artifact/1

Senza un artefatto sociale è difficile definire politiche di coordinazione o regole di interazione sociale.

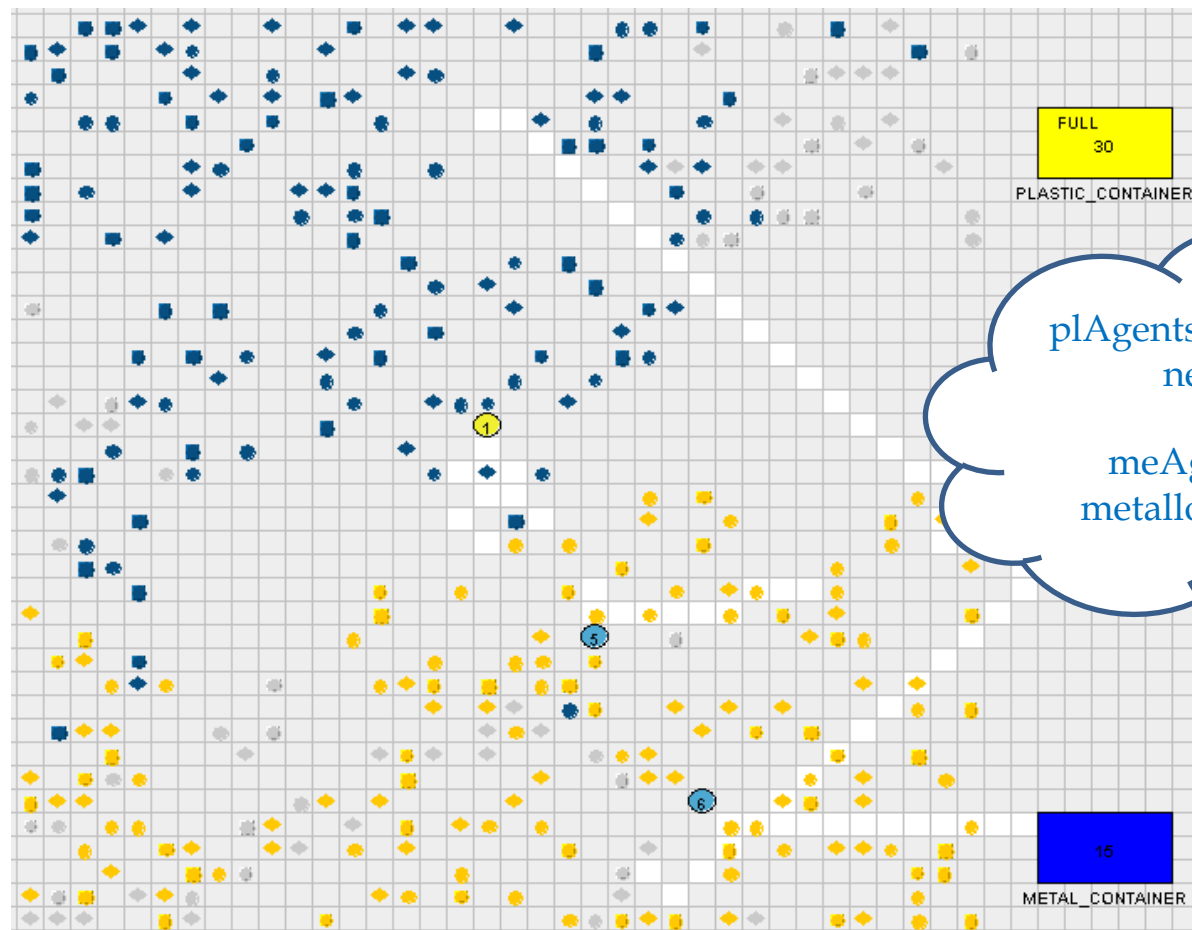
Ad esempio cosa potrebbe succedere?



Social Artifact/2

Senza un artefatto sociale è difficile definire politiche di coordinazione o regole di interazione sociale.

Ad esempio cosa potrebbe succedere?



plAgents cercano plastica
nel metallo!

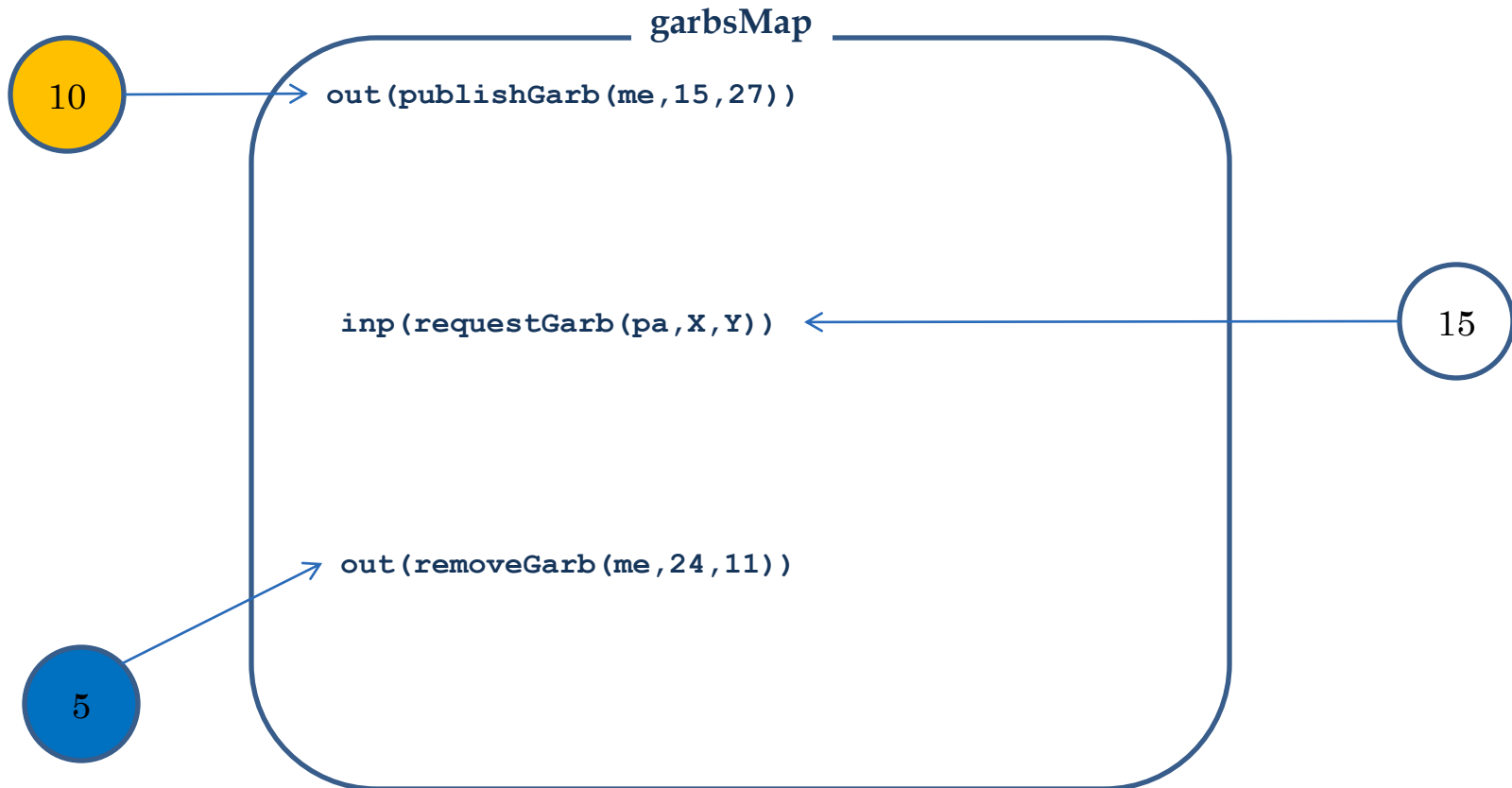
meAgents cercano
metallo nella plastica!



Tuple Centre for garbsMap

Cosa deve fare un cleanerAgent?

- Pubblicare notOkGarbs ed okGarbs che non è in grado di raccogliere
- Rimuovere okGarbs raccolti
- Fare richiesta di okGarbs se la sua ricerca non ha esito positivo



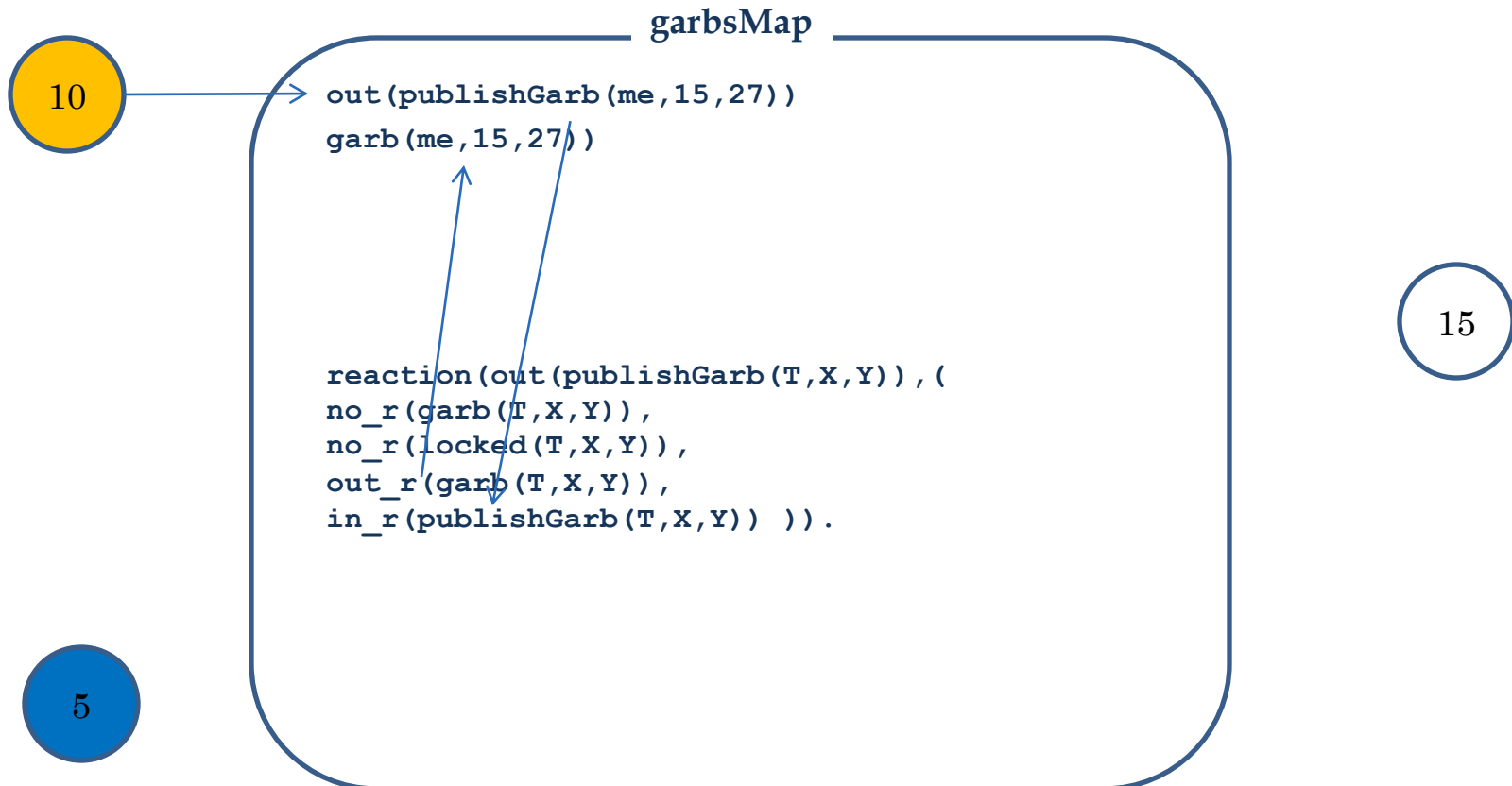
Tuple Centre for garbsMap

Come realizzare la coordinazione?

Definendo il comportamento del centro di tuple!



Le regole di coordinazione non pesano sugli agenti



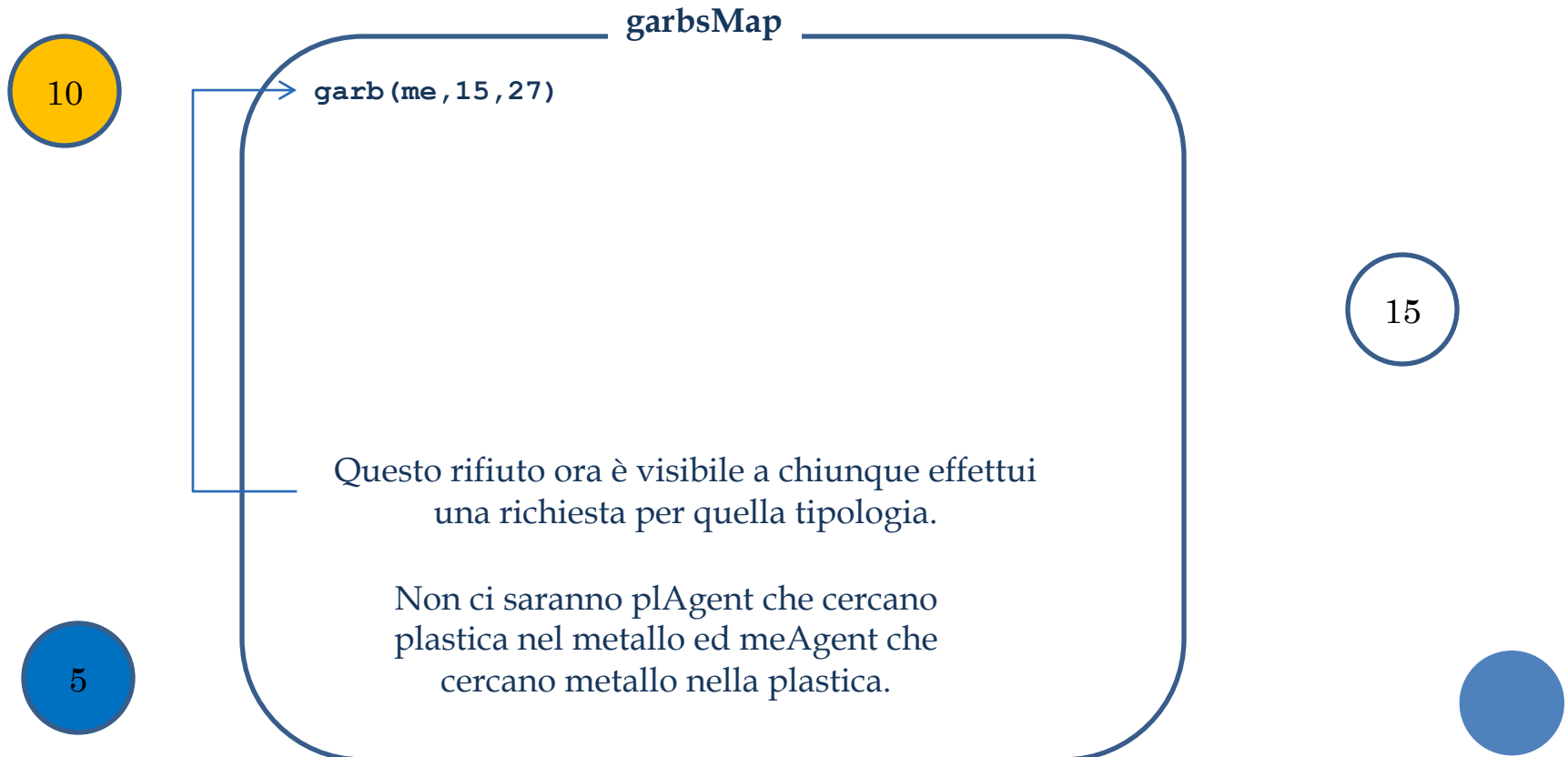
Tuple Centre for garbsMap

Come realizzare la coordinazione?

Definendo il comportamento del centro di tuple!



Le regole di coordinazione non pesano sugli agenti



Tuple Centre for garbsMap

Come realizzare la coordinazione?

Definendo il comportamento del centro di tuple!



Le regole di coordinazione non pesano sugli agenti



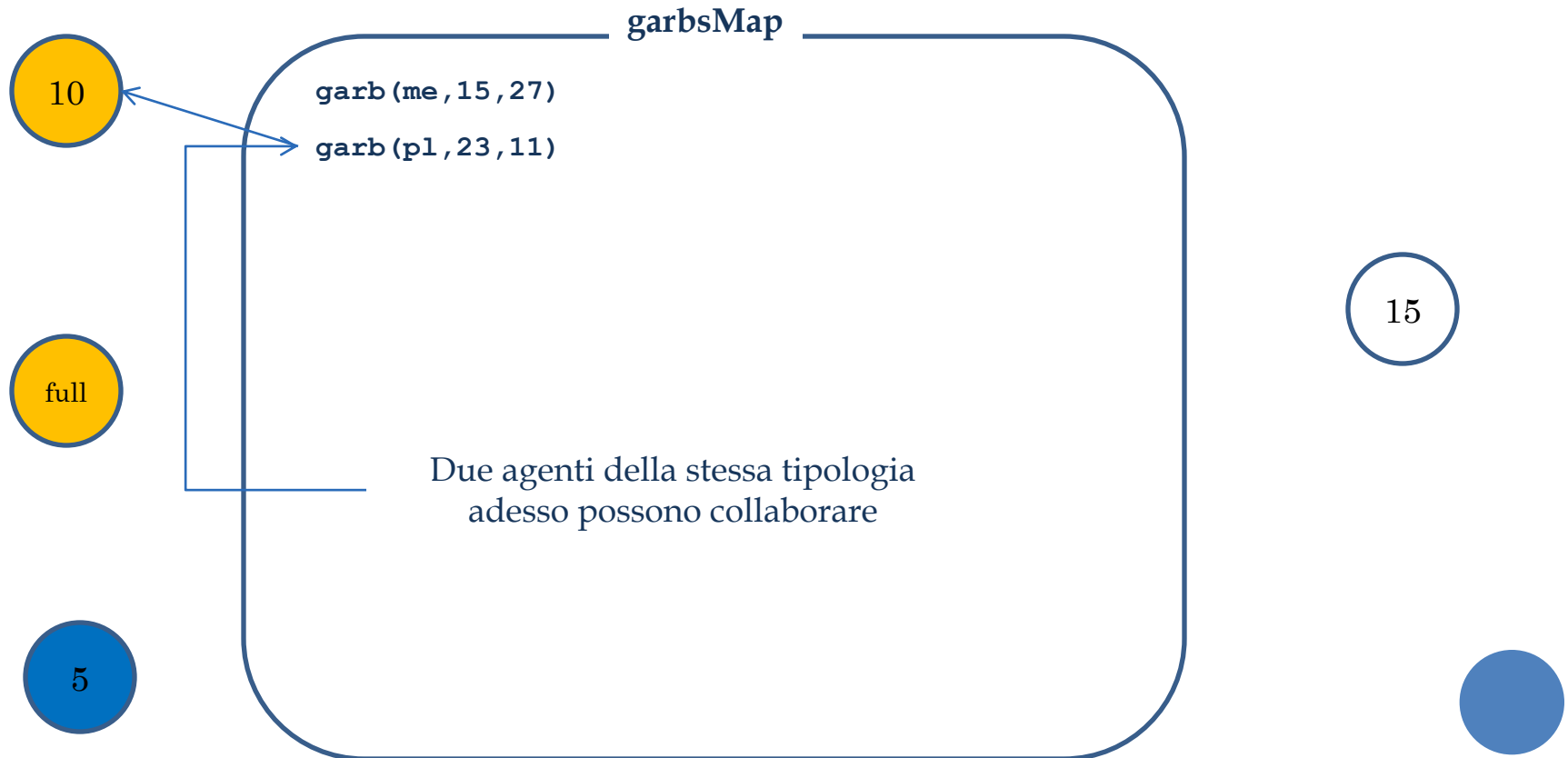
Tuple Centre for garbsMap

Come realizzare la coordinazione?

Definendo il comportamento del centro di tuple!



Le regole di coordinazione non pesano sugli agenti



Tuple Centre for garbsMap

Come realizzare la coordinazione?

Definendo il comportamento del centro di tuple!



Le regole di coordinazione non pesano sugli agenti

garbsMap

`garb (me , 15 , 27)`

`garb (pl , 23 , 11)`

`inp (requestGarb (pa , X , Y) )`

`reaction (inp (garbRequest (T , X , Y)) , (pre ,
rd_r (garb (T , X , Y)) ,
out_r (locked (T , X , Y)) ,
in_r (garb (T , X , Y)) ,
out_r (garbRequest (T , X , Y))))`

15

10

5

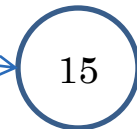
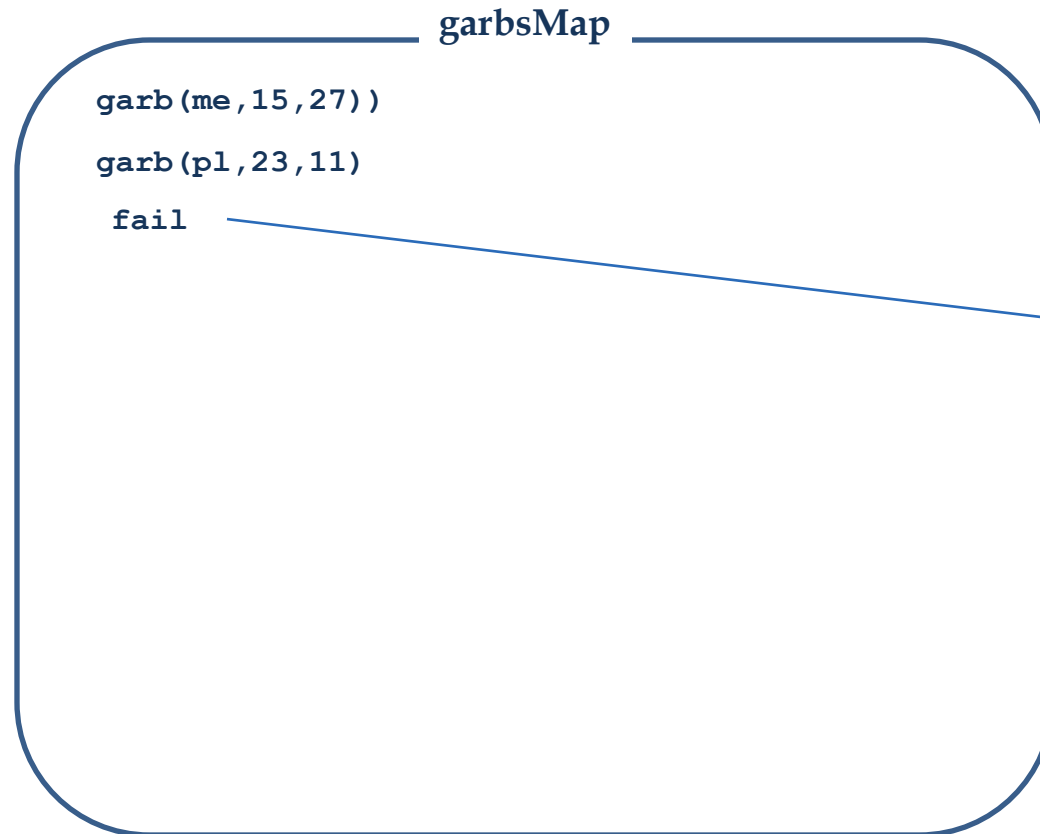
Tuple Centre for garbsMap

Come realizzare la coordinazione?

Definendo il comportamento del centro di tuple!



Le regole di coordinazione non pesano sugli agenti



Tuple Centre for garbsMap

Come realizzare la coordinazione?

Definendo il comportamento del centro di tuple!



Le regole di coordinazione non pesano sugli agenti



Tuple Centre for garbsMap

Come realizzare la coordinazione?

Definendo il comportamento del centro di tuple!



Le regole di coordinazione non pesano sugli agenti

garbsMap

garb(me,15,27))

garb(pl,23,11)

garb(pa,3,19)

locked(pa,3,19)

inp(requestGarb(pa,X,Y))

reaction(inp(garbRequest(T,X,Y)), (pre,
rd_r(garb(T,X,Y)),
out_r(locked(T,X,Y)),
in_r(garb(T,X,Y)),
out_r(garbRequest(T,X,Y)))

15

10

5

Tuple Centre for garbsMap

Come realizzare la coordinazione?

Definendo il comportamento del centro di tuple!



Le regole di coordinazione non pesano sugli agenti

garbsMap

`garb (me , 15 , 27) )`

`locked (pa , 3 , 19)`

`requestGarb (pa , 3 , 19)`

Questo rifiuto appartiene ad un agente!
Operazioni di pubblicazione o di richiesta da
parte di altri agenti falliranno.

L'agente pubblica, rimuove o richiede informazioni
riguardanti tipologia e locazione. Queste vengono
rappresentate nel centro di tuple in modo diverso in base
al comportamento che si desidera ottenere.

15

10

5

Tuple Centre for garbsMap

Come realizzare la coordinazione?

Definendo il comportamento del centro di tuple!



Le regole di coordinazione non pesano sugli agenti

garbsMap

```
garb(me,15,27)
locked(pa,3,19)
```

```
reaction(out(downloaded(A,N)), (
  in_r(nDownloaded(A,X)),
  N1 is N+X,
  out_r(nDownloaded(A,N1)),
  in_r(downloaded(A,N)) ))
```

```
out(downloaded(meAgent3,5))
```

10

15

5

Tuple Centre for garbsMap

Come realizzare la coordinazione?

Definendo il comportamento del centro di tuple!



Le regole di coordinazione non pesano sugli agenti

garbsMap

`garb (me , 15 , 27) )`

`locked (pa , 3 , 19)`

Il centro di tuple consente una
maggiore osservabilità sul modo
di lavorare di ciascun agente

`nDawnloaded (plAgent3 , 35) )`

10

15

5

riassumendo

Attraverso il garbDetector

- **Gli agenti percepiscono e modificano l'ambiente, ciascuno secondo il proprio obiettivo.**
- Non sono evitabili situazioni in cui l'ottimo lavoro di un agente può intralciare quello degli altri.
- Il tempo necessario per il raggiungimento della pulizia complessiva della discarica potrebbe essere lungo.

Attraverso la garbsMap

- **Gli agenti cooperano nel conseguimento di un obiettivo comune come vere e proprie parti di un sistema, condividendo informazioni e seguendo le regole imposte dall'artefatto sociale.**
- **E' possibile tracciare la storia del sistema ed eventualmente scrutarne il comportamento.**
- **I tempi necessari al conseguimento dell'obiettivo del sistema si accorciano.**



DEMO

1- Install garbsMap

2- Run Garbage Dump

3- Inspect garbsMap



DEMO

Run GD_noCoordination

