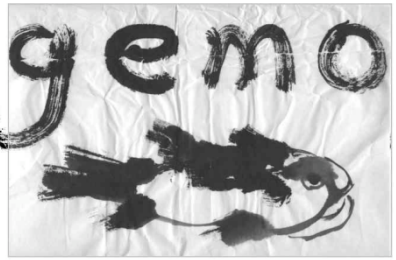
distributed consistency-based diagnosis

Vincent Armant, Philippe Dague, Laurent Simon

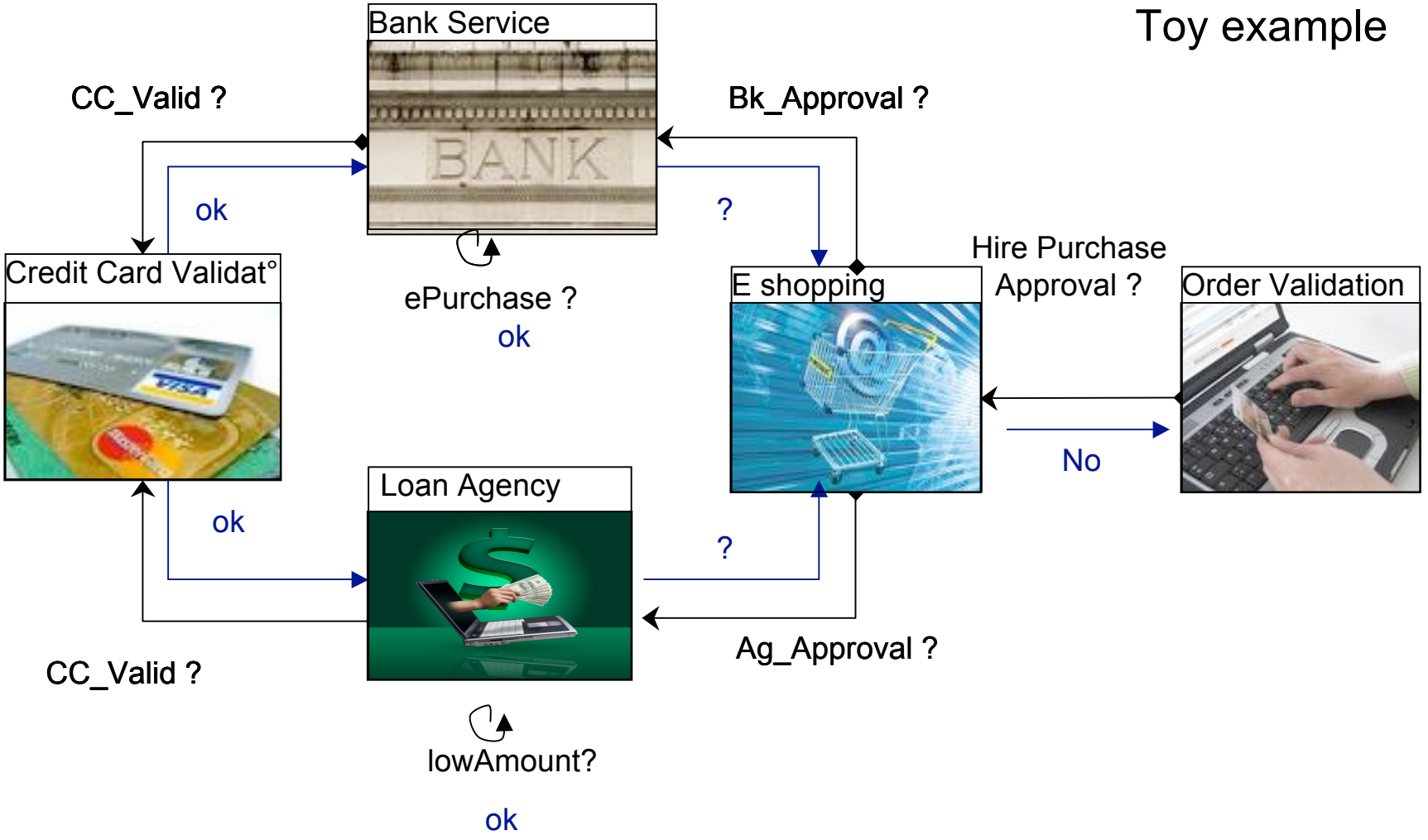


{vincent.armant, dague, simon}@lri.fr

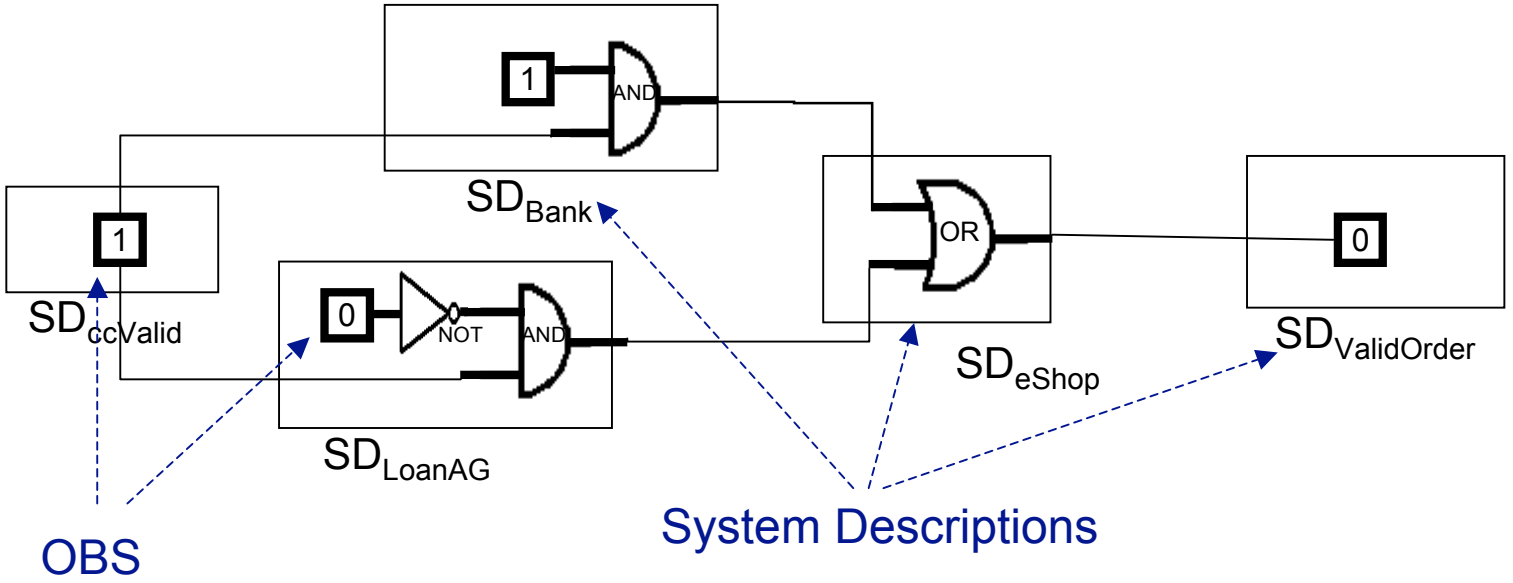
Road Map

- Consistency based diagnosis
- System description compilation
- Dealing with privacy and useless knowledge
- Our distributed, incremental, algorithm
- Conclusion and Perspectives

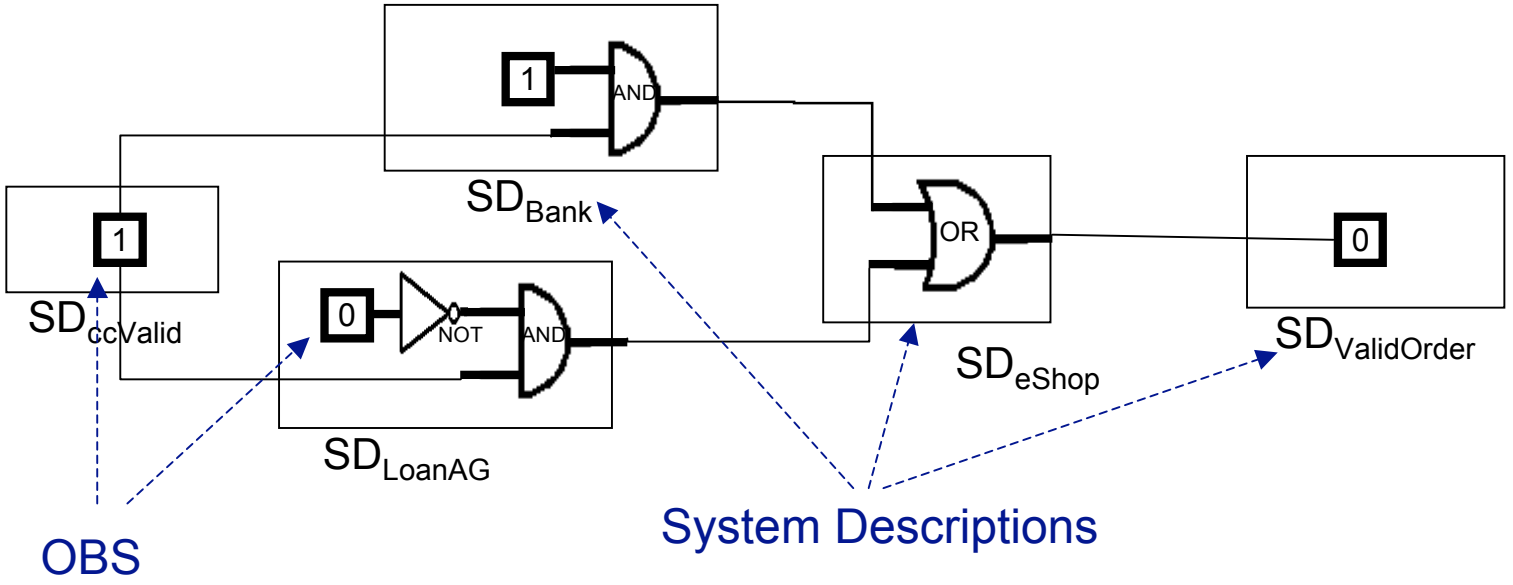
Three times web-payment certification



Modeling the behaviors



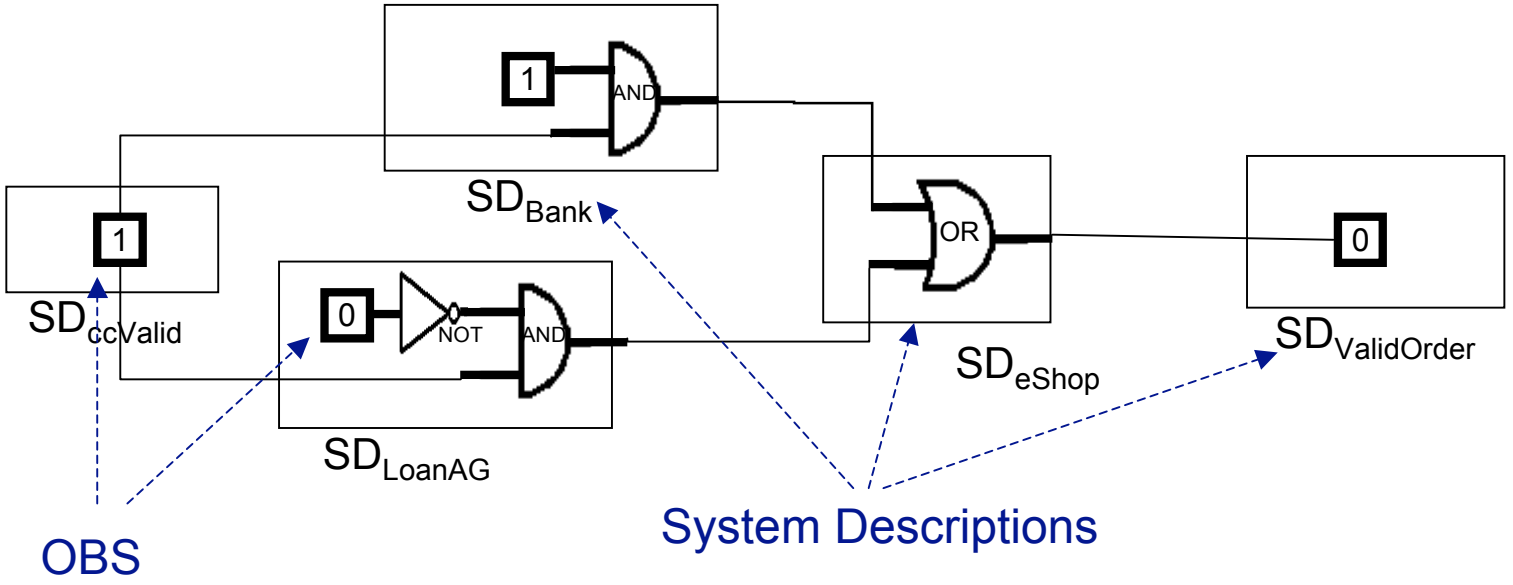
Distinguish shared and local knowledge



SD_{Bank} : $\neg ab(Bank) \Rightarrow ((ccValid \wedge e\_purchase) \Leftrightarrow bk\_approval)$

Variables: Mode Shared Local

Modeling the behaviors



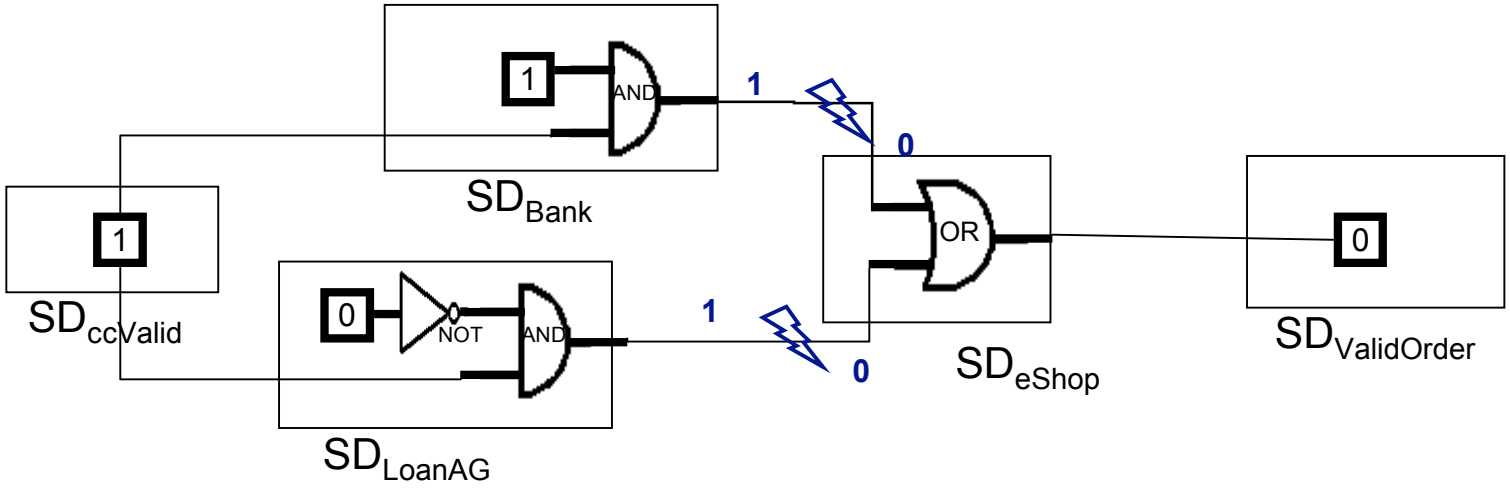
SD_{Bank} : $\neg ab(Bank) \Rightarrow ((ccValid \wedge \neg e\_purchase) \Leftrightarrow bk\_agreemt)$

Variables: Mode Shared Local

Global model : set of observations and local system descriptions

$$SD_{global} = \wedge SD_i \wedge OBS$$

Minimal conflicts



Minimal Conflict :

are components that are together inconsistent with observations

$$\bigwedge_i SD_i \wedge OBS \models C$$

s.t. ∀ C' conflict, if C' ⇒ C then C' = C

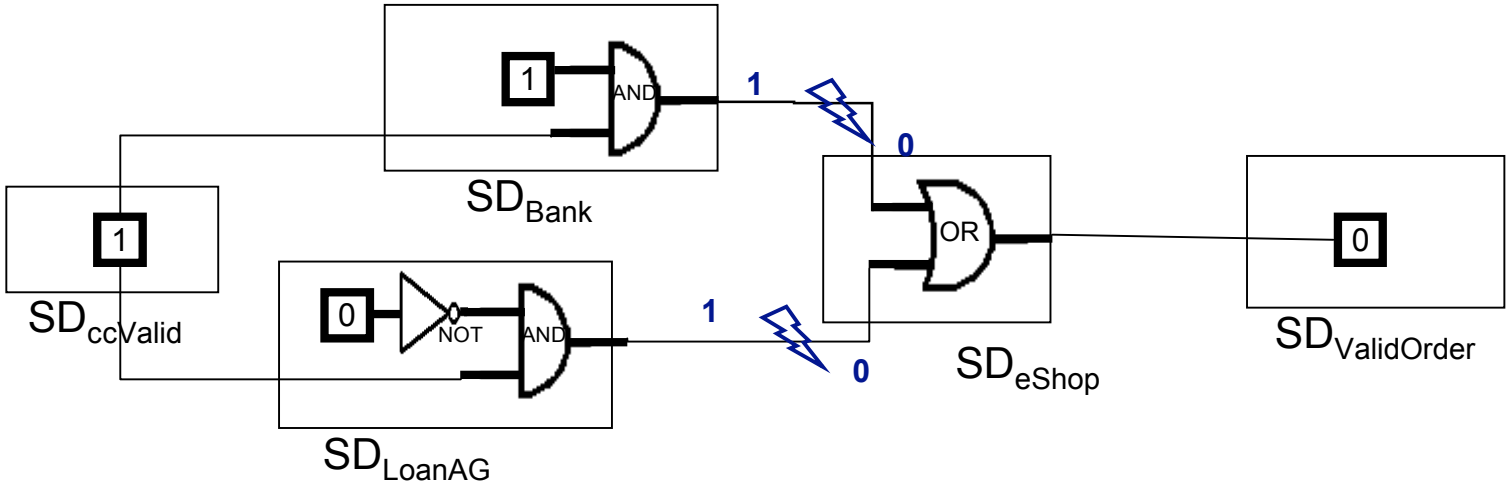
$C \subseteq F$, $F = \{ab1, \ldots, abn\}$

Example:

Ab(Bank) ∨ Ab(eShop)

Ab(LoanAg) ∨ Ab(eShop)

Minimal diagnoses



Minimal Diagnosis Δ :

Is a minimal explanation which cover **all** minimal conflicts

$\bigwedge_i SD_i \wedge OBS \wedge \Delta \wedge \overline{F \setminus \Delta} \not\models \bot$ s.t. ∀ Δ' diagnosis, if Δ' ⇒ Δ then Δ' = Δ

$\Delta \subseteq F$, $F = \{ab1, \ldots, abn\}$

Example:

Ab(Bank) ∧ Ab(LoanAg)

∨

Ab(eShop)

Challenge of distributed diagnosis

- Context : Distributed Algorithm
  - Each peer performs the same algorithm
  - The network incrementally returns diagnoses
  - The network topology is imposed
- Challenge : Global reasoning with local knowledge
  - A peer only know :
    - Its acquaintance
    - Its own description
  - A peer does not want to share some private knowledge
    - But must share any local knowledge that is “interesting” for the task

Distributed Diagnosis : related work

- Distributed model Based diagnosis
  - A model based diagnosis Framewor k for distributed System [Provan 02], [Kurien 02],…
    - Takes advantage and rearranges o the network topology
  - Minimal cardinality [Biteus 06]
    - Subset of minimal diagnosis

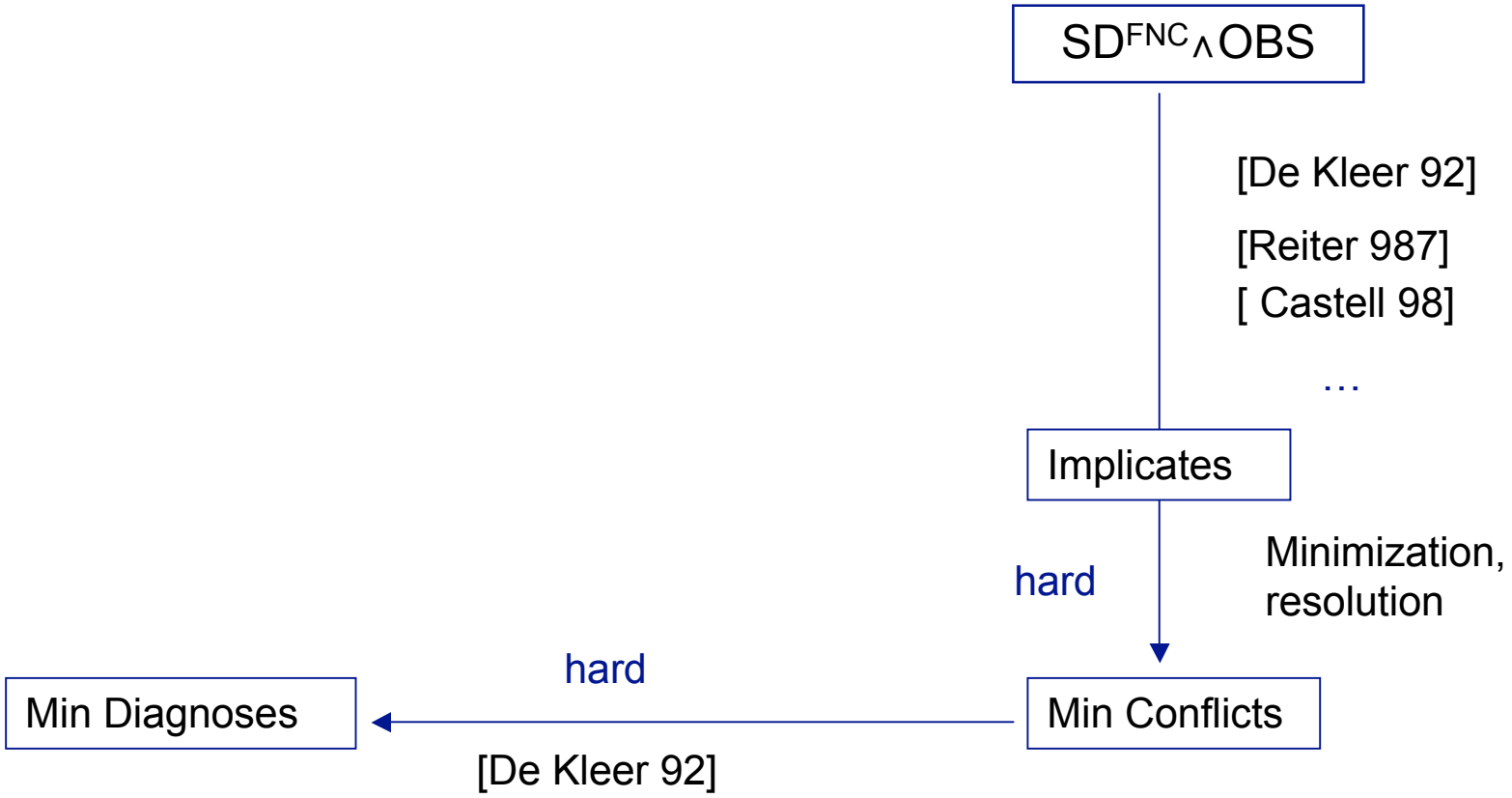
- Decentralized Diagnosis
  - Scalable Jointree Algorithm for diagnosability [Shumann, Huang 08]
  - Local Consistency and Junction Tree for Diagnosis of DES [Pr Kan John, A Grastien 08],
  - A Framework for Decentralized Qualitative Model-Based Diagnosis[Lucas Console 07]
  - A. Beneviste, E. Fabre, et al 01],…
    - Suppose a supervisor, no private knowledge

- DCSP, DCOP
  - Asynchronous weak commitment, …
    [Makoto Yokoo, Edmund H. D et al, 98], …
    - Look for conflicts first

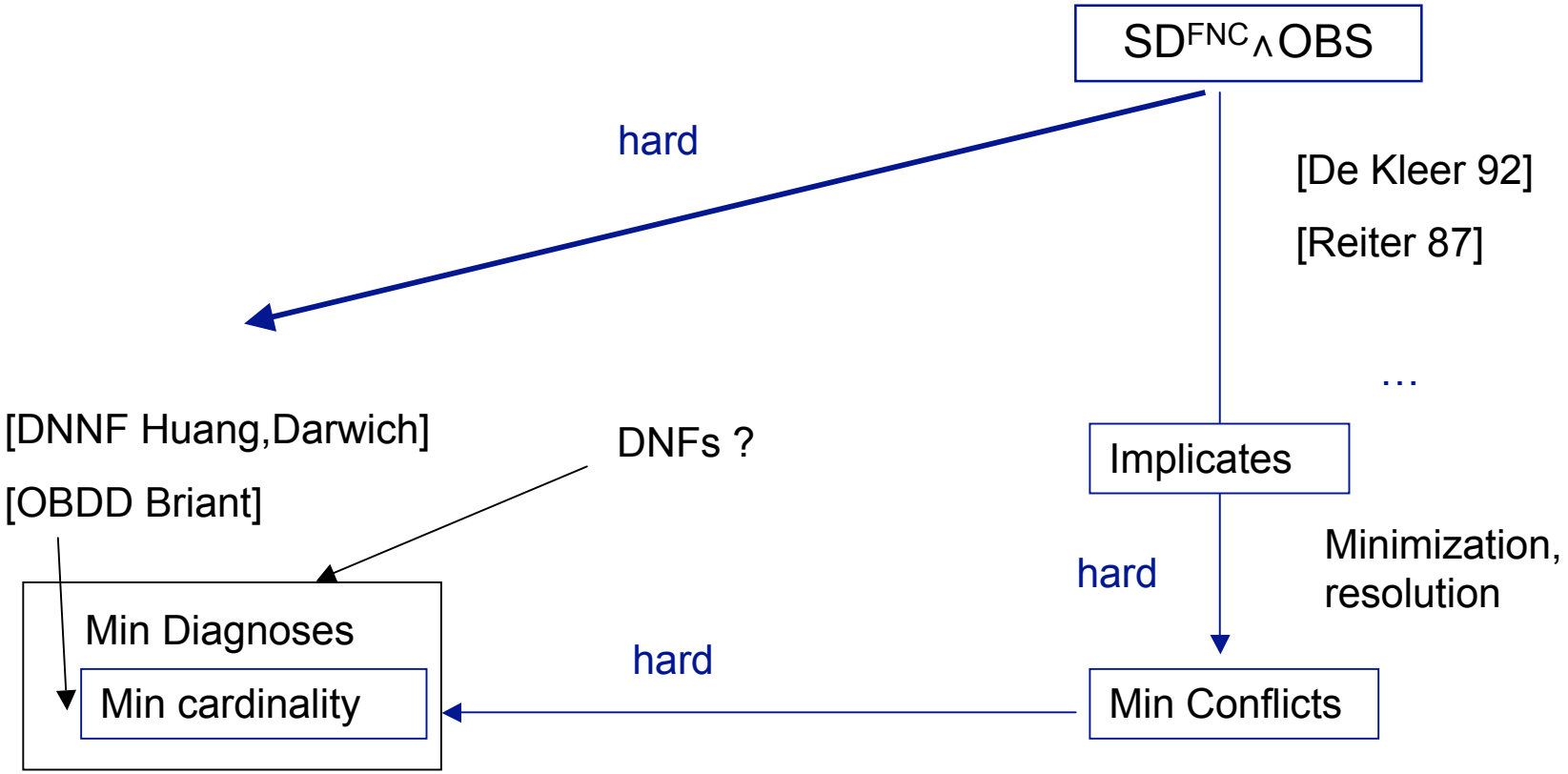
Road Map

- Consistency based diagnosis
- System description compilation
- Dealing with privacy and useless knowledge
- Our distributed, incremental, algorithm
- Conclusion and Perspectives

Diagnosis by conflicts



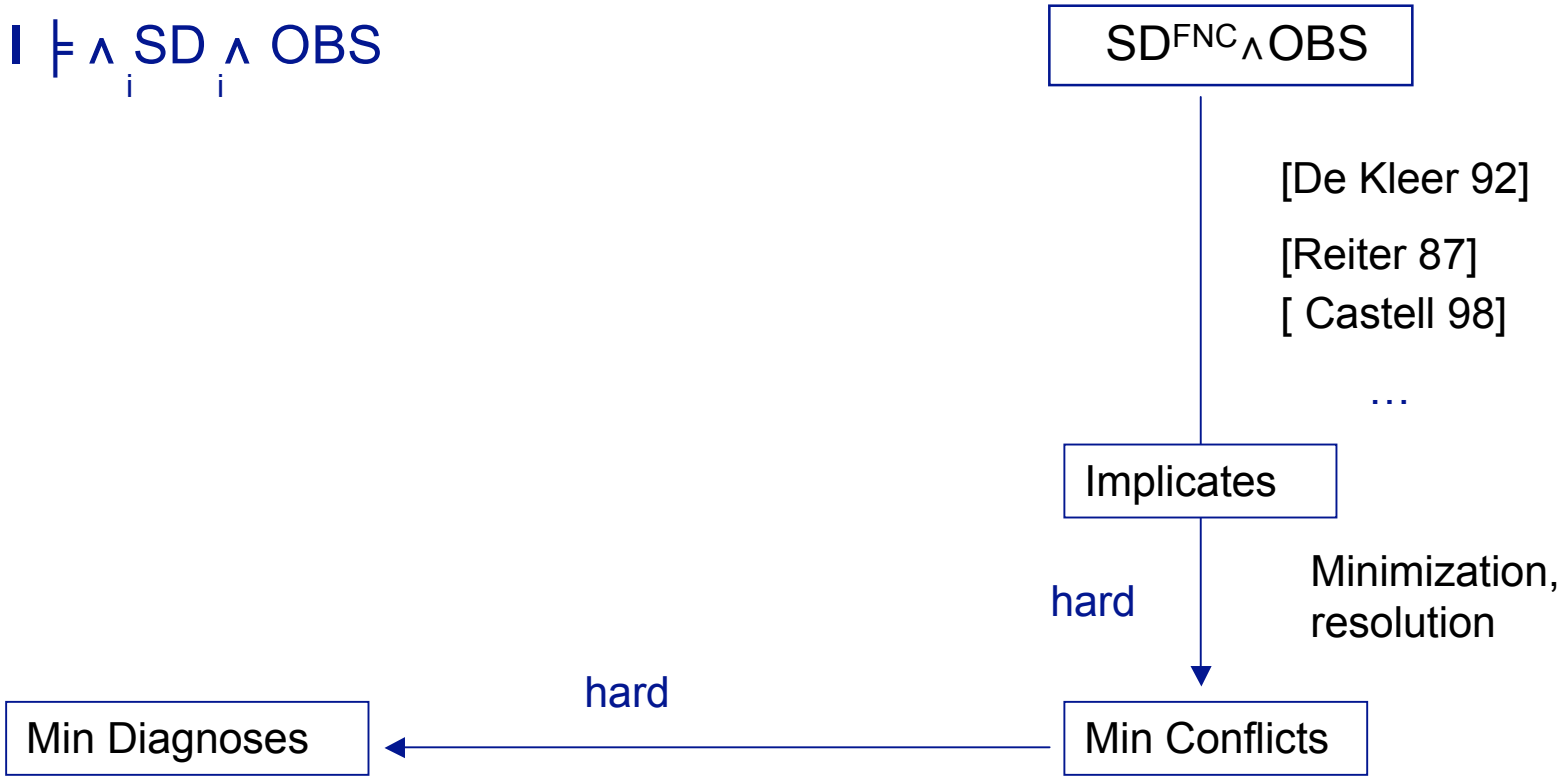
Compiling system description



Online diagnosis

Let I be an implicant of SD ∧ OBS

$I \models \bigwedge_i SD_i \wedge OBS$

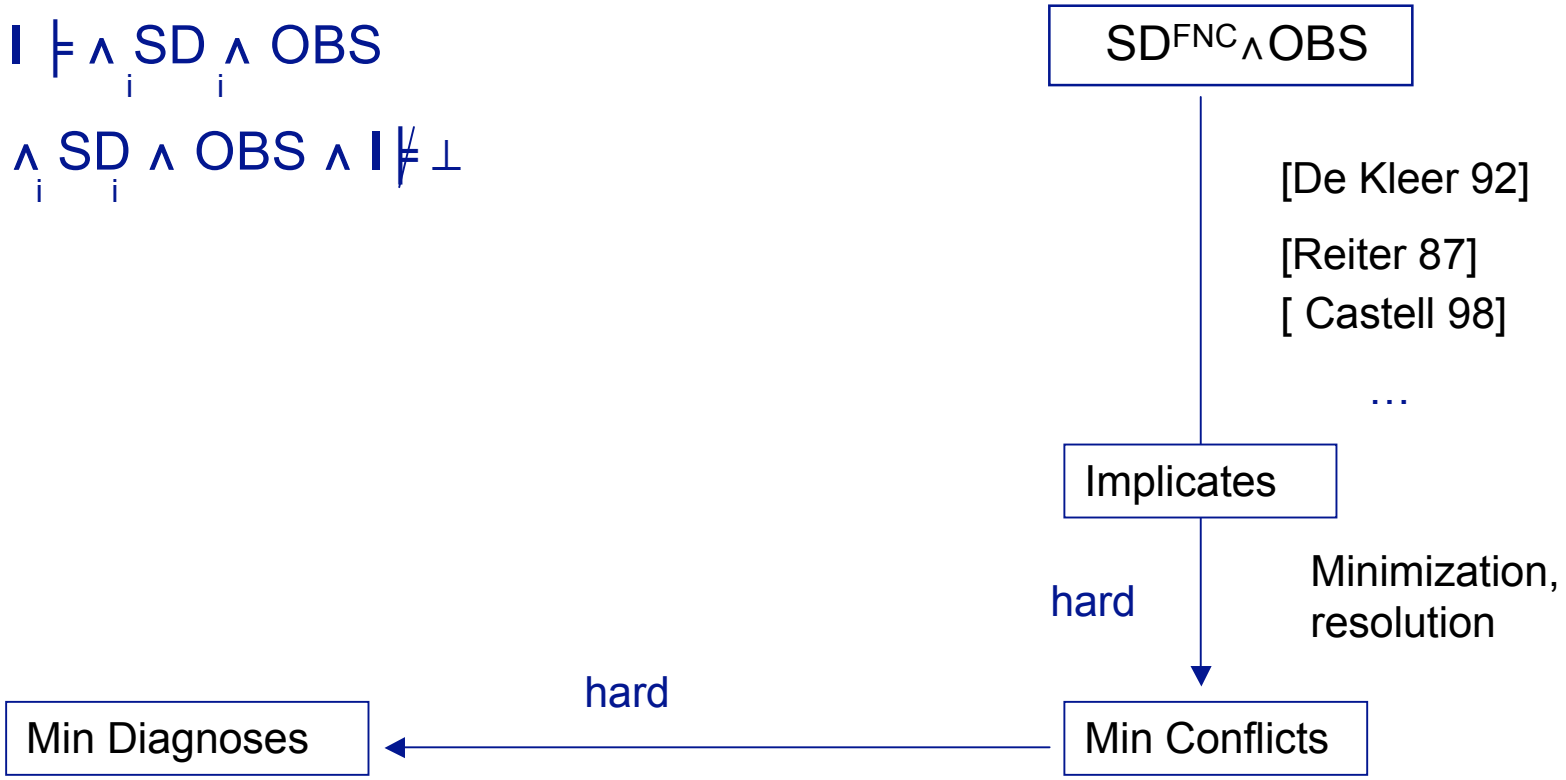


Online diagnosis

Let I be an implicant of SD ∧ OBS

$$I \models \bigwedge_i SD_i \wedge OBS$$

$$\bigwedge_i SD_i \wedge OBS \wedge I \not\models \bot$$



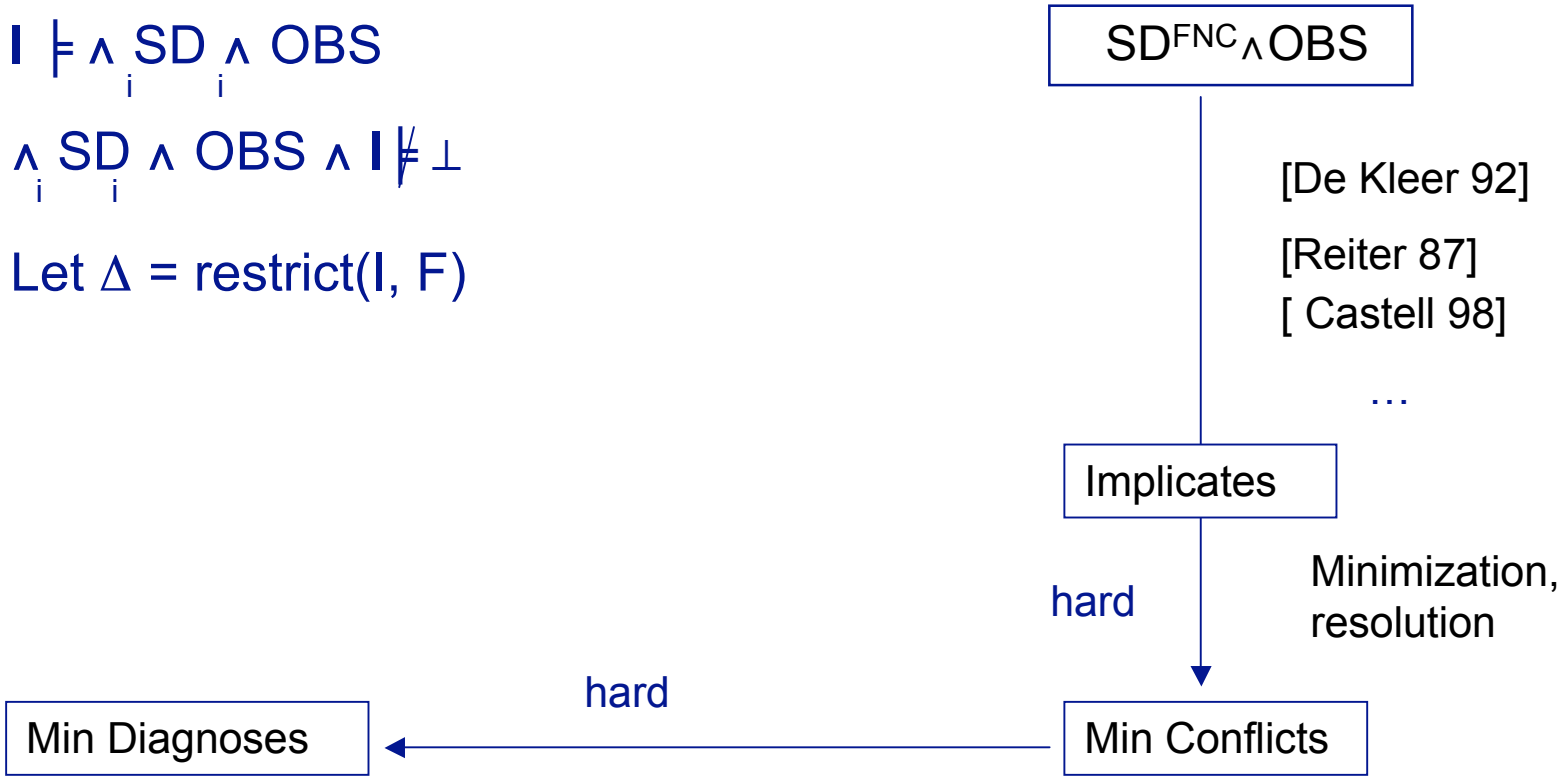
Online diagnosis

Let I be an implicant of SD ∧ OBS

$I \models \bigwedge_i SD_i \wedge OBS$

$\bigwedge_i SD_i \wedge OBS \wedge I \not\models \bot$

Let Δ = restrict(I, F)



Online diagnosis

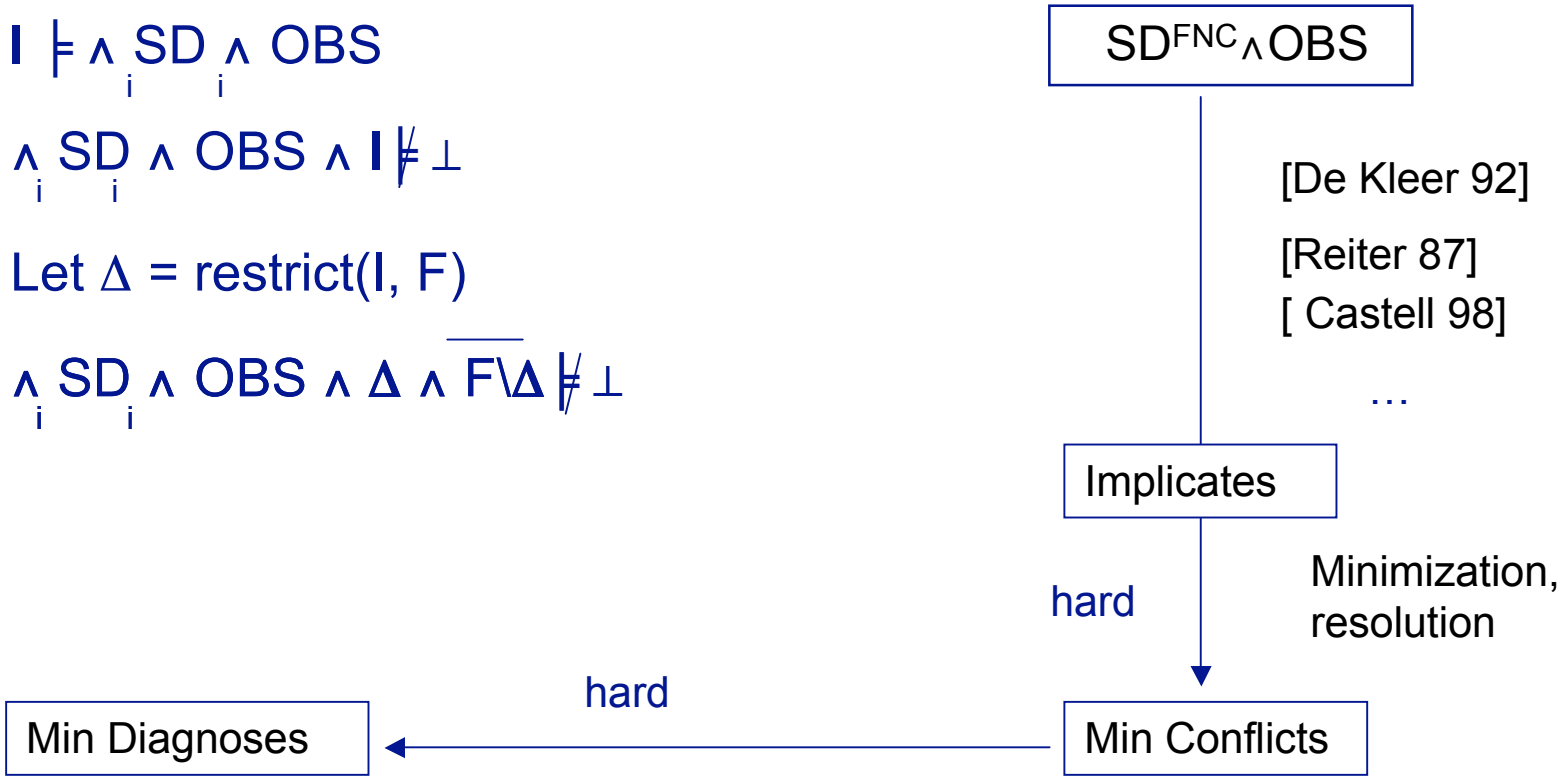
Let I be an implicant of SD ∧ OBS

$I \models \bigwedge_i SD_i \wedge OBS$

$\bigwedge_i SD_i \wedge OBS \wedge I \not\models \bot$

Let Δ = restrict(I, F)

$\bigwedge_i SD_i \wedge OBS \wedge \Delta \wedge \overline{F \setminus \Delta} \not\models \bot$



Online diagnosis

Let I be an implicant of SD ∧ OBS

$I \models \bigwedge_i SD_i \wedge OBS$

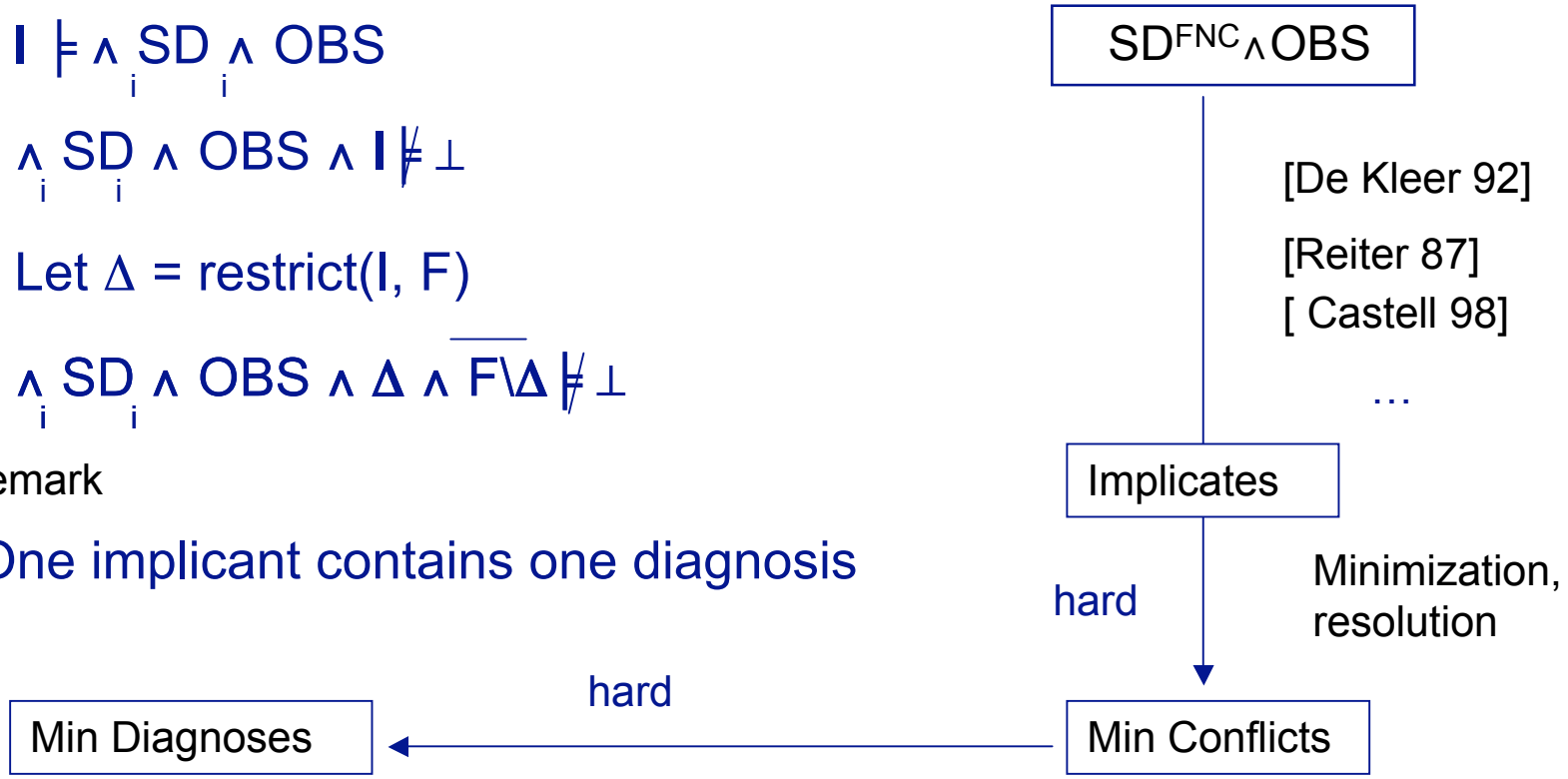
$\bigwedge_i SD_i \wedge OBS \wedge I \not\models \bot$

Let Δ = restrict(I, F)

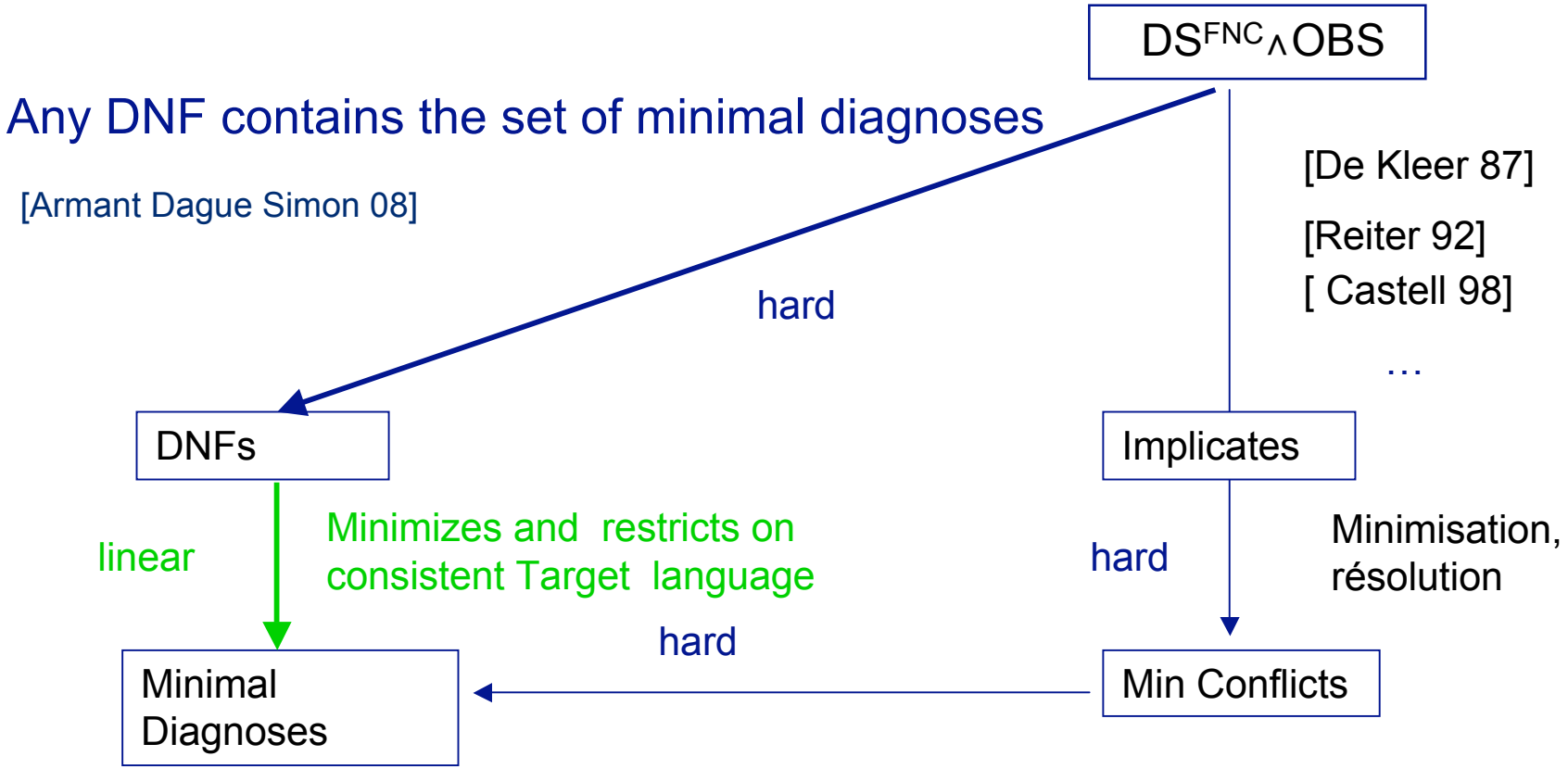
$\bigwedge_i SD_i \wedge OBS \wedge \Delta \wedge \overline{F \setminus \Delta} \not\models \bot$

Remark

One implicant contains one diagnosis



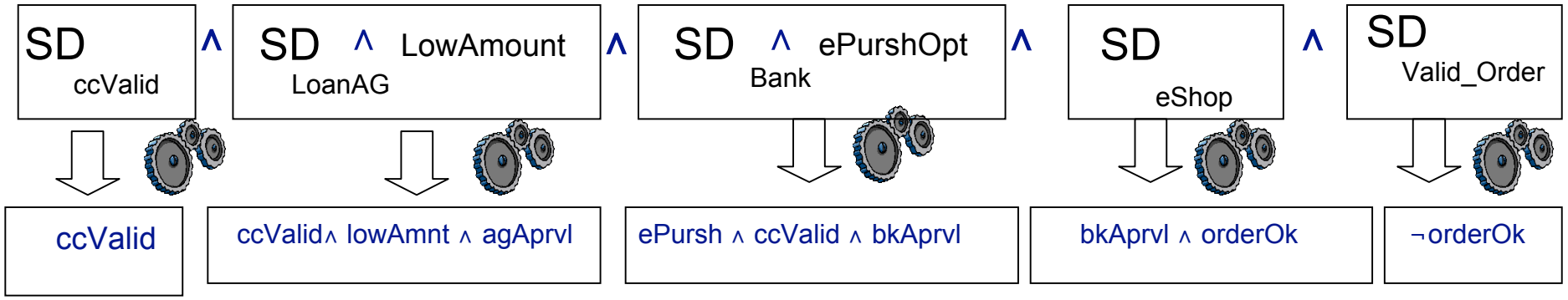
Online diagnosis



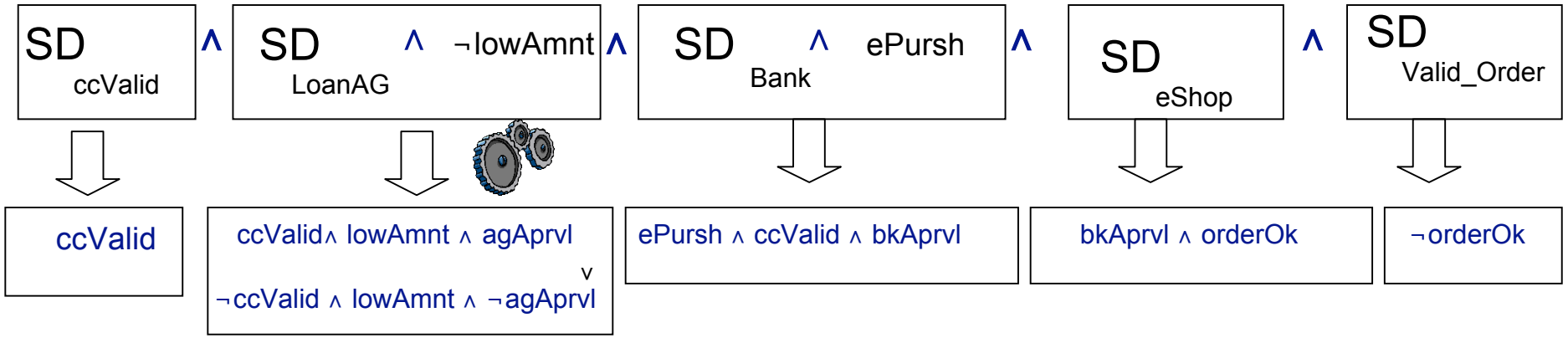
Our problem reformulated

- Hypothesis :
  - We can rewrite peer's observed description on the fly or offline
  - We can consider the global theory as a conjunction of local DNF
- Problem
  - Find all minimal sets on a target language witch are consistent with the global observed system

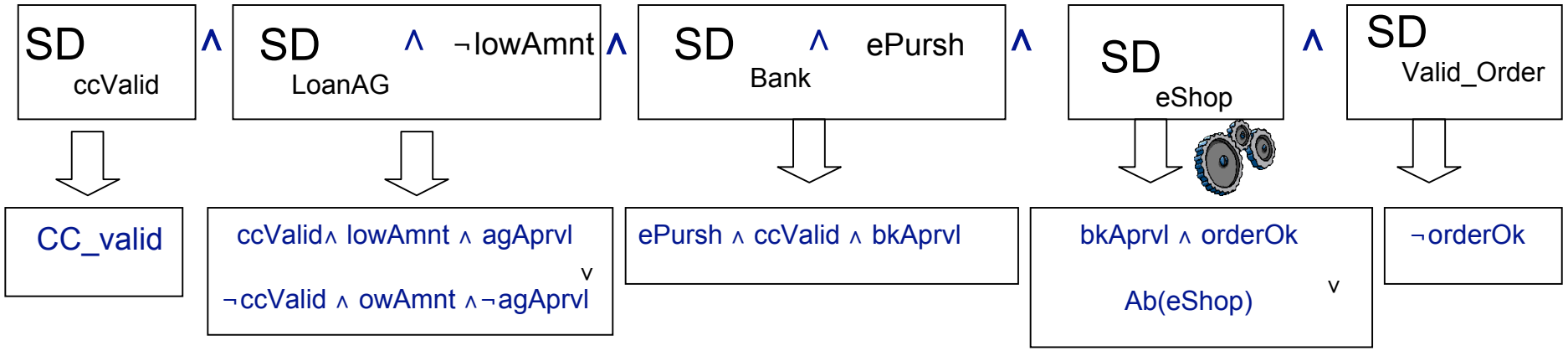
Using the power of the distributed setting



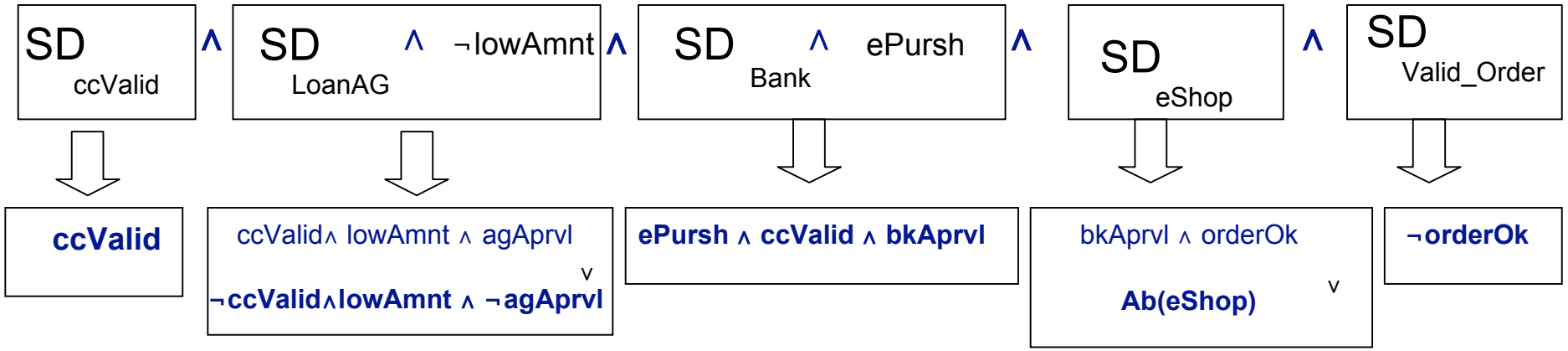
Using the power of the distributed setting



Using the power of the distributed setting



Using the power of the distributed setting



implicant

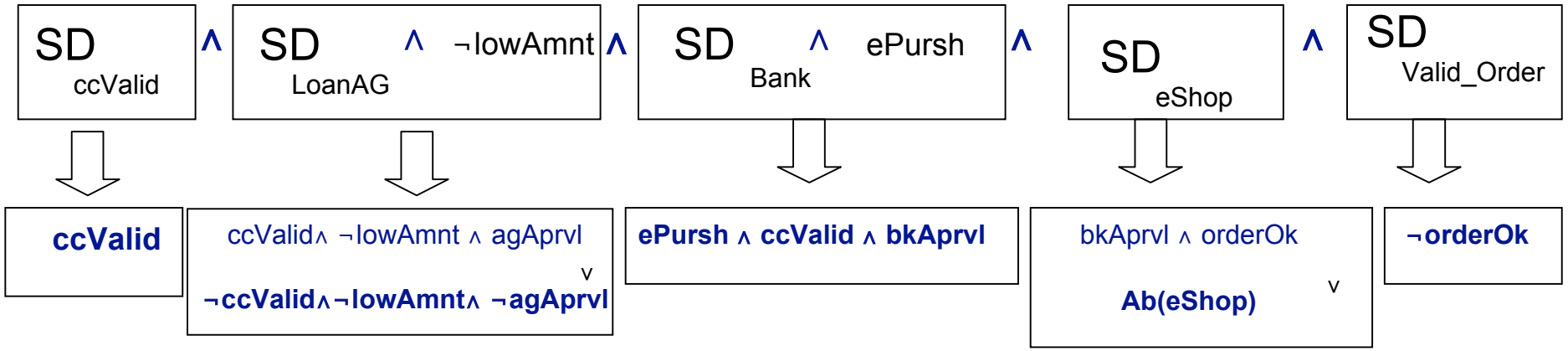
ccValid ∧ lowAmnt ∧ pretAg eOpt ∧ CBvadid ∧ pretBq ∧ Ab(eShop) ∧ ¬orderOk

diagnosis

Projection on mode variables

Ab(eShop)

Using the power of the distributed setting



implicant

ccValid ∧ lowAmnt ∧ agAprvlg eOpt ∧ CBvadid ∧ pretBq ∧ Ab(eShop) ∧ ¬orderOk

diagnosis

Projection mode variables

Ab(eShop)

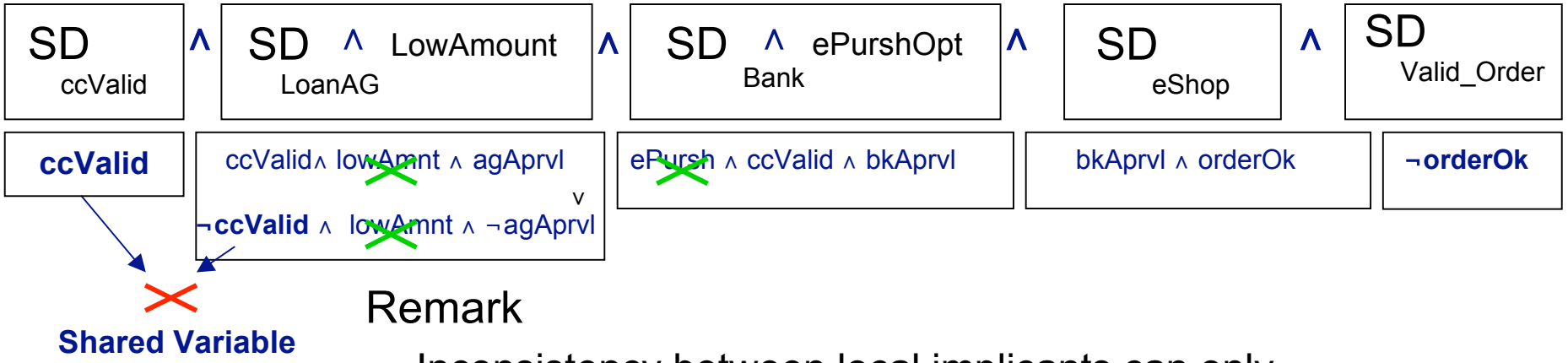
But this first approach does not respect the privacy constraint

Road Map

- Consistency based diagnosis
- System description compilation
- Dealing with privacy and useless knowledge
- Our distributed, incremental, algorithm
- Conclusion and Perspectives

Keeping local knowledge local

Private knowledge : peer's system description and local variables



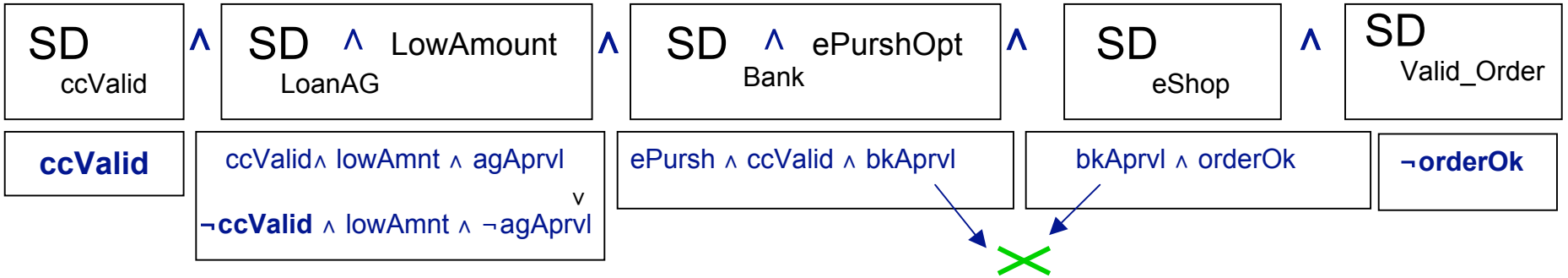
Remark

Inconsistency between local implicants can only come from shared variable

First Restrict peer's SD on shared and mode variables then distribute is equivalent to restrict the global DNF representing the whole system

[Armant Dague Simon 08]

Removing useless variables



Shared Variable

Remark

A Shared variable can be considered as a local variable of the set peers in which it appears

Removing Shared variables only appearing in Sdi, SDj , after their distribution does not affect the global consistency

[Armant Dague Simon 08]

Road Map

- Consistency based diagnosis
- System description compilation
- Dealing with privacy and useless knowledge
- Our distributed, incremental, algorithm
- Conclusion and Perspectives

Top-down bottom-up algorithm

Top down phase : construction of a distributed tree

- Initially a given peer broadcasts a request of diagnosis to its neighborhood
- When a peer has received its 1st request of diagnosis
  - It chooses the sender as parent
  - Starts computation of its local R-implicants
- When a peer receives at least 1 msg from it neighborhood

It sends its restricted implicants to its parent

```
3:  case : reqDiag
4:    /*A distributed tree is built*/
5:    if parent is not set then /* Flooding alg.*/
6:       parent ← p'
7:       send to all p neighborhood \p' : msg [reqDiag]
8:    else /* p' is not a direct child */
9:       NotChild ← NotChild ∪ {p'}
10:   end if
11:   /* Flushes all stored implicants when the subtree is known */
12:   if {parent} ∪ Child ∪ NotChild = Neighborhood
13:      Π ← flush(T_p^∨, TChild, Desc)
14:      for all I ∈ Π
15:         send to parent msg [respDiag,I,Desc ∪ {p}]
16:      end for
17:   end if
18:   /* p' is either the parent or not a direct child*/
19:   checkEnd(waitEnd, p')
```

Top-down bottom-up algorithm

Bottom-up phase : compose diagnoses of sub tree

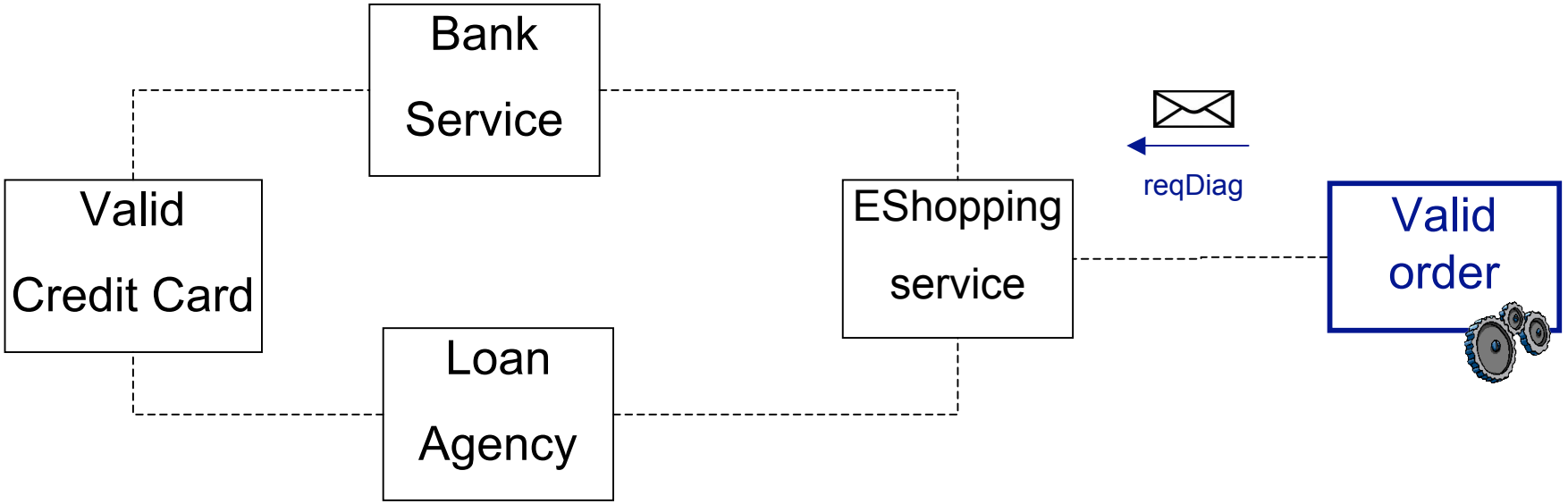
- when a peer receives a Rimplicant
  - it composes R implicants from its sub trees

If it receives at least 1 msg from its neighborhood

  - it sends to its father R-Implicants built from its restricted on useful vocabulary

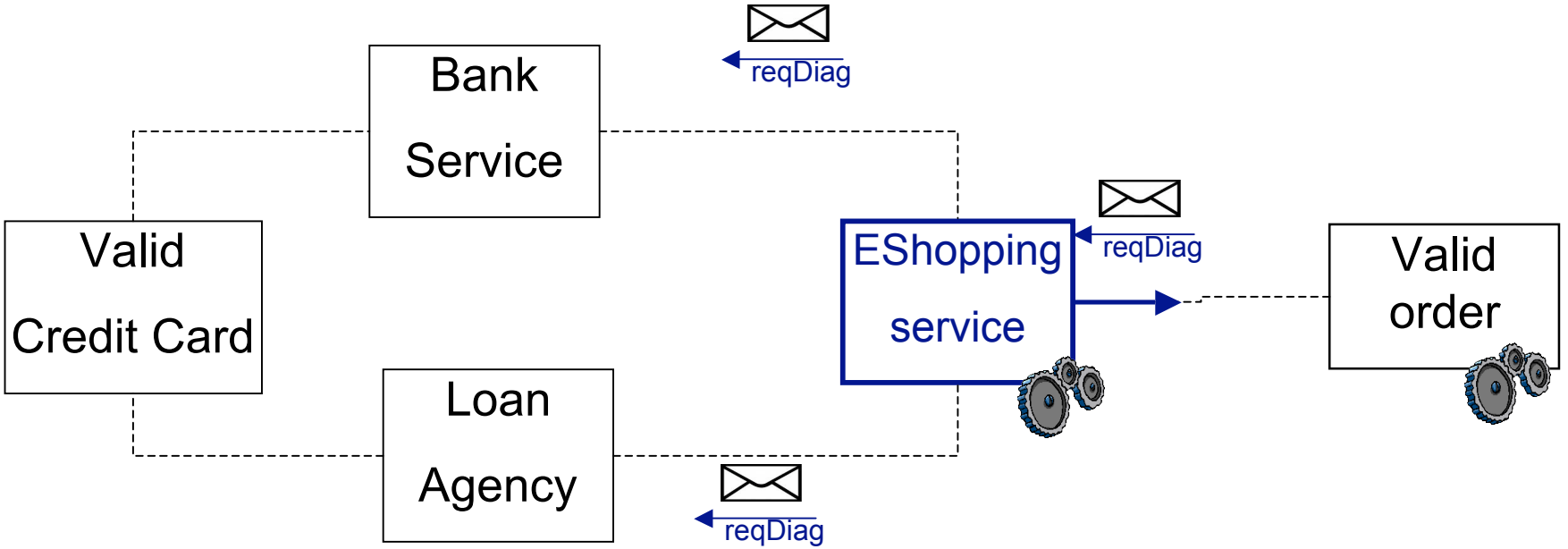
```
21: case : respDiag
22:    /* Stores the diag, or extends and propagates it */
23:    Child ← Child ∪ {p'}
24:    Desc ← Desc ∪ msg.Desc
25:    TChild[p'] ← TChild[p'] ∪ msg.rImpl
26:    /* Extends msg.rImpl only if the subtree is already known */
27:    if {parent} ∪ Child ∪ NotChild = Neighborhood
28:       Π ← extends(msg.rImpl, T_p^∨, TChild, Desc)
29:       for all I ∈ Π
30:          send to parent msg [respDiag,I,Desc ∪ {p}]
31:       end for
32:       Tresult ← min_⊆(Tresult ∪ Π)
33:    end if
```

M2DT: Minimal Diagnoses by Distributed Tree



- The peer "Valid Order" starts the diagnosis task
  - it sends a request of diagnosis
  - it begins the computation of its own implicants

M2DT: Minimal Diagnoses by Distributed Tree

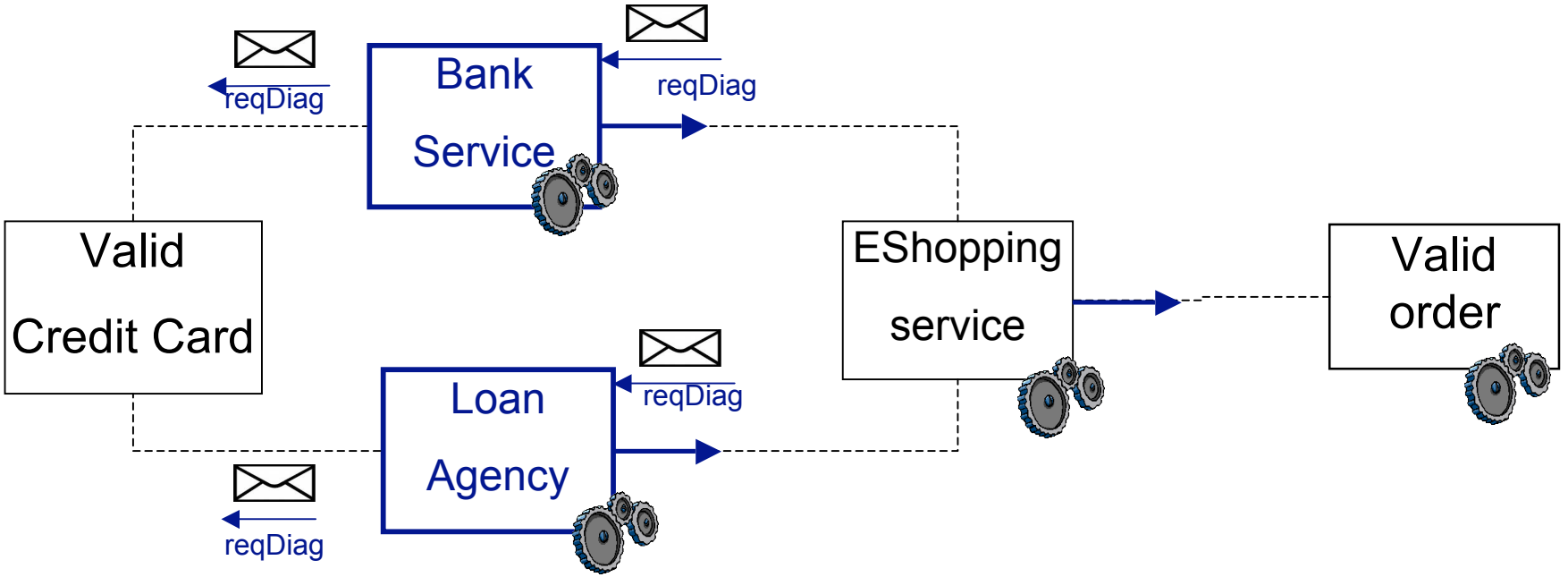


- When “eShopping service” has received the request of diagnosis
  - it chooses “Order Validation” as parent
  - it starts the computation of its local implicants :

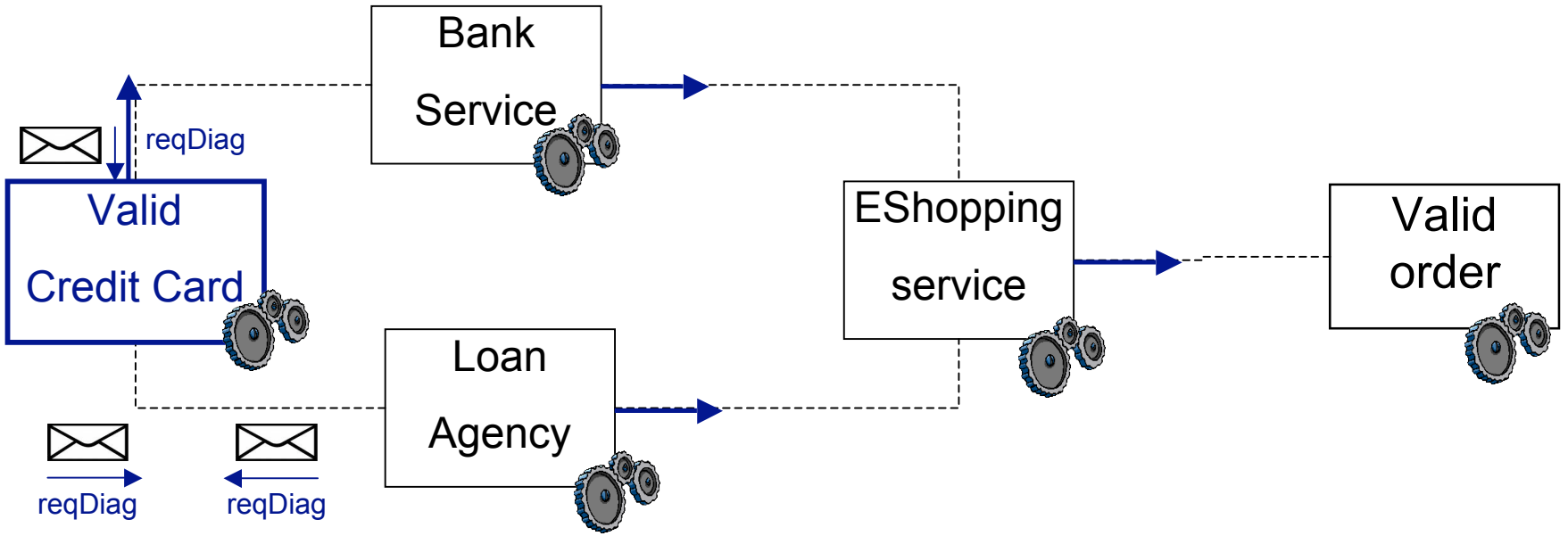
$$Bk\_Approval \wedge hire\_Purch$$

  - it Forwards the request to Bank and to the Loan Agency

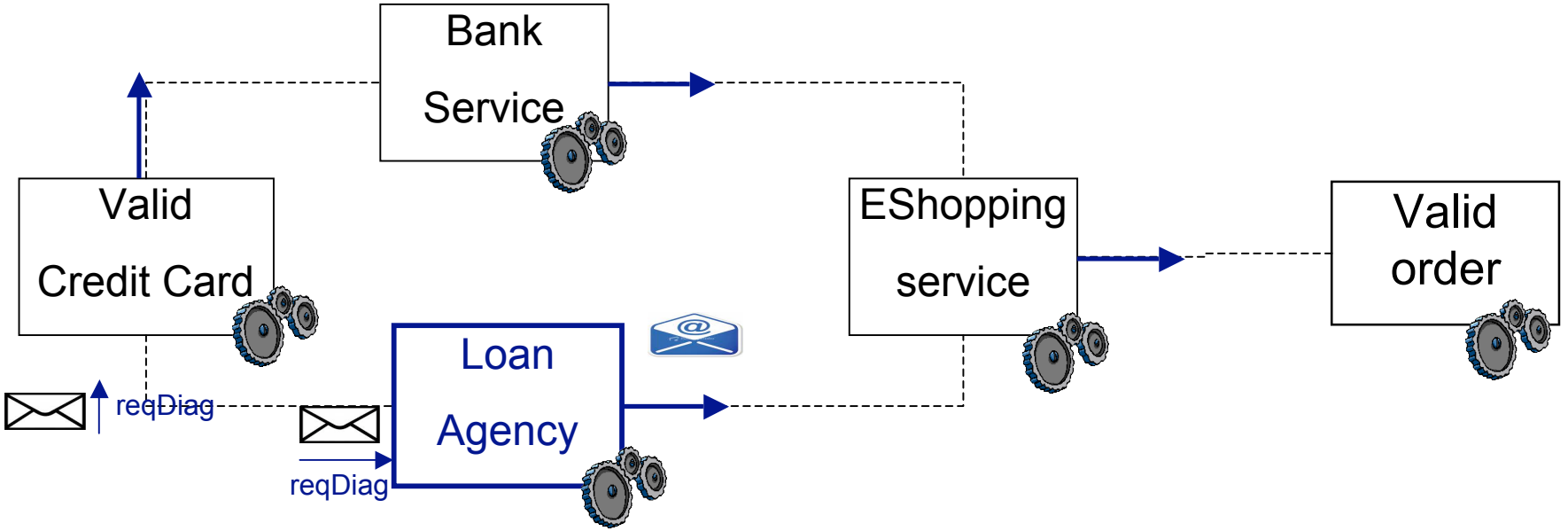
M2DT: Minimal Diagnoses by Distributed Tree



M2DT: Minimal Diagnoses by Distributed Tree

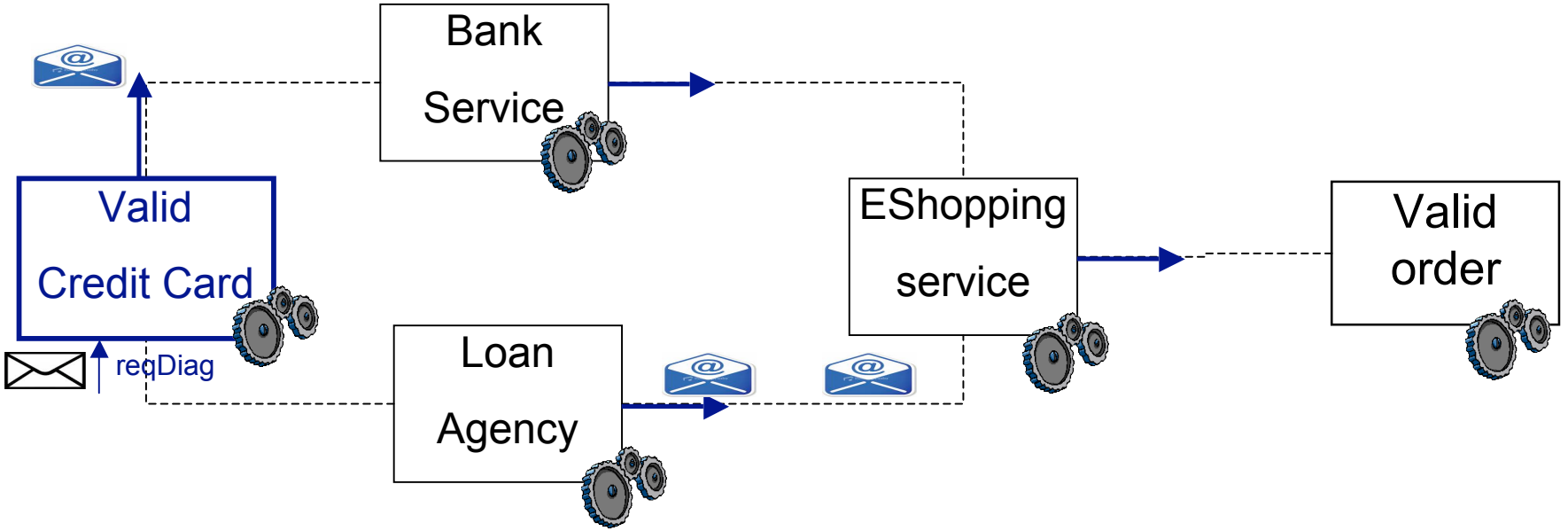


M2DT: Minimal Diagnoses by Distributed Tree



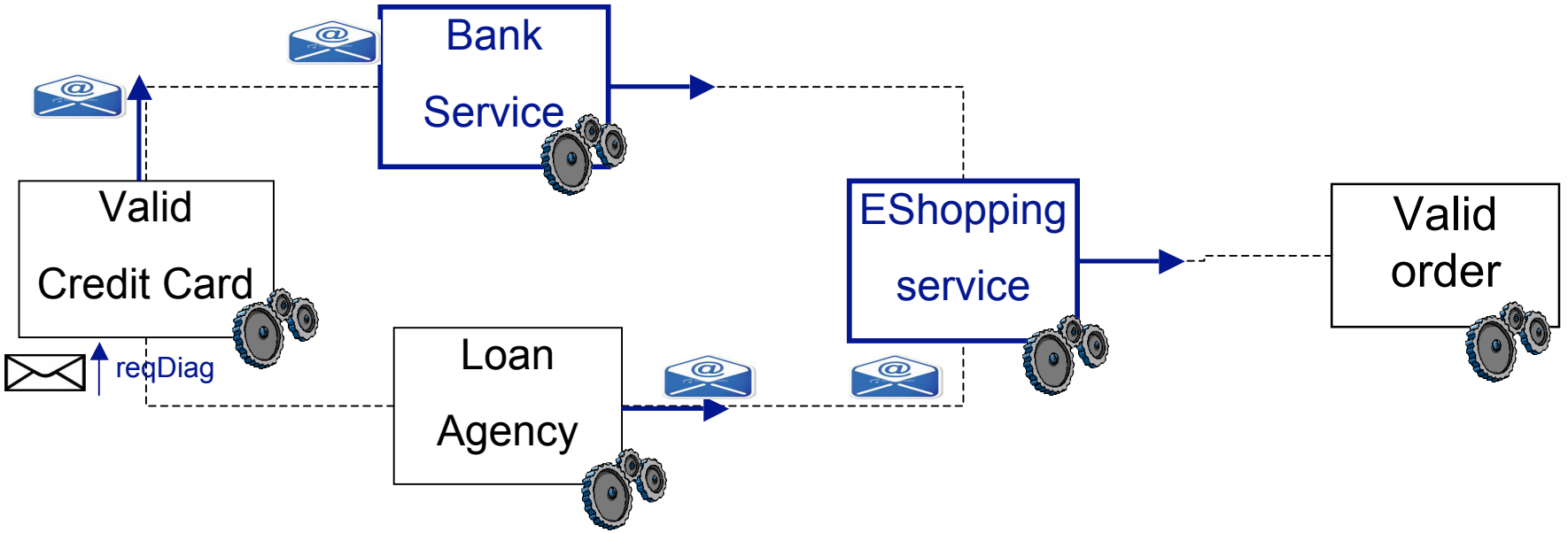
- When Loan Agency has received a msg from all its neighbours
  - it sends his restricted Implicants agAprvl to its parent

M2DT: Minimal Diagnoses by Distributed Tree

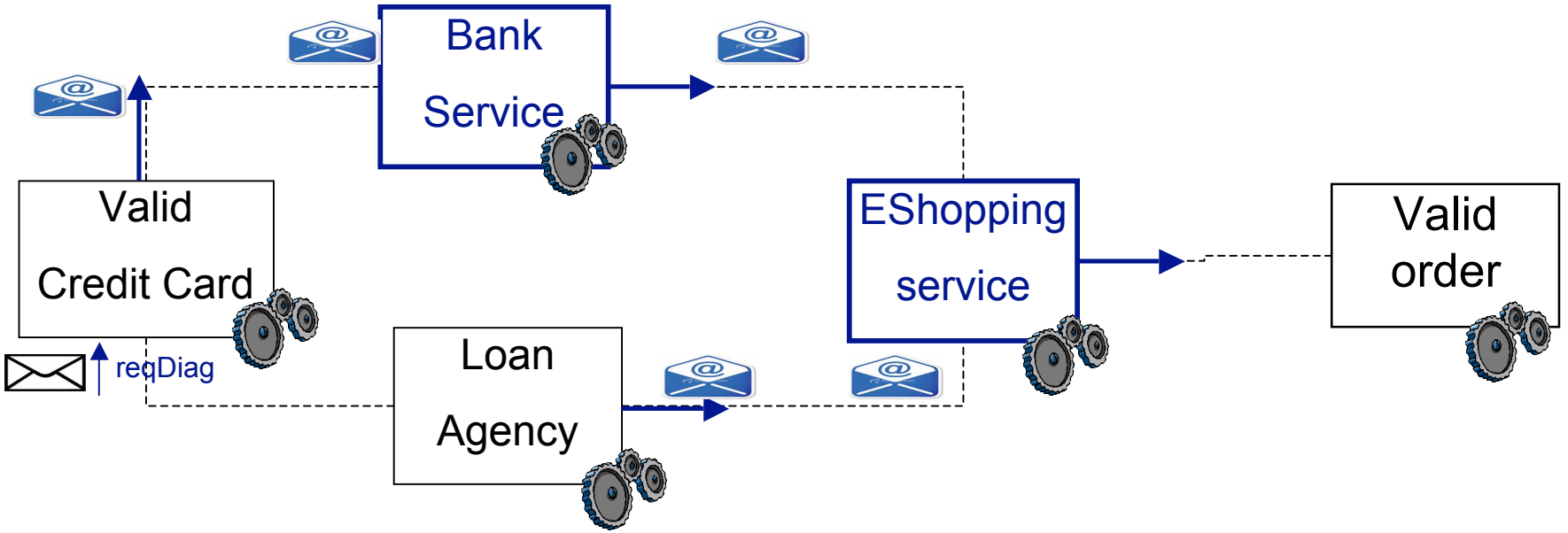


- Valid Credit Card has received a msg from all its neighbours
  - it sends his restricted Implicants ccValid to its parent
- Loan Agency continuously sends new restricted implicants to its parents

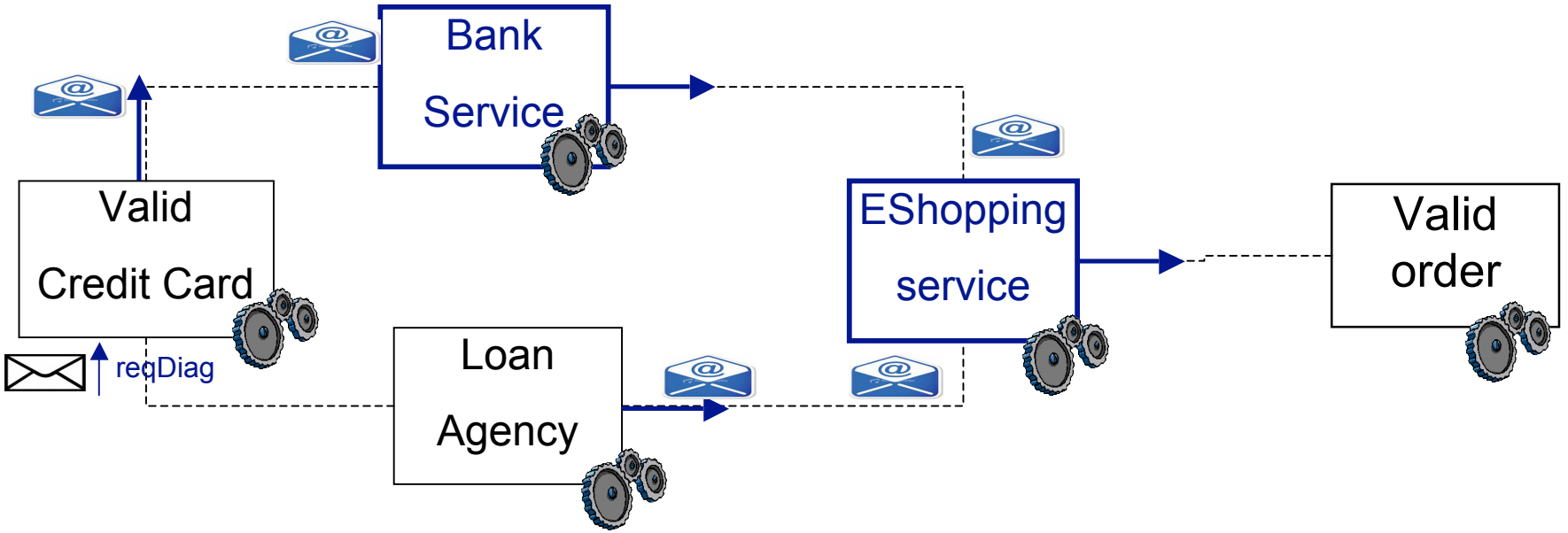
M2DT: Minimal Diagnoses by Distributed Tree



M2DT: Minimal Diagnoses by Distributed Tree



M2DT: Minimal Diagnoses by Distributed Tree



☐ When The peer eShop has received from all its neighbors,

- It can build the consistent conjunction : Bk_Apprvl ∧ ccValid ∧ Ab(eShop) ∧ hPurch
- It removes useless variables by restricting the new conjunction on shared and mode variable
- It sends the result Ab(eShop) ∧ hPurch to Order Validation Service

☐ Order Validation Service gets its first diagnosis Ab(eShop)

Road Map

- Consistency based diagnosis
- System description compilation
- Dealing with privacy and useless knowledge
- Our distributed, incremental, algorithm
- Conclusion and Perspectives

Conclusion

- Incremental computation of global diagnoses
- At the termination the peer starter is informed of all minimal diagnoses
- Local descriptions of peers are never communicate
- The entire DNF of the systems is never built
- Experimentation
  - (is going on)

Perspectives

- Privacy
  - Communicate neither the local description of peer nor the healthy variables
- Dynamic system
  - Arrival and departure of peers
  - Asynchronous arrival of observations