Reading Agendas between the lines *an exercise*

30 August 2016 - AI4J Workshop / ECAI 2016 @ Den Haag

Giovanni Sileno (g.sileno@uva.nl), Alexander Boer, Tom van Engers

Leibniz Center for Law

University of Amsterdam

Problem context

How people behave..

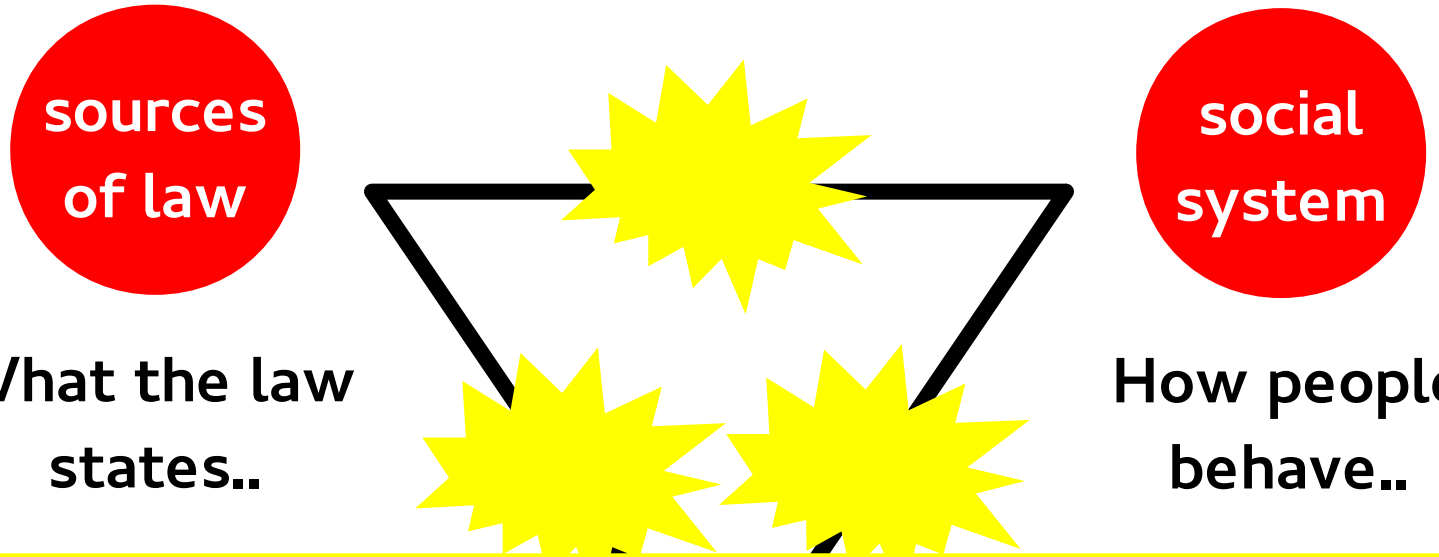
What the law states..

How people behave..

What the law states..

How people behave..

How public administrations implements the law..

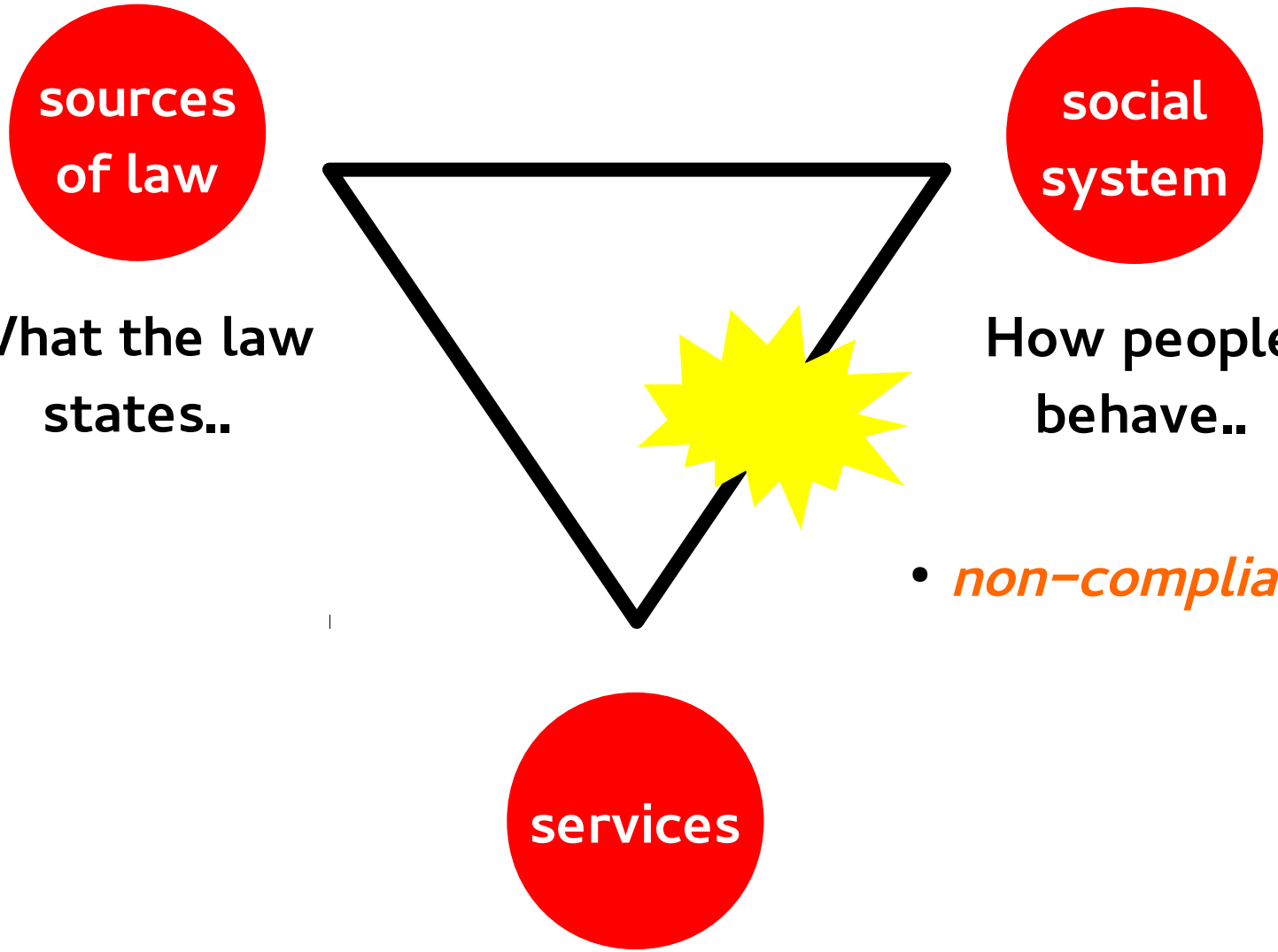


are three matters only loosely coupled

services

How public administrations implements the law..

Focus on services/social system



Focus on services/social system

sources of law

What the law states..

social system

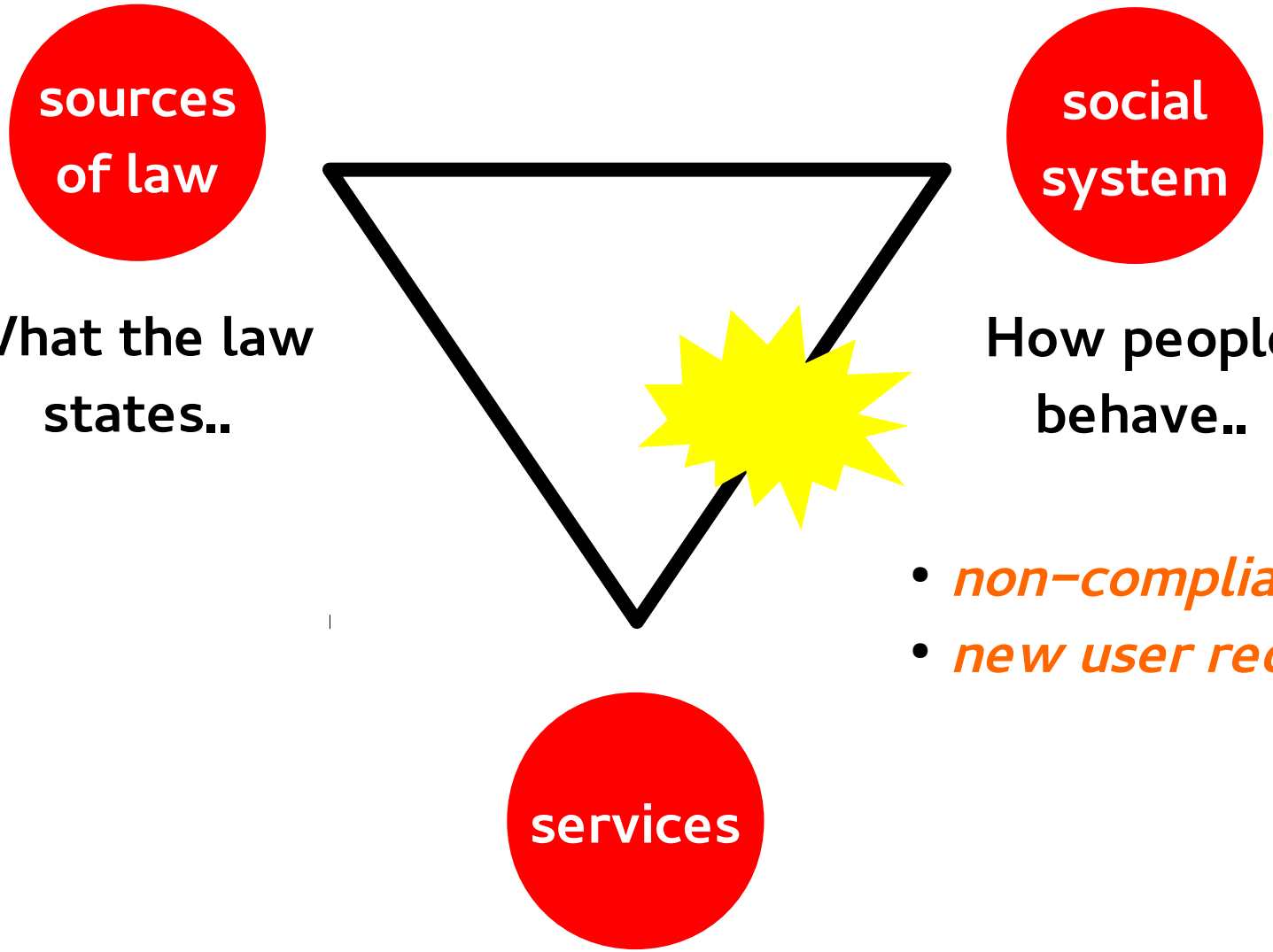
How people behave..

- *non-compliance*

services

How public administrations implements the law..

Focus on services/social system



- *non-compliance*
- *new user requirements*

Diagnosis of social systems

Diagnosis

- Diagnosis starts from the presumption that a *failure* occurred in the system.
- But what counts as a **failure**?

Diagnosis of a designed artifact

- In case of a ***designed artifact***, we know the function of the system, so if does not behave how it was supposed to, this is a failure.

Diagnosis of a designed artifact

- In case of a ***designed artifact***, we know the function of the system, so if does not behave how it was supposed to, this is a failure.

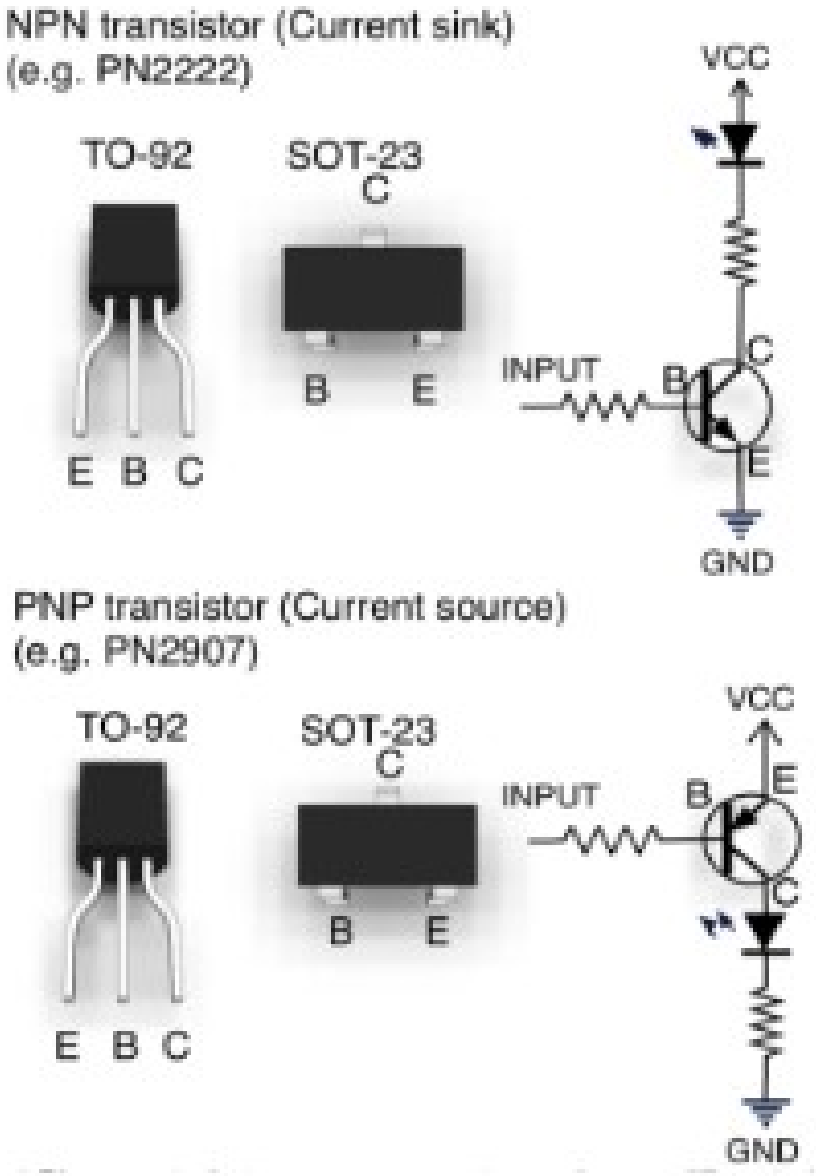
Diagnosis of a designed artifact

- In case of a ***designed artifact***, we know the function of the system, so if does not behave how it was supposed to, this is a failure.

- Two types of failure:
  - ***operational failure***
  - ***bad design***

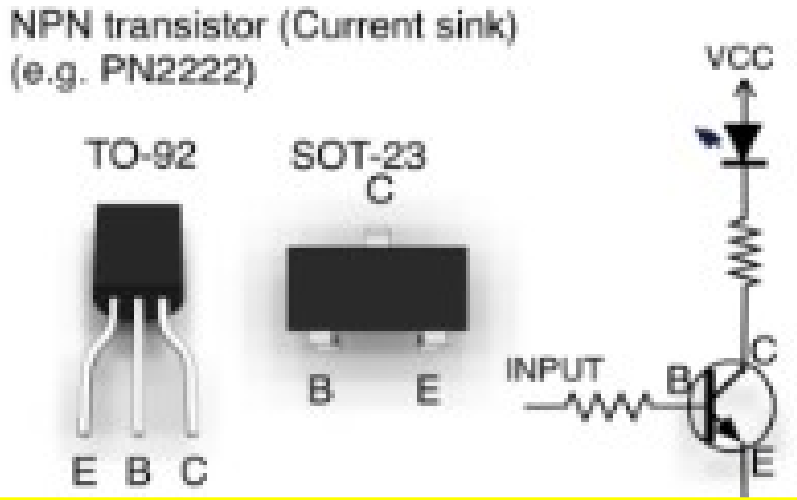
Consistency-based diagnosis

- In case of electronic components, for instance, manufacturers furnish the normal behaviour of the item.

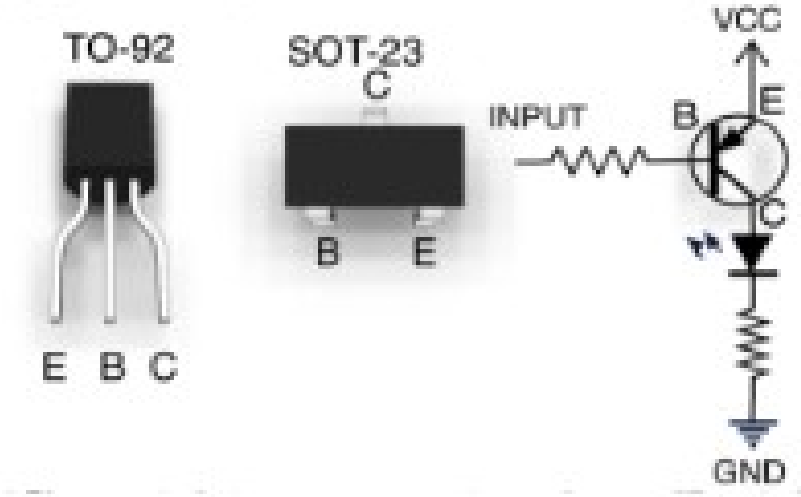


Consistency-based diagnosis

- In case of electronic components, for instance, manufacturers furnish the normal behaviour of the item.

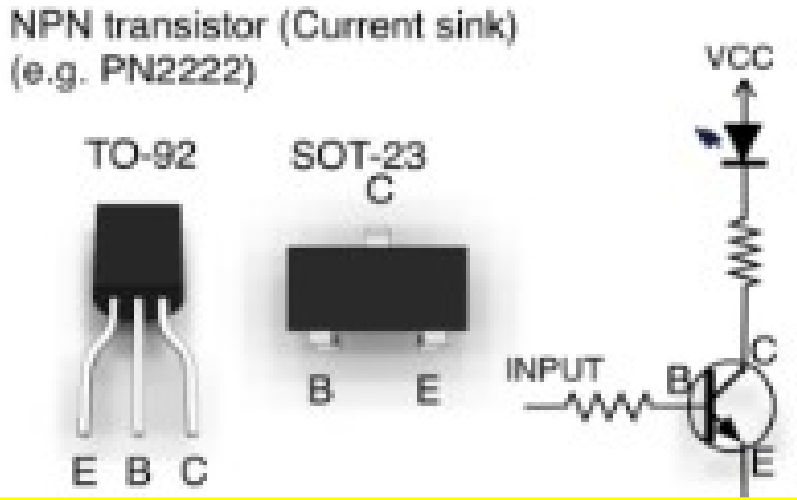


failure = *inconsistency* with nominal specifications



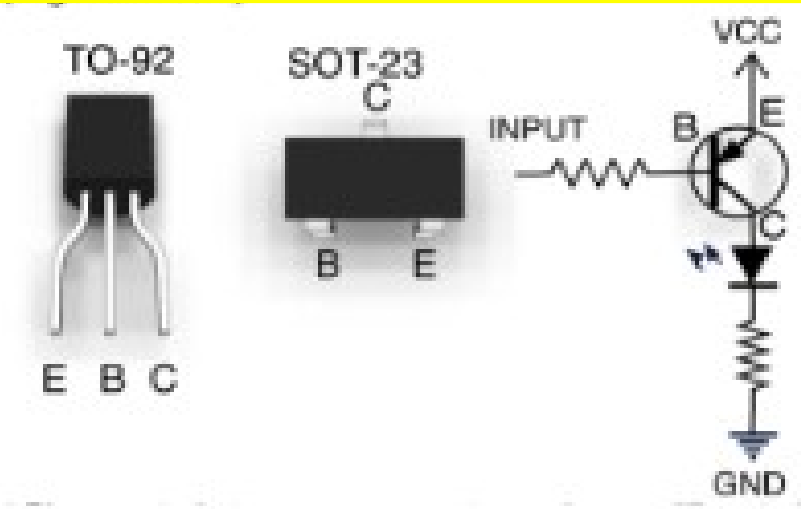
Consistency-based diagnosis

- In case of electronic components, for instance, manufacturers furnish the normal behaviour of the item.



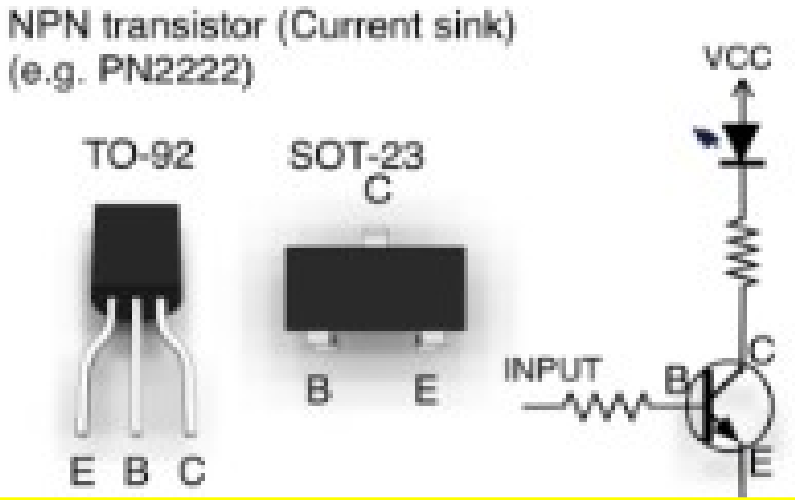
failure = *inconsistency* with nominal specifications

- *Usual diagnostic problem*: recognize the **minimal** set of components that produces the inconsistency



Consistency-based diagnosis

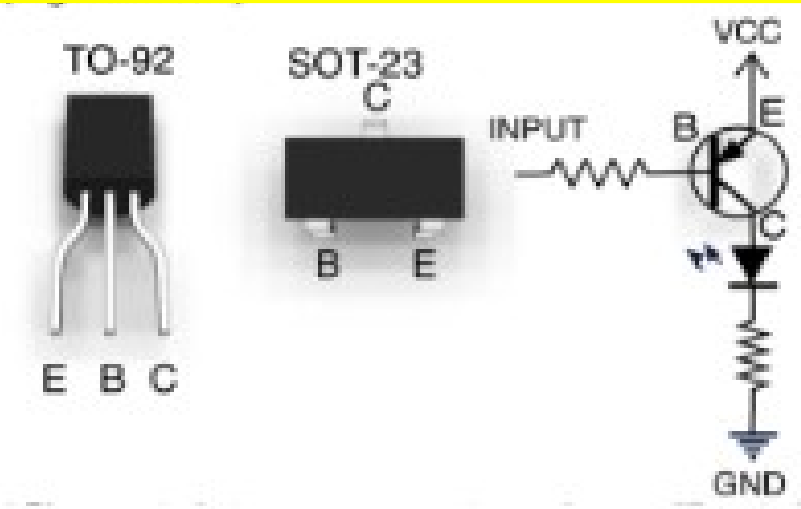
- In case of electronic components, for instance, manufacturers furnish the normal behaviour of the item.



failure = *inconsistency* with nominal specifications

In AI terms ~ default negation implies strong negation

```
-a :- not a.
```



Abductive diagnosis

- In other domains, we naturally create models of faulty behaviour, because there may be ***non-pathological*** cases in which things do not go as expected. i.e. **not all misalignments to the norm are failures.**

Abductive diagnosis

- In other domains, we naturally create models of faulty behaviour, because there may be ***non-pathological*** cases in which things do not go as expected. i.e. **not all misalignments to the norm are failures.**

failure = *consistency* with explicitly faulty model

In AI terms ~ default negation is different from strong negation

Diagnosis of a social system

- What counts as a failure in a *social system*?

Diagnosis of a social system

- What counts as a failure in a *social system*?
- A **wrong** supposedly occurs when there are *unmet social expectations* (for at least one of the participants)

Diagnosis of a social system

- What counts as a failure in a *social system*?
- A **wrong** supposedly occurs when there are *unmet social expectations* (for at least one of the participants)
- These are specified via **normative directives**: prohibitions, obligations, etc.
  - In principle, a wrong is detectable from the violation of normative directives.

Diagnosis of a social system

- What counts as a failure in a *social system*?
- A **wrong** supposedly occurs when there are *unmet social expectations* (for at least one of the participants)
- These are specified via **normative directives**: prohibitions, obligations, etc.
  - In principle, a wrong is detectable from the violation of normative directives..

    but proceeding "literally" is not sufficient.

Types of failure

e.g. *duty to pay taxes*

Types of failure

e.g. *duty to pay taxes*

- **not** to pay taxes (*syntaxic, qualitative failure*)

Types of failure

e.g. *duty to pay taxes*

- **not** to pay taxes (*syntaxic, qualitative failure*)
- pay a **wrong** amount in respect to the declared income (*syntaxic, quantitative failure*)

Types of failure

e.g. *duty to pay taxes*

- **not** to pay taxes (*syntaxic, qualitative failure*)
- pay a **wrong** amount in respect to the declared income (*syntaxic, quantitative failure*)
- pay *the correct but* **not the right** amount, by using tax evasion schemes (*semantic failure*)

Types of failure

e.g. *duty to pay taxes*

- **not** to pay taxes (*syntaxic, qualitative failure*)
- pay a **wrong** amount in respect to the declared income (*syntaxic, quantitative failure*)
- pay *the correct but* **not the right** amount, by using tax evasion schemes (*semantic failure*)

The "semantic" failure is the most difficult to be catched!

Sources of knowledge

sources of law

legal norms

Sources of knowledge

sources of law

legal norms

business process models

services

Sources of knowledge

sources of law

legal norms

Usual focus of *compliance-checking* methods

business process models

services

Sources of knowledge

sources of law

legal norms

..but not adequate to capture the social semantics

Usual focus of *compliance-checking* methods

business process models

services

Sources of knowledge

sources of law

legal norms, cases

stories, experiences

social system

business process models

services

Sources of knowledge

sources of law

legal norms, cases

stories, experiences

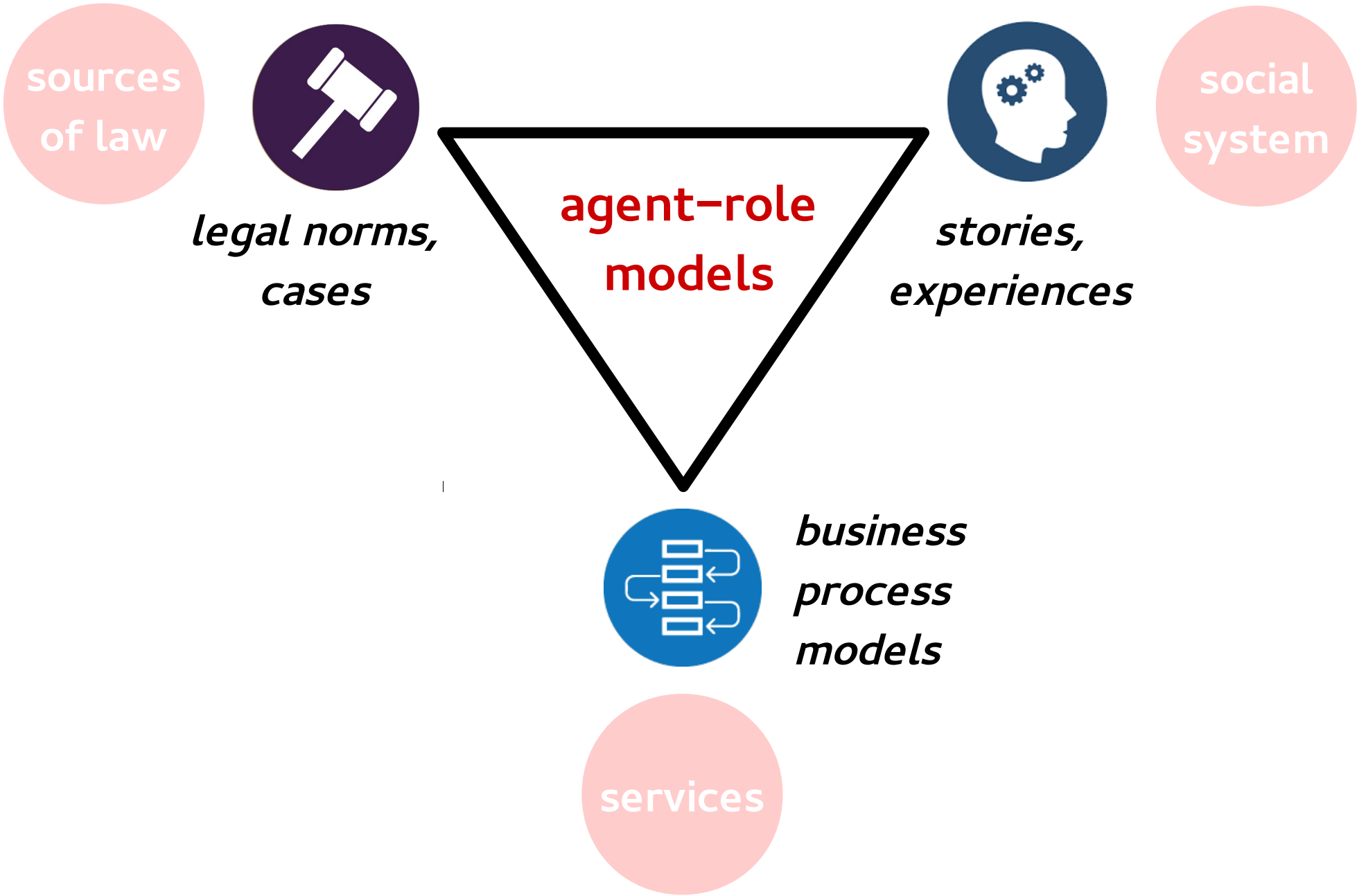
social system

Which representational ground should we consider?

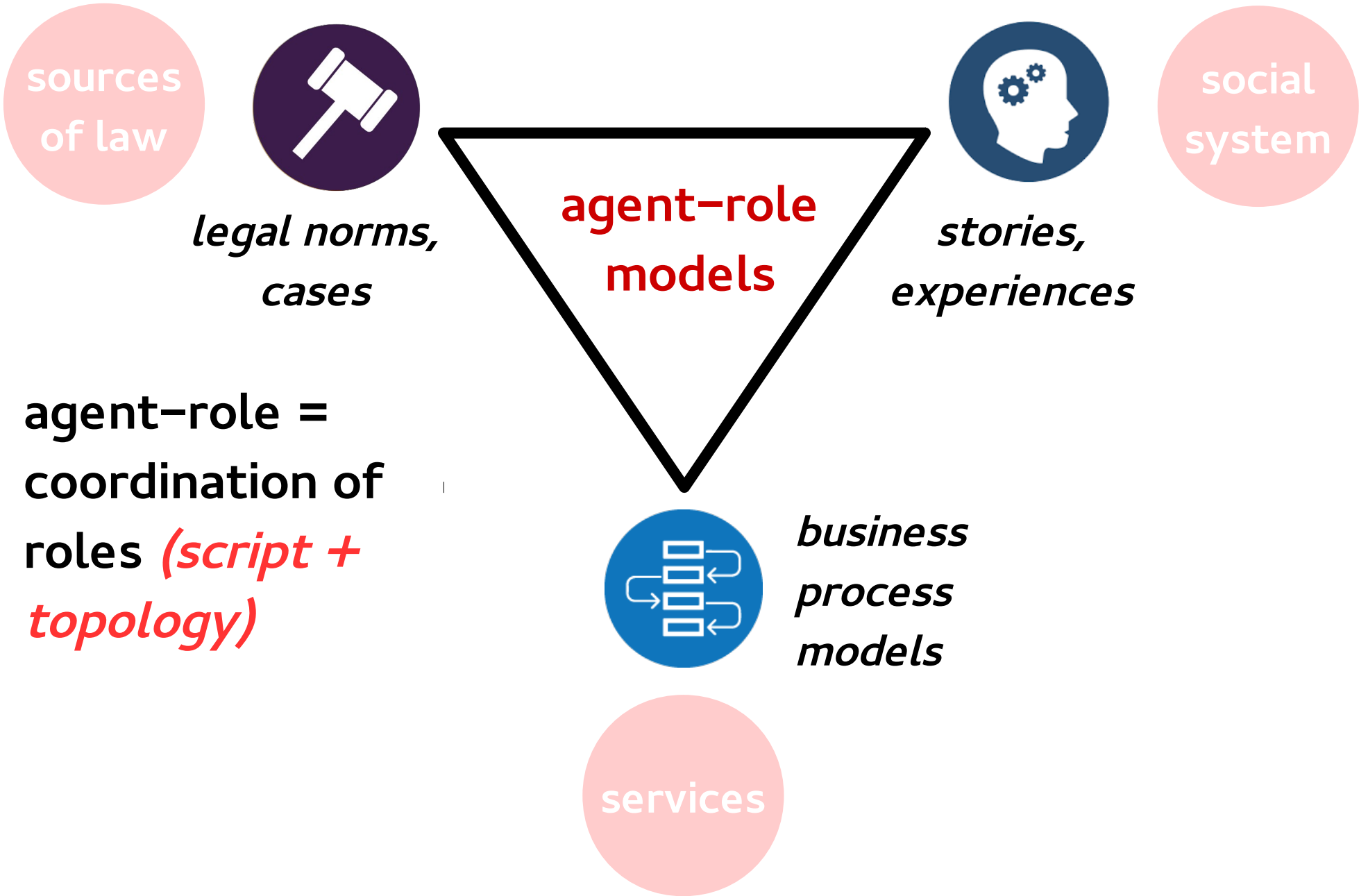
business process models

services

Integrating model: agent-role

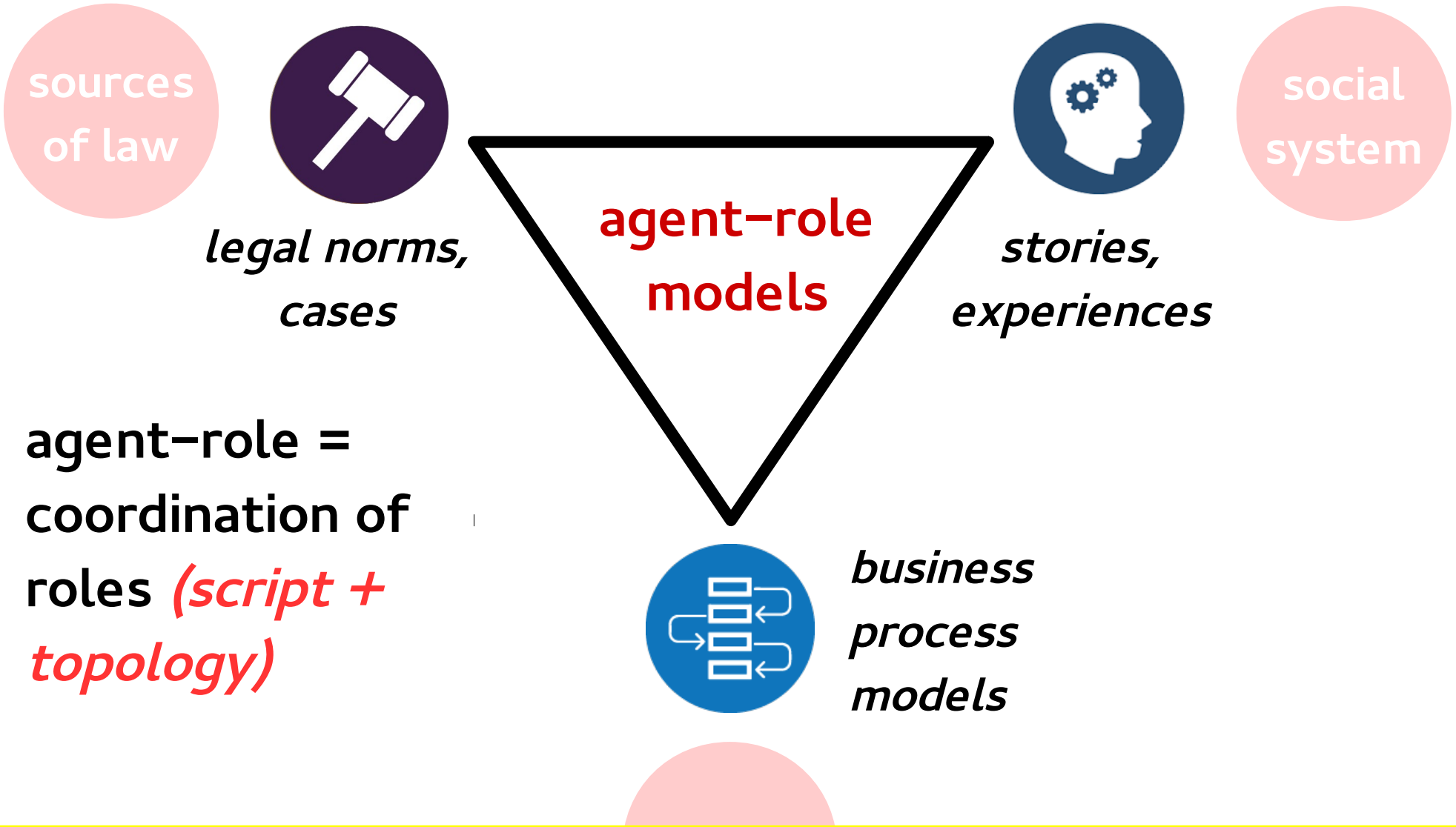


Integrating model: agent-role



agent–role = coordination of roles *(script + topology)*

Integrating model: agent-role



agent–role = coordination of roles *(script + topology)*

NB: the model abstracts the individuals

An exercise of application

Domain: Real-estate transactions

Normal sale

Real estate A, worth 500.000 €

Normal sale

Real estate A, worth 500.000 €

With 6% transfer tax, if sold 30.000 € taxes

Hidden payment scenario

X wants to give to Y 300.000 €.

X sells Real estate A (worth 500.000 €) for 200.000 € to Y

with 6% transfer tax, 10.000 €.

(taxes for a direct transfer would be higher)

Swap-scheme scenario

X wants B, Y wants A.

Owner: X. Real estate A,
worth 10.000.000 €

Owner: Y. Real estate B,
worth 10.000.000 €

6% transfer tax: if sold,
1.200.000 € (total)

Swap-scheme scenario

X wants B, Y wants A.

Owner: X. Real estate A,
worth 10.000.000 €

Owner: Y. Real estate B,
worth 10.000.000 €

"Why don't we decrease the nominal price?"

Swap-scheme scenario

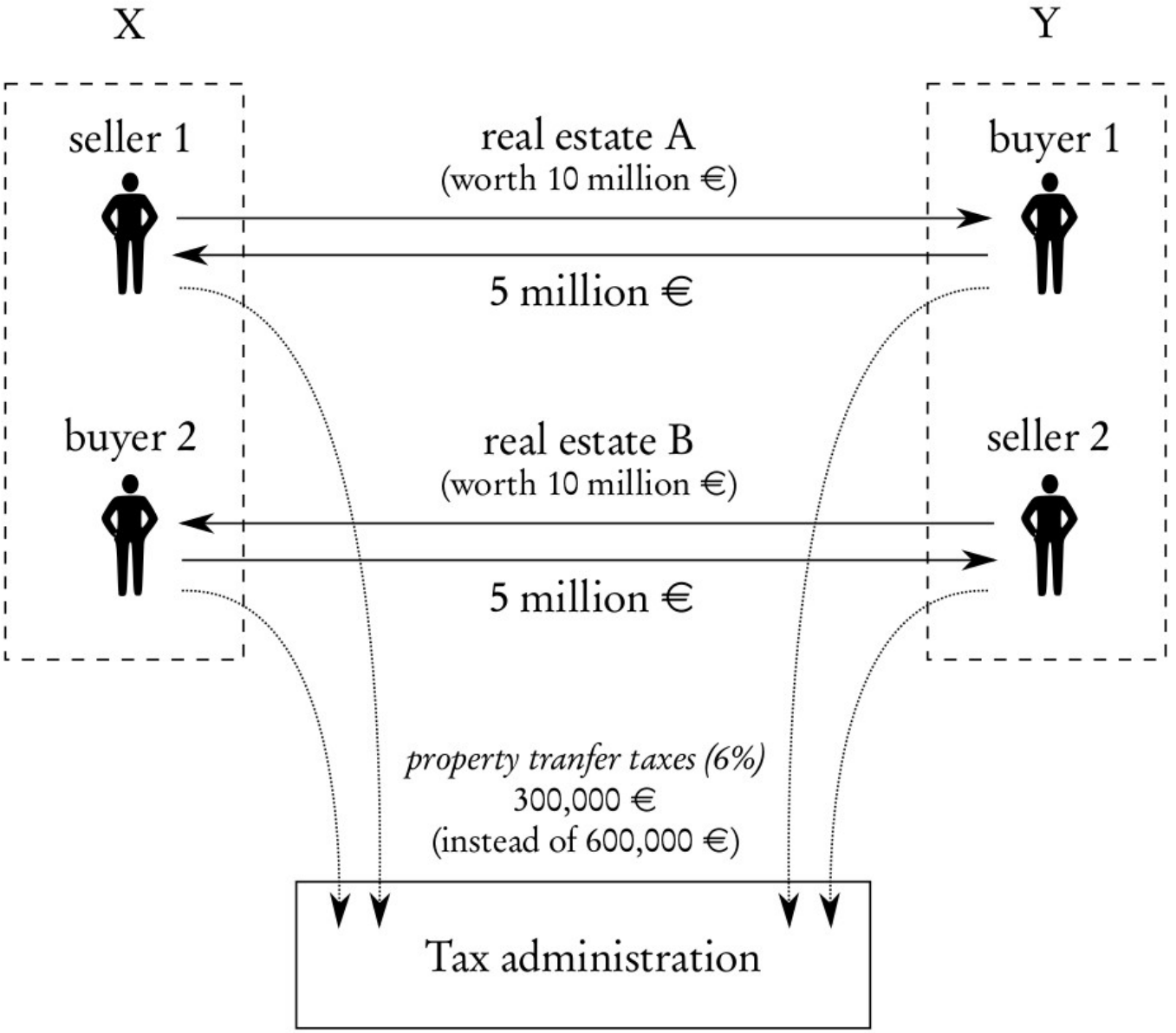
X wants B, Y wants A.

Owner: X. Real estate A,
sold for **5.000.000** €

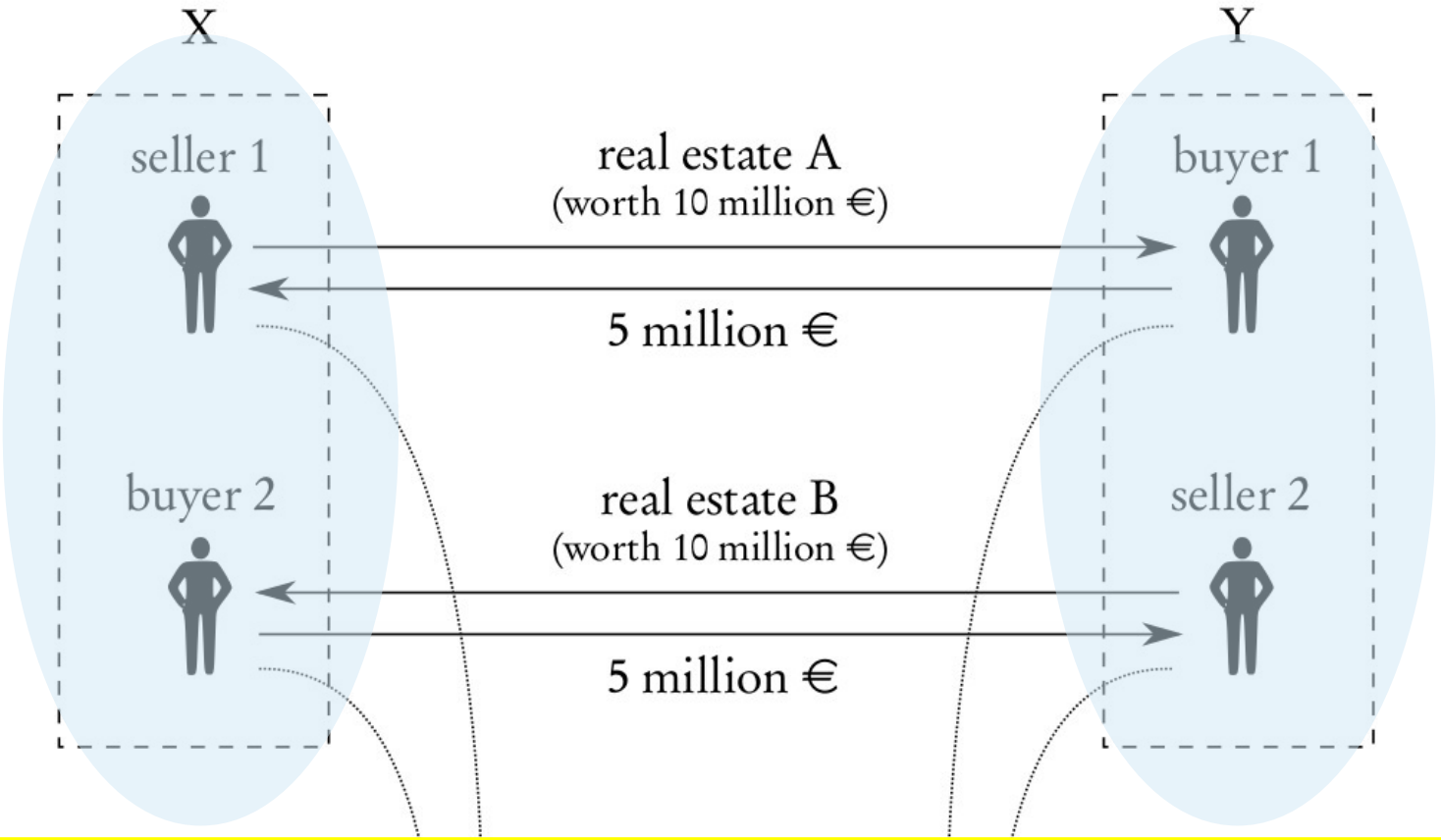
Owner: Y. Real estate B,
sold for **5.000.000 €**

6% transfer tax:
600.000 € (total)

Swap-scheme topology

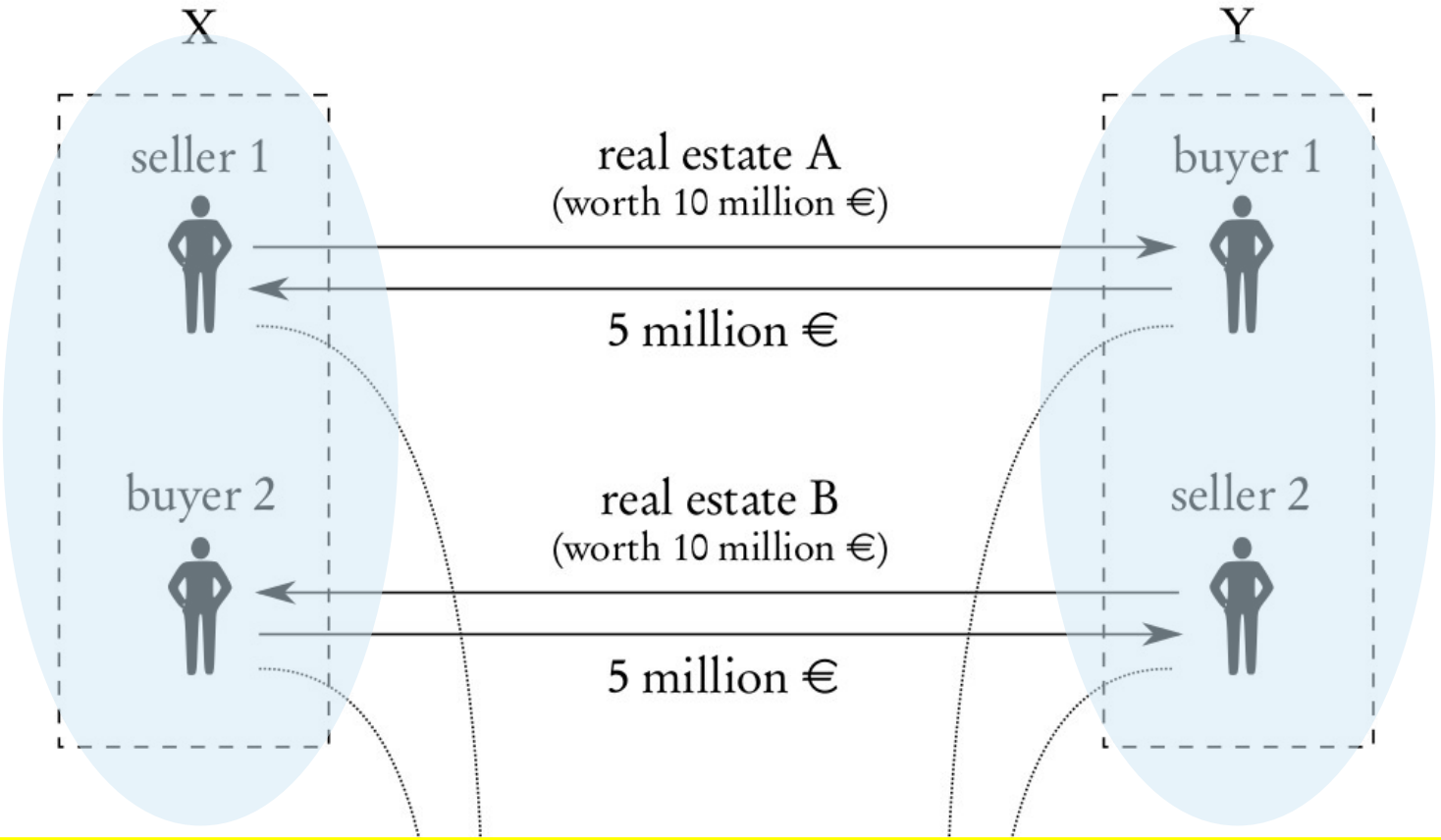


Swap-scheme topology



internal topologies – *intentional coordination*

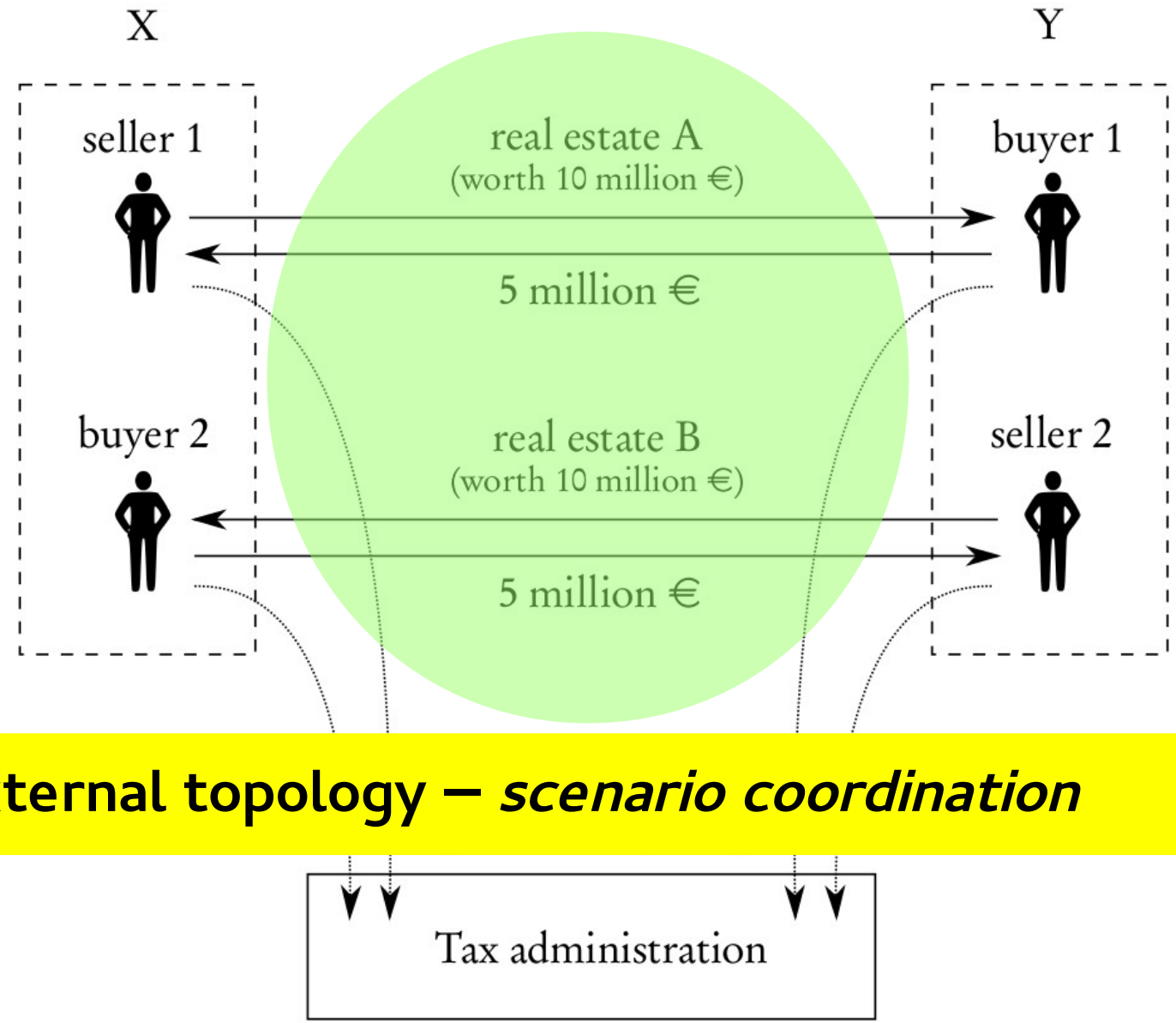
Swap-scheme topology



internal topologies – *intentional coordination*

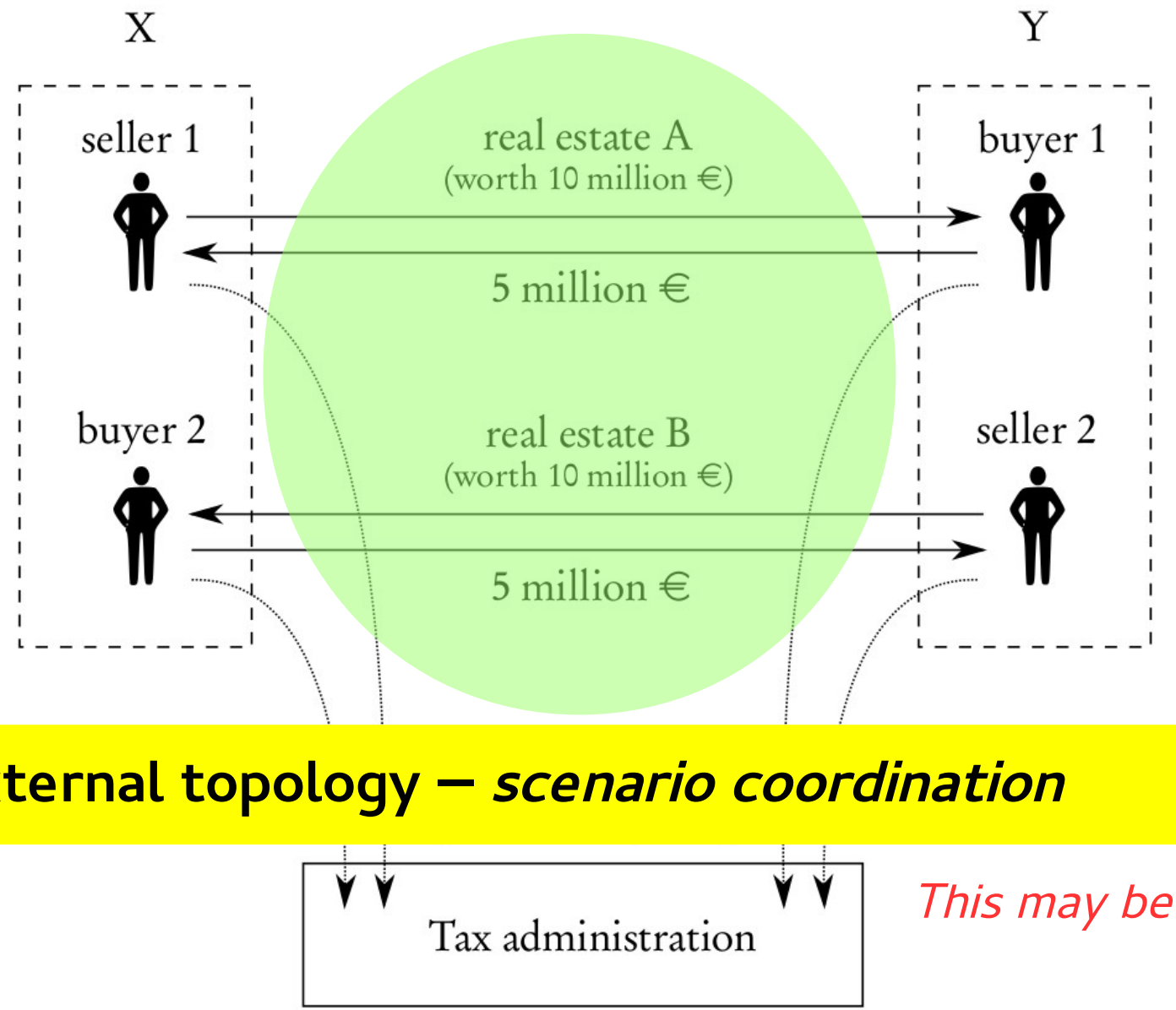
base for concentration of interests (company, family, etc.)

Swap-scheme topology



external topology – *scenario coordination*

Swap-scheme topology

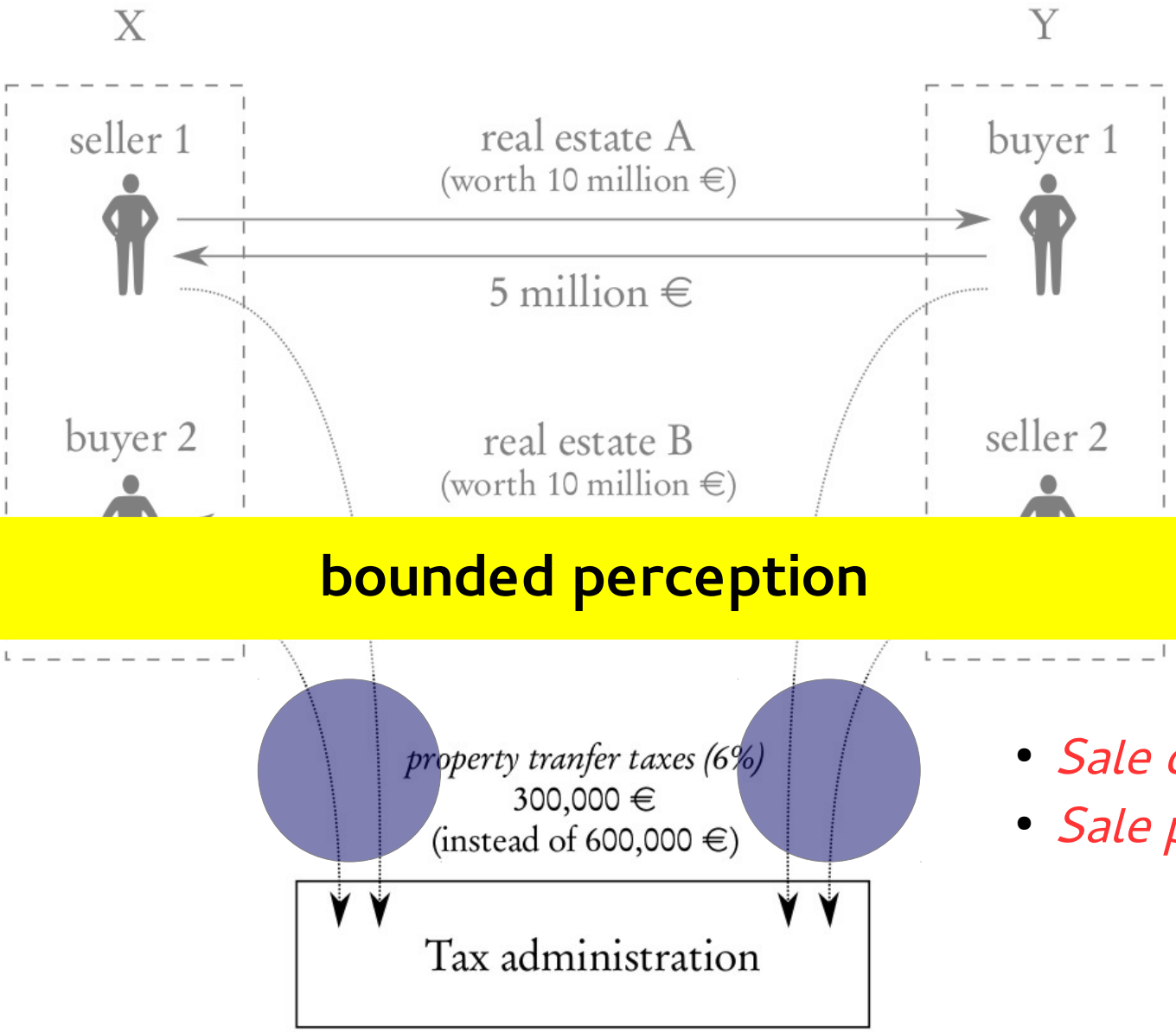


external topology – *scenario coordination*

This may be contingent...

Monitoring and diagnosis

Taking the diagnoser view



bounded perception

- *Sale declarations*
- *Sale payments*

Monitoring and diagnosis

- The observation abilities of the diagnoser are necessarily limited.
- Cognitive resources for reasoning are also limited. It does not make sense to ***interpret*** all the data.

Monitoring and diagnosis

- We consider an architecture affine to the Dual Process theory of reasoning:
  - ***monitoring*** (fast, reactive) for selecting suspicious cases
  - ***diagnosis*** (slow, reflective) for investigating them.

Monitoring

Monitoring filters can be constructed by constrast of normal cases with abnormal ones.

e.g. *duty to pay taxes*

- **not** to pay taxes (*syntaxic, qualitative failure*)
  - Use a timeout rule, asynchronous check

Monitoring

Monitoring filters can be constructed by constrast of normal cases with abnormal ones.

e.g. *duty to pay taxes*

- **not** to pay taxes (*syntaxic, qualitative failure*)
  - Use a timeout rule, asynchronous check
- pay a **wrong** amount in respect to the declared income (*syntaxic, quantitative failure*)
  - Use an operational rule, synchronous check

Monitoring

Monitoring filters can be constructed by constrast of normal cases with abnormal ones.

e.g. *duty to pay taxes*

- pay *the correct but* **not the right** amount, by using tax evasion schemes (*semantic failure*)
  - **contrast** determines threshold conditions

Monitoring

Monitoring filters can be constructed by constrast of normal cases with abnormal ones.

e.g. *duty to pay taxes*

- pay *the correct but* **not the right** amount, by using tax evasion schemes (*semantic failure*)
  - **contrast** determines threshold conditions

In our case, adequate thresholds are decided upon **average market price.**

Selection rule in Prolog

Suppose the threshold is placed at -40% of market price (e.g. after statistical analysis):

```
suspiciousPrice(Price, Estate, Time) :-
   marketPrice(MarketPrice, Estate, Time),
   Price =< (MarketPrice * 60)/100.

suspiciousSale(Seller, Buyer, Estate, Price, Time) :-
   declaration(sale(Seller, Buyer, Estate, Price, Time)),
   suspiciousPrice(Price, Estate, Time).
```

Diagnosis (main action)

At this point, we check if suspicious sales are consistent with a **swap-scheme *agent-role***:

```
ActionEvidenceOfSwap(
   sale(Seller1, Buyer1, EstateA, PriceA, T1),
   sale(Seller2, Buyer2, EstateB, PriceB, T2)
) :-
   suspiciousSale(Seller1, Buyer1, EstateA, PriceA, T1),
   suspiciousSale(Seller2, Buyer2, EstateB, PriceB, T2),
   not(EstateA = EstateB),
   not(Seller1 = Seller2), not(Buyer1 = Buyer2).
```

Diagnosis (additional evidence)

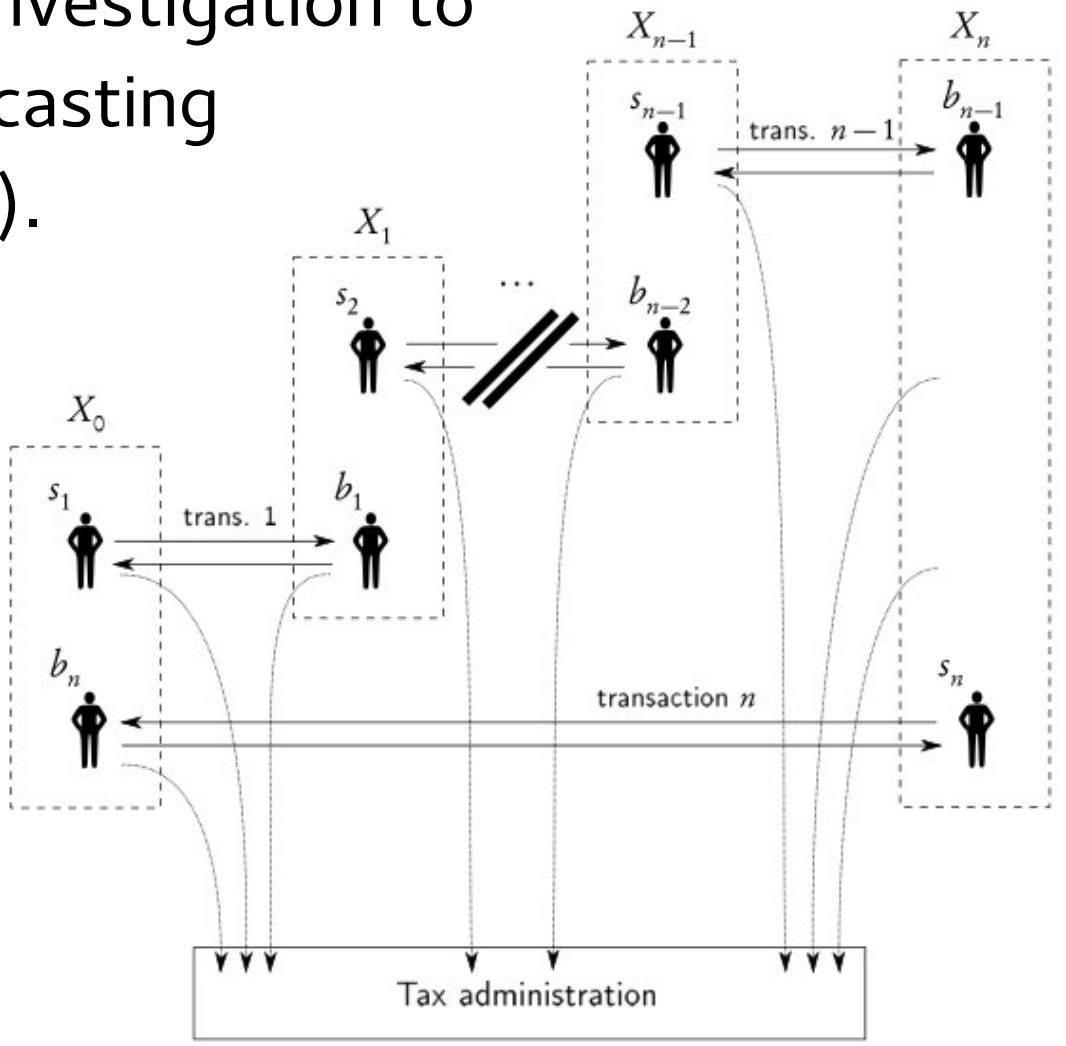
..and when suspicious coordinated behaviour is found, we look for circumstantial evidence about ***structures of concentration of interests***:

```
actionAndCircumstantialEvidenceOfSwap(
   sale(Seller1, Buyer1, EstateA, PriceA, T1),
   sale(Seller2, Buyer2, EstateB, PriceB, T2)
) :-
   actionEvidenceOfSwap(sale(Seller1, Buyer1, EstateA,
   PriceA, T1),
   sale(Seller2, Buyer2, EstateB, PriceB, T2)),
   relatedTo(Seller1, SharedStructure1),
   relatedTo(Buyer2, SharedStructure1),
   relatedTo(Seller2, SharedStructure2),
   relatedTo(Buyer1, SharedStructure2).
```

Swap-scheme generalization

- Through computational means, we can easily generalize the investigation to *n-steps* swaps (obfuscasting traces of coordination).

  ...this includes the "hidden payment" scenario!



Conclusion

Conclusion

- This exercise showed a simplified application of a more general framework

Conclusion

- This exercise showed a simplified application of a more general framework
  - The most important message is that…

Conclusion

- This exercise showed a simplified application of a more general framework
  - The most important message is that…

agency trascends individuals.

Conclusion

- This exercise showed a simplified application of a more general framework
  - The most important message is that…

    agency trascends individuals.

    ..and we require abstractions like **agent–roles** to pass to higher-level interpretation.

Further developments

- This exercise showed a simplified application of a more general framework
  - Prolog conflates strong negation with default negation. What if some ***information*** is simply ***unavailable***? Other approaches (EBA) would be more appropriate.

Further developments

- This exercise showed a simplified application of a more general framework
  - Prolog conflates strong negation with default negation. What if some ***information*** is simply ***unavailable***? Other approaches (EBA) would be more appropriate.
  - Scenarios should be collected in an **adequate representation** for the modeler taking ***causation*** into account. We suggested an extension to Petri nets, with normative and agentive positions.

Further developments

- This exercise showed a simplified application of a more general framework
  - For the **acquisition**, in previous work we proposed an ***intecremental method***, starting from UML-diagrams (similarly to requirement engineering practices)

Further developments

- This exercise showed a simplified application of a more general framework
  - For the **acquisition**, in previous work we proposed an ***intecremental method***, starting from UML-diagrams (similarly to requirement engineering practices)
  - About ***computational complexity***, existing **decomposition** techniques could be used in principle to compile the database of scenarios offline, and for realtime exploitation.