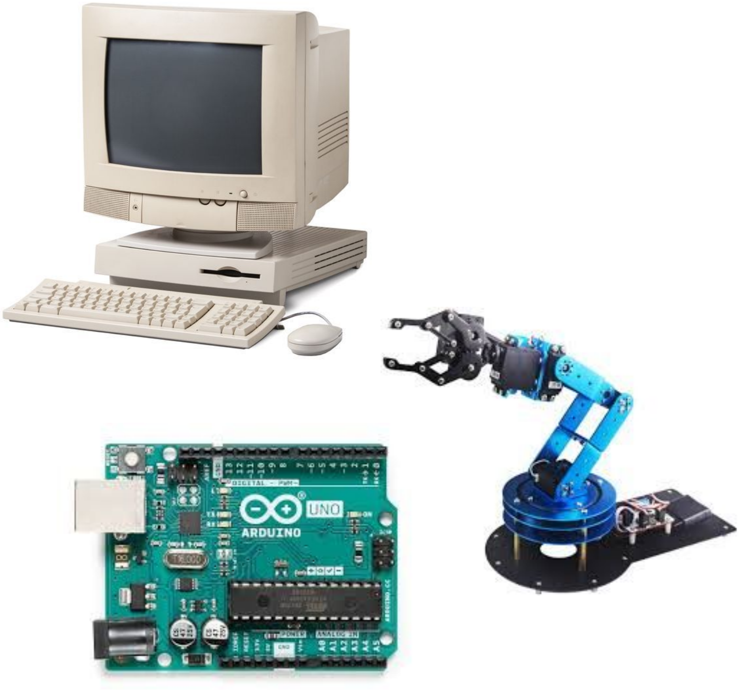
# Normware engineering: opportunities and open problems

7 November 2024, *IPA Fall Days*

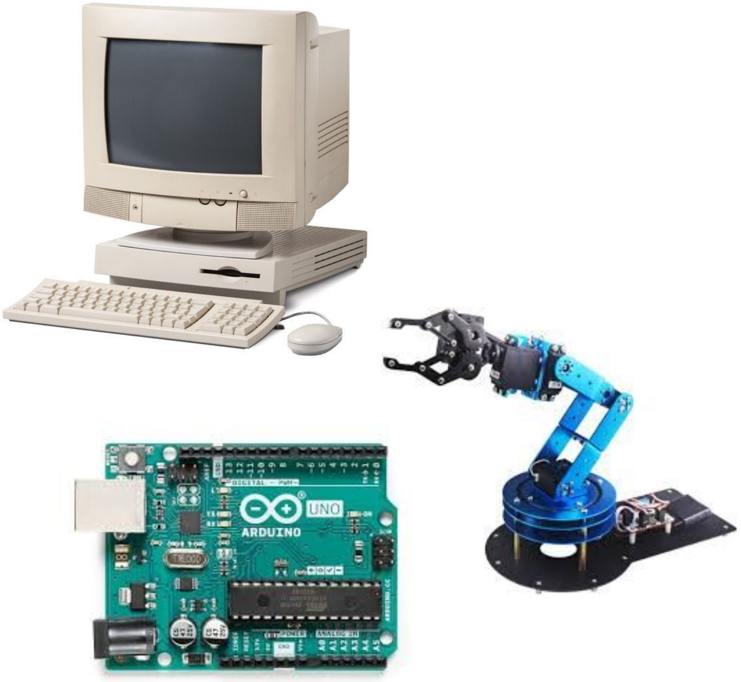
Fall Days on Models for Constructing Software

Giovanni Sileno, g.sileno@uva.nl
Socially Intelligent Artificial Systems (SIAS),
Informatics Institute, University of Amsterdam

# from individual devices...

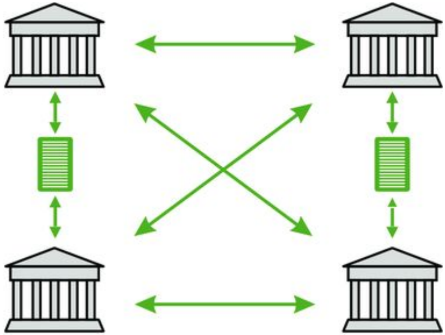


# from individual devices to digital social systems...



Social networks

Distributed Ledgers



Digital Markets

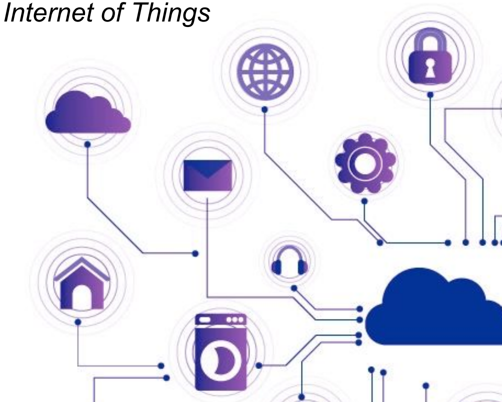
Internet of Things

# from “mechanical” to “institutional” approaches to computation...

not *instructions,* but ***contracts***, ***regulations***, ***laws***...

*Digital Markets*

*Internet of Things*



# from “mechanical” to “institutional” approaches to computation...

not *instructions*, but ***contracts***, ***regulations***, ***laws***…

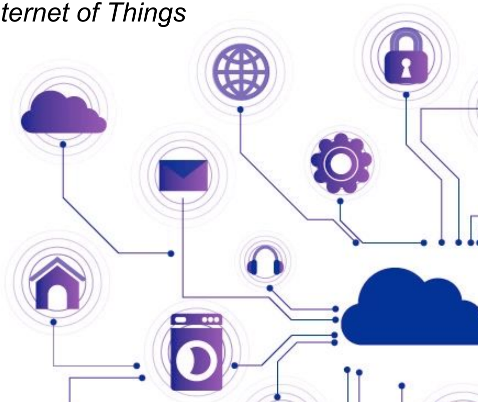
**focus on**
**PERFOMANCE**

→

**focus on**
**COORDINATING EXPECTATIONS**

*Digital Markets*

*Internet of Things*



# overarching question

- Instead of relying on infrastructures ruled by single actors, we engage with the challenges concerning:
  - how to design and deploy ***computational infrastructures***
  - in which **users may decide** and **enact their own policies**

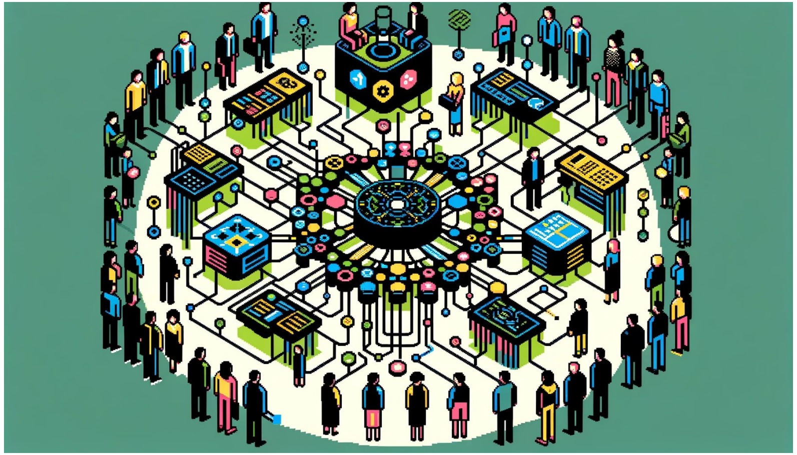
# overarching question

- Instead of relying on infrastructures ruled by single actors, we engage with the challenges concerning:
    - how to design and deploy ***computational infrastructures***
    - in which **users may decide** and **enact their own policies**

reusable components and mechanisms!

# overarching motivation

- By enabling a pluralism of interactional mechanisms via “private” regulations, we make explicit the continuity holding between computational distributed systems and social systems.



This is relevant to ***responsible computing*** initiatives, concerning eg. responsible/participatory AI or responsible Internet.

# practical issues

- extreme dispersion on what/how to specify computational regulation…

OpenFisca, Catala, FormaLex, FCL, Symboleo, Stipula, Blawx, DCR Graphs, Eiger, Orlando, Accord, CSL, Logical English, Epilog, LLD, UMLSC, TAC, BCL, DCMs, RuleML, MODELLER, LMC, CL, PENELOPE, SCIFF, eFLINT, RuleSpeak, ALEF, Publi.Codes, Avola, USoft (SBVR), …

Chun et al. (2024), Kaptijn & Klaver (2024), Parvizimosaed et al. (2022)

# practical issues

- extreme dispersion on what/how to specify computational regulation…

OpenFisca, Catala, FormaLex, FCL, Symboleo, Stipula, Blawx, DCR Graphs, Eiger, Orlando, Accord, CSL, Logical English, Epilog, LLD, UMLSC, TAC, BCL, DCMs, RuleML, MODELLER, LMC, CL, PENELOPE, SCIFF, eFLINT, RuleSpeak, ALEF, Publi.Codes, Avola, USoft (SBVR), …

Chun et al. (2024), Kaptijn & Klaver (2024), Parvizimosaed et al. (2022)

- …to add to all specific languages applied in technical tasks!

eg. BGP policies for routing, XACML policies for access/usage control, Protune, Rei, Ponder, TrustX for cloud infrastructures, …

➡ *interoperability is essentially impossible.*

# practical issues

- extreme dispersion on what/how to specify computational regulation…

OpenFisca, Catala, FormaLex, FCL, Symboleo, Stipula, Blawx, DCR Graphs, Eiger, Orlando, Accord, CSL, Logical English, Epilog, LLD, UMLSC, TAC, BCL, DCMs, RuleML, MODELLER, LMC, CL, PENELOPE, SCIFF, eFLINT, RuleSpeak, ALEF, Publi.Codes, Avola, USoft (SBVR), …

Chun et al. (2024), Kaptijn & Klaver (2024), Parvizimosaed et al. (2022)

- …to add to all specific languages applied in technical tasks!

eg. BGP policies for routing, XACML policies for access/usage control, Protune, Rei, Ponder, TrustX for cloud infrastructures, …

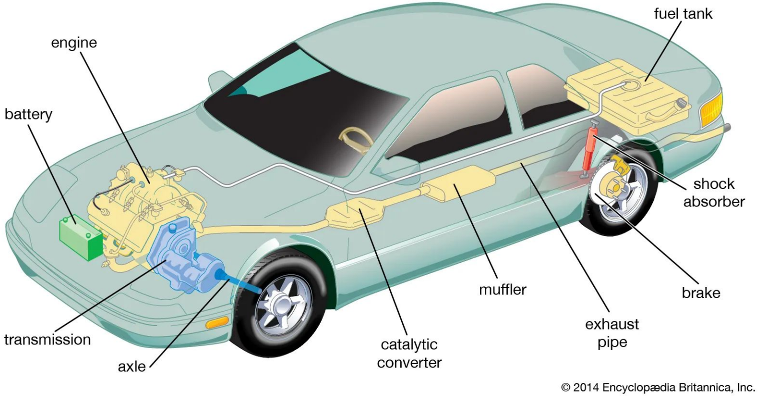
→ *interoperability is essentially impossible.* each framework/tool is constructed based on specific types of normative tasks and domain…

# What should we “standardize”?

all components at once?

# What should we “standardize”?

all components at once?

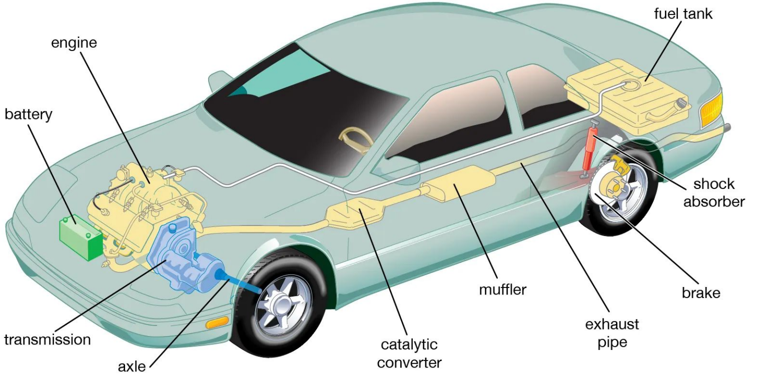


© 2014 Encyclopædia Britannica, Inc.

the structure first, and then possibly providing additional standards for the individual components

# What should we “standardize”?

all components at once?



© 2014 Encyclopædia Britannica, Inc.

***What is the structure we should look at?***

the structure first, and then possibly providing additional standards for the individual components

# Overview of my talk

- core components of regulative mechanisms: **normware** layer of design
- how to specify regulative mechanisms and the challenges that comes with (we’ll use DCPL as a sandbox)
- how this may work for actual infrastructure (revisiting the Responsible Internet proposal)

# Part I:
# what is normware?

Sileno, G., Code-Driven Law NO, Normware SI!, CRCL 2022

# A tentative ontology

HARDWARE

- **physical device**
- when running
  **⇒ physical process**
- situated in a
  **physical environment**

SOFTWARE

- **symbolic device**
- when running
  **⇒ symbolic process**
- relies on
  **physical processes**

# A tentative ontology

## HARDWARE

- **physical device**
- when running
  **⇒ physical process**
- situated in a
  **physical environment**

## SOFTWARE

- **symbolic device**
- when running
  **⇒ symbolic process**
- relies on
  **physical processes**

**?**

## NORMWARE

- **?**
- when running
  **⇒ ?**
- relies on **?**

# A tentative ontology

?

| HARDWARE | SOFTWARE | NORMWARE | |
|---|---|---|---|
| ● **physical device** | ● **symbolic device** | ● **?** | **ARTIFACT dimension** |
| ● when running<br>**⇒ physical process** | ● when running<br>**⇒ symbolic process** | ● when running<br>**⇒ ?** | **PROCESS dimension** |
| ● situated in a<br>**physical environment** | ● relies on<br>**physical processes** | ● relies on **?** | |

# Normware as artifacts…

# Normware as artifacts:
# 1. directives concerning regulation

the cookie jar
must be full

you are prohibited
to eat cookies

you can not eat
cookies

# Normware as artifacts:
# 1. directives concerning regulation

aiming to regulate situations in the world

the cookie jar must be full

aiming to regulate behaviour

you are prohibited to eat cookies

you can not eat cookies

# Normware as artifacts:
# 2. directives concerning terminology/meaning

*what is a cookie?*

*what is a jar?*

**the cookie jar must be full**

*what does it mean to be full?*

*who is you?*

*what is eating?*

**you are prohibited to eat cookies**

**you can not eat cookies**

# Normware as artifacts:
# 3. directives concerning expectations

*eating cookies → cookies are destroyed → the jar is not full*

the cookie jar
must be full

*practical* normative reasoning always require some world knowledge

you are prohibited to eat cookies

you can not eat cookies

# Normware as artifacts:
# 4. devices intended to regulate

behaviour can be regulated even if we do not have access to the inner decision-making mechanism!

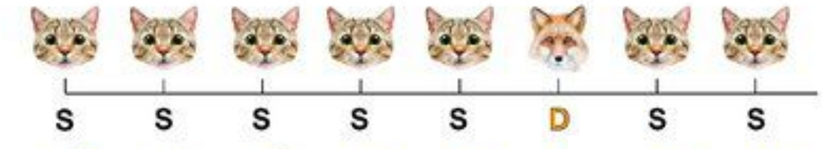
doors regulate entrances

semaphores regulate traffic

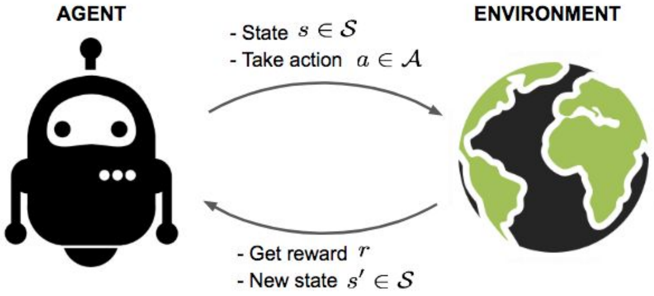
# Normware as artifacts:
# 4. devices intended to regulate

**black-boxes (eg. ML models) are also artifacts expressing some form of normativity/normality**

*"is this a cat?"*



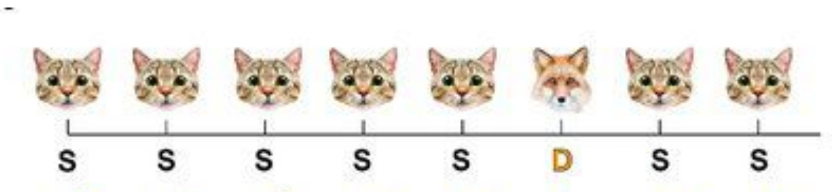
*"how to (best) behave in certain conditions?"*


AGENT
ENVIRONMENT
- State $s \in \mathcal{S}$
- Take action $a \in \mathcal{A}$
- Get reward $r$
- New state $s' \in \mathcal{S}$

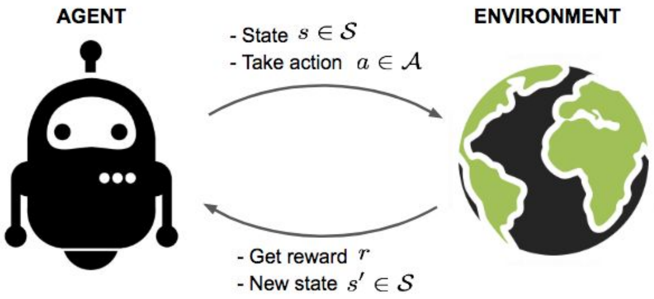
# Normware as artifacts:
# 4. devices intended to regulate

**black-boxes (eg. ML models) are also artifacts expressing some form of normativity/normality**

*"is this a cat?"*



*"how to (best) behave in certain conditions?"*


AGENT
ENVIRONMENT
- State $s \in \mathcal{S}$
- Take action $a \in \mathcal{A}$
- Get reward $r$
- New state $s' \in \mathcal{S}$

**from a functional point of view, they also count as normware!**

# Normware as processes

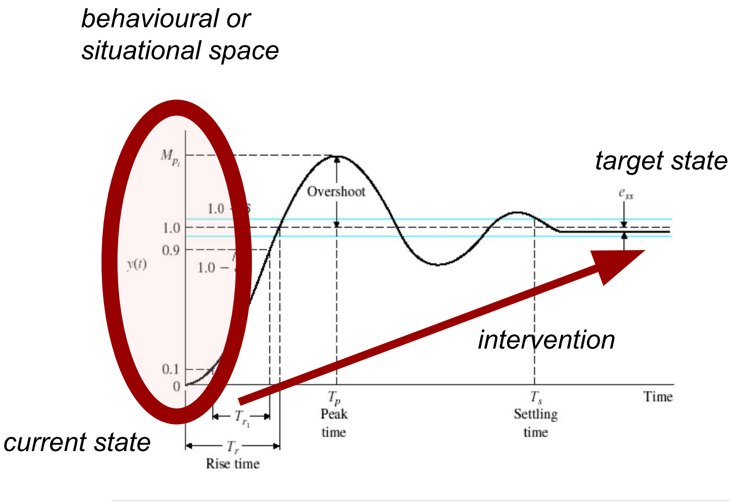
# Normware as processes:
# 1. regulation as control

Whether artificial or natural, designed or emergent,

**what counts in control is**

- the existence of some ***reference*** (the **target** of control),
- which the entity is set to either *approach* or *avoid* (the **direction** of control).

# Normware as processes:
# 1. regulation as control

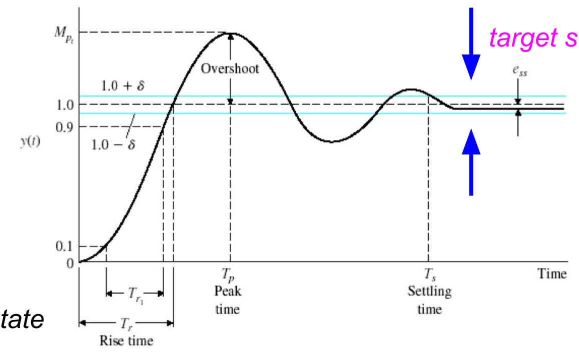


Whether artificial or natural, designed or emergent,

**what counts in control is**

- the existence of some ***reference*** (the **target** of control),
- which the entity is set to either *approach* or *avoid* (the **direction** of control).

# Normware as processes:
# 1. regulation as control



*direction*

*target state*

*current state*

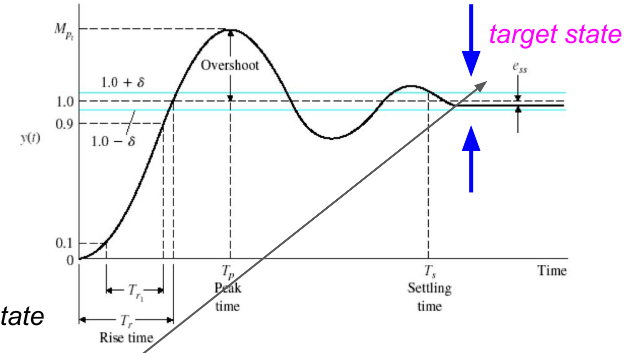
Whether artificial or natural, designed or emergent,

**what counts in control is**

- the existence of some ***reference*** (the **target** of control),
- which the entity is set to either *approach* or *avoid* (the **direction** of control).

*by defining directives by this control signature (<u>target</u>, <u>direction</u>), any regulative mechanisms can be abstracted from its implementation.*

# Normware as processes:
# 1. regulation as control



*direction*

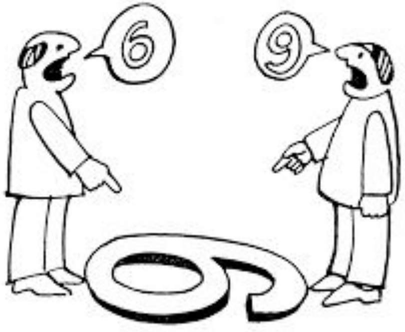
*target state*

*current state*

***what defines the references though?***

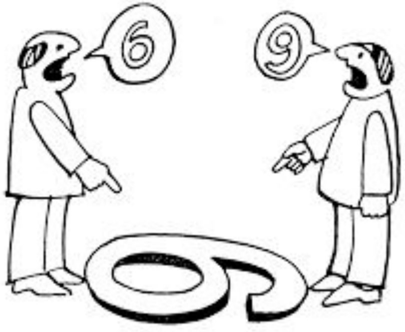
# Normware as processes:
# 2. higher-order indetermination

indeterminacy of references



# Normware as processes:
# 2. higher-order indetermination

indeterminacy of references

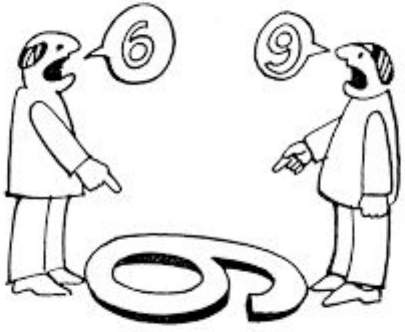


indeterminacy of directives

# Normware as processes:
# 2. higher-order indetermination

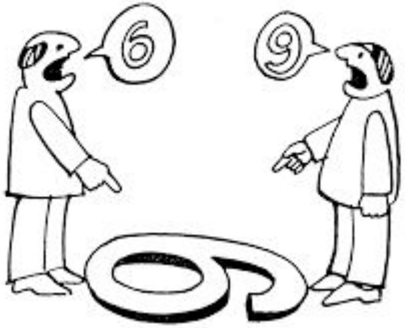
indeterminacy of references

antinomies



indeterminacy of directives

# Normware as processes:
# 2. higher-order indetermination



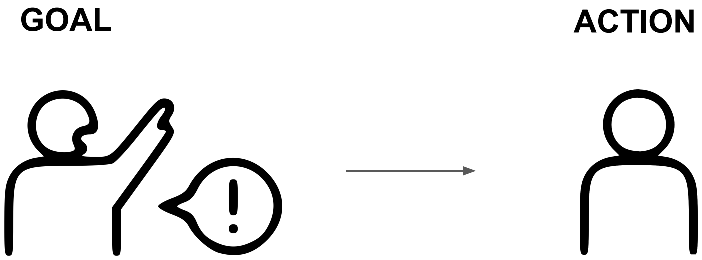
indeterminacy of references

antinomies

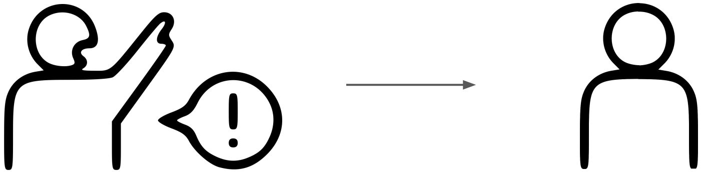
indeterminacy of directives

**mechanisms of conflict resolution**
are needed at systematic level!

# Normware: first-order control



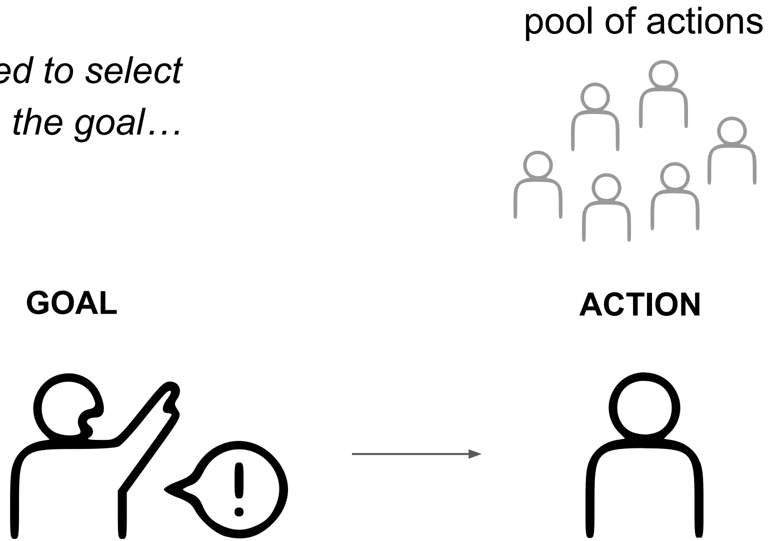
# Normware: first-order control



| | GOAL | ACTION |
|---|---|---|
| ***code-driven*** | instruction | operation |
| ***data-driven*** | labelled dataset | ML black box |

# Normware: first-order control

*In both cases, some method has been used to select the action from a pool of actions based on the goal…*



pool of actions

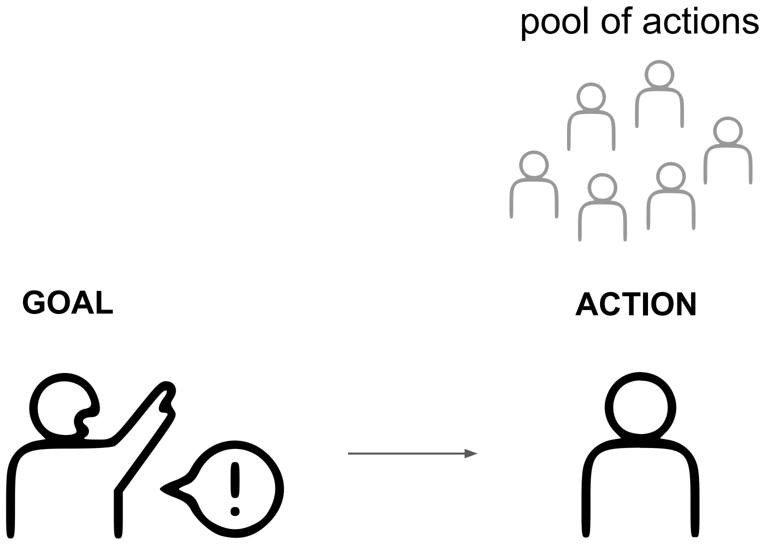
GOAL

ACTION

| | | |
|---|---|---|
| ***code-driven*** | instruction | operation |
| ***data-driven*** | labelled dataset | ML black box |

# Normware: first-order control

*But then, where the goal comes from?*

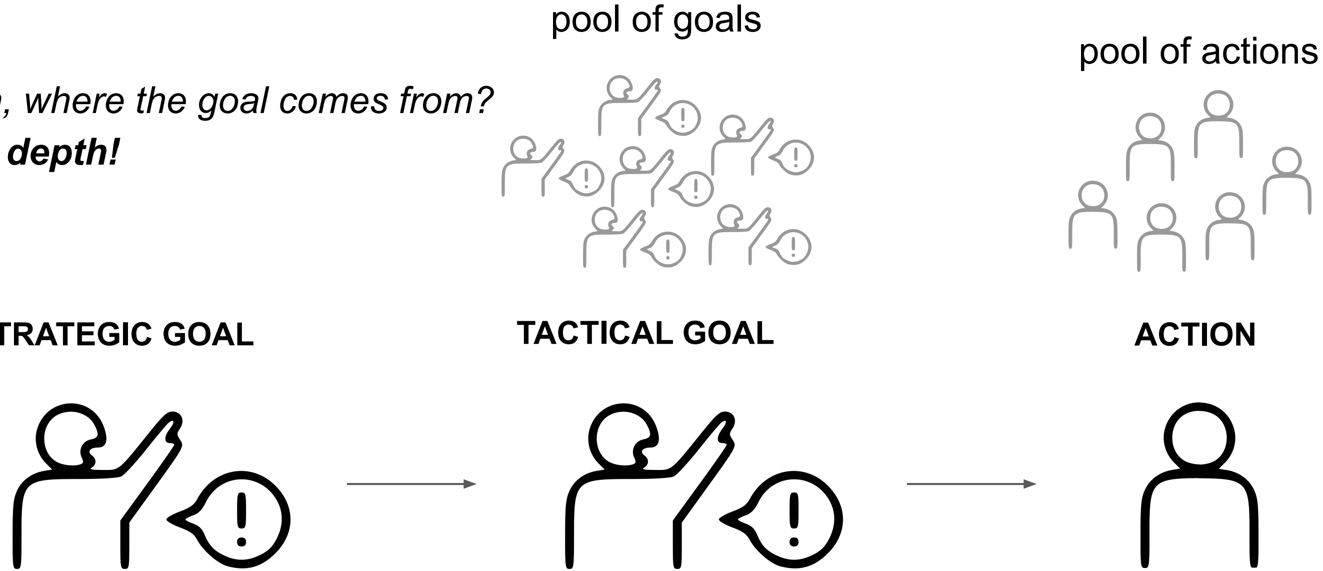


| | | |
|---|---|---|
| ***code-driven*** | instruction | operation |
| ***data-driven*** | labelled dataset | ML black box |

# Normware: second-order control

*But then, where the goal comes from?*

***We add depth!***



pool of goals

pool of actions

STRATEGIC GOAL

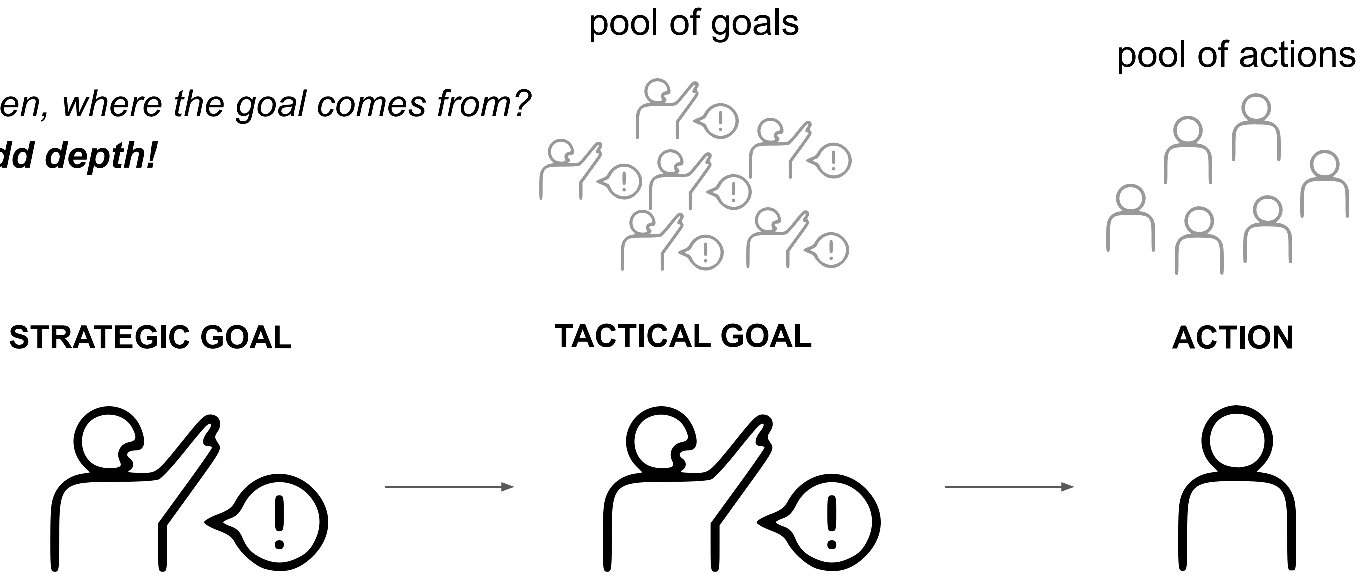
TACTICAL GOAL

ACTION

# Normware: second-order control

*But then, where the goal comes from?*

***We add depth!***

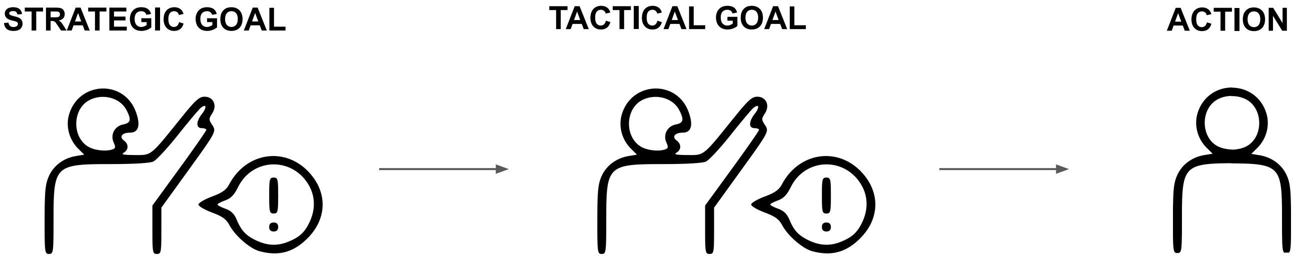


Sileno, G., Boer, A. and van Engers, T., The Role of Normware in Trustworthy and Explainable AI, Proceedings of XAILA workshop: Explainable AI and Law, in conjunction with JURIX 2018

***trustworthy AI*** *and* ***explainable AI*** *issues in ML due to lack of the strategic component*

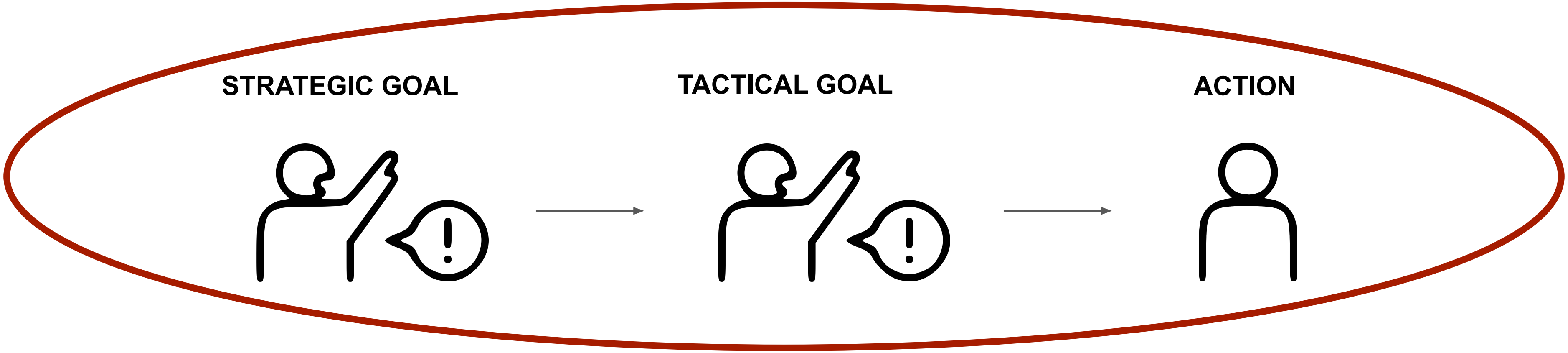
# Normware: second-order control

*cybernetic view on systems:* ***policy**, **intelligence**, **operations***



# Normware: second-order control

*cybernetic view on systems:* ***policy*****,** ***intelligence*****,** ***operations***

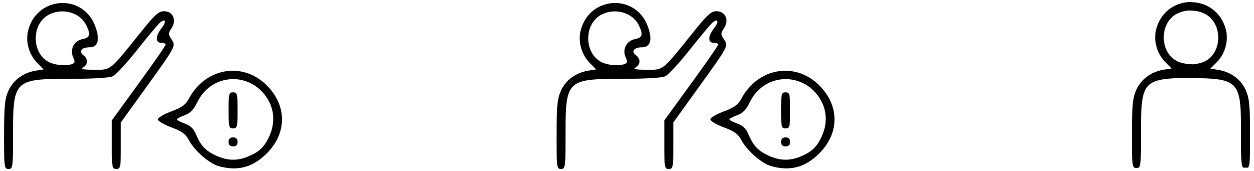


**STRATEGIC GOAL** → **TACTICAL GOAL** → **ACTION**

…yet it is about a single “organism”, not an “ecology” → **“TOTALITARIAN” architecture**

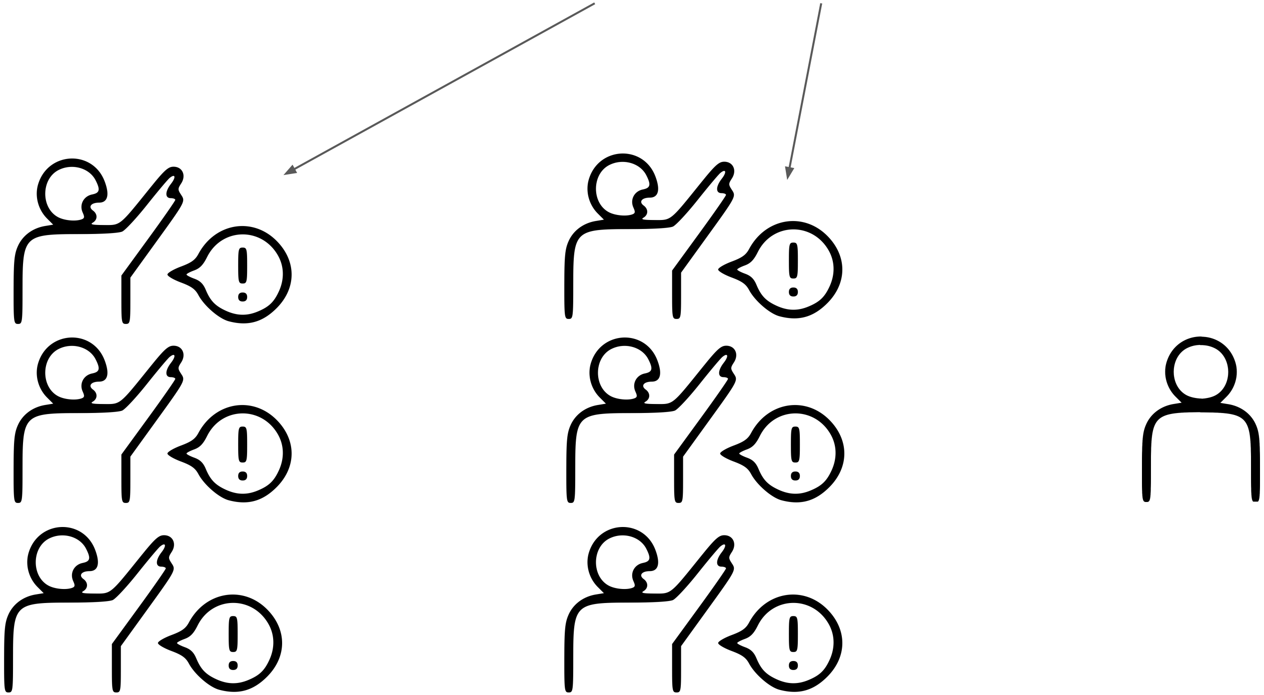
# Normware: plural second-order control

*we need to acknowledge the presence of several* ***autonomous entities****,*



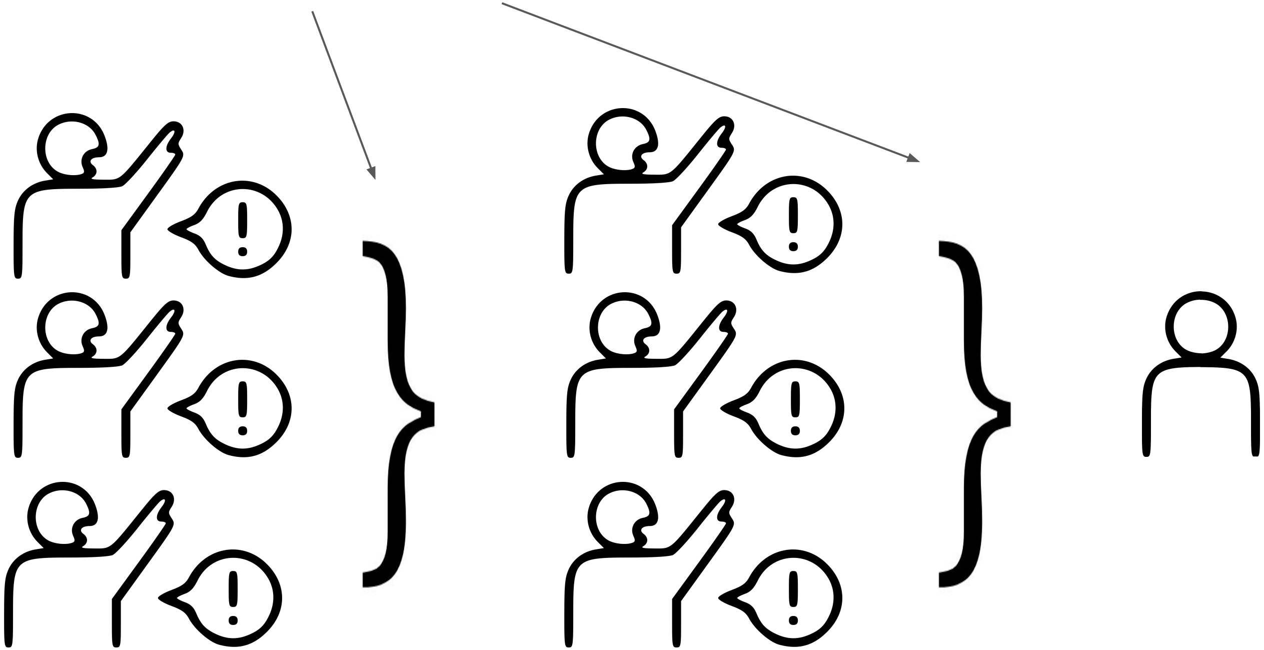
# Normware: plural second-order control

*we need to acknowledge the presence of several* ***autonomous entities****,*



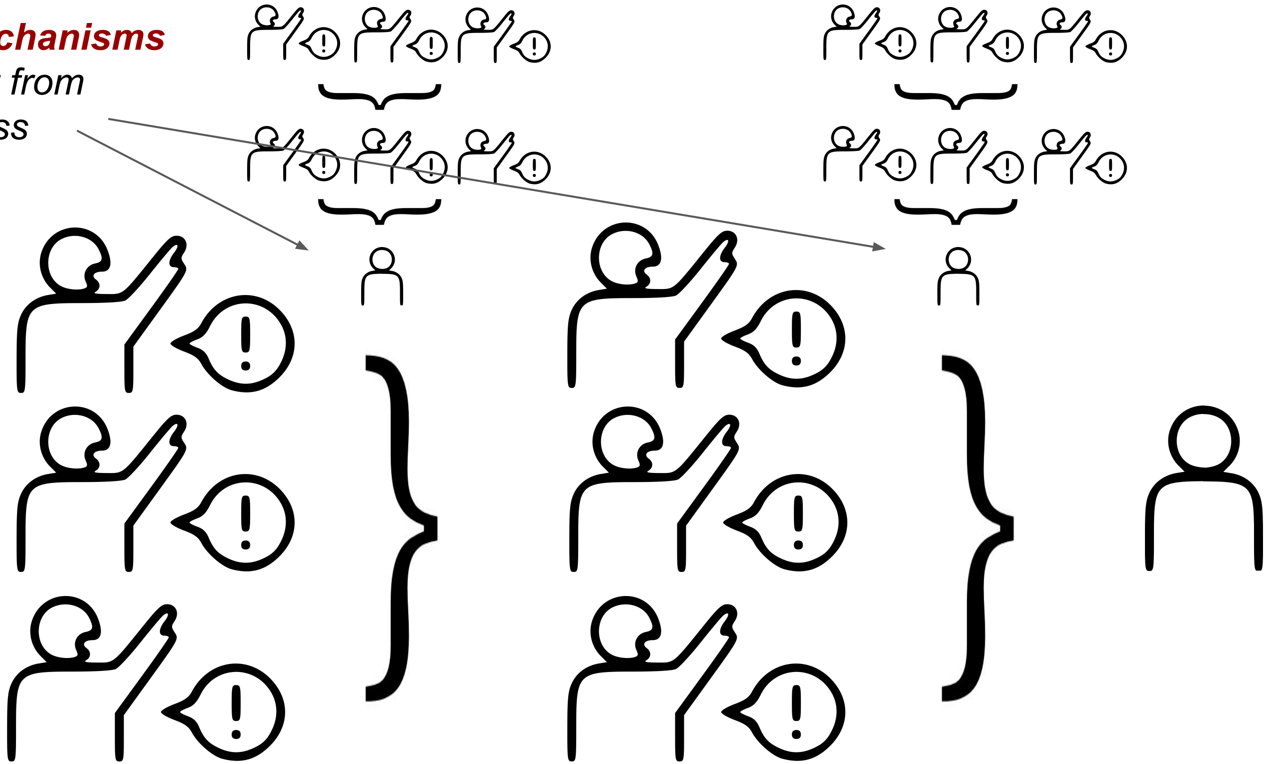
# Normware: plural second-order control

*we need to acknowledge the presence of several* ***autonomous entities****,*
*and adequate conflict* ***resolution mechanisms***



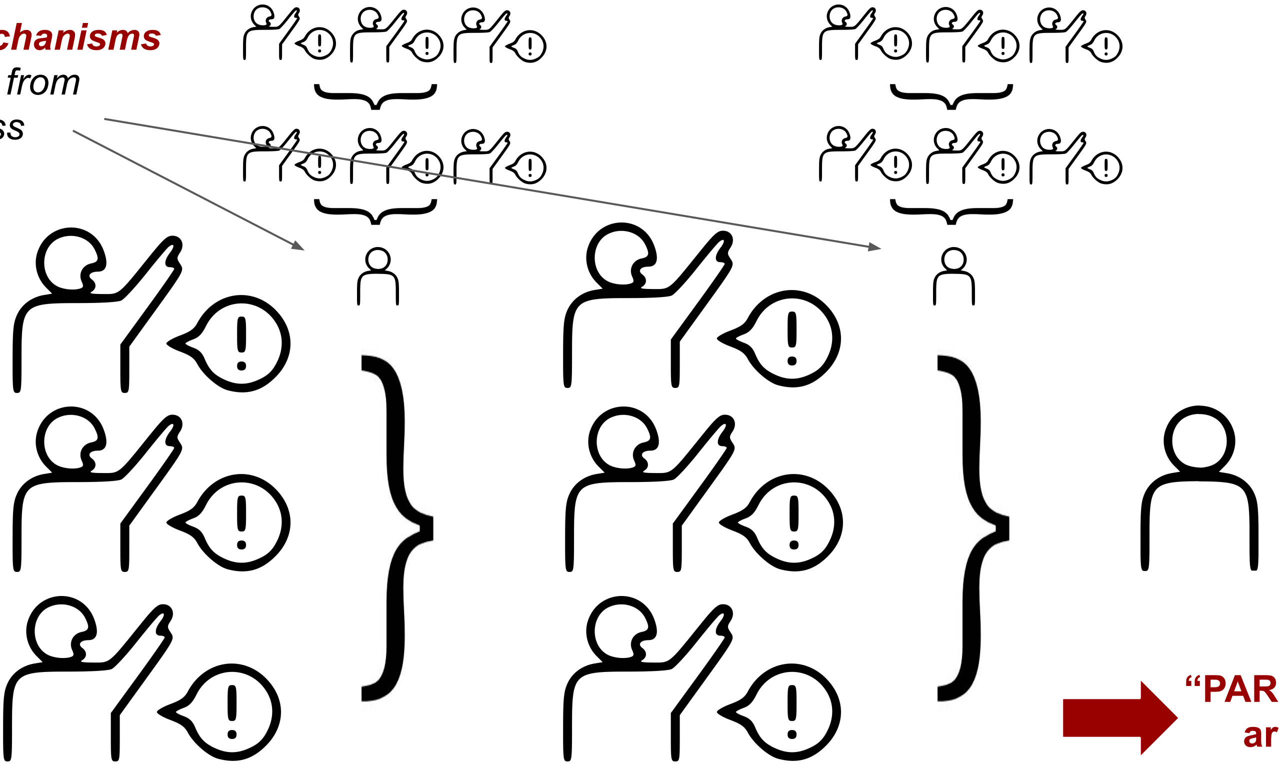
# Normware: plural second-order control

***resolution mechanisms*** *may also result from a similar process*



# Normware: plural second-order control

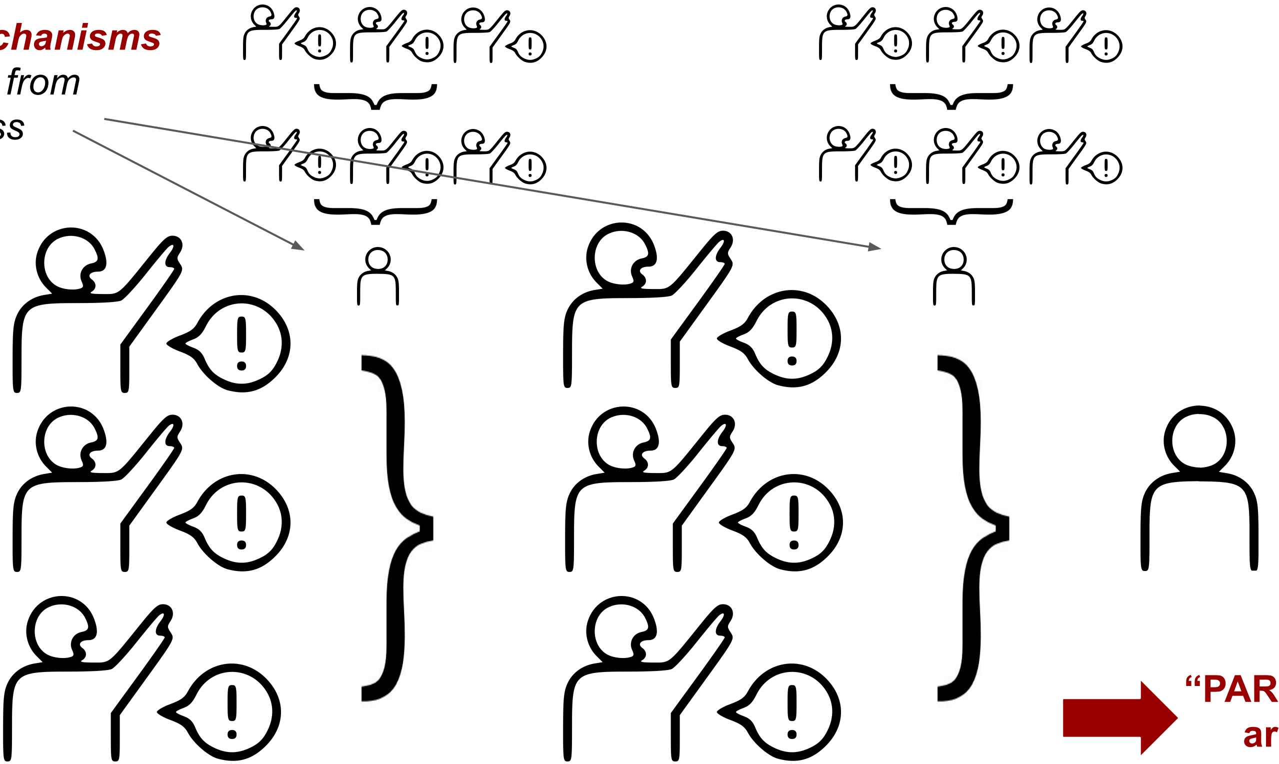
***resolution mechanisms***
*may also result from*
*a similar process*



“PARTICIPATORY”
architecture

# Normware: plural second-order control

***resolution mechanisms***
*may also result from*
*a similar process*



**“PARTICIPATORY” architecture**

Sileno, G., Grosso, P., Accounting Value Effects for Responsible Networking. Proceedings of ACM SIGCOMM 2021 Workshop on Technologies, Applications, and Uses of a Responsible Internet (TAURIN2021).

# A less tentative ontology

| | | **?** |
|---|---|---|
| **HARDWARE** | **SOFTWARE** | **NORMWARE** |
| **physical device** | **symbolic device** | **coordination device** |
| when running<br>⇒ **physical process** | when running<br>⇒ **symbolic process** | when running<br>⇒ **coordination process** |
| situated in a<br>**physical environment** | relies on<br>**physical processes** | relies on **symbolic** (possibly<br>hard-coded) **processes** |

# Relevant research directions for normware?

- **Artefact level**: going up in the abstraction ladder of specification languages…

**Imperative programming**

**Declarative programming**

**Policy-based programming**

desires/preferences as individual policies

norms as collective policies

***Normative specifications***

***Agent-based Programming***

# Relevant research directions for normware?

- **Artefact level**: going up in the abstraction ladder of specification languages…

**Imperative programming**

**Declarative programming**

**Policy-based programming**

desires/preferences as individual policies

norms as collective policies

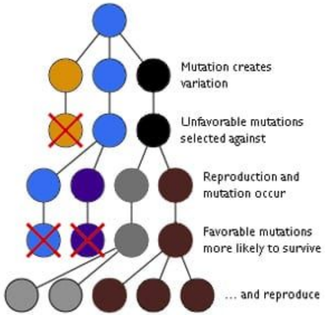
***Normative specifications***

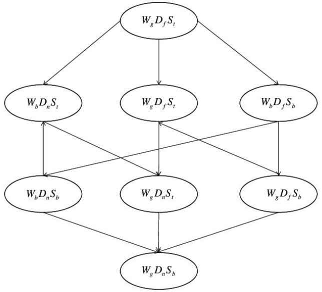
***Agent-based Programming***

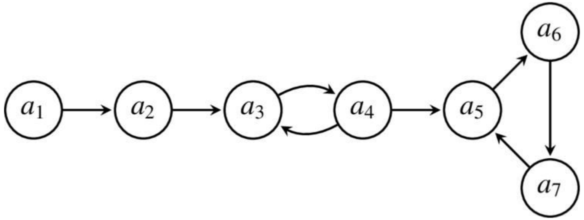
*from algorithms to governance of algorithms…*

# Relevant research directions for normware?

- **Process level:** better understanding/application of mechanisms of resolution, eg.
    - *preferential aggregation*
    - *voting systems (computational social theory)*
    - *formal argumentation frameworks*
    - *evolutionary algorithms*
    - *…*


Mutation creates variation
Unfavorable mutations selected against
Reproduction and mutation occur
Favorable mutations more likely to survive
... and reproduce





# Part II:
# specifying normware

Sileno, G., Van Binsbergen, T., Pascucci, M., van Engers, T., DPCL: a language template for normative specifications,
Workshop on Programming Languages and the Law (ProLaLa 2022), co-located with POPL 2022.

ok, we want to represent policies (normative directives), but how?

ok, we want to represent policies (normative directives), but how?

1. do we need normative concepts?
2. if yes, which normative concepts do we need?
3. what do they “mean”?

# 1. do we need normative concepts in IT?

programs in themselves
are mandatory in nature

# 1. do we need normative concepts in IT?

programs in themselves
are mandatory in nature

| | |
|---|---|
| **`a := 2 + 2`** | system ***has*** to perform 2 + 2… |
| **`?mother(maggie, bart)`** | system ***has*** to prove that… |
| **`animal :- dog.`** | system ***has*** to make animal true if dog is true |

# 1. do we need normative concepts in IT?

programs in themselves
are mandatory in nature

**PERFORMANCE**
**is expected**

→ ***the system does what we tell it to do***

# 1. do we need normative concepts in IT?

programs in themselves are mandatory in nature

**PERFORMANCE is expected**

vs **FAILURE is expected**

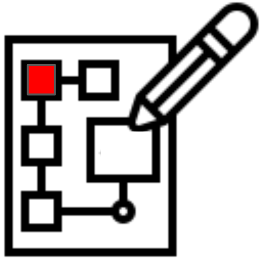
# 1. do we need normative concepts in IT?

programs in themselves
are mandatory in nature

**PERFORMANCE**
**is expected**

**vs FAILURE is expected**

**VIOLATION**
certain components
may not perform
as required



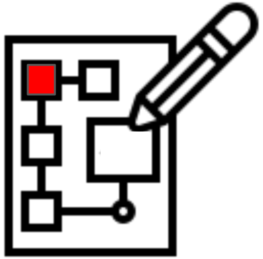
# 1. do we need normative concepts in IT?

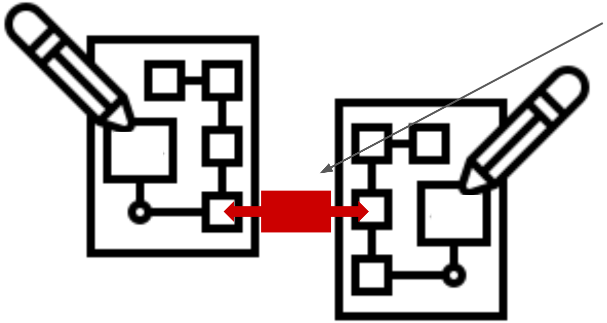
programs in themselves
are mandatory in nature

**PERFORMANCE**
**is expected**

**vs FAILURE is expected**

**VIOLATION**
certain components
may not perform
as required





**CONFLICT**
concurrent
components
may have
incompatible
requests

# 1. do we need normative concepts i

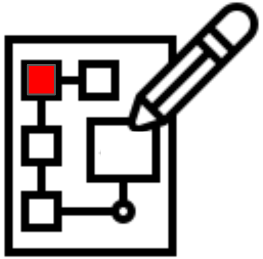
**CENTRAL PROBLEM: who will declare that there is/was indeed a failure?**

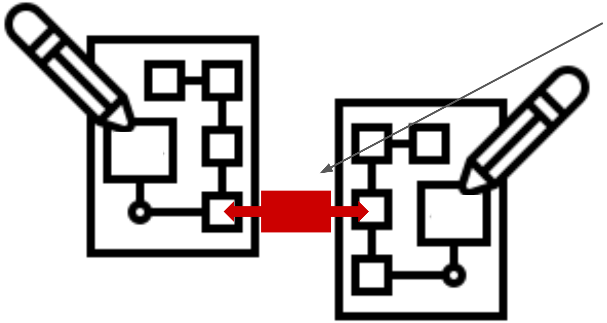
programs in themselves are mandatory in nature

**PERFORMANCE is expected**

vs **FAILURE is expected**

**VIOLATION**
certain components may not perform as required

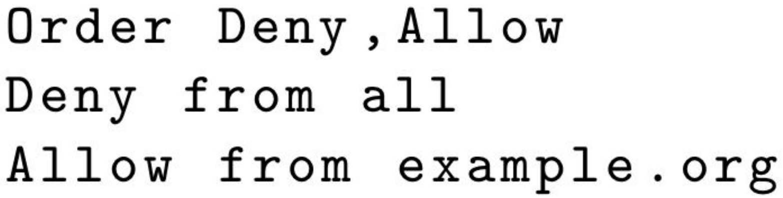




**CONFLICT**
concurrent components may have incompatible requests

# 2. which normative concepts do we need?

- Control models (e.g. access or usage control)



```
Order Deny,Allow
Deny from all
Allow from example.org
```

*example from Apache webserver configuration*

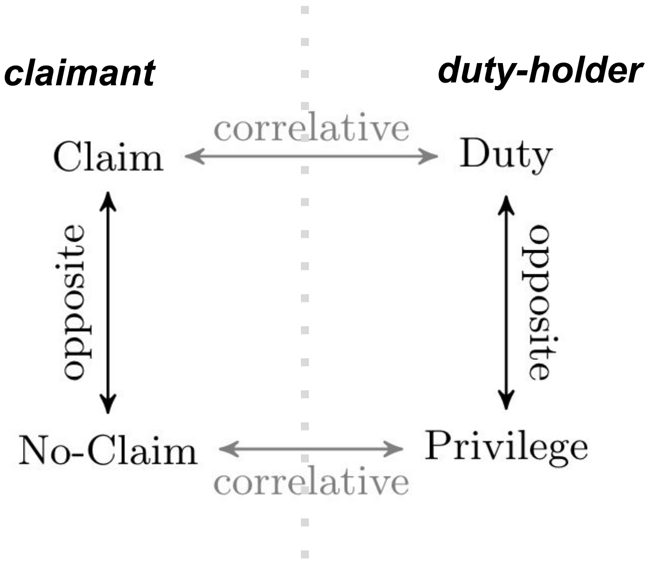
# 2. which normative concepts do we need?

- Deontic logic(s)

$$\begin{tikzcd}[column sep=huge, row sep=huge]
O\phi \arrow[r, leftrightarrow, "\text{contrary}"] \arrow[d, "\text{subaltern}"'] \arrow[dr, leftrightarrow] & F\phi \arrow[d, "\text{subaltern}"] \arrow[dl, leftrightarrow, "\text{contradictory}" description] \\
P\phi \arrow[r, leftrightarrow, "\text{subcontrary}"'] & P\neg\phi
\end{tikzcd}$$

# 2. which normative concepts do we need?

- Hohfeld’s (based on Salmond’s) normative relationships

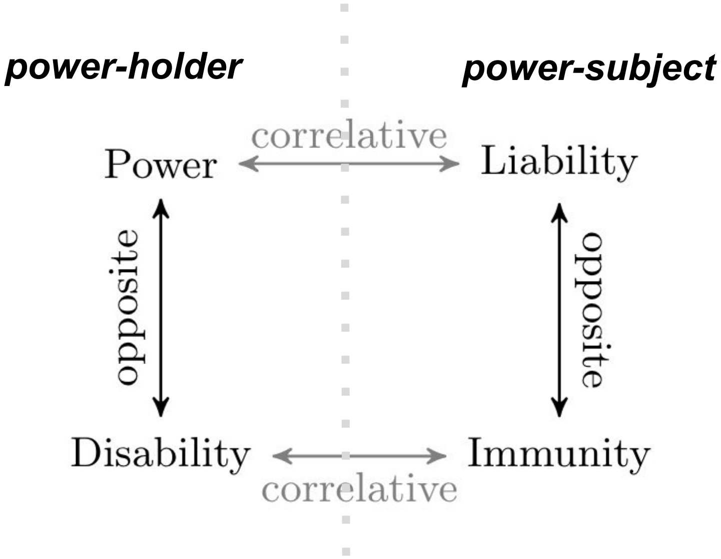


***claimant*** ***duty-holder***

Claim correlative Duty

opposite opposite

No-Claim correlative Privilege



***power-holder*** ***power-subject***

Power correlative Liability

opposite opposite

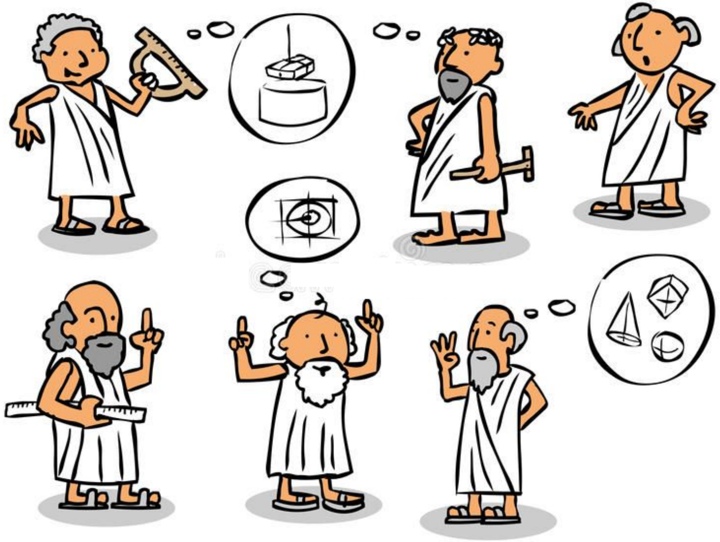
Disability correlative Immunity

# 2. which normative concepts do we need?

| | Control models | Deontic Logic(s) | Hohfeld's framework |
|---|---|---|---|
| permission | X | X | X (as liberty) |
| prohibition | X | X | X (as duty not) |
| obligation | | **X** | X (as duty) |
| power/ability | | | **X** |
| | 1 party | 1 party | **2 parties** |
| *focus on* | **actions** | **situations** | **actions** |

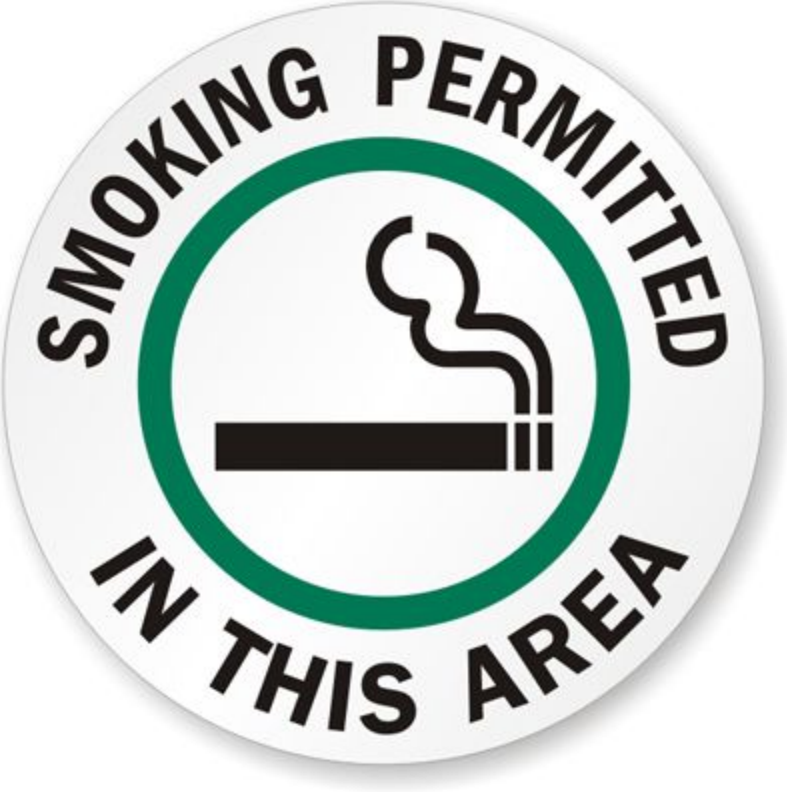
# 3. what normative concepts “mean”?

- long-standing debate
- no shared agreement
- new semantics continuously released



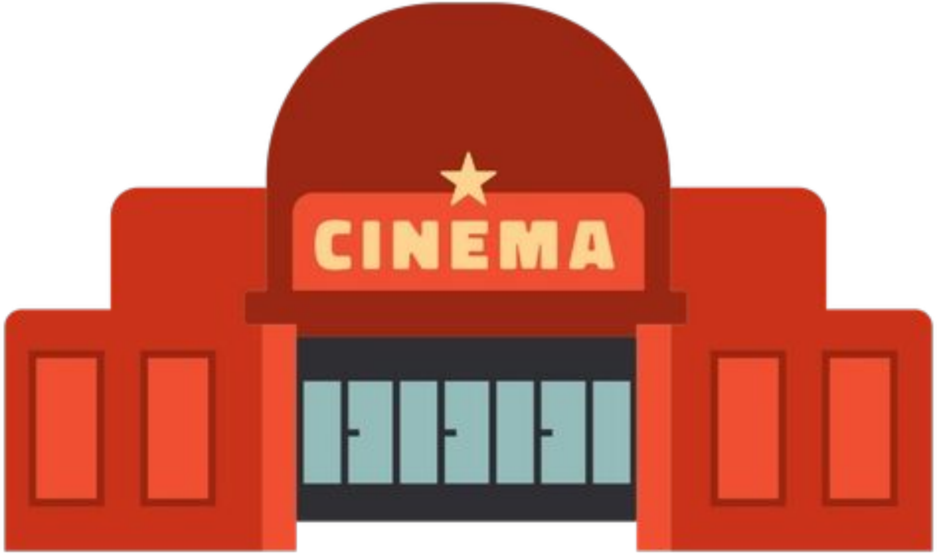
## Example 1

- You are permitted to smoke.



# Example 2

- You have to pay to see the film.



## ok, we want to represent policies (normative directives), but how?

expecting performance vs expecting failures (violations and conflicts)

1. **do we need normative concepts?**
2. **if yes, which normative concepts do we need?**

control models vs deontic logics
vs hohfeldian relationships

3. **what do they “mean”?**

…long-standing debate. no shared agreement.

# ok, we want to represent policies (normative directives), but how?

expecting performance vs expecting failures (violations and conflicts)

1. do we need normative concepts?
2. if yes, which normative concepts do we need?

control models vs deontic logics
vs hohfeldian relationships

3. what do they "mean"?

…long-standing debate. no shared agreement.

4. how to **specify** normative directives?

# Success story: ODRL (Open Digital Rights Language)

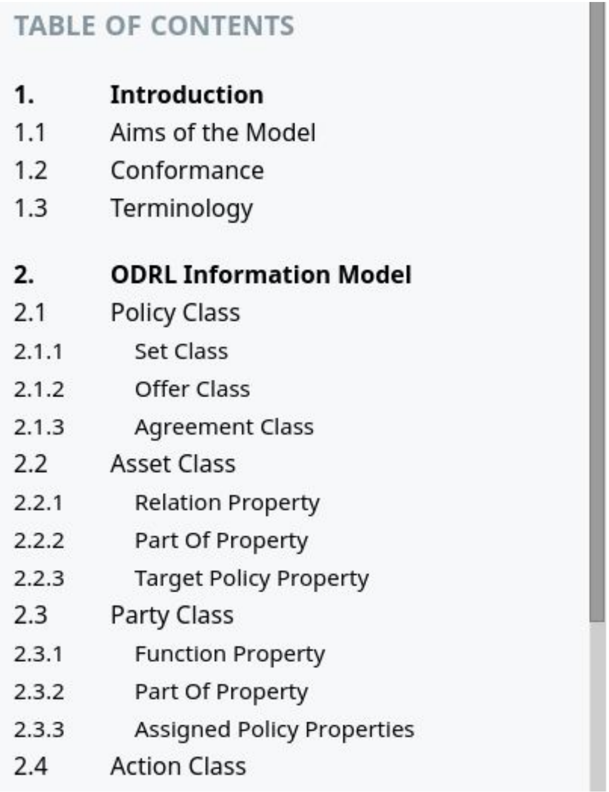
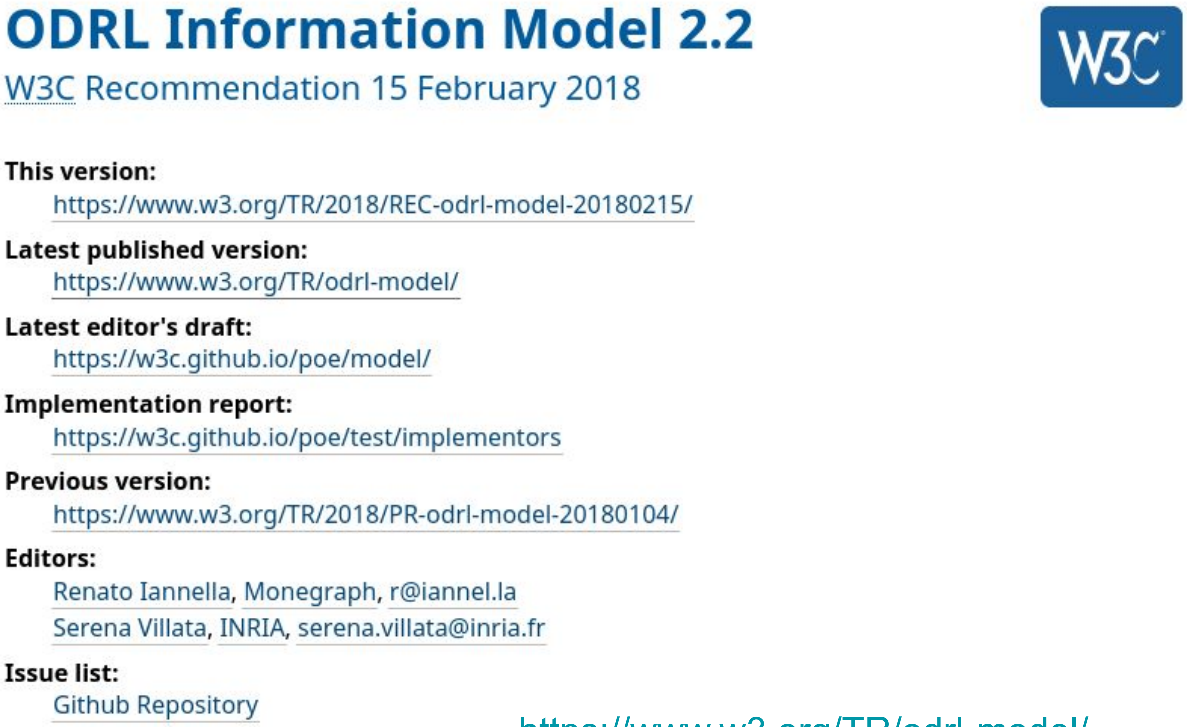


TABLE OF CONTENTS



## ODRL Information Model 2.2

W3C Recommendation 15 February 2018

**This version:**
https://www.w3.org/TR/2018/REC-odrl-model-20180215/

**Latest published version:**
https://www.w3.org/TR/odrl-model/

**Latest editor's draft:**
https://w3c.github.io/poe/model/

**Implementation report:**
https://w3c.github.io/poe/test/implementors

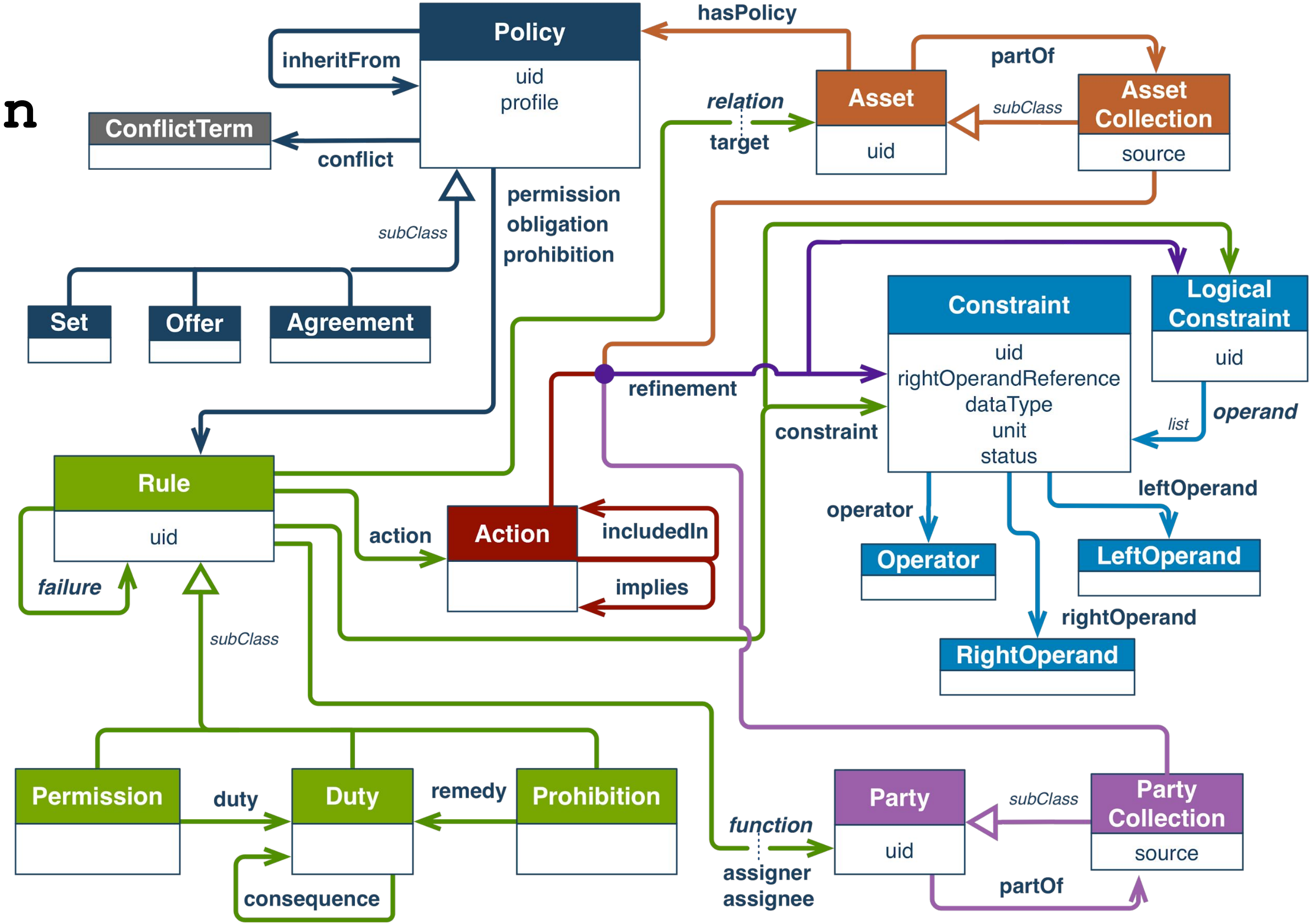
**Previous version:**
https://www.w3.org/TR/2018/PR-odrl-model-20180104/

**Editors:**
Renato Iannella, Monegraph, r@iannel.la
Serena Villata, INRIA, serena.villata@inria.fr

**Issue list:**
Github Repository

https://www.w3.org/TR/odrl-model/

# ODRL Information Model



primacy
to deontic
categories

# ODRL example

```
{
  "@context": "http://www.w3.org/ns/odrl.jsonld",
  "@type": "Offer",
  "uid": "http://example.com/policy:4444",
  "profile": "http://example.com/odrl:profile:11",
  "permission": [{
    "assigner": "http://example.com/org88",
    "target": {
      "@type": "AssetCollection",
      "source":  "http://example.com/media-catalogue",
      "refinement": [{
        "leftOperand": "runningTime",
        "operator": "lt",
        "rightOperand": { "@value": "60", "@type": "xsd:integer" },
        "unit": "http://qudt.org/vocab/unit/MinuteTime"
      }]
    },
    "action": "play"
  }]
}
```

json
data
structure

roughly: permission to org88 to play assets in collection with running length < 60 min

# ODRL example

```
{
  "@context": "http://www.w3.org/ns/odrl.jsonld",
  "@type": "Offer",
  "uid": "http://example.com/policy:4444",
  "profile": "http://example.com/odrl:profile:11",
  "permission": [{
    "assigner": "http://example.com/org88",
    "target": {
      "@type": "AssetCollection",
      "source":  "http://example.com/media-catalogue",
      "refinement": [{
        "leftOperand": "runningTime",
        "operator": "lt",
        "rightOperand": { "@value": "60", "@type": "xsd:integer" },
        "unit": "http://qudt.org/vocab/unit/MinuteTime"
      }]
  }]
}
```

json data structure

**almost any IT practitionner is able to read through it**

roughly: permission to org88 to play assets in collection with running length < 60 min

# Our modeling playground in a nutshell

- **JSON-like syntax**
- foundational ontology
  - **objects** vs **events**
  - **transformational rules** vs **reactive rules**
- normative concepts from **Hohfeld's framework**

named **~~DPCL~~ DCPL**

*Duty, Claim, Power, Liability* or
*Digital Contracts Programming Language*

https://github.com/gsileno/DCPLschema

Sileno, G., van Binsbergen, T., Pascucci, M., van Engers, T., *DPCL: a Language Template for Normative Specifications*, Workshop on Programming Languages and the Law (ProLaLa 2022), co-located with POPL 2022 https://arxiv.org/abs/2201.04477

# **DCPL: basic entities**

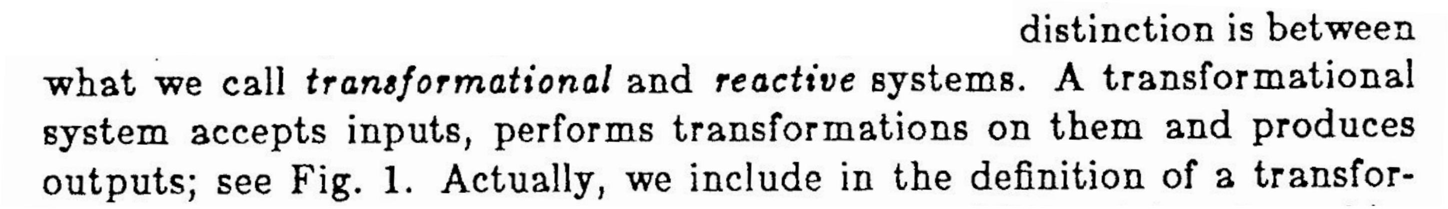
We follow the common-sensical distinction:

- states: **`condition, object, agent`**
- (transition) events:
  - primitive events: **`#action`**
  - production/removal events: **`+object`**, **`-object`**
  - qualification/disqualification events: **`object in group`**, ...

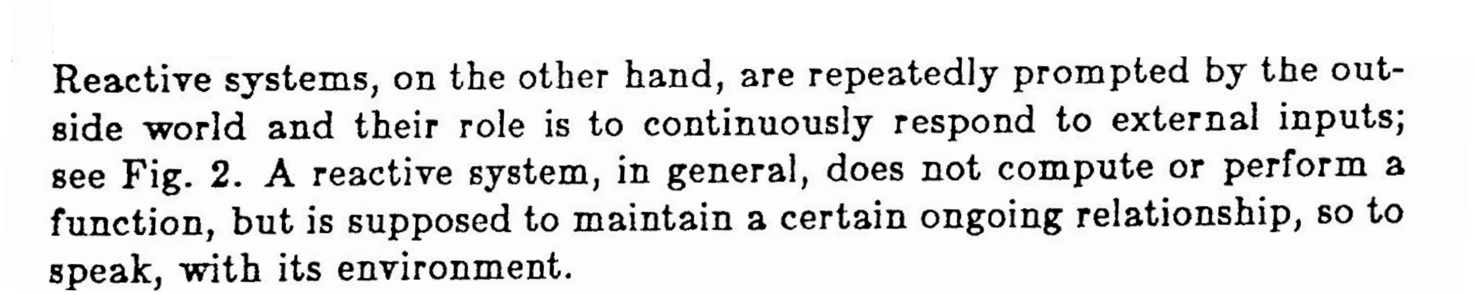
this is confirmed in legal core ontologies like

# DCPL: conditioning rules

- **Transformational vs reactive systems**



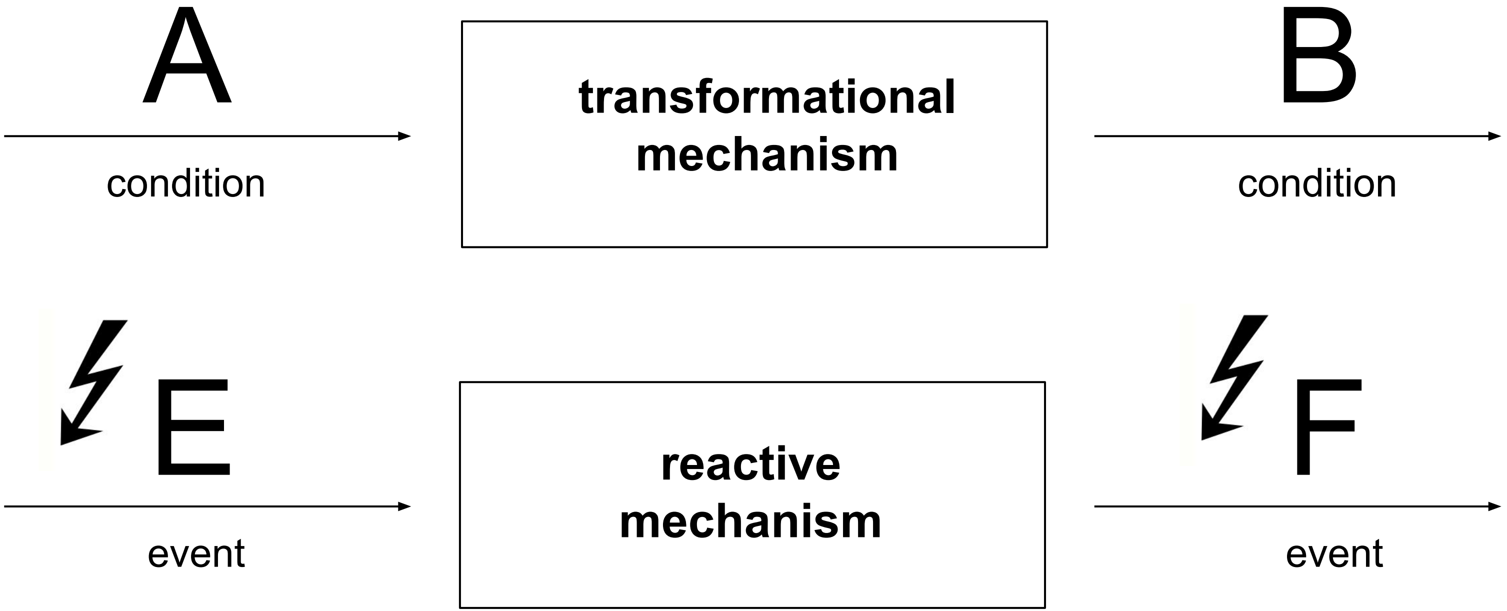
distinction is between what we call *transformational* and *reactive* systems. A transformational system accepts inputs, performs transformations on them and produces outputs; see Fig. 1. Actually, we include in the definition of a transfor-



Reactive systems, on the other hand, are repeatedly prompted by the outside world and their role is to continuously respond to external inputs; see Fig. 2. A reactive system, in general, does not compute or perform a function, but is supposed to maintain a certain ongoing relationship, so to speak, with its environment.

Harel, D., & Pnueli, A. (1985). On the development of reactive systems. Logics and Models of Concurrent Systems, 477–498.

# Transformational vs reactive



A

condition

transformational mechanism

B

condition

E

event

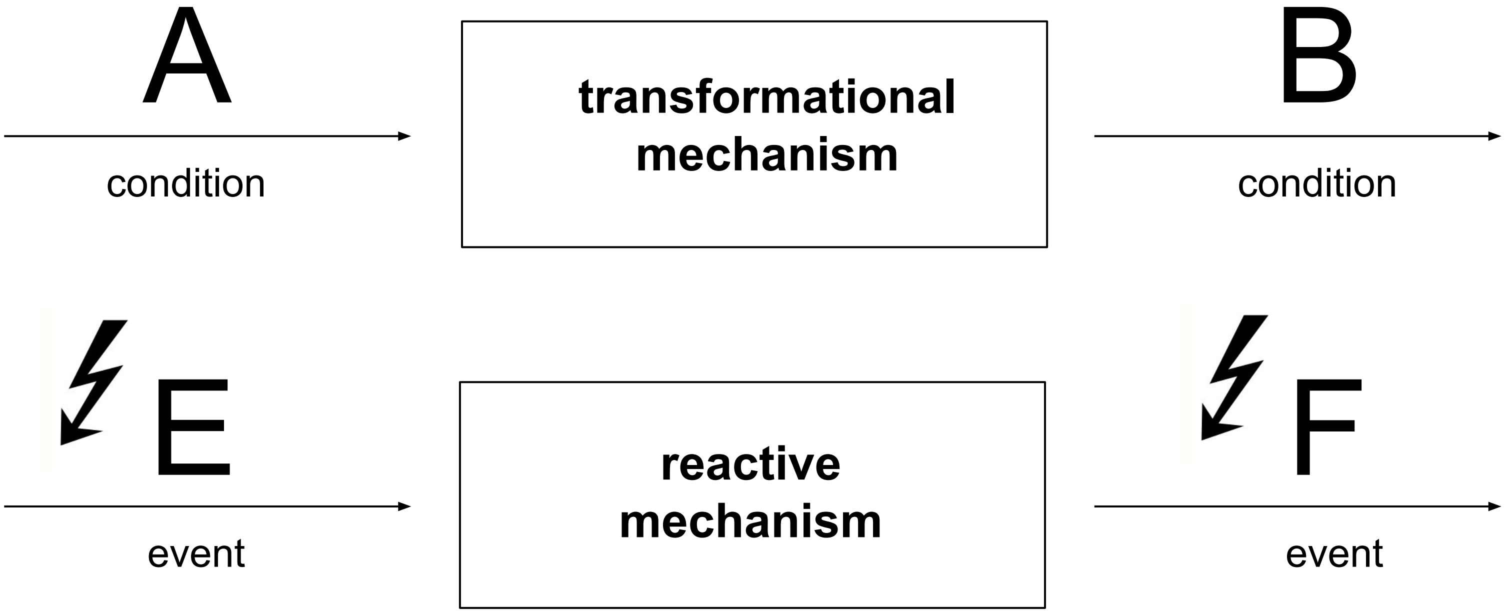
reactive mechanism

F

event

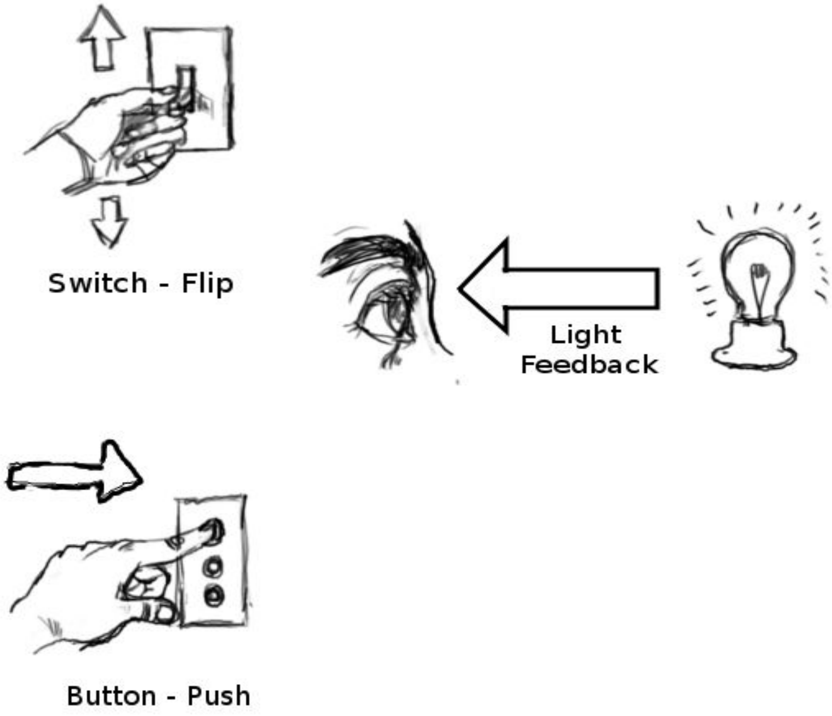
# Transformational vs reactive

*coupling input with output*

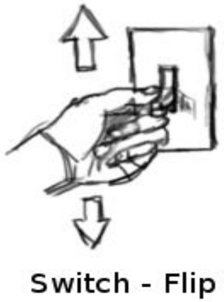


*decoupling input from output*

# Transformational or reactive?

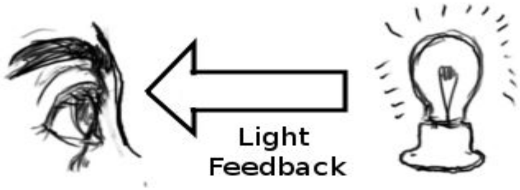


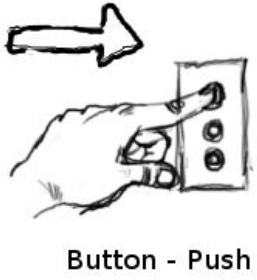
# Transformational or reactive?



### TRANSFORMATIONAL

If the switch is in top position, the light is on.

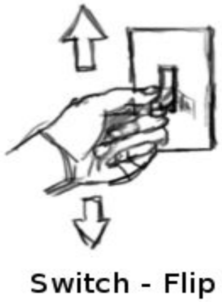




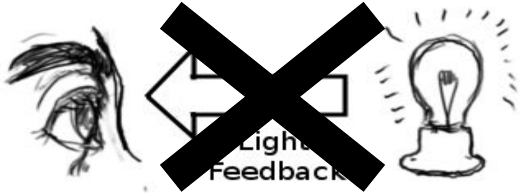
### REACTIVE

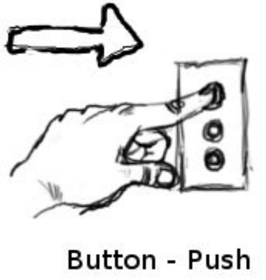
If the button is pressed, the light changes of state.

# On enabling/disabling conditions



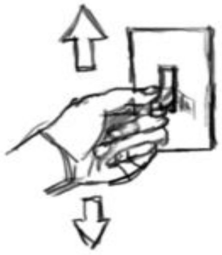
POWER
OUTAGE



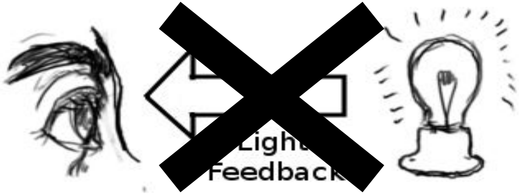


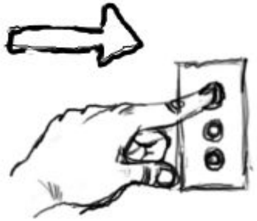
**NO ELECTRICITY → NO LIGHT**

# On enabling/disabling conditions



Switch - Flip



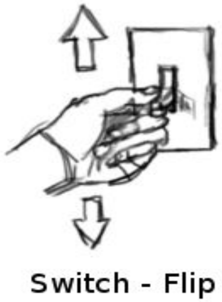


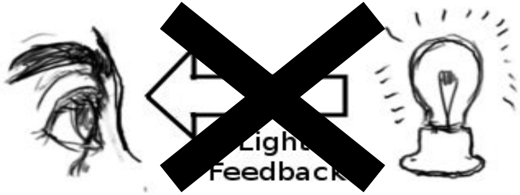
Button - Push

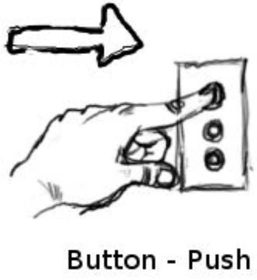
**NO ELECTRICITY** → **NO LIGHT**

**LIGHT** functioning → **ELECTRICITY** holding

# On enabling/disabling conditions







**Transformational (meta-)mechanism**

**NO ELECTRICITY** → **NO LIGHT**

**LIGHT** functioning → **ELECTRICITY** holding

# Institutional mechanisms are just the same

Bikes count as vehicles.

Raising a hand counts as a bid.

# Institutional mechanisms are just the same

Bikes count as vehicles.

*If (as long as) an object is a bike,*
*then that object is deemed a vehicle.*

Raising a hand counts as a bid.

*If you raise a hand,*
*you create a bid.*

# Institutional mechanisms are just the same

Bikes count as vehicles.

*If (as long as) an object is a bike, then that object is deemed a vehicle.*

**TRANSFORMATIONAL**

Raising a hand counts as a bid.

*If you raise a hand, you create a bid.*

**REACTIVE**

# Institutional mechanisms are just the same

Bikes count as vehicles.

*If (as long as) an object is a bike, then that object is deemed a vehicle.*

**TRANSFORMATIONAL**

Raising a hand counts as a bid.

*If you raise a hand, you create a bid.*

**REACTIVE**

within the jurisdiction of the parking regulation

within the auction regulation

# DCPL: conditioning rules

- Transformational rules (as long as the premise is true, the conclusion is true):

```
raining -> wet
bike -> vehicle
```

- Reactive rules (when the antecedent occurs, the consequent occurs):

```
#rain => +wet
#raise_hand => +bet
```

# DCPL: conditioning rules

- Transformational rules (as long as the premise is true, the conclusion is true):

```
raining -> wet
bike -> vehicle
```

- Reactive rules (when the antecedent occurs, the consequent occurs):

```
#rain => +wet
#raise_hand => +bet
```

- Contexts are generally involved in transformational rules:

```
auction -> { #raise_hand => +bet }
```

# DCPL: parameters and refinements

Any entity can be refined via some parameter, eg. in the case of actions:

```
#give {
    agent: john
    item: apple
    recipient: paul
}

#eat {
    agent: paul
    item: apple
}
```

# DCPL: power frame

```
power {
    holder: student
    action: #register { instrument: holder.id_card }
    consequence: holder in member
}
```

a power reifies an

**(institutional) causal mechanism**

- conditioned by **qualification** of agent
- conditioned by **procedure** of action
- affecting a limited **domain of competence**

# DCPL: duty frame

```
duty {
    holder: john
    counterparty: university
    action: #teach { recipient: student }
}
```

# DCPL: duty frame

```
duty {
    holder: john
    counterparty: university
    action: #teach { recipient: student }
}
```

a duty reifies an expectation (of “good”) for the counterparty

# DCPL: duty frame

```
duty {
    holder: john
    counterparty: university
    action: #teach { recipient: student }
    violation: john.online is False
}
```

a duty reifies an expectation (of “good”) for the counterparty

*sometimes languages enable violations to be defined independently of the content of the duty*

# DCPL: prohibition frame

*another example of “**semantic neutrality**”**:** not all logics consider the “prohibition to do A” the same as the “obligation of not doing A”*

```
prohibition {
    holder: john
    action: #go { destination: swimming }
}
```

# DCPL: prohibition frame

*another example of “**semantic neutrality**”**:** not all logics consider the “prohibition to do A” the same as the “obligation of not doing A”*

```
prohibition {
    holder: john
    action: #go { destination: swimming }
    termination: -winter
}
```

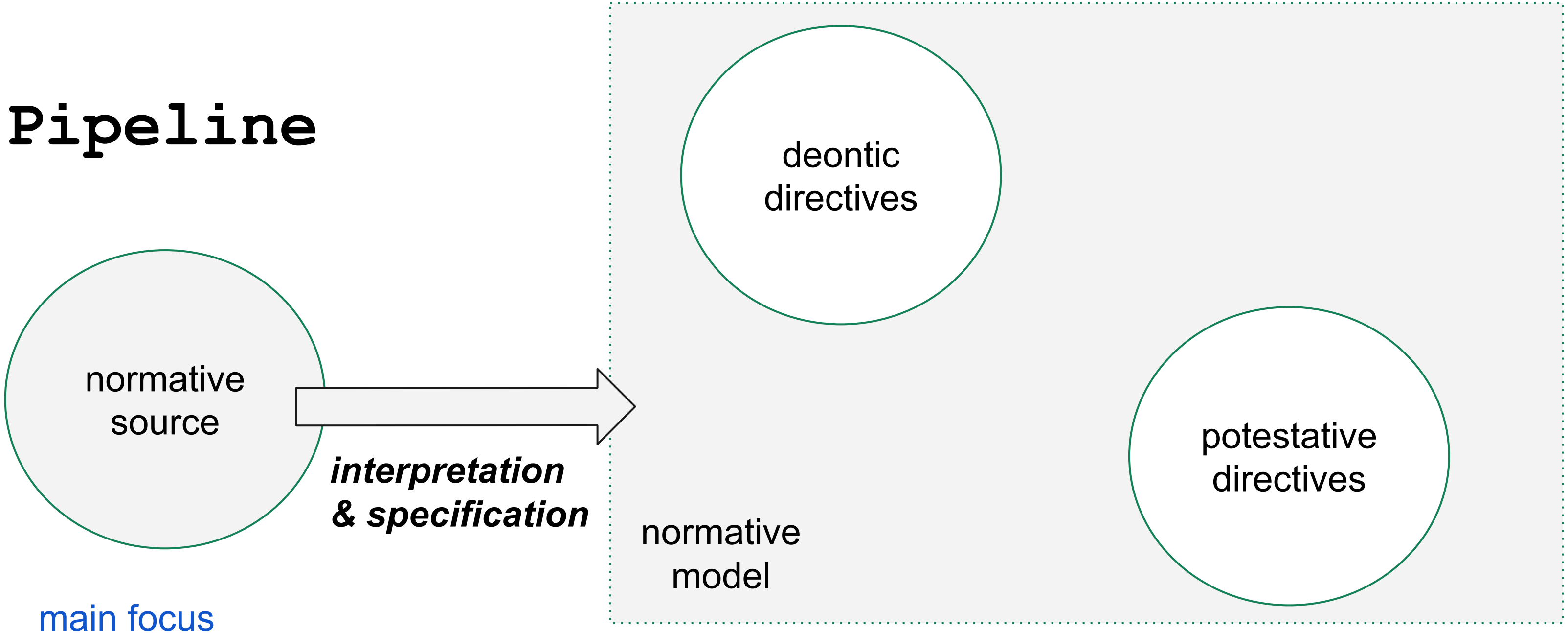
*sometimes normative directives have terminating events independent of performance*

DCPL also provides **`liability`**, **`liberty`**,  … which may not have correspondences to other languages

# Pipeline

normative source

# Pipeline



normative source

*interpretation & specification*

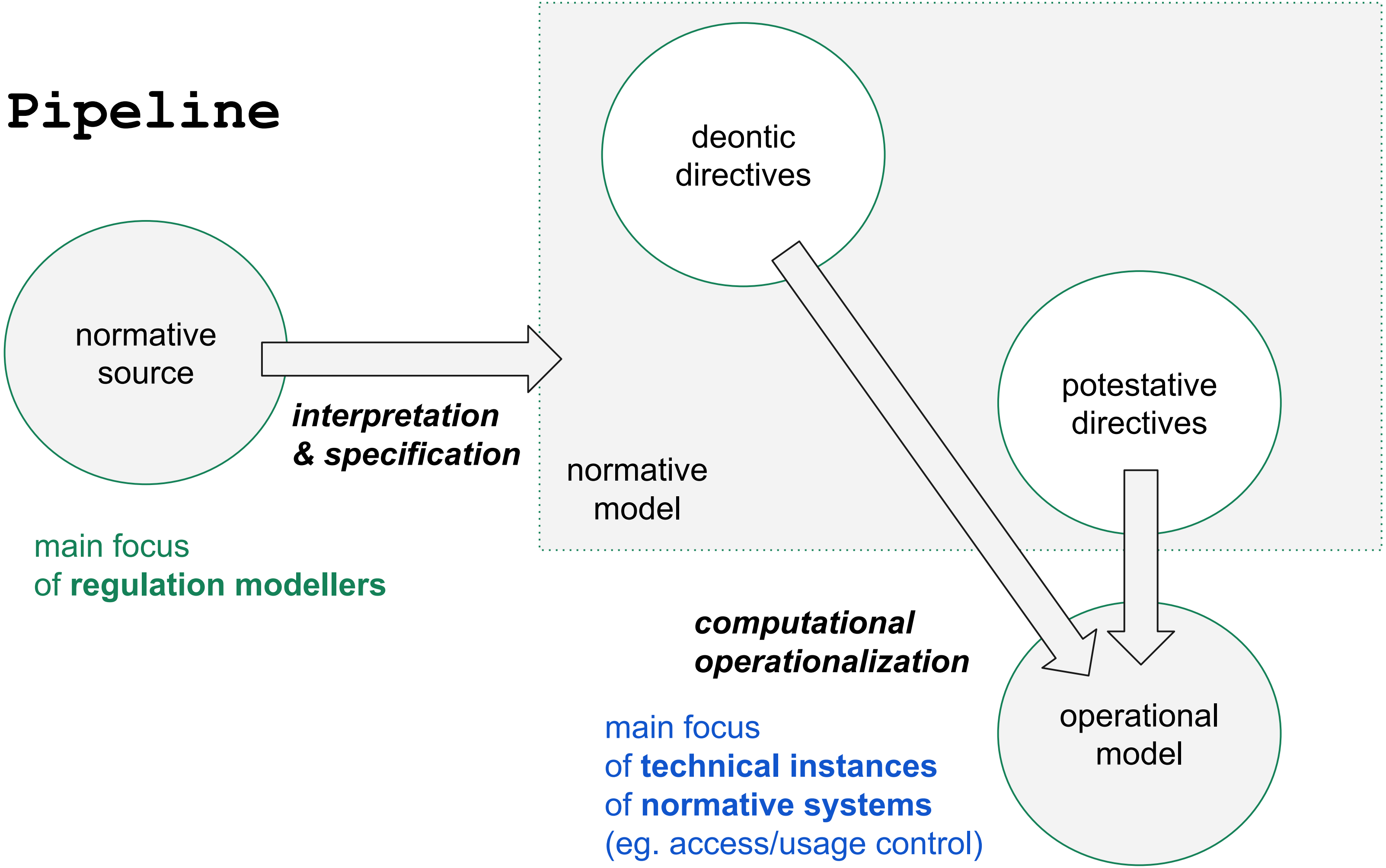
deontic directives

potestative directives

normative model

main focus of **regulation modellers**

# Pipeline



normative source

***interpretation & specification***

deontic directives

potestative directives

normative model

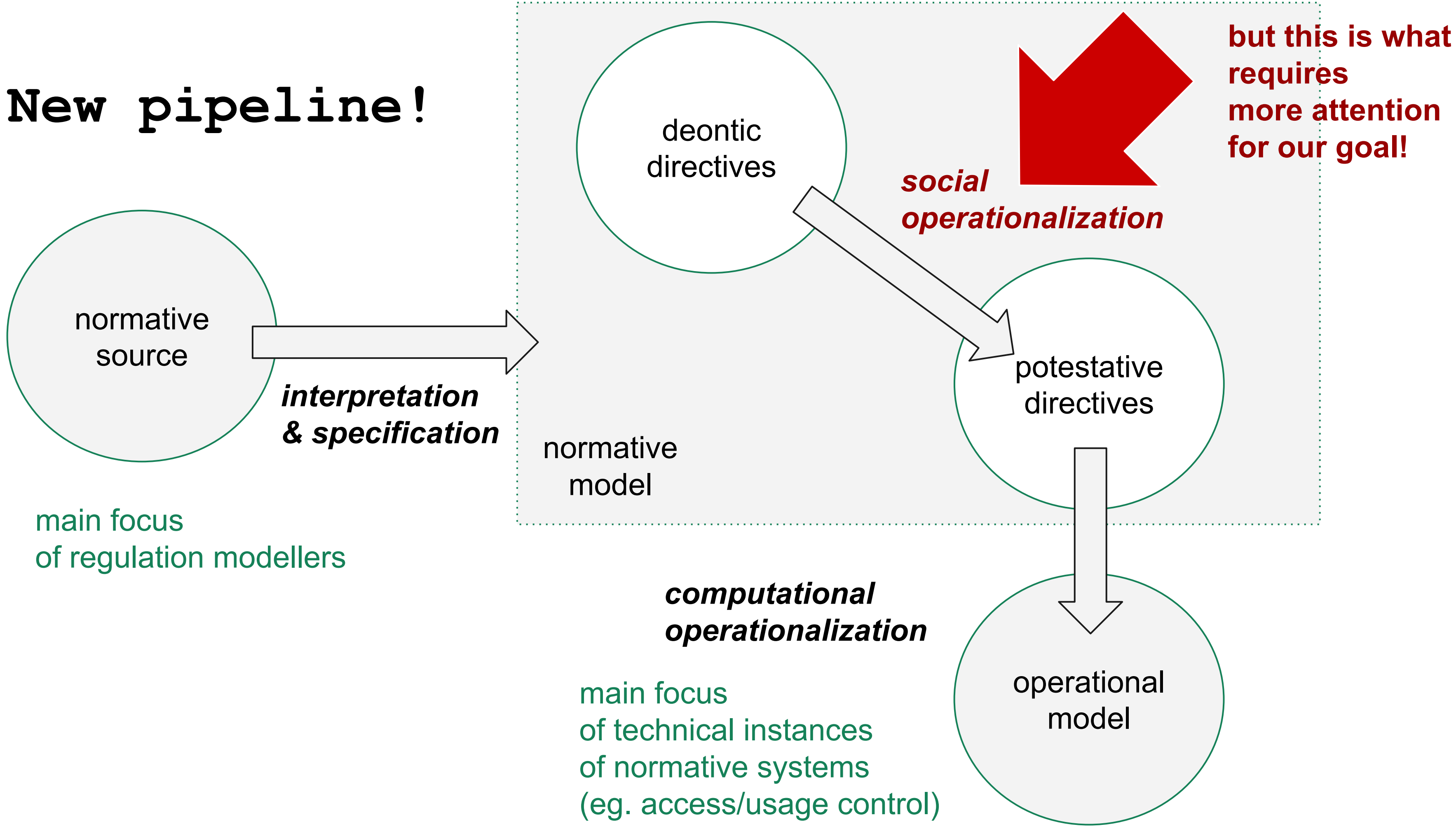
***computational operationalization***

operational model

main focus of **regulation modellers**

main focus of **technical instances** of **normative systems** (eg. access/usage control)

# New pipeline!



normative source

*interpretation & specification*

main focus of regulation modellers

deontic directives

*social operationalization*

potestative directives

normative model

but this is what requires more attention for our goal!

*computational operationalization*

operational model

main focus of technical instances of normative systems (eg. access/usage control)

# Rewriting: all is about power!

- All conditions (e.g. preconditions, violation, termination) implicitly refers to a power that may (should?) be assigned to someone.
- This is an actual step in **policy operationalization** in administrative settings.

# Rewriting: all is about power!

- Unfolding a violation construct to the power to declare that violation…

```
prohibition p {
   action: #smoke
}
```

```
              p -> {
                 #smoke => +power {
                    holder: *
                    action: #declare_violation { item: p }
                    consequence: +p.violated
                 }
              }
```

# Rewriting: all is about power!

- More in general any duty comes with two powers: one to declare fulfilment, another one to declare violation.

```
duty d {
    holder: john
    counterparty: paul
    action: #pay
    violation: timeout
}
```

# Rewriting: all is about power!

- More in general any duty comes with two powers: one to declare fulfilment, another one to declare violation.

```
duty d {
    holder: john
    counterparty: paul
    action: #pay
    violation: timeout
}
```

*here we assign these powers to the counterparty, the claimant*

```
d -> {
     john.#pay => +power {
          holder: paul
          action: #declare_fulfillment { item: d }
          consequence: +d.fulfilled
     }
     timeout => +power {
          holder: paul
          action: #declare_violation { item: d }
          consequence: +d.violated
     }
}
```

# Rewriting: rules as duties & powers

- Transformational rules can be seen not only as “epistemic” duties (about producing knowledge), but also as powers!

```
bike -> vehicle
```

⇨

```
bike -> {
    duty {
        holder: *
        action: +vehicle
    }
    power {
        holder: *
        action: #state { item: vehicle }
        consequence: +vehicle
    }
}
```

mandatory view

ability view

# Rewriting: rules as duties & powers

- Transformational rules can be seen not only as “epistemic” duties (about producing knowledge), but also as powers!

```
bike -> vehicle
```

⇨

```
bike -> {
    duty {
        holder: *
        action: +vehicle
    }
    power {
        holder: *
        action: #state { item: vehicle }
        consequence: +vehicle
    }
}
```

mandatory view

LESS IMPORTANT IN A SOCIAL COORDINATION SETTING!

ability view

# Rewriting: maintenance duties

- Unfolding maintenance duties (about states of affairs) in terms of duties of actions

**maintenance duty**

```
duty d1 {
    target: g1
}
```

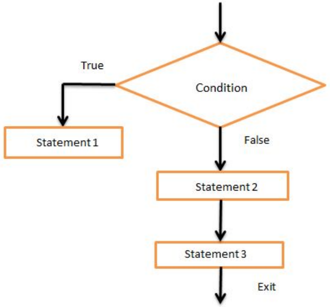
⇒

**achievement duty**

```
d1 -> {
    ~g1 -> duty { action: +g1 }
    g1 -> prohibition { action: -g1 }
}
```

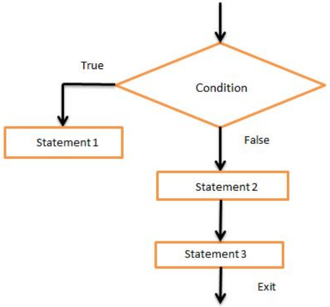
**avoidance duty**

# A rather unexplored dimension?



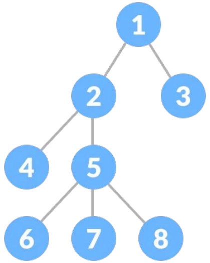
CONTROL
FLOW

# A rather unexplored dimension?

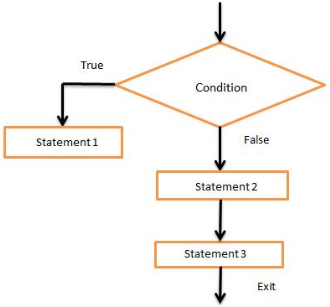


CONTROL
FLOW

DATA
STRUCTURE

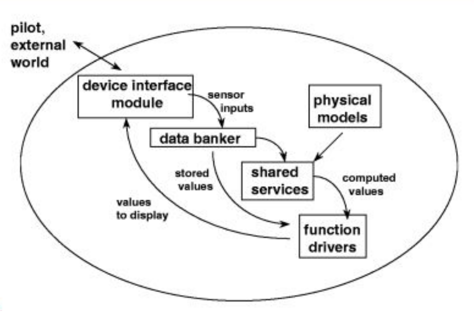


# A rather unexplored dimension?

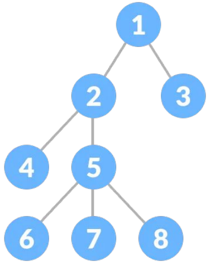


CONTROL FLOW

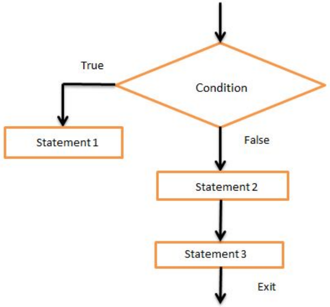
DATA FLOW



DATA STRUCTURE

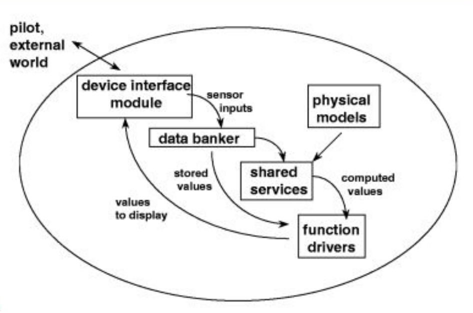


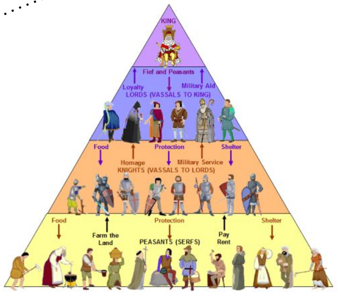
# A rather unexplored dimension?



CONTROL FLOW

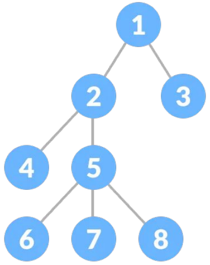
DATA FLOW





CONTROL STRUCTURE

DATA STRUCTURE



*this is the domain of normware: **roles, power relationships, interventions*** points!

# Part III:
# An application: unfolding the Responsible Internet proposal

Sileno, G., Grosso, P.,, Accounting Value Effects for Responsible Networking Proceedings of ACM SIGCOMM 2021 Workshop on Technologies, Applications, and Uses of a Responsible Internet (TAURIN2021).

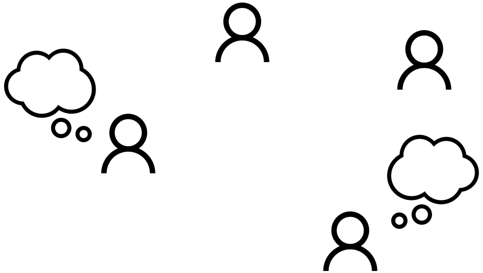
# Data-sharing has practical effects

because having access to relevant information has value for agents!

# Data-sharing has practical effects!

because having access to relevant information has value for agents!

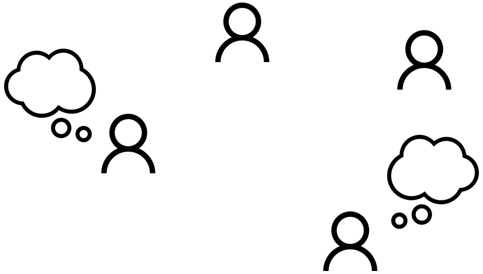
**people, organizations, systems which act to achieve certain purposes**



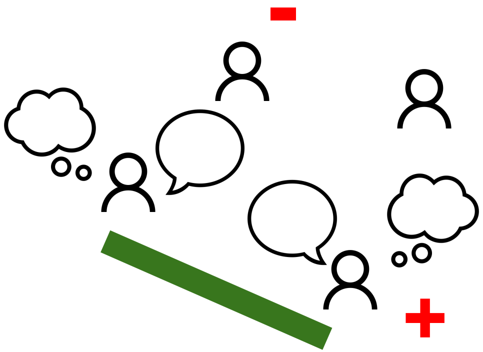
no exchange

# Data-sharing has practical effects!

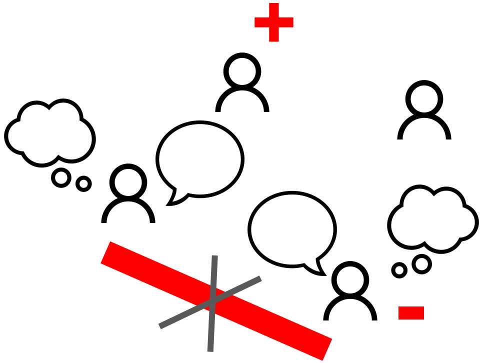
because having access to relevant information has value for agents!



no exchange



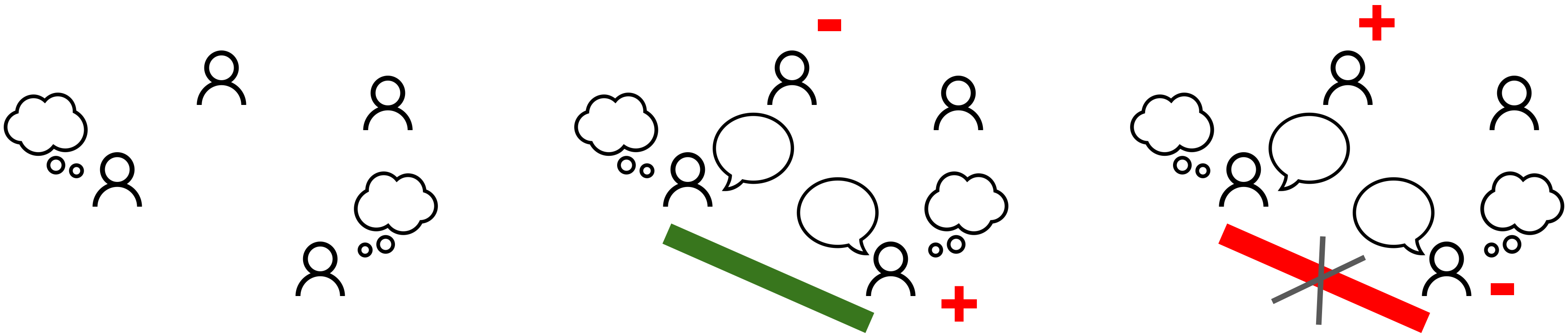
exchange enabled/allowed



exchange disabled/disallowed

# Data-sharing has practical effects!

because having access to relevant information has value for agents!



no exchange

exchange enabled/allowed

exchange disabled/disallowed

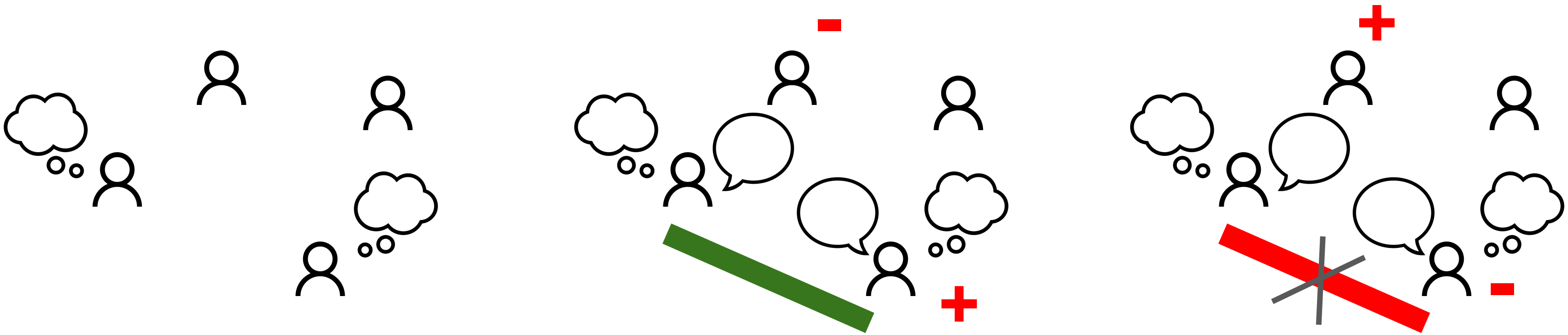
- technologies provide new abilities

**e.g. Internet with new forms of knowledge sharing, aggregation, making business, etc.**

**ability** (power)
dimension

# Data-sharing has practical effects!

because having access to relevant information has value for agents!



no exchange

exchange enabled/allowed

exchange disabled/disallowed

- societies introduce checks & balances

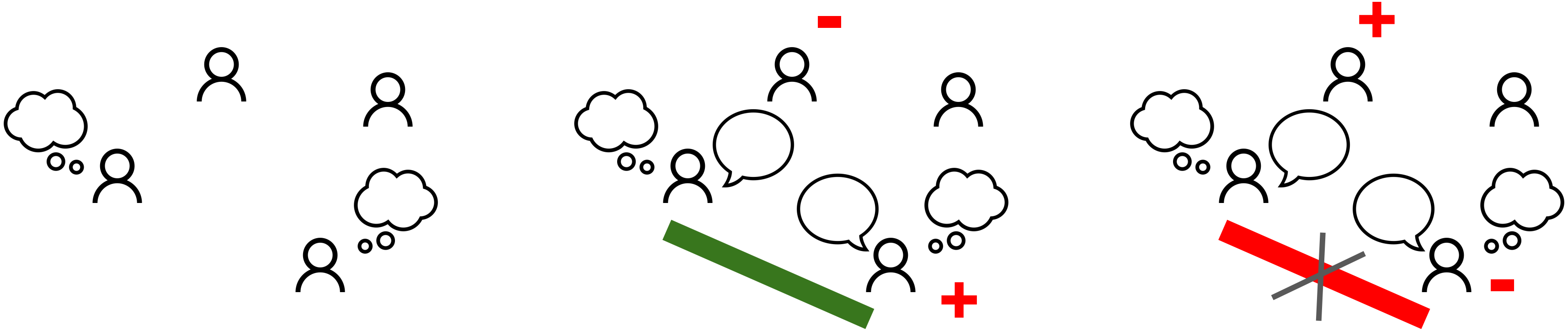
➡ **e.g. norms on privacy, data regulation (e.g. GDPR), competition laws, etc.**

ability (power) dimension

permission dimension

# Data-sharing has practical effects!

because having access to relevant information has value for agents!



no exchange

exchange enabled/allowed

exchange disabled/disallowed

- societies introduce checks & balances

**e.g. norms on privacy, data regulation (e.g. GDPR), competition laws, etc.**

***How these checks and balances are reflected at infrastructural level?***

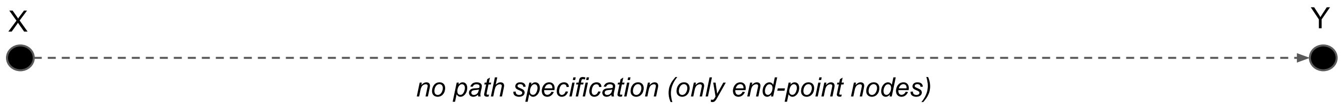
# Data transmission as “logistic” task

How to transport data from node X to node Y?

*no path specification (only end-point nodes)*

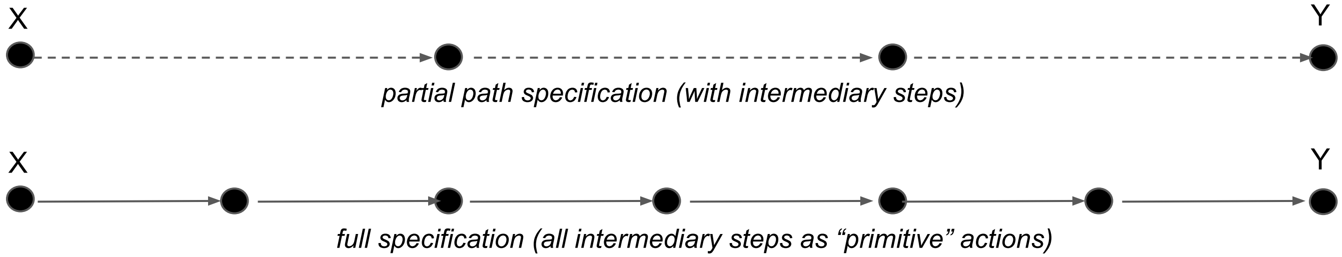
# Data transmission as "logistic" task

How to transport data from node X to node Y?



*no path specification (only end-point nodes)*

- Enabling transmission from X to Y requires the network to provide some form of **routing services.**

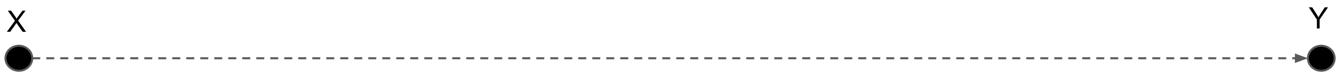


*partial path specification (with intermediary steps)*

*full specification (all intermediary steps as "primitive" actions)*

# Data transmission as "logistic" task

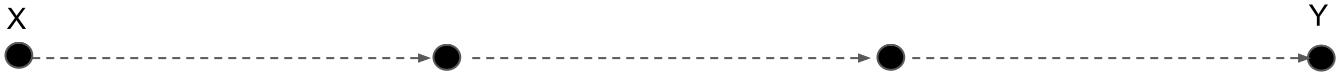
How to transport data from node X to node Y?



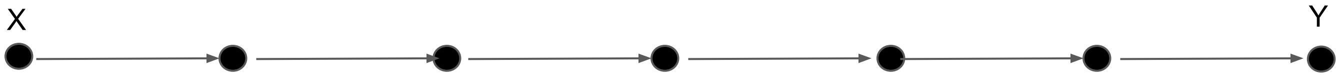
*no path specification (only end-point nodes)*

- Enabling transmission from X to Y requires the network to provide some form of **routing services.**

For ***inter-domain routing***, network operators typically rely on BGP policies and community tags.



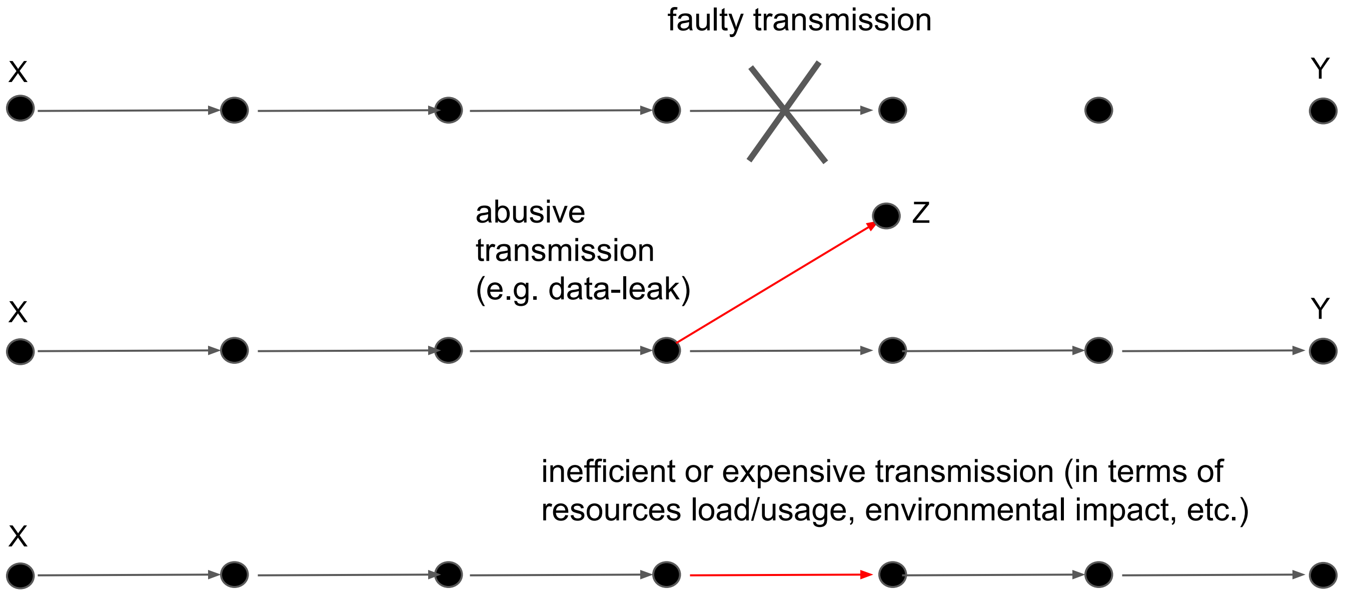
*partial path specification (with intermediary steps)*



*full specification (all intermediary steps as "primitive" actions)*

# Data transmission as “logistic” task

Main issues possibly occurring at network level:

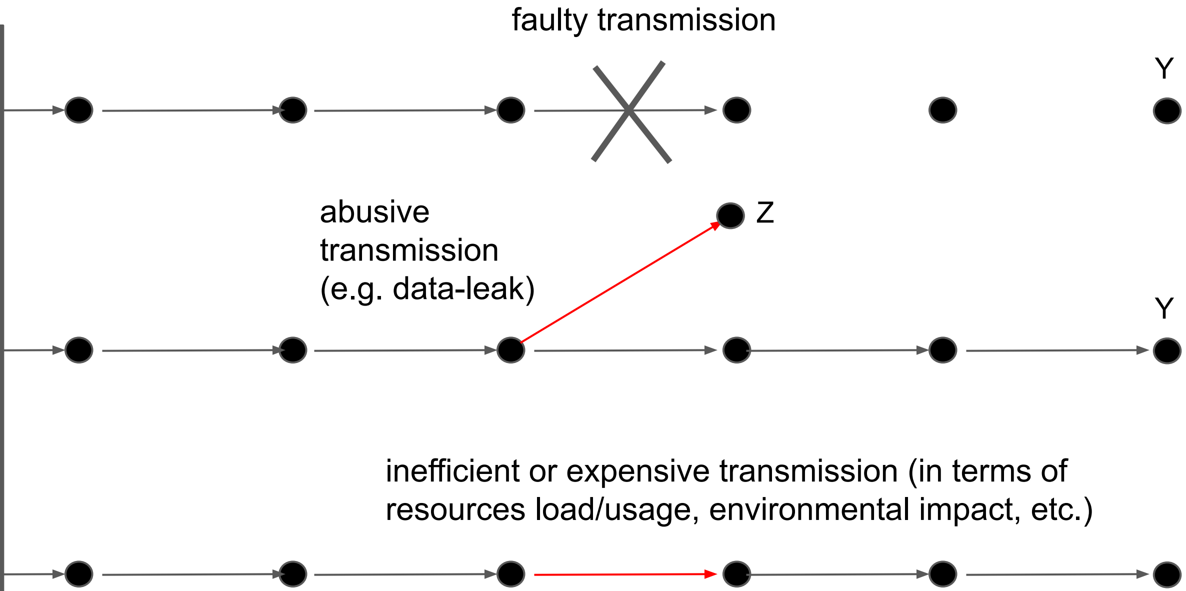

faulty transmission
X
Y
abusive
transmission
(e.g. data-leak)
Z
X
Y
inefficient or expensive transmission (in terms of
resources load/usage, environmental impact, etc.)
X

# Data transmission as “logistic” task

Main issues possibly occurring at network level:

Intuitively, a ‘responsible’ networking should reduce these issues.

but...

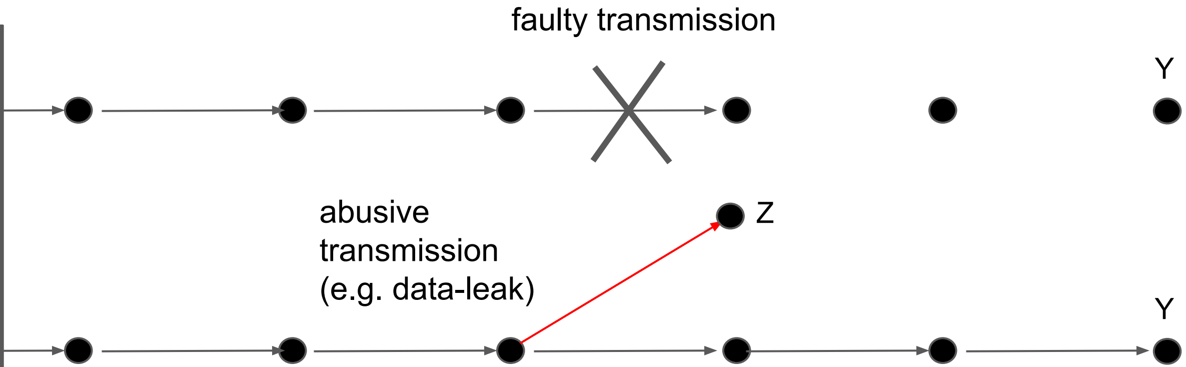

faulty transmission
Y
abusive transmission (e.g. data-leak)
Z
Y
inefficient or expensive transmission (in terms of resources load/usage, environmental impact, etc.)

# Data transmission as “logistic” task

Main issues possibly occurring at network level:

Intuitively, a ‘responsible’ networking should reduce these issues.

but...



inefficient or expensive transmission (in terms of resources load/usage, environmental impact, etc.)

***who defines what is faulty, abusive, expensive?***

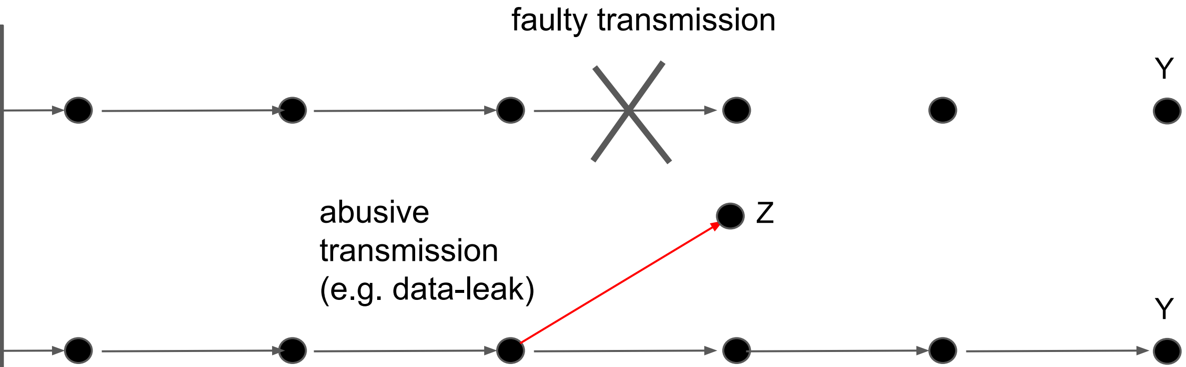
***who monitors? who prevents (predicts) or reacts to failures?***

# Data transmission as “logistic” task

Main issues possibly occurring at network level:

Intuitively, a ‘responsible’ networking should reduce these issues.

but...



inefficient or expensive transmission (in terms of resources load/usage, environmental impact, etc.)

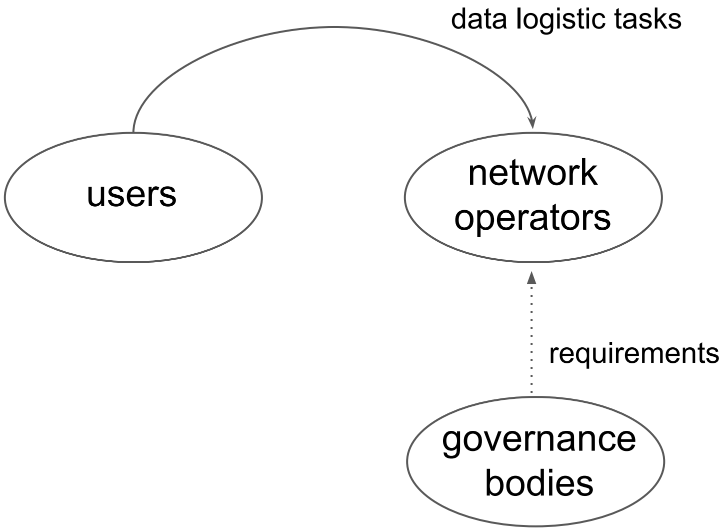
***who defines what is faulty, abusive, expensive?***
***who monitors? who prevents (predicts) or reacts to failures?***

→ “**responsibility**” is a matter of **social coordination** policy

# Internet social structure

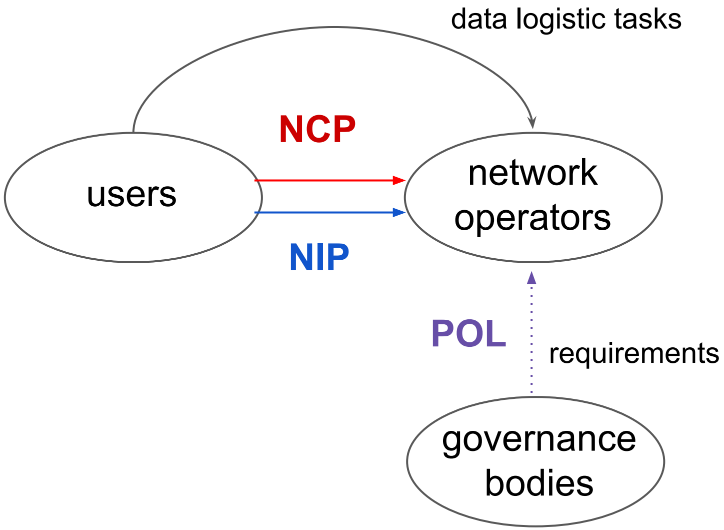
Three main roles can be recognized around Internet’s activities:

- users (applications, software agents, etc.)
- network operators
- governance bodies



# Responsible Internet social structure

The Responsible Internet proposal (Hesselman et al., 2020) essentially envisions to **redistribute** control and monitoring abilities to ***users***, supported by regulations issued by relevant societal stakeholders.



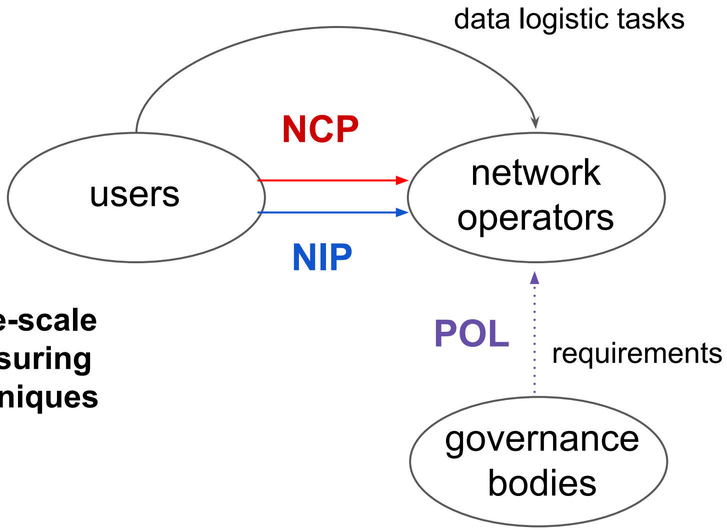
Cristian Hesselman, Paola Grosso, Ralph Holz, Fernando Kuipers, Janet Hui Xue, Mattijs Jonker, Joeri de Ruiter, Anna Sperotto, Roland van Rijswijk-Deij, Giovane C.M. Moura, Aiko Pras, and Cees de Laat. *A Responsible Internet to Increase Trust in the Digital World.* Journal of Network and Systems Management 28, 4 (2020), 882–922.

# Responsible Internet social structure

The Responsible Internet proposal (Hesselman et al., 2020) essentially envisions to **redistribute** control and monitoring abilities to ***users***, supported by regulations issued by relevant societal stakeholders.

**open-source programmable networks**
(e.g. for routing, via BGP policies)

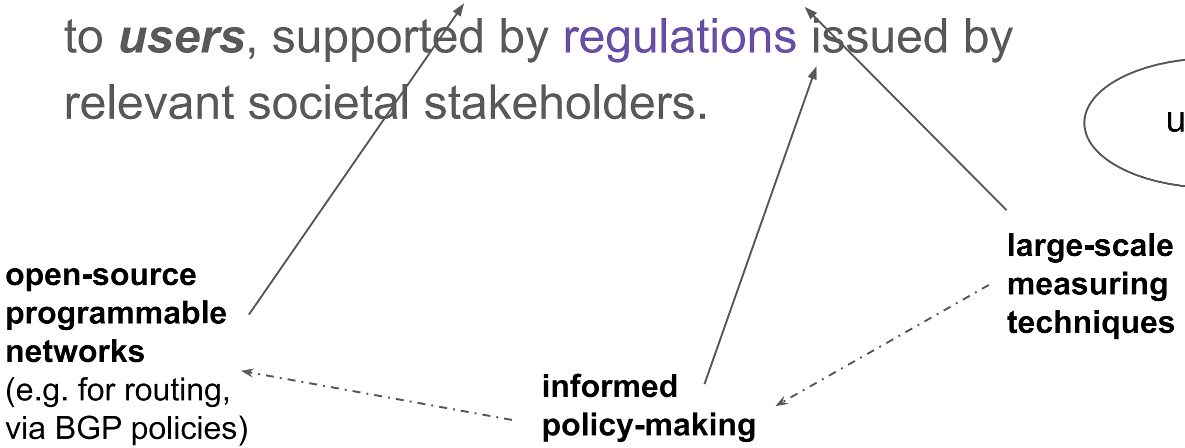
**large-scale measuring techniques**



Cristian Hesselman, Paola Grosso, Ralph Holz, Fernando Kuipers, Janet Hui Xue, Mattijs Jonker, Joeri de Ruiter, Anna Sperotto, Roland van Rijswijk-Deij, Giovane C.M. Moura, Aiko Pras, and Cees de Laat. *A Responsible Internet to Increase Trust in the Digital World.* Journal of Network and Systems Management 28, 4 (2020), 882–922.

# Responsible Internet social structure

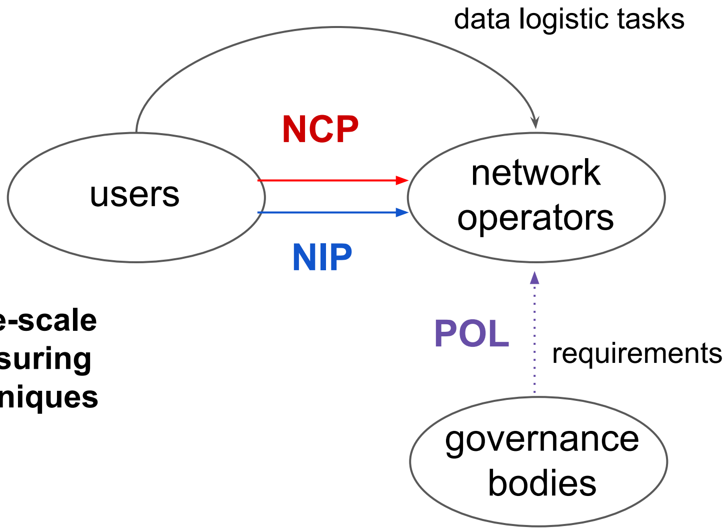
The Responsible Internet proposal (Hesselman et al., 2020) essentially envisions to **redistribute** control and monitoring abilities to ***users***, supported by regulations issued by relevant societal stakeholders.



**open-source programmable networks**
(e.g. for routing, via BGP policies)

**informed policy-making**

**large-scale measuring techniques**



Cristian Hesselman, Paola Grosso, Ralph Holz, Fernando Kuipers, Janet Hui Xue, Mattijs Jonker, Joeri de Ruiter, Anna Sperotto, Roland van Rijswijk-Deij, Giovane C.M. Moura, Aiko Pras, and Cees de Laat. *A Responsible Internet to Increase Trust in the Digital World.* Journal of Network and Systems Management 28, 4 (2020), 882–922.

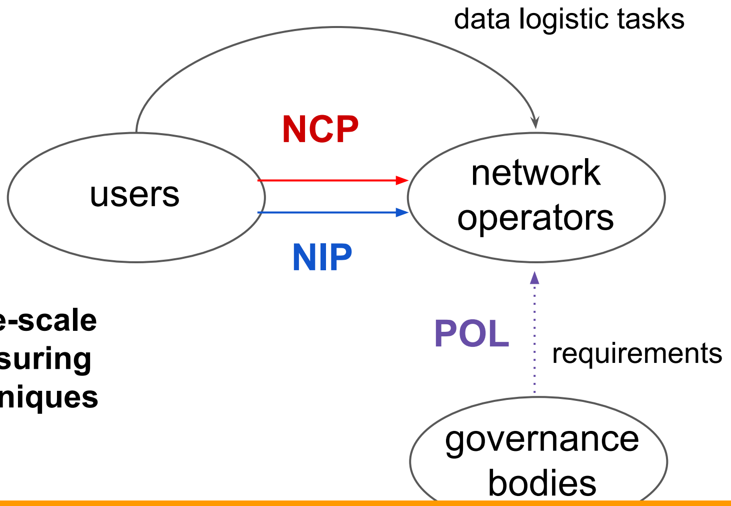
# Responsible Internet social structure

The Responsible Internet proposal (Hesselman et al., 2020) essentially envisions to **redistribute** control and monitoring abilities to ***users***, supported by regulations issued by relevant societal stakeholders.

**open-source programmable networks**
(e.g. for routing, via BGP policies)

**informed policy-making**

**large-scale measuring techniques**



**IS THIS COMPLETE?**

Cristian Hesselman, Paola Grosso, Ralph Holz, Fernando Kuipers, Janet Hui Xue, Mattijs Jonker, Joeri de Ruiter, Anna Sperotto, Roland van Rijswijk-Deij, Giovane C.M. Moura, Aiko Pras, and Cees de Laat. *A Responsible Internet to Increase Trust in the Digital World.* Journal of Network and Systems Management 28, 4 (2020), 882–922.

# Our paper raises two critiques

- **RESPONSIBILITY GAP**: Low-level programmability (e.g. for routing, via BGP policies) is not sufficient to capture and behaviourally operationalize the value structure of users.

- **REGULATIVE CONTINGENCY:** Power-relationships between roles should not be hard-coded (that is, should be partially programmable).

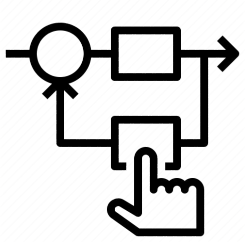
# Our paper raises two critiques

- **RESPONSIBILITY GAP**: Low-level programmability (e.g. for routing, via BGP policies) is not sufficient to capture and behaviourally operationalize the value structure of users.

- **REGULATIVE CONTINGENCY:** Power-relationships between roles should not be hard-coded (that is, should be partially programmable).

# Requirements for responsibility

An agent has (agentive) *responsibility* if it:



has the ability to **control** its own behaviour

has the ability to **foresee** the associated outcomes

has the ability to **assess** actions according to a certain preference/value structure

Sileno, G., Boer, A., Gordon, G., Rieder, B., Like Circles in the Water: Responsibility as a System-Level Function. Proceedings of 3rd XAILA workshop: Explainable and Responsible AI and Law, in conjunction with JURIX 2020 (2020)

# Requirements for responsibility

**necessary e.g. to identify wrong behaviour**

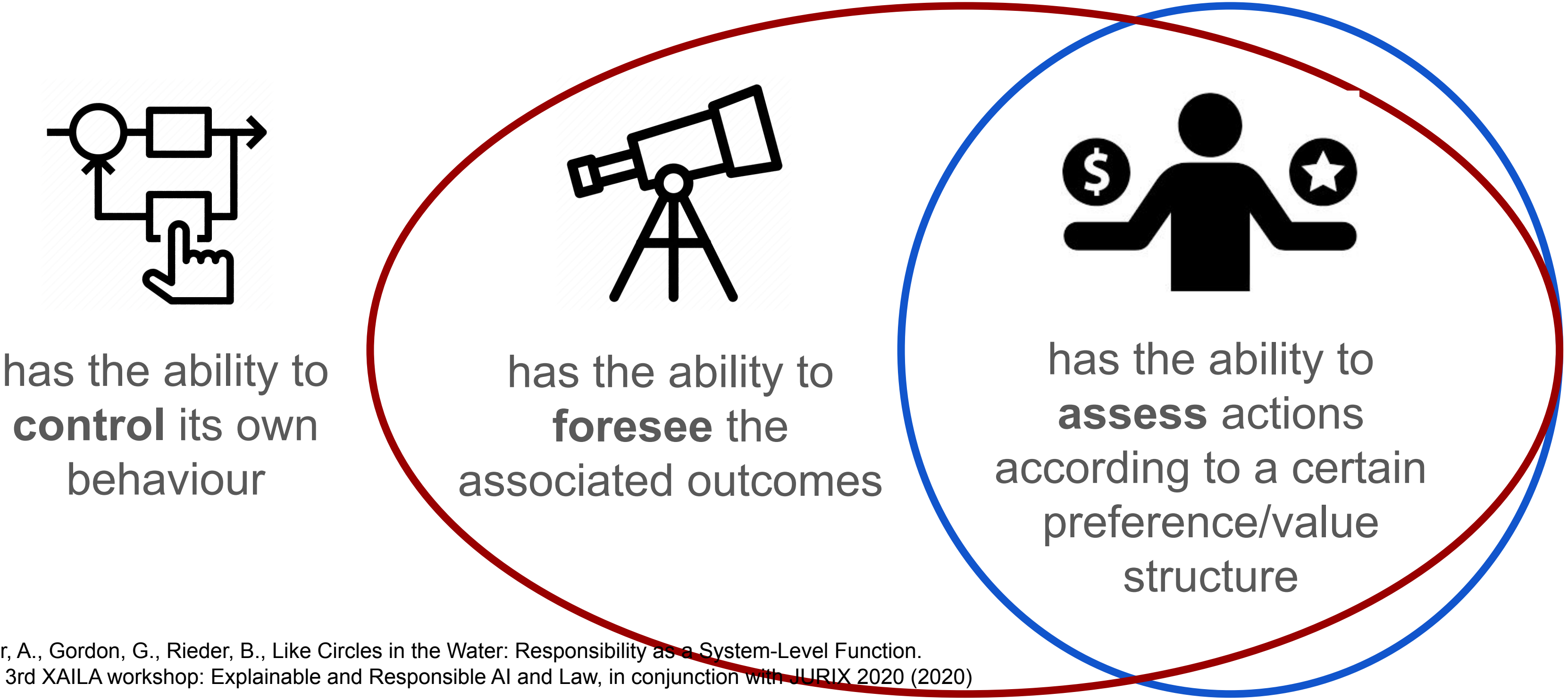
An agent has (agentive) *responsibility* if it:

- has the ability to **control** its own behaviour
- has the ability to **foresee** the associated outcomes
- has the ability to **assess** actions according to a certain preference/value structure

Sileno, G., Boer, A., Gordon, G., Rieder, B., Like Circles in the Water: Responsibility as a System-Level Function. Proceedings of 3rd XAILA workshop: Explainable and Responsible AI and Law, in conjunction with JURIX 2020 (2020)

# Requirements for responsibility

An agent has (agentive) *responsibility* if it:



**necessary e.g. to inhibit wrong behaviour**

**necessary e.g. to identify wrong behaviour**

has the ability to **control** its own behaviour

has the ability to **foresee** the associated outcomes

has the ability to **assess** actions according to a certain preference/value structure

Sileno, G., Boer, A., Gordon, G., Rieder, B., Like Circles in the Water: Responsibility as a System-Level Function. Proceedings of 3rd XAILA workshop: Explainable and Responsible AI and Law, in conjunction with JURIX 2020 (2020)

# Responsibility gap

**In the *Responsible Internet* proposal, users gain *controllability* by low-level programmability (via the NCP).**

has the ability to **control** its own behaviour

has the ability to **foresee** the associated outcomes

has the ability to **assess** actions according to a certain preference/value structure

# Responsibility gap

In the *Responsible Internet* proposal, users gain ***controllability* by low-level programmability** (via the **NCP**).

**...but nothing is said about the two other components.**

has the ability to **control** its own behaviour

has the ability to **foresee** the associated outcomes

has the ability to **assess** actions according to a certain preference/value structure

# Responsibility gap

In the *Responsible Internet* proposal, users gain ***controllability* by low-level programmability** (via the **NCP**).

...but nothing is said about the two other components.

has the ability to **control** its own behaviour

has the ability to **foresee** the associated outcomes

has the ability to **assess** actions according to a certain preference/value structure

**HOW CAN WE REPAIR THIS?**

# Reducing the responsibility gap

**[1] We need a model of how the world functions.**

→ **EXPECTATIONS artefact**

has the ability to **control** its own behaviour

has the ability to **foresee** the associated outcomes

has the ability to **assess** actions according to a certain preference/value structure

# Reducing the responsibility gap

has the ability to **control** its own behaviour

**[1] We need a model of how the world functions.**

**EXPECTATIONS artefact**

has the ability to **foresee** the associated outcomes

**[2] We need a model of what is valuable in the world.**

**high-level POLICY artefact**

has the ability to **assess** actions according to a certain preference/value structure

# Reducing the responsibility gap

has the ability to **control** its own behaviour

**[1] We need a model of how the world functions.**

**EXPECTATIONS artefact**

*norm as in* *<u>normal</u>*

has the ability to **foresee** the associated outcomes

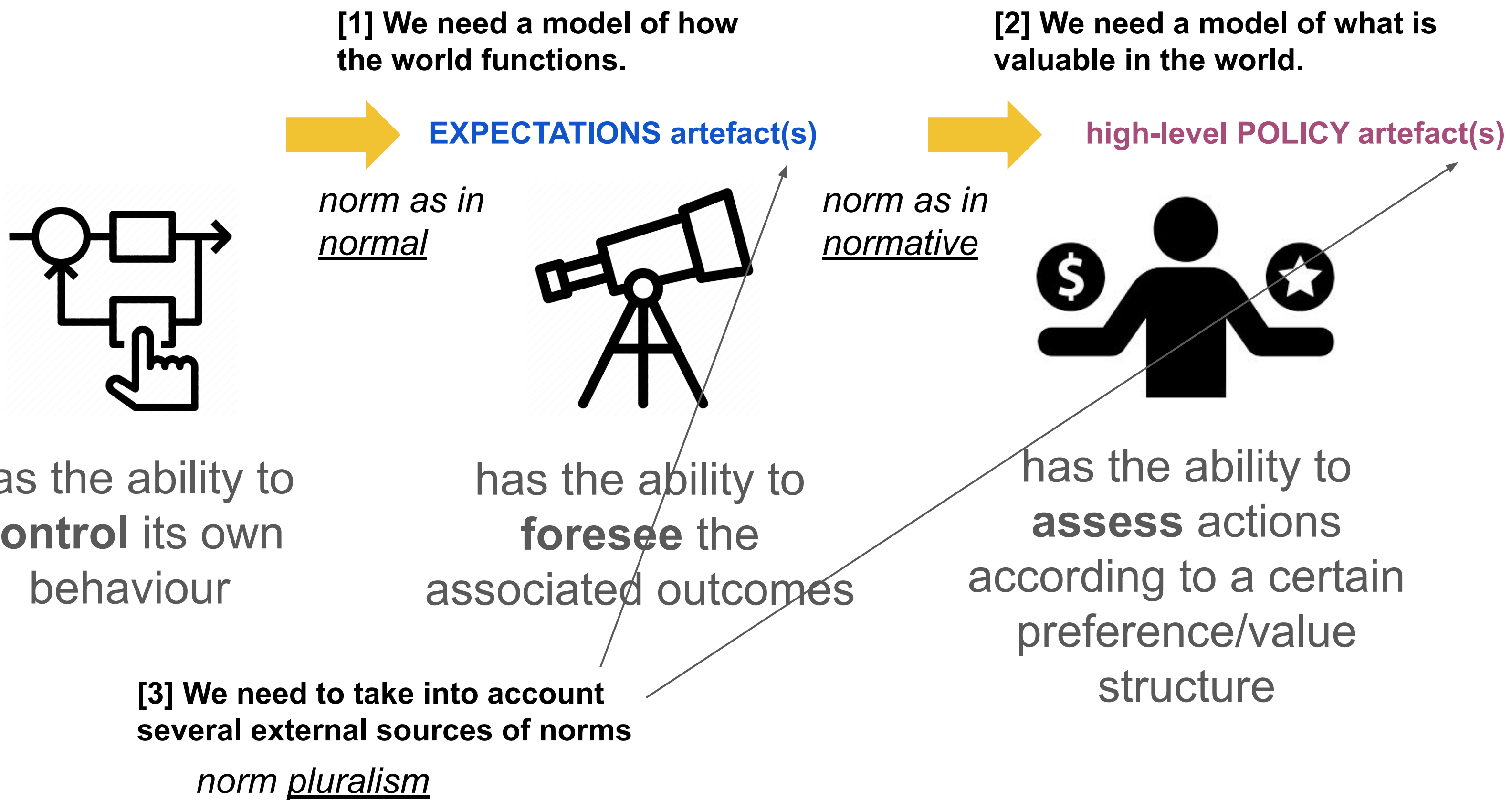
**[2] We need a model of what is valuable in the world.**

**high-level POLICY artefact**

*norm as in* *<u>normative</u>*

has the ability to **assess** actions according to a certain preference/value structure

# Reducing the responsibility gap



**[1] We need a model of how the world functions.**

**[2] We need a model of what is valuable in the world.**

**EXPECTATIONS artefact(s)**

**high-level POLICY artefact(s)**

*norm as in* *normal*

*norm as in* *normative*

has the ability to **control** its own behaviour

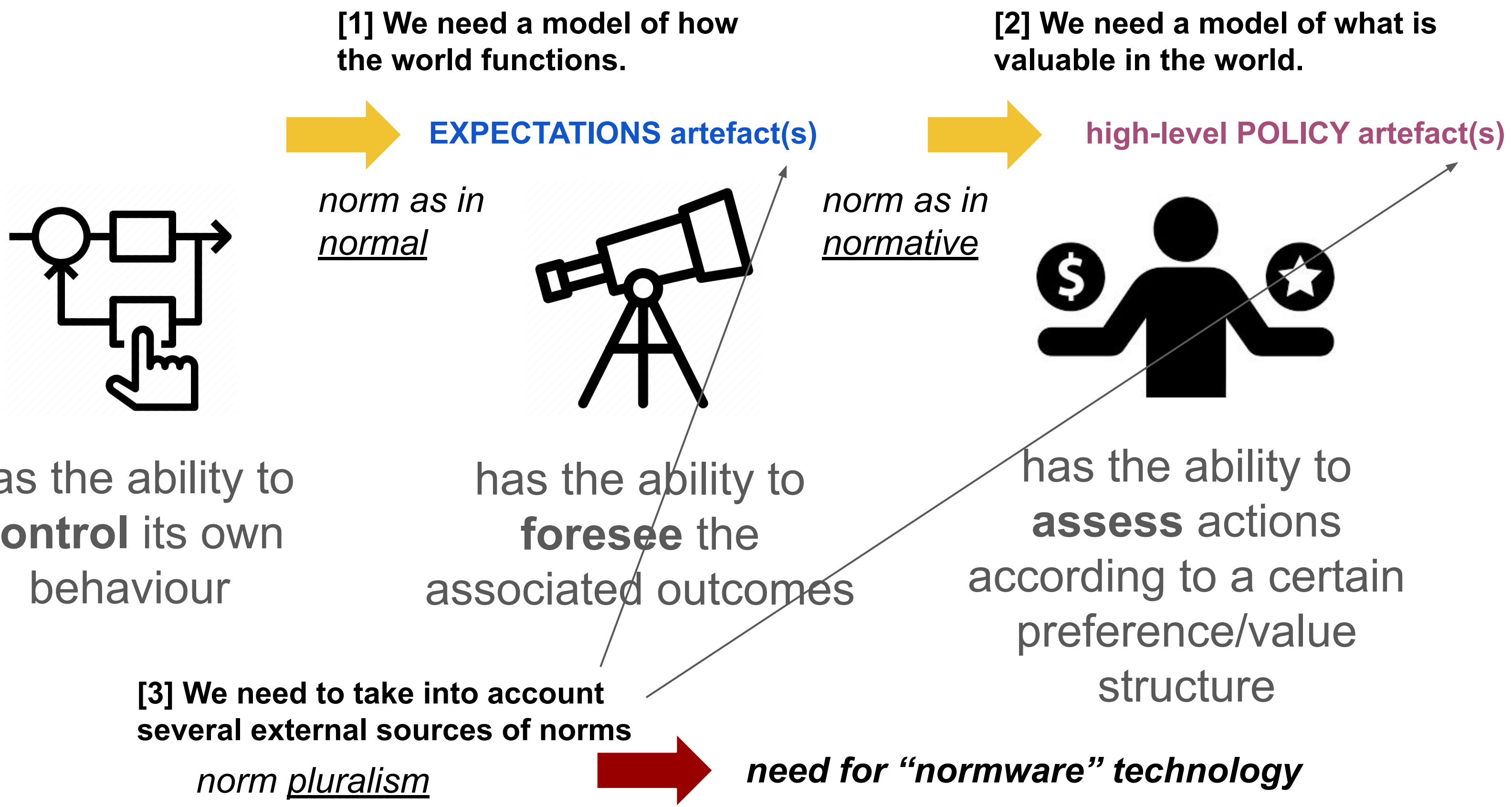
has the ability to **foresee** the associated outcomes

has the ability to **assess** actions according to a certain preference/value structure

**[3] We need to take into account several external sources of norms**

*norm pluralism*

# Reducing the responsibility gap



**[1] We need a model of how the world functions.**

**EXPECTATIONS artefact(s)**

*norm as in <u>normal</u>*

**[2] We need a model of what is valuable in the world.**

**high-level POLICY artefact(s)**

*norm as in <u>normative</u>*

has the ability to **control** its own behaviour

has the ability to **foresee** the associated outcomes

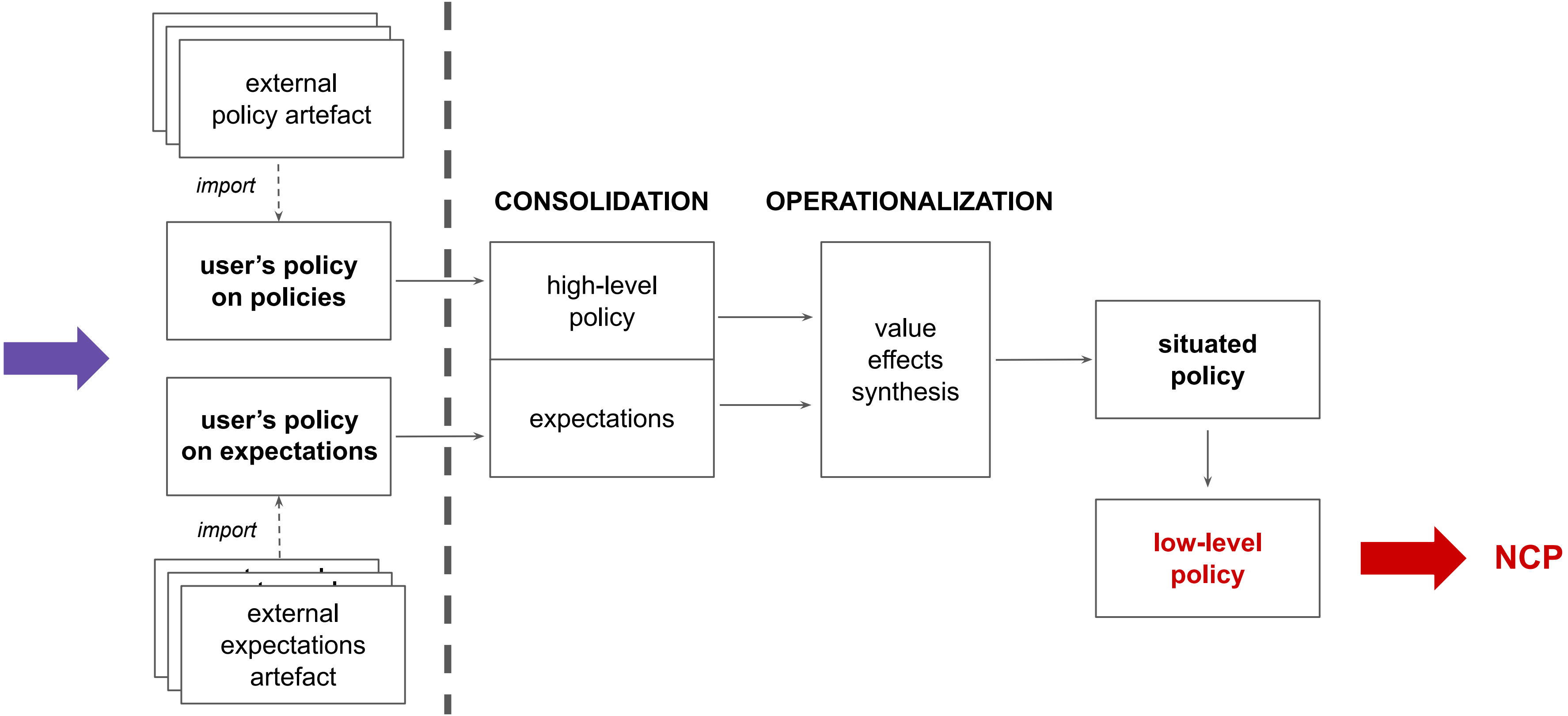
has the ability to **assess** actions according to a certain preference/value structure

**[3] We need to take into account several external sources of norms**

*norm <u>pluralism</u>*

***need for "normware" technology***

# From higher-level to lower-level policies



# Our paper raises two critiques

- **RESPONSIBILITY GAP**: Low-level programmability (e.g. for routing, via BGP policies) is not sufficient to capture and behaviourally operationalize the value structure of users.

- **REGULATIVE CONTINGENCY:** Power-relationships between roles should not be hard-coded (that is, should be partially programmable).

# Regulative contingency

Users, network operators, and the various governance bodies have all legitimate interests to play a role in policy-making.

Prototypical conflictual design choice: *anonymity vs accountability*.

# Regulative contingency

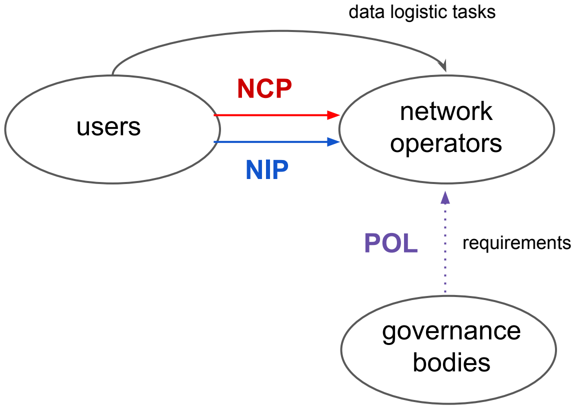
Users, network operators, and the various governance bodies have all legitimate interests to play a role in policy-making.

Prototypical conflictual design choice: *anonymity vs accountability*.

Governmental, public agencies are users of the infrastructure, and play a role in the infrastructure governance bodies.

The Responsible Internet proposal says that POL

- should ***be informed*** by NIP and
- should ***drive*** the NCP.



# Regulative contingency

Users, network operators, and the various governance bodies have all legitimate interests to play a role in policy-making.

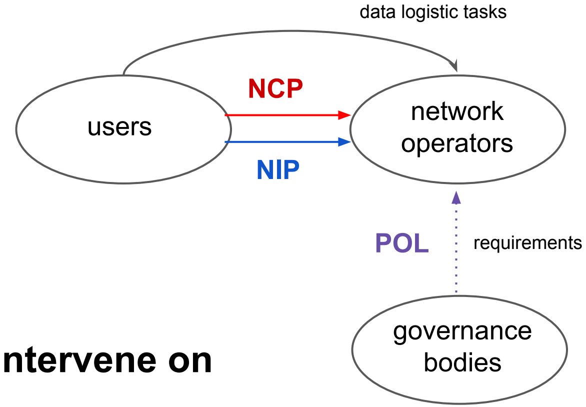
Prototypical conflictual design choice: *anonymity vs accountability*.

Governmental, public agencies are users of the infrastructure, and play a role in the infrastructure governance bodies.

The Responsible Internet proposal says that POL

- should ***be informed*** by NIP and
- should ***drive*** the NCP.

**but how? to what extent regulators can intervene on users' activity on the infrastructure?**



data logistic tasks

users

NCP

NIP

network operators

POL

requirements

governance bodies

# Regulative contingency

Users, network operators, and the various governance bodies have all legitimate interests to play a role in policy-making.

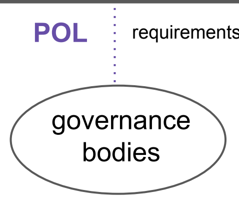
Prototypical conflictual design choice: *anonymity vs accountability*.

There is no definitive, global solution: checks & balances vary on a local basis.

- Power-relationships between roles should not be hard-coded, but programmable.

- should ***drive*** the NCP.

**but how? to what extent regulators can intervene on users' activity on the infrastructure?**



# Regulative contingency

Users, network operators, and the various governance bodies have all legitimate interests to play a role in policy-making.

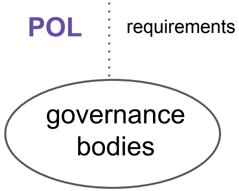
Prototypical conflictual design choice: *anonymity vs accountability*.

There is no definitive, global solution: checks & balances vary on a local basis.

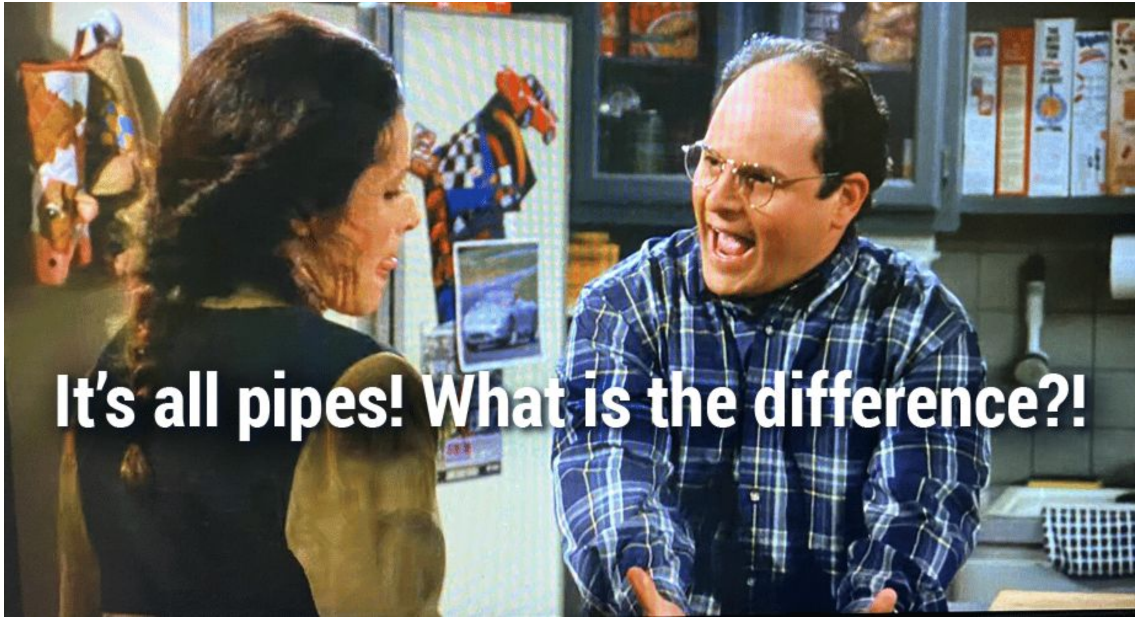
- Power-relationships between roles should not be hard-coded, but programmable. → *an additional use case for a policy-based technology*

- should ***drive*** the NCP.

**but how? to what extent regulators can intervene on users’ activity on the infrastructure?**



# Traditional principle of the internet



"dumb pipe" principle

# New principles?

**“give me eyes, and I’ll know where I’ll go”** Computation cannot be “responsible” if the computational agent has no means to evaluate the effect of its actions, and then to prevent wrong outcomes.

# New principles?

**“give me eyes, and I’ll know where I’ll go”** Computation cannot be “responsible” if the computational agent has no means to evaluate the effect of its actions, and then to prevent wrong outcomes.

**“pipes are dumb, water drinkers are not”** Networks are supposed to operate blindly with respect to the content they transport, by making decisions on packets and unaware of the value of the whole transactions. *But this information is (to some extent) available at the users’ endpoints!*

In full control, users should be able to provide some artefact specifying their preference/value structure and their expectations. Network operators should operate, still blindly, just according to these directives.

# New principles?

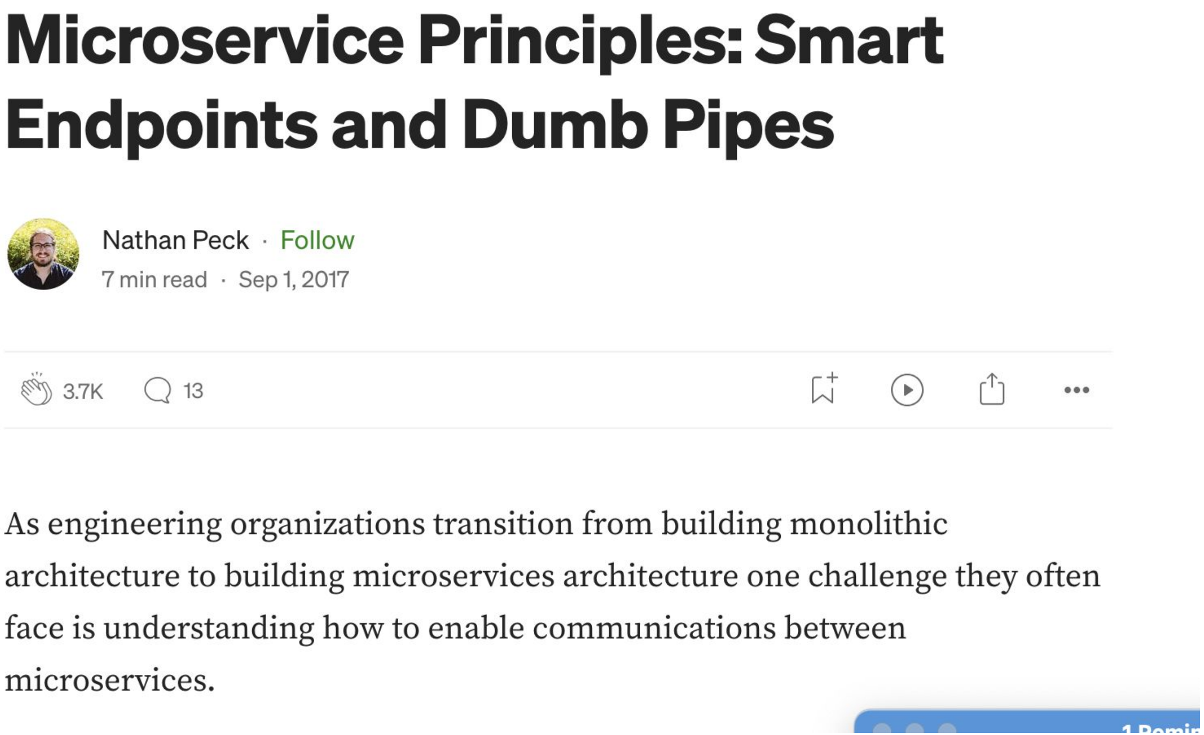
**“do not hard-code what is soft-coded”** It is premature, if not wrong, to aim to a definitive solution concerning power-relationships (e.g. full-control for users and full-blindness for network operators). Too many local contextual factors intervene to set which are the “right” checks and balances. We need ***programmability*** also at this level. *But what to program?*

# New principles?

**“do not hard-code what is soft-coded”** It is premature, if not wrong, to aim to a definitive solution concerning power-relationships (e.g. full-control for users and full-blindness for network operators). Too many local contextual factors intervene to set which are the “right” checks and balances. We need ***programmability*** also at this level**.** *But what to program?*

**“what works, it may work”** For a global infrastructure like the Internet, possible starting points would be normative constructs and frameworks developed in non-computational contexts, as in international law, or most plausibly in *international private law*, already operative across very diverse jurisdictions.

# Relevant for micro-services too!



## Microservice Principles: Smart Endpoints and Dumb Pipes

Nathan Peck · Follow

7 min read · Sep 1, 2017

As engineering organizations transition from building monolithic architecture to building microservices architecture one challenge they often face is understanding how to enable communications between microservices.

# Part IV:
# Opportunities and Open challenges

# Normware offers better modularity

- Plenty of assumption exist in traditional approaches/tools. For instance, who monitors for violations? But there are others!

# Normware offers better modularity

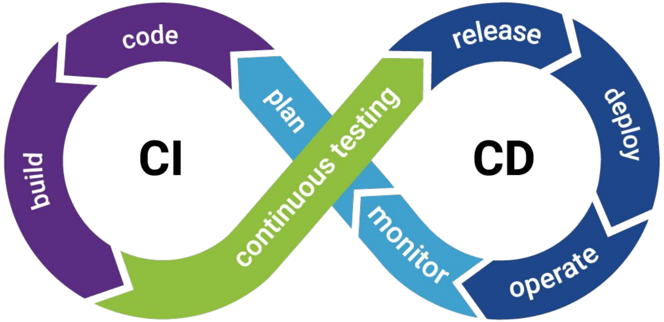
- Plenty of assumption exist in traditional approaches/tools. For instance, who monitors for violations? But there are others!
- Applying a normware stance, you are enabled to
  - separate policy from expectations components. Facts should be elaborated from different modules.
  - separate directives-related issues from resolving conflicts issues.
  - separate ***social operationalization*** from ***computational operationalization***!

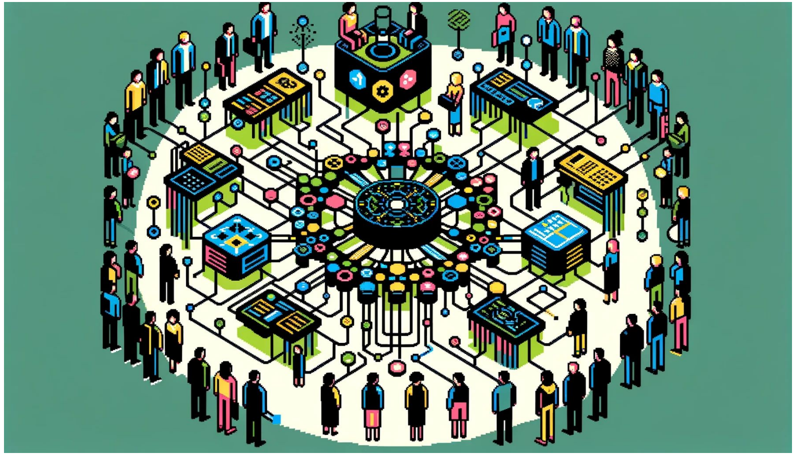
# Normware offers better modularity

- Plenty of assumption exist in traditional approaches/tools. For instance, who monitors for violations? But there are others!
- Applying a normware stance, you are enabled to
  - separate policy from expectations components. Facts should be elaborated from different modules.
  - separate directives-related issues from resolving conflicts issues.
  - separate ***social operationalization*** from ***computational operationalization***!
- This gives spaces to **control structure** design, depending on the task: eg. authorization/intervention, forensics, auditing, testing, verification, and so on.

# Computational counterpart of governance

- From continuous integration to continuous governance?

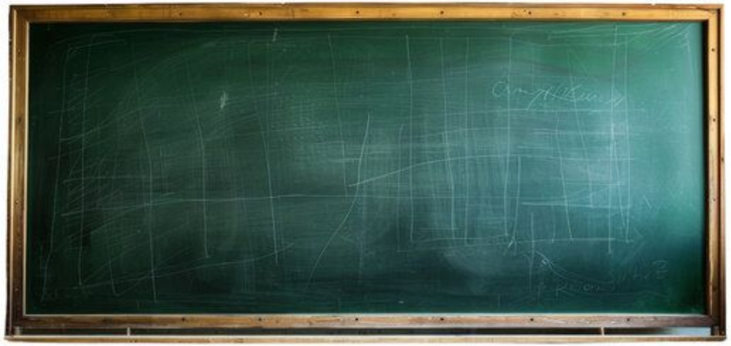




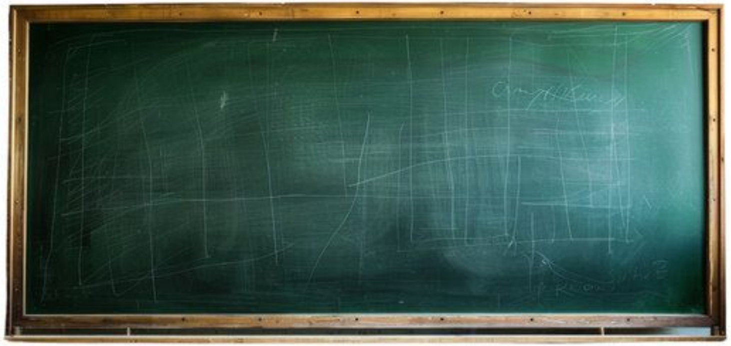
continuous authorizations/interventions, forensics, auditing, testing, verifications!

# Open theoretical problems

- Identities! Dynamic multiple inheritance is a known tough problem.

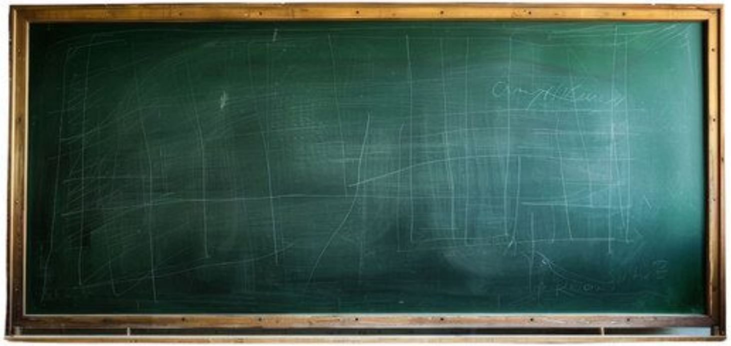


# Open theoretical problems



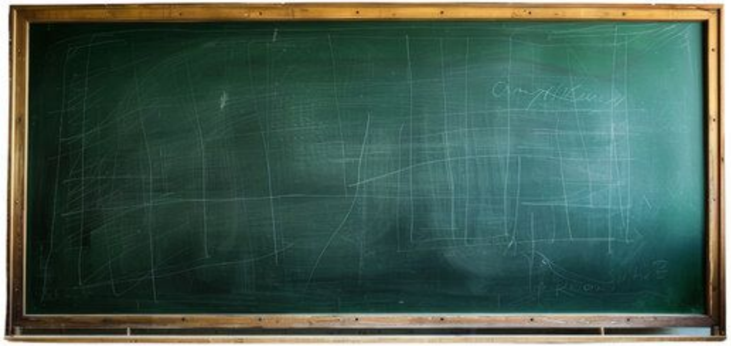
- Identities! Dynamic multiple inheritance is a known tough problem.
- Against extensionality: here all is about roles (formal semantics are based usually on set theory).

# Open theoretical problems



- Identities! Dynamic multiple inheritance is a known tough problem.
- Against extensionality: here all is about roles (formal semantics are based usually on set theory).
- Transformational and reactive components need to coexist (verification approaches typically take a declarative/functional or an imperative perspective)

# Open theoretical problems



- Identities! Dynamic multiple inheritance is a known tough problem.
- Against extensionality: here all is about roles (formal semantics are based usually on set theory).
- Transformational and reactive components need to coexist (verification approaches typically take a declarative/functional or an imperative perspective)
- Seemingly, very little is theorized about “consolidation”

# Normware engineering: opportunities and open problems

7 November 2024, *IPA Fall Days*

Fall Days on Models for Constructing Software

Giovanni Sileno, g.sileno@uva.nl
Socially Intelligent Artificial Systems (SIAS),
Informatics Institute, University of Amsterdam