

Lu-Chi Liu, MSc<sup>1</sup>; L. Thomas van Binsbergen, PhD<sup>2</sup>

<sup>1</sup>University of Amsterdam, <sup>2</sup>Centrum Wiskunde & Informatica

### eFLINT: Act, Duty, and Fact Frames

- eFLINT is an action-based language for formalizing norms as found in regulations, policies, code-of-conducts, business protocols, sharing agreements, etc.
- Ontological concepts and relations are described through **fact-type** declarations.
- Powers and duties are derived from applicable actions and expected actions. (**act-type** and **duty-type** declarations)

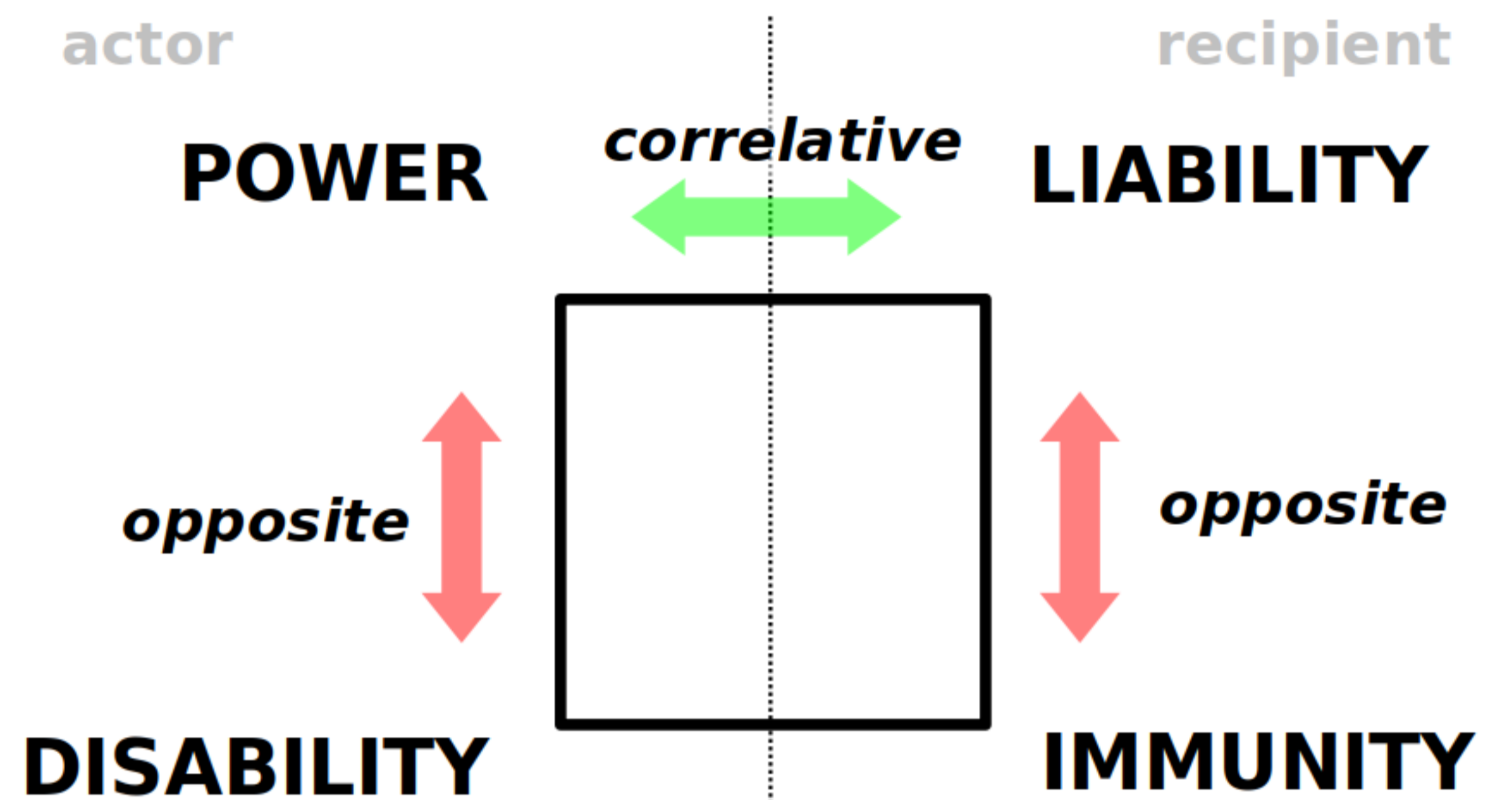
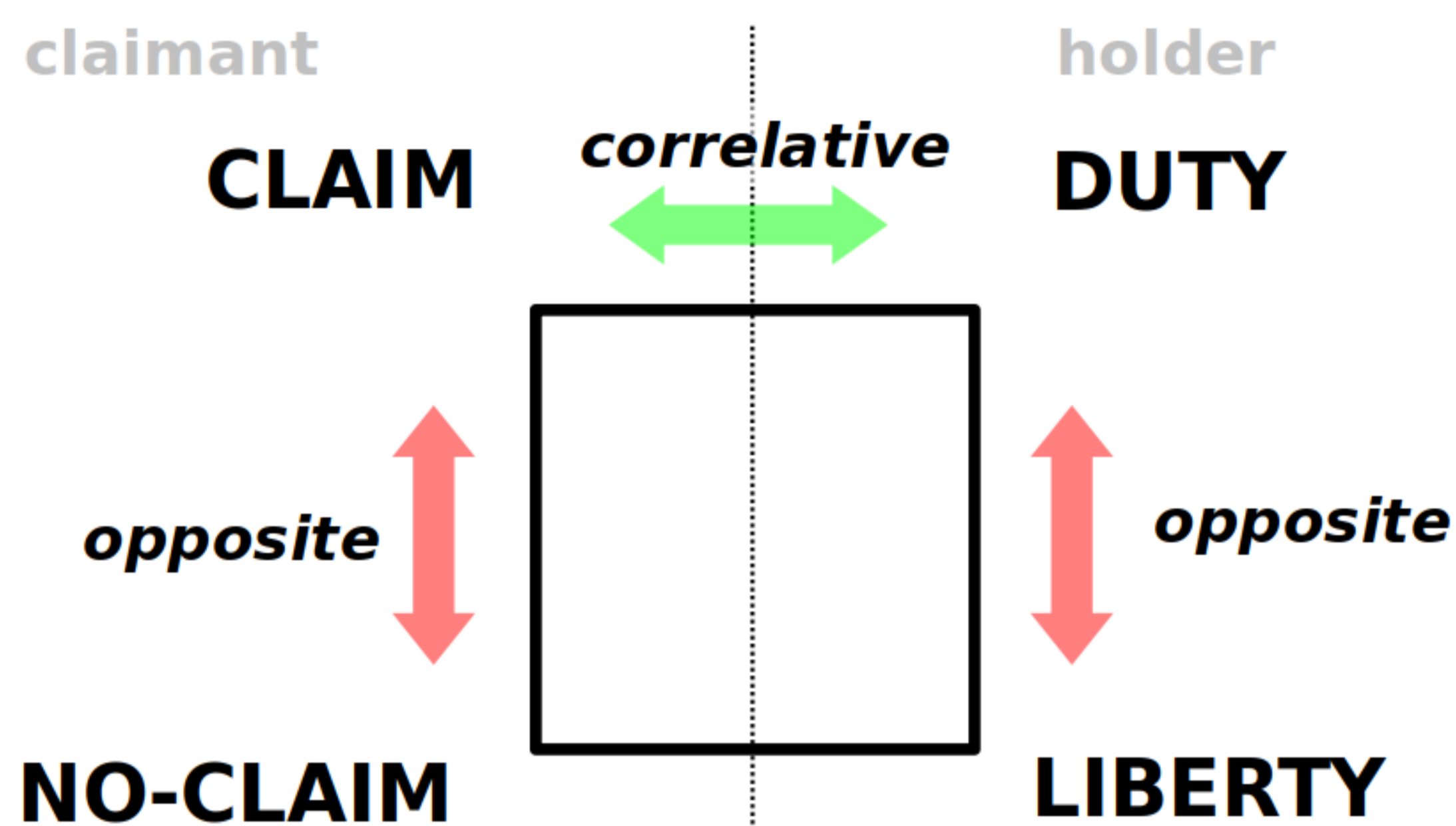
```

Fact person
Placeholder parent For person
Placeholder child For person

Fact natural parent
  Identified by parent * child
Fact adoptive parent
  Identified by parent * child
Fact legal parent
  Identified by parent * child
  Present when adoptive parent() || natural parent()
  
```

Fact frame

### How to Interpret Norms?



How do we write down an interpretation formally?

- An interpretation is a collection of power-liability relations and duty-claim relations. When powers are enacted (by performing actions), the effects may change which power-liability and duty-claim relations are present.
- An interpretation is written down formally in eFLINT by declaring act-types with associated actor, recipient, and conditions and duty-types with associated duty-holder, claimant and an optional violation condition.

- X is in a power-liability relation with Y at a particular moment in time if at that moment X has an act with recipient Y with a satisfied precondition. The effects of enacting on that power are described in the postconditions of the act.
- X is in a duty-claim relation with Y at a particular moment in time if at that moment X is the duty-holder of a duty with claimant Y. The duty is violated (Y has a valid claim) if the duty is present and the violation condition of the duty holds.

### Example: GDPR Duty to Remove

```

// actors
Fact subject      Identified by Bob
Fact controller   Identified by ControllerX
Fact data_type    Identified by PhoneNumber, Address

// relations
Fact personal_data Identified by subject * data_type
Fact data_stored_by Identified by personal_data * controller

// norms
Duty duty_to_remove_data
  Holder controller
  Claimant subject
  Related to data_type
Act ask_remove_data
  Actor subject
  Recipient controller
  Related to data_type
  Conditioned by !duty_to_remove_data()
  Creates duty_to_remove_data()
Act remove_data
  Actor controller
  Recipient subject
  Related to data_type
  Conditioned by
    data_stored_by(personal_data(), controller)
  Terminates
    data_stored_by(personal_data(), controller),
    duty_to_remove_data()
  
```

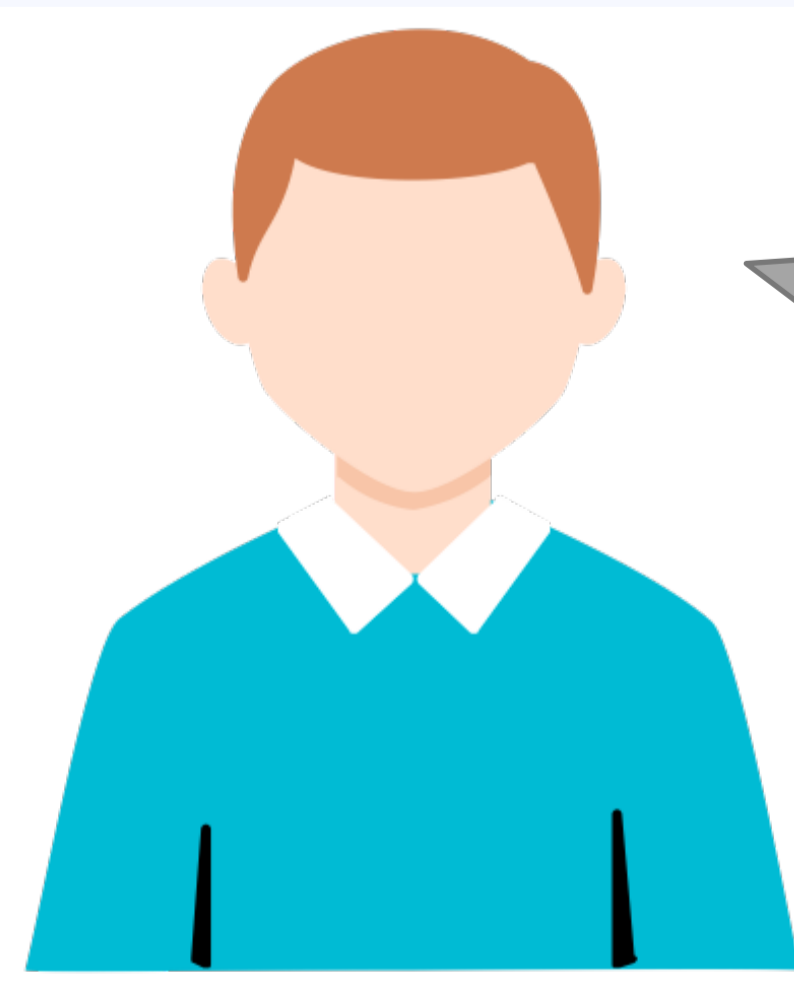
Duty frame:  
Holder &  
Claimant

Act frame:  
Actor &  
Recipient

```

// initial state
ask_removal. remove_data.

// scenario
Create instance +data_stored_by(personal_data(Bob, Address), ControllerX).
                +data_stored_by(personal_data(Bob, PhoneNumber), ControllerX).
Perform action  !ask_remove_data(Bob, ControllerX, PhoneNumber).
                !ask_remove_data(Bob, ControllerX, Address).
                !remove_data(ControllerX, Bob, Address).
Query          ?duty_to_remove_data(ControllerX, Bob, PhoneNumber). -> Query succeeded
                ?duty_to_remove_data(ControllerX, Bob, Address). -> Query failed
  
```



According to GDPR, if I ask you to remove my data, you have the duty to do so!

#### Acknowledgements

The Calculamus-Flint project is a collaboration between the University of Amsterdam, TNO, ICTU, the Ministry of Justice and Security and the Ministry of Finance, and aims to reduce the efforts made by governmental organizations towards implementing the rules and services described in laws and policies. This work has been supported by the NWO project Secure Scalable Policy-enforced Distributed Data Processing(628.009.014) which is part of the NWO research program Big Data: Real Time ICT for Logistics.

