Digital Enforceable Contracts (DEC): Making Smart Contracts Smarter

Lu-Chi Liu, Giovanni Sileno, Tom van Engers

Complex Cyber Infrastructure Group, Informatics Institute, University of Amsterdam

Background

- Current smart contracts have limited capabilities of normative representations, making them distant from actual contracts.
- Normative contents (duty and power) can be modeled into logic-based representation.
- DEC provides a general architecture where various enforcement mechanisms are enabled by normative reasoning. For example, to check whether an action will lead to a duty.

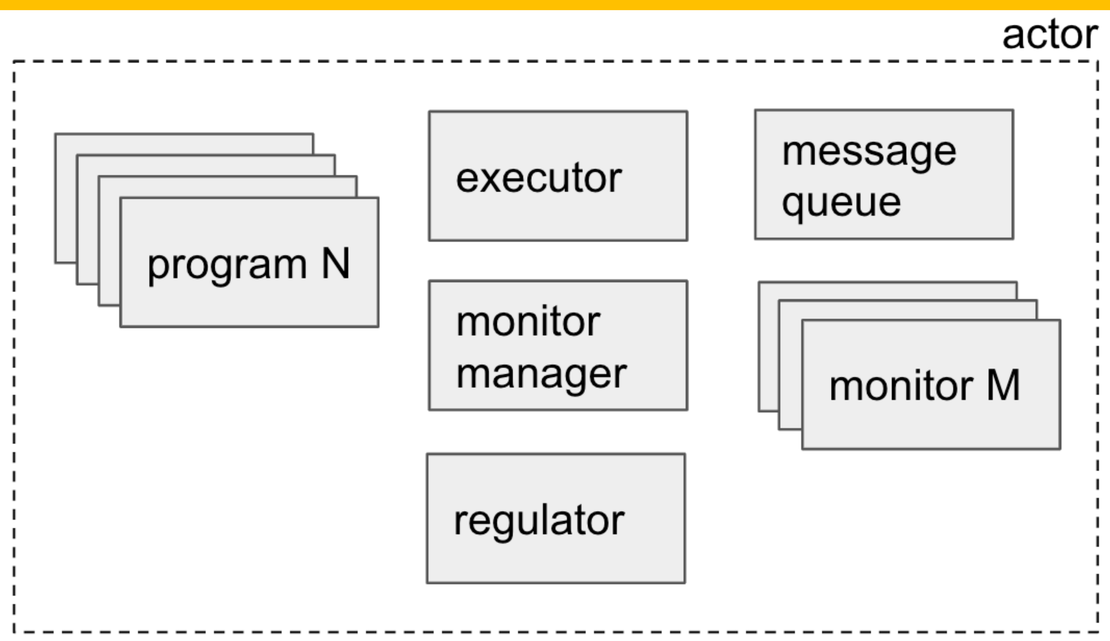
```
// written in eFLINT
Act request to modify consent
  Actor subject
  Recipient controller
  Related to consent, other purpose
  Conditioned by
    consent && consent.purpose != other purpose
  Creates duty to modify consent()

Duty duty to modify consent
 Holder controller
 Claimant subject
 Related to consent, other purpose
```

Norms related to GDPR

Actor-based Modular Architecture

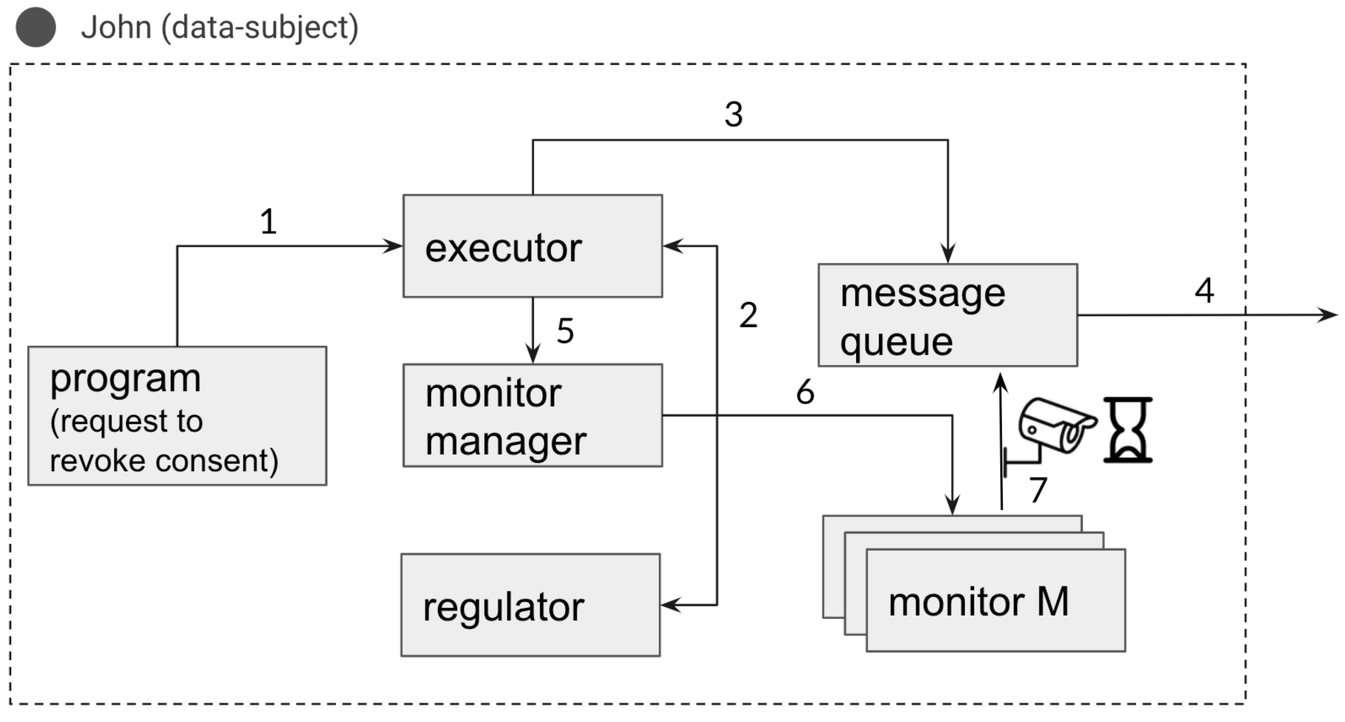
The architectural model is composed of a selected set of modules providing the functionality to run enforcement constructs.

Actor (the minimal unity of agency):
Program - plan to achieve a given design goal
Executor - internal control of the actor
Message queue - communication channel
Monitor - listeners that hook to events or facts
Monitor manager - handle monitors
Regulator - normative reasoning

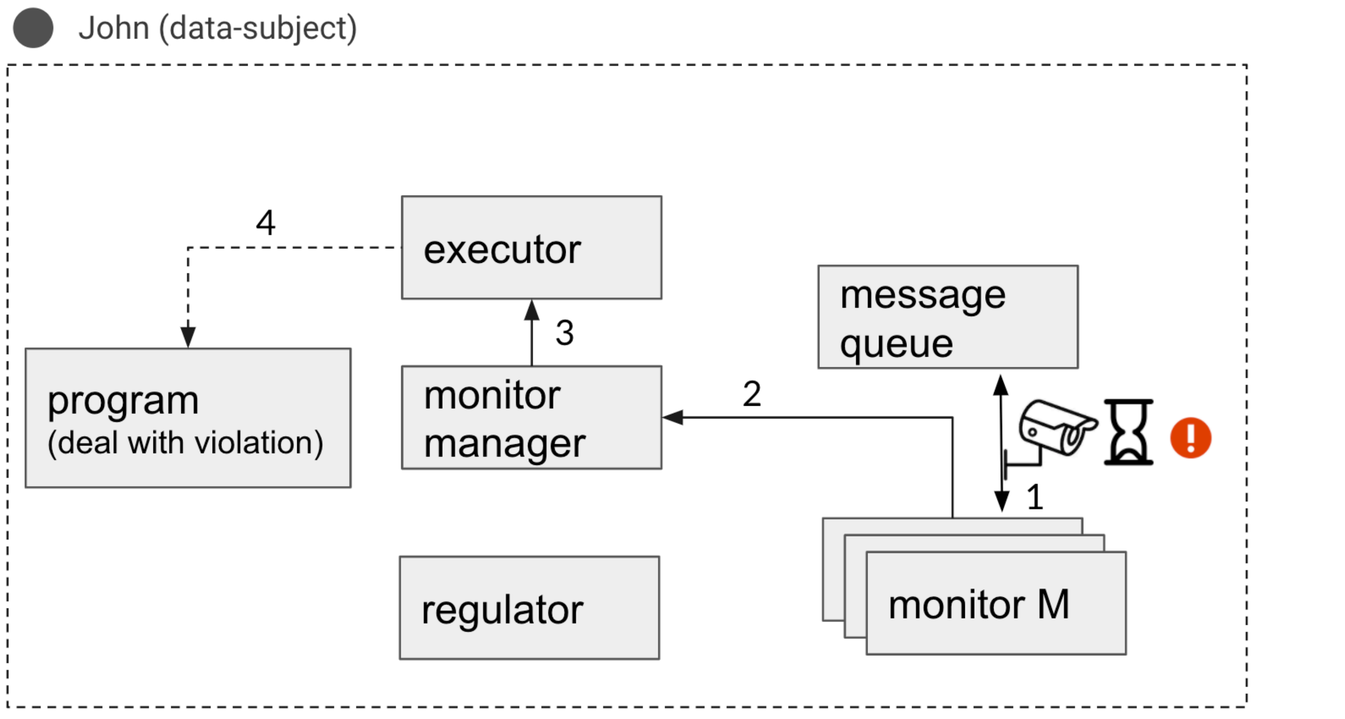


Prototype being developed using Akka-typed actor-oriented programming framework

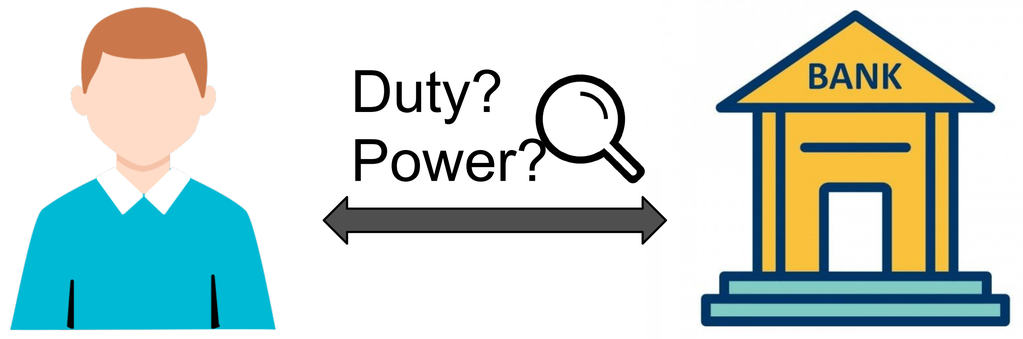
Example: A Data-sharing Scenario with GDPR



1) John (data-subject) attempts to revoke his consent of using his data from Bank (data-controller).
2) The executor sends query to the regulator to check related permissions and duties. (According to GDPR, Bank, as data-controller, has the duty to fulfill this request.)
3) The executor sends this request to the queue.
4) The request is then sent to Bank.
5) The executor asks monitor manager to create a monitor to check for violation.
6) A monitor is created.
7) The monitor checks messages from Bank with a timeout mechanism.



1) When the duty is due and not fulfilled, the monitor will be aware of this violation.
2) The monitor reports the violation.
3) Monitor manager notifies the executor of the violation.
4) The executor takes actions to deal with the violation.



Acknowledgements
This research is funded by the Dutch Organization for Scientific Research (NWO) under contracts 628.009.014 (SSPDDP project) and 628.001.001 (DL4LD project).

References

[1] Lu-Chi Liu, Giovanni Sileno, and Tom van Engers. Digital enforceable contracts: making smart contracts smarter. JURIX. 2020.

[2] L. Thomas van Binsbergen, Lu-Chi Liu, Robert van Doesburg, and Tom van Engers. eFLINT: a domain-specific language for executable norm specifications. In Proceedings of GPCE '20. ACM, 2020.