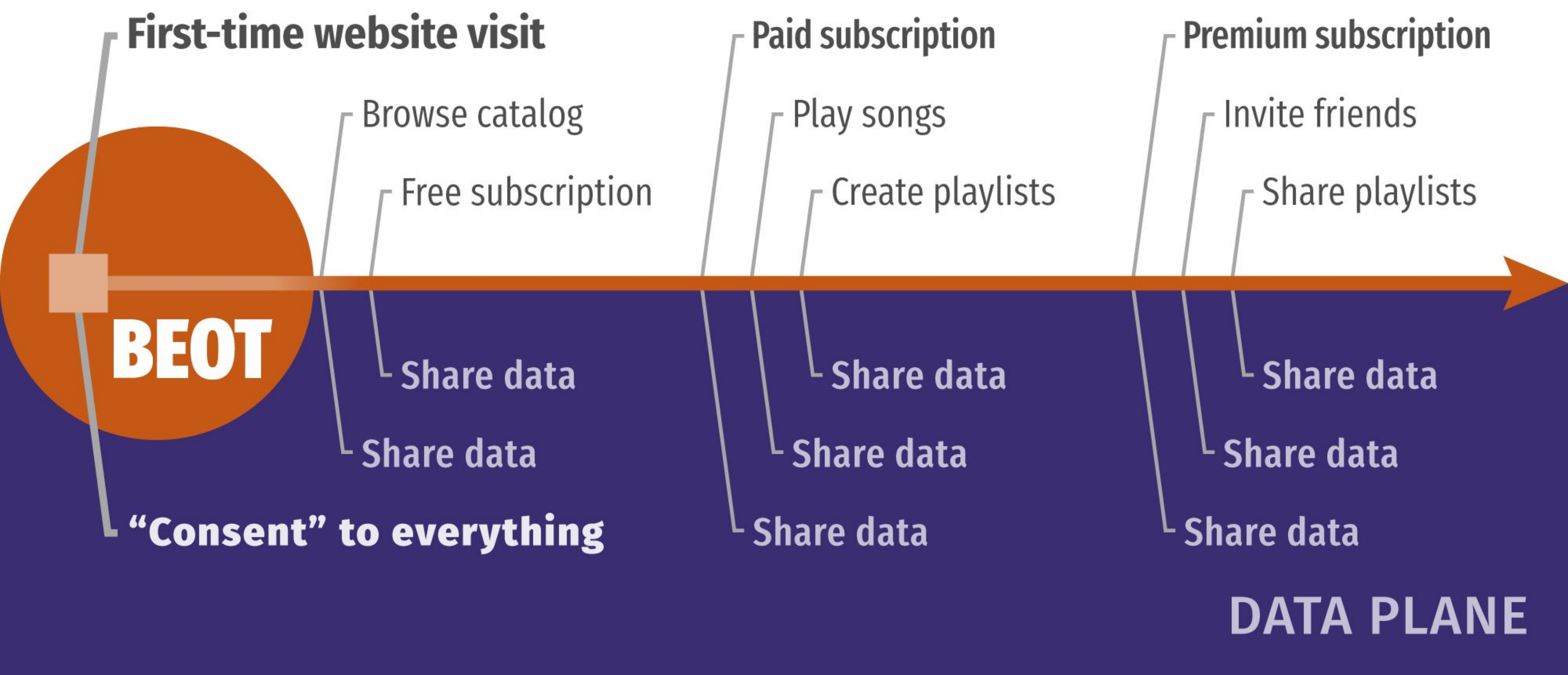
Automated Policy Negotiation: a Four-Course Meal

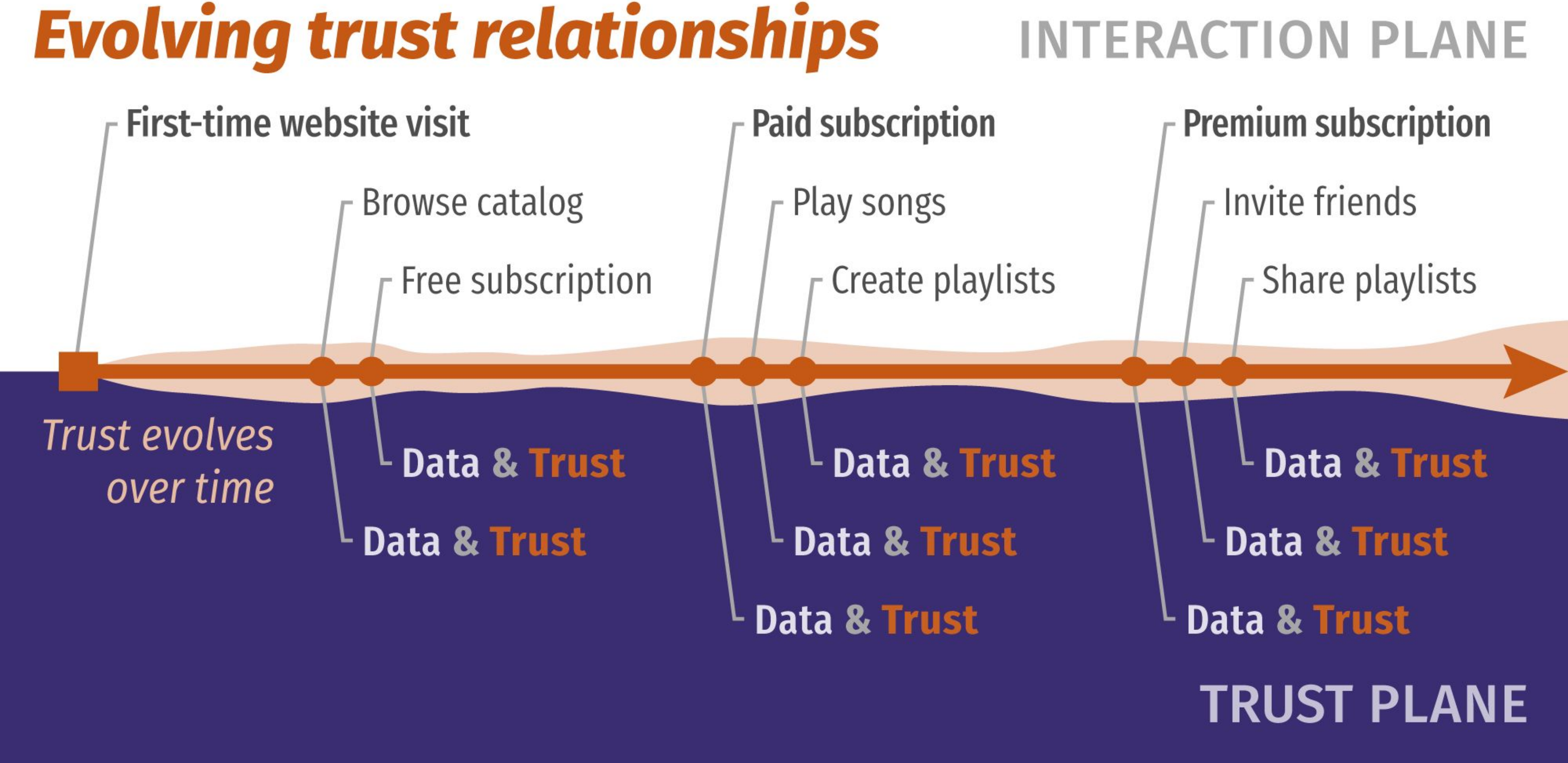
Patrick Hochstenbach, Beatriz Esteves, Ruben Verborgh

patrick.hochstenbach@ugent.be
beatriz.esteves@ugent.be

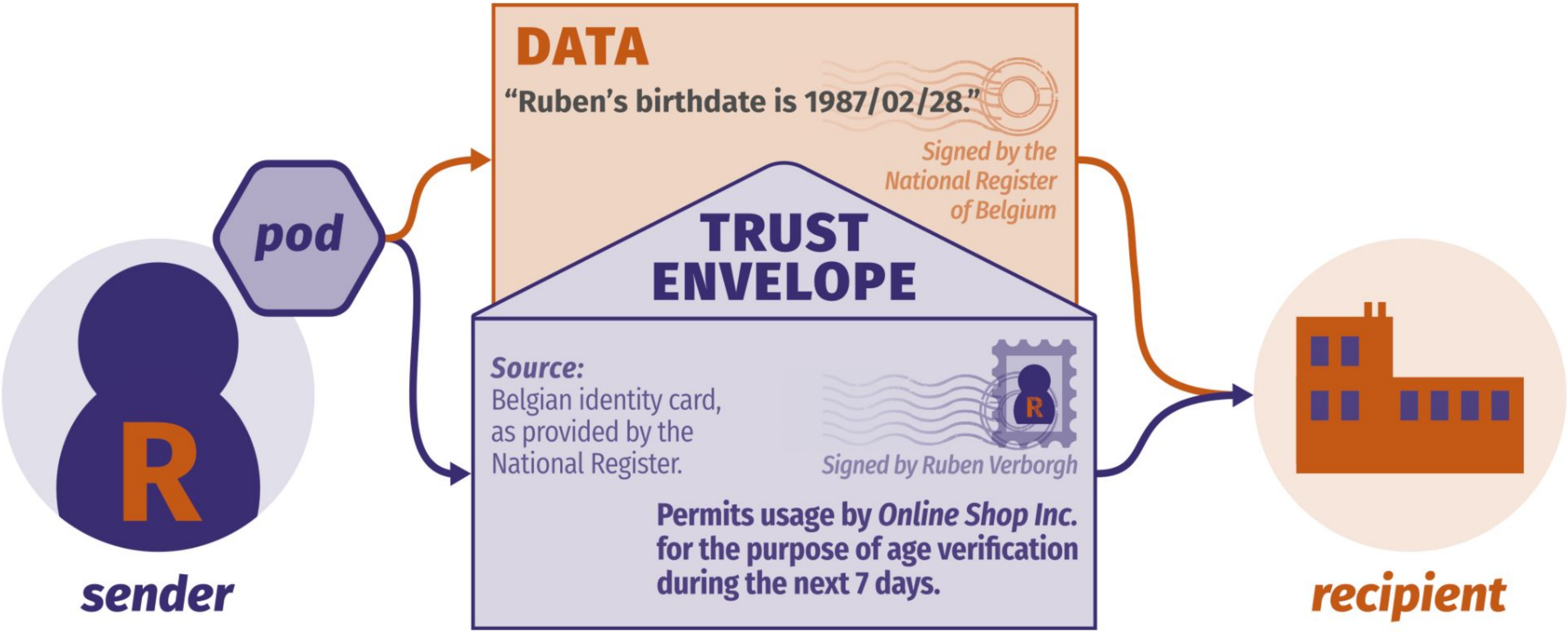
Trustless consent model



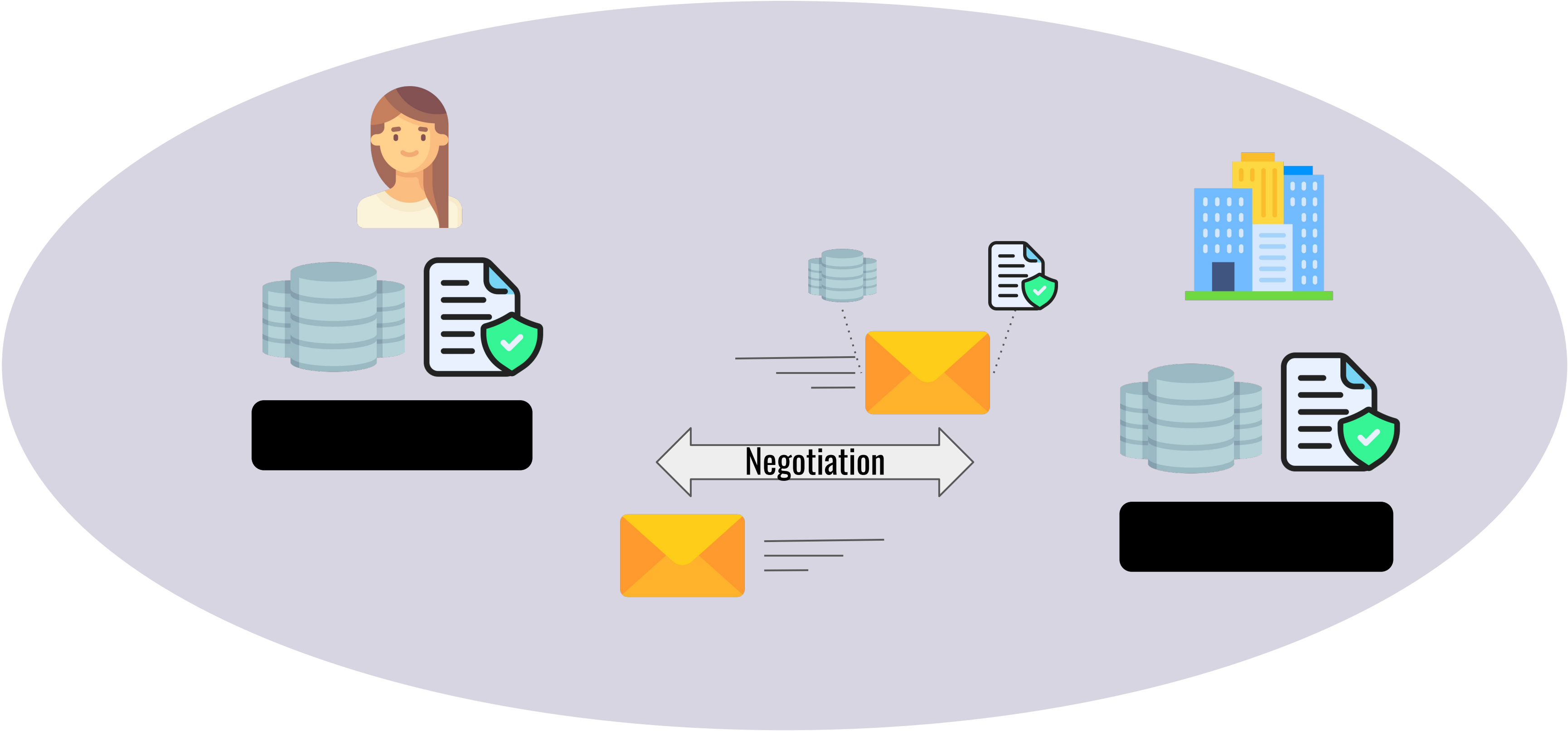
Evolving trust relationships



Trust envelopes as vehicles of history & destiny

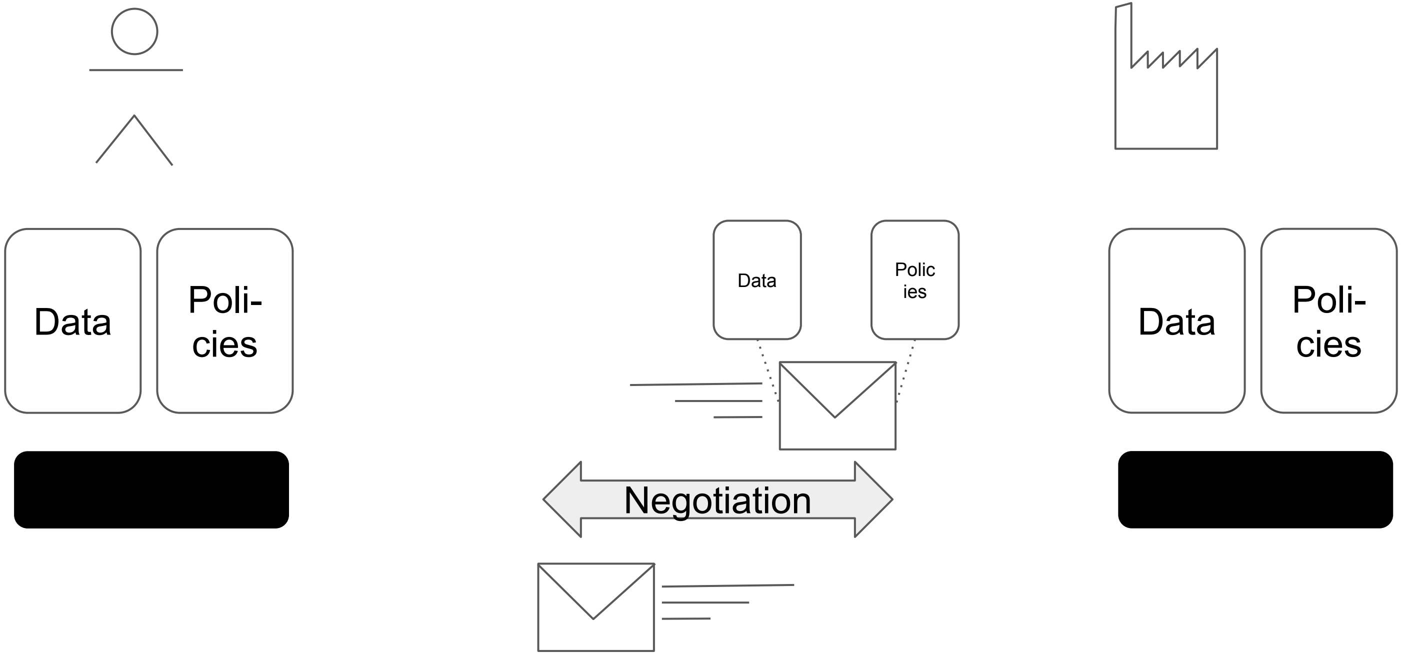


Techno-legal systems? The 10.000 meter view



Icons from Freepik, Iconiyo, Wahyu.Setyanto, Anggara, Freepik

Techno-legal systems? The 10.000 meter view



Automated Policy Negotiation: A Four-Course Meal

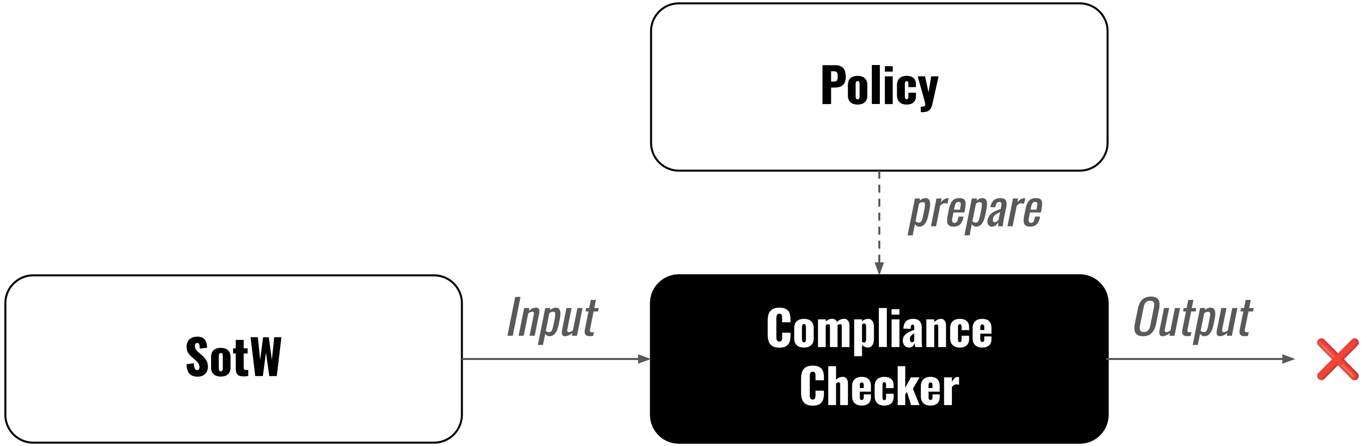
1. Challenges
2. Policies as Computer Programs
3. Related Work
4. Conclusions & Future Work

Automated Policy Negotiation: A Four-Course Meal

What are typical tasks these machines should be capable to do?

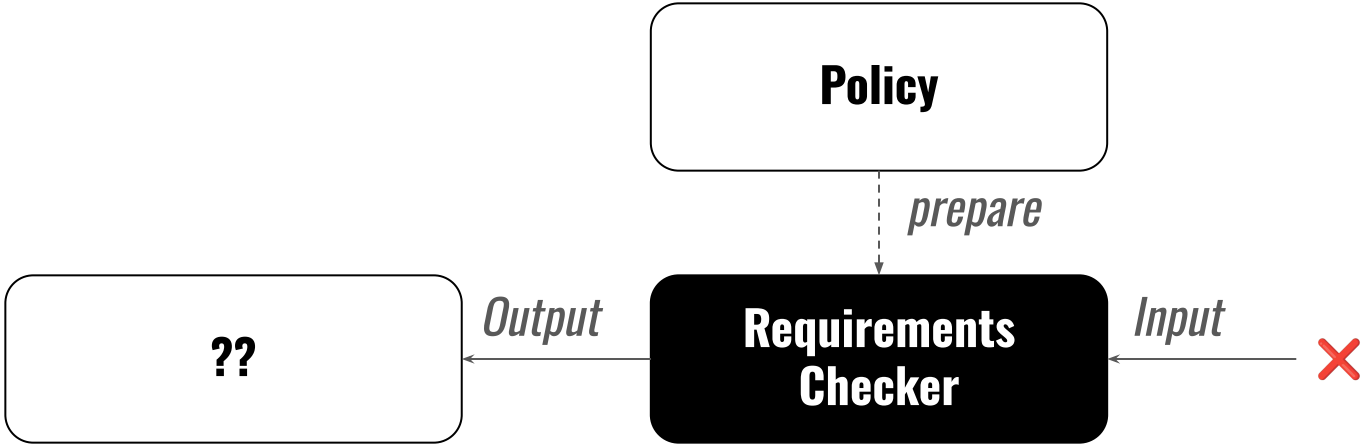
Compliance checking

Given an policy as input, the machine should be able to calculate in a particular state of the world complies with the policy norms.



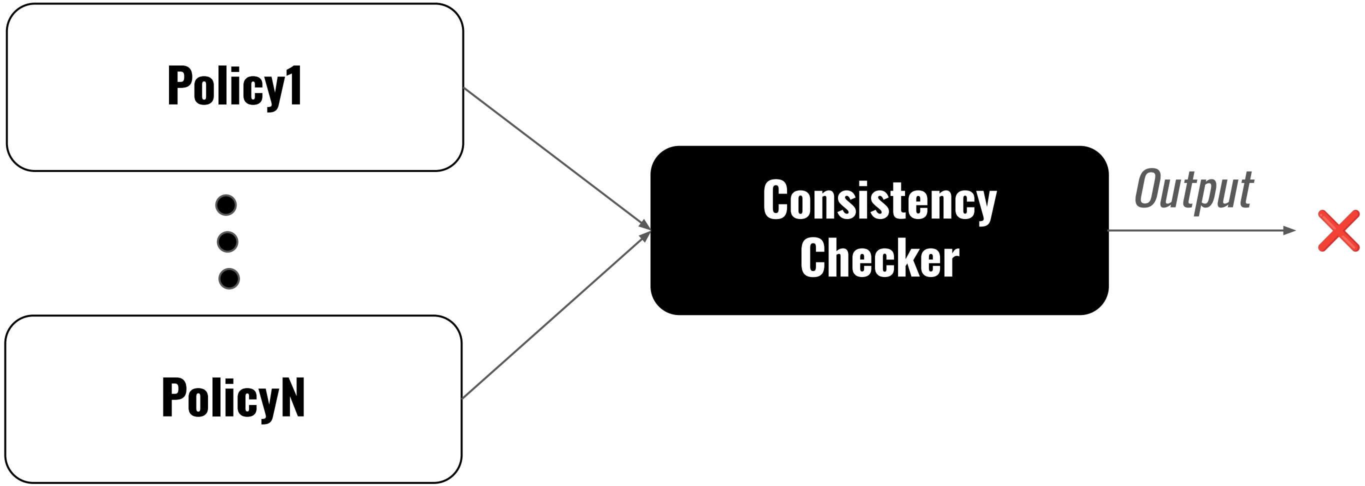
Requirements checking

This is the inverse process of the previous compliance checking. If a "computer says no," we need to understand why and what actions we can take to change the "no" into a "yes."



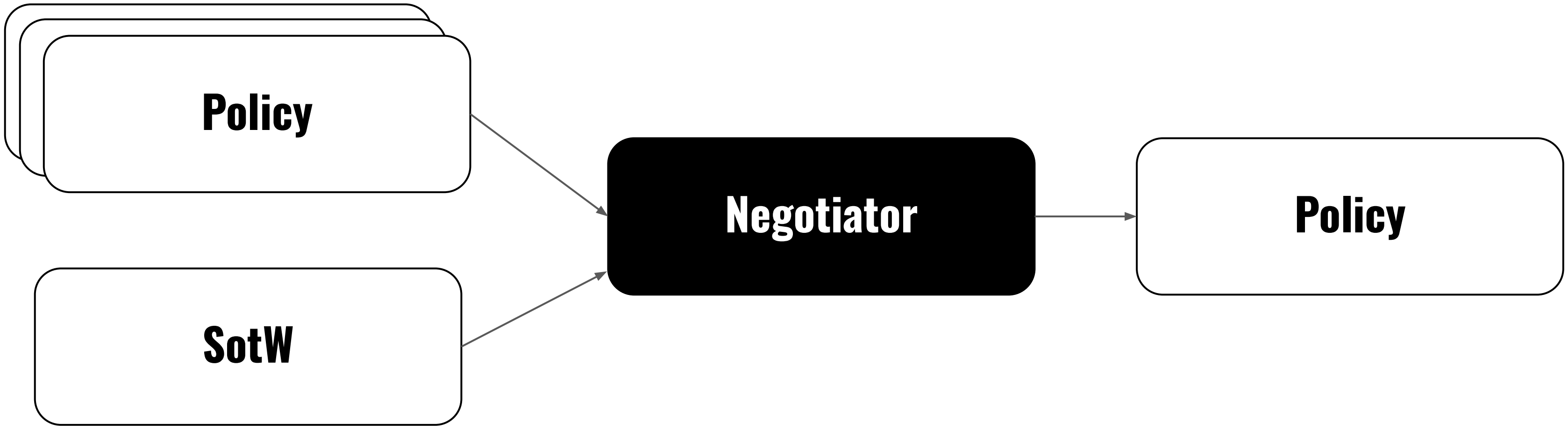
Consistency checking

This machine get policies as input and it needs to know if there are inconsistencies between these policies. Inconsistent policies are void and useless and potential dangerous if not detected.



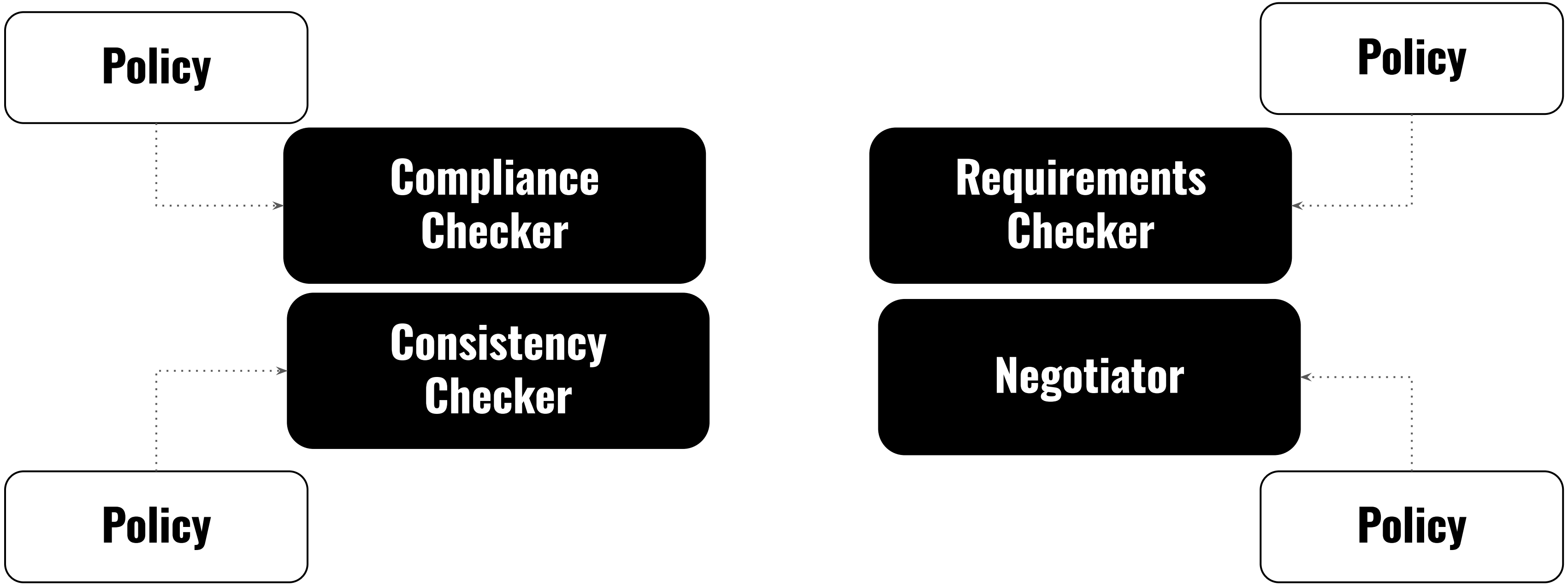
Negotiation

The negotiation process requires a combination of customer policies, company policies, and potentially a state of the world to arrive on a new policy for a particular use-case.



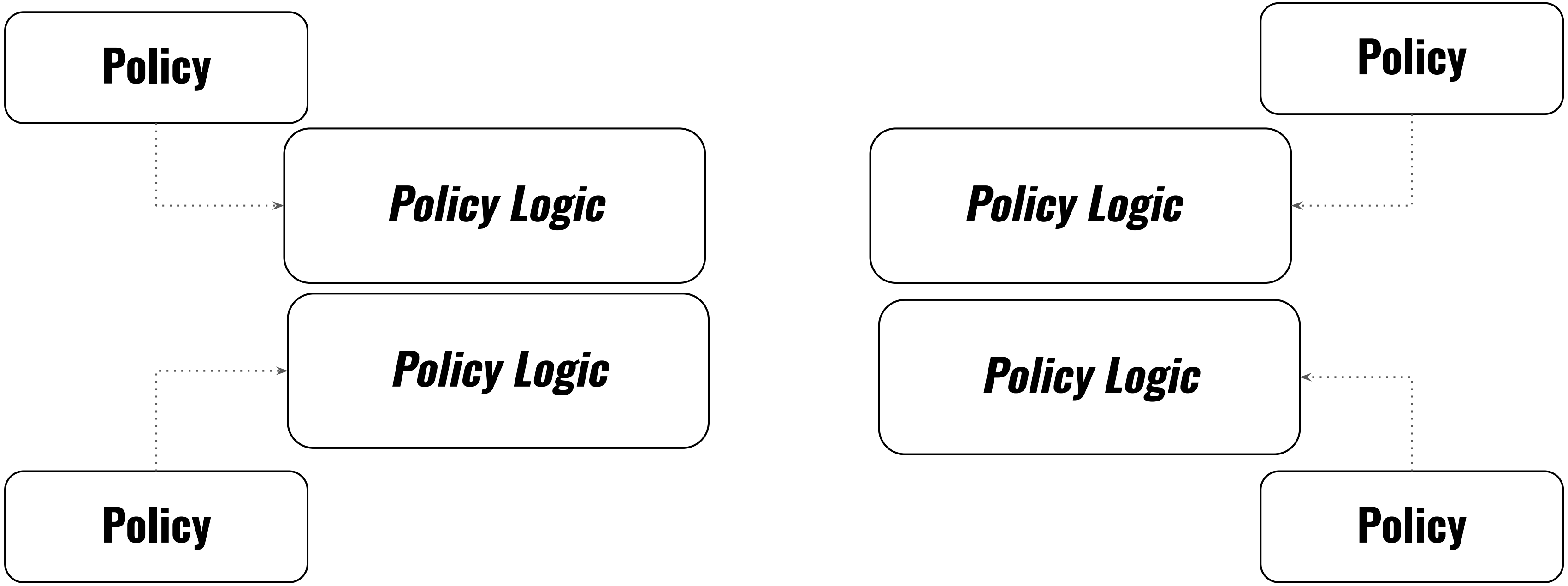
These four challenges are related

The logic as expressed in the policies need to commute between applications.



These four challenges are related

The logic as expressed in the policies need to commute between applications.



Automated Policy Negotiation: A Four-Course Meal

1. Challenges
2. **Policies as Computer Programs**
3. Related Work
4. Conclusions & Future Work

Policies are, in effect, computer programs

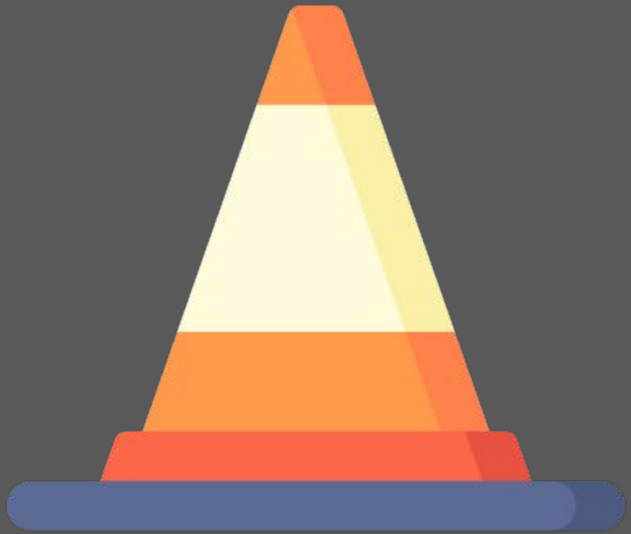
- Policy logic is currently defined by their implementations.

- There were high hopes that Semantic Web logic would automatically provide us common logic suitable for expressing the richness of our policy languages.

- However, in effect, what we see is a balancing act:
  - Implementing the requirements of deontic+defeasilble+(more?) in a particular framework
  - Requiring multiple of these framework, each with their own choice of what logic to implement to be interoperable
  - Making this all scalable

Automated Policy Negotiation: A Four-Course Meal

1. Challenges
2. Policies as Computer Programs
3. **Related Work**
4. Conclusions & Future Work



WORK in PROGRESS

Challenge	References	Language	Formalization
Compliance checking	Wieringa 1994 [15]	Libary Regulations	Prolog
	Chong 2006 [16]	LicenseScript	Prolog
	Gandon 2017 [17]	LegalRuleML	OWL+SPARQL
	Fornara 2019 [18]	ODRL	OWL+RIF
	De Vos 2019 [20]	InstAL	ASP
	Kirrane 2021 [19]	SPECIAL	OWL
	Van Binsbergen 2022 [22]	eFLINT	Haskell
	Francesconi 2023 [11]	LegalRuleML	OWL+SPARCL+SHACL
	Robaldo 2023 [21]	LegalRuleML	ASP/SHACL
	Slabbinck 2025 [13]	ODRL	Prolog
Consistency checking	Gangadharan 2007 [23]	ODRL/L(S)	*Theory*
	Sensoy 2012 [24]	OWL-POLAR	OWL+Pellet
	Villata 2012 [25]	CC	*Theory*
	Rotolo 2013 [26]	CC,ODC,GNU,...	*Theory*
	Costantino 2018 [14]	CNL4DSA	Maude
	Pellegrini 2018 [27]	ODRL	ASP
	Inclezan 2023 [28]	$\mathcal{AOPL}$	ASP
Requirements checking	Pandit 2018 [29]	GDPRov, GDPRtEXT	SPARQL
	Okoyomon 2019 [30]	*Consent screens*	Regex
	Hamdani 2021 [31]	*Privacy policies*	NLP
	Akaichi 2023 [10]	ODRL	OWL+SHACL
	Adhikari 2025 [32]	*Survey paper*	-
Negotiation	Baarslag 2017 [33]	-	*Protocol*
	Kiruthika 2020 [12]	*Survey paper*	-
	Yumasak 2024 [34]	ODRL	IDSA
	IDSA 2025 [35]	ODRL	*Protocol*

Challenge	References	Language	Formalization
Compliance checking	Wieringa 1994 [15]	Libary Regulations	Prolog
	Chong 2006 [16]	LicenseScript	Prolog
	Gandon 2017 [17]	LegalRuleML	OWL+SPARQL
	Fornara 2019 [18]	ODRL	OWL+RIF
	De Vos 2019 [20]	InstAL	ASP
	Kirrane 2021 [19]	SPECIAL	OWL
	Van Binsbergen 2022 [22]	eFLINT	Haskell
	Francesconi 2023 [11]	LegalRuleML	OWL+SPARCL+SHACL
	Robaldo 2023 [21]	LegalRuleML	ASP/SHACL
	Slabbinck 2025 [13]	ODRL	Prolog
Consistency checking	Gangadharan 2007 [23]	ODRL/L(S)	*Theory*
	Sensoy 2012 [24]	OWL-POLAR	OWL+Pellet
	Villata 2012 [25]	CC	*Theory*
	Rotolo 2013 [26]	CC,ODC,GNU,...	*Theory*
	Costantino 2018 [14]	CNL4DSA	Maude
	Pellegrini 2018 [27]	ODRL	ASP
	Inclezan 2023 [28]	$\mathscr{AOPL}$	ASP
Requirements checking	Pandit 2018 [29]	GDPRov, GDPRtEXT	SPARQL
	Okoyomon 2019 [30]	*Consent screens*	Regex
	Hamdani 2021 [31]	*Privacy policies*	NLP
	Akaichi 2023 [10]	ODRL	OWL+SHACL
	Adhikari 2025 [32]	*Survey paper*	-
Negotiation	Baarslag 2017 [33]	-	*Protocol*
	Kiruthika 2020 [12]	*Survey paper*	-
	Yumasak 2024 [34]	ODRL	IDSA
	IDSA 2025 [35]	ODRL	*Protocol*

Challenge	References	Language	Formalization
Compliance checking	Wieringa 1994 [15]	Libary Regulations	Prolog
	Chong 2006 [16]	LicenseScript	Prolog
	Gandon 2017 [17]	LegalRuleML	OWL+SPARQL
	Fornara 2019 [18]	ODRL	OWL+RIF
	De Vos 2019 [20]	InstAL	ASP
	Kirrane 2021 [19]	SPECIAL	OWL
	Van Binsbergen 2022 [22]	eFLINT	Haskell
	Francesconi 2023 [11]	LegalRuleML	OWL+SPARCL+SHACL
	Robaldo 2023 [21]	LegalRuleML	ASP/SHACL
	Slabbinck 2025 [13]	ODRL	Prolog
Consistency checking	Gangadharan 2007 [23]	ODRL/L(S)	*Theory*
	Sensoy 2012 [24]	OWL-POLAR	OWL+Pellet
	Villata 2012 [25]	CC	*Theory*
	Rotolo 2013 [26]	CC,ODC,GNU,...	*Theory*
	Costantino 2018 [14]	CNL4DSA	Maude
	Pellegrini 2018 [27]	ODRL	ASP
	Inclezan 2023 [28]	$\mathscr{AOPL}$	ASP
Requirements checking	Pandit 2018 [29]	GDPRov, GDPRtEXT	SPARQL
	Okoyomon 2019 [30]	*Consent screens*	Regex
	Hamdani 2021 [31]	*Privacy policies*	NLP
	Akaichi 2023 [10]	ODRL	OWL+SHACL
	Adhikari 2025 [32]	*Survey paper*	-
Negotiation	Baarslag 2017 [33]	-	*Protocol*
	Kiruthika 2020 [12]	*Survey paper*	-
	Yumasak 2024 [34]	ODRL	IDSA
	IDSA 2025 [35]	ODRL	*Protocol*

- Early implementations based on **logic programming** languages, e.g., Prolog
- **Rise of the Semantic Web languages**: challenges in covering all the deontic logic requirements, e.g., prohibition requires some form of negation
- **Combination of languages** has the potential to provide the necessary expressivity

Challenge	References	Language	Formalization
Compliance checking	Wieringa 1994 [15]	Libary Regulations	Prolog
	Chong 2006 [16]	LicenseScript	Prolog
	Gandon 2017 [17]	LegalRuleML	OWL+SPARQL
	Fornara 2019 [18]	ODRL	OWL+RIF
	De Vos 2019 [20]	InstAL	ASP
	Kirrane 2021 [19]	SPECIAL	OWL
	Van Binsbergen 2022 [22]	eFLINT	Haskell
	Francesconi 2023 [11]	LegalRuleML	OWL+SPARCL+SHACL
	Robaldo 2023 [21]	LegalRuleML	ASP/SHACL
	Slabbinck 2025 [13]	ODRL	Prolog
Consistency checking	Gangadharan 2007 [23]	ODRL/L(S)	*Theory*
	Sensoy 2012 [24]	OWL-POLAR	OWL+Pellet
	Villata 2012 [25]	CC	*Theory*
	Rotolo 2013 [26]	CC,ODC,GNU,...	*Theory*
	Costantino 2018 [14]	CNL4DSA	Maude
	Pellegrini 2018 [27]	ODRL	ASP
	Inclezan 2023 [28]	$\mathscr{AOPL}$	ASP
Requirements checking	Pandit 2018 [29]	GDPRov, GDPRtEXT	SPARQL
	Okoyomon 2019 [30]	*Consent screens*	Regex
	Hamdani 2021 [31]	*Privacy policies*	NLP
	Akaichi 2023 [10]	ODRL	OWL+SHACL
	Adhikari 2025 [32]	*Survey paper*	-
Negotiation	Baarslag 2017 [33]	-	*Protocol*
	Kiruthika 2020 [12]	*Survey paper*	-
	Yumasak 2024 [34]	ODRL	IDSA
	IDSA 2025 [35]	ODRL	*Protocol*

Challenge	References	Language	Formalization
Compliance checking	Wieringa 1994 [15]	Libary Regulations	Prolog
	Chong 2006 [16]	LicenseScript	Prolog
	Gandon 2017 [17]	LegalRuleML	OWL+SPARQL
	Fornara 2019 [18]	ODRL	OWL+RIF
	De Vos 2019 [20]	InstAL	ASP
	Kirrane 2021 [19]	SPECIAL	OWL
	Van Binsbergen 2022 [22]	eFLINT	Haskell
	Francesconi 2023 [11]	LegalRuleML	OWL+SPARCL+SHACL
	Robaldo 2023 [21]	LegalRuleML	ASP/SHACL
	Slabbinck 2025 [13]	ODRL	Prolog
Consistency checking	Gangadharan 2007 [23]	ODRL/L(S)	*Theory*
	Sensoy 2012 [24]	OWL-POLAR	OWL+Pellet
	Villata 2012 [25]	CC	*Theory*
	Rotolo 2013 [26]	CC,ODC,GNU,...	*Theory*
	Costantino 2018 [14]	CNL4DSA	Maude
	Pellegrini 2018 [27]	ODRL	ASP
	Inclezan 2023 [28]	$\mathcal{AOPL}$	ASP
Requirements checking	Pandit 2018 [29]	GDPRov, GDPRtEXT	SPARQL
	Okoyomon 2019 [30]	*Consent screens*	Regex
	Hamdani 2021 [31]	*Privacy policies*	NLP
	Akaichi 2023 [10]	ODRL	OWL+SHACL
	Adhikari 2025 [32]	*Survey paper*	-
Negotiation	Baarslag 2017 [33]	-	*Protocol*
	Kiruthika 2020 [12]	*Survey paper*	-
	Yumasak 2024 [34]	ODRL	IDSA
	IDSA 2025 [35]	ODRL	*Protocol*

- Focus on software **licenses**
- Maude is a **declarative programming environment** used for specifying and analyzing formal models of systems, including consistency checks
- ASP to find **inconsistencies**, **underspecified**, and **ambiguities**

Inconsistencies – rules that contradict each other

Underspecified – rules that never trigger

Ambiguities – rules that permit an action in one possible state of the world but forbid it in another possible state of the world

Challenge	References	Language	Formalization
Compliance checking	Wieringa 1994 [15]	Libary Regulations	Prolog
	Chong 2006 [16]	LicenseScript	Prolog
	Gandon 2017 [17]	LegalRuleML	OWL+SPARQL
	Fornara 2019 [18]	ODRL	OWL+RIF
	De Vos 2019 [20]	InstAL	ASP
	Kirrane 2021 [19]	SPECIAL	OWL
	Van Binsbergen 2022 [22]	eFLINT	Haskell
	Francesconi 2023 [11]	LegalRuleML	OWL+SPARCL+SHACL
	Robaldo 2023 [21]	LegalRuleML	ASP/SHACL
	Slabbinck 2025 [13]	ODRL	Prolog
Consistency checking	Gangadharan 2007 [23]	ODRL/L(S)	*Theory*
	Sensoy 2012 [24]	OWL-POLAR	OWL+Pellet
	Villata 2012 [25]	CC	*Theory*
	Rotolo 2013 [26]	CC,ODC,GNU,...	*Theory*
	Costantino 2018 [14]	CNL4DSA	Maude
	Pellegrini 2018 [27]	ODRL	ASP
	Inclezan 2023 [28]	$\mathscr{AOPL}$	ASP
Requirements checking	Pandit 2018 [29]	GDPRov, GDPRtEXT	SPARQL
	Okoyomon 2019 [30]	*Consent screens*	Regex
	Hamdani 2021 [31]	*Privacy policies*	NLP
	Akaichi 2023 [10]	ODRL	OWL+SHACL
	Adhikari 2025 [32]	*Survey paper*	-
Negotiation	Baarslag 2017 [33]	-	*Protocol*
	Kiruthika 2020 [12]	*Survey paper*	-
	Yumasak 2024 [34]	ODRL	IDSA
	IDSA 2025 [35]	ODRL	*Protocol*

Challenge	References	Language	Formalization
Compliance checking	Wieringa 1994 [15]	Libary Regulations	Prolog
	Chong 2006 [16]	LicenseScript	Prolog
	Gandon 2017 [17]	LegalRuleML	OWL+SPARQL
	Fornara 2019 [18]	ODRL	OWL+RIF
	De Vos 2019 [20]	InstAL	ASP
	Kirrane 2021 [19]	SPECIAL	OWL
	Van Binsbergen 2022 [22]	eFLINT	Haskell
	Francesconi 2023 [11]	LegalRuleML	OWL+SPARCL+SHACL
	Robaldo 2023 [21]	LegalRuleML	ASP/SHACL
	Slabbinck 2025 [13]	ODRL	Prolog
Consistency checking	Gangadharan 2007 [23]	ODRL/L(S)	*Theory*
	Sensoy 2012 [24]	OWL-POLAR	OWL+Pellet
	Villata 2012 [25]	CC	*Theory*
	Rotolo 2013 [26]	CC,ODC,GNU,...	*Theory*
	Costantino 2018 [14]	CNL4DSA	Maude
	Pellegrini 2018 [27]	ODRL	ASP
	Inclezan 2023 [28]	$\mathcal{AOPL}$	ASP
Requirements checking	Pandit 2018 [29]	GDPRov, GDPRtEXT	SPARQL
	Okoyomon 2019 [30]	*Consent screens*	Regex
	Hamdani 2021 [31]	*Privacy policies*	NLP
	Akaichi 2023 [10]	ODRL	OWL+SHACL
	Adhikari 2025 [32]	*Survey paper*	-
Negotiation	Baarslag 2017 [33]	-	*Protocol*
	Kiruthika 2020 [12]	*Survey paper*	-
	Yumasak 2024 [34]	ODRL	IDSA
	IDSA 2025 [35]	ODRL	*Protocol*

Three approaches in the literature:

- **Syntactical analysis** of the policy language, e.g., matching of human- and machine-readable representation
- Analysis of the **deeper underlying logic** of the policy language, e.g., using deterministic processes to formalize and analyze the policies
- **Non-deterministic processing**, e.g., machine learning to analyse the policies

Challenge	References	Language	Formalization
Compliance checking	Wieringa 1994 [15]	Libary Regulations	Prolog
	Chong 2006 [16]	LicenseScript	Prolog
	Gandon 2017 [17]	LegalRuleML	OWL+SPARQL
	Fornara 2019 [18]	ODRL	OWL+RIF
	De Vos 2019 [20]	InstAL	ASP
	Kirrane 2021 [19]	SPECIAL	OWL
	Van Binsbergen 2022 [22]	eFLINT	Haskell
	Francesconi 2023 [11]	LegalRuleML	OWL+SPARCL+SHACL
	Robaldo 2023 [21]	LegalRuleML	ASP/SHACL
	Slabbinck 2025 [13]	ODRL	Prolog
Consistency checking	Gangadharan 2007 [23]	ODRL/L(S)	*Theory*
	Sensoy 2012 [24]	OWL-POLAR	OWL+Pellet
	Villata 2012 [25]	CC	*Theory*
	Rotolo 2013 [26]	CC,ODC,GNU,...	*Theory*
	Costantino 2018 [14]	CNL4DSA	Maude
	Pellegrini 2018 [27]	ODRL	ASP
	Inclezan 2023 [28]	$\mathcal{AOPL}$	ASP
Requirements checking	Pandit 2018 [29]	GDPRov, GDPRtEXT	SPARQL
	Okoyomon 2019 [30]	*Consent screens*	Regex
	Hamdani 2021 [31]	*Privacy policies*	NLP
	Akaichi 2023 [10]	ODRL	OWL+SHACL
	Adhikari 2025 [32]	*Survey paper*	-
Negotiation	Baarslag 2017 [33]	-	*Protocol*
	Kiruthika 2020 [12]	*Survey paper*	-
	Yumasak 2024 [34]	ODRL	IDSA
	IDSA 2025 [35]	ODRL	*Protocol*

Challenge	References	Language	Formalization
Compliance checking	Wieringa 1994 [15]	Libary Regulations	Prolog
	Chong 2006 [16]	LicenseScript	Prolog
	Gandon 2017 [17]	LegalRuleML	OWL+SPARQL
	Fornara 2019 [18]	ODRL	OWL+RIF
	De Vos 2019 [20]	InstAL	ASP
	Kirrane 2021 [19]	SPECIAL	OWL
	Van Binsbergen 2022 [22]	eFLINT	Haskell
	Francesconi 2023 [11]	LegalRuleML	OWL+SPARCL+SHACL
	Robaldo 2023 [21]	LegalRuleML	ASP/SHACL
	Slabbinck 2025 [13]	ODRL	Prolog
Consistency checking	Gangadharan 2007 [23]	ODRL/L(S)	*Theory*
	Sensoy 2012 [24]	OWL-POLAR	OWL+Pellet
	Villata 2012 [25]	CC	*Theory*
	Rotolo 2013 [26]	CC,ODC,GNU,...	*Theory*
	Costantino 2018 [14]	CNL4DSA	Maude
	Pellegrini 2018 [27]	ODRL	ASP
	Inclezan 2023 [28]	$\mathscr{AOPL}$	ASP
Requirements checking	Pandit 2018 [29]	GDPRov, GDPRtEXT	SPARQL
	Okoyomon 2019 [30]	*Consent screens*	Regex
	Hamdani 2021 [31]	*Privacy policies*	NLP
	Akaichi 2023 [10]	ODRL	OWL+SHACL
	Adhikari 2025 [32]	*Survey paper*	-
Negotiation	Baarslag 2017 [33]	-	*Protocol*
	Kiruthika 2020 [12]	*Survey paper*	-
	Yumasak 2024 [34]	ODRL	IDSA
	IDSA 2025 [35]	ODRL	*Protocol*

- **Not many examples of fully automated agents** that can negotiate policies
- **IDSA has semi-automated** for contract negotiation
- Machines could be involved in providing **feedback on the consistency of negotiated policies**, **explaining the consequences of the negotiated policies**, and running some sample scenarios

Automated Policy Negotiation: A Four-Course Meal

1. Challenges
2. Policies as Computer Programs
3. Related Work
4. **Conclusions & Future Work**

Conclusions & Future Work

- A four course meal is required to create a fully automated techno-legal system that does not rely on a "all or nothing" trust.
- What is blocking us is the definition of a formal policy logic.
  - This should not be left to implementers of policy languages.
- High hopes are/were that standard Semantic Web languages would provide the required deontic, defeasible, and other features of such a formal logic.
- There is a renewed interest in symbolic logic that does provide a richer set of logic features.
- Is it possible to have marriage between Semantic Web and a richer set of logic features?
  - In our group, we believe that Notation3 and RDF Surfaces, both based on first-order logic with powerful negation and a rich set of built-ins, could inspire such a recipe.

Automated Policy Negotiation: a Four-Course Meal

Patrick Hochstenbach, Beatriz Esteves, Ruben Verborgh

patrick.hochstenbach@ugent.be
beatriz.esteves@ugent.be