Towards a Multi-Agent Network for The Netherlands National Police

Bas Testerink

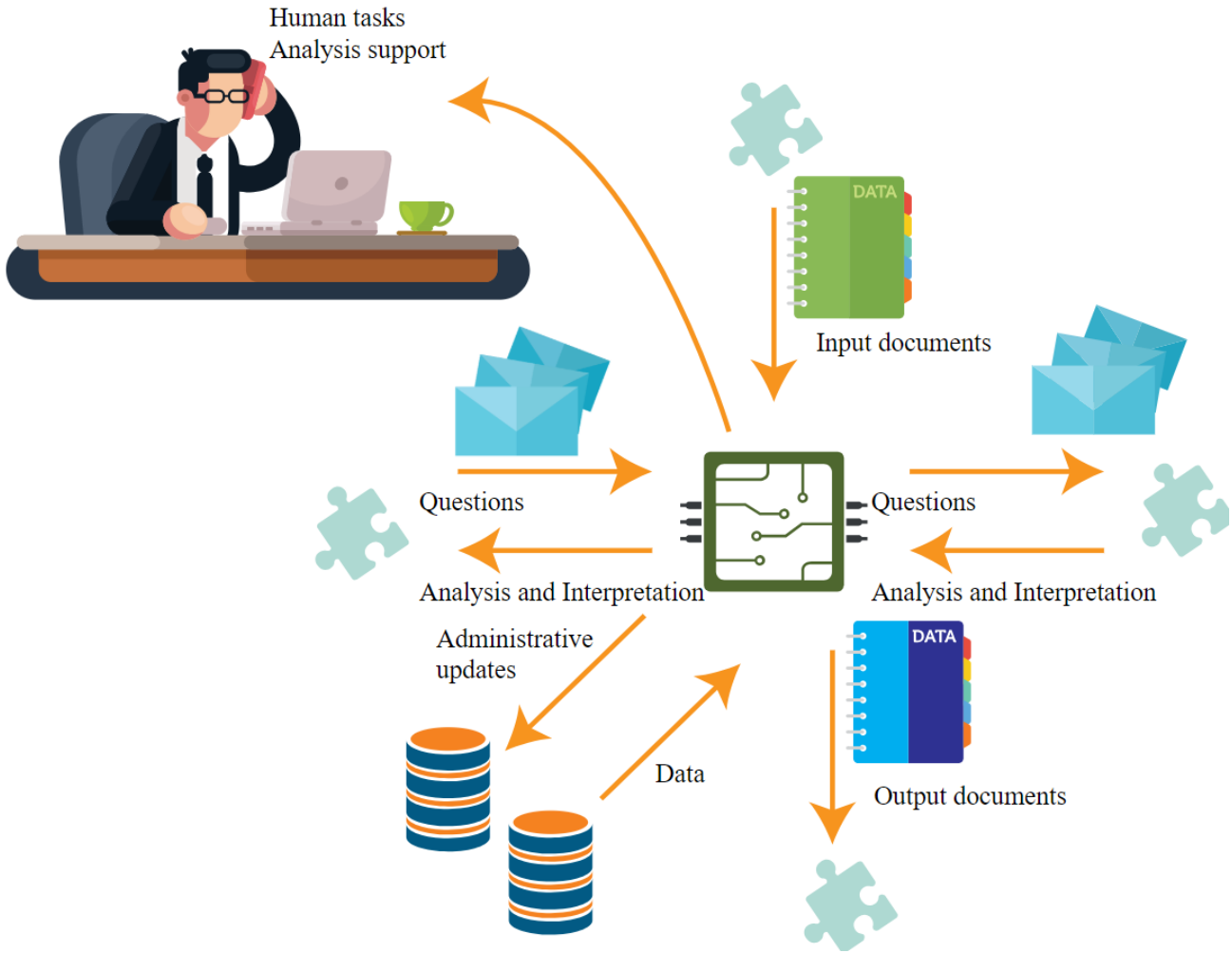
Background

Bas Testerink
The Netherlands National Police
Central Unit
Police Profession/Police A.I. Lab
What I do
I act as a bridge between the police and academia.

Decentralized A.I. supported decision-making in legal environments.

1. The general idea and examples
2. An MVP agent architecture
3. A toy example demonstration
4. Some considerations
5. If there's time left: technical details

Human-Machine Team



Human-Machine Team

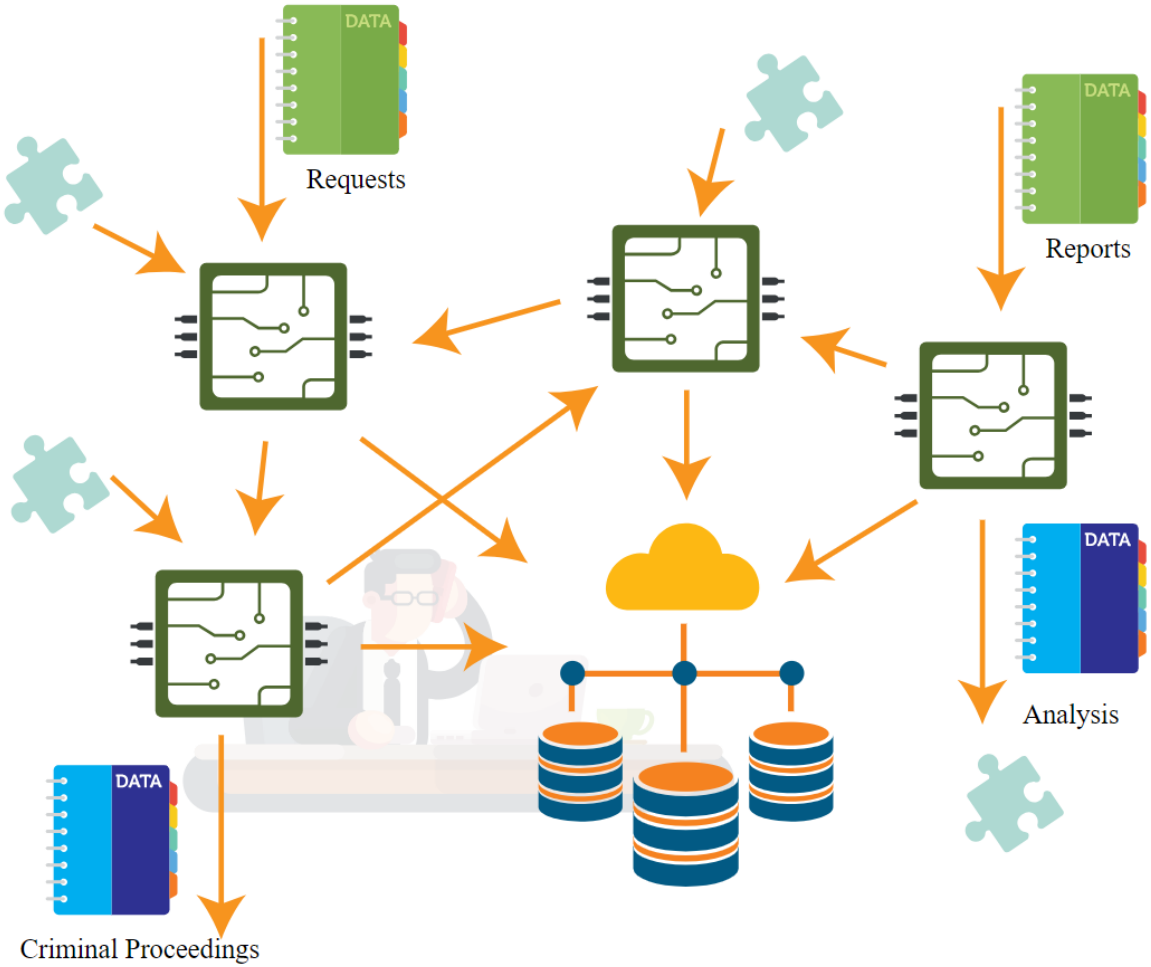
Capability enhancement:

- Multi-modal search
- Anomaly detection
- Unsupervised clustering of data

Decision-support and -automation:

- Information product production and processing
- Digitization vs analog processes with digital means
- Hypothesis testing (scenario reasoning, simulations)

The Police as a Human-Machine Team

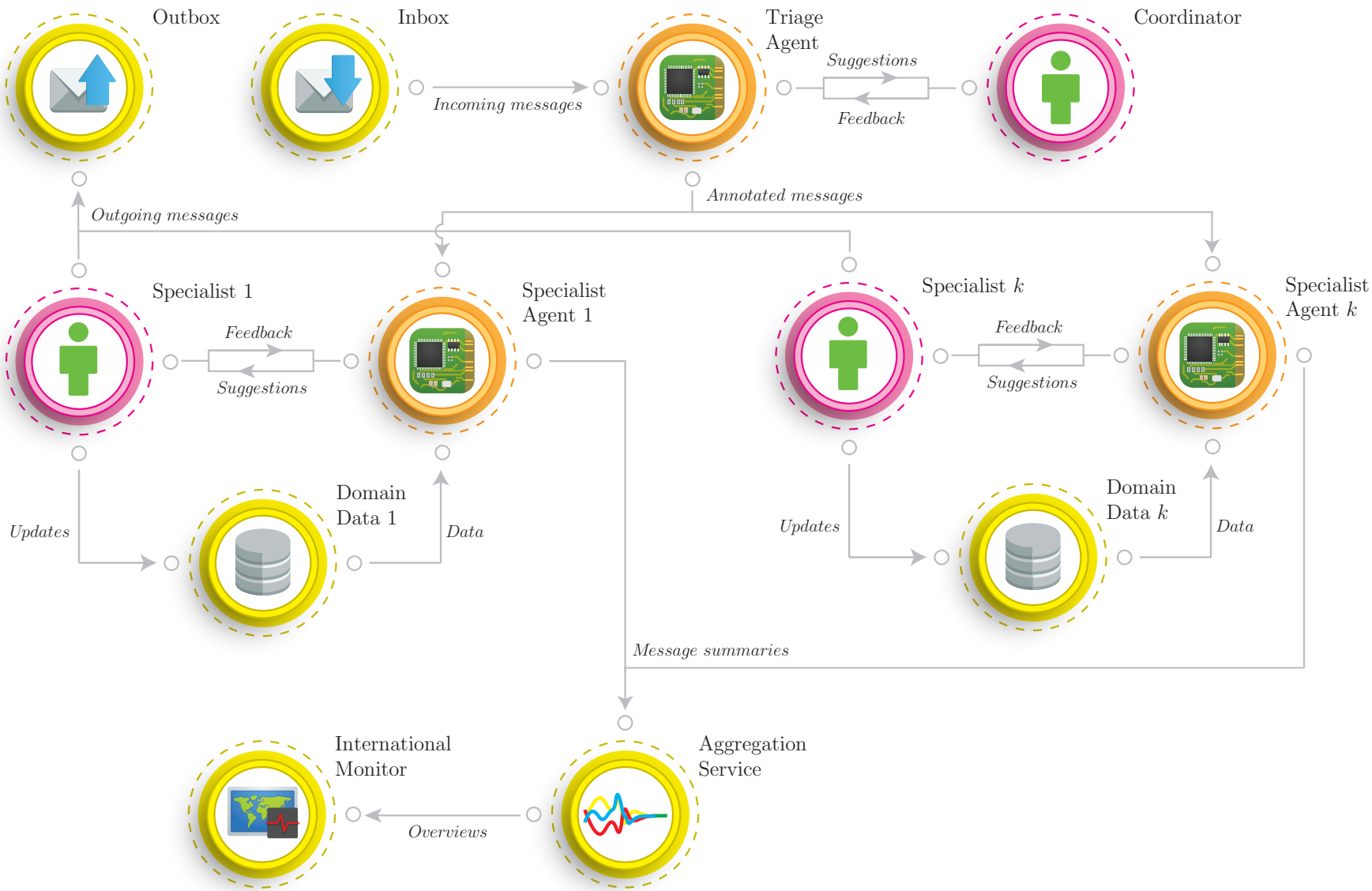


1 Project = 1 (Single/Multi) Agent System. The aim: semi-autonomous law-enforcement processes.

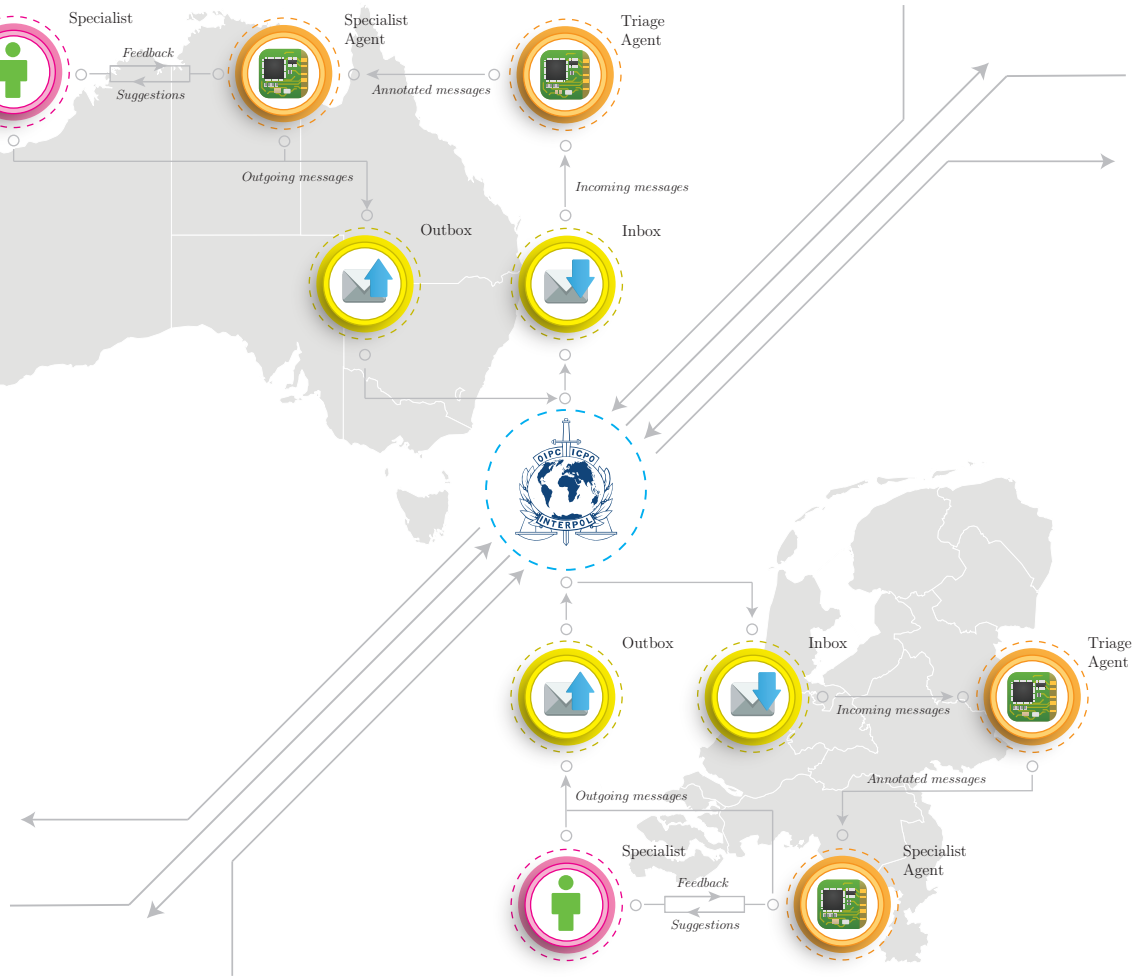
Example: online intake of reports

The agent judges the content of the report on legal context and asks about legally relevant but missing information.

Example: international request handling



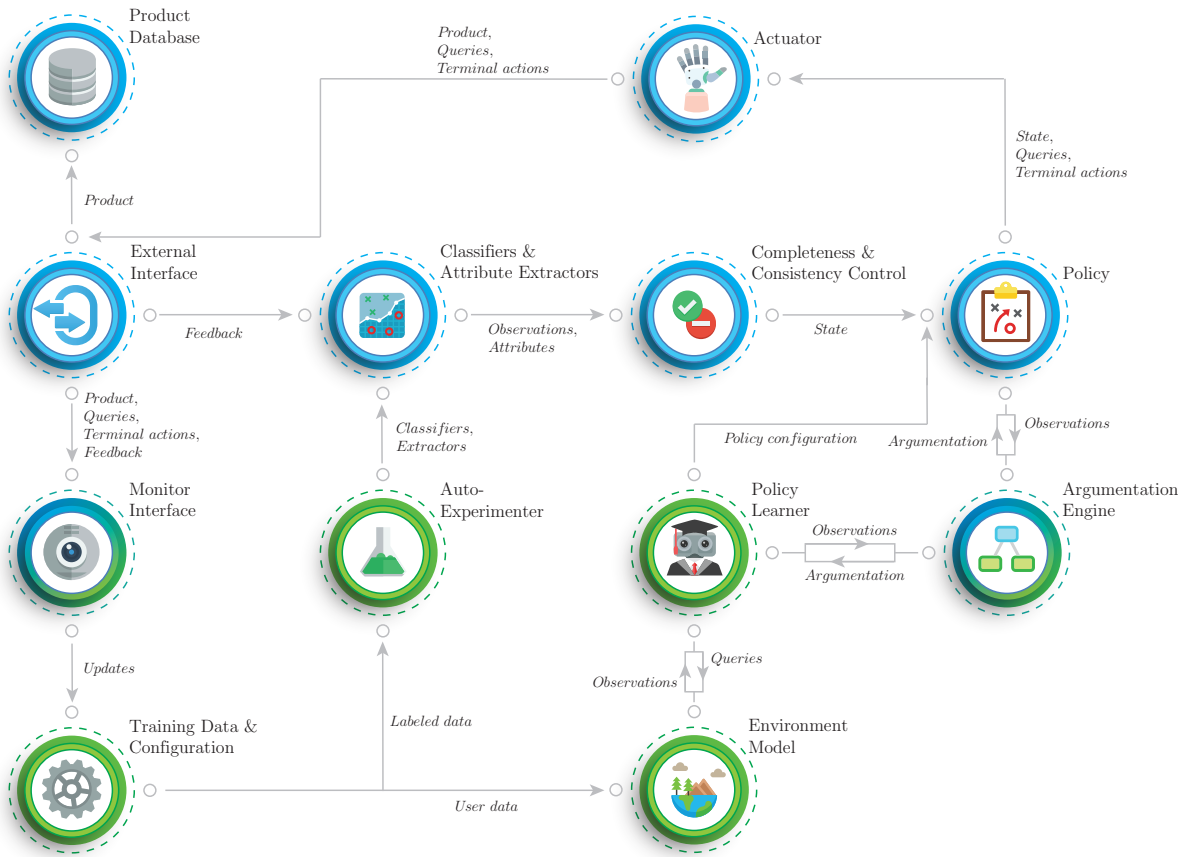
Example: international request handling



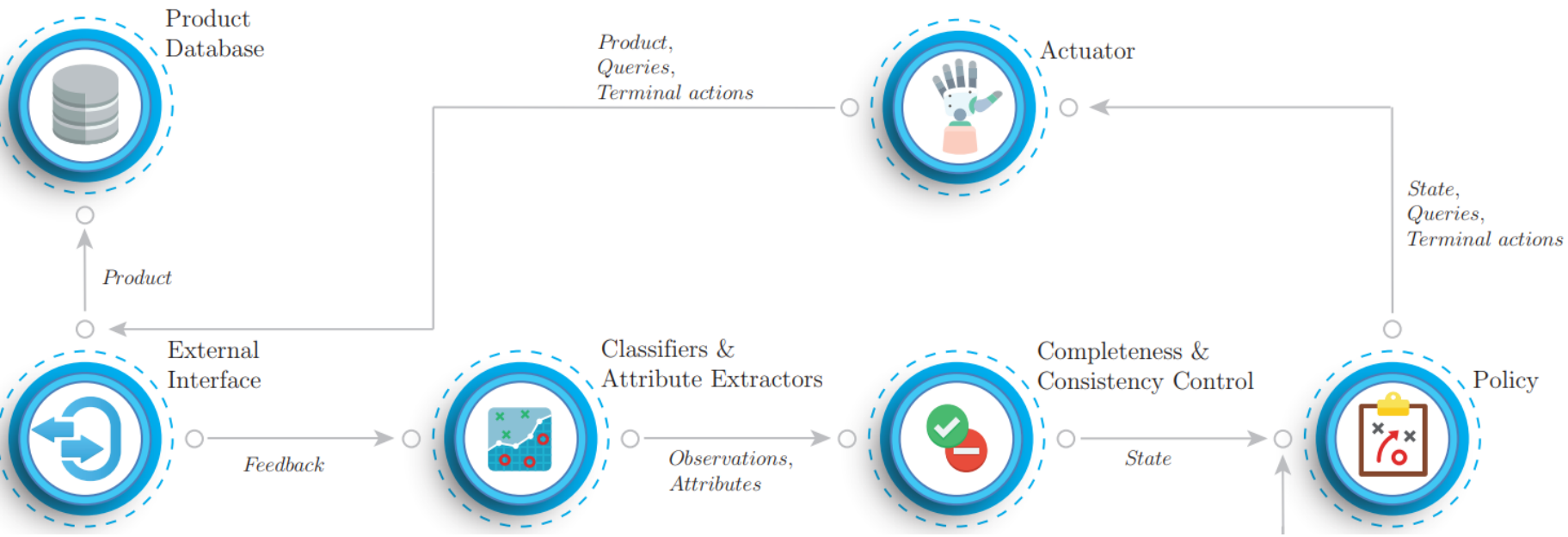
Demands

- Accuracy: Avoid high-impact errors
- Transparency: Explain automated decisions
- Controllable: Detectable and repairable errors
- Proportional: Minimization of resource spending and information/data gathering.

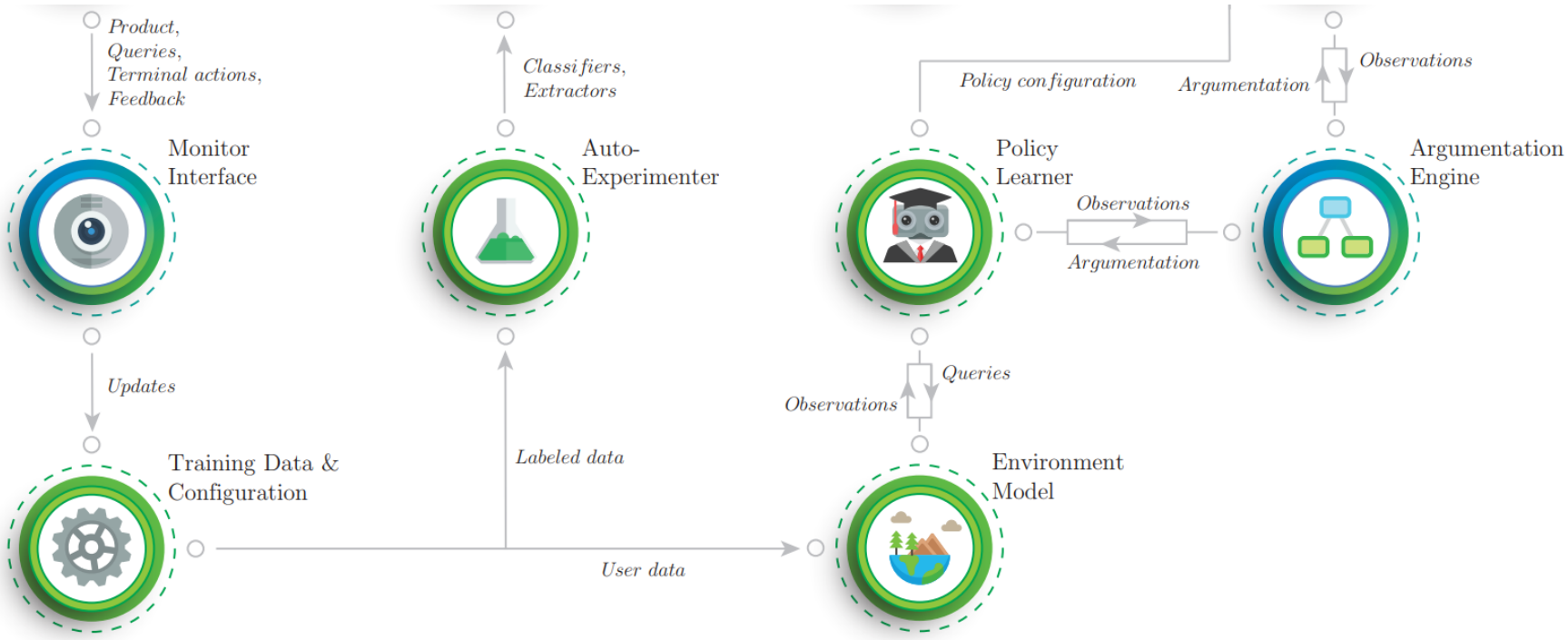
Machine Architecture



Machine Architecture



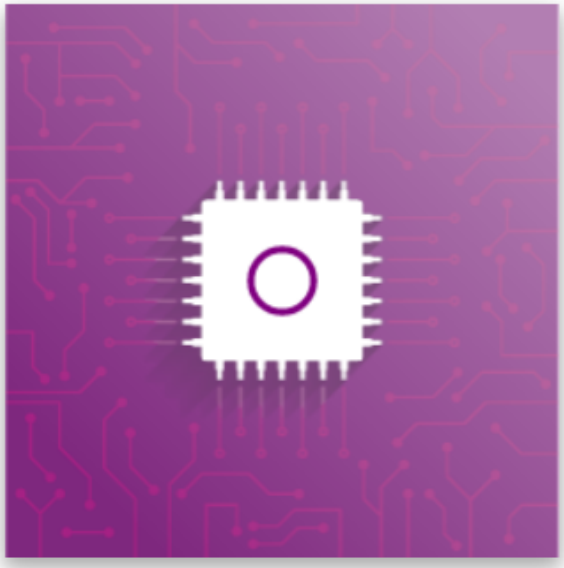
Machine Architecture



Demands

- Accuracy: Automated experimentation with supervised learning
- Transparency: Argumentation interwoven with decision making
- Controllable: Human-in-the-loop design
- Proportional: Reinforcement learning for policy optimization combined with legal relevancy based on argumentation

Toy Example: International Requests



Toy Example: International Requests

We are looking for Jan Jansen. We wonder whether he traveled to your country. He is wanted for murder. Please only reply in case of a positive hit.

Toy Example: International Requests

Language

English

Sentence Detect, Tokenizer and POS Tagger

- 0: We are looking for Jan Jansen .
- 1: We wonder whether he travelled to your country .
- 2: He is wanted for murder .
- 3: Please only reply in case of a positive hit .

Entity Detect

- person (1)

Toy Example: International Requests

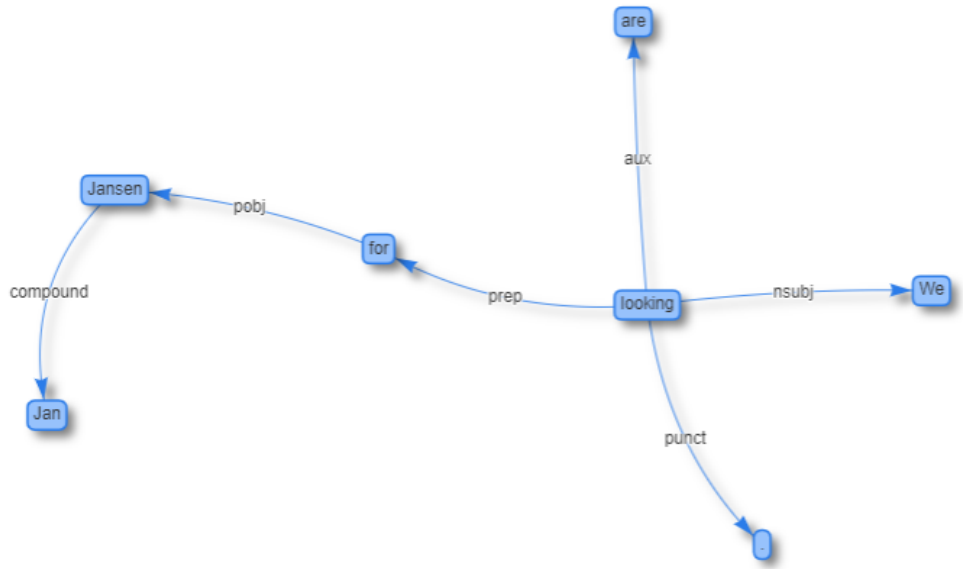
▼ 0: We are looking for Jan Jansen .

Sentence: "We are looking for Jan Jansen."

Token	POS Tag
We	PRON
are	VERB
looking	VERB
for	ADP
Jan	PROPN
Jansen	PROPN

Toy Example: International Requests

Classifier	Label
Request/Commit	Yes
o_positive_only_reply	No



Toy Example: International Requests

Input Classifiers

Classifier	Label
o_positive_only_reply	Yes
~o_positive_only_reply	No
o_crime_suspect	Yes
~o_crime_suspect	No
o_travel_question	Yes

Toy Example: International Requests

Argumentation

Argumentation Setup

Rules

```
o_positive_only_reply,~positive=>~t_feedback
~request=>~t_feedback
request,request_handled=>t_feedback
o_travel_question=>request
o_travel_question,~o_travel_hit=>request_handled
o_travel_question,~o_travel_hit=>~positive
o_travel_question,o_travel_hit=>positive
o_travel_question,o_travel_hit,~o_transfer=>request_handled
o_travel_question,o_travel_hit,o_transfer,o_discussed_travelling_with_peer=>request_handled
```

Topics

```
t_feedback
t_relay_intel
```

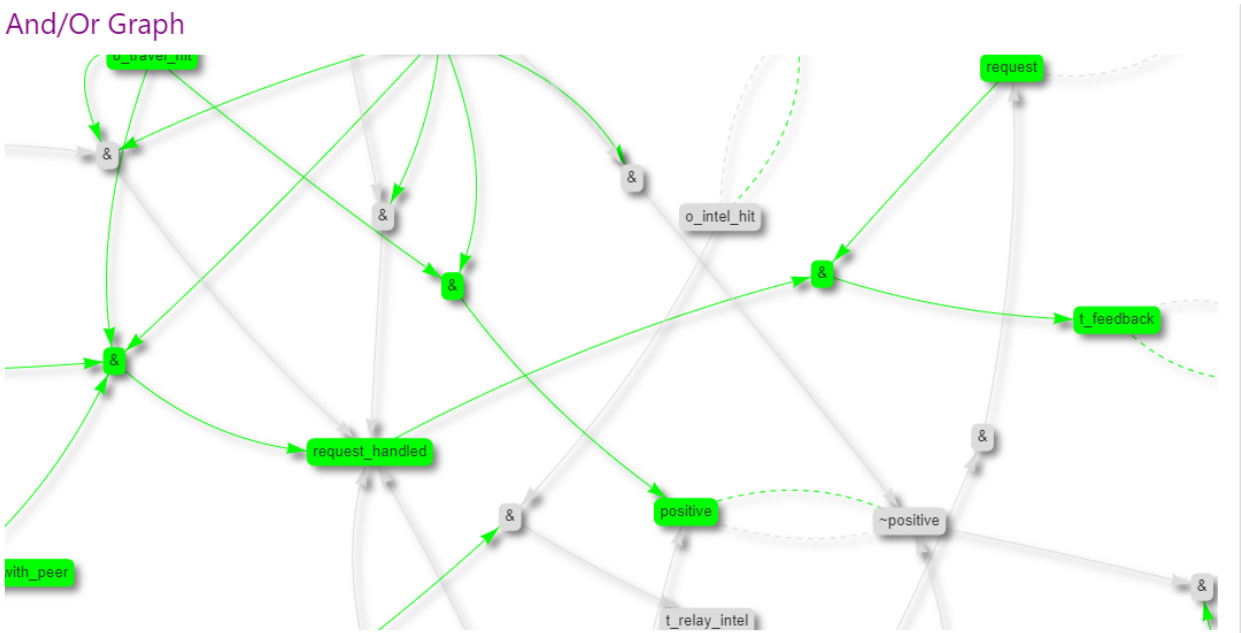
Observable

```
o_positive_only_reply
~o_positive_only_reply
o_travel_question
~o_travel_question
o_travel_hit
~o_travel_hit
o_crime_suspect
~o_crime_suspect
o_check_id
```

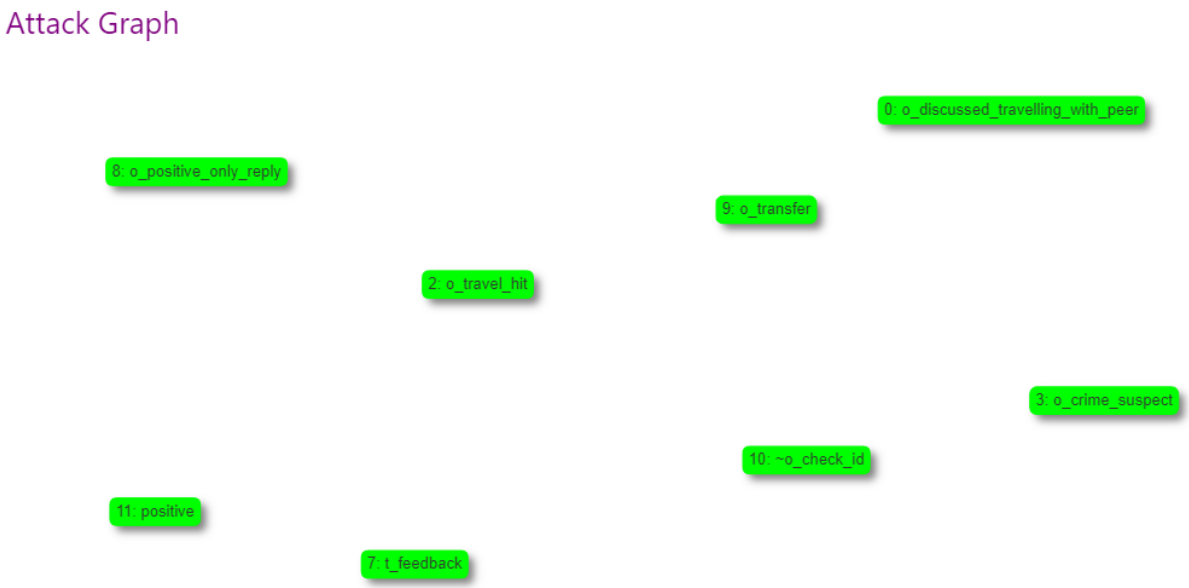
Observations

```
o_positive_only_reply
o_crime_suspect
o_travel_question
~o_check_id
```

Toy Example: International Requests



Toy Example: International Requests



Toy Example: International Requests

Relevance Map

Topic	Relevant
t_feedback	
t_relay_intel	

Toy Example: International Requests

Suggested Course of Action

Verdict: Send feedback

Toy Example: International Requests

Subject:

Re: Test subject

Content:

To whom it may concern

We have processed your request and have come to the conclusion that:

Jan Jansen indeed travelled to the Netherlands. Furthermore, Jan Jansen also transferred in The Netherlands. Please find attached additional information.

With kind regards,

Send

Toy Example: International Requests

Findings

Priority	NL Relevant	Theme	Intent
Normal	✔ Yes	Misc	Request for information

- No GBA-check was executed.
- The intel-check returned a negative result.
- A subject travelled to The Netherlands.
- A subject transferred in The Netherlands.

Toy Example: International Requests

Motivation

An defensible argument can be made for replying with feedback for the sender.

Feedback should be sent back to the sender if the sender made a requests for which the investigation is done. This was applicable because:

- The sender submitted a request. If the sender asked a travel question, then that counts-as a request. This was applicable because:
  - The sender asked a travel question. (this was observed by a classifier)
- The investigation for the sender is done. If the sender asked a travel question, and there was a travel hit including a transfer, and this transfer is discussed with the relevant peers, then the investigation is done. This was applicable because:
  - The sender asked a travel question. (this was observed by a classifier)
  - There was a positive hit on the travel database. (this was observed by a classifier)

Toy Example: International Requests

- All found transfers have been discussed with the relevant peers. (this was observed by a classifier)

It was not detected whether newly discovered intel should be relayed internally. There was no applicable rule.

1. The following rule was not applicable:
   - If the requests contains a crime suspect which occurs in the intel database, then intel should be relayed.

   Because: It was not detected whether there was a positive hit on the intel database. (this couldn't be observed)

Toy Example: International Requests

1: Initial action

DETAILS
Read the initial mail

RESULT SUMMARY
Read mail with subject: Test subject

2: Check travel database

DETAILS
?- check_db(Jan, Jansen, travel, Hit).

RESULT SUMMARY
Result: o_travel_hit

3: Check transfers

DETAILS
?- check_db(Jan, Jansen, transfer, Hit).

RESULT SUMMARY
Result: o_transfer

4: Discuss travelling with peer

DETAILS
?-conversation(0).

RESULT SUMMARY
Dummy conversation result.

5: Check intel

DETAILS
?- check_db(Jan, Jansen, intel, Hit).

RESULT SUMMARY
Result: ~o_intel_hit

6: STOP

DETAILS
Suggest course of action

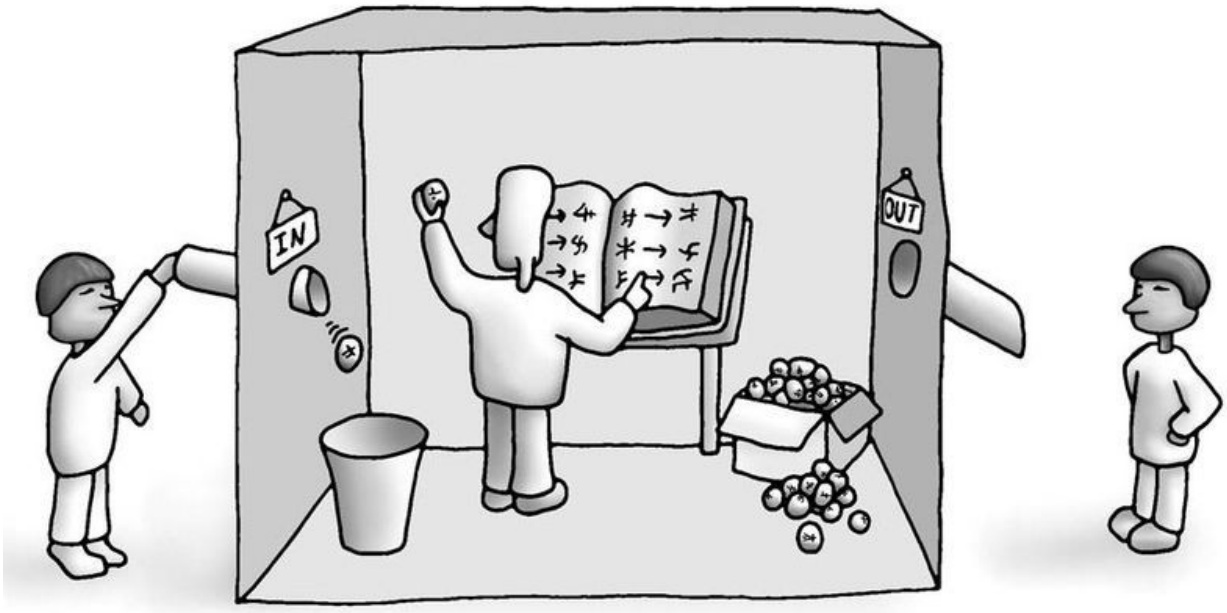
RESULT SUMMARY
Send feedback

A.I.: more than algorithms

- Maintenance considerations: what skills are required from ‘ground’ personnel and IT?
- How to re-educate people to train and correct A.I. instead of doing the job themselves?
- Who determines what the correct legal and ethical guidelines are on the software level?
- How to engage governmental, scientific, non-profit, hobbyist and business communities?
- What constitutes proper transparency, explainability, fairness and responsibility for an actual application?

Dangers

How does a computer executing a program differ from a person executing bureaucratic protocols?



Are we really entering a legal & ethical new era with A.I. or are we simply creating the steam engine equivalent for bureaucracy?

Importance of formal considerations

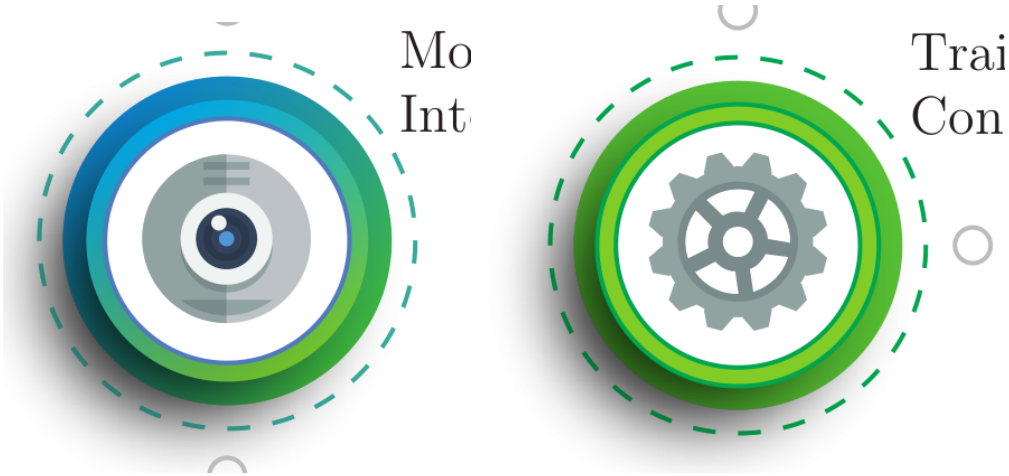
- Formal specification may allow for model-checking.
- Formally defined system traces may allow for runtime verification.
- Formal specification allows for a translation of formal theory to practice without burdening a logician/philosopher with coming up with a full real-world example.
- Likely future development: Legislation being translated to formal properties which then are verified in order to certify software that contains autonomous decision-making.

Concluding thoughts

- A.I. offers great opportunities and potential dangers.
- We need to experiment and fail-fast in order to learn and adapt.
- Today we discussed an example agent architecture.
- The police A.I. lab is a good environment for the valorisation of academic ideas on responsibility, legality and machine ethics.
- The police A.I. lab will produce many suggestions, results and examples in the coming years.

(the following slides are some more technical details)

Training Data



- Data is gathered in the business process by the monitor interface. This maintains accuracy and provides (after-the-fact) checks on agent behavior.
- Often we need to ‘bootstrap’ a project
- For the latter we use internally developed labelling tools with uncertainty and Thompson sampling

Auto-Experimenter

- For any data set and any deadline, produce a classifier
- Manual experimentation is boring and time-consuming
- Automated supervised learning is still in its infancy
- We're still exploring this part, but it's important for maintenance of data-driven applications

Solution approaches

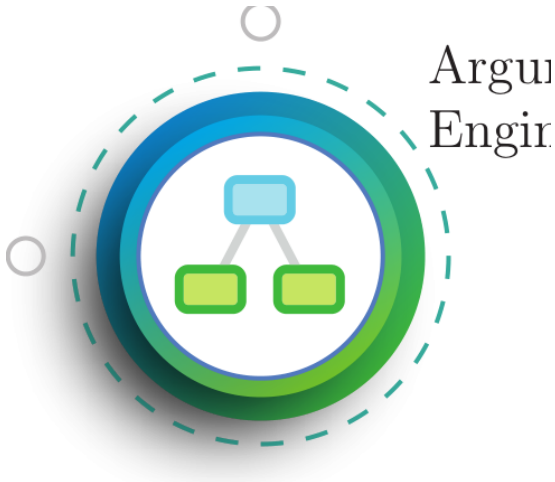
- A one-size-fits-all classifier.
- Metaclassifier.
  - Grid search
  - Sequential model-based optimization (SMBO)
  - Genetic algorithms

We're looking into SMAC as a candidate: *Sequential Model-Based Optimization for General Algorithm Configuration*. Frank Hutter, Holger H. Hoos and Kevin Leyton-Brown. 2011 (basically train a regression model that predicts for an algorithm + hyperparameters how 'good' it will perform and then use that model in deciding which next algorithm + parameters to try)

Notes on an Auto-Experimenter

- Impacts accuracy because it provides the classifiers and attribute extractors that turn unstructured data to structured data
- The auto-experimenter is not transparent
- The auto-experimenter is an integral part of controllability: this part is responsible for classification errors
- The auto-experimenter improves maintenance efficiency for data-driven applications

Argumentation



- We draw upon legal informatics, in particular computational argumentation, for legal reasoning
- Computational argumentation can be used as a basis for explainable A.I.
- However, we need to integrate computational argumentation in the machine-learning driven architecture

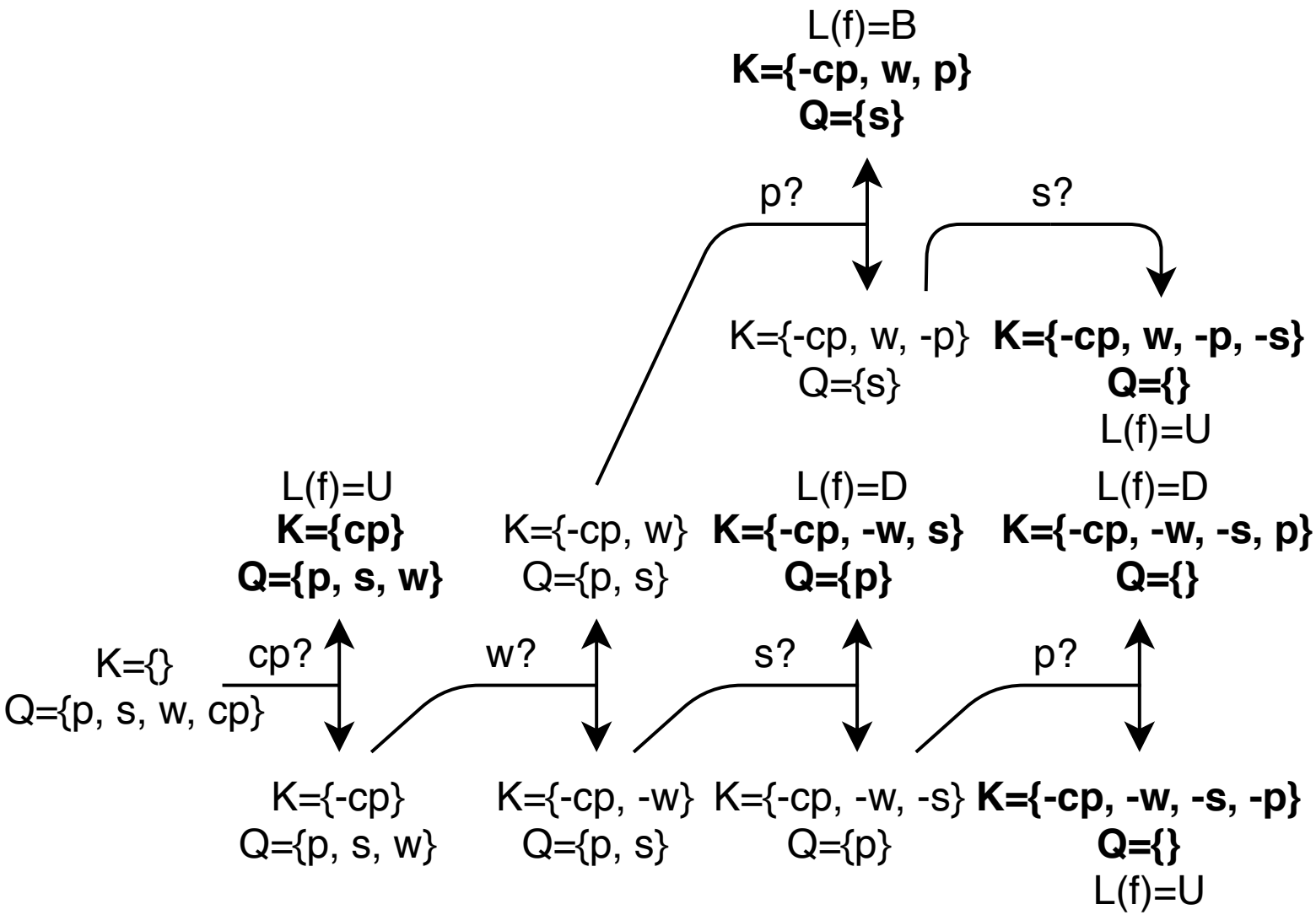
Argumentation

- We developed a sound polynomial-time approximation algorithm that tells whether more information may change the final decision of the agent
- The argumentation formalism is a simplified version of ASPIC

Policy

- We maximize the proportionality of the agent's data gathering
- We also want to guarantee that we can explain decisions based on their legal context
- For this we transform the argumentation framework to an MDP (argumentation stability is the main influence on the reward function)
- For an MDP we can use various known techniques to create the optimal policy or approximation thereof

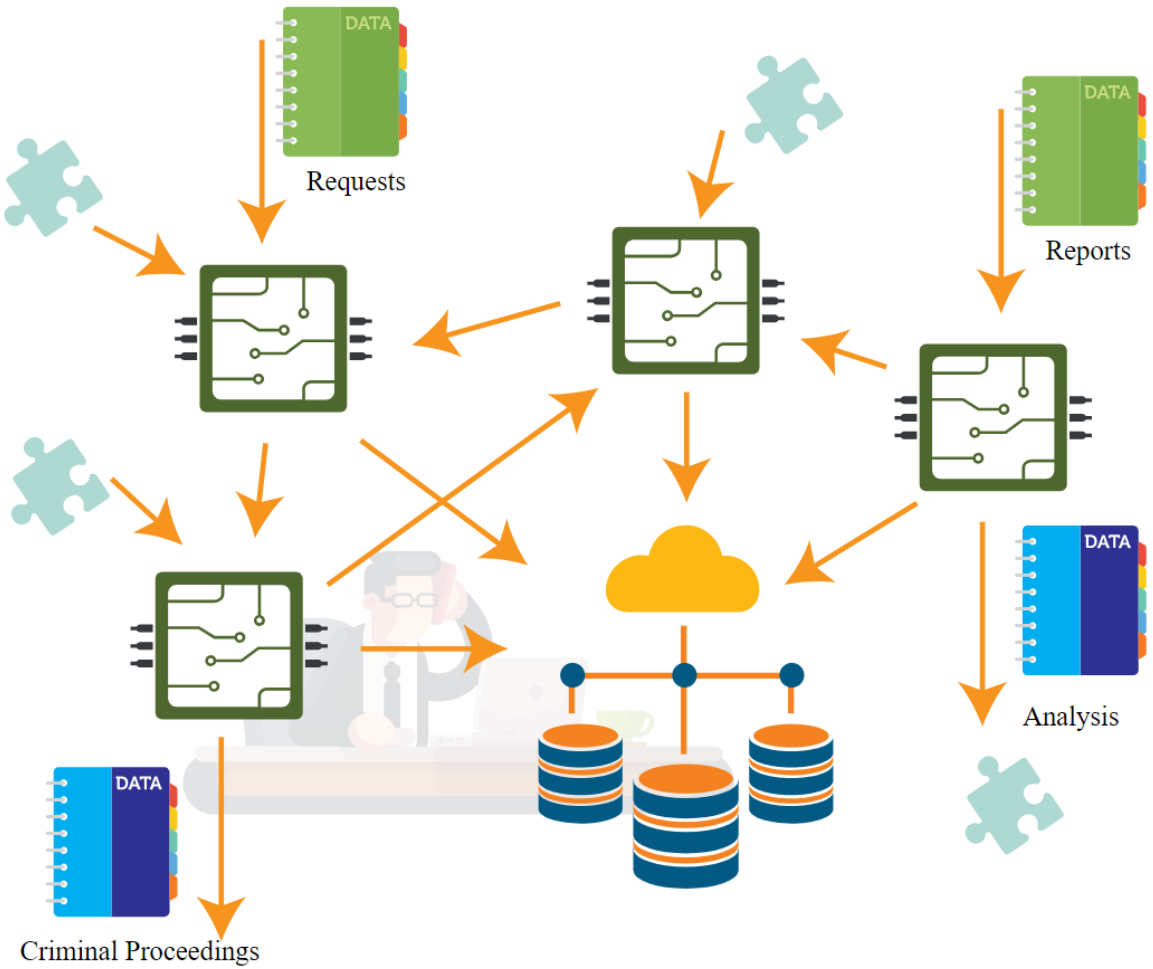
MDP Policy


L(f)=B
K={-cp, w, p}
Q={s}
p?
s?
K={-cp, w, -p}
Q={s}
K={-cp, w, -p, -s}
Q={}
L(f)=U
L(f)=U
K={cp}
Q={p, s, w}
K={-cp, w}
Q={p, s}
L(f)=D
K={-cp, -w, s}
Q={p}
L(f)=D
K={-cp, -w, -s, p}
Q={}
K={}
Q={p, s, w, cp}
cp?
w?
s?
p?
K={-cp}
Q={p, s, w}
K={-cp, -w}
Q={p, s}
K={-cp, -w, -s}
Q={p}
K={-cp, -w, -s, -p}
Q={}
L(f)=U

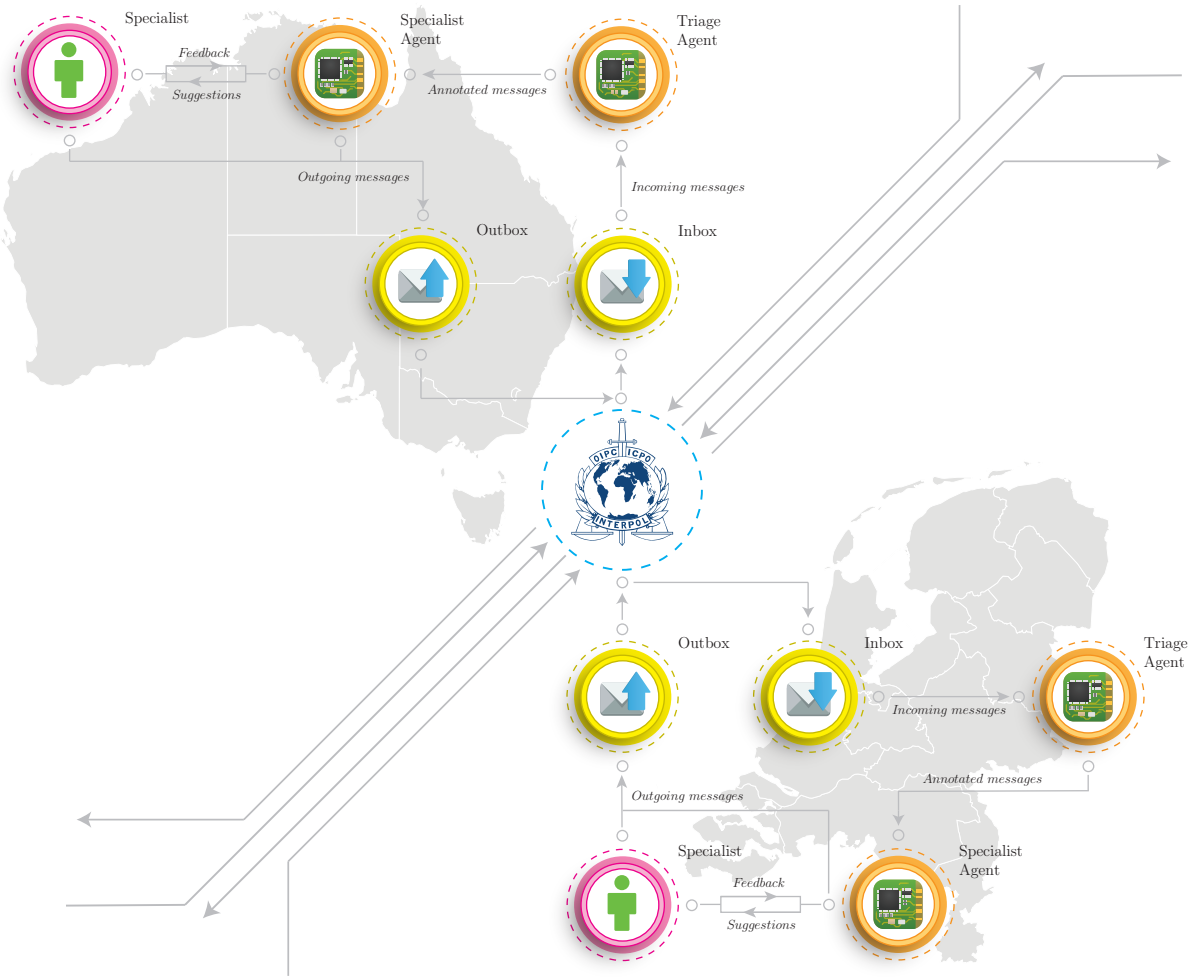
Environment Model

- Getting the transition probabilities requires some model of the environment.
- The appropriate model heavily depends on the application at hand.
- Per application a data-mining expert has to look into this part.

Let's discuss the arrows between agents



Let's discuss the arrows between agents



Argumentation Dialogues

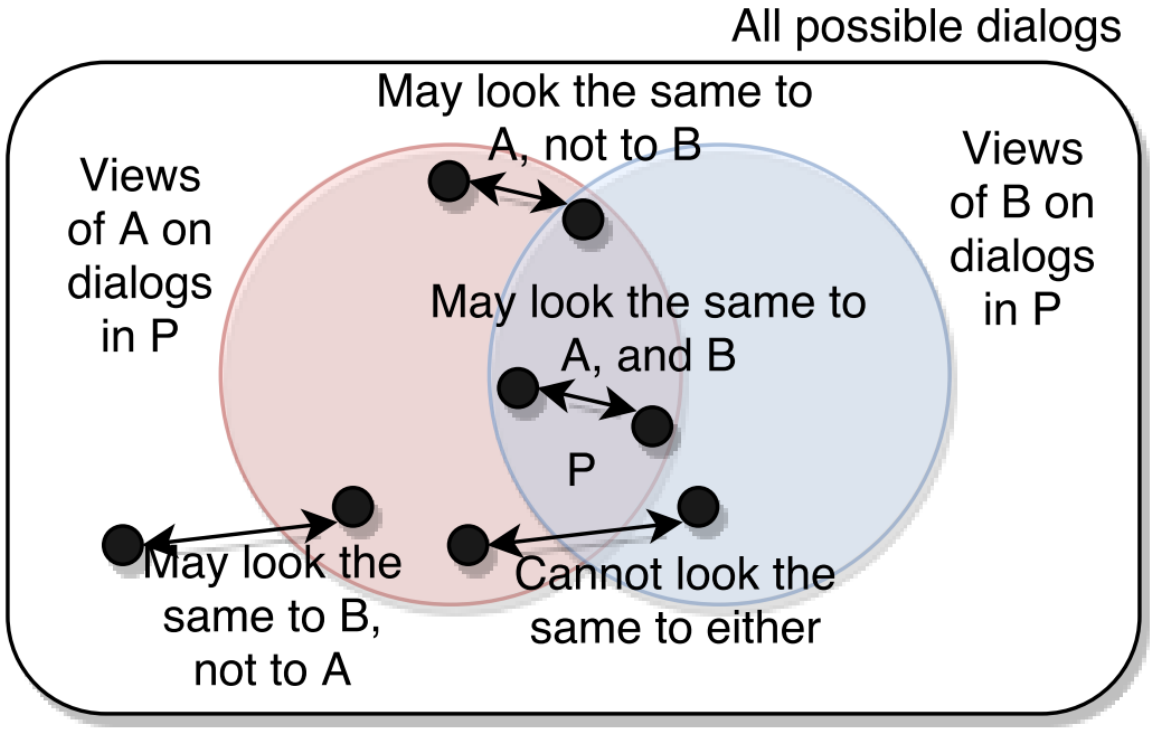
- We use argumentation as a core part of an agent's explainability
- Argumentation dialogues are therefore a natural choice for their communication
- But we want to be able to verify communicative behaviour

Challenge

How do we specify a communication protocol such that all agents are guaranteed to be able to know whether their own actions are legal?

- No middleware allowed that sees all communication (security)
- Different knowledge/tooling per agent
- No full control over all participants

P2P Suitable



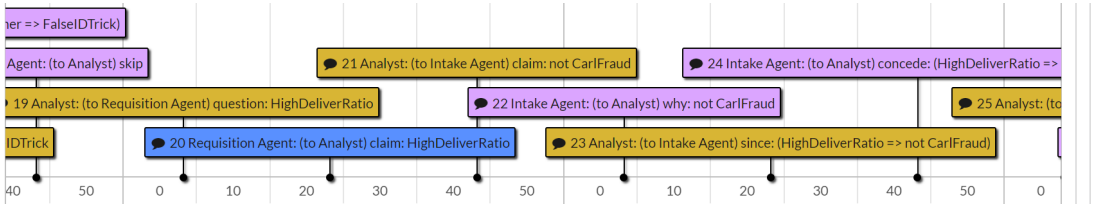
A protocol is P2P suitable iff for every violation at least one agent (the cause of the violation) witnesses this violation.

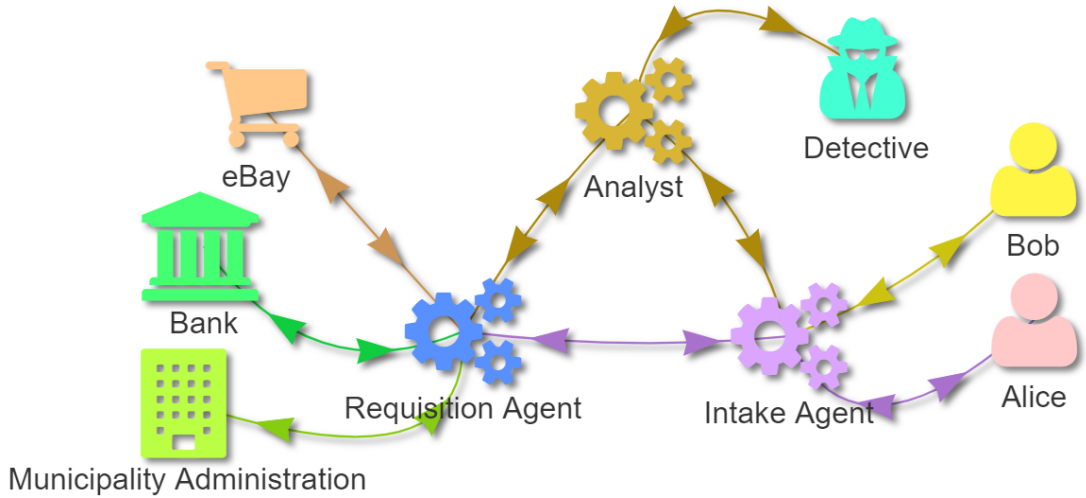
Components

- Dialogue Graph: We capture the state of the agent in a dialog with a graph (i.e., its arg1 & arg2 log based on O'Keefe's distinction)
- Dialogue Templates: A template tells when a locution is allowed to be sent/received and how this is interpreted in the dialog graph (i.e., how it is interpreted from an argumentation point of view).
- Template-based System: A template-based system is one where the agents maintain dialog graphs and use templates to interpret locutions.
- We prove that template-based systems are P2P suitable.

Dialogue

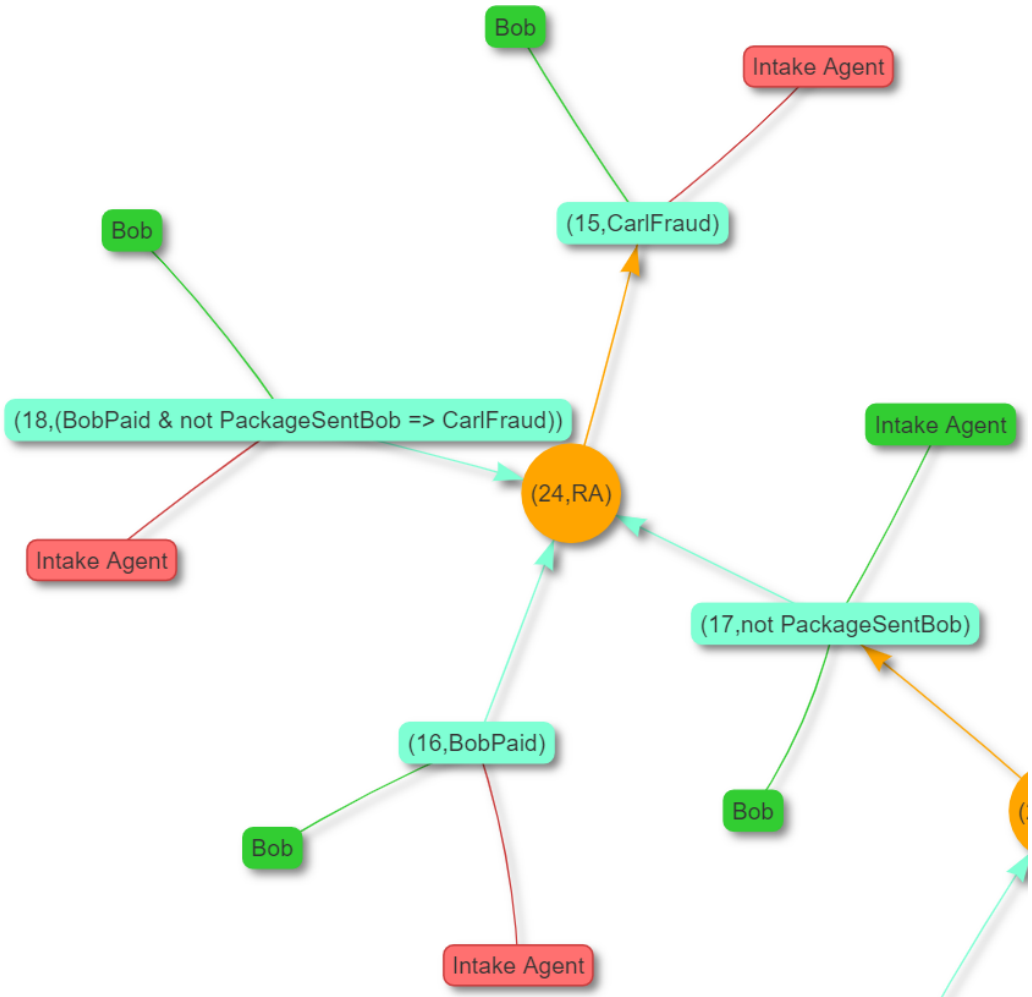
We want to interpret dialogues from an argumentation perspective.





Argument 1

Argument 1 captures the structure of arguments.

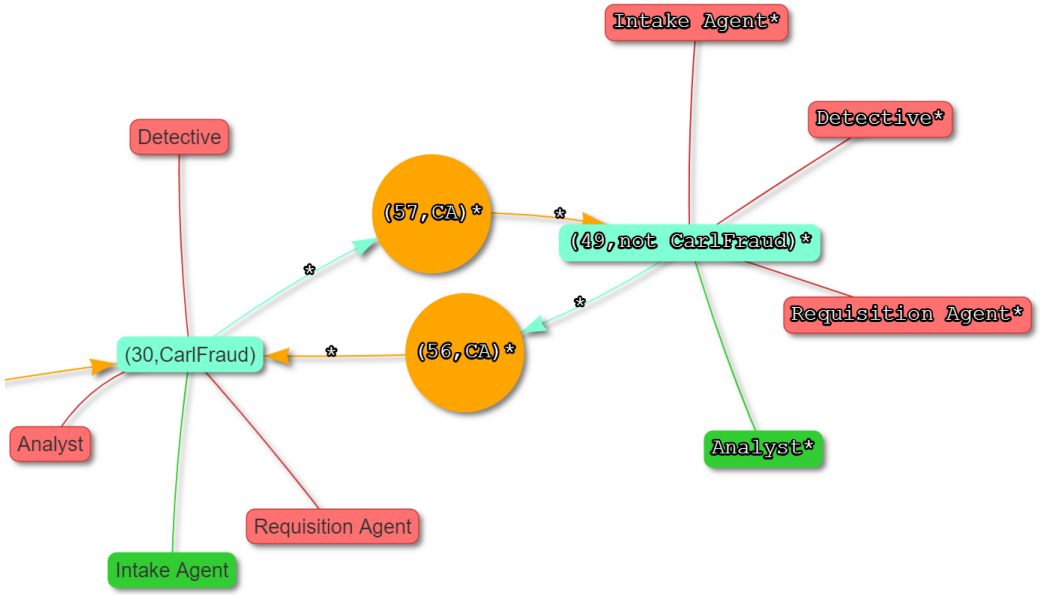


Argument 2

Argument 2 captures the dialogue structure of argumentative dialogues.

Illocutionary Force

The illocutionary force of a message is an update over the argument 1/2 structure.

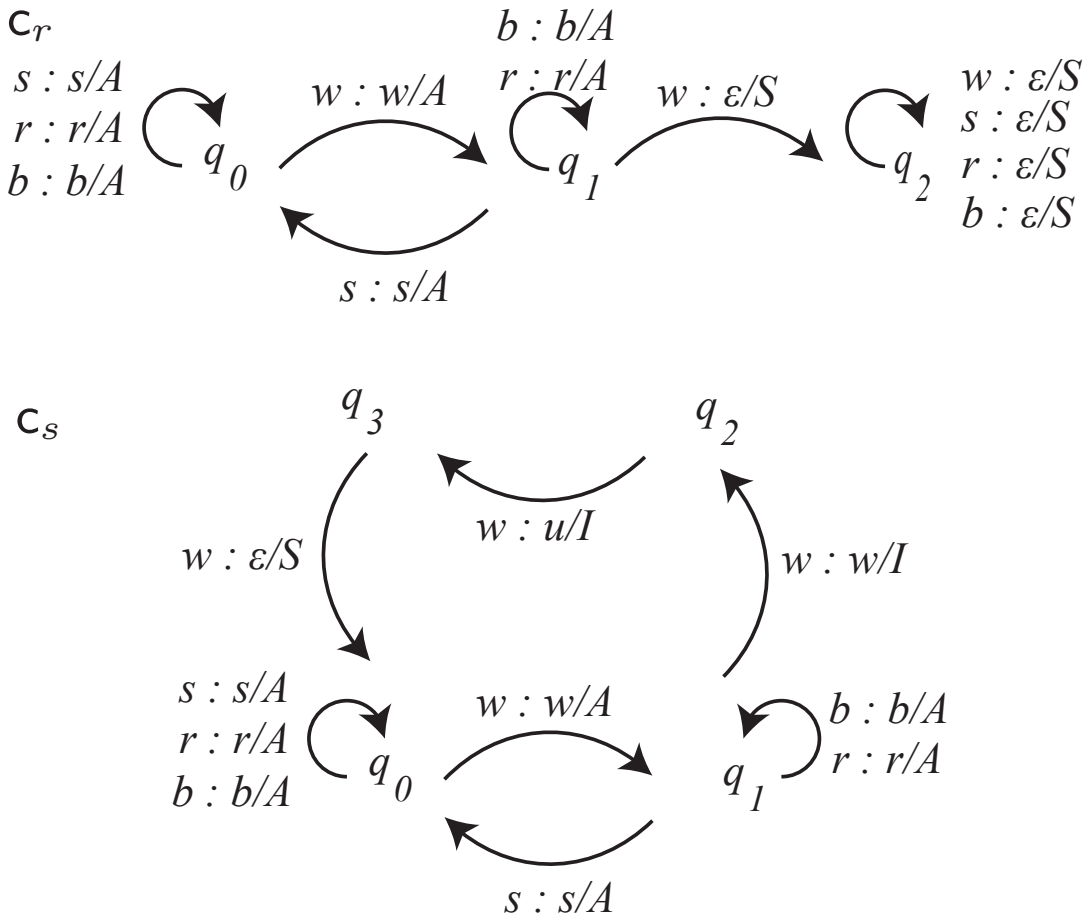


Runtime Monitoring and Control

Having a P2P suitable protocol is only the start. We still need to know how we may enforce it.
Our approach: *Model Check What You Can, Runtime Verify the Rest.* Hinrichs, T. and Sistla, A. and Zuck, L. HOWARD-60. 2014.

Controller automaton

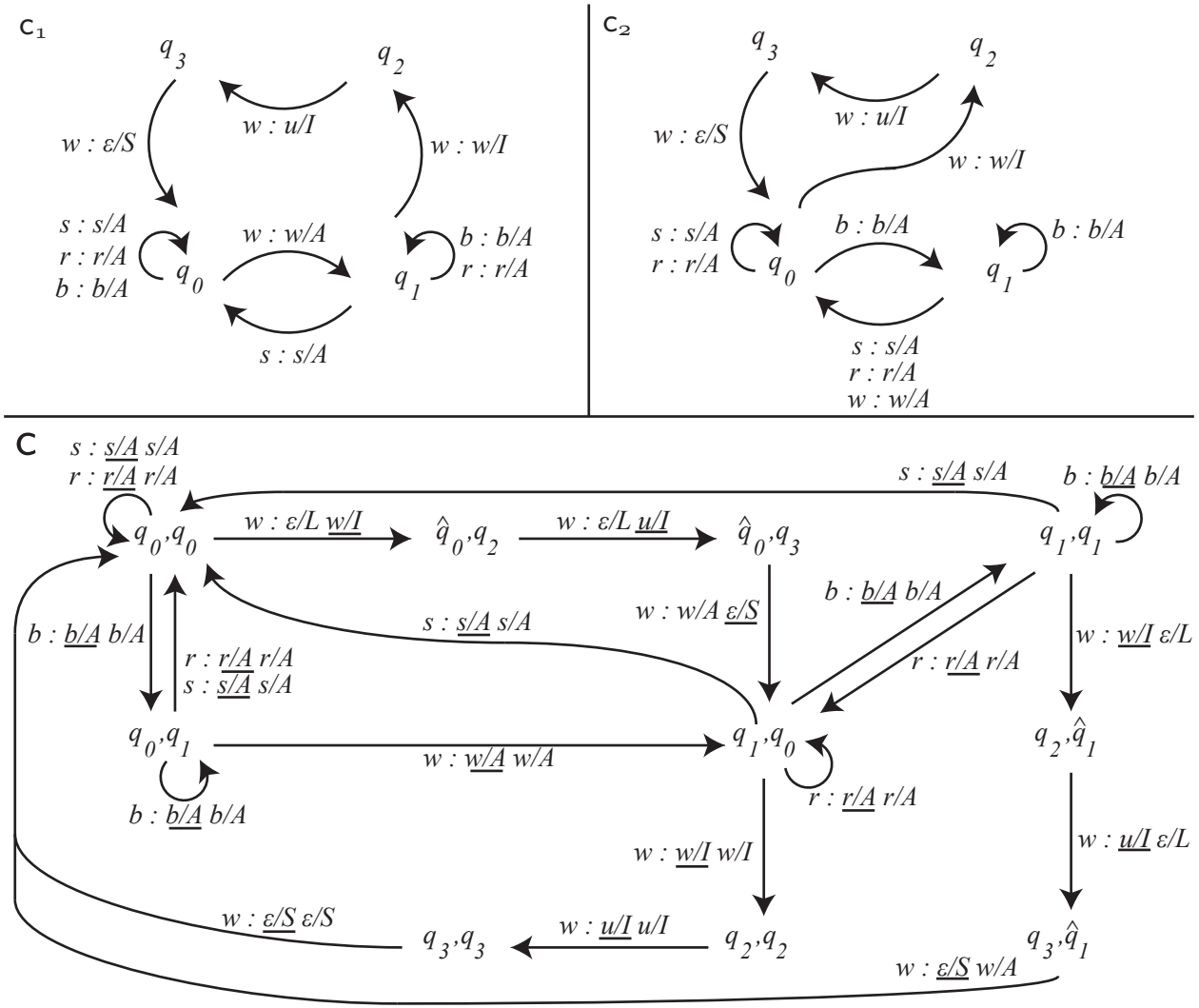
Edit Automata: Automaton based models for runtime controllers



Collaborative Controllers

1. Model of concurrently applied controller automata
2. Revision conflict resolution is captured by a selection function
3. Challenge: find a procedure for constructing the revision function for a set of property enforcing controller automata

Collaborative Controller



Relation to MAS for police

1. P2P suitable protocols guarantee that a collaborative controller is possible.
2. The dialogue templates of P2P suitable protocols are building blocks for synthesizing runtime controllers.
3. If each agent has its own runtime controller, then all the controllers of the agents combined are the collaborative runtime controller
4. Current research: design time verification combined with runtime verification for cross-jurisdiction autonomous software

Engineering

How do we actually develop agents?

- Describe parts of the solutions as design patterns and develop libraries to support their application
- Aspect-oriented programming may provide a good solution to the separation of concerns regarding (runtime) verification and business logic
- Use open-source available data science techniques
- SMBO is available as auto-weka and autosklearn
- The argumentation engine and policy learner are in-house developed
- We try to maintain a service-oriented architecture
- Software such as MCAPL provides a basis for model checking

Some other current developments

- Simulation of organized crime as way to obtain an environment model.
- Human-computer interaction, including speech to text, for the actuator.
- Organizational/social change and impact.
- Expansion of natural language processing tooling, in particular application of LSTM's for attribute extraction.