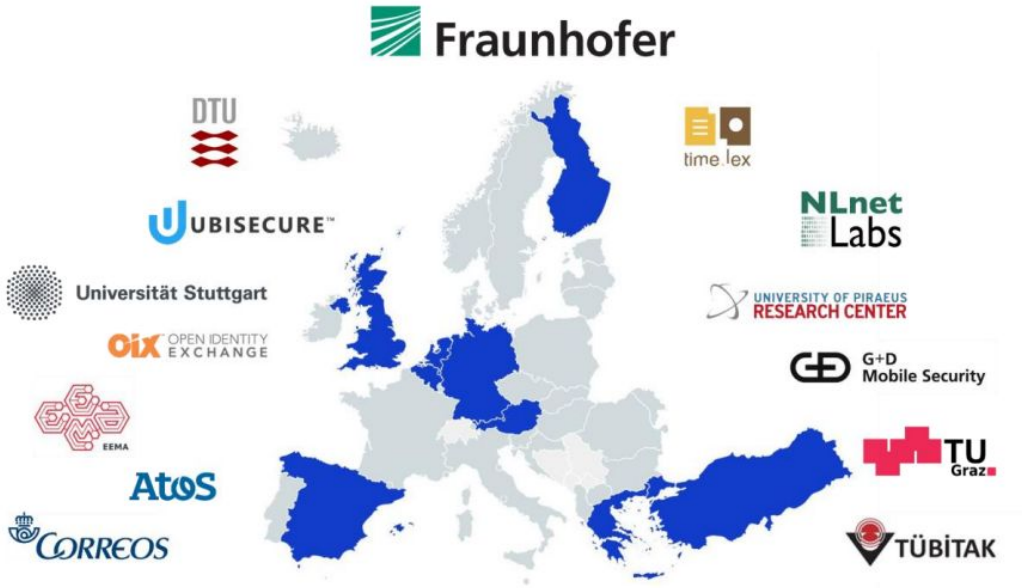
LIGHT*est*

A Lightweight Infrastructure for Global Heterogeneous Trust Management



Trust Verification & Policies in LIGHT*est*

LIGHT*est* has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

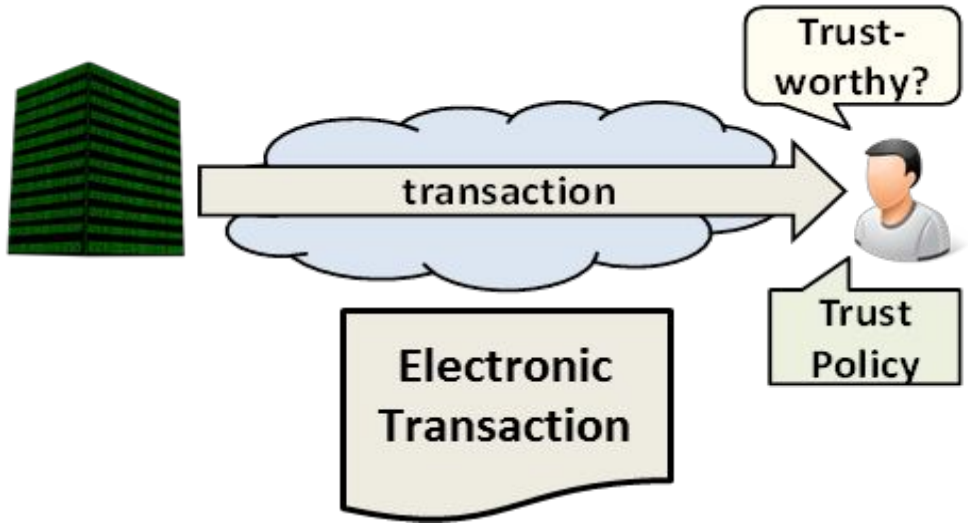
Outline

- **Formats**
- Policies in Lightest
- An introduction to TPL
- The TPL Interpreter
- Parser
- Natural Language Processing
- GTPL

LIGHT^est has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

The Electronic Transaction

- An electronic transaction is a container (of a given format)
- Contains several documents or sub-containers
- Documents and containers are associated with an electronic identity, e.g., via electronic signature
- Electronic Signatures: the electronic equivalent of a traditional manual signature, placed on a piece of paper.
  - Qualified ones are even cryptographically secure



LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Transaction Format

- In Lightest there are standardized “templates” for Transactions…
- ...so called Formats
- Known by all parties
- Formats may have a specialized parser
- Transactions have to conform to be valid
- Header: same in every Transaction
- Body is domain specific
- That is why we call formats also “domain templates”

LIGHTest has received funding from the European Union’s Horizon 2020 research and innovation programme under G.A. No. 700321

Subformats

- Formats that can be plugged into formats
- Necessary for a building brick approach
- In addition Formats may need a specific parser (e.g. certificate subformat)
- ...TODO.

LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Format as blank XML

```
<?xml version="1.0" encoding="UTF-8"?>

<form format="theAuctionHouse2019" specification="theAuctionHouse_v2">

        <person>
                <name type="string" />
                <street type="int" />
                <city type="string" />
                <country type="string" />
        </person>

        <lot_number type="int" />
        <bid type="int" />
        <certificate type="cert" />
</form>

```


LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Format as XSD

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
           attributeFormDefault="unqualified">
    <!-- XML Schema Generated from XML Document with http://xmlgrid.net -->

    <xs:element name="form">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="person">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="name" type="xs:string"></xs:element>
                            <xs:element name="street" type="xs:string"></xs:element>
                            <xs:element name="city" type="xs:string"></xs:element>
                            <xs:element name="country" type="xs:string"></xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="lot_number" type="xs:int"></xs:element>
                <xs:element name="bid" type="xs:int"></xs:element>
            </xs:sequence>
            <xs:attribute name="format" type="xs:string"></xs:attribute>
        </xs:complexType>
    </xs:element>
</xs:schema>
```


LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

ASIC

- Associated Signature Container
- Standard for packaging container types
- Standard for associating the data and cryptographic parts
- We use it as standard for Lightest Transactions
- Internal structure of an ASiC container
  - A **root** folder for all the container content, including folders that reflect the structure of the content
  - A "**META-INF**" folder inside the above mentioned root folder that holds files that contain metadata about the content, including the associated signature/and or time assertion files
- This is where Delegation (before lunch session) and transaction comes together again
- Finally qualified signatures is applied

LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Outline

- Formats
- **Policies in Lightest**
- An introduction to TPL
- The TPL Interpreter
- Parser
- Natural Language Processing
- GTPL

LIGHT^est has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

What are Policies used in general

- A policy is a deliberate system of principles to guide decisions and achieve rational outcomes
- A policy between two parties helps building trust
- Lightest uses policies to automatically decide whether an incoming “request” is to be trusted
- A policy decides what to accept and what to discard

LIGHTest has received funding from the European Union’s Horizon 2020 research and innovation programme under G.A. No. 700321

What are Lightest policies in Lightest? What are they enabling you to do?

- Incoming transaction must be verified and reviewed (e.g. the seller must know if the buyer is authorised to buy in a companies name)
- Lightest enables you to have automated decisions
- To be automated parameters need to be set
- Lightest lets you write multiple policies to every regard
- Feeded to the ATV (or more precisely the Interpreter), policies will take care for you for incoming transaction

LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Outline

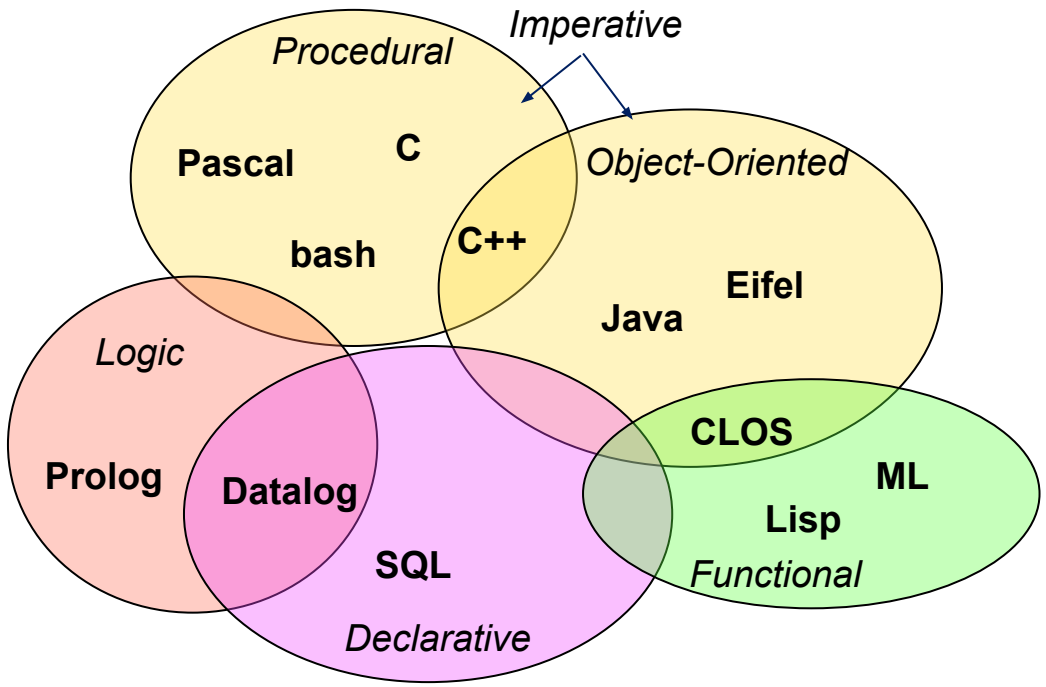
- Formats
- Policies in Lightest
- **An introduction to TPL**
- The TPL Interpreter
- Parser
- Natural Language Processing
- GTPL

LIGHT^est has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Introduction Logic Programming

- Logic Programming is declarative programming paradigm (in contrast to imperative: Pascal, C, Java,...)
- We use a subset of first-order logic, called Horn clause logic
- Most widely used logic language is Prolog (**Pro**gramming in **log**ic)
- Advantages of logic programming:
  - No need to distinguish programs from database
  - Natural support of pattern-matching and meta-programming
  - Ease of representing knowledge

Programming Language Paradigms



LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Introduction to TPL

- made up of facts and rules
- a fact asserts a property or relation:
  - e.g. *parent(alice, bob).*
  - means Alice is a parent of Bob
- a rule infers a property or relation based on preconditions:
  - e.g. *parent(X, Y) :- mother(X, Y).*
  - means that Person X is the parent of Person Y, if X is the mother of Person Y

parent(alice, bob). → Fact

parent(X, Y) :- mother(X, Y) → Rule

Predicate

Head Body

→ Predicate definitions consist of clauses.
→ Facts do not have body because they are always true.

LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Introduction to TPL

- A predicate head consists of:
  - predicate name
  - arguments (not necessarily)
- The body basically is made up of:
  - subgoals (calls to a predicate)
  - terms
- These terms can be:
  - Constants e.g. jane, alice,...
  - Variables e.g. X, Person1,...

```
parent(alice, bob).                    → Fact
parent(    X,    Y) :-  mother(X, Y)   → Rule
```

Predicate

Head

Body

→ Predicate definitions consist of clauses.
→ Facts do not have body because they are always true.

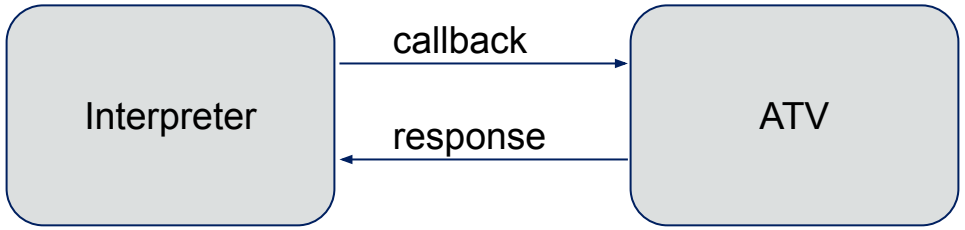
LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Main Concepts TPL - Build-in Predicates

- Furthermore TPL supports built-in predicates
- ...which trigger events on the ATV

- Custom predicates possible through plug-in system

- Built-in predicates:
  - extract
  - lookup
  - trustlist
  - trustscheme
  - verify_signature
  - verify_hash



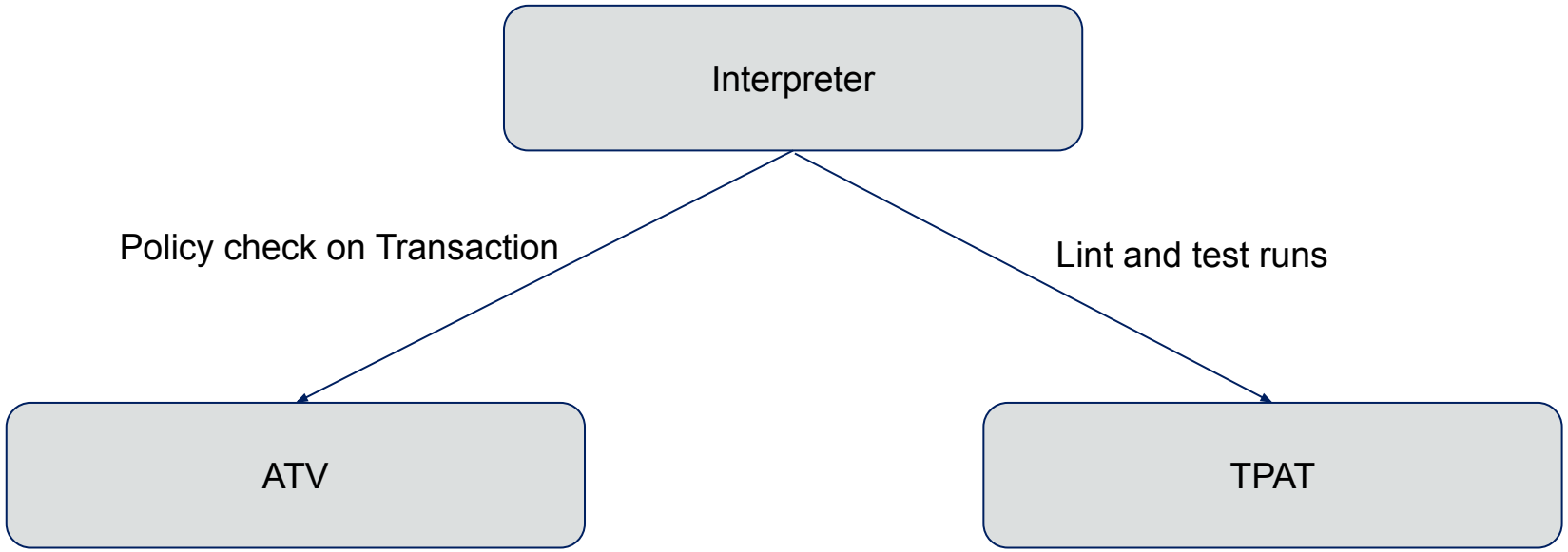
LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Outline

- Formats
- Policies in Lightest
- An introduction to TPL
- **The TPL Interpreter**
- Parser
- Natural Language Processing
- GTPL

LIGHT^est has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

The Interpreter and its interaction with ATV and TPAT



LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Policy snippets and what happens in the ATV background Part 01

- How the interpreter and the ATV work together
- Using an example of delegation
- Qualified delegation can be split:
  - checking the mandate
  - checking the mandator's key
  - checking temperedness

```
checkQualifiedDelegation(Document, Mandate) :-
    checkMandate(Document, Mandate),
    checkMandatorKey(Document, Mandate),
    checkValidDelegation(Document, Mandate).
```


LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Policy snippets and what happens in the ATV background Part 02

- check mandate is pointing to the correct person
- is the purpose correct?

```
checkMandate(Document, Mandate) :-
    extract(Mandate, format, delegation),
    extract(Mandate, proxyKey, PkSig),
    verify_signature(Document, PkSig),
    extract(Mandate, purpose, place_bid).
```


LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Policy snippets and what happens in the ATV background Part 03

- mandator is trustworthy
- is the Delegation's signature correct

```
checkMandatorKey(Document, Mandate) :-
    extract(Mandate, issuer, MandatorCert),
    extract(MandatorCert, trustScheme, TrustSchemeClaim),
    trustscheme(TrustSchemeClaim, eIDAS_qualified),
    trustlist(TrustSchemeClaim, MandatorCert, TrustListEntry),
    extract(TrustListEntry, pubKey, PkIss),
    verify_signature(MandatorCert, PkIss).
```


LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Policy snippets and what happens in the ATV background Part 04

- check that the Delegation is untempered

```
checkValidDelegation(Document, Mandate) :-
    extract(Mandate, delegationProvider, DP),
    lookup(DP, DPEntry),
    extract(DPEntry, fingerprint, HMandate),
    verify_hash(Mandate, HMandate).
```


LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Outline

- Formats
- Policies in Lightest
- An introduction to TPL
- The TPL Interpreter
- **Parser**
- Natural Language Processing
- GTPL

LIGHT^est has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

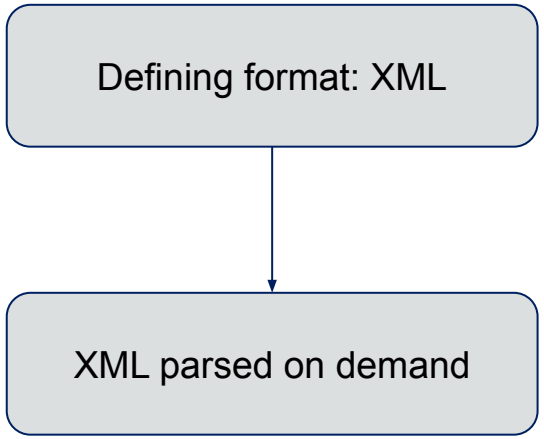
Parser for ATV & TPAT

- Parsers in ATV and TPAT are used to load transactions respectively their formats
- Two kinds of parsers:
  - Generic
  - Custom
- Subformats need custom parser
- Verifies if the transaction is formally valid
- Independent from the interpreter
- Parser is more an interface to extract information out of transactions

LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Generic Parser

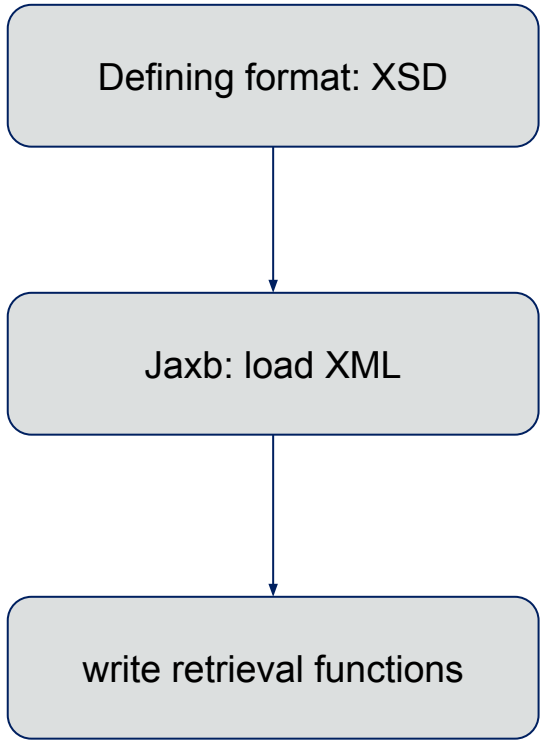
- Format: blank XML
- Transaction data: filled out XML
- Static
  (e.g. no dynamically sized lists supported)
- Querying simple Fields only



LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Custom Parser with XSD and Java JBX

- Format: XSD
  - basically cont. from formats with XSD
- Transaction data: filled out XML
- Dynamic
  - Using Jaxb
  - Fields retrieval redirected over Java
- Querying smart fields possible



LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Outline

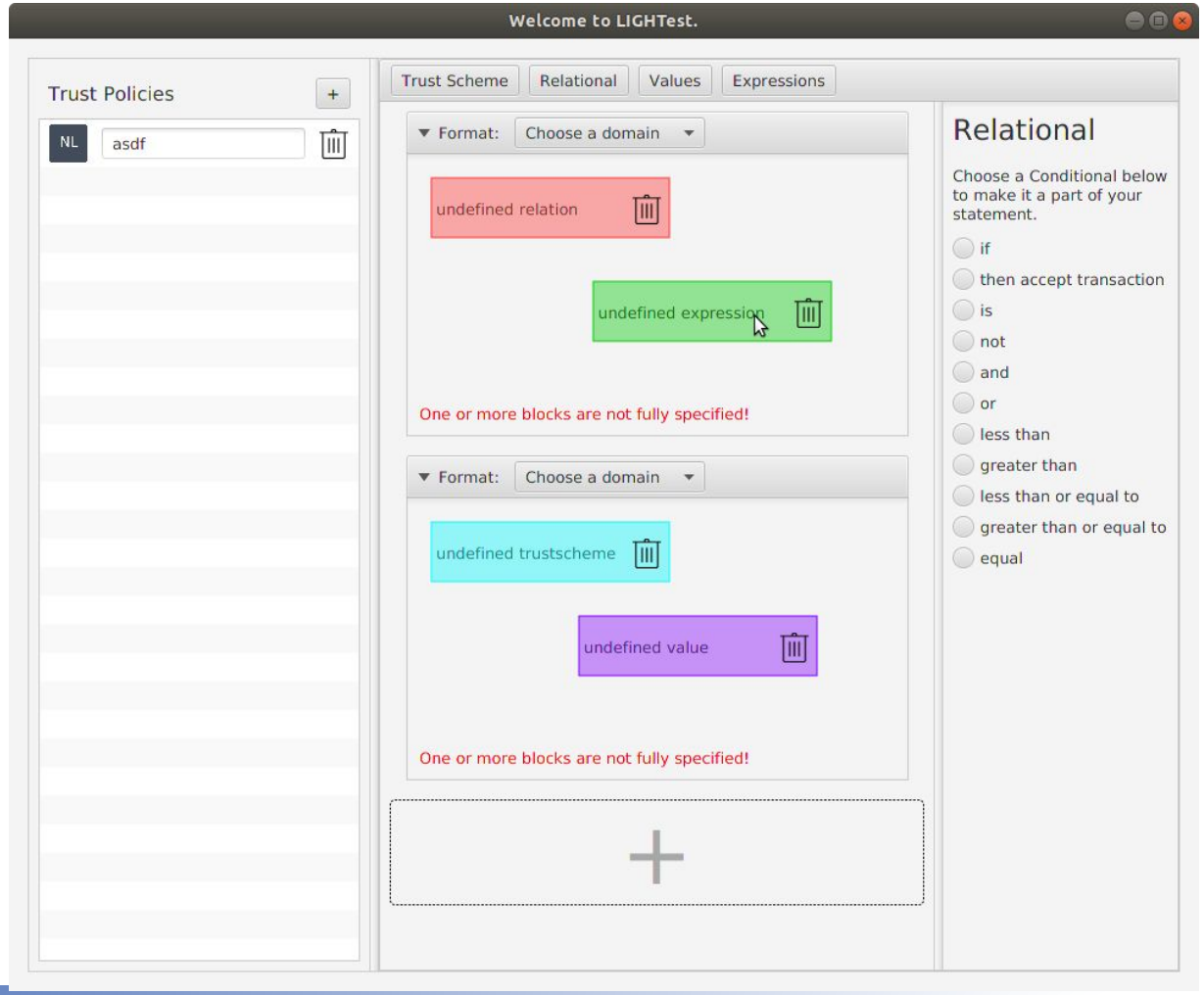
- Formats
- Policies in Lightest
- An introduction to TPL
- The TPL Interpreter
- Parser
- **Natural Language Processing**
- GTPL

LIGHT^est has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

TPAT & NL recap

- **N**atural **L**anguage is the intermediate level in the TPAT (**T**rust **P**olicy **A**uthoring **T**ool)
- A building blocks based approach to create policies in a natural sounding way
- Statements are *OR* connected
- Typical block statement:

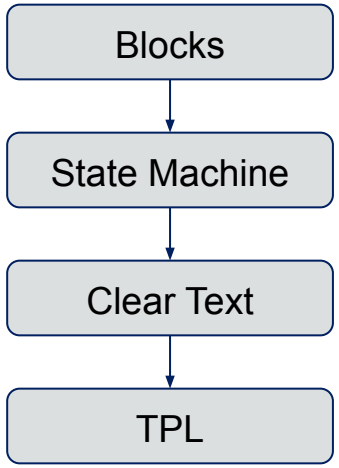
  If <condition> and <condition> then accept



LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

What is Clear Text in Lightest terms? How are NL Blocks translated?

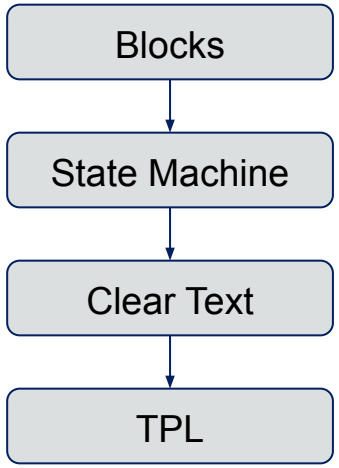
- The User stucks Blocks together
- A State Machine is used
  - user correction support
  - translation to Clear text
- Clear Text is basically Blocks translated to a text string of a certain form
- State Machine offers even more possibilities (e.g. recommender system supported autocompletion)



LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Clear Text to TPL

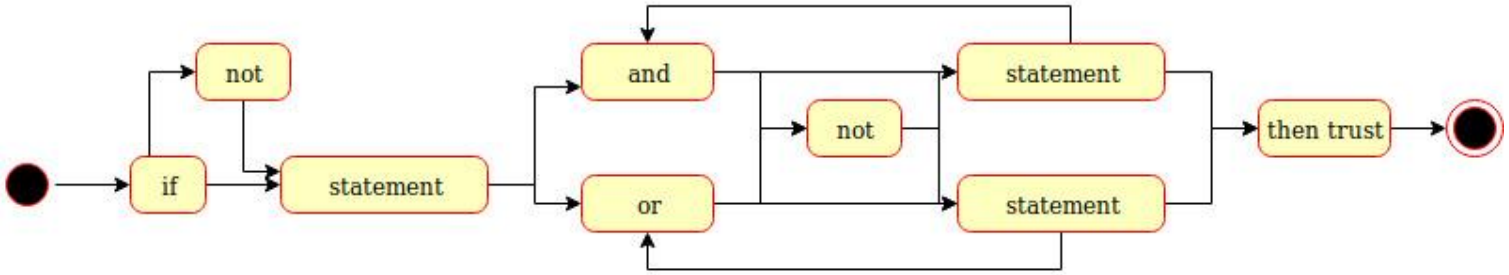
- Done by the technical university denmark
- Antlr to cross compile
- One Clear Text instruction may extend to multiple TPL instructions (magic words)
- The whole procedure gets triggered on every drop of a block, and saves out TPL-code to a file
- Only one way: TPL to blocks not possible



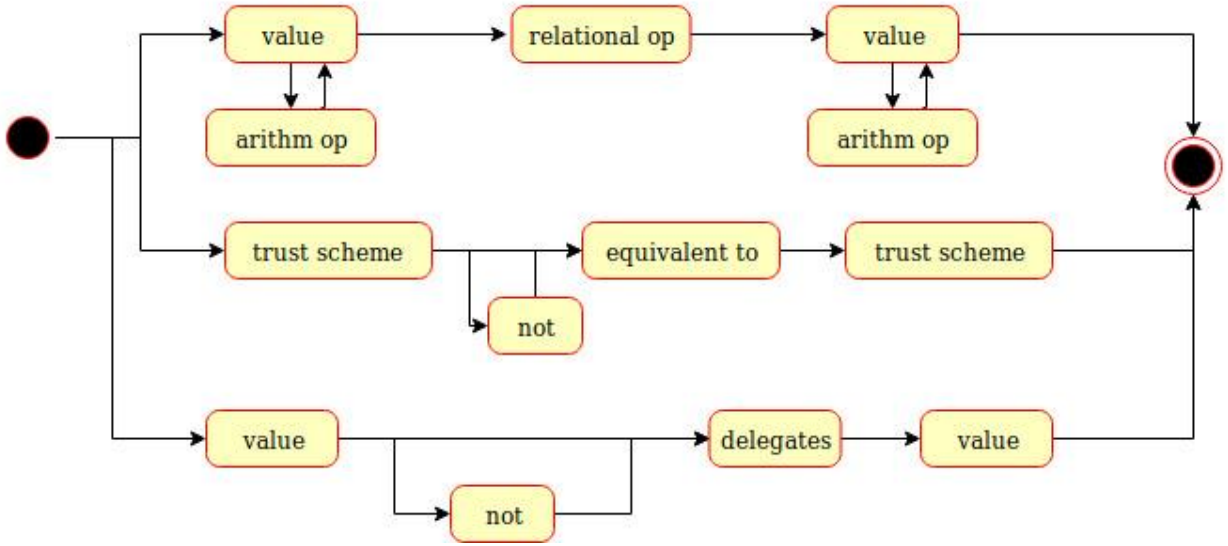
LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

NL State Machine

State Machine:



Statement:



LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

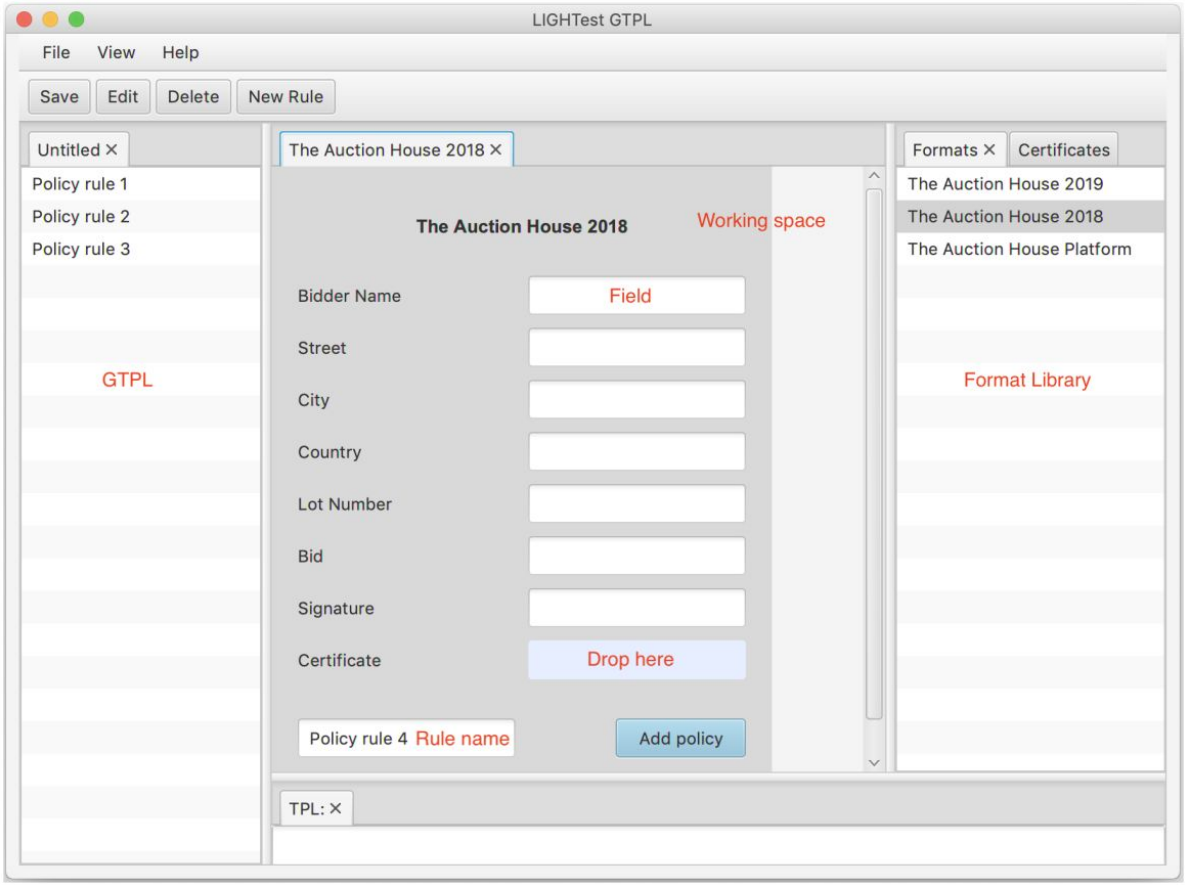
Outline

- Formats
- Policies in Lightest
- An introduction to TPL
- The TPL Interpreter
- Parser
- Natural Language Processing
- **GTPL**

LIGHT^est has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

What is GTPL

- Similar to the Delegation Publisher
- For companies:
  - Creating policy formats done by an employee with domain expertise and programming knowledge (e.g. IT finance engineer)
  - Business people then use one of the created formats and fill out the transaction
- Formats can be added using a plugin system



LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Q & A

Thank you for your attention

LIGHTest has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Q&A Bonus Slides

LIGHT^est has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321