Department of Informatics
of the Technical University of Munich
in cooperation with
fortiss

Project Report in Advanced Practical Course
Blockchain technology for public sector innovation (IN2106, IN4212)

DLT to Support Dutch Immigration Service

Authors:

Lukas Braunstorfer (ga54caj@mytum.de) Master in RCI

Daniel Illner (ge98riy@mytum.de) Master in RCI

Supervisor:

Prof. Dr. Helmut Krcmar (TUM - Chair for Information Systems)

Advisors:

Dian Balta (fortiss)

Anastasios Kalogeropoulos (fortiss)

Matthias Buchinger (fortiss)

Submission Date: 13.07.2020

Abstract

The immigration process in the Netherlands faces more complexity and a variety of problems nowadays. In order to overcome these problems, innovative ideas and concepts are essential. Therefore, TNO in combination with fortiss and the Technical University Munich initiated a project aiming for new vision and solutions.

In this report we want to introduce a concept to overcome those issues and that should function as a first stage concept to initiate a process that will force change in the public sector, more specifically in the immigration process. Our concept relies on the connection of three different ideas and technical implications. The use case and prototype provided is based on a small part of the Dutch Aliens Act which is represent by the domain-specific language system FLINT. The prototype is a distributed ledger technology system based on Evidentia and implements the decision process of providing a resident permit for students or highly skilled workers.

Contents

List of Figures

List of Tables

List of Abbreviations

BAMF	Federal Office for Migration and Refugees
DEN	Distributed Evidence Network
DLT	Distributed Ledger Technology
ETB	Evidential Tool Bus
FLINT	Formal Language for the INTerpretation of Sources of norms
fortiss	fortiss GmbH
IND	Dutch Immigration and Naturalisation Service
NGO	Non-governmental organization
TNO	Netherlands Organization for Applied Scientific Research

1 Introduction

This project was introduced through TNO aiming for new concepts and solutions regarding the public sector in the Netherlands. TNO in this context is constantly striving for innovative ideas that might be used on a wide range of applications. For this project the focus will lie on the innovation towards the immigration process that is handled by the Dutch Immigration and Naturalisation Service (IND) and that is facing increasing complexity year by year. Regarding to the latest research, the biggest problem of the IND is a lack of decision-making capacity [3]. Especially the year 2015 was a decisive point, when the IND was dealing with major challenges due to the high influx of asylum seekers [4]. Currently, the Dutch Immigration Service handles a 6-figure number of applications that again comprehend various types [4]. The high number of cases in combination with legal complexity leads to delay within the process. There are incurring fines for being behind schedule for some of their cases. Besides this economical damage for the IND and the public sector, the process creates harm for the individual that is often left in unstable situations for years. You can imagine, that these situations reach out to a cultural level that causes misunderstanding, distrust and instabilities within society. Without diving too deep into details, this transforms a technical question into a much wider perspective, which is indeed tempting.

Having mentioned these problems and insufficiencies, this project aims towards an innovative concept for the process of immigration, which will be more efficient, transparent and reliable in the end. This applies to the immigration department itself as well as the individual that is seeking for asylum, a temporary resident permit or even citizenship.

To reach this ambitious goal formulation, the project introduces a concept based on distributed ledger technology and therefore aims towards a digital transformation from a central-based approach to a decentralized system. As digital transformation involves reconsidering existing processes, establishing new ways of working, and promoting innovative ways of thinking, the project will demonstrate a possibility in order to encourage change. Seeking for automation and transparency as the key factors, distributed ledger technology facilitates a fully digital exchange of information among all members of the system, and this exchange is both fast and secure. Thus, everyone can instantly be made aware of the completion of essential steps within a given process, where-upon other authorities can immediately initiate follow-up steps accordingly. This leads to a highly automated process in a secure fashion.

In this paper we present our approach, which implements a small fragment of legal text into a distributed ledger technology system that represents a process of providing a resident permit to an applicant. We will explain the related work and idea giving concepts in the second section. The third chapter will provide a general explanation of the scenario and use case, followed by the detailed description of the prototype. In the end we will discuss our findings, the possibilities and limits of the current prototype as well as suggestions for improvement.

2 Related Work

Our approach is based on the work of three different projects that differ in concept, technical aspects as well as the principal goals. But these projects offered perspective, ideas, uniqueness and technical support that lead to an innovative concept of implementing and building an immigration process from scratch.

The framework that the conceptual structure and organization is based on, is a pilot project for the German asylum procedure done by the Federal Office for Migration and Refugees in Germany. By using a blockchain solution the BAMF aims to facilitate an exchange of information in the sub-processes ‘registration, creation of an application file and personal interview’, ‘referral’, and ‘ruling and next steps’. This exchange of information is to be timely and fully digital, significantly minimizing the duration of procedures and the expenditure of resources [2]. Meanwhile, security aspects could be considered more efficiently, document procurement could be brought forward to an early point in the process, and repatriation could be organized sooner. It will accelerate the processing of selected stages in the asylum procedure and increase the quality of information regarding rulings on asylum applications. Their system architecture is implemented via Hyperledger Fabric and is divided into three layers. One layer provides existing workflow management systems and data repositories of the respective authorities. Above this, they introduce the integration layer, which hosts a dashboard that enables displaying content from the backend layer and the blockchain layer, depending on the role and access rights of the authority. In addition, they introduce a privacy service, which enables a data protection-friendly mapping not only of unique, pseudonymized reference characteristics on the blockchain but also of technical characteristics in the backend systems [2]. This aspect is of importance as our process aims for transparency on the one hand but handles sensitive data on the other. The third level connects the various existing workflow management systems and data repositories to form a network in which information can be securely shared. Important to note here, is that this network structure and the technical realization influenced the process in a way that an idea was formed how such a distributed ledger technology system could work and generate advantages in comparison to the almost exclusively used centralized system. It might be useful to further follow this pilot project, if our concept evolves to a more sophisticated prototype.

As the immigration process is based on the Dutch Aliens Act, the act plays a central role in automating the process by specifying a set of rules and framing the procedure. The Calculemus method described in this paper [7] introduces an explicit interpretation of the Dutch Aliens Act by using the FLINT language. The method has been used before to make normative interpretations of regulations that form the basis for specific services to be delivered by governmental agencies and of legal cases [6]. The Formal Language for the INTerpretation of Sources of norms (FLINT) has been developed for the explicit interpretation of sources of norms. It is a semi-formal language that has evolved from working on real-life cases. The language consists of three frames: act frames, duty frames and fact frames. It is implemented as .json format. The Act Frame is used to describe normative actions performed by an actor that results in normative changes addressed to an agent that is either receiving the results or is an interested party. As it describes all the aspects of the function that changes a state due to a normative action performed by an agent, it is called the act frame. The act is valid if it is performed in a state that meets a precondition. If the action is performed and the precondition is met, the action will result in normative facts. Regarding our project, this is important, as the preconditions indirectly define our peer organizations, service functions and the workflow. The Duty Frame represents deontic normative relations and consists of a duty, or obligation, that is in effect the state in which an institutional act ought to be performed in the future, or ought to have been performed in the past in case of a violation of the duty. In terms of this project, where the duties will not be considered, the third frame the FLINT language introduces (fact frame), is again relevant. The fact frame can be used to make detailed statements on the precondition of an act. The precondition consists of a function of institutional facts connected by Boolean or arithmetic operators. Every fact in the function of a fact frame, can be the subject of a new fact frame. The level of detail that is pursued depends on the purpose of the analysis. That relates back to the statement before and indirectly will define the framework and structure of the prototype.

The technical implementation of this project is realized by a framework called Evidentia [5]. Evidentia resembles a multi-layered architecture with strong emphasis on modularization. The main idea and most important point regarding the presented prototype are that the architecture defines a tool bus for the generation and continuous maintenance of evidence. Furthermore, a logical foundation with theoretical and operational semantics for the translation of executable business logic is provided and overall the architecture supports a distribution of the tool bus among various actors in a decentralized manner that is accessible as a secure substrate and enables building trusted and accountable evidential transactions. As stated, the architecture consists of three components. The component Cyberlogic, that will provide solid theoretical foundations in logic and proof theory, remains untouched in our prototype. Therefore, the focus is on the remaining two components, the Evidential Tool Bus (ETB) and the Distributed Evidence Network (DEN). The ETB enables services and workflows for the implementation of verifiable evidential transactions. Furthermore, this setup allows the execution of parameterized queries to generate claims stating the evidence. The logic is implemented based on Datalog and each ETB node keeps track of all claims it generates. The DEN is basically a Hyperledger Fabric network that secures a distributed execution of the ETB services and keeps track of all transactions between the nodes based on the Hyperledger policy. It makes sure that identity is valid and provides a policy management, verifiable evidential transactions as well as enforcement of business logic execution. Hence, the DEN represents the infrastructure for trusted and accountable transactions [5].

3 Scenario and Use Case

As stated above, the use case and prototype combine the ideas of the different concepts. In this section, the actual concept of the use case will be introduced. The foundation of the workflow is given by a part of the Dutch Aliens Act responsible for the immigration of highly skilled workers and students. This part of the law is translated into machine-readable code by FLINT in the form of a .json file [1]. This code snipped consists of 4 act frames and several facts, duties and preconditions that are transformed into facts again. The act frames predefine the general workflow of the use case. The applicant that aims for a resident permit in order to work or study in the Netherlands, acts upon these act frames and creates normative facts. The four act frames are:

Act Frames	<<disregarding a temporary regular residence permit>> <<granting application for regular residence permit>> <<rejecting a temporary regular residence permit>> <<granting a temporary regular residence permit>>

Table 1: Definition of the Act Frames

These acts can be met by fulfilling certain preconditions and facts. The structure set and created by these acts that will impose the workflow of the prototype can be seen in Figure 2: Workflow of the Prototype. For the implementation of the prototype we reach out on a specific part of all the acts and facts in order to sustain feasibility. Compare Figure 1: Whole content of Act frames and Facts given by [6] and Figure 2: Workflow of the Prototype in order to get an impression of the complexity and variety of the whole functionality provided by this code snipped [1]. The Figure 1: Whole content of Act frames and Facts given by [6] is rather for showing the whole structure then focusing on the details.

The facts are institutional, and the applicant must provide certain documentation or information to fulfill these conditions, e.g. “alien has a valid border-crossing document”. The key idea of our concept is to meet these facts by integrating independent organizations to the network that deliver proof that this fact is valid or not. You may call this evidence. At this point Evidentia comes to play and provides exactly the framework to build up a system that consists of several peer nodes that act as organizations within the network and create certain evidence whenever needed. For example, the fact “alien has a valid border-crossing document” can be confirmed by an independent border control authority that is registered in the network under the hood of organization “Border Control”. The evidence is put on the blockchain, therefore transparent to every node participating in the network and overall tamper proof, secured by the systems architecture (implications from Hyperledger Fabric). The organizations we obtained from [1] are:

- Coordinator, Requester, Employer, University, Bank, Border Control, Immigration Department, Police, Medical Association

These organizations are registered in the Distributed Evidence Network and part of the prototype we implemented. As you have many different units being part of an organization, Evidentia allows to create many different ETB nodes running under the hood of one organization. In this way, the framework of Evidentia enables the integration of many different institutions as peer nodes. At the operation phase of the system, transactions are carried out as follows. The process starts when the applicant as a requester organization creates a workflow from its personal ETB node. Once the workflow is started certain requests are made and Evidentia automatically discovers an ETB node that has a service for handling these requests and directs these to the intended node. A certain Doctor for instance, that runs an ETB node on its own system, can act on the request and provides evidence e.g. a medical examination for the applicant. These procedures are done many times with different functions, fact and peer nodes, until the whole process is done, and the applicant gets a decision.

4 Prototype and Architecture

With the general scenario and use case in mind, this section aims to describe the prototype and its architecture in detail. As the project obviously has its constraints the implementation of the prototype faces these as well. We want to point out the steps we took and the results we obtained. Before we head on to the discussion where the possibilities, adaptions, limitations, downsides and future ideas will be described.

The network and its functionality is implemented via Hyperledger Fabric named Distributed Evidence Network. Its purpose und functions stay the same as described in section 2. For this prototype the DEN will simply be a local network that runs on a single computer. It contains the nine different organizations with their unique certificate and private and public keys. The main work of the prototype is done via the different ETB nodes that represent each organization. For the prototype, we only use one ETB node for each organization.
The whole workflow of the prototype is implemented via Datalog in the ETB node of the Requester, that acts as an applicant for a resident permit. The workflow is shown as a flowchart in the Figure 2: Workflow of the Prototype. For convenience and comprehensibility, the following table prints parts of the workflow written in Datalog.

```
foreign_national_satisfies_conditions_for_the_granting_of_a_vvr_bep(ApplicantInfo):-
        N_alien_constitutes_a_danger_to_public_policy_or_national_security(ApplicantInfo,Ok5),
        N_foreign_national_will_owe_a_fee_for_processing_an_application(ApplicantInfo,Ok6),
        foreign_national_has_a_valid_provisional_residence_permit(ApplicantInfo,Ok7),
```

foreign_national_has_a_basic_level_of_knowledge_of_the_Dutch_language (ApplicantInfo,Ok8),
foreign_national_has_a_valid_border_crossing_document(ApplicantInfo,Ok9),
foreign_national_is_prepared_to_cooperate_in_a_medical_examination(ApplicantInfo,Ok10),
purpose_for_which_the_stay_is_authorised(ApplicantInfo).

Table 2: Part of the workflow within the ETB node "Requester"

The applicant must prove that he satisfies the conditions for the granting of a resident permit. This fact invokes several additional embodied facts, that must be provided. These conditions are connected via AND operator and the master fact is only true for having fulfilled every other condition. The way this works, is that the facts are queried sequentially. Each fact is bounded to a certain organization that is able serve that fact. "N_alien_constitutes_a_danger_to_public_policy_or_national_security" will be processed to the responsible organization "Police" and the applicant can find a registered police authority that will set the fact to True or False. Further development could lead to more sophisticated information that just a True or False, that could lead to more complex decision making but for the prototype, we stick to this binary decision type.

5 Discussion and Conclusion

In this section we want to present our results and discuss important aspects whether they are positive or negative. Furthermore, it is crucial to mention that the prototype is indeed not more than an idea of a system that offers a lot of potential but is far from being a product such an important public service like the immigration department can rely on. Therefore, the first part of the discussion is about the limitations and problems we were facing the process of implementing the prototype.

On the one hand, Evidentia provides a lot of functionality regarding our concept and the general idea and structure offer a great deal of possibilities, but on the other hand Evidentia is still on an early development stage. That causes lack of features and problems at some extend. There are some proposals, we want to discuss in order to improve the Evidentia framework, especially regarding our use case. Firstly, the functionality to embed a workflow within another existing workflow. That would enable building more complex structures like a whole Dutch Aliens Act would impose. Currently, you can only build a system with one main workflow. It also would allow us to implement workflows in parallel and moreover implement two separate workflows, one for the immigration department and one for the requester. You basically need both in order to secure the workflow by the immigration department but also offer the requester enough flexibility to call specific peer nodes that provide a certain service (e.g. a specific doctor, that he or she went to). Secondly, there is an issue when adding a service to the ETB node. Whenever a service is added, the “Generic Wrapper” object in Java is called and creates the wrapper functions of the specific service. This is a problem because this would destroy the predefined ETB node and all its service functionality. You must add a service in order to register the ETB node to the network. In the long run, you want to hand out a predefined and complete ETB node to run at a server of a doctor, other than letting the user prepare the functionality of the node. You can adapt to that by coding the generic wrapper with the functionality of every organizations manually, but that seems a bit unpleasant and not necessary if you would design that in a different way. Thirdly, a method, that allows to explicitly choose a certain ETB node that should execute a service is missing. The relates back to the point that you want to build a system where many different ETB nodes of the same organization are part of the network. You need a functionality that allows that flexibility, if you planning to put it on a bigger stage. Another crucial feature, that is missing currently, is a method to predefine and fix the services a certain organization can offer and execute. This is key, as the whole architecture makes no sense if an Employer node can provide medical examination by just adding this service manually. We might suspect that is done in the DEN and we were not able to discover that sufficiently during the implementation stage but wanted to point out this feature is not part of the prototype functionality, but it is crucial on a certain level. Another aspect regarding automation, are the circumstance that you have to prepare the CA-certificates manually in each ETB node. That means copy and paste all certificates created during the process of ramping-up the DEN into the intended folder of each ETB node. The more extensive the project gets, the larger the influence of this aspect is. Fifthly, when executing a service in a workflow and the output of the service consisting of a file, e.g. json format, then errors occur when another service wants to access this output. As the subsequent service depends on the content of the preceding service, the system does not allow this pass. It is currently only working when using string variables. This functionality will be crucial in complex systems where you want to exchange sophisticated data not only string variables. A last aspect that limited our prototype to a certain complexity, was processing power in combination with the systems capacity management. After a certain amount of running ETB nodes and added claims within these nodes the RAM is completely used. We are not totally sure, if there is some leakage within the Evidentia structure, but we thought this is worth mentioning as this might be an issue someday, when you apply this structure to bigger prototypes.

Accordingly, to the early stage of Evidentia, the FLINT system is also in a development phase and therefore has its flaws and downsides that occurred to us during this project. We want to discuss these problems as well in order to initiate development as we think that this domain-specific language can offer a lot of potential regarding automating law text and legal processes. The greatest problem, that is still unsolved, is the missing option to automatically generate workflows and services by parsing FLINT files. There is a machine-readable version of the law compressed in json format but this itself gives no automation. There is a great need for a intelligent parsing system that enables to gather information and structure from this machine-readable code snipped. Otherwise and like in our case, you need some human perspective to create structure from this code file. The json format is indeed better to interpret the structure for using it in another software application, as it predefines statements that can be distinguished via AND, OR operators than rather gaining this structure from raw law text, but for a true automation a more sophisticated parsing system is essential. The code snipped [1] is further missing a consistent denomination of all the fact, acts and duties. There are facts that represent the same conditions and normative facts, so are equal, but named slightly different. This condition is insufficient. This will not allow a parsing system to work properly (besides using a human-like AI that is able to handle natural language in a very interpretive way), but also complicated the process of finding structure for us a lot. Besides translating the whole file into English language, this condition was quite time consuming for us. Thirdly, in order to create a fully automated process in our framework with Evidentia, FLINT itself will not provide any information about which organization is allowed to provide certain evidence. We had to come up with certain authorities, that might provide certain evidence. This should be written down in law form as authorities act in their scope laid down by legal organization, but FLINT does not include these scopes. Moreover, FLINT will not determine the input and output for certain services that must be implemented by the system. For instance, the fact “applicant is willing to do medical examination”, we had to imply the output to be Boolean and the input to be information about the applicant. This is not stated in the FLINT file but implied by the human interpreter.

However, the acts, facts and duties defined in FLINT give a solid structure how to generate the desired evidence and offers potential to enable automated decision making based on legal text.

Besides these technical issues by the used and implemented systems FLINT and Evidentia, there are further preliminaries that must be satisfied in future applications. The first major challenge to integrate the system on a larger stage, is that all involve parties need to adapt the system and taking part in the network as ETB nodes. Therefore, a simple browser application or app is crucial to ensure usability, as well as convince a doctor for instance to be part of this process. Another crucial aspect is that this technology forces the law to adapt. Law text must consider that the evidence we want to create is valid otherwise the system is illegal. You might compare this to automated driving, where you create new circumstances and law obviously has to adapt to these new findings. Furthermore, law text must adapt to more Boolean decision making if that is possible. There are decisions that might be soft, in a way that you can not define a hard cut but let it flexible to more human interpretations. This is an issue and we are not fully aware if that is completely solvable. Creating fully automated decision processes without involving human interpretation as part of the process itself, might be infeasible. Another technical aspect is the question who is running the Distributed Evidence Network. Will it be done by the immigration department only? You cannot expect from a certain organization that is hosts the whole network. Therefore, you have to consider a precise solution who is in favor of the DEN, so that is not quite a centralized system again. This also implicates the need for different Orderer nodes that validate all the actions going on while the process is running. Will this also be done by the immigration department only? These questions will need to be answered when implementing it on a bigger stage. An optional suggestion that we were thinking of is an organization that is registered in the network but is not part of the evidence process itself but acts like an overviewing organization. This could be done by journalistic authorities or NGOs acting like The Fourth Estate within the process. This idea leads to more transparency reconciling the initial goal formulation.

Acknowledgements

We would like to thank Jeroen Breteler from TNO for introducing this project, his valuable contribution, helpful suggestions and overall support. Furthermore, we would like to thank the three advisors of the course Dian, Matthias and Thasos for their feedback and support. Especially, Tasos for the support regarding the Evidentia system. The work was done in the course of the module “Blockchain technology for public sector innovation” and is supported by TUM.

References

[1] Discipl. *Vreemdelingenwet.flint.json*. https://github.com/discipl/discipl-law-reg/blob/cb6472ec9ccba673622238982a33337aa3b12af8/test/Vreemdelingenwet.flint.json. Accessed 12 July 2020.

[2] Federal Office for Migration and Refugees. 2019. Development of a GDPR-compliant Blockchain Solution for the German Asylum Procedure.

[3] Immigration and Naturalisation Service. *Researchers: IND's biggest problem is a lack of decision-making capacity*. https://ind.nl/en/news/Pages/Researchers-IND%27s-biggest-problem-is-a-lack-of-decision-making-capacity.aspx. Accessed 12 July 2020.

[4] Immigration and Naturalisation Service. 2017. The IND in 2016 – Annual Report.

[5] Nigam, V., Beyene, T., Balta, D., Cherifa, T., Ruess, H., and Shankar, N. 2019. evidentia: Evidential Transactions for a Digital World.

[6] van Doesburg, R., van der Storm, T., and van Engers, T. CALCULEMUS: Towards a Formal Language for the Interpretation of Normative Systems.

[7] van Doesburg, R. and van Engers, T. 2019. Explicit interpretation of the Dutch Aliens Act: Specifications for Decision Support Systems and Administrative Practice.

Appendix

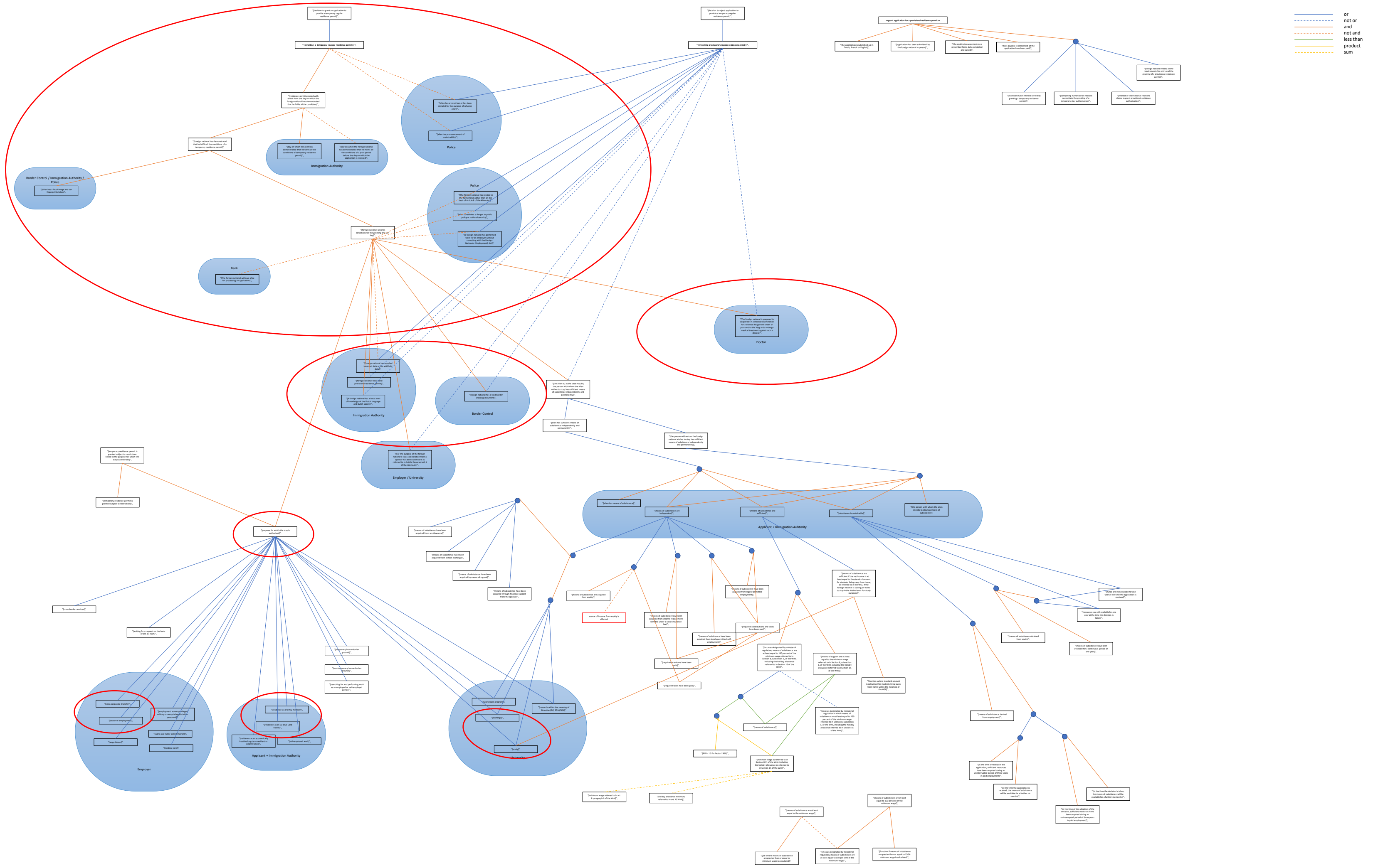

Figure 1: Whole content of Act frames and Facts given by [6]

The red circles show the parts that lead to the Figure 2: Workflow of the Prototype

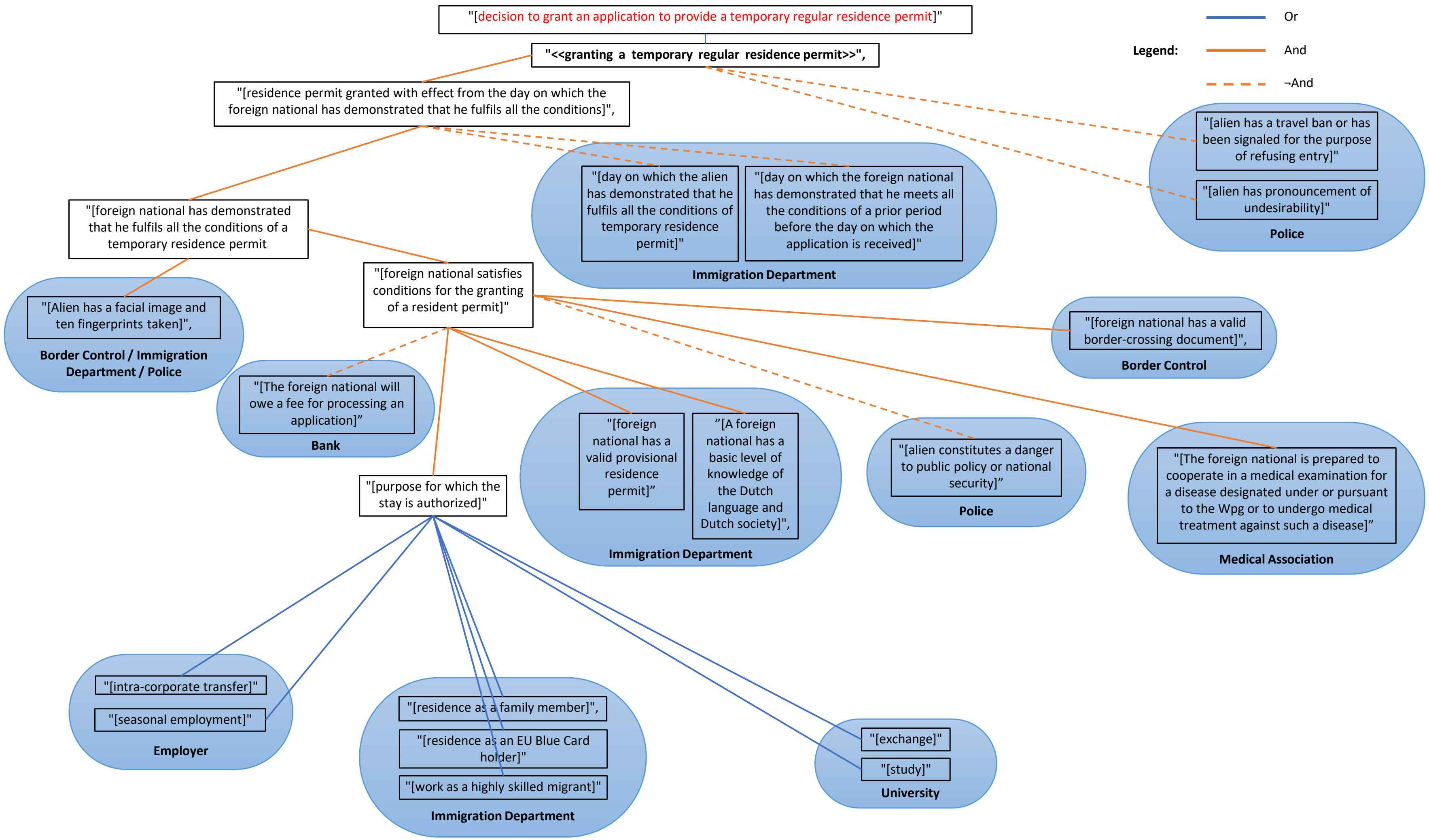

Figure 2: Workflow of the Prototype