Application of RPA for Cross-border Business Processes based on the Example of Intra-Community Supplies

Dominic Alexander Neu (corresponding author)1,2[dominic.neu@dfki.de](mailto:dominic.neu@dfki.de)

Alessandro Benke1,2  
[alessandro.benke@dfki.de](mailto:alessandro.benke@dfki.de)

Robert Müller1,2  
[robert.mueller@dfki.de](mailto:robert.mueller@dfki.de)

Peter Fettke1,2  
[peter.fettke@dfki.de](mailto:peter.fettke@dfki.de)

1Institute for Information Systems, German Research Center for Artificial Intelligence (DFKI), Saarland Informatics Campus, Saarbruecken, Germany  
2Institute for Information Systems, Saarland University, Saarbruecken, Germany

# Abstract

Cross-border business process usually require significant amounts of paperwork. This contribution shows how several software robots can be synchronized in order to automate complex business processes. A novel approach of concurrent legal assessment capable of significantly reducing processing times is presented which is only possible by employing software robots for legal assessment. It is demonstrated how the Heraklit modelling language can be employed to design information architectures and information flows within and between multiple bots.

Keywords: RPA, intra-Community Supplies, Tax Process Automation, Heraklit modelling

# Introduction

The complexity of performing business grows continuously. An increase in compliance partly causes this. Under normal circumstances, tax compliance is a very labour-intensive and time-consuming aspect, contrary to other business activities. This applies in particular to the area of value-added tax (VAT) compliance. In the EU, transactions between entrepreneurs are often carried out across borders, so-called intra-community supply. Fraudulent entrepreneurs commit VAT evasion amounting to 60 billion euros every year by incorrectly claiming input VAT reimbursements. (Ismer & Schwarz, 2019) This fraudulent behaviour has led to extensive compliance regulations in this area, which are complex and highly penalised if disregarded. (EU-commission, 2017)

To ensure VAT compliance, mass data is processed. Therefore, automation technologies such as Robotic Process Automation (RPA) and Artificial Intelligence (AI) are interesting tools for standardisation. (Fettke & Risse, 2018) A variety of currently human routine tasks is necessary to fulfil VAT compliance. For this reason, VAT is suitable for implementing automation initiatives. Robot process automation (RPA) strives in this context since it works on the user interface, enabling lightweight integration across the various systems involved (Bal, 2019). In business practice intra-community supplies demand significant amounts of manual work like collecting and evaluating documents. This compliance work causes friction in intra-community trade and leaves opportunities for fraudulent activities. Our solution helps to integrate and evaluate information from different sources and thus standardizing and automating the process.

Typical applications for RPA are a combination of manual human-centric work and already automated tasks of existing systems. RPA provides software artefacts (so-called bots or agents) that interact with different systems, enter information, and thus at least partially replace repetitive human work. (van der Aalst, Bichler, & Heinzl, 2018) In connection with tax processes, automation implies the technical implementation of a process without human or minimal human intervention. RPA replaces human intellectual work (Houy, Hamberg, & Fettke, 2019). In contrast to the automation of processes in ERP systems, in which processes and the system must be changed, the ERP system is retained by RPA, and existing processes can be automated. (Czarnecki & Fettke, 2021, S. 4) Challenges in the tax area exist due to various external stakeholders involved in tax processes, missing system standardisation in cross-border situations causing system and media breaks. This is underlined by the current literature. Classical administration and accounting tasks are tackled through RPA case studies (Cooper, Holderness, Sorensen, & Wood, 2019) (Houy, Hamberg, & Fettke, 2019), while taxation processes remain rather unexplored territory. (Sala, 2020) stresses the importance of tax automation while suggesting starting with repetitive value added taxes and (Mezzio, Stein, & Stein, 2019) highlight the importance of understanding and analysing tax processes prior to automation.

As a consequence, the following section gives a brief definition of Robotic Process Automation to emphasise the benefits of automation through RPA in the context of Business Process Management (BPM). Then the relevance of accurate process modelling as a transfer format from the business and compliance needs onto the implementation details of RPA robots is discussed. Finally, we present Heraklit as an infrastructure to model computer-integrated systems to accomplish this task. (Houy, Hamberg, & Fettke, 2019, S. 63)

# RPA: Problem Solutions Research Field

Robotic Process Automation has become an increasing trend in research and practice. A broad variety of cases have been presented (Romao, Costa, & Costa, 2019) (Aguirre & Rodriguez, 2017). RPA is an innovative approach to transform the process execution without changing the underlying application system. First, authors have characterised those systems by the ease of implementation – no or only minimal programming effort – and the simple integration in existing application landscapes through user interfaces. Others emphasise the imitation of tasks that a human would do (van der Aalst, Bichler, & Heinzl, 2018). Defining the term RPA is not that easy, as it is more an idea inspired by practice instead of a carefully developed concept. It is, therefore, a term used for a wide variety of automation techniques.

Increasing the automation of a process through technological progress is not a new idea but a core topic of various business, informatics, and engineering disciplines, that has been discussed for decades. The innovation potential resulting from the interaction of processes and technology was already addressed in the 1990s in the context of business process reengineering. The core focus of those traditional process automation approaches is an execution of process flows according to mainly static requirements that are either hardcoded into application systems (ERP, CRM, …) or defined in a business process management system (BPMS). While these systems provide highly reliable integration of many tasks, they imply the costly and time-consuming process of customisation and adaptation to the needs of the individual processes in the enterprise.

The automation of business processes via software robots, on the other hand, starts with another perspective: The manual interaction of humans with existing applications is observed with the goal of imitating this behaviour with a software robot. This bot then takes over the execution specific activities, which contains the handling of involved application systems through the existing presentation layer. Hence, neither processes nor application systems are changed, and data is only handled via existing applications (Czarnecki & Fettke, 2021).

The functional scope of an RPA system is thereby defined through the selection and description of the process to be automated. Individual activities might range from simple emulation of routine tasks to autonomously reaching complex decisions. Although automation starts with the observation of human actions, proper automation of a manual process does not necessarily mean that all tasks are performed in an equivalent manner (Czarnecki & Fettke, 2021). This makes it difficult to understand and manage the transformation from manual to automated processes. Furthermore, it might be helpful to extend the software robot by further functionalities that could help to improve existing processes or even develop entirely new processes.

The main benefit of RPA lies in the lightweight integration of systems. This opens the opportunity to automate activities, where the economic benefit potential is usually in a range that would not allow a substantial reengineering project. Additionally, the idea of a software robot implies a flexible and agile entity. The composition of different software robots forms a highly dynamic and complex system.

To manage and control the transformation process of manual steps to RPA bots, we propose the modelling language Heraklit. It has been developed to depict the dynamics of business processes and their interplay with information systems while accurately tracking the role of static data elements.

Through the usage of Heraklit, the status quo of the business process can be described with the same language as the automated bot-based process. Heraklit will be used to model "what" an RPA-bot will do instead of "how" a bot will execute its actions. The model thus abstracts from the concrete implementation of a bot but is tangible enough to be used as an implementation instruction. As will be shown in chapter 3.3, the language also allows documenting the execution of bots on an abstract and comprehensible level.

# Methodological approach: Modelling RPA-based systems with Heraklit

Heraklit is a general-purpose modelling language based on symbolic petri nets. Heraklit summarises objects and operations on objects of all kinds in a structure: individual objects, documents, abstract objects, services, data, deadlines, invoices with payment terms, composed objects et cetera. The signature is a purely symbolic representation of several structures at the same time. Structures and signatures allow Heraklit to model static aspects of business processes.

The dynamics and state changes of individual objects are illustrated via the well-founded and proven concept of petri nets.

Lastly, the architecture of larger complexes can be described through the notion of modules. These have interfaces with gates, allowing them to interact with other modules. (Fettke & Reisig, 2020)

A distributed run is a causal petri net, containing one event **and** its predicates for every transition that fired. It therefore describes the occurrence of actions. (Reisig, 2013)

These features render Heraklit an ideal toolset for modelling RPA-based process automation. The modular architecture allows for modelling complex systems of interdependent bots performing complex tasks. Using distributed runs, it can be shown how information is processed and passed on by and between the bots. Heraklit will be used in the context of a design-oriented research approach. It is employed to describe the information architecture and the dynamic of transforming information objects in a precise way.[[1]](#footnote-2)

# Case Example: VAT, Heraklit -Model

## VAT-Compliance in Intra-Community Supplies

Intra-Community supplies can be exempted from VAT, according to Art. 138 VAT-Directive (2006/112/EC)[[2]](#footnote-3). Depending on the perspective, intra-community deliveries and intra-community-acquisitions are differentiated.[[3]](#footnote-4) In simplified terms, the entrepreneur making the supply must fulfil three conditions to qualify for exemption: Confirmation of the actual transport, determination of the entrepreneur status, and the receiving goods are subject to VAT (Amand, 2016, S. 98).

Intra-Community supplies may involve two parties, but they get significantly more complex when several companies are involved. In a chain transaction, for example, the supply is only physically moved between two parties, but the financial transaction is between at least three parties. (Dietsch, 2018) For simplicity reasons, this chapter covers only intra-community supplies between two parties.

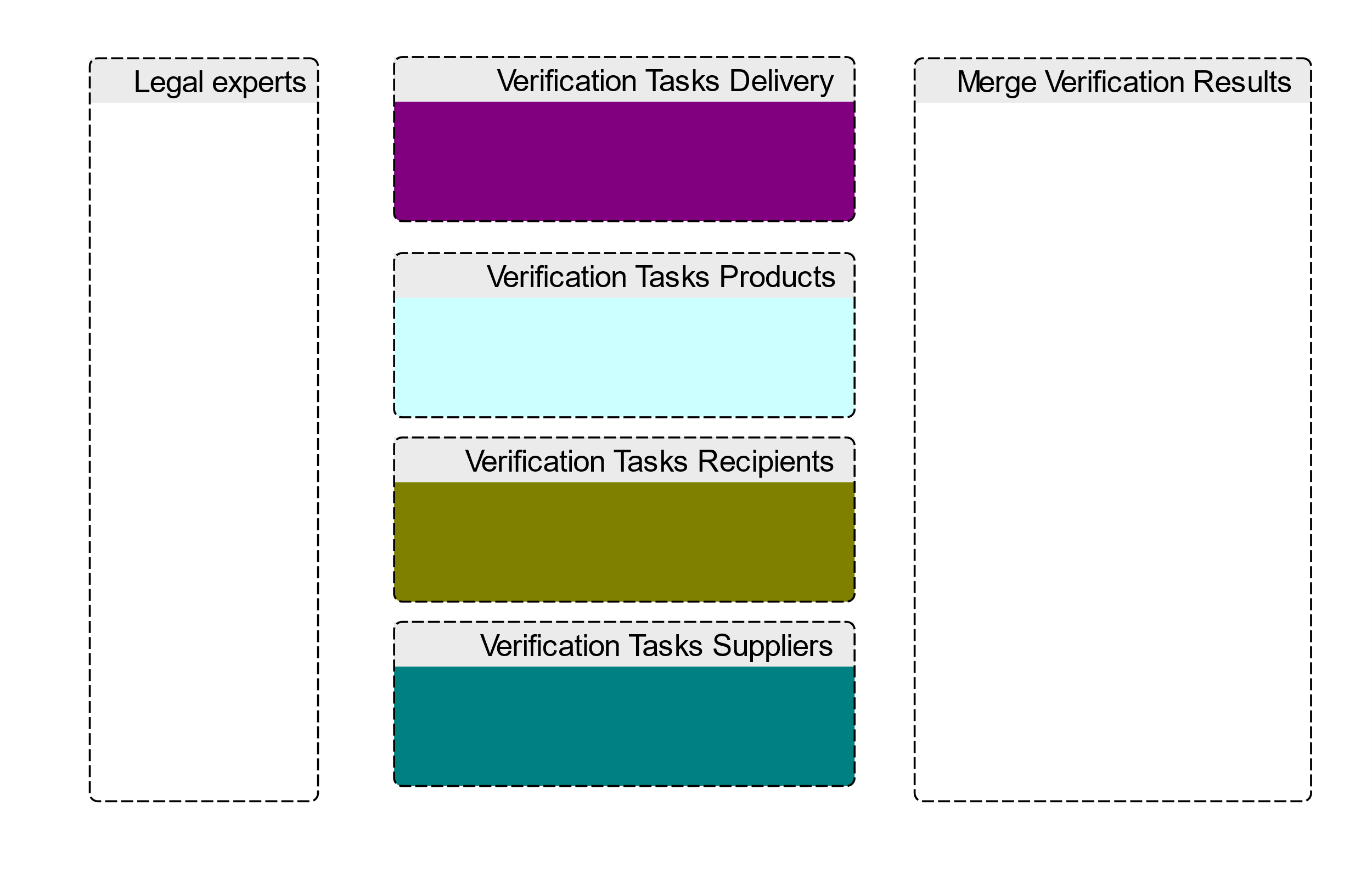


Figure 1: Module composition with legal experts

The high amount of value-added tax poses a substantial risk for entrepreneurs if misjudged. In Addition, an intra-community supply can be revoked by the cause of missing or incomplete documentation. This is especially challenging for small and medium companies with non-regular transactions to unknown recipients. For these reasons, the intra-community supply plays well as a candidate for Robotic Process Automation: high volume, standardised process, well-described decision rules and need for precise documentation.

In the following, the status quo of the verification process as it is usually performed by legal experts is shown. The process can be broken down into four categories of verification steps and the final consolidation of the individual results. The verification steps themselves are executed by a legal expert in an arbitrary sequential order. The documentation of a legal assessment, on the other hand, always follows a strict order to make it more readable for others. To model this process, we therefor differentiate between six different modules. The arbitrary sequential working order of the legal expert is show in the module “legal experts” (figure 2). The module “Merge Verification Results” takes the individual results and merges them sequentially according to the legal best practice (figure 3). Whereas the remaining four modules are used to categorise the verification tasks. They do not contain any dynamic behaviour but simply pass the result from the left interface of the module to the right interface. The composition of these modules is shown in Figure 1.

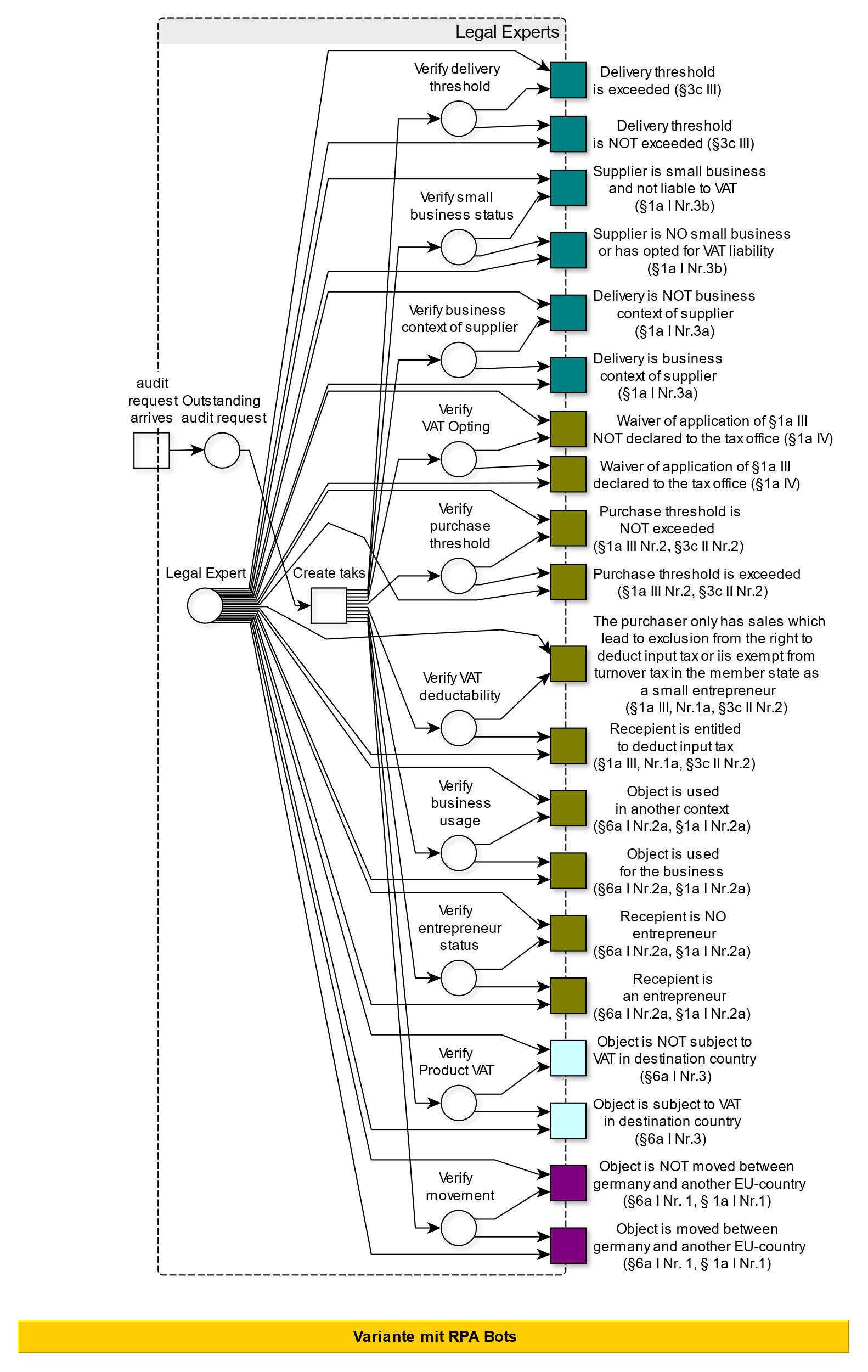


Figure 2: Module Legal experts with inner dynamics of verification steps

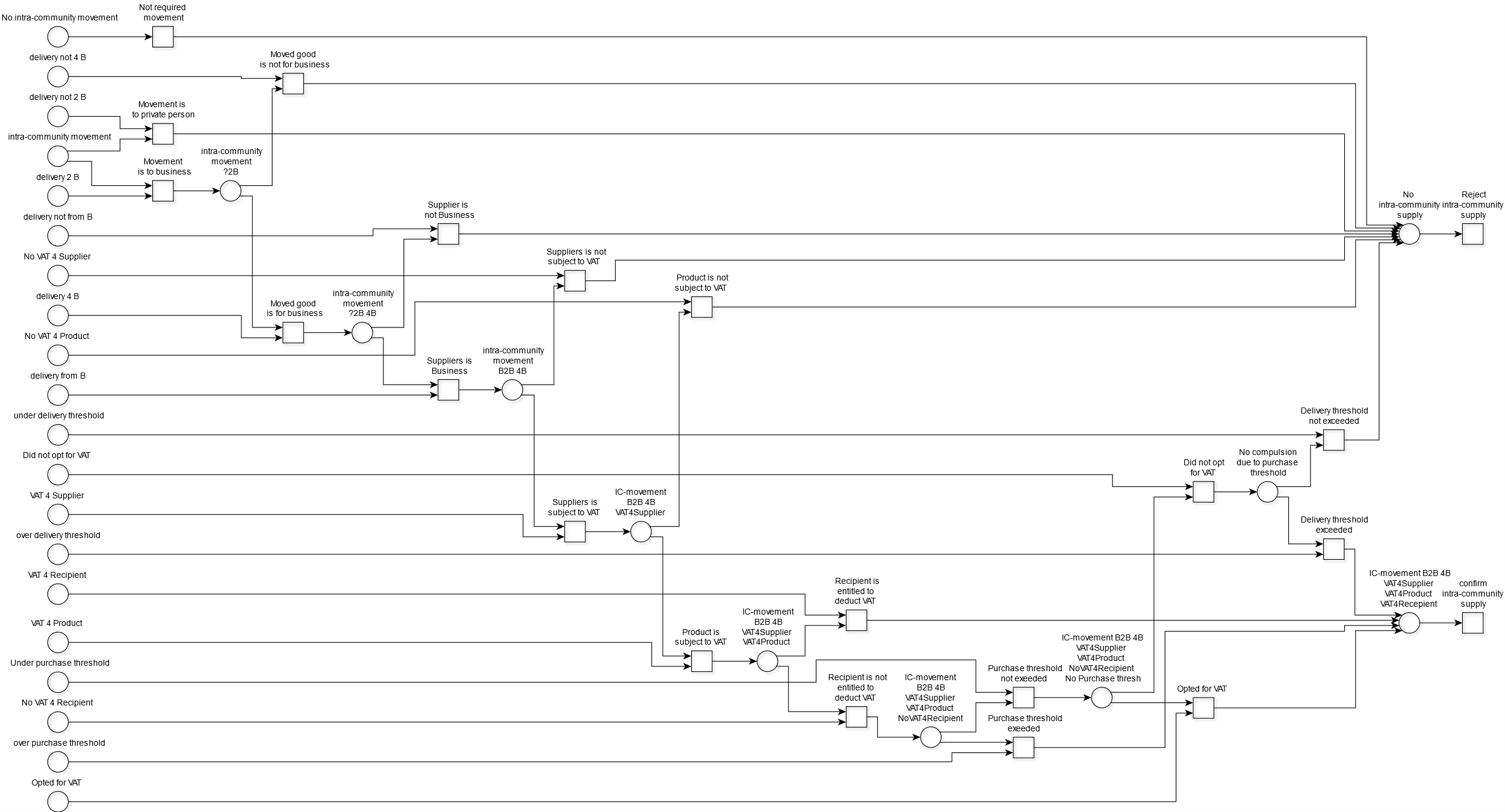


Figure 3: Sequential Merging of results for the documentation (2 and 4 are used as abbreviations for “to” and “for” respectively. Furthermore, B is the abbreviation for “business”)

Although this approach is directly transferable to a bot-based assessment, there are several arguments to split this task into several independent verifications. By reducing the scope of one bot to a smaller process chunk, the maintainability of the systems is increased. This modularisation concept (or micro-services) is well established in software development (van Vliet, 2008). Furthermore, the modularisation of the verification process enables the abstraction of lower-level actions to higher-order activities/subprocesses and thereby enhances the understandability of the system. Finally, by breaking down the work into smaller chunks, the resources can be scaled more precisely to the need of each individual verification task.

Another problem of the process model in figure 2 is the detailedness. Although the model might be sufficient as a work instruction for a legal expert, it lacks more detailed instruction on the data involved in the process and the sub-steps necessary to perform each action. The model is not sufficiently concrete to be used as an RPA-Bot construction manual. We, therefore, propose a second, more detailed module (RPA Solution) that clearly outlines the required data for each step and provides actions of lower granularity that are ready to be implemented algorithmically. The module can substitute the "legal experts" module in figure 1 while the verification tasks and the merging of the results remain identical.

Since the law is a science in which much depends on contextual circumstances, we must presuppose some premises so that the algorithmic implementation corresponds to the legal examination procedures. Some of these premisses are general practices accepted by the courts; others are enterprise-specific and need to be evaluated individually by the applicant.

## RPA-Solution model

To model the legal assessment of intra-community supplies, the modelling language Heraklit is used. It amalgamates the detailed description of static data involved and the dynamic behaviour of a process. In Addition, its strict formalism simplifies the transformation into an algorithmic implementation.

To start, the static data objects involved in the process are enumerated and composed. Together they form the signature of the model:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Base Sorts** | | | **Variable names** | |
|  | Legal entity | | enty, enty\_1,  enty\_2 |
|  | Product | | prod, prod\_1 |
|  | Product Category | | categ |
|  | VATID | | vatid |
|  | Country | | cntry,  cntry\_1,  cntry\_2 |
|  | Status | | stat, stat\_recv, stat\_supp, stat\_deliv, stat\_prod |
| **Derived Sorts** | |  |  | |
|  | Company | Legal entity × Country | comp |
|  | Partner Status | Company × Status | - |
|  | Receiver | Company × VATID | recv |
|  | Order Positions | P(Product) | opos |
|  | Order | Company × Receiver × Order Positions | ordr |
| **Functions** | | | | |
|  | cat: Product -> Product category | | |

Table 1: Signature of the model

Through this signature, the required static elements are further characterised without a reference to a specific data format. It should be noted that the sort "Order" can either be an invoice (for the recipient of an intra-community supply) or a purchase order (for the supplier). Although the logical structure of an invoice is prescribed in more detailed form (e.g. invoice number, tax rate, etc.), the signature is reduced to the data elements necessary to perform an intra-community-supply verification. The invoice is assumed to conform with additional formal requirements from other regulations.

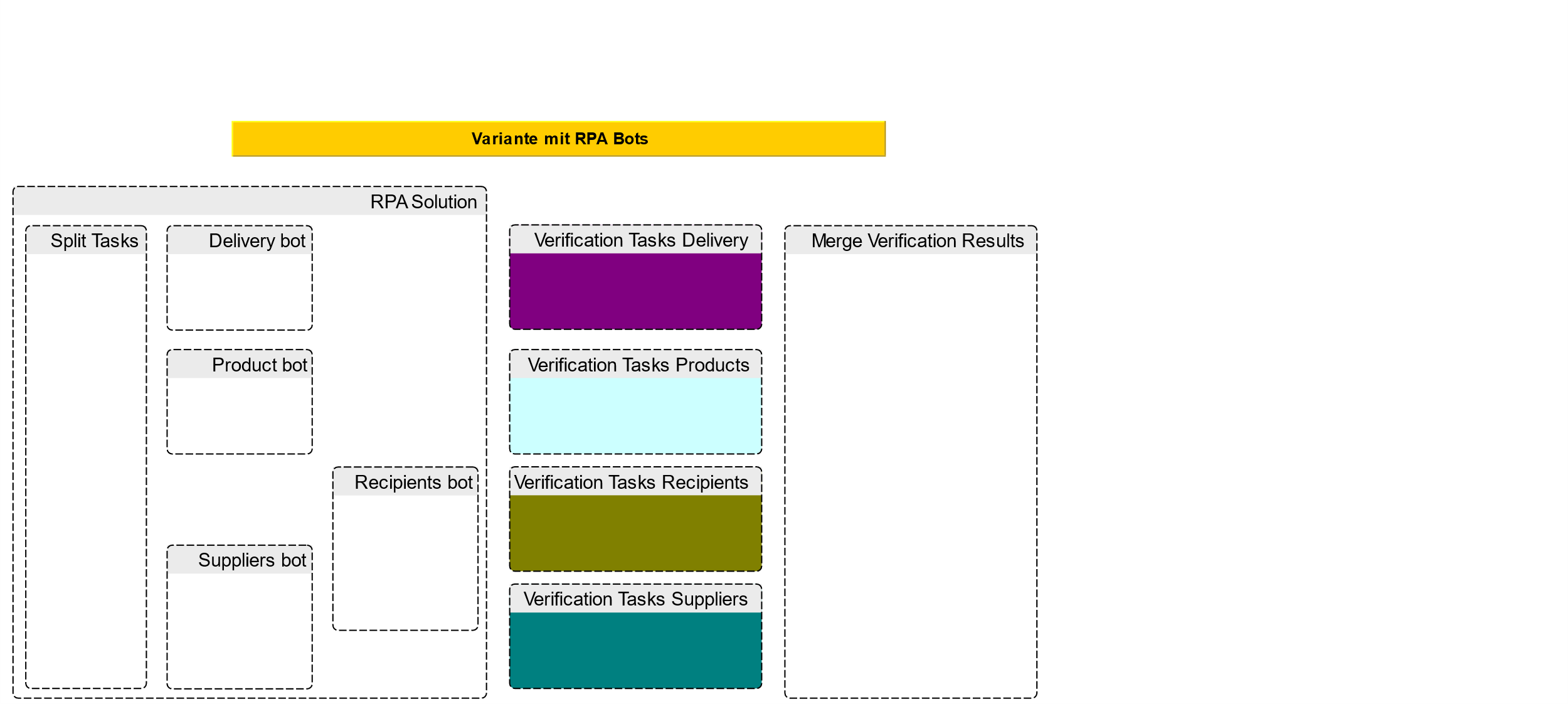


Figure 4: Composition of Module "RPA Solution"

Following the static elements, the composition of each submodule is presented in figure 4. An order can arrive at any time, and when it does, it is split into several smaller data objects, each containing only the necessary information to perform the following verification. This is done in the module Split Tasks. Afterwards, each individual verification is performed in a separate module. The modules Verify receiver and Verify supplier investigate the required status of the companies involved and share a common data object, the Partner Status, but other than that, they are independent. The module Verify delivery checks for sufficient documentation of the shipment to the foreign EU country. The module Verify products ensures that the individual products are subject to the VAT in the destination country. At last, the module Merge Verification results merges all four outcomes and proposes one final assessment.

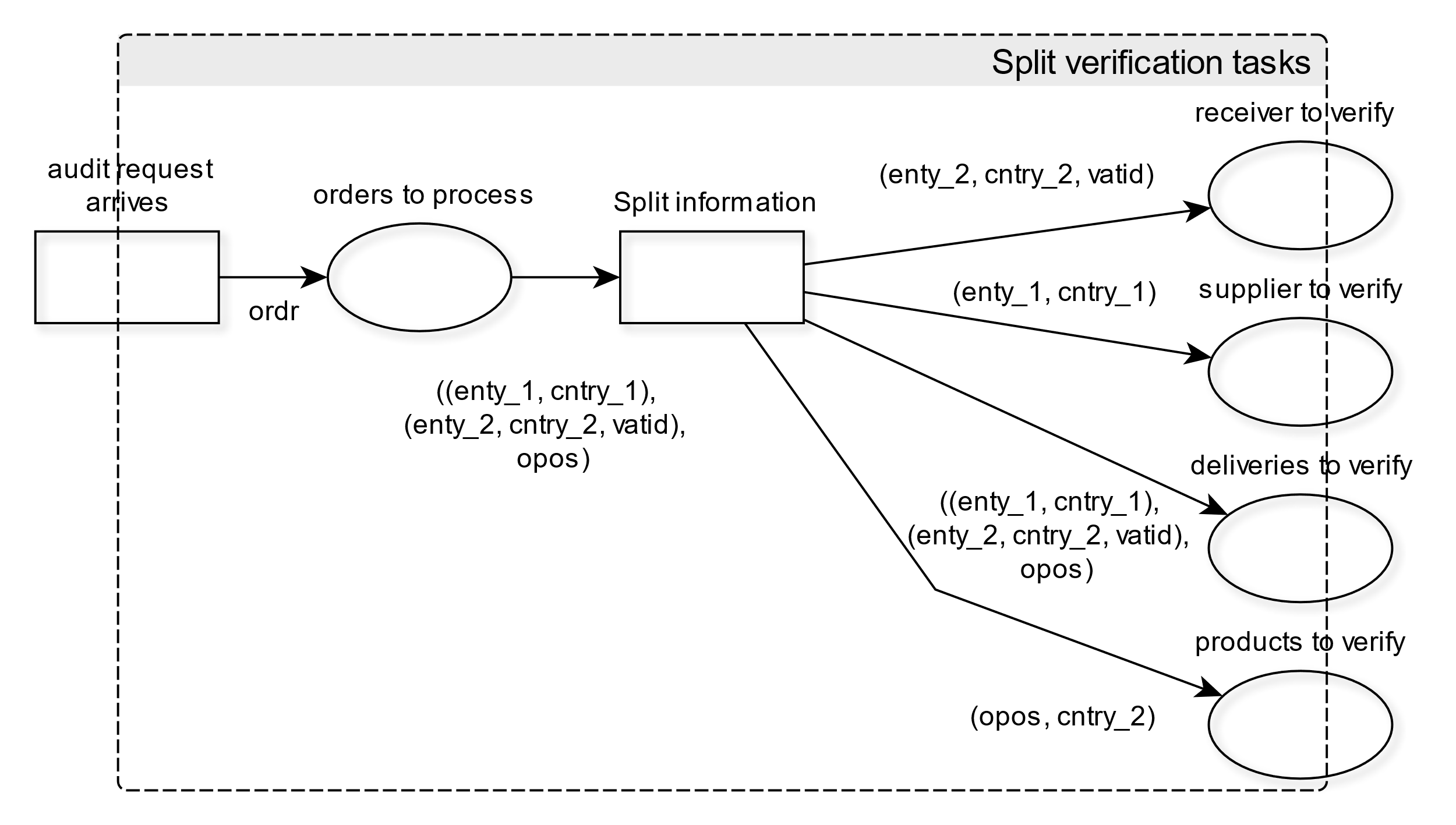


Figure 5: Module "Split Verification Tasks"

The assessment of the receiver is split into three consecutive steps. First, the specification of the VAT-ID is verified. By this, the receiver signals that this purchase is on behalf of a company liable for turnover tax, and it is intended for usage in his business.[[4]](#footnote-5) Afterwards, the VAT-ID is checked for its validity, which, in Germany, can be done through the portal of the Federal Central Tax Office[[5]](#footnote-6). Although an invalid VAT-ID is handled equivalently to a missing VAT-ID, further inquiries to the receiver might be an appropriate escalation step before rejecting the intra-community supply. As a final step, the status of the receiver is retrieved from an internal customer database. The status signals whether a company is excluded from intra-community supplies according to some legal norm. Since this status is somewhat unlikely to change throughout the year, it is updated periodically in the database and not checked individually for every invoice.

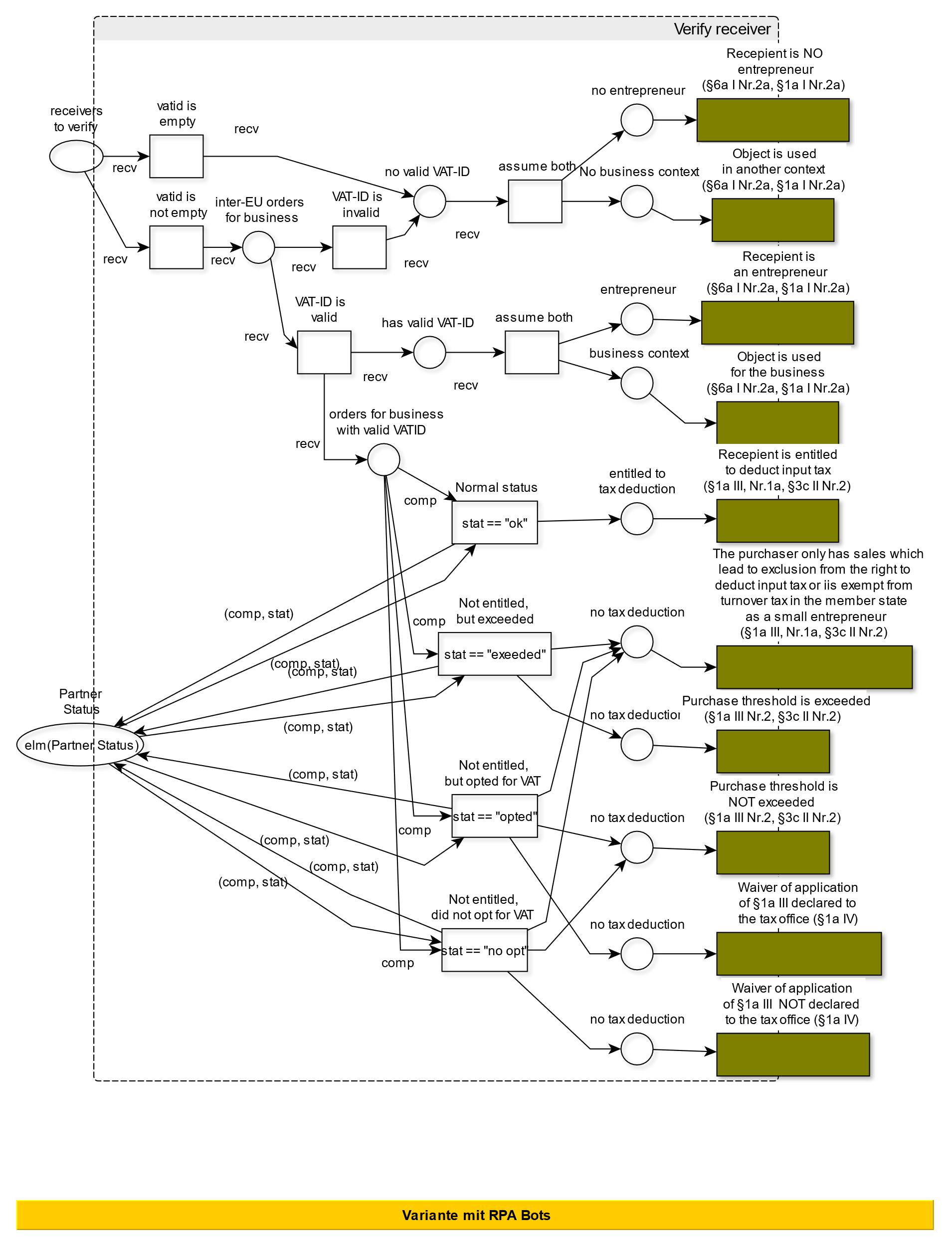


Figure 6: Module "Verify Receiver"

This premise will also depend on the company in focus. For example, a manufacturing enterprise will likely have just a handful of clients in fixed countries. In this case, the partner status might be updated yearly in a spreadsheet or the ERP system. An online retailer, on the other hand, would encounter new clients from different countries daily. In this setting, the verification of the partner status itself should as well be modelled and automated.

The verification of the supplier follows a similar pattern to that of the receiver, with the distinction that a supplier does not specify a VAT-ID. Therefore, it is only the status in the database that is verified.

The verification of the delivery is divided into two steps. Firstly, the requirement of delivery between two distinct EU countries is verified. This is achieved by comparing the two countries of each partner involved. For the delivery verification, it is not sufficient that two EU companies are involved. The proper execution of the product movement from one country to another must be documented via confirmation of arrival. Therefore, as a next step, the existence of confirmation of arrival relating to this order is ensured.

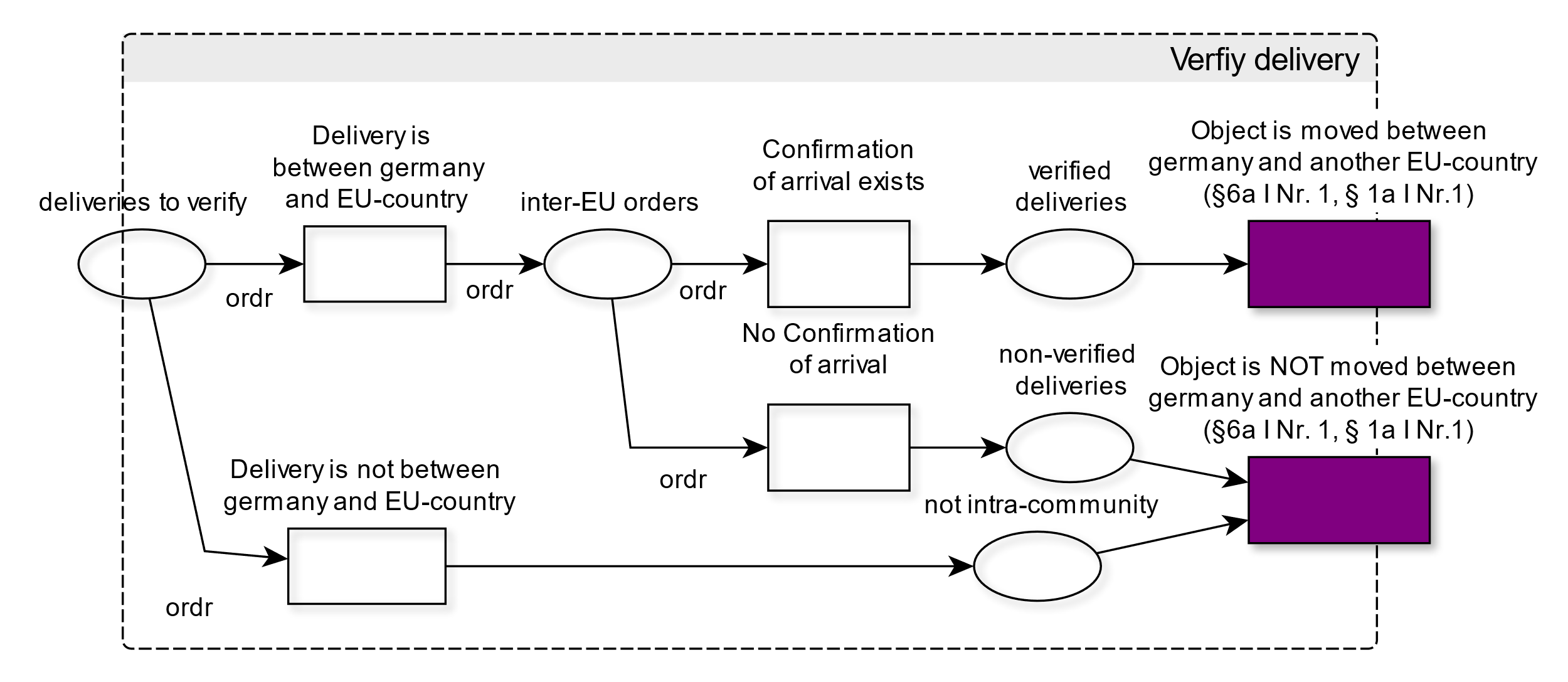


Figure 7: Module "Verify delivery"

National law and court decisions strongly influence the exact formal requirements of a valid confirmation of arrival. (Langhein, 2017, S. 144 ff.) The detailed validation of a CoA is therefore abstracted in this context. However, in a practical setting, the verification of the CoA itself should be modelled as an inner module containing the detailed verification steps.

The remaining verification task checks for each product if it is subject to the VAT in the destination country. Although this step is part of our model, the explicit implementation will depend on the required rigorousness defined by the company and the products in question. Many exceptions differ throughout the different member states of the EU. Depending on the products sold by the company, several approaches might be recommendable: assume the product is always subject to VAT, infer the result from the product category or hand over each case to a human worker and process the result. In our model, we infer the tax liability from the product category.

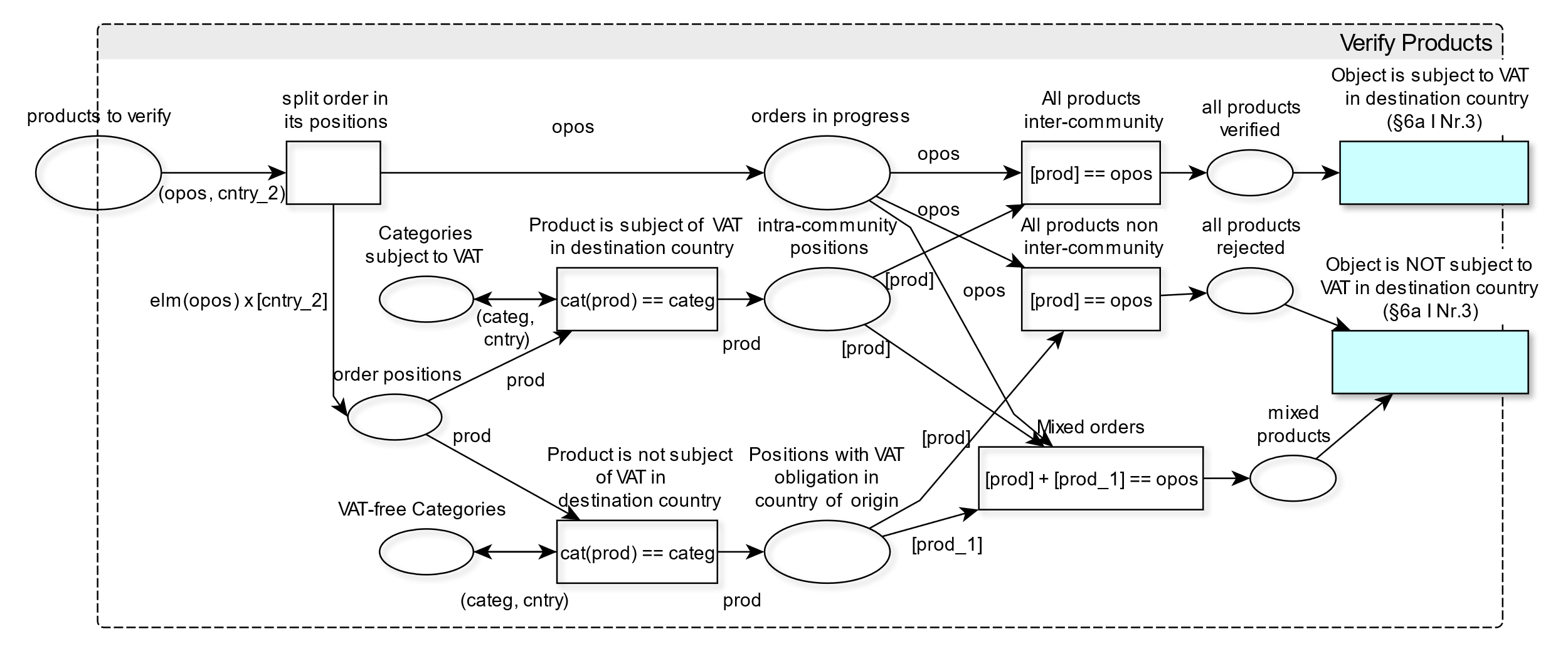


Figure 8: Module "Verify Products"

Like the invalid VAT-ID, another escalation step to handle mixed orders might be recommendable, but mixed orders are treated as non-VAT liable for simplicity reasons.

After each verification has been executed concurrently, the intermediate results are merged into a final one in the module Merge Verification results. This is basically a logical join on all the possible combinations of individual module outcomes and is identical to the merging performed by a legal expert.

## Intra-community supply: Verification example

To present the necessary steps in the legal assessment of VAT exemption, the verification schema is presented by one exemplary case. The evaluation is generally based on the German VAT act.[[6]](#footnote-7) However, not only the national legal situation is decisive, but also the legal situation in the other Member States (e.g., the delivery and the acquisition thresholds are specific to each country). Furthermore, the national laws and court decisions further influence the formal requirements concerning the invoice and the assessment documentation.

In this case, the Spanish Medical supplies SE delivers cough syrup to the German MyCorp GmbH. Cough sirup belongs to the product category of medicine and is subject to the VAT in Germany. Moreover, the Germany Company supplied its valid VAT-ID when placing the order. This action signals the Medical Supplies SE that MyCorp GmbH is liable to the VAT in Germany and that the cough syrup will be used in the company's business context. MyCorp GmbH and medical supplies SE are known business partners; therefore, it has been verified for each company that they have not been excluded from vat exemption through some legal norm.

The statics of this verification example are shown in Table 2, where each sort is initialised with its corresponding data elements. The events of the verification process are aligned with their connected data objects in the distributed run of the schema. Figure 9 shows the distributed run for this example and also constitutes a precise documentation of the verification process and its result.

|  |  |  |  |
| --- | --- | --- | --- |
| **Base Sorts** | | **Variable names** | |
|  | Legal entity | [MedicalSupplies SE, MyCorp GmbH] |
|  | Product | ["cough syrup"] |
|  | Product Category | [medicine] |
|  | VATID | [DE314159265] |
|  | Country | [Germany, Spain] |
|  | Status | [Excluded, "Not excluded"] |
| **Derived Sorts** | |  | |
|  | Company: | [(MedicalSupplies SE, Spain), (MyCorp GmbH, Germany)] |
|  | Partner Status: | [((MedicalSupplies SE, Spain), not excluded),  ((MyCorp GmbH, Germany), not excluded)] |
|  | Receiver: | ((MyCorp GmbH, Germany), DE314159265) |
|  | Order Positions: | [[["cough syrup"]] |
|  | Order: | [((MedicalSupplies SE, Spain), ((MyCorp GmbH, Germany), DE314159265), ["cough syrup"])] |
| **Functions** | | | |
|  | cat: Product -> Product category | cat(cough syrup) = medicine |

Table 2: Initialised Signature

Contrary to classical logs, there is no total order in the events. Usually, a timestamp is used to create a total order of the events. In this case, the order only defined through the casual relationship of events. In Figure 9 it can be seen, that the individual verification steps (turquoise, dark cyan, olive, purple) are independent and concurrent, whereas all verifications need to finish before the final result can be produced (brown events). To get a more detailed look at the event inscriptions, Figure 10 depicts an exerpt of the distrubuted run, containing the verification steps that relate to the product itself.

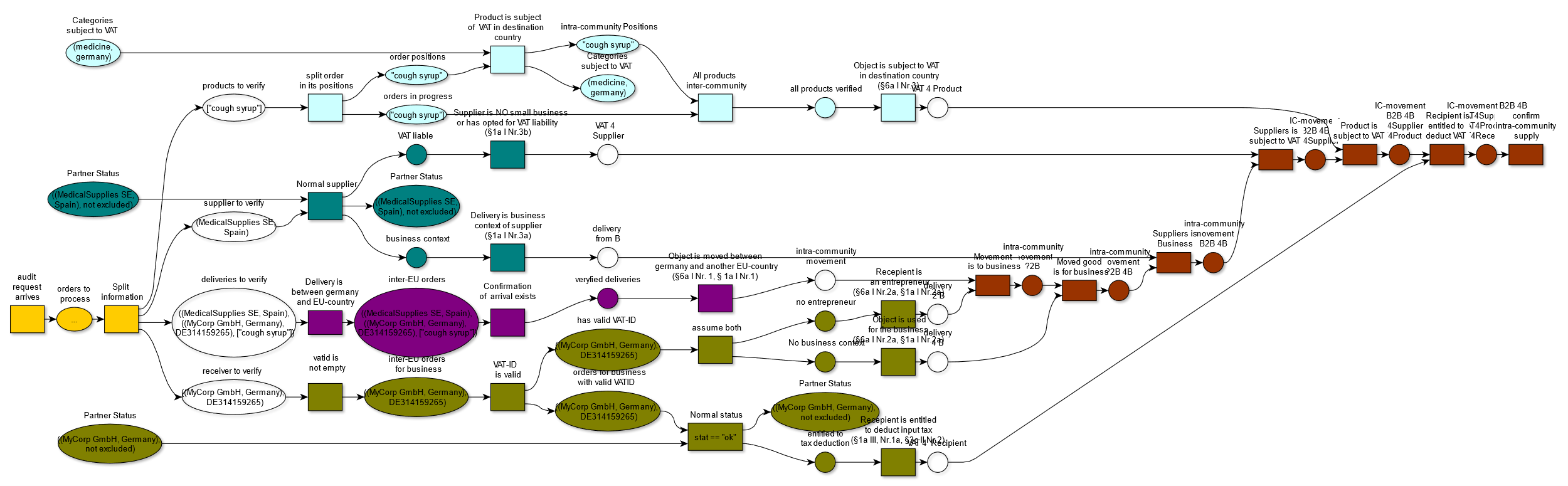


Figure 9: Distributed run as event log[[7]](#footnote-8)

For example, the event "Object is subject to VAT in the destination country (§6a I Nr.3)" shows that the bot verified the specified norm. By traversing the distributed run to the left, the reason for this decision becomes evident: The product in question (cough sirup) belongs to the product category medicine and is therefore subject to VAT.

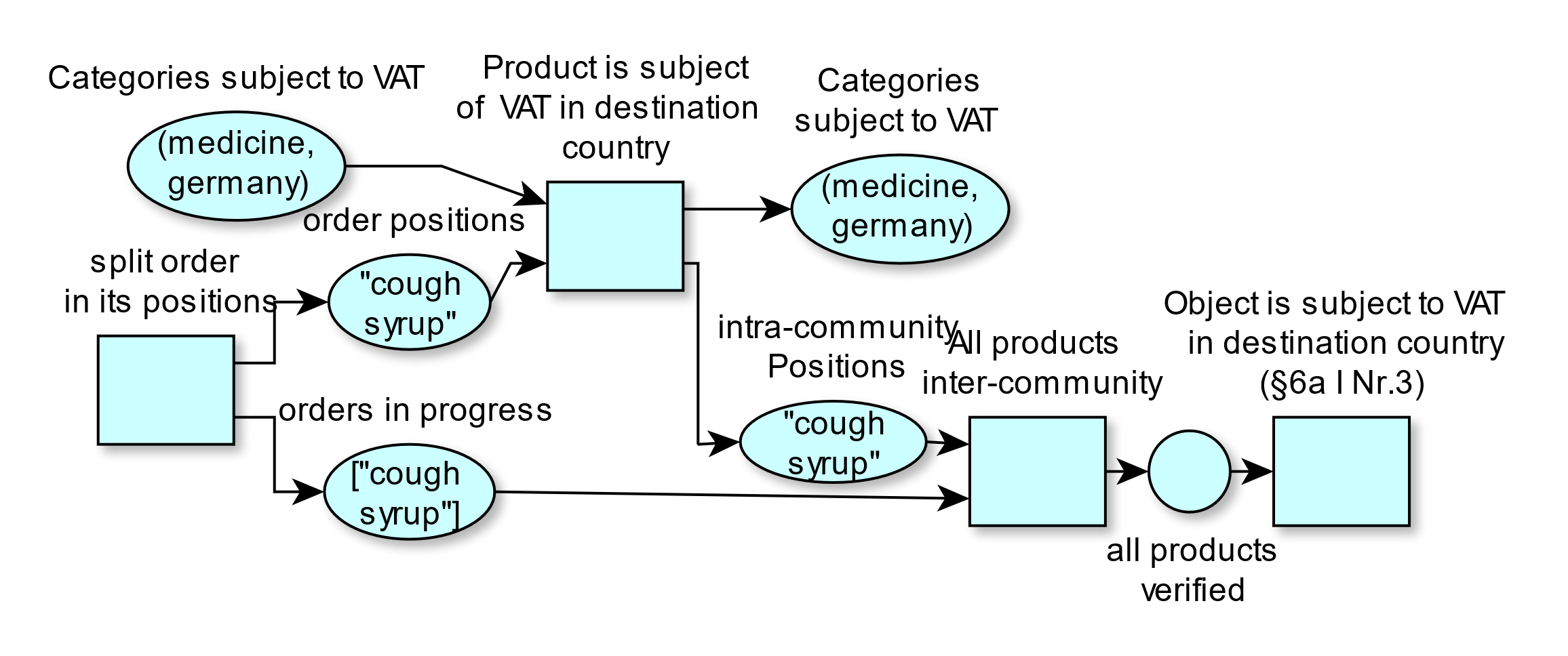


Figure 10: Exempt of distributed run

In conclusion, Heraklit is capable of modelling the RPA system and documenting the execution of processes. The acyclic graph allows extracting causal relationships between events and data objects, thereby enhancing the explainability and understandability of RPA-Systems for business users and legal experts.

## Process automation through multi-agent RPA

The model in section 4.1 describes the distributed process of intra-community supplies. Although the dynamics and the involved data objects are defined, the model is technology agnostic. The following section will describe the transfer of this model onto specific multi-agent RPA implementations.

Research has shown that the division of larger process automation into smaller fragments reduces maintenance costs and improves the whole system's understandability. This is also reflected in the Heraklit model. For an initial understanding of the assessment process, only the modules and their interfaces are utilised. The inner logic of each process is abstracted. This representation can be directly transferred to the RPA implementation. Each module can be implemented as a separate RPA-Agent, and the interfaces depict the communication channels and the expected information exchange.

A principal design question when constructing modular RPA automation is whether to connect bots via choreography or orchestration. In an orchestrated execution, a central party like a process engine or a workflow management system communicates with each bot individually and controls their collaboration. (Mendling & Michael, 2008) Workflow engines are centralised services that were designed to orchestrate processes within an organisation. The idea to control process step executions in multiple systems through one centralised system suffers from the following problems: A central workflow engine is a Single Point of Failure (SPOF) and hard to scale to large, distributed enterprise settings. In Addition, they require a single agreed-upon process model. Although this might be possible in intra-enterprise processes, this becomes highly unlikely in inter-enterprise settings. This is especially true for value-added tax assessments since a single invoice might only involve two entrepreneurs. Still, the company assesses massive amounts of invoices with numerous suppliers and clients.

A choreographic approach, on the other hand, imposes fewer restrictions on the stakeholders. The stakeholders instead agree upon a model describing how information is exchanged. Each stakeholder can independently decide on the internal architecture of the services he contributes as long as they implement the underlying choreography model. (Mendling & Michael, 2008) This approach is more modular and robust to changes.

The Heraklit Modell imposes a choreographed solution where the gates at each module interface represent the information channels. Each module can be implemented as a separate RPA-Bot, and the interface defines the required communication with other parts of the system. Furthermore, this architecture can be scaled to other tax-related automation without the need for one unified model. Therefore, Aalst describes RPA as "BPMS for the poor" since it is a loosely coupled and cheap connection of IT systems with a business process perspective. (van der Aalst W. M., 2021)

# Discussion

This case study showed the ability of Heraklit to describe the logic of an intra-community verification process. We were able to model the status quo of the legal experts and the automated process with RPA-bots. Through modelling the verification process with legal experts, the complexity of taxation automation became visible. Even though multiple simplifying assumptions were made, the model complexity had to be reduced through modularisation.

The contribution focuses on the modelling perspective of complex interdependent systems of bots to automate entire workflows and legal decision making. The implementation is not in the scope of this contribution.

Splitting the verification process into smaller, independent bots not only enhances the understandability but also leverages the benefits of concurrent computing and modularisation:

* The bots are easier to maintain since the complexity of one bot is reduced.
* For semi-automated processes, modularisation allows for more elaborate tasks to be left to human workers.
* Hardware provisioning can be adapted more adequately to the requirements of each bot. For example, neural network-based optical character recognition in a document might need GPU acceleration, whereas classical front-end automation performance economic.

In this context, it should be mentioned that a sequential control flow that breaks when encountering KO-criteria might be more resource-efficient in obtaining a result since the execution would stop at an earlier point in the verification process. On the other hand, a concurrent solution ensures that each verification task is completed and that the resulting documentation is always comprehensive.

Moreover, the graph-based documentation of the process events enables business users, tax office officials and legal experts to extract causal relationships. Although, the distributed form of legal assessment might seem unusual to legal experts. This is primarily due to the fact that a verification assignment would not be split into such small tasks. The communication overhead to pass a minimal verification task to a co-worker would exceed the time to perform the actual verification. For example, the time requirements for writing an email with the assignment: Please check the following VAT-ID is equivalent to entering the VAT-ID into the form of the Federal Central Tax Office. This does not apply to RPA-Bots communicating with each other.

Furthermore, we believe the presentation of RPA-execution logs as distributed runs which are based on the process model developed by business users, dramatically enhances the understandability of such systems for non-technical employees.

By using individual transitions for each verification step, we were able to link individual bot activities to their corresponding legal norm. This is another enhancement of the documentation since each activity is linked to its semantics. Through this link, tax office officials are enabled to validate the semantic equivalency between the bot actions and the intended meanings of the legal norm.

Unfortunately, to create an RPA implementation that fully automates the verification process, further aspects of the model need to be specified. As mentioned earlier, the validation of the confirmation of arrival should be modelled in itself, and the maintenance of the partner status should be considered for automation. Through the modularisation capabilities of Heraklit, these models can be integrated seamlessly into the existing work. The modularisation furthermore allows to switch individual components: the module "product verification", for example, can have different internal dynamics depending on the needs of the company. If multiple reference model alternatives are presented by academia, the company only needs to insert the module with congruent premises.

# Outlook

At first sight, verifying intra-Community supplies constitutes a perfect candidate for automation: high volume throughput, standardised decision rules, extensive documentation requirements and millions of European enterprises as potential beneficiaries.

The difficulties arise in the assessment of each individual case. Although the general decision rules are specified only by a handful of norms, they are refined and detailed in a wide range of tax directives, implementation ordinances and regional court decisions. This requires the automation model to differentiate between a multitude of cases. In addition, legal norms require a factual assessment which means a legal expert decides if a rather open formulation is applicable in a complex situation. This requires context information and experience which are hard to leverage in current algorithmic implementations. Automations are at their best when an unambiguous measurand is constrained using a fixed domain of values. Therefore, in our opinion, the task of auditors will predominantly be to define heuristics that map an unambiguous measurand to an acceptable range of values. These heuristics must be defined in such a way that their results are congruent with the auditors' common assessments in many circumstances.

The difficulties in automating the entire workflow shown throughout this case study support the effectiveness of Heraklit as an intermediary modelling language between subject-matter experts and explicit implementations. By conceptualising the architecture, the statics, and the dynamics of an RPA solution, we uncovered unsolved questions that need to be addressed prior to the execution phase of an RPA project. Additional research could build upon this insight and examine further automation cases in public administrations.

In the next step, we plan to extend our model to cover a wider range of cases and implement a prototype based on this reference model. The open research questions are whether the Heraklit model is expressive enough to create an RPA implementation based on it and if the distributed run serves as an enhanced and easily comprehensible execution log.

# References

Aguirre, S., & Rodriguez, A. (2017). Automation of a business process using robotic process automation (RPA): A case study. *Workshop on engineering applications* (S. 65-71). Springer.

Amand, C. (2016). The Impossible Proof of Intra-Community Supplies of Goods. *International VAT Monitor*, S. 98.

Bal, A. (15. April 2019). Indirect Tax Compliance: Process Automation or Process Transformation? *Tax notes international*, S. 255-258.

Cooper, L. A., Holderness, D., Sorensen, T. L., & Wood, D. A. (2019). Robotic Process Automation in Public Accounting. *Accounting Horizons*, S. 15-35.

Czarnecki, C., & Fettke, P. (2021). Robotic Process Automation. Positioning, Structuring, and Framing the Work. In C. Czarnecki, & P. Fettke, *Robotic Process Automation. Management, Technology, Applications* (S. 3-24). De Gruyter.

Dietsch, D. R. (2018). Umsatzsteuer 4.0 - wie Blockchain grenzüberschreitende Reihengeschäfte transparenter machen könnte. *MwStR*, S. 813.

EU-commission. (2017). *, Towards a single EU VAT area - Time to act. Amended proposal for a Council Regulation amending Regulation (EU) No 904/2010 as regards measures to strengthen administrative cooperation in the field of value added tax, COM 2017 706 final.* Brussels.

Fettke, P., & Reisig, W. (2020). *Heraklit – die erkenntnistheoretisch motivierte Modellierung rechnerintegrierter Systeme.*

Fettke, P., & Risse, R. (27. July 2018). Blockchain: Wird eine sog. "real time" Tax Compliance möglich? *Der Betrieb*, S. 1748-1755.

Houy, C., Hamberg, M., & Fettke, P. (2019). Robotic Process Automation in Public Administrations. *Digitalisierung von Staat und Verwaltung. Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI)* (S. 62-74). Münster: Gesellschaft für Informatik (GI).

Ismer, & Schwarz. (2019). Combating VAT Fraud through Digital Technologies: A Reform Proposal. *IVM*, S. 240.

Langhein, N. (2017). *Umsatzsteuer und Steuerplanung.* Wiesbaden: Springer Fachmedien.

Mendling, J., & Michael, H. (2008). From WS‐CDL choreography to BPEL process orchestration. *Journal of Enterprise Information Management*.

Mezzio, S., Stein, R., & Stein, S. (2019). Robotic process automation for tax. *Journal of Accountancy*, S. 61-63.

Reisig, W. (2013). *Petri Nets. Modeling Techniques, Analysis Methods, Case Studies.* Berlin Heidelberg: Springer.

Romao, M., Costa, J., & Costa, C. J. (2019). Robotic process automation: A case study in the banking industry. *2019 14th Iberian Conference on information systems and technologies (CISTI)* (S. 1-6). IEEE.

Sala, M. (2020). A road map for tax function automation. *Tax Adviser*, S. 1-6.

van der Aalst, W. M. (2021). Process Mining and RPA. In C. Czarnecki, & P. Fettke, *Robotic Process Automation - Management, Technology, Applications* (S. 223-239). Berlin: De Gruyter.

van der Aalst, W., Bichler, M., & Heinzl, A. (2018). Robotic Process Automation. *Business & Information Systems Engineering*, S. 269-272.

van Vliet, H. (2008). *Software Engineering: Principles and Practice.* Hoboken: Wiley & Sons.

1. For additional details, we refer to http://www.heraklit.org/ [↑](#footnote-ref-2)
2. Council Directive 2006/112/EC of 28 November 2006 on the common system of value added tax, OJ L 347/1. [↑](#footnote-ref-3)
3. In this section, we will use the term intra community supplies for acquisitions as well as deliveries. [↑](#footnote-ref-4)
4. Art. 43 Para. 2 MWSt SysRl [↑](#footnote-ref-5)
5. <https://evatr.bff-online.de/eVatR/index_html> [↑](#footnote-ref-6)
6. Value Added Tax Act (Umsatzsteuergesetz) in the version of the notice of 21. February 2005 (BGBl. I S. 386), which was last changed through Article 3 of the Act of 10. March 2021 (BGBl. I S. 330). [↑](#footnote-ref-7)
7. Inscriptions are not readable and only added for completeness. The general structure is of interest. [↑](#footnote-ref-8)