Individual Service Clearing as a Business Service: A Capable Reference Solution for B2B Mobility Marketplaces

Michael Strasser[1], Nico Weiner[1] and Sahin Albayrak[2]
[1]*Bosch Software Innovations GmbH, Schöneberger Ufer 89, 10785, Berlin, Germany*
[2]*DAI Laboratory, Technical University of Berlin, Ernst-Reuter-Platz 7, 10587, Berlin, Germany*

Keywords: Individual Service Clearing, Service Marketplaces, Service Platforms, Service Interface Design, Smart City Mobility Services.

Abstract: The paper presents an approach for individual and outsourced service clearing within Business-to-Business marketplaces for mobility services in a Software as a Service fashion. The lack of service clearing possibilities within today's marketplaces solutions for mobility services have been confirmed by interviews experts who highlight the need for clearing. To enable service clearing, appropriate interfaces which enable access upon data are required. Current solutions lack those interfaces and thus service clearing to charge service transactions is not possible. The paper discusses interfaces for outsourced service clearing according to their design, data and role dependencies. A reference solution is implemented to demonstrate the feasibility of the interfaces and the overall clearing approach. A clearing algorithm has been developed to validate the interfaces' reliability and correctness. Our presented clearing approach enables marketplace participants to outsource the transaction clearing to a provider which offers clearing capabilities. The work on hand contributes to interface and protocol standardization in respect to service clearing and marketplace interconnectivity.

1 PROBLEM DESCRIPTION

The number of service consumers and providers for mobility services (e.g. car sharing, charging, parking, routing) increases constantly. It is anticipated that this trend will probably result in a higher number of service offerings, demands and transactions (Thitimajshima et al., 2015). As a logical consequence, more business relationships will be established via electronic service marketplaces. These marketplaces represent the environment in which participants can accomplish service trades. These marketplaces are pure B2B oriented and do not deal with end-customers. End-customers are registered with service consumers. The increasing demand for convenient mobility will subsequently increases the usage mobility services. That circumstance leads to more transactions which are processed by a service operator and probably routed by a marketplace. One of the service operator's duties is to have capabilities to uniquely assign a single transaction to one particular contract (Pfeiffer and Bach, 2014). This is necessary to assure that a service consumption (transaction) is cleared (charged) according to the conditions defined in the business contract.

It is likely that the growth of service consumptions results in a higher effort for service clearing (transaction billing). Therefore it is feasible to assume that certain service operator want to outsource the complete service clearing process to a third party. This party has to have comprehensive knowledge and approved and standardized processes in the service clearing domain. For individual service clearing, a respective clearing operator has to register with a marketplace and offer clearing services to other participants as software as a service (SaaS). Such a clearing operator is called service consumption clearing operator (SCCO). A service operator, for instance for charging stations, who want to outsource the service clearing process can contract a SCCO's clearing service. Once they are business partners, the charging station operator can assign the SCCO's clearing-service to all his services which he wants to be cleared (charged) by the SCCO. Therefore, a service marketplace for mobility services has to provide mechanisms to ensure accurate data access on internally stored data. Interviewed experts[1] point out that individual

[1]The interviewees' affiliation are presented in the acknowledgment section.

Strasser, M., Weiner, N. and Albayrak, S.
Individual Service Clearing as a Business Service: A Capable Reference Solution for B2B Mobility Marketplaces.
In *Proceedings of the 6th International Conference on Cloud Computing and Services Science (CLOSER 2016) - Volume 2*, pages 27-38
ISBN: 978-989-758-182-3

service clearing does not exist in electronic marketplaces for mobility services but is required. Individual in this context means that a marketplace participant can do the clearing on his own or contract and entitle a professional SCCO. However, due to missing Application Programming Interfaces (API)s, outsourced service clearing is one of many use cases which cannot be accomplished because access to data stored within the marketplace is not possible. As discussed by Gubbi et al., a service infrastructure has to provide appropriate APIs which can be used to enable flexible billing (Gubbi et al., 2013).

The rest of the paper is organized as follows. Section 2 presents the current state of the art in B2B service clearing in marketplaces for mobility services. Section 3 introduces the interfaces which are deemed as necessary to enable communication with the marketplace. Section 4 discusses the elaborated approach for individual service clearing. The evaluation of the interfaces and the elaborated and prototypically implemented service clearing algorithm are presented in Section 5. The paper ends with a conclusion in Section 6 and a short outlook on further research in Section 7.

2 CURRENT STATE OF THE ART

Research projects[2] explore possibilities about how to provide services for the mobility domain, like charging services for electric vehicles, sharing services for bike or car as well as services which help to find free parking lots. A system which provides access to such services is called for example *e-hub* (Grieger, 2003), *service provisioning platform*, (Å gerfalk et al., 2006), *Value Added Service Supplier* (Legner, 2007) *intermediary* (Heinrich et al., 2011), *Platform* (Buchinger et al., 2013) or *electronic marketplace*, *trading communities*, *trading exchanges* (Turban et al., 2015). Companies[3] also provide Information and Communication Technology (ICT) based solutions over which mobility service operators can offer and mobility service consumers can access mobility services. These mobility services enable end-customers of service consumers to experience the capabilities of future mobility. An electronic marketplace is the environment for service operators (offer services) and service consumers (consume services) to trade services like goods (Ghenniwa et al., 2005). An electronic marketplace matches sellers and buyer and encourage them for service trades by providing appropriate trading capabilities (Turban et al., 2015).

Current marketplaces for electric mobility services lack the possibility to enable data access on internally processed and data from outside. This is a limitation and prevents further use cases and the realization of *Critical Success Factors* (CSF). Interviewed experts have emphasized that it is mandatory to access transaction data, contract data, participant information or service descriptions via APIs. By the current date, neither an appropriate protocol nor respective interface standards are available to guide the development of respective interfaces regarding comprehensive communication with the outside world. Without comprehensive data access, internal data is locked inside a marketplaces and cannot be processed further by any participant expect the marketplace itself.

Costs accrue for being a member of a marketplace (depends on a marketplaces' revenue model (Johnson, 2013)) but also for using a service provided via a marketplace (interviewed experts). An identified business case (also indicated by interviewed experts and literature) is business to business (B2B) clearing also known as service charging. Clearing is considered as a CSF of electronic marketplaces (Balocco et al., 2010; Johnson, 2013). The experts point out that service provisioning marketplaces have currently no elaborated clearing approach or completely lack the possibility to enable service clearing within their system. Pfeiffer and Bach point out that a business model for clearing does currently not exists and that "the billing process is more cost-intensive than the overall billing amount" (Pfeiffer and Bach, 2014). These are the reasons why it is currently not operated within marketplaces or platforms. However, clearing mechanisms have to be in place as transactions are not free of charge (Rust and Zahorik, 1993; Buyya et al., 2008; Johnson, 2013; Thitimajshima et al., 2015). Even though no appropriate business models for transaction clearing exists, the rise of new demands or the development of new capabilities might enable cross selling or combining of capabilities which justifies certain clearing models. A clearing service is deemed as a value added service in within electronic commerce conducted via a marketplace (Turban et al., 2015).

Marketplaces for electric charging infrastructure like Green eMotion and Smartlab's e-clearing.net perform do B2B service clearing for their participants right within their systems. There is no choice for their participants to appoint another clearing operator.

[2]For example: Green eMotion (Green-eMotion, 2015), CROME (CROME, 2014), Olympus (Olympus, 2015), Streetlife (Streetlife, 2015)

[3]For example: Hubject (Hubject, 2015), Parku (Parku, 2015), Multicity (Multicity, 2015), Smartlab (smartlab Innovationsgesellschaft mbh,)

CROME or Hubject do not support service clearing inside their platform and leave it to their participants who have to agree on a common clearing method. The limited service clearing mechanism is a disadvantage of all of today's platforms and marketplaces for mobility services. Current solutions do not provide appropriate interfaces that enable access to internally stored data or do not even store data accordingly (interviewed experts). This cripples the possibility for individual service clearing where participants outsource the clearing process and appoint an individual clearing service operator which meets their requirements. The state of the art in transaction clearing in marketplaces for mobility service is in summary that either the marketplace itself seizes service clearing or participants have to do it somehow outside the marketplace. Marketplaces like Amazon or iTunes as well as eBay and other known marketplaces or trading platforms offer a handful payment options to their participants. Individual service clearing via SCCOs and internal B2B transaction clearing is not foreseen in their solutions.

The paper on hand is a comprehensive expansion of (Strasser et al., 2016). Its contributions are presented in List 1 below.

- A comprehensive outline of the problem domain and the current state of the art
- An extended overview of required interfaces for future marketplaces along with an simplified architecture
- An expanded presentation of the defined clearing process
- A deeper discussion of the elaborated clearing interfaces and their implementation
- An extensive presentation of the developed service clearing protocol and its parameters
- An open demonstration of the elaborated clearing algorithm
- An performance evaluation of the clearing algorithm

List 1: Contribution of the paper.

3 MARKETPLACE INTERFACES

Marketplaces have to provide appropriate APIs to enable access on internal data which is fundamental for outsourced and individual service clearing. A comprehensive set of marketplace interfaces enables the accomplishment of various additional use cases, for example interconnectivity between service marketplaces (Strasser and Albayrak, 2015; Strasser and Albayrak, 2016) or Service Level Agreement (SLA) and condition monitoring. The marketplace developed within this work consists of Service Oriented Architecture (SOA). The marketplace's communication interfaces use the Simple Object Access Protocol (SOAP). The interfaces are developed in a role-based manner instead of a data-driven manner. The interfaces are furthermore differentiated between public and private. Context, service or participant relevant data can only be accessed via private interfaces. This kind of data might be classified as confidential and only accessible for registered marketplace participants. General information about a marketplace and its participants, services, supported domains or operation area can be retrieved via public interfaces which do not require a marketplace participation. List 2 presents interfaces which have been identified to be of public interest. Thus they are accessible without any marketplace registration. Registered participants can use those interfaces too.

- Public interfaces:
  - *GetMarketplaceInformation*
  - GetParticipantOverview
  - GetServiceOverview
  - *GetMarketplaceConditions*
  - SetRegisterParticipant

List 2: Public marketplace interfaces.

List 3 provides an abstract of private interfaces which have been identified to be of private interest. They provide access to confidential and participant dependent information and thus can only be accessed by registered participants. These interfaces are assumed as necessary to enable the communication with other service marketplaces as well as to enable individual and outsourced service clearing. Of course, respective processes have to be implemented too. The interfaces for the outsourced service-clearing are emphasized in bold. The interfaces for the marketplace interconnectivity are emphasized in italic. The interfaces are named according their functionality and purpose.

Set and *Get* indicate whether information can be retrieved (pull) from or send (push) to a marketplace. The interfaces' grouping towards public or private is not final. It might change once more use cases become reasonable. Interfaces which are listed in both lists provide more information if they are applied in private mode. Due to security mechanisms, a API requester is validated if it is a registered marketplace participant. This is done by a certificate. Then it is

- Private interfaces:
  - GetParticipantOverview
  - GetServiceOverview
  - GetParticipantInformation
  - GetServiceInformation
  - SetCreateServiceOfferQuotation
  - SetCreateServiceSearchQuotation
  - GetServiceOfServiceType
  - **GetTransactionData**
  - **GetClearingRelatedContracts**
  - **GetContractData**
  - *SetUploadMarketplaceData*
  - *SetUploadParticipantData*
  - *SetCloseContract*
  - *SetUploadAllAvailableServices*
  - *GetEndCustomerAuthorization*
  - *GetMarketplaceConditions*
  - *GetSearchServiceQuotation*
  - *SetAcceptServiceQuotationCondition*

List 3: Private marketplace interfaces.

checked what role the requester has to decide what data can be accessed.

The interfaces are role-base driven rather than data driven. That enables re-using and re-purposing of interfaces while being able to limit the data access accordingly. For security reasons, private and public interfaces are separate web-services. This is done to relieve each interface and because the interfaces can have different service level agreements (SLA)s. Each web-service consists of various operations which represent one of the above identified interfaces. Extensible markup language (XML) schema validation is used to validate the input data. This ensures data type integrity and avoids malfunction. The private web-service validates user-credentials and certificates. Figure 1 presents a simplified overview about how marketplace data is accessed from external.

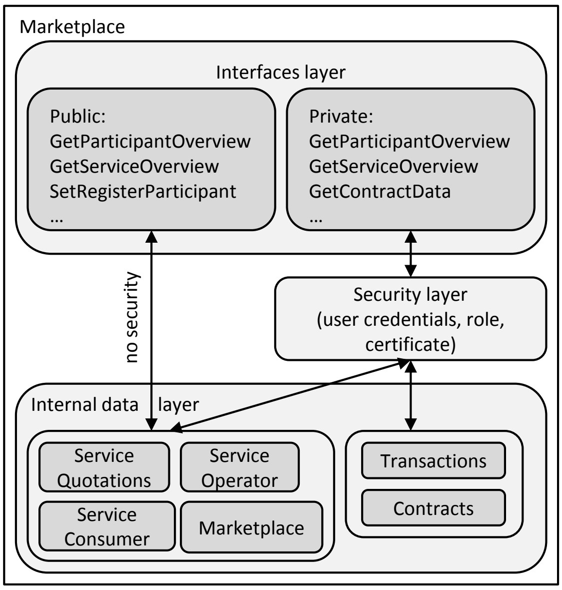

Figure 1: Interfaces to access internal marketplace data.

Security mechanisms have to be in place to validate requests and to determine which data the requester is allowed to access. Even though the security layer can also be applied for public interfaces, it is assumed that security checks are costly and thus should only be conducted when necessary.

4 ACCESSING AND PROCESSING INTERNAL MARKETPLACE DATA

The proposed marketplace interfaces facilitate the realization of various use cases. Request routing, performance and efficiency monitoring, Service Level Agreement (SLA) validation or any other data processing or measuring use cases can be developed. However, the interfaces also enable individual service clearing for B2B service transactions. Clearing services require access at internal marketplace data from outside. The elaborated marketplace provides those interfaces and processes which are necessary to access all required data to create an invoice for service consumption. Service-clearing particularly requires access to contract and transaction data.

4.1 Service Consumption Data

The following mobility use case exposes how transaction data incurs within a mobility service marketplace. A service operator (SO) has parking infrastructure. SO is registered with a service marketplace (MP) and offers a service via the MP which enables to access his infrastructure. The service's functionalities are for example i) to open a barrier arm, ii) to guide a driver to an empty parking spot or iii) to determine the duration of the parking. Service consumer (SC) is also registered with MP. SC searches for a service with capabilities to access parking infrastructure. Therefore, SC i) contacts SO, ii) negotiates the service's conditions and iii) settles a contract with SO. A digital representation of the paper contract is stored inside MP. The end-customer (EC) of SC cannot, at first, access the parking infrastructure of SO because

SO does not know EC. For SO is EC a stranger and thus forwards the request to MP. MP checks all of SO's contracts and forwards the service request to all of SO's business partners. SC retrieves the service request and recognizes EC. SC responses accordingly and MP forwards the response to SO. SO opens his parking infrastructure for EC because SC has acted as EC's guarantor. As soon EC leaves the parking area, SO sends a *park detail record*, which contains information about the parking process, to MP which forwards the record to SC.

4.2 Individual Service Clearing - An Outsourced Invoice Approach

In the current context, service clearing is done in respect to B2B service consumption among marketplace business partners (service operator and consumer). Outsourcing the service clearing process to a third party calls Luttge *outsourced invoice* (Luttge, 2001). This approach is, according to the interviewed experts, not implemented by any mobility service marketplace. However, the experts assume it as a fundamental functionality and according to Turban et al., a trading platform should provide mechanisms to arrange payment (Turban et al., 2015). Even though Vidal et al. (Vidal et al., 2011) describe in detail what kind of data has to be in place to clear an electric vehicle charging process via a service marketplaces, they miss the chance to implement and present a working solution.

To outsource the service clearing process to a SCCO a service operator must comply to the following process: A SCCO-A offers a clearing service via a marketplace (MP). Another service operator (SO-B), who offers parking infrastructure services, closes a contract with SCCO-A. Then, SO-B assigns the contracted clearing service of SCCO-A to his service XY. This entitles SCCO-A to access those contracts and transactions which relate to SO-B's XY service. The service description of XY should particularly emphasize that a third party (SCCO-A) has been entitled to accesses all accrued transactions and the contracts related to service XY. A service consumer (SC) who signs a contract with SO-B for service XY agrees on the service's conditions and thus accepts that SCCO-A accesses all clearing relevant data. This agreement is important because the transaction data represents a kind of an end-customer's mobility profile. Thus SC has to inform his end-customers about the third party access. SCCO-A gathers the necessary data according to the defined clearing schedule and calculates the price which SC has to pay SO-B. Due to the individually, SO-B can appoint another SCCO to his other services but also can do the clearing on his own without third party support. This functionality is not in place by current available solutions but is prototypically implemented within this research.

The role-heritage diagram in Figure 2 presents the roles used in the use-case diagram presented in Figure 3. The role *Clearing Service Provider* is not a individual role in particular but represents a possible character of a service provider. The generic use-case diagram illustrates the roles, systems and processes which are part of a individual and outsourced clearing process. The elaborated Use-Case diagram is neither an extension nor a modification of (Pfeiffer and Bach, 2014) e-roaming clearinghouse. It presents a succeeding step in which e-roaming is actually charged according to negotiated conditions written in a contract during, what (Pfeiffer and Bach, 2014) calls, settlement.

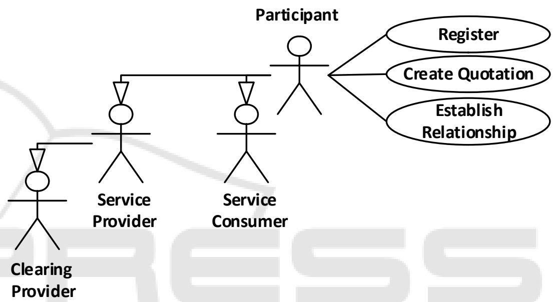

Figure 2: Overview role inheritance.

To achieve individual and outsourced service clearing various logical and functional challenges have to be approached. An abstract of identified challenges is shown in List 4. The list does not claim to be complete. Once our proposed solution is implemented in commercial solutions it is believed that further challenges will be identified. The challenges furthermore depend on the design of the service clearing as well as on the implemented contracting mechanism.

Legal regulations are not part of the research's scope. The interviewed experts confirm the complexity of service clearing but unfortunately do not propose a suitable solution. No standardized protocol or data set exists on which data exchange via marketplace APIs can be build on. The relevant domain protocols have capabilities to support the marketplace in its operation only. Marketplace participants are able provide to information to the platform but cannot retrieve necessary information about contracts and transactions or other data useful for future business cases. Access on data is only possible for data which has been provided by the participants before. No guidelines are available which emphasize the required steps to enable data access from outside and,

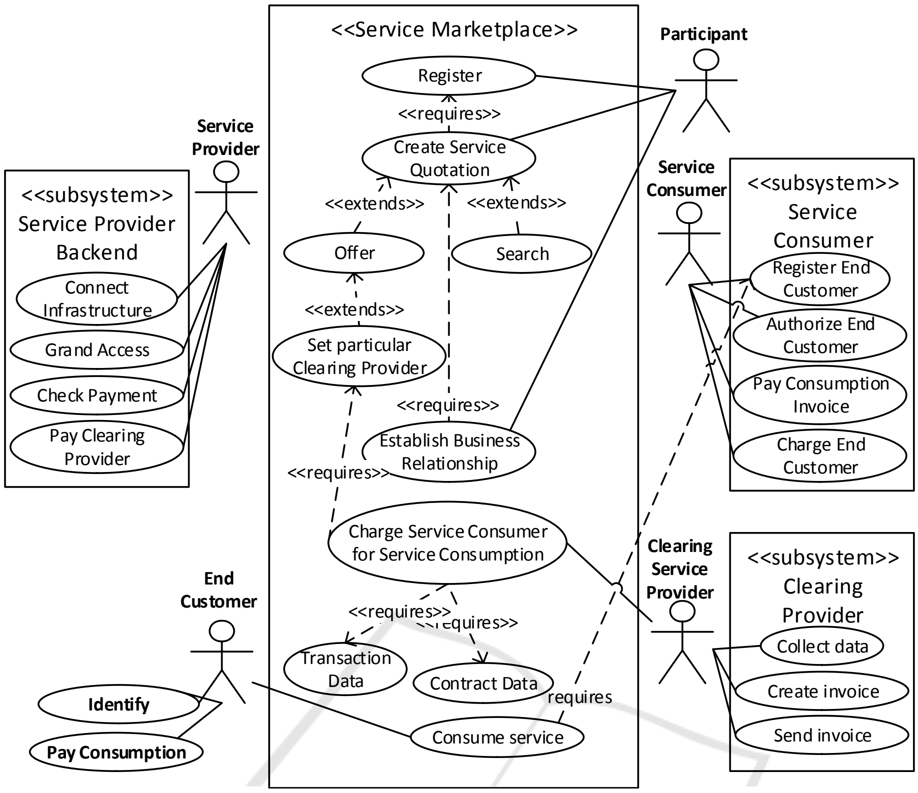

Figure 3: Use-Case diagram for individual service clearing.

- Authorize access at other participants' data
- Control data access and check if it is valid
- Avoid multiple data access by multiple SCCOs
- Identify which data is allowed to be accessed
- Notify service consumers about SCCOs data access
- Define process if a service's SCCO has changed
- Validate how often data is accessed
- Determine processes and tasks to close a clearing contract
- Avoid that a SCCO outsources his clearing duty to other SCCOs
- Evaluate the robustness of the interfaces
- Evaluate whether interfaces can be used for cross-marketplace service clearing

List 4: Challenges according to service clearing.

even more important, no research was found which identified what interface functionalities are required by mobility marketplace participants. Therefore, this work provides insights into the required steps, tasks and data for doing so while identifying the need for individual and outsourced service clearing.

4.3 Service Clearing Interfaces

Table 1 presents the elaborated interfaces required for outsourced and individual service clearing. These interfaces are used by a SCCO to access clearing relevant data. The clearing functionality has been realized with several light wise interfaces to achieve a better performance, to reduce maintenance work and to enable interface re-usability.

It is pointed out that marketplace interfaces should not, if applicable, serve one particular use case only. Use case relevant interfaces process and return data for a particular use case, thus they are probably not applicable for other use cases. The developed marketplace and its interfaces are use case independent and role-based driven. Thus participants and there roles are differentiated and the data returned respectively. The interfaces in Table 1 are examples of role-based driven interfaces because they return data according to the provided certificate and identifiers. The first can only be used, at the moment, by a SCCO while the remaining two can be used by all registered marketplace participants for informative reasons. A role validation is necessary to decide which data has to be

Table 1: Interfaces for service-clearing.

Interface Name	Description
GetClearingRel - atedContracts	Returns a list of contracts in which the requester is set as SCCO
GetContractData	Returns contract details like participant contact data and usage price plan
GetTransaction - Data	Returns transaction data related to the contract and the SCCO's client

delivered to the requester.

Due to the interfaces' use case independence implementation, they can be used individually or in combination. Individual usage would be, for example, to check transactions or to check when a specific contract expires. Combined usage is about using an interface's output for another interface's input. This is done for service clearing and presented in Figure 4. At first, a SCCO checks his contracts to identify his clients by using the *GetClearingRelatedContracts* interface. The contracts define the services for which the SCCO is entitled to do service clearing. Once SCCO knows which services are under his clearing responsibility, SCCO uses the *GetContractData* interface to retrieve the contracts which have been signed between his clients and their business partners. With those contracts SCCO knows the service consumption conditions (price plan) and other contact data. The *GetTransactionData* interface is used to retrieve all transaction data related to a specific service. Once all data is gathered a SCCO computes the costs of all retrieved transactions according to the contract and the specified price plan. A potential capability might be that SCCO sends the final invoice to his clients' business partners on behalf of his client. As previously outlined, a service operator is able to entitle different clearing operators to do the service clearing for different services but only one SCCO per service at the same time. From a SCCO point of view it is relevant which interface is used first. The presented input and output parameters in Figure 4 are a summary of the most important parameters. Each interface requires a minimal set of input information. To avoid the accidentally disclose of data, the interface returns as little information as necessary in accordance to the requesters role and the role's scope. This procedure contributes to the assurance of privacy and security. The marketplaces' interfaces are developed to actively contribute to protocol standardization for marketplace in the mobility domain and therefore the protocols parameters are presented. The lessons learned while develop the presented processes and interfaces can be used for the implementation of additional interfaces necessary for the use case introduced in the beginning. This is also a reason why independent but role-based driven interfaces are required.

Figure 5 presents a Business Process Diagram (BPD) which demonstrates the access via the *GetTransactionData* interface with all validations. If the input parameters pass all validations then the transaction data is returned.

4.4 Interface Implementation

The interfaces have been implemented using *inubit*, a Java and XML based Business Process Modeling application. The process engine runs on Apache Tomcat. The *AccountId* is the most general identifier and is used to identify a requester. The *ContractId* is indispensable to determine a contract and subsequently retrieve the contract details. The prototypical implementation of the introduced clearing interfaces has proven that these two identifiers are sufficient to determine all account, contract and transaction data. Special roles like SCCO have separate identifiers to avoid confusion when using the interfaces. The *GetClearingRelatedContracts* is an interface that specifically requires a *SCCOID* to uniquely identify a clearing operator rather than a general *AccountId*. The *SCCOID* replaces the *AccountId*. All interfaces check whether an *AccountId* fits to the given *ContractId* or not. This is done to determine a requester's access scope. For example, using the *GetTransactionData* interface with the same *ContractId* but different *AccountId*s might returns with i) a different data set or ii) with a subset of the former return data or iii) with the very same data. The reason is that the given *AccountId* is a) set in the contract as business partner or b) is set as a clearing operator which is allowed to access this particular data. The interfaces need to be designed to detect and differentiate all possible scenarios and respond accordingly. This is also presented in Figure 5.

Figure 6 on the next page shows the *GetContractData* response. It contains the *ContractId* and all contract details. The contract details depict the business partners. The service operator is shown in *ProviderAddress* and the respective service consumer in *ConsumerAddress*. The service operator entitled the SCCO to do the service-clearing on its behalf. The price conditions are necessary to charge each transaction individually in accordance to the defined usage price plan. A operator of a service can define a basic price for each service transaction. However, it is also possible that special time slots are defined. The

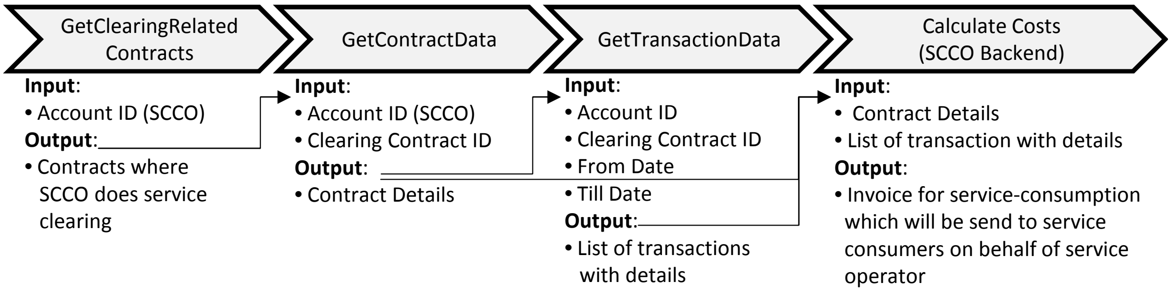

Figure 4: Interface sequence for service clearing.

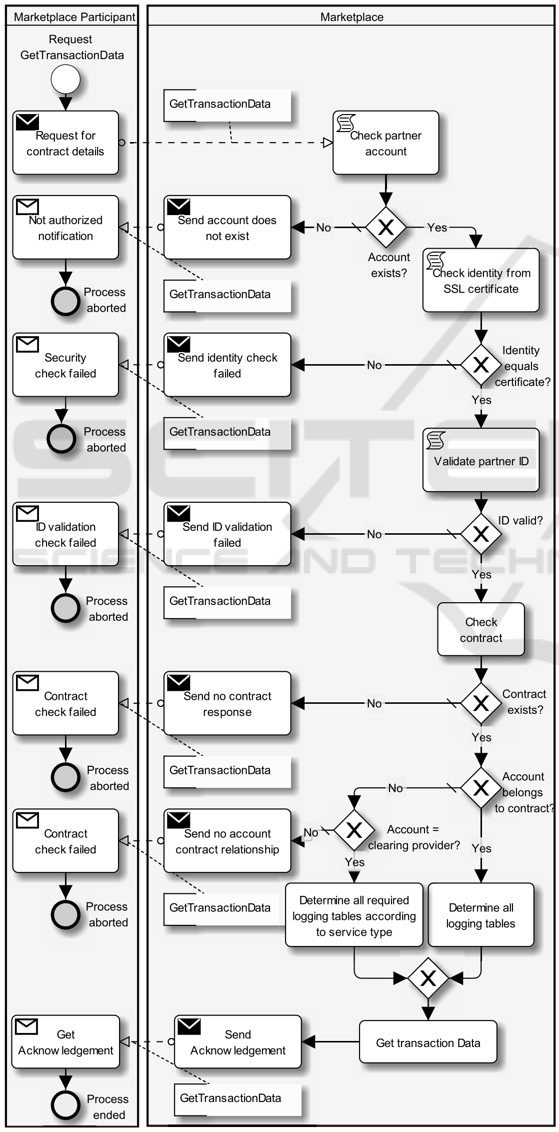

Figure 5: Process to access own or third party transaction data.

prototypically implementation has shown that prices of such time slots can differ. During the prototyping it was detected that the time slots require a validation as they should not overlap each other. Furthermore would it be necessary, in case other consumption conditions can exist, that a prioritization mechanism is in place. The address details are necessary to send the invoice to the service consumer. Future marketplaces have to consider these findings and implement them accordingly to satisfy the clearing requirement pointed out by the literature and the interviewed experts.

The *GetTransactionData* response is presented in Figure 7 on the next page. The response contains details about a transaction's status, operation and processing time. The time is important because it indicates to which time slot a transaction belongs and how it has to be charged. A general usage price is charged if a transaction does not fit into any special time slot. A transaction can maximal fit into one special time slot. All prices are taken from the contract to which a transaction belongs. A transaction is free of charge if it was not processed successfully. These characteristics have been identified during the implementation and particularly during the testing. Test data was defined and the tests executed without any human interaction. Tests were executed to check whether the identified interfaces and their design is valid and fulfill the requirements of individual service clearing or if certain parameters are still undetected. Furthermore did the tests show what data has to be available and accessible and what is the data access sequence.

5 CLEARING SERVICE PROTOTYPE

Visual Rules, a modeling framework for Business Rules Management (BRM) has been used to develop a service for service clearing. The service runs as a web service within a rule execution engine which itself runs on Apache Tomcat.

The clearing service uses and evaluates the capabilities and the completeness of the introduced interfaces for service clearing. It accesses automati-

```
<GetContractDataResp>
  <ContractDetails>
    <Contract>
      <ContractId>20003</ContractId>
      <ServiceName>ClearingService
      </ServiceName>
      <CreationDate>2015-01-15T11:36.12
      </CreationDate>
    </Contract>
    <ConsumerAddress>
      <Name>EMP</Name>
      <Address>
        <...>
      </Address>
    </ConsumerAddress>
    <ProviderAddress>
      <Name>CPO</Name>
      <Address>
        <...>
      </Address>
    </ProviderAddress>
    <Pricing>
      <UsageBasedPlan>
        <IsDeleted>false</IsDeleted>
        <BasicPrice>0</BasicPrice>
        <UsagePrice>0.2</UsagePrice>
        <TimePeriod></TimePeriod>
        <Currency>EUR</Currency>
        <TimeSpecification>
          <startDay>Monday</startDay>
          <endDay>Tuesday</endDay>
          <startTime>09:00</startTime>
          <endTime>20:00</endTime>
          <usagePrice>0.1</usagePrice>
        </TimeSpecification>
      </UsageBasedPlan>
    </Pricing>
  </ContractDetails>
</GetContractDataResp>
```


Figure 6: Structure of GetContractData response.

cally internal marketplace data from outside as a third party. The service clearing process complies to the sequence presented in Figure 4. The service's core consists of an domain independent clearing algorithm. Domain independent means that this algorithm does not differentiate or evaluate the content of a provided contract or transaction. As long as the provided data passes the interface schema, the algorithm can calculate the total costs for any kind of service transaction that has been processed via the marketplace. Without the access at transaction and contract data, a third party clearing operator would not be able for doing so via a marketplace. The algorithm is able to process any number of special time slots while considering only transactions which have been processed successfully. The developed algorithm is presented below.

Let S be the set of all relevant service consumptions:

$$S = \{x \mid x \text{ is service consumption}\}$$


```
<GetTransactionDataResp>
  <TransactionDetails>
    <Operation>STARTCHARGE</Operation>
    <RequestStatusCode>
      <sessionStatusName>Success</sessionStatusName>
    </RequestStatusCode>
    <RequestTime>2015-05-15T17:04:47</RequestTime>
    <ResponseTime>2015-07-15T17:04:54</ResponseTime>
    <RequestStatusCode>
      <sessionStatusName>Success</sessionStatusName>
    </RequestStatusCode>
  </TransactionDetails>
  <TransactionDetails>
    <Operation>STARTCHARGE</Operation>
    <RequestStatusCode>
      <sessionStatusName>Success</sessionStatusName>
    </RequestStatusCode>
    <RequestTime>2015-05-18T13:38:32</RequestTime>
    <ResponseTime>2015-05-18T13:38:39</ResponseTime>
    <RequestStatusCode>
      <sessionStatusName>Success</sessionStatusName>
    </RequestStatusCode>
  </TransactionDetails>
  <TransactionDetails>
    <Operation>STARTCHARGE</Operation>
    <RequestStatusCode>
      <sessionStatusName>Success</sessionStatusName>
    </RequestStatusCode>
    <RequestTime>2015-05-19T19:59:55</RequestTime>
    <ResponseTime>2015-05-19T20:00:02</ResponseTime>
    <RequestStatusCode>
      <sessionStatusName>Success</sessionStatusName>
    </RequestStatusCode>
  </TransactionDetails>
</GetTransactionDataResp>
```


Figure 7: GetTransactionData response data with transaction status, type and times.

For $s \in S$ is

$$\text{status}(s) := \begin{cases} \text{successful} & \text{if} \quad s \text{ successful} \\ \text{failure} & \text{if} \quad s \text{ failure} \end{cases}$$

That defines a function

$$\text{status}: \begin{array}{ccc} S & \longrightarrow & \{\text{successful}, \text{failure}\} \\ s & \longmapsto & \text{status}(s) \end{array}$$

For $s \in S$ let t_s be the timestamp (ts) that corresponds to the service consumption s. That defines a function

$$\text{ts}: \begin{array}{ccc} S & \longrightarrow & \{t \mid t \text{ is ts}\} \\ s & \longmapsto & t_s \end{array}$$

T is the set of special time slots specified in the contract:

$$T = \{Z \mid Z \text{ is special time slot}\}$$

Now let P be a function that maps special time slots to related prices.

$$P: \begin{array}{ccc} T & \longrightarrow & \mathbb{R}_{\geq 0} = \{r \in \mathscr{R} \mid r \geq 0\} \\ Z & \longmapsto & P(Z) \end{array}$$

The final formula TC is thus

$$TC = BP \cdot \chi_{\{\text{status}(s) \mid s \in S\}}(\text{successful}) + \sum_{Z \in T} \sum_{s \in S_{suc}} \chi_{Z}(\text{ts}(s)) \cdot$$

$$P(Z) + \sum_{s \in S_{suc,\,T^c}} UP$$

whereby:

$$\begin{aligned} S_{suc} &= \{s \in S : \text{status}(s) = \text{successful}\} \\ S_{suc,\,T^c} &= \{s \in S_{suc} \mid \forall\, Z \in T : \text{ts}(s) \notin Z\} \end{aligned}$$

All successful transactions are checked whether they fit into a special time slot or not. The respective usage costs are accumulated. If at least one transaction is successful then the basic usage price is charged too. The total costs is the sum of all individual transaction costs plus the basic usage price. Figure 8 depicts the output of the service-clearing service. The provided data is the one presented in Figure 6 and Figure 7.

```
<transactionList>
 <entry>
    Transaction: ChargeAuthorizationStart from:
    Friday, 2015-05-15 17:04:47 costs: 0:20
 </entry>
 <entry>
    Transaction: ChargeAuthorizationStart from:
    Monday, 2015-05-18 13:38:32 costs: 0:10
 </entry>
 <entry>
    Transaction: ChargeAuthorizationStart from:
    Tuesday, 2015-05-19 19:59:55 costs: 0:10
 </entry>
</transactionList>
<totalCost>0.40</totalCost>
<numberTransaction>3</numberTransaction>
<numberSucTransactions>3</numberSucTransactions>
```

Figure 8: Output of the developed service for service-clearing.

The value of *numberTransaction* in Figure 8 indicates how many transactions have been processed by the algorithm. The number of transactions with are actually charged, based on its status, are given by *numberSucTransactions*. Both values can be either equal or *numberTransaction* is higher than *numberSucTransactions*. The *transactionList* in Figure 8 depicts all processed transactions. This overview is provided within the invoice. It shows that different prices for the transactions have been charged. Two transactions fit into a special time specification and one does not. Therefore two transactions are charged with a special price (0.10 Euro) and one with the general usage price (0.20 Euro). Because the basic usage price is zero the total cost for the charged transactions is 0.40 Euro. Furthermore does the *transactionList* depict, that an overall number of three transactions have been processed and three transactions are considered for the total costs calculation.

Different test sets have been executed to demonstrate the algorithms performance. The test results are shown in Table 2.

Table 2: Rule model and algorithm performance measurement.

Processed Data Sets	Time Slots	Execution Time	Read Data
500	3 5 10	825-845 ms 980-1000 ms 1045-1060 ms	222 ms
500	3 5 10	1200-1270 ms 1300-1400 ms 1400-1450 ms	360 ms
2000	3 5 10	1600-1610 ms 1710-1735 ms 1790-1825 ms	572 ms
3000	3 5 10	1930-2020 ms 2120-2190 ms 2180-2360 ms	703 ms

Because the test performance depends on the underlying IT infrastructure and internet connection, a scenario has been chosen that assumes that a electronic marketplace hosts the clearing service. Transaction data is not queried from a database somewhere in the network but from a local CSV file. This avoids that the infrastructure has a deep impact on the tests. The transaction data column shows how many test transactions are processed by the rule engine and the time slots shows how many special time slots have been checked against the transactions. The execution time shows the complete execution time of the rule model, inclusive reading the test data, validate the data and compute the price algorithm. The *Read Data* column indicates how long it the clearing process took to read the test data.

6 CONCLUSION

The test cases demonstrated that the elaborated clearing service processes correctly. This implies that the interface design, implementation and execution sequence are valid. Therefore, data access across system boundaries has been achieved. The tests have confirmed our identified data which has to be accessed and combined properly to achieve service clearing at all. For the individual service clearing, the tests have shown that it is necessary to entitle certain roles to access and process third party data. This functionality is key and without no individual and outsourced service clearing is possible via a marketplace. Furthermore did the tests proof that role-based driven interfaces are suitable for the problem context and provide a powerful mechanism to enable independent interface consumption. The paper demonstrated how individual service clearing can be achieved by outsourcing it to respective marketplace participants. As pointed out by the experts, to the current day all available B2B service marketplaces in the electric mobility domain lack individual and outsourced service clearing. With the presented solution on hand, they can start implementing this urgently required functionality. They can use the presented reference implementation as a foundation to close this gap. Overcoming service marketplaces' system boundaries to access all kind of stored data is a step towards marketplace interconnectivity (Strasser and Albayrak, 2016).

This functionality is already available for flight or hotel portals, but in the mobility domain the capability for data exchange among different systems is still fiction. One of the problems why it is not yet possible to interconnect mobility systems is, according to the interviewed experts, that (electric) mobility services and marketplaces are sill under development and have no mass market. The experts point out that companies hesitate to invest money, do not open their systems and do not tightly collaborate with each other. The developed and presented interfaces have enabled the required integration and provision of individual and outsourced service clearing. They can also be used as a foundation for future interface implementations.

7 OUTLOOK ON FUTURE WORK

As Buyya et al. (2008) recognized, protocol extension is key for future service interactions. The designed interfaces can be used as a foundation to start discussions about a standard for B2B service clearing in service provisioning marketplaces. The experience gained during the research and the interface implementation will contribute to discussions about required interfaces for marketplace interconnectivity. This research area is supposed to be a topic which deserves high attention as it is of high urgency for shifting today's way of mobility. Once a marketplace provides the suggested interfaces for service clearing and interconnectivity between different marketplaces is not fiction anymore, the proposed clearing approach might be used as a foundation to enable the clearing of service requests which have been roamed between fragmented service marketplaces. For this research we will build on the proposed interconnectivity approach elaborated by (Strasser and Albayrak, 2016) as this already enables the exchange of certain data sets among fragmented service marketplaces via an appropriate and reliable connection management.

ACKNOWLEDGEMENT

This work has been conducted within a project (support code 16SBB007C) funded by the German Federal Ministry of Economics and Technology.

Interview partners were from Bosch SI, IBM R&D, T-Systems, Siemens, Smartlab GmbH, EnBW, BridgingIT, Agency for Electric Mobility, Berlin, Federal Highway Research Institute (BASt), Department of Traffic and Infrastructure (BW), University of Applied Science, Ludwigshafen (all Germany), Gireve (France), Proeftuin-EV - Flemish Institute for Technological Research (Belgium), Bosch SI (Singapore/Asia). The interviews have been conducted between July - August 2015.

REFERENCES

Å gerfalk, P., Bannon, L., and Fitzgerald, B. (2006). Action in Language, Organisations and Information Systems. Number March.

Balocco, R., Perego, A., and Perotti, S. (2010). B2b eMarketplaces Aclassification framework to analyse business models and critical success factors. *Industrial Management & Data Systems*, 110(8):1117—-1137.

Buchinger, U., Lindmark, S., and Braet, O. (2013). Business Model Scenarios for an Open Service Platform for Multi-Modal Electric Vehicle Sharing. In *SMART 2013: The second International Conference on Smart Systems, Devices and Technologies*, pages 7–14.

Buyya, R., Yeo, C., and Venugopal, S. (2008). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *Proceedings - 10th IEEE International Conference on High Performance Computing and Communications, HPCC 2008 (2008)*, pages 5–13. IEEE.

CROME (2014). CROME. http://crome-project.eu/. Last accessed on 2014-05-15.

Ghenniwa, H., Huhns, M. N., and Shen, W. (2005). eMarketplaces for enterprise and cross enterprise integration. *Data & Knowledge Engineering*, 52(1):33–59.

Green-eMotion (2015). Green eMotion. http://www.greenemotion-project.eu/. Last accessed on 2015-11-05.

Grieger, M. (2003). Electronic marketplaces: A literature review and a call for supply chain management research. *European Journal of Operational Research*, 144(2):280–294.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660.

Heinrich, B., Huber, A., and Zimmermann, S. (2011). Make-and-Sell or Buy of Web Services. In *19th European Conference on Information Systems (ECIS)*.

Hubject (2015). Hubject. http://www.hubject.com/?lang=en. Last accessed on 2015-11-5.

Johnson, M. (2013). Critical success factors for B2B e-markets: a strategic fit perspective. *Marketing Intelligence & Planning*, 31(4):337–366.

Legner, C. (2007). Is there a Market for Web Services? - An Analysis of Web Services Directories. In *Service-oriented computing-ICSOC 2007 workshops*, pages 29—-42. Springer.

Luttge, K. (2001). E-Charging API: Outsource Charging to a Payment Service Provider. *Intelligent Network Workshop, 2001 IEEE*, 00(C):216–222.

Multicity (2015). Multicity. https://www.multicity-carsharing.de/en/. Last accessed on 2015-11-05.

Olympus (2015). Olympus. http://www.proeftuin-olympus.be/en/home-1.htm. Last accessed on 2015-11-05.

Parku (2015). parku. https://parku.ch/?lang=en. Last accessed on 2015-11-05.

Pfeiffer, A. and Bach, M. (2014). An E-Clearinghouse for Energy and Infrastructure Services in E-Mobility. In *Operations Research Proceedings 2012*, pages 303–308. Springer International Publishing.

Rust, R. T. and Zahorik, A. J. (1993). Customer Satisfaction, Customer Retention, and Market Share. *Journal of retailing*, 69(2):193–215.

smartlab Innovationsgesellschaft mbh. smartlab. http://smartlab-gmbh.de/. Last accessed on 2015-05-23.

Strasser, M. and Albayrak, S. (2015). Conceptual Architecture for Self-Discovering in Fragmented Service Systems. In *7th International Confrence on New Technologies, Mobility and Security (NTMS), 2015*, Paris.

Strasser, M. and Albayrak, S. (2016). Smart City Reference Model : Interconnectivity for On-Demand User to Service Authentication. In *12th International Conference on Industrial Engineering (ICIE) (to be published)*.

Strasser, M., Weiner, N., and Albayrak, S. (2016). Outsourced Invoice Service : Service-Clearing as SaaS in Mobility Service Marketplaces. In *Network Operations and Management Symposium (NOMS) (to be published)*.

Streetlife (2015). Streetlife. http://www.streetlife-project.eu/index.html. Last accessed on 2015-11-05.

Thitimajshima, W., Esichaikul, V., and Krairit, D. (2015). Developing a Conceptual Framework to Evaluate Public B2B E-Marketplaces. In *Proceedings of the Pacific Asia Conference on Information Systems 2015*.

Turban, E., King, D., Lee, J., Liang, T.-P., and Turban, D. C. (2015). *Electronic Commerce - Business-to-Business E-Commerce*. Springer.

Vidal, N., Iaarakkers, J., Marques, R., Scuro, P., Caleno, F., Matrose, C., and Bolczek, M. (2011). Report on Billing and stakeholders architecture and ICTs recommendations. Technical Report 241295, G4V.