IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 17, 2019, accepted July 15, 2019, date of publication July 29, 2019, date of current version August 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2931762

Modeling of Collaboration Archetypes in Digital Market Places

LU ZHANG[1], REGINALD CUSHING[1], LEON GOMMANS[1,2], CEES DE LAAT[1], AND PAOLA GROSSO[1]

[1]Systems and Networking (SNE) Lab, University of Amsterdam, 1098 XH Amsterdam, The Netherlands
[2]Air France KLM Group, 1119 NX Amsterdam, The Netherlands

Corresponding author: Lu Zhang (l.zhang2@uva.nl)

This work was supported in part by the Dutch Top Consortia for Knowledge and Innovation, Institute for Advanced Logistics, (TKI Dinalog) Ministry of Economy and Environment, The Netherlands, and in part by the Dutch Commit-to-Data initiative.

ABSTRACT With everyone collecting and generating value out of data, this paper focus on distributed data trading platforms, digital market places (DMPs). The DMPs can handle the intricacies of data sharing: how, where, and what can be done with the traded data. Here, we represent collaborations among involving parities in DMPs in the form of archetypes and model them with numeric representations for easier manipulation with standard mathematical tools. We also develop an algorithm that aims to map any customer-defined trust-dependent application request into a best-fit infrastructure archetype in a DMP. Also, we propose multiple metrics that allow evaluate and compare competing the DMPs systemically from more dimensions: coverage, extensibility, precision, and flexibility. We demonstrate the effectiveness of these metrics in a concrete use case.

INDEX TERMS Digital market places (DMP), trust, collaboration archetypes, evaluation metrics.

I. INTRODUCTION

In the era of big data, the amount of collected data is increasing dramatically [1], [2]. Sharing and utilizing such data can generate great value and improve collaborations among parties [3]. But security and privacy concerns may arise, especially in scenarios that members are normally competing with each other [4]. Newly emerging Digital Market Places (DMP) concept aims to facilitate such trusted big data sharing for a specific purpose [5], [6]. In this paper, we propose a method to match applications to the closest infrastructures, in the form of archetypes, in a DMP. We also define a set of metrics to evaluate and compare with competing DMPs.

A DMP is a membership organization bringing parties together to share data assets for achieving a common goal. A well-known example is Airbnb. It constructs a distributed computing platform which allows providers and consumers to trade and share their data asset and creates a trusted infrastructure for data processing. A DMP may be governed by a consortium to prevent asset exposure. The transactions within a DMP must comply with a digital contract, agreed by all members, to regulate everything from data movement to algorithm execution.

A potential DMP customer normally participates in different DMPs for different applications. Because both collaborating partners and collaboration purposes are varying with requirements of individual application. For example, airline companies would like to predict the necessity of aircraft maintenance with AI/ML algorithms. They can certainly benefit from a more accurate prediction by gathering data of the same type of aircraft. Certainly keeping data sovereignty is crucial since the data is shared with competitors. But one company may need to collaborate with a different set of airline partners for different aircraft types. And the collaboration request changes correspondingly with different trust among involving parties.

This begs a question: *How to map applications into best fit infrastructure patterns in a specific DMP?* Also, it is quite interesting to have a deeper understanding and a more systematic description of the capability and features of those DMPs. The concept of DMPs is, though very promising, a relatively new research field. As far as we know, there are no established and standardized metrics to evaluate the performance of DMPs and compare competing ones. The main contributions of our work are:

- We model multi-party collaborations numerically with 3D matrices; We also develop an algorithm to reason on the mathematical representations of collaborations

The associate editor coordinating the review of this manuscript and approving it for publication was Kuo-Hui Yeh.

with an effort to match any concrete complicated collaboration request into the best fit distributed computing archetype from the DMP.

- Define multiple metrics to evaluate a DMP from various aspects; namely, we identify *coverage* and *extensibility* as metrics to describe properties and features of a DMP itself; and *precision* and *flexibility* describe the performance associated with a specific user request to the DMP.

II. DMPS AND COLLABORATION MODELS

A DMP is a membership organization to support members to achieve a common goal by data asset sharing. Figure 1 illustrates a high-level framework of a DMP. The movement and processing of data objects and compute objects are governed by an *Agreement* achieved by all members, such as data suppliers and algorithm providers, in this DMP instance. The *Infrastructure Pattern* is dependent on concrete *Agreement* for each DMP instance and those rules are enforced by underlying *Data Exchange Infrastructure* with future network capabilities.

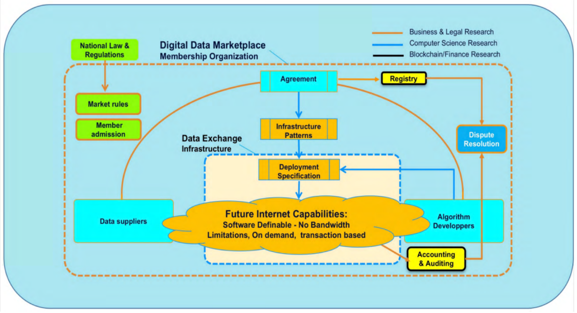

Business & Legal Research
Computer Science Research
Blockchain/Finance Research
National Law & Regulations
Digital Data Marketplace Membership Organization
Agreement
Registry
Market rules
Member admission
Infrastructure Patterns
Data Exchange Infrastructure
Dispute Resolution
Deployment Specification
Data suppliers
Future Internet Capabilities: Software Definable - No Bandwidth Limitations, On demand, transaction based
Algorithm Developpers
Accounting & Auditing

FIGURE 1. A high-level framework of an example Digital Market Places (DMP).

The *Agreement* of a DMP instance contains information about how data and compute objects flow, where to perform the execution and how intermediate results aggregate and so on. Collaboration models are defined to describe such restrictions and serve a role in connecting the *Agreement* to the underlying digital infrastructure. For example, different collaboration models might have different vulnerabilities and threats, which require different defense mechanisms in the underlying infrastructure to achieve optimization between security and performance.

Normally, collaboration models are defined and described from both the DMP operator perspective and potential customer perspective. Here we clarify some terminologies for better explanations. From the DMP operator side, we call those collaboration models as *Collaboration Archetypes*. From a DMP customer side, we call those collaboration models as *Application Requests*.

A. COLLABORATION ARCHETYPES

Each DMP may support one or more collaboration archetypes to allow potential customers to choose from.

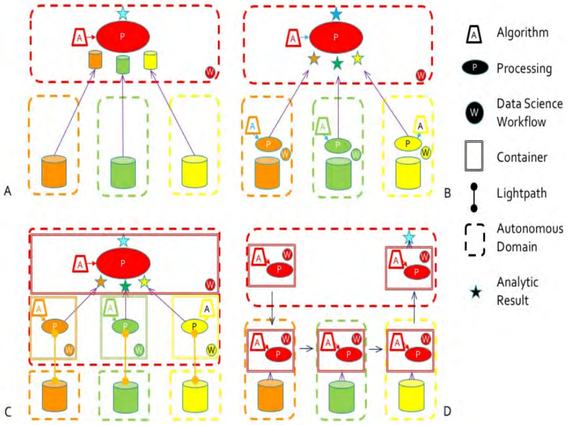


FIGURE 2. Example collaboration archetypes of a Digital Market Places (DMP).

Figure 2 illustrates four collaboration archetypes. Multiple parties, located in distributed places, aggregate their data and compute objects for a result to achieve a common goal. In Archetype A, all the data are transferred and aggregated in the compute object provider. In Archetype B, compute objects come to data providers and data are processed locally and separately. Intermediate results are then merged in compute object provider. For archetype C, the data and compute meet in a trusted 3rd party. The data from each data set is processed separately for an intermediate result and then merged at compute object provider. For archetype D, data are processed locally in each database by the compute object transferred from its provider. However, the intermediate results are not merged in one physical location, like archetype A, B, C, but aggregated in a cascaded manner.

Based on the definition in [7], archetypes are defined as an original model or type based on which similar things are patterned. We call these collaboration models, from DMP perspective, archetypes because they only capture the main features but are not specific to some details. Those details include the concrete participating parties and the total number of parties for the collaboration.

B. APPLICATION REQUEST

A potential customer, a group of members who would like to collaborate for data sharing for a common goal, may come to a DMP with a concrete collaboration request and seek a best-fitted collaboration archetype. We call such collaboration models as application requests.

Application requests describe how the involving members would like to share their assets in the specific application. Normally application requests are included in the *Agreement* and highly depending on the trust relationships among involving members.

Figure 3 describes a concrete application request. Party A would like to perform its algorithm on the data from Party B. But Party A and B do not trust each other, so they employ a trusted third party C and send their compute and data to

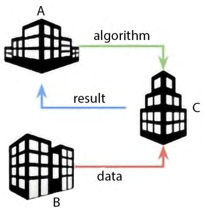


FIGURE 3. An example application request from a potential DMP customer.

it. Party C executes A's algorithm on B's data and sends the result back to A.

A customer-defined application request may comprise both hard requests and soft requests. Hard requests are not negotiable and must be fulfilled in the collaboration process. However soft requests could be adjusted to better fit any existing collaboration archetype.

III. MODELING OF MULTI-PARTY COLLABORATIONS

To manage and manipulate multi-party collaborations among participating members in a DMP, we should, in the first place, model them properly. They are modeled with numeric representations because we believe this would give us standard mathematical tools to further reason about them. For example, we can measure the similarity between an archetype and an application request by computing mutual distance with those mathematical representations.

Firstly, a bilateral collaboration relationship can be fully described by four attributes:

1) *Source* is the resource provider;
2) *Target* is the resource consumer;
3) *Collaboration level* represents the concrete approach of resources exchange;
4) *Collaboration scope* describes which resource could be shared between specific parties [8].

Collaborations among participating members may take place in multiple scopes, *data scope*, *algorithm scope* and *intermediate result scope*. More scopes can be added when necessary, e.g. geographical locations.

TABLE 1. Collaboration levels under individual scopes.

Collaboration levels	*Data Scope*	*Algorithm Scope*	*Intermediate Result Scope*
1	Remote Mounting	Partial Algorithm	Cascaded Aggregation
2	Direct transfer	Entire Algorithm	Parallel Aggregation

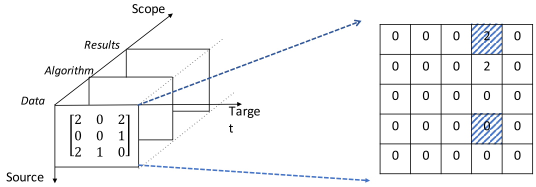
Also, collaboration level captures important information about concrete collaborating approaches of each scope between parties. Those features may influence the implementation and performance of underlying digital infrastructures. Table 1 explains the concrete collaborating approaches represented by collaboration levels under each scope. These values are ordered and larger numbers indicate a stronger collaboration, which implies more trust between *source* and *target* parties.

In *data scope*, the collaboration levels indicate whether the data is accessed by the *target* with directly data transfer or remote file system mounting. In *algorithm scope*, *partial algorithm* means that *source* only shares the necessary part of its algorithm, dedicated to individual partners, to reduce information exposure. *Entire algorithm* means that the total algorithm is shared for all distributed partners and this certainly requires more trust from *source* to *target*. In *intermediate result scope*, collaboration levels represent whether the intermediate result is aggregated in a parallel manner, illustrated in Figure 2 (C), or a cascaded manner, illustrated in Figure 2 (D).

A bilateral collaboration relationship is represented as $\{source, target, scope_1 : level_1 \cdots , scope_n : level_n\}$.

For each scope, a multi-party collaboration relationship can be modeled as a labeled weighted graph and represented as its corresponding adjacent matrix.

We denote the graph as $G(V, E, W)$. The set of nodes V represent participating members. The edges set E represent bilateral collaboration relationships and weights W represent corresponding collaboration levels. For example, w_{ij} is the *collaboration level* from member i to member j. We also use labels to indicate whether a bilateral collaboration relationship belongs to hard or soft requests when modeling an application request.


Scope
Results
Algorithm
Data
Target
Source
$\begin{bmatrix} 2 & 0 & 2 \\ 0 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix}$

0	0	0	2	0
0	0	0	2	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

FIGURE 4. Modeling of a multi-party collaboration relationship. On the left we see the relations between sources and targets for the three scopes; on the right, we zoom in on one specific scope, where the crossed-out cells represent hard requests.

As illustrated in Figure 4, a multi-party collaboration relationship among multiple members is effectively modeled as a 3D matrix. Each 2D matrix along scope-axis is the adjacent matrix of a graph under a specific scope.

IV. SELECTION OF COLLABORATION ARCHETYPES IN A DMP

Each DMP may support multiple collaboration archetypes to meet individual application requests. The requests may vary over applications and even vary in time. Therefore it is highly beneficial to develop an algorithm to perform the matching procedure from any incoming application request to a collaboration archetype supported by DMP.

We define similarity measures between collaboration models, which is effectively quantified as a distance metric. Either a collaboration archetype or an application request can be mapped as a point in a discrete space by calculating their mutual distances.

The algorithm aims to select a collaboration archetype which fully satisfies hard requests from customer and best fits the soft requests. Here ''best fit'' means the highest similarity, which is described by minimum distance to the input application request.

A. ALGORITHM OVERVIEW

The matching algorithm consists of two stages, filtering (Stage I) and archetype selection (Stage II). Figure 5 describes the algorithm flowchart.

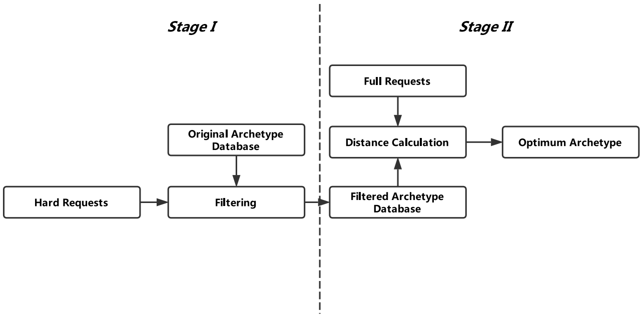


FIGURE 5. Flow chart of the archetype selection algorithm. Stage I is concerned with filtering the archetypes based on hard requests, and Stage II calculating distances to identify the optimal archetype.

At Stage I, all collaboration archetypes from *Original Archetype Database* are filtered with *Hard Requests* given by a potential customer. After *Filtering*, a subset of archetypes are kept in *Filtered Archetype Database* for further processing and the corresponding searching space shrinks. All the remaining archetypes are acceptable by potential customers for the compliance with *Hard Requests*.

At Stage II, we first calculate the distances between *Full Application Request* and remaining archetypes in *Filtered Archetype Database*. Then select the *optimal archetype* as the one with minimum distance towards *Full Application Request*.

The operational details of each stage are described in the remaining part of this section.

B. STAGE I: FILTERING WITH HARD REQUESTS

An application request includes three scopes, as discussed in Section III, and we perform the filtering stage scope-wise. Suitability under one specific scope does not necessarily mean a completely identical adjacent matrix. For example, if an application requires no 3rd party, any matrix with all-zero entries in the corresponding positions are qualified. The mechanism is illustrated in Figure 6.

Scope Priority depends on the ratio of hard request entries in each scope. Higher priority is achieved for more non-negotiable request entries. A tree structure is automatically generated with inputs of *Scope Priority* and *Original Archetype Database*.

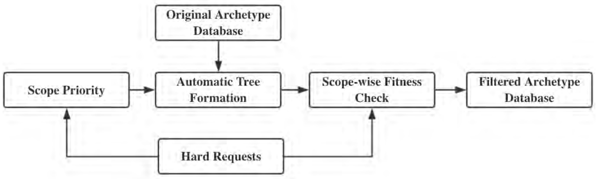


FIGURE 6. Stage I components performing the filtering.

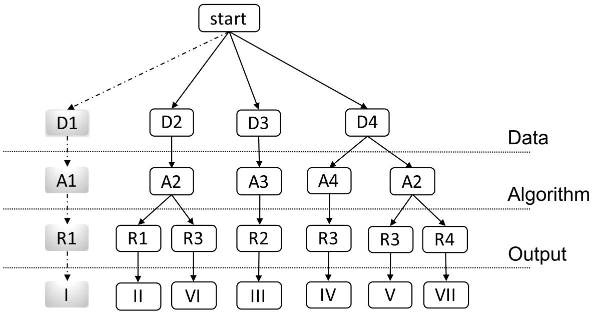


FIGURE 7. An example tree structure formed by the filtering mechanism.

An example tree structure with *Scope Priority* [data, algorithm, output] is shown in Figure 7. The path from *start* to a concrete collaboration archetype consists of matrices under each scope and different archetypes may share the same scope-level matrix. If the data scope matrix D1 does not satisfy the hard request, all its children nodes are excluded from the search space.

C. STAGE II: DISTANCE CALCULATION AND ARCHETYPE SELECTION

We should define a distance calculation method, which can measure the dissimilarities among collaboration models effectively. A smaller distance is expected for two collaboration models who are intuitively more similar.

What do we mean with similarities among collaboration models? Firstly, multi-party collaborations are more similar if more bilateral collaboration relationships are equivalent. Secondly, two bilateral collaboration relationships are more similar if they are identical in more scopes. Thirdly, the existence of a collaboration between parties weights more in our similarity assessment than the level to which they collaborate. The distance calculation method is illustrated in Figure 8.

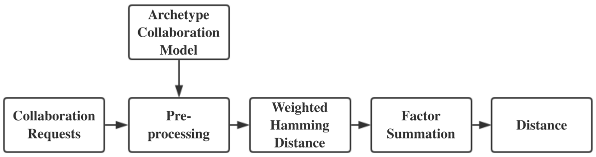


FIGURE 8. Stage II components performing the distance calculation for individual collaboration archetypes.

Firstly, we pre-process both *Application Request* and *Collaboration Archetype* for more commensurate comparison. In *Pre-processing* module, we adjust the dimension of collaboration archetypes in the database to the dimension of the input application request, which is equal to the number of involved parties. Also, we extract all non-zero vectors along the scope axis, each of which represents a bilateral collaboration relationship. We call such vectors *bilateral relationship vector* and each vector can be denoted as $\{source, target, (level_1, level_2, level_3)\}$. Also, *source* and *target* in the *bilateral relationship vector* are represented as the roles of involving parties instead of concrete matrix indexes. The purpose is to eliminate the influences of how those members are positioned into a collaboration matrix to represent their application requests. Those *bilateral relationship vectors* from both application request and collaboration archetype are passed from *Pre-processing* module to the next.

In the *Weighted Hamming Distance* module, we calculate weighted Hamming distances between pairs of *bilateral relationship vectors* with equivalent $\{source, target\}$ [9].

The distance between two collaboration models is achieved by summing up all the individual Hamming distances generated from *Weighted Hamming Distance* module. This is performed in the module *Factor Summation* and the mathematics equation 1 is

$$D(\mathrm{CM}_i, \mathrm{CM}_j) = \sum_{s=0}^{P-1}\sum_{t=0}^{P-1}\sum_{k=0}^{S-1} w_k[\mathrm{level}(i)_{s,t,k} \neq \mathrm{level}(j)_{s,t,k}], \qquad (1)$$

where CM_i denotes ith collaboration model. It can be either a customer-defined application request or a collaboration archetype supported by a DMP. P, S denote the number of involved parties and number of defined scopes respectively. $\mathrm{level}(i)_{s,t,k}$ denotes the collaboration level from source s to target t at kth scope in collaboration model i. w_k is the weight of Hamming distance, which is jointly decided by scope priority and collaboration entries.

As discussed previously, the *source* or *target* are represented as roles of members rather than index. So there may be multiple *bilateral relationship vectors* with same $\{source, target\}$. The distance is the minimum value of all results computed from all *bilateral relationship vector* combinations between two collaboration models. We aim to find an optimum archetype for a concrete application request by considering all possible arrangements of members when they put themselves into the matrix to represent their application request.

V. EVALUATION METRICS OF A DMP

As we discussed in the previous sections, application requests can be matched into most similar collaboration archetypes in a DMP.

For potential customers it is interesting to know a-priori how easily one of their application requests can be fulfilled by a particular DMP; for DMP operators it is important to assess how well they can serve their user base generally.

Suppose that two DMPs all support an equal number of archetypes. They may performance differently according to particular customer-defined application requests or mutual distances among archetypes in the discrete space. For example, if all archetypes of a DMP are concentrated in a small area, it might have less capability to fulfill overall application requests than a DMP whose archetypes are sparsely distributed. We propose multiple metrics that allow more nearly complete evaluation of a DMP:

- *Coverage*: How well the overall application requests can be satisfied by a DMP with a certain mismatch.
- *DMP Extensibility*: What is the potential richness of a DMP by decomposing and composing collaboration archetypes.
- *Application Extensibility*: How elastic an application request is for achieving a perfect match with a given DMP.
- *Precision*: How well the supported collaboration archetypes of a DMP fit an application request.
- *Flexibility*: How easily an application request can be satisfied generally.

Metrics like *coverage* and *DMP extensibility* are not related to individual requests but represent a general feature of a DMP. However, *precision*, *flexibility* and *application extensibility* depend on both concrete customer-defined application requests and DMP itself.

Besides conceptual definitions, we also define quantization methods for each metric, which we will introduce in detail in the following.

A. COVERAGE

With metric *coverage*, we can assess how well the overall application requests can be satisfied by the archetypes of a given DMP. It is intuitively clear that *coverage* highly depends on how we define customer satisfaction. In our work, a potential customer is considered as satisfied if the distance, between her application request and the optimum archetype, is not larger than a pre-defined value. We call the parameter *affordable distance* and denote it as D_A.

First, we try to identify the number of overall application requests. Suppose a DMP supports collaboration archetypes $A = \{A_1, A_2, \ldots, A_n\}$. Let P, S, and C denote the number of participating parties, number of defined scopes, and number of collaboration levels respectively. Since the diagonal elements are invalid in a collaboration matrix, the number of entries containing effective collaboration information N_E is

$$N_E = (P^2 - P) * S \qquad (2)$$

Theoretically, the total number of possible collaboration models with fixed P, S and C is

$$N_T = C^{N_E} \qquad (3)$$

In reality, this number of feasible collaboration models is much smaller. On the one hand, not all collaboration matrices describe a valid collaboration model. On the other hand, multiple mathematically different collaboration matrices might represent the same collaboration model due to symmetry. We will develop a feasibility validation model in future work.

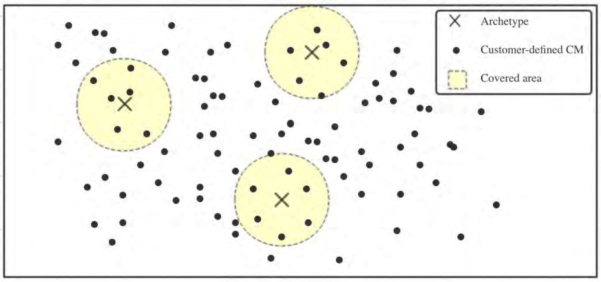


FIGURE 9. Illustration of coverage in discrete space, with archetypes identified as crosses, application requests as dots, and covered areas represented by the yellow circles.

As illustrated in Figure 9, the covered area of ith archetype A_i is modeled as a sphere with radius of the affordable distance D_A. The total covered area of multiple collaboration archetypes is the union of individual covered area.

Ultimately, *coverage* is quantified as the percentage of the application requests, that fall into the covered area of supported archetypes, over the total number of overall collaboration models.

$$coverage = \frac{N_{\text{covered}}}{N_T}, \tag{4}$$

where N_{covered} denotes the number of application requests that fall into total covered area of the DMP.

Coverage is calculated by computing the distances between all possible application requests and supported archetypes. But this leads to heavy computational burden and the complexity grows exponentially with larger C and P.

We develop an optimization algorithm to reduce computation complexity. The general principle is to exclude those application requests, which surely fall outside the covered areas, before simulation.

Described in algorithm 1, $N_{nz,A}$ is the number of non-zero entries in the collaboration matrix of a supported archetype and w_h is the maximum weight of Hamming distance in equation 1. We sort overall application requests with the number of non-zero entries in their collaboration matrices and AR_i is the set of application requests with i non-zero entries. $AR_{covered,i}$ denotes number of covered application requests in AR_i.

For instance, if there are four and seven non-zero entries in an archetype matrix and an application request matrix respectively, then at least three entries are not overlapped and contribute to a distance of $3 * w_h$. So there is a limit for the number of non-zero entries in the application request matrix to achieve a distance smaller than D_A. This maximum number of non-zero values $N_{nz,\max}$ is calculated from line 2 to 6 in Algorithm 1. Next, the algorithm deals with each AR_i with an increasing number of nonzero entries i and computes a $AR_{covered,i}$. When i is larger than $N_{nz,A}$, there are also limitations about how these entries distribute in the collaboration matrix and the number of iterations could be further reduced as indicated in lines 11 and 12.

Algorithm 1 Optimization Algorithm for *Coverage* Calculation

```
1:  Input $D_A$, $N_{nz,A}$ and $w_h$
2:  if $D_A$ is even then
3:      $N_{nz,\max} = \frac{D_A}{w_h} + N_{nz,A}$
4:  else
5:      $N_{nz,\max} = \frac{D_A+1}{w_h} + N_{nz,A}$
6:  end if
7:  for $AR_i \in \{AR_0, AR_1, \ldots, AR_N\}$ do
8:      if $i \leq N_{nz,A}$ then
9:          compute $AR_{covered,i}$ by iterating all request $\in AR_i$
10:     else
11:         reduce $AR_i \rightarrow AR_{re,i}$ by restricting matrix deployment
12:         compute $AR_{covered,i}$ by iterating all requests $\in AR_{re,i}$
13:     end if
14:     $AR_{covered} = AR_{covered} + AR_{covered,i}$
15: end for
```

B. DMP EXTENSIBILITY

DMP Extensibility measures the potential richness of a DMP by recombining collaboration archetypes.

Each archetype can be decomposed into multiple basic blocks. Each basic block describes collaborations among two or three parties of the same trust domain and we call them primitives. Different collaboration archetypes may share the same primitives. The primitive set of a DMP is the union of primitives of its supporting collaboration archetypes.

Suppose the primitive set of a DMP is $P = \{P_l | l = 1, 2, \ldots, N\}$ and a new collaboration archetype can be constructed as

$$A = r_1P_1 + r_2P_2 \cdots + r_NP_N = \sum_{l=1}^{N} r_lP_l, \tag{5}$$

where r_i denotes the number of repeating times of each primitive.

DMP extensibility is a measure of the ability to enrich DMP by archetype recombination. It can be measured as

$$\text{DMP } Extensibility = 1 - \frac{N_{A,o}}{N_{A,e}} \tag{6}$$

where $N_{A,o}$ denotes the number of original archetypes of a DMP and $N_{A,e}$ denotes the number of possible archetypes with the primitive combination.

C. APPLICATION EXTENSIBILITY

Application extensibility describes the elasticity of an individual application request in achieving a perfect match towards a given DMP. It is quantified as the percentage of unmodified soft entries over all the soft entries in the collaboration matrix. We set the metric as $-\infty$ if a zero distance is not reachable with this DMP by adjusting soft entries in the application. *Application extensibility* is calculated as

$$App\ Extensibility = 1 - \frac{N_{m,soft}}{N_{soft}}, \tag{7}$$

where N_{soft} denotes the number of soft entries in a collaboration matrix and $N_{m,soft}$ denotes the number of modified soft entries for a perfect match. This metric is related to *flexibility* in Section V-E. This metric is conditional and is only valid when there are soft requests in the application request.

D. PRECISION

Precision describes how well the supported archetypes of a DMP match a specific application request of potential customers. This metric is calculated as

$$precision = 1 - \frac{D_{\mathrm{min}}}{D_{\mathrm{A}}} \tag{8}$$

$$D_{\mathrm{min}} = min(Distance(\mathrm{AR}, A_i)), \tag{9}$$

where D_{min} denotes the distance between an application request *AR* and the optimum archetype in the DMP, D_{A} is aforementioned *affordable distance*.

If a perfectly matched archetype exists in a given DMP with $D_{\mathrm{min}} = 0$, *precision* regarding to the application request is 1. If D_{min} is exactly D_{A}, the *precision* turns out to be 0. Otherwise if D_{min} is significantly larger than D_{A}, *precision* results in a negative value.

E. FLEXIBILITY

The metric *flexibility* measures the strictness of an application request. It is quantified as

$$Flexibility = 1 - \frac{N_h}{N_E}, \tag{10}$$

where N_h denotes the number of hard request entries in a collaboration matrix, N_E denotes the number of entries containing efficient information, which can be calculated by equation 2.

F. INTELLIGENT SELECTION ALGORITHM

With the values of proposed metrics for each DMP, the customer will get information about which DMP meets his or her application request best.

Algorithm 2 explains the concrete procedure of metric analysis. It aims to select the 'best' DMP who can provide a perfect matched collaboration archetype for the application request with minimum modification effort and relatively higher *coverage*.

First of all, the algorithm sorts all DMPs on *coverage* in a descending order to ensure that the winner always has the highest *coverage* among the qualified members.

In the first step, it analyzes *precision*, described from lines 3 to 8, to check if any DMP can provide a perfectly matched collaboration archetype without any modification. If so, it selects this DMP and ends the procedure.

In the second step, the algorithm checks which DMP can provide an exactly matched archetype by only extending the application request. This is done by analyzing metrics of *flexibility* and *Application Extensibility*. Line 9 checks if there are any soft requests in this application request. Line 10 checks whether the distance can be shortened to zero by just soft request adjustments. If so, the DMP with minimum modification of application request, a minimum value of *Application Extensibility*, is selected.

Finally, the algorithm enriches the DMP candidate pool by archetype recombination and checks whether a DMP, in the enriched pool, can fully satisfy the application request. They are indicated from line 16 to 22.

Algorithm 2 Intelligent Selection Algorithm With a Specific Application Request

```
 1: Input application request → AR
 2: Sort DMPs with coverage in descending order → $\mathrm{DMP_{rank}}$
 3: for $\mathrm{dmp}_i \in \mathrm{DMP_{rank}}$ do
 4:     if precision($\mathrm{dmp}_i$, AR) = 1 then
 5:         $\mathrm{dmp}_i \rightarrow \mathrm{dmp_{opt}}$
 6:         go to output
 7:     end if
 8: end for
 9: if flexibility(AR) > 0 then
10:     if ∃ app extensibility ≥ 0 then
11:         Select $\mathrm{dmp}_i$ with maximum app extensibility
12:         $\mathrm{dmp}_i \rightarrow \mathrm{dmp_{opt}}$
13:         go to output
14:     end if
15: end if
16: Extend $\mathrm{DMP_{rank}}$ by primitive composition → $\mathrm{DMP_e}$
17: for $\mathrm{dmp}_i \in \mathrm{DMP_e}$ do
18:     if precision($\mathrm{dmp}_i$, AR) = 1 then
19:         $\mathrm{dmp}_i \rightarrow \mathrm{dmp_{opt}}$
20:         go to output
21:     end if
22: end for
23: output:
24: Return $dmp_{opt}$
```

VI. APPLICATION USE CASE: DATA LOGISTICS

Our proposed metrics are intended to aid the DMP operators and DMP users to optimally define their archetypes and make better decisions. In this section, we will evaluate the effectiveness of the metrics with project DL4LD [10]. We will be applying the research results presented here in this context.

A. DL4LD

The goal of the DL4LD project is to help the Dutch logistics sector with IT tools that promotes digital business processes, with particular support for the trustworthy sharing of sensitive data. Specifically, DL4LD shows how to establish, digitally, sufficient trust to execute a data-transaction between two ad-hoc logistic partners. This includes the digital negotiation of legal contracts for data sharing and data operations. DL4LD also shows how digital contracts are input for automated setting up of the required digital infrastructure.

At this moment in the project, we have defined seven archetypes for secure data sharing and digital collaboration for logistic parties.[1]

An example use case from DL4LD concerns airlines. Airline companies, e.g. KLM and AirFrance, would like to predict the need for aircraft maintenance by operating AI/ML algorithms on the aircraft data. It is commonly known that a more reliable prediction result is achieved by better availability of training data. It is beneficial for those companies to gather the data of the same aircraft type for collaborative computing. But these companies are competing with each other and normally have a preferred collaboration model for privacy and confidentiality consideration.

B. SPATIAL DISTRIBUTION AND MUTUAL DISTANCES

We first looked at the spatial distribution of all seven DL4LD archetypes.

We computed the pair-wise mutual distances among all the archetypes. The corresponding results with four parties are shown in Table 2. The resulting matrix is upper-triangular because of the symmetry property of distances in space.

TABLE 2. Mutual distances between archetypes defined in project DL4LD.

	I	II	III	IV	V	VI	VII
I	0	10	12	12	12	12	12
II	-	0	14	5	4	2	4
III	-	-	0	16	16	16	16
IV	-	-	-	0	1	3	2
V	-	-	-	-	0	2	1
VI	-	-	-	-	-	0	3
VII	-	-	-	-	-	-	0

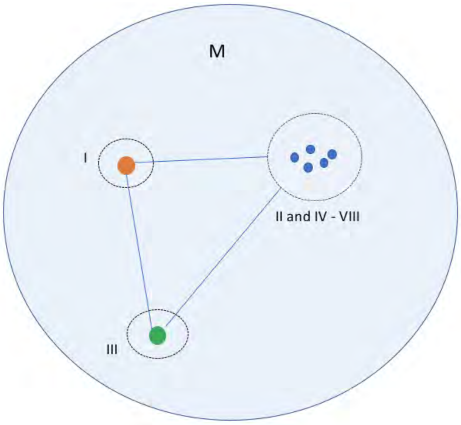


FIGURE 10. Spatial distribution of archetype collaboration models in the DL4LD project.

According to these relative distances, we can visualize the spatial distribution of those archetypes. As illustrated in Figure 10, archetype I and III are more isolated with others and archetype II, IV, V, VI, and VII are clustered together. This computation result is in accordance with the similarity between archetypes.

VII. METRICS EVALUATION FROM DMP OPERATOR PERSPECTIVE IN DL4LD

In this section, we evaluate DMPs, who support different archetype sets by computing and analyzing *coverage* and *DMP extensibility*.

A. EXPERIMENTAL DESIGN

We assign the total seven archetypes into various subsets and suppose each of them is supported by an individual DMP. The number of all possible archetype combinations with a particular set size is shown in Table 3. We will compute *coverage* and *DMP extensibility* of all those individual DMPs.

TABLE 3. The number of possible archetype combinations with increasing set size.

Archetype Set Size	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Number of Subsets	7	21	35	35	21	7	1

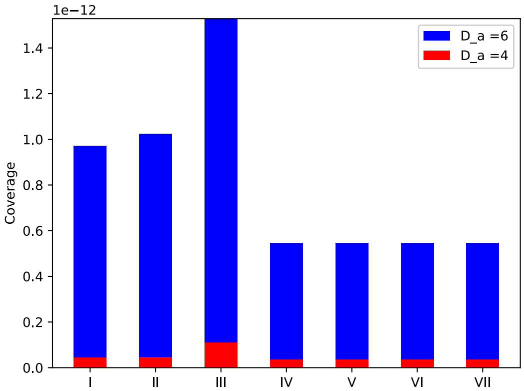


FIGURE 11. Individual *coverage* of each archetype, with $D_A = 4$ and $D_A = 6$ respectively.

B. ANALYSIS AND DISCUSSION

Figure 11 shows the *coverage* of each archetype with affordable distance $D_A = 6$ and $D_A = 4$. Every single archetype may have different capabilities to serve the overall request space with an identical pre-defined covered area. Archetype III has the highest *coverage*, which implies a higher density of feasible application requests in its neighboring space. Also, the value of affordable distance D_A plays an important role in *coverage*.

A DMP operator may get more complete information about its supported archetypes by computing and analyzing metric *coverage*. For instance, the DMP operator may expect that implementing archetype III and corresponding infrastructures is more beneficial for the ability to meet overall collaboration requests.

More generally, *coverage* of all other archetype sets are computed with optimization algorithm discussed in Section IV. The corresponding computation results are illustrated in Figure 12.

[1] https://bitbucket.org/uva-sne/dl4ld_public_documents/src

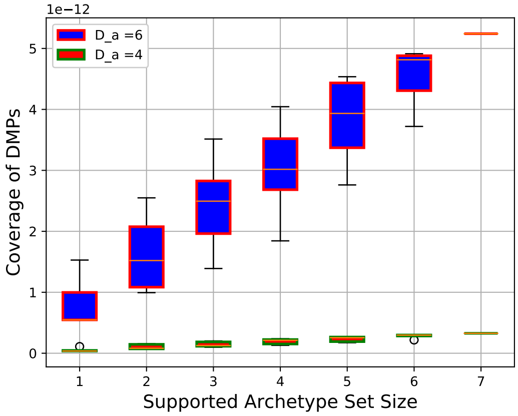


FIGURE 12. ***Coverage*** **as a function of increasing archetype set size with $D_A = 4$ and $D_A = 6$ respectively.**

In Figure 12, each group represents *coverage* of DMPs supporting archetype sets with equal size. It is not surprising that *coverage* increases approximately in a linear manner with a larger archetype set size. If a DMP operator implements and supports more collaboration archetypes, it certainly has a higher possibility to satisfy more requests. But it is usually more expensive.

By analyzing data of proposed metrics, a DMP operator may find a better solution between implementation cost and achieved *coverage*. Shown in Figure 12, most inter-quartile range boxes have overlap values with their neighbors. This indicates that a DMP, who supports a larger number of archetypes, may result in a relatively lower *coverage*. One DMP operator or customer may beneficially select a specific archetype set who has higher *coverage* but lower archetype size.

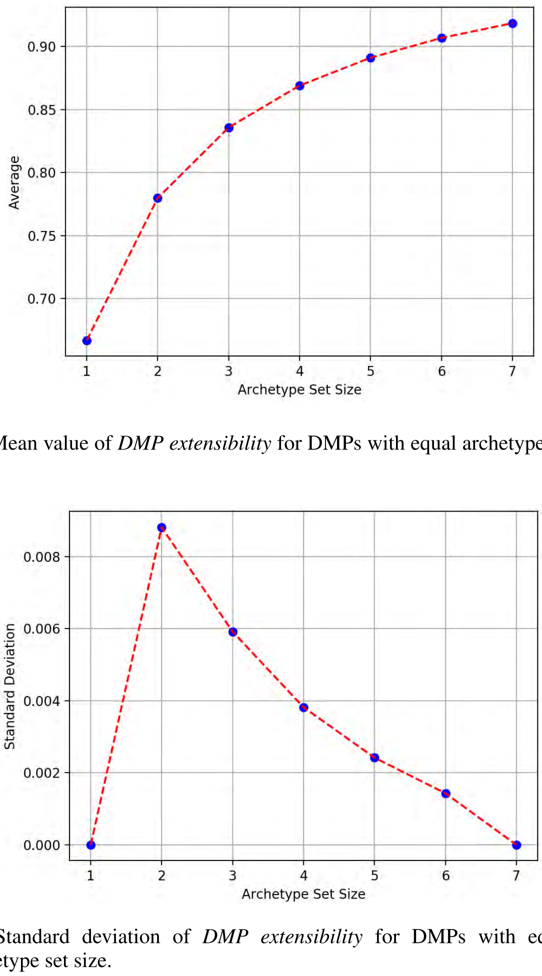
Similar with *coverage*, *DMP extensibility* is also an evaluation metric defined from DMP operator perspective and independent of particular collaboration requests. It represents the richness a DMP can achieve by constructing new archetypes by primitive composition. In some scenarios, a DMP with lower *coverage* may have higher *DMP extensibility*.

Figure 13 shows statistic information about the values of *DMP extensibility* in DL4LD. *DMP extensibility* increases non-linearly with more supported archetypes. The mean value increases faster when the supported archetype size grows from 1 to 4 and becomes relatively stable after the number reaches 5. The standard deviation of *DMP extensibility* for DMPs with equal archetype set size is very small. It is because that every archetype in DL4LD has only one primitive.

For *DMP extensibility*, it may be more interested to investigate how *coverage* or *precision* would increase after DMP extension. We would discuss some of them in next Section.

VIII. METRIC EVALUATION WITH SPECIFIC APPLICATION REQUESTS IN DL4LD

In this section, we evaluate multiple DMPs in DL4LD by computing all the five metrics with two concrete application requests of the airline use case. An optimum DMP is selected for each scenario by analyzing those metrics intelligently with Algorithm 2.



(a) Mean value of *DMP extensibility* for DMPs with equal archetype set size.

(b) Standard deviation of *DMP extensibility* for DMPs with equal archetype set size.

FIGURE 13. **DMP extensibility as a function of archetype set size.**

A. DESCRIPTION OF SPECIFIC APPLICATION REQUESTS

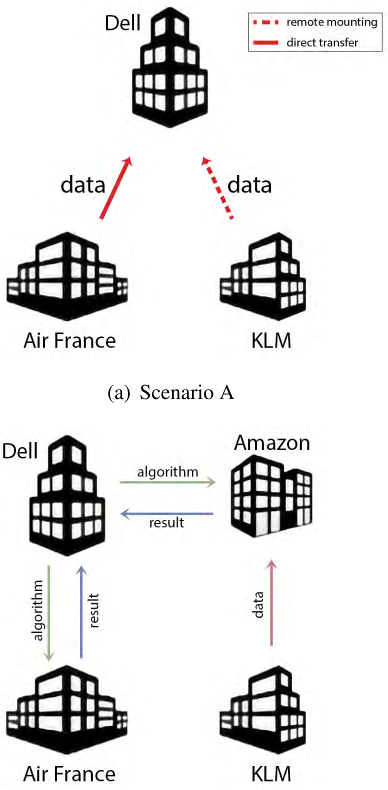
Two scenarios describe collaboration among Airline Companies. The involved parties are KLM, AirFrance, and Dell.

1) SCENARIO A

As illustrated in Figure 14(a), both AirFrance and KLM trust Dell in *data scope* and provide their aircraft data to it. Dell aggregates the data and performs its AI algorithm on it. However, KLM prefers sharing its data only by remote mounting and AirFrance allows the direct transfer, both of which are negotiable and belong to soft requests of this application.

2) SCENARIO B

This scenario is more complicated and is described in Figure 14(b). One data provider AirFrance does not trust Dell in *data scope* but Dell trusts it in *algorithm scope*. Dell first sends its AI algorithm to AirFrance, who would send the *intermediate result* back after operating on its local data. Another data provider KLM and Dell do not trust each other and agreed to use Amazon as a trusted 3rd party to perform



(a) Scenario A

(b) Scenario B

FIGURE 14. Two example application requests for a digital collaboration of airline companies in DL4LD.

the computation and the *intermediate result* is also sent back to Dell. Finally, Dell can merge the *intermediate results* from both sides and offer a prediction result. All the asset sharing is through direct transfer and no soft requests involve in this collaboration.

B. METRICS ANALYSIS WITH INTELLIGENT DMP SELECTION

In this section, we show a concrete example about how to choose a suitable DMP with specific application requests among competing DMPs with algorithm explained in Section V-F. The application requests are described in detail as scenarios A and B and available DMPs are shown in Table 4. The table describes each DMP with its supported archetype set.

TABLE 4. Available DMPs and its supported archetypes defined in DL4LD.

DMP	Supported Collaboration Archetypes
DMP_1	I, II, III, IV, VII
DMP_2	I, II, III, V, VII
DMP_3	I, II, III, V, VI
DMP_4	I, III, IV, V, VII
DMP_5	II, III, IV, VI, VII

1) METRIC ANALYSIS FOR SCENARIO A

Table 5 shows the proposed metrics of all DMPs for application request A. Rank those DMPs with *coverage* in descending order and no DMP achieves a full *precision*. Existence of soft requests contributes to a non-zero *flexibility*, which is a pre-condition for calculating *application extensibility*. A positive *application extensibility* indicates that a perfect matched archetype, for the specific request, can be provided by the DMP by modifying the application. Finally, DMP_1 is selected as optimum for this specific scenario.

TABLE 5. Metrics evaluation of various DMPs for scenario A.

	DMP_1	DMP_2	DMP_3	DMP_4	DMP_5
Coverage (1e−12)	4.29	4.28	4.26	3.69	3.65
Precision	0.83	0.83	0.83	0.83	-0.67
Flexibility	0.06	0.06	0.06	0.06	0.06
App extensibility	0.5	0.5	0.5	0.5	$-\infty$

TABLE 6. Metrics evaluation of various DMPs for scenario B.

	DMP_1	DMP_2	DMP_3	DMP_4	DMP_5
Coverage (1e−12)	4.29	4.28	4.26	3.69	3.65
Precision	0	0.17	0.33	0.17	0.33
Flexibility	0	0	0	0	0
App extensibility	–	–	–	–	–
Exact match after extension	F	F	T	F	T

2) METRIC ANALYSIS FOR SCENARIO B

The computed metrics of application request B for all available DMPs are shown in Table 6. Based on the value of *precision*, the fitness from those five DMPs to application request B is much lower than that of A. Since there is no soft requests, *flexibility* $= 0$. Consequently, metric *application extensibility* is invalid under this scenario. Then we further explore whether a perfect match can be achieved by archetype recombination. According to the last row in Table 6, DMP_3 is selected as optimum for the ability to offer an exact match and relatively higher *coverage*.

IX. RELATED WORK

DMPs are found in the literature to primarily describe specific online platforms that enable transactions among participating parties [11]. A very well known example is Airbnb [12], which is focused on putting peers, i.e. homeowners and short term renters, in contact. Business to business (B2B) platforms also relies on DMPs to create additional value for participating parties [13], [14].

The common approach to a DMP is that of a platform whereby the DMP provider becomes a trusted party [15]. This model entails that data and algorithms have to move to a secure trusted location provided by the provider. Our model of a DMP is a distributed model where autonomous parties build trust relations between them and move data and algorithms accordingly.

Reference [16] defines DMP as a platform coordinating supply and demand of digital products, a collection of data containing specific information, among providers and consumers. They define a distributed business process model and corresponding supported P2P based network [17]. But no work is involved in linking digital agreement with digital infrastructures.

Our work is generically focused on modeling collaborations in DMPs and defining fundamental building blocks in such architectures. This is the first comprehensive step, to the best of our knowledge, towards a systematic description of DMPs.

Toward this general definition of DMPs, we built upon concepts that have been explored before, also in our research group. The two main concepts we adopt are trust, and derived from trust policies.

Reference [18] has been the first to identify the need for a thorough and comprehensive definition of trust among participants in the marketplace. They also saw the trust as the starting point for the whole chain of resource and services authorization among parties. Subsequent work has further elaborated this concept, as we can see in [19]. We use this idea of trust as the underlying mechanism that allows us to model collaboration across scopes.

Trust is indeed the starting element to create actionable policies. Policy-driven systems are well known in the literature [20], [21]. In the work we presented here we do not cover the implementation choices needed to translate the collaboration models into actual components, software and hardware, in the DMP. This is the focus of ongoing work.

X. CONCLUSION AND FUTURE WORK

In this paper, we presented a model for describing DMP capabilities, which in turn express the underlying collaboration relationships between participating parties. Our model opens up a number of novel approaches to tackle a still unresolved problem: how to map applications into such policy-driven infrastructures.

Traditionally, applications are described as work flows and pipelines which describe an application as a composition of smaller tasks with their control and data inter-dependencies. The DMP brings an additional component to applications which is the application archetype i.e. the transaction flow between parties that needs to take place for the application to successfully run and adhere to the policies.

We showed that if the DMP collaboration archetype and the application request are consistently described we can map them together. This mapping allows us to identify the closeness of requests, i.e. the application, and the offered infrastructure, i.e. the DMP. We showed that the evaluation and comparison of competing DMPs are allowed and supported by having consistent and generic metrics, namely *coverage*, *extensibility*, *precision* and *flexibility*.

We applied our model and metrics to a specific use case, DL4LD, to illustrate how these methodologies are applied to in the real world. One concrete example is to allow for an intelligent selection of DMPs under specific scenarios.

There are many more directions to explore with our work in the future. Despite the compatibility between DMP archetypes and an application request, which is the main focus of current work, we can also consider other factors, e.g. achievable security level and performance cost, to facilitate a multi-criteria decision making of available archetypes for a specific application scenario. Another attractive research topic might be the risk minimization of DMP applications. We can investigate how to identify risks of a specific archetype generically and what monitors would be needed to create barriers around risks.

ACKNOWLEDGMENT

This paper builds upon the work done within the Dutch NWO Research project "Data Logistics for Logistics Data" (DL4LD, www.dl4ld.net).

REFERENCES

[1] S. Sagiroglu and D. Sinanc, "Big data: A review," in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, May 2013, pp. 42–47.

[2] Y. Demchenko, P. Grosso, C. De Laat, and P. Membrey, "Addressing big data issues in scientific data infrastructure," in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, May 2013, pp. 48–55.

[3] S. Dalmolen, H. Bastiaansen, H. Moonen, W. Hofman, M. Punter, and E. Cornelisse, "Trust in a multi-tenant, logistics, data sharing infrastructure: Opportunities for blockchain technology," in *Proc. 5th Int. Physical Internet Conf. (IPIC)*, 2018, pp. 299–309.

[4] C. Clifton, M. Kantarcioğlu, A. Doan, G. Schadow, J. Vaidya, A. Elmagarmid, and D. Suciu, "Privacy-preserving data integration and sharing," in *Proc. 9th ACM SIGMOD Workshop Res.Issues Data Mining Knowl. Discovery (DMKD)*, 2004, pp. 19–26. [Online]. Available: http://doi.acm.org/10.1145/1008694.1008698

[5] S. Liebowitz, "Rethinking the networked economy: The true forces driving the digital marketplace," in *Proc. AMACOM Division Amer. Manage. Assoc.*, Dallas, TX, USA, 2002, pp. 69–73.

[6] A. Zerdick, K. Schrape, A. Artope, K. Goldhammer, U. T. Lange, E. Vierkant, E. Lopez-Escobar, and R. Silverstone, *E-conomics: Strategies for the Digital Marketplace*. Berlin, Germany: Springer, 2013.

[7] S. Prabhakaran, S. Raman, J. E. Vogt, and V. Roth, "Automatic model selection in archetype analysis," in *Pattern Recognition*. Berlin, Germany: Springer, 2012, pp. 458–467. [Online]. Available: http://link.springer.com/10.1007/978-3-642-32717-9_46

[8] A. Jøsang, E. Gray, and M. Kinateder, "Simplification and analysis of transitive trust networks," *Web Intell. Agent Syst., Int. J.*, vol. 4, no. 2, pp. 139–161, 2006.

[9] M. Norouzi, D. J. Fleet, and R. R. Salakhutdinov, "Hamming distance metric learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1061–1069.

[10] TD Consortium. (2018). *Data Logistics for Logistics Data*. [Online]. Available: https://www.dl4ld.net/

[11] S. Cisneros-Cabrera, A. Ramzan, P. Sampaio, and N. Mehandjiev, "Digital marketplaces for industry 4.0: A survey and gap analysis," in *Proc. Workshop Conf. Virtual Enterprises*. Vicenza, Italy: Springer, 2017, pp. 18–27.

[12] A. Fradkin, "Search, matching, and the role of digital marketplace design in enabling trade: Evidence from airbnb," *SSRN Electron. J.*, Apr. 2017. doi: 10.2139/ssrn.2939084.

[13] D. Zahay, D. Schultz, and A. Kumar, "Reimagining branding for the new B2B digital marketplace," *J. Brand Strategy*, vol. 3, no. 4, pp. 357–372, 2015.

[14] A. Ordanini and A. Pol, "Infomediation and competitive advantage in B2B digital marketplaces," *Eur. Manage. J.*, vol. 19, no. 3, pp. 276–285, 2001.

[15] M. Schoop and T. List, "To monitor or not to monitor-the role of trusted third parties in electronic marketplaces," in *Information Age Economy*. Heidelberg, Germany: Springer, 2001, pp. 605–618.

[16] M. Schmees, "Distributed digital commerce," in *Proc. 5th Int. Conf. Electron. Commerce*, 2003, pp. 131–137.

[17] H. Tran, M. Hitchens, V. Varadharajan, and P. Watters, "A trust based access control framework for P2P file-sharing systems," in *Proc. 38th Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 2005, p. 302c.

[18] L. Gommans, J. Vollbrecht, B. Gommans-de Bruijn, and C. de Laat, "The service provider group framework: A framework for arranging trust and power to facilitate authorization of network services," *Future Gener. Comput. Syst.*, vol. 45, pp. 176–192, Apr. 2015.

[19] A. Deljoo, T. van Engers, R. Koning, L. Gommans, and C. de Laat, "Towards trustworthy information sharing by creating cyber security alliances," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 1506–1510.

[20] M. Sloman, "Policy driven management for distributed systems," *J. Netw. Syst. Manage.*, vol. 2, no. 4, pp. 333–360, 1994.

[21] C. Bennett, E. Esseiva, T. Kol, and R. Stevens, "Policy driven access to electronic healthcare records," U.S. Patent 11/316 262, Jun. 28, 2007.

LU ZHANG received the B.Sc. degree from Shandong University, China, and the M.Sc. degree from RWTH Aachen University, Germany. She is currently pursuing the Ph.D. degree with the Systems and Networking Lab (SNE), University of Amsterdam. Her research interests include information security, container networks, and novel networking infrastructures.

REGINALD CUSHING is currently a Postdoctoral Researcher with the Systems and Networking Lab (SNE), University of Amsterdam. His research interests include distributed computing, computing paradigms, programmable infrastructures, and alternative computing. He is also involved in PROCESS and DL4LD projects.

LEON GOMMANS received the Ph.D. degree in computer science from the University of Amsterdam, in 2014, after defending his thesis on multi-domain authorization for e-Infrastructures. After completing his Ph.D. degree as a Guest Researcher at the University of Amsterdam, he became Science Officer at the Air France KLM IT Technology Office, where he brings academic research alongside business use cases to unlock new value and opportunities. His current research interests include secure, fair, and trusted data sharing in B2B context, using the concept of digital data marketplace. Data driven aircraft maintenance represents a main use-case.

CEES DE LAAT currently chairs the System and Network Engineering (SNE) Lab, Faculty of Science, Informatics Institute, University of Amsterdam. He serves with the Lawrence Berkeley Laboratory Policy Board for ESnet, is a Co-Founder of the Global Lambda Integrated Facility (GLIF), the Founder of GRIDforum.nl, and a Founding Member of CineGrid.org. His group has been a part of a.o. EU projects GN4-2, SWITCH, CYCLONE, ENVRIplus and ENVRI, Geysers, NOVI, NEXTGRID, and EGEE and national projects DL4LD, SARNET, COMMIT, GIGAport, and VL-e. He is also a member of the Advisory Board of Internet Society Netherlands and Scientific Technical Advisory Board of SURF Netherlands. More information available at: http://delaat.net/.

PAOLA GROSSO is currently an Associate Professor with the Systems and Networking Lab (SNE), University of Amsterdam. She is the Coordinator and the Lead Researcher of all the group activities in the field of multi-scale networks and systems. Her research interests include the creation of sustainable e-Infrastructures, relying on the provisioning, and design of programmable networks. She currently participates in several national projects, such as SARNET, DL4LD, EPI, and SecConNet and in EU H2020-funded projects, such as FED4FIRE+, GN4., and ENVRIPLUS. More information available at: https://staff.fnwi.uva.nl/p.grosso/.