QUALIFIED DATA EXCHANGE

Preprint · February 2023

DOI: 10.13140/RG.2.2.28753.81769

1 author:

Rieks Joosten
TNO
39 PUBLICATIONS **72** CITATIONS

SEE PROFILE

QUALIFIED DATA EXCHANGE

An Introduction

January 2023

TNO innovation for life

Rieks Joosten

Table of contents

1 Introduction 4
1.1 Getting a handle on complexity 5
1.2 Another starting point 6
1.3 Furnishing work from this premise 7
1.4 Looking through different glasses 8

2 Qualified Data Exchange (QDX) 12
2.1 QDX Governance 14
2.2 QDX Management 15
2.3 Policies Management 15
2.4 QDX marketplace 17
2.5 QDX matching 17
2.6 Operational data exchange 18

3 Reflection 20

1 Introduction

For example, those engaged in data sharing do so in order to have more/better information, which may come from different sources.[1] The idea is that this gives you more (or better) insight, which enables you to make better decisions, act more appropriately, work more efficiently/effectively, innovate, form communities, and so on. Others share data because there is a weighty societal interest involved, or because laws and regulations require it.[2]

An example: The Municipal Debt Relief Act (Wgs) requires all municipalities in the Netherlands to be able to receive signals about payment arrears from parties providing a critical service to citizens, so that a decision can be made on this basis whether to invite a resident for a discussion about their debts and the possibilities for assistance in this regard. This means that all "fixed charge partners" (water companies, energy suppliers, housing corporations, etc.) should be able to share data with all municipalities in the Netherlands where they have customers. Figure 1 (on the next page) shows a (very) small part of this network.

The figure shows that each of the (more than 300) individual municipalities may receive signals from one or more water companies, housing corporations or other landlords, energy and/or heat suppliers, etc. There are more than 300 parties in the Netherlands that (should) provide such signals[3] . That amounts to hundreds of relationships. However, what the figure does not show is that there is also dynamic complexity: over time, signal providers will be added, municipalities may be merged, and so on. What the figure also fails to show is that there are many other situations in which data must be shared between these and also other parties, or better yet, between different departments or business units. Each of these has its own data needs, and establishes (and maintains) relationships with various other parties. And because these departments also have to comply with different legal or otherwise regulatory frameworks for different tasks, the complexity on the shop floor is even many times greater than you might expect at first glance.

[1] Digicampus (2020). Government as a partner in data sharing. See also: NLAIC, building block data sharing.
[2] NVB, bank-and-data.
[3] there are at least 30+ energy suppliers, 10 water utilities, over 300 housing associations (I'm sure there are more landlords), 11 health insurance companies (each with one or more labels). Source: Internet

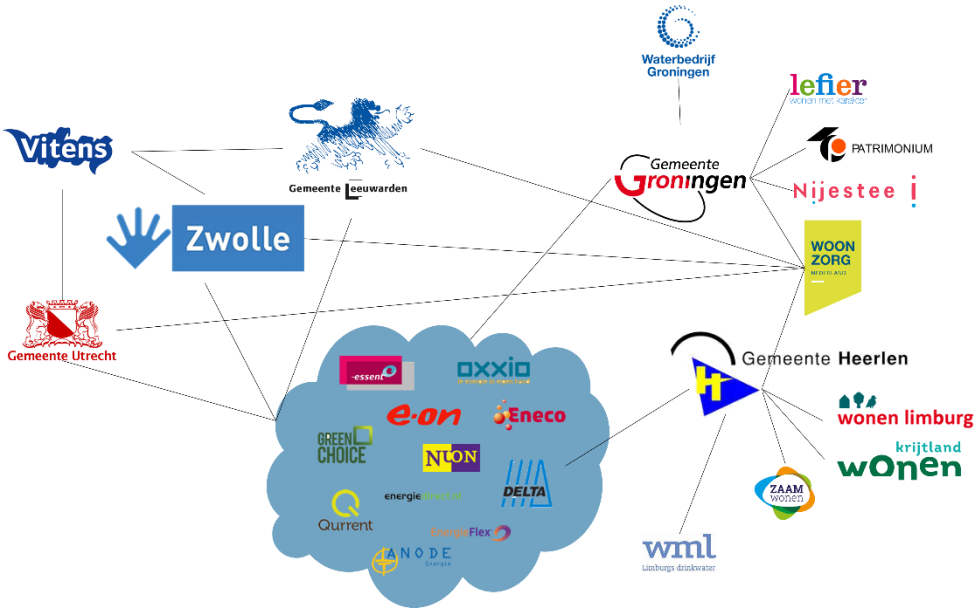

Figure 1: Some relationships between parties that exist to comply with Art. 3 Wgs.

1.1 Getting a handle on complexity

Wanting to have a picture or overview of (departments of) organizations, their mutual relations, the data they exchange, and so on, is often inspired by the desire to get a grip on this. After all, it can then be explained why data is requested, what it is, and what is being done with it. This in turn is necessary to demonstrate that work is being done within the set legal or otherwise mandatory frameworks. Such a picture is usually constructed with a top-down approach: tasks are derived from laws and regulations, which must be assigned to organizations. Task objectives, in turn, consist of subtasks, which are assigned to organizational units (departments) - sometimes also from other organizations. This continues until they are finally assigned to (functional) "roles" within an organization (part), or individual officers. This detailing, which is a combination of mandating and delegating, involves more than just assigning tasks (packages) to performers. These types of assignments also include rules within which a those tasks are to be performed, what may (or may not) be done in the process. As an overview shows more of this kind of detail, it becomes more and more complex, until eventually it is no longer an overview, but just a cluttered mess. It is never really complete, consistent and/or coherent, current, and therefore not really useful. The desire to get a grip on that situation does not become a reality with this.

We also see that in some domains (for example, healthcare, or information security) frameworks are being established, or following standards (such as ISO 9001 or ISO 27001) are being made mandatory. Again, this is often done to get a better handle on the complexity of these types of data exchanges. These high-overhead frameworks are often fleshed out in manuals or other tools intended to be used on the shop floor. And while they often contain very useful things, we see that on the shop floor the amount of such frameworks and tools is also quite large. To keep things manageable, the shop floor has to choose which, or what of them, to use (and what not to use). There are no uniform criteria for determining this, so one chooses what is workable and what is defensible. Whether that then also means that one has a grip on the complexity is very questionable.

1.2 Another starting point

We suggest a different starting point and one that is based on achieving concrete results on the shop floor that are known who will use them and for which they must then be fit for purpose. In black and white terms, this would mean that on the shop floor, activities are only carried out if the implementers know what concrete results are to be delivered that are 'fit for purpose', i.e. suitable for what will be done with them by the users of those results. In practice, the soup will never be eaten so hot, but the reverse - just doing what one thinks is necessary without thinking too much about it - can easily lead to routines that are ultimately inefficient or otherwise undesirable.

By way of example, let's imagine a municipality (official) whose duties include making decisions about whether or not to invite citizens to a meeting about debt assistance, as referred to in Article 3, paragraph 1b Wgs. To carry out this task (and make such decisions), it needs data. Each municipality is autonomous when it comes to determining on the basis of what data their officials are expected to make such decisions. Which choice they make also has consequences, by the way: wrong choices can lead to complaints (for example, about violating someone's privacy, or being treated unequally), or making the wrong decision (someone is not invited who should be eligible).

The autonomy that parties have when it comes to making decisions and collecting data for them implies (in theory) that they can make this very simple (for example, by rolling a die), or very complicated (having taken into account everything that could go wrong). In practice, it means that a party strives to make it as simple as possible, while covering the most important risks. Therefore, the data requested serve both to determine exactly what is being decided and to cover risks. Because risks are constantly changing (for example, compliance with ever-changing laws and regulations), the data requested for a certain decision will also change from time to time.

An important risk that must be covered is that of 'invalid' (invalid) data. A piece of data is 'valid' not only if it has the correct meaning (in IT we are talking about 'syntax and semantics'), but also if its truth can be trusted. This trust need not be absolute: if the risks involved in making an invalid decision are negligible, then it does not matter so much. We see this, for example, when buying alcoholic beverages online: the user must check that he is holder over 18 years of age. The meaning of such a checkmark is clear, but whether its truth can really be trusted is very questionable. But as long as it has no adverse consequences for the webshop, it does not matter. It is quite different with a mortgage loan. If the mortgage lender does not properly verify the truth of the applicant's being over 18, the transaction can be legally invalidated with adverse consequences.

Getting a grip on complexity does not mean keeping an overview of it. It is enough if for every activity to be performed on the shop floor, it is known what the result should be, who will use that result, and what he will (or should be able to) do with it next. This applies to collecting data as much as it does to making decisions, or participating in meetings, and so on. This can be "decentralized" to be monitored, evaluated, and if necessary, corrected.

In determining what kind of data is needed for a particular task, we choose as a starting point what implementers, on the shop floor, need to perform that task. From this starting point, we can then determine whether such data are available, where they might come from, and with what assurances they are qualified to be "valid" to be used within the task at hand.

1.3 Furnishing work from this premise

This change of perspective (from top-down overviews to "decentralized" bottom-up looking at what is needed) enables organizations to achieve their goals by explicitly linking them to concrete activities with outcomes suitable for doing what they are intended to do with them.

Suppose a municipality aims to "comply with Art 3, paragraph 1b Wgs". This means that one or more officials are given the task of collecting data in such a way that it can be determined which of that municipality's residents may be eligible for debt assistance, and to invite them for an initial interview about it. The data must therefore be 'valid' for this purpose.[4] The municipality must then determine what these data are, where they must or may come from, and how to determine whether they are valid for the intended purpose. This will be different for each municipality; not only because housing associations, water utilities, etc. may differ from one municipality to another, but also because there are (unique) local initiatives[5] that exist in one municipality and not in others.

A translation into implementation will then have to be made within the municipality. This consists of choosing the communication channels that may (or should) be used for data collection, and determining the set of "actors" (people or devices) that request, collect and validate the data through these channels. Attention to communication channels is necessary because each type of channel[6] has its own mechanisms for obtaining certain assurances needed to determine data validity. Sending data through the mail provides little assurance about its sender, for example, or its timeliness. Had the same data been sent electronically, there may be more certainties associated with that: for example, a sender can be determined with more certainty if there is a digital signature under the data, or the data is sent over an SSL connection. The assurances associated with a particular communication channel (and associated method) may be provided by technical measures, but may also be of a legal nature, or follow from a system of agreements to which the communicating parties have committed themselves.

Once the communication channels and actors through which data can be obtained have been identified, it may also be necessary to establish 'policies' for obtaining and validating the data. By a 'policy' we mean a set of rules, work instructions and/or other guidance intended for a specific type of actor (employees of a certain department or with a certain function, or certain IT systems) and for performing a specific task (i.e.: coherent set of actions). The idea is that when such an actor performs (part of) such a task, the actor has this policy at his disposal, he can read and (correctly) interpret it, and thus perform the task in the manner intended by the organization.

Finally, it will be necessary to make (and keep) the communication channels themselves available, and to ensure that there are enough human and non-human actors to do the actual work. Of course, individual communication channels and actors can be used for more than one task. Creating and maintaining such a "mapping" can help to efficiently allocate the people and resources that organizations use to do the work that leads to achieving their goals.

The above text, written from the position of the data processor, applies equally to setting up the work for data providers. An organization must also begin to determine what (types of) data it wants to be able to provide, and create an offer for that purpose. Such an offer describes not only the syntax and semantics of the data itself, but also

---

[4] And, to keep it organized, you also need to prioritize. Not every signal may be equally important, and some data is easier to obtain than others.

[5] Example: 'WIJ teams' we find (on the Internet) only in Groningen and Eindhoven.

[6] By a communication channel, we mean the set of means and activities that (can) be used to send data from a sender to an addressee, and exchange meta-data about it. The latter involves, for example, requesting/sending an acknowledgement of receipt, or proof that the data is still valid (not revoked or withdrawn).

other properties (which ones, of course, the organization gets to decide). It may be a description of how the data was created (e.g., through a KYC process, or as a result from a certified (technical, or administrative) process), what qualifications its implementer(s) had, and so on. These kinds of descriptions are necessary for other parties if they are going to determine whether they will be able to use this data (and whether it is valid) for their purpose(s).

In addition, the organization will need to establish (and describe in an offer) practical matters, such as through which communication channels the data will be made available (and at what "address" of such a channel this will be done), and what conditions must be met in order for a request for the provision of such data to be processed. This enables one's own organization, as well as other parties who want to be able to request such data, to make the policies for the actors who will do the associated operational work, and also to organize that there are sufficient (and adequately qualified) people and resources who can perform these tasks.

1.4 Looking through different glasses

We can describe this (shop floor) perspective a bit more tightly (more formally). This can help architects, process designers and the like in advising/supporting management when they are faced with the task of determining which data are needed for which of their goals, how to operationally determine that they are valid, through which communication channels data can be obtained and/or delivered, which actors will be tasked with this and what qualifications they must then meet. A more formal description makes it possible to create concrete lists of requirements, wishes, "mappings" and the like that can facilitate the actual work within an organization. By later comparing such lists from different parties, perhaps patterns can be found that are useful for setting up (more generic) facilities that are not separate from what is needed on the shop floor. [7]

However, designing, implementing and managing information processes and associated data exchanges from the "shop floor" perspective does not only have benefits. It can also cause (sometimes intense) feelings of discomfort, uncertainty, despondency, anxiety and the like among designers and implementers. It involves not only a different approach, but also a different way of thinking. It is similar to putting on a new pair of glasses, where the 'old glasses' are the current way of thinking, and the new glasses are the model of thinking that we will summarize in the next chapter[8] . 'Putting on new glasses' means that first the old glasses have to be taken off, i.e. you have to temporarily(!) put aside the way you are used to looking at data sharing and actually start using this new thinking model. That takes getting used to. But the habituation effects disappear again once you get used to the new way of thinking and figure out how it works.[9] And then you can weigh the pros and cons and decide if, or when, you are going to use these 'new glasses'.

The new thinking model on data sharing we describe below consists of a piece of terminology[10] and a description of their interrelationships. The words in bold in this document are terms we explicitly define here.[11]

---

[7] With a top-down approach, we often see more of a tendency toward furnishing "one size fits all" solutions, which can cause quite a bit of distress in the workplace - especially if this includes people.

[8] Those interested in learning more about this should refer to the eSSIF Lab Parties, Actors and Actors model.

[9] Habituation effects when changing perspectives (theories) are well known in history, for example, in the transition from a geocentric to heliocentric worldview, from Newtonian to relativistic and quantum mechanics, and still, for example, in fathoming (visual) illusions.

[10] We do so by providing criteria for each term that the reader can use to determine whether something qualifies as (instance of) that term, in order to minimize misunderstandings.

[11] These, and related terms, are described in detail in the eSSIF-Lab framework (and glossary).

We call an **entity** (i.e.: something that exists) a **party** if it sets its own goals, maintains its own knowledge, uses that knowledge to achieve those goals, and does so all in an autonomous (a sovereign) way. These are typically people and organizations. We call an entity an **actor if** it can perform actions. Typical examples are people and machines (computers). An **action** is any unit of work performed within a certain context by one actor, on behalf of one party as a single (indivisible) operation[12] . An example is signing a letter. We say that the actor who performs the act on behalf of the party does so in the role of **agent** (for that party); the party in that context performs the role of **principal** for the actor.[13]
Organizations cannot perform acts and therefore do not qualify as actors. For instance, TNO cannot sign a contract or hire an employee - for that TNO needs an actor who performs such actions on behalf of TNO, such as a person who is a member of the Board of Management, or one who performs the role of HR employee within TNO. By the way, it is best to keep using the common language in which organizations simply perform actions (as in: "TNO hired 5 employees today"). We must then realize that the *actual* meaning is that there is an actor who performs this action on behalf of the organization[14] .

People qualify not only as actors (after all, they can perform acts) but also as parties (after all, they have their own goals, maintain their own knowledge, and so on). When a person performs an action, they can do so on behalf of themselves (they are then their own agent and principal), but also (in the role of agent) on behalf of another party, for example, their employer.[15]

Because each action is performed on behalf of one party, that party also determines the rules according to which an actor must perform that action. These are laid down in (detailed) work instructions, (high-over) policies, and other kinds of artifacts that we commonly call "**policies**," the content of which belongs to the knowledge of the party establishing them. Policies deal, for example, with how to make a certain decision, what data are needed to do so, when something is 'true,' under what conditions those data are **valid**, i.e. lead to a valid/right decision, and so on. Actors are expected to know and follow the policies, at least insofar as they apply to the actions the actors (may) perform[16] . When drawing up policies, it is often assumed that the implementing actors already have certain knowledge (and skills), which are therefore not specified as yet. As an example, a work instruction on administering medication is not about how to administer a pill, drops, a syringe, etc., but more about the place, time and circumstances in which it should be done. It has been assumed that the people (actors) who do it know how to administer pills, drops, or a syringe themselves. A robot or computer (actor) also possesses knowledge itself (its program code), and it is deployed according to the policies of the party on whose behalf this actor performs actions.

---

[12] See: "Practice Book for Process Architects," van Gorcum, 2002. This defines operations as the basic blocks from which processes and procedures are composed.
[13] An actor can perform different acts for different parties in a certain period of time, thus fulfilling the role of "agent" for different parties. However, the actor performing a single act does so as the agent of (exactly) one party.
[14] The process by which a party is given the opportunity to deploy a certain actor (human or otherwise) to perform (or be able to perform) certain actions is called onboarding.
[15] We reserve the term "employee" (of a party) for an actor who is onboarded by that party, i.e., that actor has been granted by that party the right (or duty) to perform certain types of acts on behalf of that party, and has been enabled by that party to do so.
[16] The "onboarding" of an actor by a party establishes its rights and obligations, and also arranges for the actor to have all necessary conditions, resources, etc.

This model of thinking sees parties as completely autonomous (sovereign) entities when it comes to their knowledge (goals, work rules, etc.). Thus it is not self-evident that they obey the law: they choose (consciously or unconsciously) if, or to what extent, they do so. That the model thereby models actual reality will be agreed by anyone who has at one time or another (consciously or unconsciously) chosen to drive too fast, or through a red light, or who remembers reports about organizations that have (again) "gone wrong.
Party autonomy is an important principle that we explicitly consider when it comes to data sharing. After all, it means that each party decides for itself what its mission and other goals are, what and how things are done, and what rules are followed or not. And also which other parties they interact with, what they do in them, and how.

For data sharing, this means that each party will have to determine for itself - on a technical, organizational and legal level - which data it needs for which specific purpose, from which source(s) such data must come, what their syntax and semantics are, what makes the data reliable, and what other properties they must have to be "valid" in order to be able and allowed to be used for that specific (set of) purpose(s). We elaborate on that further below.

Finally, we define the term '**role**' (of an entity - typically a party or an actor) as a set of characteristics that this entity has and/or actions that the entity is allowed to perform and/or pieces of knowledge that the entity possesses, in a certain context. We will describe the roles 'data consumer' and 'data provider' further on. However, there are other roles; some are defined in a law, others are only specified for use within a single party (for example, through its policies).

It is important not to confuse a "role" with the party or actor performing such a role (in a certain context). After all, in a different place or time, that party might fulfill a different role. A party offering data will, in order to decide whether or not it is going to comply with a request to do so, first fulfill the role of data consumer (through which it collects data in order to make that decision), and if that turns out to be positive, it will (in the role of data provider) start providing the requested data.

When we say that <role *name*> does something, we mean that an entity fulfilling the role <*role* name> at some point in time does the related thing. For example, if a data consumer requests data, and later a data provider provides data, these roles may be fulfilled by the same party. But they may also be fulfilled by different parties.

In summary, the new perspective consists mainly of replacing the terms "organization" and (the different variants of) "person" (natural person, legal person) with the terms "party" and "actor," which, however, have a very specific meaning with which we make a different distinction: a party is an autonomous entity that manages its own (subjective) knowledge, and an actor is an entity that can do things. When an actor does something, it always does so on behalf of a party, and that party then also provides the knowledge (policies) that the actor uses to perform the act(s) in the way this party has conceived. When we (still) say that a party performs an action (does something), we always mean that there is some actor who performs this action on behalf of this party.

2 Qualified Data Exchange (QDX)

There is a lot involved in data sharing and this is usually done from the perspective that data that is available somewhere should also be usable elsewhere[17] . The prevailing view is that syntax and semantics should be well established ('semantic interoperability') and that all kinds of ('trust') frameworks are going to help to actually start (re)using that data.

We are going to add the perspective of individual and autonomous data processors (data consumers) to this vision. From this point of view, (obtained) data must serve some purpose. Perhaps it serves to record a decision, or to perform some (other) action with it. It is then important that that data must be suitable (valid) for that purpose: after all, a decision based on invalid data may entail undesirable consequences (risks). This viewpoint emphasizes that a data processor must not only be able to **verify** the data it obtains, i.e. determine that the data has the intended syntax and semantics, but ALSO be able to **validate it**, i.e. determine whether it is valid for the purpose for which it intends to use it[18] .

Qualfied Data are data that meet all the conditions specified by the party requesting such data in order to establish that the (obtained) data are valid (valid) for (further) processing in order to realize a well-defined purpose. Conditions may relate to syntax and semantics, but also to the way the data were created, guarantees regarding provenance and/or integrity, etc.

Qualified Data Exchange (**QDX**) is a way of looking at data sharing with as its main starting points the autonomy (sovereignty) of all parties, and the from that (subjectively inferable) things like policies for task execution, data needs, validity criteria, and so on. QDX is a model that identifies from different perspectives (roles) what is involved in requesting Qualfied Data and offering data that (for certain parties and purposes) can count as Qualified Data. Figure 2 shows a (simplified) overview of this.

The right-hand column shows how QDX works from the perspective of a party who needs data to perform a certain type of operation (processing). He needs to know how to ask for it in such a way that the answers are verifiable and validatable. The actors doing the associated work need rules, work instructions, etc. for this (which we call here "policies"), so that in their operational context they know how to ask for, and validate, the right data. These policies are created and maintained earlier (design-time[19]) in the 'policies management' process for different opertional ('run-

[17] Here we look primarily from the perspective of data providers and/or data ecosystems.
[18] Checking syntax and semantics is necessary, but not sufficient. Depending on the intended purpose, it may also be necessary to know who determined this data (e.g., the government, a bank, a water utility), and/or how it was done (e.g., a real measurement, or an estimate), and/or certain (legal) rights can be derived from the data, etc.
[19] That is: in the preparatory work. That is part of activities belonging to (the risk management part of) process design and/or information modeling.

time'[20]) contexts[21] (different communication channels[22] and/or types of actors). The QDX governance process determines what processing can be done on behalf of the party, what data are needed to do so, through what communication channels they may be requested, and when they are valid for a certain processing.

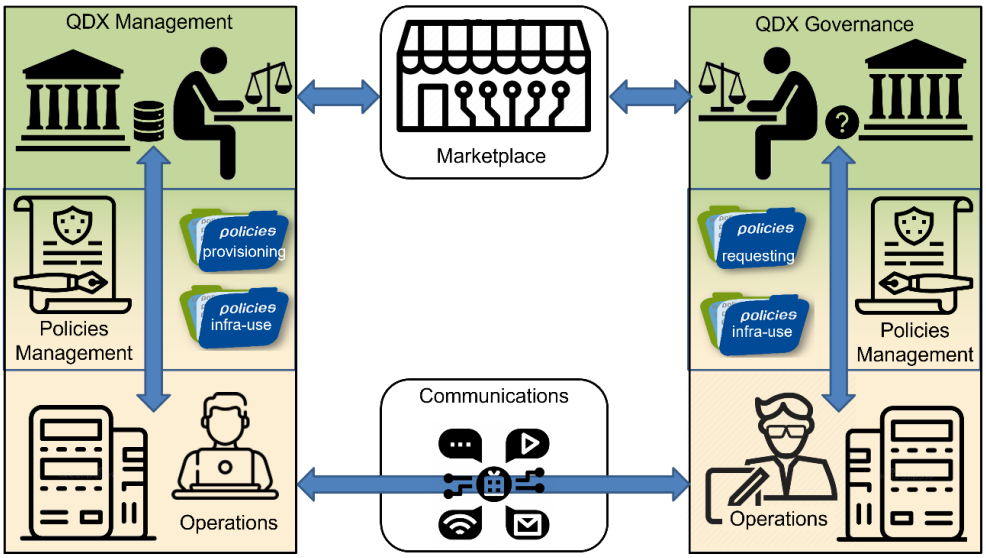

Figure 2 QDX Overview

The left column shows how QDX works from the perspective of a party that can provide data. Its QDX management process identifies what data that is, what the terms of delivery are, and what metadata is going to be published so that parties who could use such data can use that metadata to decide whether, and if so for what processing, they are going to use the offering. Again, these decisions should be translatable to the operational environment, i.e., into policies that can be used in the various operational contexts for data delivery by the relevant actors to be able to assess requests for the delivery of certain data, and if accepted, arrange for the data to be delivered.

At the top center, we see that demand (from the right column) and supply (from the left column) should be able to be matched. In fact, this is a functionality similar to a marketplace: the provider should be able to publish his offers, and the user should be able to find the offers he needs from them. It implies that providers must be able to advertise their offers in such a way that processors can determine whether, or for which operations they want to be able to perform with them, is suitable. It also implies that processors should be able to know where to find those advertisements so that they can set their policies.

---

[20] The term 'run-time' refers to the time in which actual operations/actions are performed. This is in contrast to "design-time," which refers to the time in which such operations/actions are specified.
[21] That is: in operational work, that is, when a single (real) decision is made in one specific situation by one specific actor.
[22] By a communication channel, we mean the set of means and activities that (can) be used to send data from a sender to an addressee, and exchange meta-data about it. The latter involves, for example, requesting/sending an acknowledgement of receipt, or proof that the data is still valid (not revoked or withdrawn).

Below, we see that actors (in the right column) can request data from actors in the left column. They all use the specific policies of the party on whose behalf they perform their respective task. That's what "data sovereignty" means. Down the middle, we also see that such actors, for the actual exchange, can use communication services provided by other parties. That could be, for example, a postal company (for physical exchanges), but also a network provider (for electronic exchanges), or something else.

In the following sections, we elaborate on some of these blocks.

2.1 QDX Governance

An actor (person or IT system) performing on behalf of a party (run-time[20]) performs an action that requires data, such as making a decision, must have knowledge of a policy showing what data is, and what validity criteria[23] belong to it. So this policy must have been previously (design-time) specified[24] and established[25] , and also made available in an adequate way to each actor performing actions of that kind. Design-time choices are thus made, and translated into (possibly several[26]) policies, that make the run-time exchange of (valid) data possible (or impossible).

By **QDX governance**, we mean the design-time process that a party performs, in which it specifies and establishes what kind of data are needed for what kind of operations (if they are performed on its behalf), and what validity criteria are associated with them. To that, the party will include the risks involved in performing that operation if data were to be used that is incorrect, does not relate to the correct "subject," is too dated, is unreliable (according to the judgment of the data consumer, of course), should not legally be used, etc. As a result, for each type of operation, the party specifies the frameworks for the data and collateral that are not only directly relevant to performing the operation, but also to validating data and mitigating risk.

These frameworks may also impose restrictions on the ways (communication channels) in which data are exchanged.
The result of the QDX governance process is a specification of the frameworks for obtaining data of interest to the party for the various processing operations the party performs with it in order to achieve its own goals. These frameworks contain all the information needed by the actors performing the 'Policies Management' process to translate them into the rules, work instructions and other artifacts that are then used by operational actors when collecting data, so that it is done in ways that fit within the frameworks set by the Party.

---

[23] That is: criteria (linked to a specific purpose) that can be used by a specific type of actor to determine whether a set of (supplied) data is valid to be used for that purpose.
[24] specifying is: writing down the content of the policy (clear, unambiguous, consistent, complete, etc.).
[25] To establish (a policy) is: to decide that the policy should actually be used.
[26] Different types of actors need different types of policies. For a human, a policy must be readable in the language the human is proficient in - international companies may need policies in multiple languages. For IT systems, "machine readable policies" are needed - files that can be used by these types of systems in such a way that they will perform the actions as (design-time) intended.

2.2 QDX Management

A party that possesses data and wishes to share it must ensure that it is made adequately available to parties that would/should/can use it. The data consumer will also need to disclose ("market") details about such an offer.

By **QDX management** we mean the process that a party carries out, in which it determines which (type of) data it wants to and will issue (syntax, semantics), under what conditions the data will be issued (for example, to whom, what they can do with it, etc.), what types of assurances are offered to customers in the process (for example, regarding the way in which the data was created, the qualifications of those who did so, or regarding its timeliness, and so on). It also establishes frameworks regarding operational delivery. For example, restrictions may be imposed on the communication channels to be used, which customers may or may not be supplied, and so on.

In doing so, the data provider will consider the risks involved in providing this data (in the various modalities), for example to parties who may not have/use this data, or if the law places restrictions on sharing it. He will also consider whether there is a business case for it: for some data, it may be possible to charge a fee; for others, it may be a legal obligation.

The result of the QDX management process is a specification of a party's frameworks for delivering data. These frameworks contain all the information needed to

- enable the actors performing the 'Policies Management' process to create and manage the rules, work instructions and other artifacts (policies), which are then used by operational actors to determine whether (a) a received request to supply data should be rejected, and (b) if such a request is honored, where the data can then be searched for and in what manner it may then be provided to the requester. This ensures that data delivery takes place only in ways that fit within the frameworks set by the Party.
- going through the process that results in what we call a **QDX advertisement** (for a certain set of data), i.e., a document that contains all the data a potential data buyer needs to be able to decide whether he can use the advertised data for one or more of his purposes, whether it is valid for that purpose, and also whether he can then make his own (user) policies with it. Therefore, a QDX advertisement contains not only the syntax and semantics of the offered data, but also information about collateral and delivery conditions, the communication channels used for that purpose, the 'addresses' or 'endpoints' to which requests should be sent, the structure of such requests, etc.

2.3 Policies Management

By Policies Management, we mean a process that serves to "translate" the frameworks established in a governance or management process into operational reality.

The scope of such a process is usually broader than just the collection or delivery of data: it is also used for the translation of management/ governance frameworks to other primary and secondary processes. In the figure, however, it is explicitly about the frameworks that relate to the delivery and/or gathering of data.

Looking through the new, more formal glasses, we assume that for each processing (delivery or collection) of data by a party, it has been established through which communication channels this is done, and which (classes of) actors in the operational work may and can perform certain tasks. Such an inventory in itself is not very complicated, but as the number of such processing increases it can be a lot of work to make this explicit, and also to keep it up to date.

In addition, two other inventories are needed, namely of the various:

- communication channels, and the properties they have that are relevant to policy making. For example, if data exchanges occur electronically,
    - through a network covered by the Telecom Law, then, for example, the confidentiality of communications is guaranteed
    - via an SSL connection, then confidentiality is guaranteed, for example, and there is also certainty about the party with whom communications are being made;
    - via an IDS connector, then on top of that are guarantees that the party being communicated with will adhere to a certain set of agreements;

  Similarly, non-electronic communication channels (such as sending data by mail, or by courier) may possess properties that are relevant to know when making data exchange policies.
- actors, and the various properties that data-sharing policy makers want to make use of when writing those policies. For example, for human actors, job requirements (aggregated in a functional role) may be important so that a policy can be created that says actors may only collect personal data if they can do so in accordance with the AVG[27] . Which (other) features are important will vary from party to party.
  Similarly, non-human actors (computers/applications) may have properties that may be important for performing data exchange operations. This will mainly concern whether, or to what extent, they can (technically) handle certain communication channels

Setting up these two inventories properly can be quite a job. After all, it is necessary to determine which properties to include in the inventories, and that in turn depends on what may be needed when writing the policies.

The maintenance of these inventories involves

- hiring or dismissing actors or (temporarily) suspending them. This in itself is not much work, but it does require a certain discipline on the part of administrators that is not always obvious.
- managing the properties of communication channels resp. actors that should have a place in the inventories. If from governance/management processes come decisions for which the policy writers need different/new properties, that will have to be documented in the inventories. However, the idea is that this is not going to happen very often.

Using these (we assume well-maintained) inventories, a QDX policy for a particular data delivery or data query can be translated relatively easily to the communication channels and actors doing the operational work. This, of course, does not alter the fact that any policy must be appropriate for the (type of) actor that has to work with it. If that is an IT application, then this policy will probably consist of program code, or as a configuration file. For human actors, it may be a work instruction that is stated in a natural language that the person in question has sufficient command of.

---

[27] General Data Protection Regulation.

The result of the Policies Management process (at least as far as QDX is concerned) is a set of policies, one for each combination of a (type of) actor and a (type of) communication channel, in such a way that these actors can operationally perform the tasks associated with supplying or retrieving data for a particular processing operation within the frameworks of the party on whose behalf they are doing so.

2.4 QDX marketplace

A QDX marketplace is a (physical or logical) place where parties (in their role as data providers) can advertise their data offerings, and (in their role as data consumers) can find out what data is being offered by other parties. What is typical of a QDX marketplace is that a data offer is not only about the type of data (syntax and semantics), but it also lists all kinds of other data that data consumers need to determine whether they can use the data in one or more of their data processing operations - i.e.: whether it is valid for such data processing.

We think that a QDX marketplace should ideally be situated within a 'community' (or ecosystem), i.e. within a group of parties that already have something to do with each other. After all, such communities are quite capable of making agreements in a relatively simple manner that offer guarantees that are important for data consumers (but also for data producers) to label data as 'valid' for certain processing. Then the simple fact that parties are members of such a community already provides important assurances, which may make obtaining even more assurances unnecessary, and thus greatly facilitates the exchange of data.

A QDX marketplace can be effectively realized as a platform and/or (online) catalog containing (all data of) QDX ads. Within a community, this place will be easily communicated so that parties can upload their ads to it, and view those of others. These catalogs will (for the time being) at least have to be readable by people, because they should be able to decide with the data from them whether they want to start using this kind of data (and thus have run-time retrieved). We see them as an extension to existing data catalogs, where the extension is that collateral and delivery conditions are also specified.

2.5 QDX matching

By **QDX matching** we mean all the actions taken by parties in their roles as data consumers and data providers during design-time to match supply and demand of data. This resembles 'semantic interoperability', where syntax and semantics of data are matched, leading to a standardized supply to which data consumers then conform (run-time). But here it is emphatically also about tuning what the associated certainties are that can or must be included, and how a data consumer can verify them to validate the 'normal' data.

Matching the supply and demand of data is conceptually the same as matching the supply and demand of any other product (or service). As such, there are various ways to shape it. For example, it can be done "remotely," where a data provider advertises its data ("products") and waits to see who is going to take them, and the data consumer somehow sees those ads come along (as "spam," or because he has searched for them) and then sees what he can use. The previously mentioned catalogs can play a useful role here.

In the context of data sharing, it is somewhat more nuanced, because it is not actually about the tangible ('tangible') data, but about the intangible ('intangible') information represented by this data. Each party is autonomous and self(standing) decides what data to use to represent certain information it knows. Terms like "reliable," or "true" will mean different things to different parties. And the mapping (i.e.: semantics) that the data provider chooses to use must not only be retrievable by the data consumer, but moreover, the mapped information (the concepts behind it) must fit into that data consumer's mental models. This is called: "semantic interoperability," and it requires a more intrusive way of tuning than is required, for example, between a supplier and consumer of ordinary products (a radio, or TV).

2.6 Operational data exchange

All of the foregoing serves to enable operational (run-time[20]) to exchange data. More concretely: a party that wants to perform (or have performed) an action for which certain data are needed, has in principle everything it needs to ask for these data, and to determine from the response to these questions whether these are the data it needs, and also whether they are valid or not. This applies to all combinations of contexts and actors (people, machines) made possible by such a party (design-time).

The same applies to a party that wants to perform (or have performed) acts that involve handling a request for the delivery of certain data (the result of which should be whether or not that request is honored), and - if such a request is honored - to collect the requested data and send it to the requested destination through the chosen communication channel.

Therefore, in order to actually request the data that is needed, and receive a response, at least one communication channel is needed that makes this possible. Parties will want to limit the number of communication channels, as each one involves setup and management costs. On the other hand, a party will want to make its data available in multiple ways to enable as many parties as possible to start taking it away as well. Furthermore, each exchange modality has its own unique properties, which can help to fulfill the validity criteria specified by data consumers (for certain data/purposes). We discussed this in some detail earlier (under "Policies Management").

3 Reflection

QDX is based on a number of principles that are not self-evident in practice. For example, we regularly encounter that when setting up an electronic data collection process it is thought that the first step is that 'the user must log in': after all, you need to know who you are dealing with. It is often tacitly or otherwise assumed that the user is also the party responsible for the accuracy, topicality, etc. of the data. This applies both to human 'users' and in situations where the data comes from another party's IT system (where that system is often authenticated through a PKI certificate). Such ideas often stem from procedural thinking; one imagines *how* the process works and that is then set up.

QDX requires staying focused on one's goals, the associated results, who is going to use those results and what they should be able to do with them next. That's more about the "what" than the "how. And then you almost can't help but think about what kind of data is needed for that, where it should come from, and what makes it valid to be used. It is then no longer obvious that the party actually providing the data is the same as the party that is the source (author) of the data. It is then conceivable for citizens to collect data about themselves that originate from a multitude of parties (their employer, the tax authorities, banks, and so on), to be shared when necessary with parties who need them.

It then also becomes easier to consider that certain data processing does not actually need to be done at all by the party that needs its results. This is especially true for (automatable) calculations or reasoning. There are techniques, such as Multi-Party Computation (MPC), in which (parts of) such calculations are performed by parties that possess certain data, and only the results are shared. In this way, calculations can be done (elsewhere) that a party could never have done independently, for example because the data used is too sensitive and therefore should never be shared.

QDX makes a clear distinction between parties (entities maintain their own subjective knowledge, and make decisions about what of it to share or what they need from others) and actors (entities that can do something - this is mainly about operational requesting, delivering and validating data). Making this distinction makes it easier to think of new forms of data delivery and data retrieval. It then no longer matters so much whether data comes directly from "the source," is delivered via an intermediary/intermediary, via a "vault" (whether in the cloud or not), as long as there are sufficient assurances - for example, about its source, its integrity, timeliness, reliability, etc. - that make the data valid for the purpose for which the data consumer wants to process it.

Although QDX makes it pretty clear what is involved for a single processing operation, for most parties it is true that they have to deal with many - often very many - processing operations. And what is clear for a single processing operation is no longer clear for such a multitude of operations. You have to be able to organize well to keep an overview, and that is no mean feat.
But you can also start collaborating. In practice, that often already happens: there are already a lot of communities whose members cooperate among themselves,

know who/how someone can be trusted, etc. That makes us think that those *communities* can also play an important role in organizing QDX-related work.

We call such an (existing) partnership a **QDX community** if (in addition to its existing goals) it also aims to start facilitating the idea of QDX for the participating organizations. This can be done, for example, by:

- listing the types of processing that many of the cooperating parties perform;
- specifying a (minimum/maximum) datasets (syntax and semantics) relevant to performing such processing;
- establishing (minimum/reasonable/maximum) validation criteria for these datasets;
- specifying qualifications for these datasets, especially those that facilitate evaluation of validation criteria, and how they will be communicated in QDX advertisements;
- Setting up a QDX marketplace in which all participating parties can place their QDX ads;
- selecting one or more communication channels that can (or should) be used within the collaboration for mutual data exchanges;
- Establishing (minimum/reasonable/maximum) qualification standards for actors performing tasks in providing or consuming data;
- writing the policies for the combinations of (agreed upon) communication channels and actors (meeting certain qualification standards) so that they can easily exchange the "standardized" types of data.
- Establishing a "trust framework" appropriate to the needs and capabilities of members to make it as easy as possible operationally to determine which data (suppliers) are trustworthy, with which parties data may be shared, and so on - whatever is of interest to members of the partnership.
- and so on.

The idea of communities is increasingly used, and is then known as a "data ecosystem" or a "data space. This usually involves establishing governance over an infrastructure for exchanging data between members (and possibly non-members), for which there are already many options, such as IDS, iShare, SSI, Gaia-X, FIWARE, etc.

One of the first (successful) data spaces is the Smart Connected Supplier Network (SCSN)[28] , which provides data exchange between parties in the manufacturing industry. This is an example based on the IDS infrastructure. The European Commission is encouraging the creation of data spaces, and we expect the numbers to grow significantly in the coming years.[29]
The importance of a focus such as QDX places on achieving goals/results, and in particular on establishing the validity of the data needed to achieve them, will only increase. For example, the idea behind "Explainable AI" is actually just that: establishing the validity of AI outcomes for use in specific processing. The recently made public ChatGPT, and especially how it was immediately used and abused, only underscores this importance.

We expect that both governments and private, commercial parties will benefit from a way of looking at data sharing that is based primarily on the "what" (rather than the "how"), and therefore better focuses on these kinds of concerns and addresses

[28] See https://smart-connected.nl/nl. Their processes are also available online.
[29] See European Commission - Shaping Europe's digiatl future: Staff working document on data spaces.

them. This will not solve all the problems, but it will provide a new perspective on data sharing.

For (the parties involved in) the ELSA Poverty and Debt Lab, such a new perspective can also provide new inspiration for the (re)design of processes. We expect a party that starts looking at its own processes from this perspective, and sets up the corresponding QDX processes, to get a better grip on its own data management, to be better able to fulfill the roles of data consumer and data supplier, and where necessary to account for the choices made. This better control over its own processes will therefore lead to more optimal service delivery.