Government Transparency Institute

Public procurement data processing

Government Transparency Institute

Version 1.0

June 2025, Budapest, Hungary

Government Transparency Institute

Summary

This report provides a high-level summary of the main steps of public procurement data processing applied by the Government Transparency Institute (GTI) to European public procurement data in line with its Global Public Procurement Dataset (GPPD)[1]. The data processing steps build on a set of algorithms originally developed in the EU-funded Horizon 2020 DIGIWHIST project, which were further extended and improved by GTI in more recent years.[2] Data on public tenders are published in various formats across the globe. Despite significant efforts to increase transparency in the field[3], analysis-ready datasets that capture all the key aspects of tendering practices and outcomes still require a significant data collection and cleaning effort. Some sources publish procurement notices daily, such as contract notices, awards, and modifications at Tenders Electronic Daily (TED). Others publish structured data (CSVs) consolidated at the tender level monthly (e.g., Italy), or provide API access (e.g. Moldova). As a result, these administrative datasets are in various formats and quality across countries and hence require domain knowledge of each data source and an elaborate set of algorithms that crate a cross-country comparable dataset.

The life cycle of the data ingested into the data processing pipeline are described in five steps: scraping, parsing, cleaning, matching, and mastering. This document does not provide a comprehensive description of the entire process; instead, it highlights the key steps applied to the collected public procurement data to create standardized and cross-country comparable datasets.

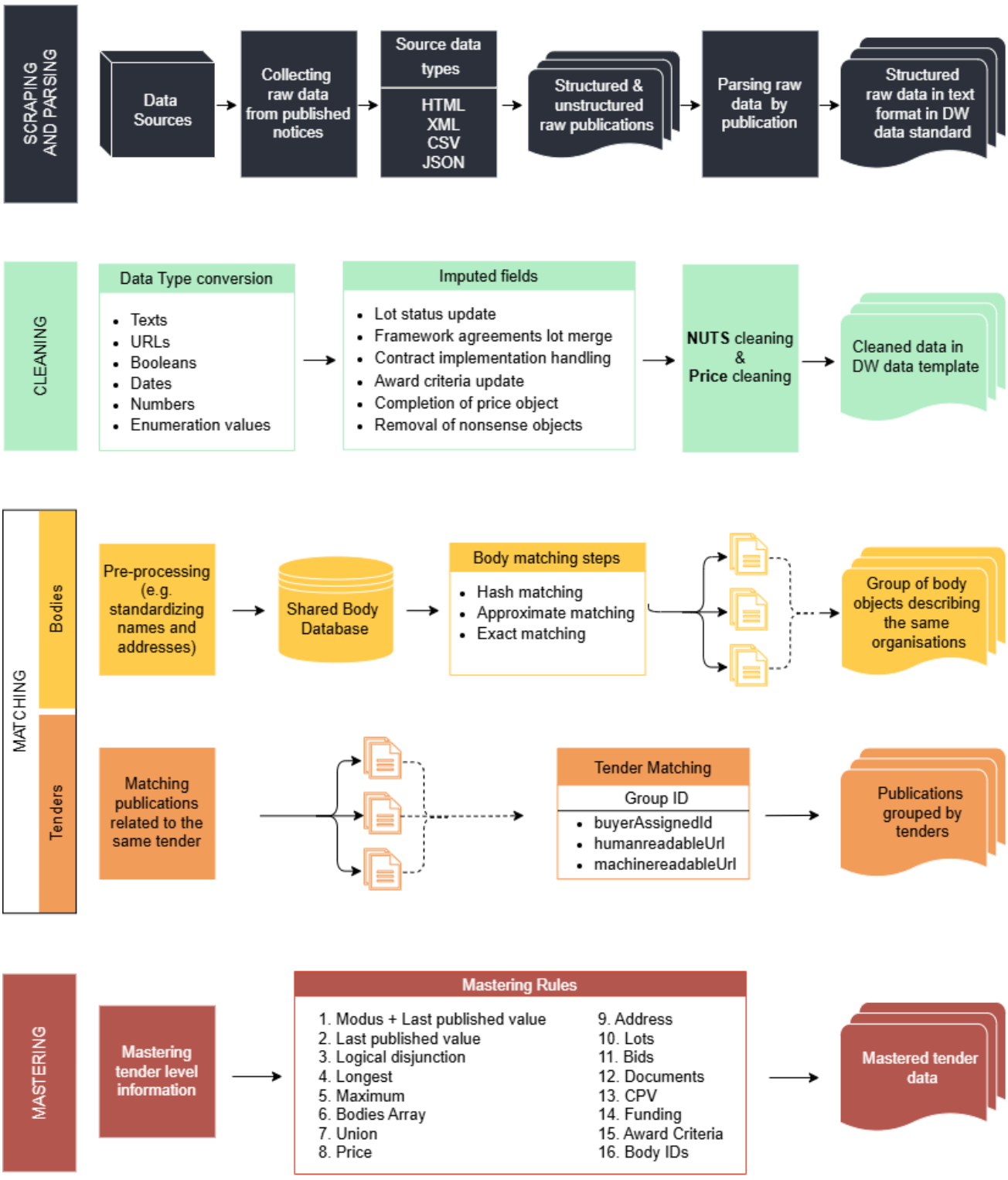
[1] https://doi.org/10.1016/j.dib.2024.110412

[2] This work was enabled by the generous and long-standing support by a range of funders, in particular the Open Society University Network, the Open Society Foundations, the EEA and Norway Grants Fund for Regional Cooperation, the EU's Horizon-2020 funding, and the UK Foreign, Commonwealth and Development Office's Anti-corruption Evidence Programme. In addition, the Government Transparency Institute has devoted its own funds to the maintenance and improvement of the datasets since 2018. We wish to extend our gratitude to all the programmers and research staff who have contributed to these datasets over the past decade, with special acknowledgement of those at Datlab and Precognox.

[3] Over the past decade, the push for greater transparency and data quality in public procurement has led many countries to adopt Electronic Government Procurement (E-GP) systems. This effort has been further supported by policies from organizations like the World Bank and OECD, and by standards such as the Open Contracting Data Standard (OCDS) by the Open Contracting Partnership.

Government Transparency Institute

The process overview



Government Transparency Institute

Scraping

The creation of GTI's GPPD starts by mapping all possible sources of public procurement data across Europe and selecting a subset of sources that can be processed with sufficient completeness and consistency. The aim is to cover the maximum number of public tenders by collecting data from the official country-specific public procurement publication sites and to collect historical data dating back to the earliest publications possible. As a result of continuous data source mapping, we process TED and national data sources from 17 countries from the EU-27[4]. The processed sources vary and include HTML web portals, XML files (e.g. uploaded to an FTP server), JSON records available from APIs, and CSV data dumps.

Many sources require dedicated solutions, however, the default scraping architecture collects publications in two steps; first it collects all addresses (URIs, URLs, ftp file paths, etc.) pointing to single publications, second it traverses the collected addresses and downloads raw data available on the source. During the processing of each source, the outputs of separate processes are saved in a dedicated database, along with the raw data collected during scraping. Where possible, scrapers use a source-specific heuristic for updating datasets during later collection rounds. During the download process each publication is attributed with a persistent identifier generated by hashing predefined unique variables, such as URLs or IDs available on the source. These persistent identifiers are useful for tracking source changes and the availability of publications.

[4] For example, we process three types of TED publications (2006-2011, 2011-2023, and 2023- eForms versions), as well as, national data sources, which often means multiple national data portals, such as ezamowienia.gov.pl and bzp.uzp.gov.pl sources in Poland.

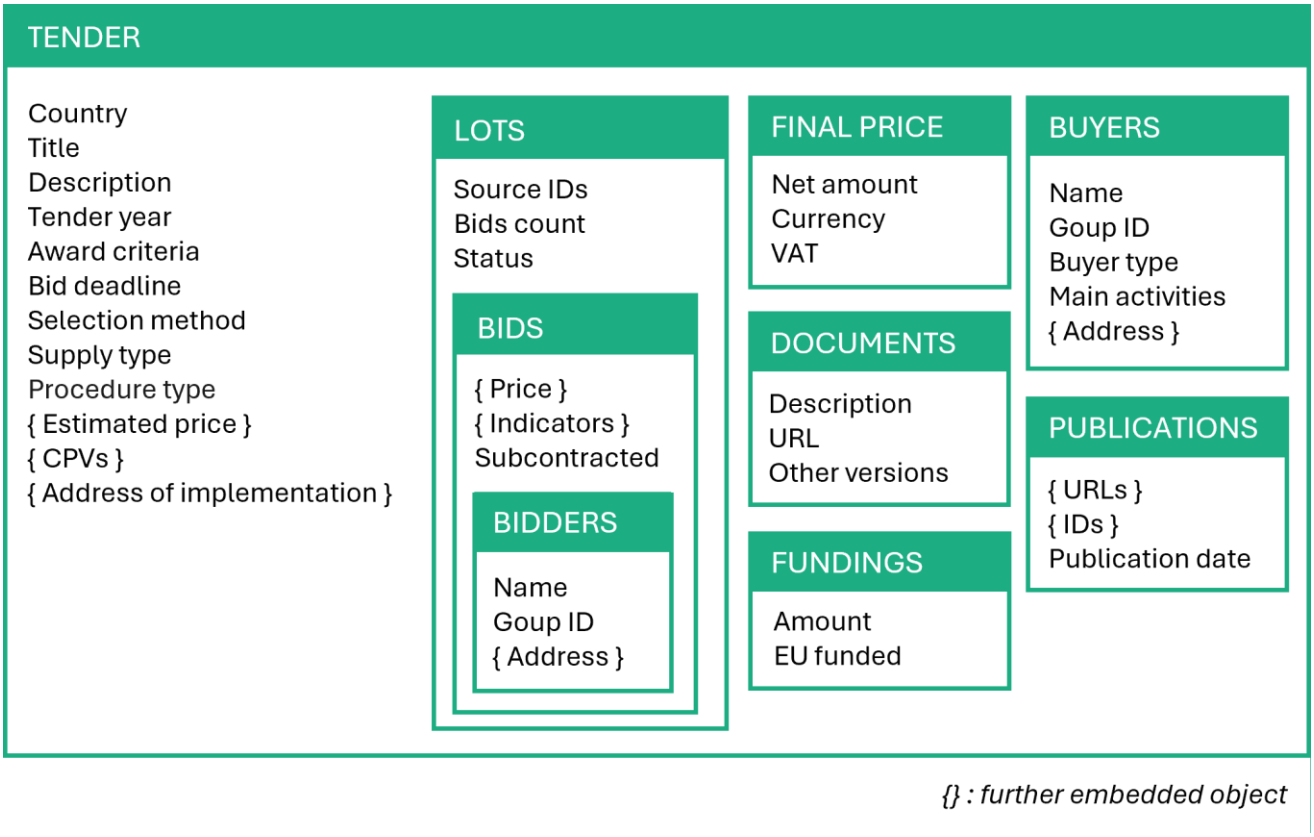
Government Transparency Institute

Parsing

Once the raw data (the procurement notices) has been downloaded, we parse the information stored in them as text values into a data template in JSON format. As the main data structure was established by DIGIWHIST (DW), we refer to the data structure as DW for simplicity. The raw publication values are mapped using manually constructed annotations into the best fitting DW data standard field and by the end crawled announcements are stored in a separate structured data object – i.e. there is a direct relation between one parsed document and the corresponding raw document. At this point all publications are saved separately into the parsed tender database and the values found on the source are not modified in any way.

The DW structure is a relational dataset template primarily informed by the TED publication template and other announcement templates used in European countries. It allows to store tendering information flexibly, allowing to accommodate not only a wide set of fields published on the source but also data enhancements such as connecting lots and contracts, organizations and derived indicators (Figure 1). This data structure reflects the nature of public procurement processes around the world: one entry corresponds to a unique tender, while tenders can be split into multiple lots, and lots can be related to multiple bids (both winning and losing).

FIGURE 1. GENERAL STRUCTURE OF THE DIGIWHIST DATA TEMPLATE



{} : further embedded object

Sources can contain up to several dozen publication types that describe alternative tendering procedures, therefore, complex sources are weighed by annotators so as to include those that are most numerous and most relevant to tendering processes. These publication types are most commonly Call for Tender Notices, Contract Award Notices, Contract Modification Notices, and Contract Cancellation Notices.

The example below demonstrates how variables in a TED XML file are parsed into the DW JSON format (Figure 2).

FIGURE 2. EXAMPLE OF PARSING TED XML

Cleaning

Data cleaning is a process that transforms text data into structured typed data by a) implementing various data type conversions, b) imputing missing information in cases when information can be derived indirectly on specific data fields, c) NUTs code cleaning, and d) price cleaning. Accordingly, each source is processed using a general cleaning algorithm, and a source specific cleaning step implemented using a configuration file written by the source experts.

Data Type Conversion

We convert text data into the following potential data types: a) text (string), b) URLs, c) Boolean, d) dateTime (or date), e) double, f) float, g) enumerated fields. We discuss each conversion below (Table 1).

TABLE 1. DATA TYPE CONVERSION METHODS

Data Type	Conversion method
Text	Text cleaning consists of several modifications. The variables transformed into strings are split into 'short' and 'long' strings and different sets of rules are applied to clean the two. The rules applied on short strings are the following: • All Unicode spaces are replaced with ordinary spaces and all invisible Unicode characters are removed • All trailing white spaces are removed • All HTML4 entities are replaced by a proper character • All white spaces are replaced by a single space character The rules applied on long strings are the following: • All occurrences of the Unicode spaces are replaced with ordinary space and all occurrences of Unicode invisible characters are removed • Specific HTML tags are replaced by the new line character (, <p>, ,) • All HTML4 entities are replaced by a proper character
URL	The most common typos are being fixed and replaced in published data if the original value is not in a proper URL form. If even after these fixes a URL is not in proper form, it is erased from a clean DB.
Boolean	All values that are defined as a Boolean value (true or false) are first cleaned from all Unicode spaces and invisible characters in the same way as short and long strings. As a second step, the original text represented values are converted into true or false values using a library function implemented by Apache software foundation.
Dates	Each source uses a different date format based on local conventions. Many sources even use multiple date formats. When developing programs for data extractions developers detected all possible formats used in a specific source. When converting text values to date values all possible date formats are used for transformation. When a transformation is successful a particular field is stored as a date. If all transformations fail, a value for particular field is not stored

Numbers	Just as Boolean values or dates, numbers are also published differently in each country – for example, some countries use a comma as decimal separator, while others a dot. Therefore, we test several different number formats to find the best transformation from text to number value. Before the transformation can start, the text value is preprocessed as a short text. This means all ballast information like trailing empty spaces, new line characters, multiple empty spaces etc. are replaced or removed from the test. If all transformations fail, the value for the particular field is not stored.
Enumeration values	To be able to provide analysis of the final data we need to convert some fields from national or source specific values to uniform enumeration values. The mapping tables are manually created for each source and the cleaning algorithm applies these mappings to the parsed data. Enumerated values include, but are not limited to: lot status, supply type, tender size, procedure type, body identifier type, body identifier scope, buyer type, buyer activity type, document type, award criteria category, correction type, unit type, selection method, and publication form type.

Imputed fields

We store several data points relating to a tendering process that are not directly published. The source-specific configuration file will contain imputation rules, such as assigning values based on the available data. Imputed information is useful for filtering in later analysis and can allow general categorizations.

As an example, the *Lot Status* imputed field represents the stage of execution the lot was at time of crawling. Its assignable values include ANNOUNCED, AWARDED, CANCELLED, etc. The configuration file of a given source will contain instructions, such as: label as ANNOUNCED if the bidding deadline has not passed yet.

NUTS codes and region names

Conversion files from postcode to NUTS code are available for European countries on http://ec.europa.eu/eurostat/tercet/flatfiles.do. The available address data is enriched by NUTS codes where a postcode is available.

Price/currency cleaning

Exchange rates are used to transform local currencies to standardized ones (EUR) to make prices comparable between countries. The pipeline uses a web service and a cache DB for historic rates that can be shared among country-specific DIGIWHIST databases.

Body matching

The data structure conceptualises buyer organisations and companies (bidders) as body objects. Parsed and cleaned body objects will have several attributes, including name, various address fields, and different IDs (e.g. tax identifier). Body matching is the process that aims to group all body objects that describe the same real-world entity together. In effect, the body objects grouped together receive a shared group identifier.

Body objects are saved during parsing and cleaning based on what can be collected from the source, however, due to organisational changes or inconsistent publication practices both buyers and companies can appear in the cleaned dataset with multiple name variants, with or without multiple identifiers, and with different address fields. Furthermore, the available text can contain encoding errors, alternative languages, punctuations, symbols and other special characters that can cause inconsistencies during text cleaning.

In the pipeline, each country process includes an algorithmic body matching and mastering step. These steps use a shared body database to identify and match all body objects that refer to the same entity. This ensures that if a body already exists in the database and appears again in another data source, it is recognized as the same entity.

- Data pre-processing
- Hash matching
- Exact matching
- Approximate matching

Data pre-processing

During data pre-processing, the fields used in the matching algorithm – names and addresses – are standardised to create a ‘digest’ (explained below). The name and address fields are crucial for comparing records to determine if they refer to the same entity, though a match alone is not definitive. To ensure accurate comparisons, names are standardized by trimming spaces, converting to lowercase, normalizing whitespace, removing accents, and applying special replacements. Special replacements unify variations in naming—such as different forms of business entities or synonyms like "Uni" and "University"—by standardizing them to a common value, ensuring consistent identification of the same organization.

Addresses go through a similar standardisation process, and are additionally concatenated to a street, city and country format, or if a structured address is not available then raw address and country.

Following standardization, a digest is calculated for each record as a pre-filtering deduplication step. It is created as a concatenation of name digest + separator + address digest (Table 2). This value is used for performance reasons to reduce a pool of bodies for which an approximate matching score is calculated.

TABLE 2. DIGEST VALUE MATCHING

name	address	digest
Oxford Uni.	Wellington Square, Oxford OX1 2JD, UK	oxford uni.|wellington square, oxford, uk
Oxford University	Wellington Square, Oxford, United Kingdom, OX1 2JD	oxford uni.|wellington square, oxford, uk

Hash matching

Hash matching groups records that appear identical by generating a consistent hash from the alphabetically ordered body ID values (where body ID is the concatenation of the ID and its scope) and the concatenated standardized name. This method improves performance and prevents false positives, but it only matches records that are exactly the same in the selected fields, such as name and VAT number, even if other attributes like address differ.

Exact matching

The exact matching process compares bodies to existing bodies in the shared body database based on the following body variables: standardized name, standardized address, and all available identifiers. The following steps are applied to find the best group of bodies for a body that is being matched:

- Two bodies are grouped together if their ID and ID scope are equal.
- Two bodies are considered to be an exact match if at least two of their non-empty standardized variables match perfectly. For example, if the standardized name and an identifier, or two different identifiers, or an identifier and a standardized address are the same, the two bodies are grouped together.
- If such a match occurs, the Body is assigned as a member of a group and matching ends.

Government Transparency Institute

Approximate matching

Additional variables are used to calculate a score for a body-to-body match. An S_i score is calculated for unmatched bodies, which is the weighted average of five component similarities—components and their weighting are the following:

Component	Assigned weight
Standardized name	1
Standardized address	1
Postcode	0.2
NUTS	0.2
ID match	1

The body will be assigned to the group for which it has the highest score, as long as the score is above a set threshold. (See example).

APPROXIMATE MATCHING EXAMPLE

Two near-identical companies with a single digit difference in identifier and single character difference in name - *weighted average threshold set to 0.75*

Consider a new body record **Item A** and an existing record **Body 1** present in the Shared Body Database:

Attribute	Item A	Body 1
Standardized name	“acme engineering solutions gmbh”	“acme engineering solution gmbh”
Standardized address	“musterstraße 12, 10115 berlin”	“musterstrasse 12, 10115 berlin”
Body identifiers	VAT DE 123456789	VAT DE 123456788
Postcode	“10115”	“10115”
NUTS codes	[“DE300”]	[“DE300”]

1. **Name similarity** (trigrams):
   “solutions” vs. “solution” differ only by the “s” → **name similarity ≈ 0. ninety-something**
2. **Address similarity** (trigrams):
   “Musterstraße” vs. “Musterstrasse” differ only in the German sharp-s encoding → **address similarity ≈ 0. ninety-something**
3. **Identifier similarity**:
   VAT numbers differ in exactly one digit → **bodyId similarity = 0.8**

4. **Postcode similarity**:
   Identical → **postcode similarity = 1.0**
5. **NUTS similarity**:
   Identical → **NUTS code similarity = 1.0**

The weighted average will be around **0.89.**

Since 0.89 > 0.75, **Item A** is considered an approximate match to **Body 1**, and matching stops on the strongest candidate.

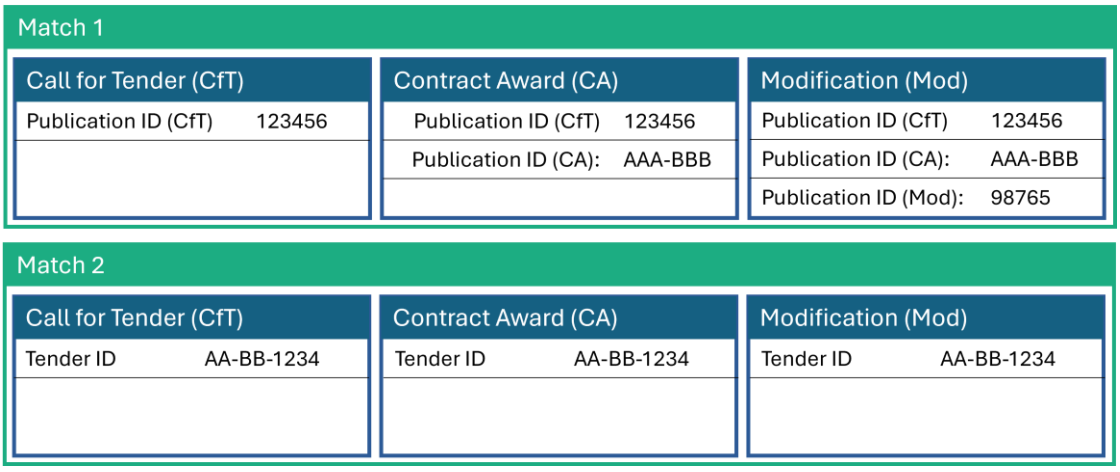
Publication matching

Publications related to the same tender can be linked together by several different approaches to complete the final data structure attributed to the given tender. The exact matching approach depends on the source. For most data sources we apply a single algorithm but there are cases where a combination of matching rules is applied (Figure 3). Such methods include, but are not limited to:

- Publication reference matching: identifiers used on the data source or in URLs are used to match different publications describing the same tendering process.
- Tender ID matching: publications are matched based on a tender ID assigned by the public procurement system.
- Buyer assigned ID matching: publications are matched based on a tender ID assigned by the buyer organisation.

At the end of the matching process each record receives a group ID, but records are still saved separately.

FIGURE 3. MATCHING EXAMPLE



Mastering

Once all publications describing a real-world public procurement tender are linked together using the group ID variable, a final image of a tender, that contains all known information about the tender's

lifecycle, is created. In this step, a rule determines which information should be stored in the final, ‘mastered’ tender data. As several publications may contain the same information relating to a tendering process (for example, a product code), a rule must decide which value to store as the one that best represents the real-world process. In practice, the mastering algorithm iterates over all matched records, which were assigned a Group ID during the matching process. The first record in the group is used as the basis, and data available in the other records in the group are added according to the rules described in the Annex.

Prices

All prices in the DW data model are converted into both national currencies (those coming from national portals) and EUR. At the end of the data processing, where possible, each price object contains three values: 1) netAmount, 2) netAmountEur, 3) netAmountNational.

Deduplication

During the mastering process each mastered tender record receives a Boolean property indicating if the record should be kept after filtering out duplicate tenders coming from different sources. This “opentender” flag is calculated based on the tender’s country and source, publication date, and estimated or final value. Importantly, tenders without any case-specific rules are judged by value thresholds: €135 000 for supplies/services, or €5 186 000 for works, using the best available price (final, estimated, or summed winning-bid amounts).

Framework agreement and DPS mastering

The mastering process includes a special heuristic applied to Framework Agreements and Dynamic Purchase System tenders. These publications are first ingested as “ANNOUNCED” whenever a contract notice is published without any award or implementation information. In this state, each lot defined in the notice is carried forward exactly as published, marked with status ANNOUNCED, and using the published estimated prices. At this stage no bidders are recorded, since there is simply an invitation to participate and no shortlist or award yet.

If a framework or DPS award notice appears listing all qualified suppliers before any individual contracts are issued, the pipeline collapses the original lot structure into a single synthetic lot representing the entire agreement. That lot is flagged as PREAWARDED and each qualified firm gets its own row as a “winning” bidder. Because actual prices are not disclosed at this point (and are often placeholders), the total estimated value of the framework is simply divided equally among all qualified suppliers to approximate potential spend.

Finally, as individual “minitenders” or call-off notices arrive, each is added back into the same tender as its own lot with status AWARDED. These rows carry the real contract details such as title, CPV, description, and record the single winning supplier and their actual award price. Meanwhile, the PREAWARDED synthetic lot remains in the data to represent the following: who was shortlisted for the overall framework and exactly how much was spent on each executed call-off.

Government Transparency Institute

Annex
Variable by variable mastering rules

Matched publications store information on the same characteristics of a tender multiple times, that might be even conflicting. For example, the estimated tender price is published in the Call for Tenders and the Contract Award announcement as well that might or might not be the same. At the data mastering stage, a set of rules are defined for each variable that picks the value (i.e. a tender characteristic) that is the most likely representation of reality. Different subsets of variables are subject to different generic mastering rules (Table A1 – Table A6).

TABLE A1. MASTERING RULES

Rule	Definition
Modus and last published value	1. Take all values and pick the most frequent. 2. In the case of comparing bodies, two bodies are considered the same if they have identical groupId (ie belonging to the same group of matched bodies) 3. If there are more values of the same frequency then select the latest published
Last published value	1. Sort all values by publication date 2. Pick the latest published not empty value
Logical disjunction	Makes a logical disjunction and can be applied to fields containing TRUE/FALSE value. It is evaluated in the following steps: 1. if at least one value is TRUE then the master value is TRUE, otherwise 2. if at least one value is FALSE then the master value is FALSE, otherwise 3. the master value is empty
Longest	Selects the longest text value from all possible values.
Maximum	This rule selects the maximum value from all possible values - for example, it picks the latest date, highest number etc.
Bodies array	Some variables represent an array of bodies like buyers or bidders. Even if matched arrays from different publications contain usually one item, the algorithm has to be capable to handle a situation when arrays contain more than one items. 1. if all arrays contain only 1 body the one with the highest completeness score (described in Body matching chapter) is selected 2. if at least one array contains more than one value, the master value is the union of all published bodies
Union	This rule is applied to variables that are stored as arrays and it tests whether two fields stored in an array are equal. If this condition is fulfilled a union of all arrays can be made. This means all published values are present in a master value and each value is present just once.Table 9
Price	All price objects are handled using this rule 1. All objects that contain netAmout value are taken into consideration 2. For <= 2 prices, use the latest published price

	a. If there are two prices without a publication date, use a random value b. If there is one price object without associated publication date information, pick the one that has publication date information associated as a master value 3. For > 2 prices, we find the netAmount MEDIAN (for an even number of prices, the first of the two middle ones is picked).
Address	The whole address object is selected, individual fields are not merged. For example, if there are two matched tender publications and both contain the address of implementation, one of them is picked as a master value. It is the one with the highest scoring where 1. NUTS has priority 2. otherwise the number of non-empty fields 3. In case of the same score, the last published address is taken
Lots	Since each publication can contain multiple lots and each publication related to the same tender can contain a different number of lots (e.g. contract award publication containing information only about awarded lots vs. contract notice announcing all lots) corresponding lots have to be grouped together before variable by variable mastering can start. This chapter describes how lots from matched tenders are grouped together. Each particular field is then mastered using one of the above or below described rules. 1. Tenders with only one lot: skip the algorithm and put them all into one group 2. Tenders with multiple lots: calculate the matching ratio MR for each cross tender lot-lot pair: a. MR = MS / C, where MS is the matching score - sum of scores from all the comparisons, C is the number of comparisons - number of comparisons on non-null values (null values are not compared) b. compare on following attributes: i. bidsCount (exact match 1, otherwise 0) ii. selectionMethod (exact match 1, otherwise 0) iii. contractSignatureDate (exact match 1, otherwise 0) iv. estimatedPrice.netAmountEur (exact match 1, otherwise 0) v. main cpv code (exact match 1, otherwise 0) vi. title (exact match 1, otherwise 0) vii. winning bids bidders (match of at least one bidder 1, otherwise 0) viii. contractNumber (exact match 1, otherwise 0) ix. lotNumber (exact match 2, otherwise 0) x. positionOnPage c. sort by MR and match lots with MR >= 0.5 (the higher score wins) i. groups are created starting from the best match ii. if the next best match creates an invalid group (only one lot from each publication can be present in one group) than it's skipped

	iii. lots that do not match anything create separate lots
Bids	Fields related to the same bid have to be grouped together before variable by variable mastering within bids can be applied since each lot can have multiple bids. Similarly as for lot objects, each particular field within the bid object is then mastered by one of the variable level rules. Bids are assigned to the same group within tender based on bidder identifiers. The assumption is that each bidder (i.e. a company or a set of companies if the bid is a joint bid) can participate once per lot: 1. if two bids from the same bidder based on bidderId are found that are related to the same lot, we consider them the same bid. 2. if a bid cannot be assigned to any existing group of bids, a new group is created – hence if the same bidder published in a way that the body matching algorithm does not identify them as the same, a new bid is recorded (i.e. a duplicate) for the same bidder.
Documents	Before mastering of the document starts all documents from all matched tenders or bids are grouped, each group describing one document of a final tender or bid. Groups of documents are then mastered variable by variable using one of the rules described above or below. The grouping rule is very simple; all documents with the same URL are considered to be the same document.
CPV	CPV objects are stored as an array and a mastered array of codes is created by the union of all occurring values. Two CPVs are equal when their code values are equal. After the array of all occurring values are created, multiple CPV codes can be market as 'main' CPV code. The following rules are applied to pick one specific code as main: 1. Pick the most specific code (i.e. the most detailed 8-digit code). 2. If multiple codes have the same specificity level, the most recently published most detailed CPV code is picked. 3. If multiple equally detailed codes are published on the same most recent date a random one is picked.
Funding	Funding objects are stored as an array and a mastered array of funding objects are created by the union of all occurring values. Two funding objects are equal when the source and isEuFunded variables have the same values. If multiple funding objects are found that are considered to be the same based on these two variables, the one with more values are kept in the final dataset.
Award criteria	Theoretically, the list of award criteria is only meaningful if the sum of weights amounts to 100%. In order to not introduce any errors in mastering the various criteria, values coming from different publications of the same tender are not combined, and instead the following mastering rules are applied: 1. The list of criteria published in the most recent announcement with weights summing up to 100% is selected as a master value. 2. If none of the announcements have a total criteria weight of 100%, the one with the highest sum of weights is selected as a master value. 3. If multiple different criteria combinations have the same sum of weights, the algorithms picks one randomly as a master value.
Body IDs	The Body ID object has three fields: id, type, scope. Multiple body IDs of the same type and scope would confuse a master body object, hence if multiple different IDs of the same type and scope appears, the following rules are applied:

	1. The one that comes from the company DB record is preferred. 2. If no ID comes from the company DB record, the most frequent value is preferred. 3. If there is no most frequent value, the most recently published value is selected.

TABLE A2. UNION MASTERING RULE DETAILS

Data type	Equality condition
Publication	Two publications are considered the same when sourceId, machineReadableUrl, humanReadableUrl, publicationDate and version are equal. Empty value equals anything.
URL	Two URLs are equal when the string representation of URLs are the same.
Enum	Same enumerated values.
String	Same string values.
Corrections	All corrections are included in a final array of corrections

Entities

These tables describe which rules are applied to which variable within a specific entity.

TABLE A3. TENDER

Rule	Fields
Modus + Last published value	buyerAssignedId, title, titleEnglish, procedureType, nationalProcedureType, isAcceleratedProcedure, description, descriptionEnglish, maxBidsCount, supplyType, size, furtherInformationProvider, specificationsProvider, bidsRecipient, specificationsCreator, appealBodyName, mediationBodyName, maxFrameworkAgreementParticipants, estimatedDurationInMonths, estimatedDurationInDays, estimatedDurationInYears, envisagedCandidatesCount, envisagedMinCandidatesCount, envisagedMaxCandidatesCount, awardDeadlineDuration, country
Last published value	bidDeadline, documentsDeadline, estimatedStartDate, estimatedCompletionDate, awardDecisionDate, contractSignatureDate, limitedCandidatesCountCriteria, selectionMethod, cancellationDate, cancellationReason, isWholeTenderCancelled, enquiryDeadline, awardDeadline
Logical OR	documentsPayable, isDocumentsAccessRestricted, isCentralProcurement, isJointProcurement, isOnBehalfOf, hasLots, areVariantsAccepted, hasOptions, isCoveredByGpa, isFrameworkAgreement, isDps, isElectronicAuction, isEInvoiceAccepted
Longest	deposits, eligibilityCriteria, personalRequirements, economicRequirements, technicalRequirements, excessiveFrameworkAgreementJustification,
Bodies array	buyers, onBehalfOf, administrators, supervisors, candidates, approachedBidders

Union	publications, courtProceedings, courtInterventions, npwpReasons, eligibleBidLanguages
Price	documentsPrice, estimatedPrice, finalPrice,
Address	documentsLocation, addressOfImplementation
Modus + Last published value	buyerAssignedId, title, titleEnglish, procedureType, nationalProcedureType, isAcceleratedProcedure, description, descriptionEnglish, maxBidsCount, supplyType, size, furtherInformationProvider, specificationsProvider, bidsRecipient, specificationsCreator, appealBodyName, mediationBodyName, maxFrameworkAgreementParticipants, estimatedDurationInMonths, estimatedDurationInDays, estimatedDurationInYears, envisagedCandidatesCount, envisagedMinCandidatesCount, envisagedMaxCandidatesCount, awardDeadlineDuration, country
Last published value	bidDeadline, documentsDeadline, estimatedStartDate, estimatedCompletionDate, awardDecisionDate, contractSignatureDate, limitedCandidatesCountCriteria, selectionMethod, cancellationDate, cancellationReason, isWholeTenderCancelled, enquiryDeadline, awardDeadline
Logical OR	documentsPayable, isDocumentsAccessRestricted, isCentralProcurement, isJointProcurement, isOnBehalfOf, hasLots, areVariantsAccepted, hasOptions, isCoveredByGpa, isFrameworkAgreement, isDps, isElectronicAuction, isEInvoiceAccepted
Longest	deposits, eligibilityCriteria, personalRequirements, economicRequirements, technicalRequirements, excessiveFrameworkAgreementJustification,
Bodies array	buyers, onBehalfOf, administrators, supervisors, candidates, approachedBidders
Union	publications, courtProceedings, courtInterventions, npwpReasons, eligibleBidLanguages
Price	documentsPrice, estimatedPrice, finalPrice,

TABLE A4. LOT

Rule	Fields
Modus + Last published value	contractNumber, estimatedDurationInMonths, estimatedDurationInDays, estimatedDurationInYears, maxFrameworkAgreementParticipants, envisagedCandidatesCount, envisagedMinCandidatesCount, envisagedMaxCandidatesCount, bidsCount, validBidsCount, electronicBidsCount, foreignCompaniesBidsCount, SMEBidsCount, otherEUMemberStatesCompaniesBidsCount, onEUMemberStatesCompaniesBidsCount
Last published value	awardDecisionDate, contractSignatureDate, completionDate, cancellationDate, cancellationReason, selectionMethod, limitedCandidatesCountCriteria, status, estimatedStartDate, estimatedCompletionDate

Logical OR	isElectronicAuction, isFrameworkAgreement, isDps, isCoveredByGpa, areVariantsAccepted, hasOptions, isAwardedToGroupOfSuppliers
Longest	title, titleEnglish, description, descriptionEnglish, eligibilityCriteria

TABLE A5. BODY

Rule	Fields
Modus + Last published value	name, email, contactPoint, contactName, phone, buyerType
Logical OR	isPublic, isSubsidized, isSectoral, isSme
Union	mainActivities
Address	address

TABLE A6. BID

Rule	Fields
Modus + Last published value	subcontractedProportion
Last published value	disqualificationReason
Logical OR	isWinning, isDisqualified, wasInRequestedQuality, wasFinishedOnTime, wasForEstimatedValue, isSubcontracted, isConsortium
Bodies array	bidders, subcontractors
Union	unitPrices, payments
Price	price, subcontractedValue

TABLE A7. DOCUMENT

Rule	Fields
Modus + Last published value	title, type, signatureDate, version, order, language
Last published value	description, format
Maximum	publicationDateTime
Union	otherVersions, extensions