

LLI WHITE PAPER

Contract metadata from a legal perspective | What data to track on contracts, and how

Baltasar Cevc / Maik Ebersoll / Klaus Gresbrand / Susanne Marks / Kai Jacob /
Jutta Löwe / Dmitri Geidel



LLI Whitepaper

Contract metadata from a legal perspective

What data to track on contracts, and how

Baltasar Cevc
Maik Ebersoll
Klaus Gresbrand
Susanne Marks
Kai Jacob
Jutta Löwe
Dmitri Geidel

TABLE OF CONTENTS

1. Introduction	6
1.1. Who should read this?	6
1.2. What is metadata and how does it relate to contracts?.....	6
1.3. Why bother about metadata?	9
2. Where to start	11
2.1. Setting the goal: What metadata is worth tracking?.....	11
2.2. Taking inventory: Leveraging what’s there	13
2.3. Doing it right: How to assure metadata quality	15
2.3.1. Rigorous automation.....	16
2.3.2. Respecting the human element and planning to avoid or catch errors	16
2.3.3. Emerging approaches to address metadata quality	19
2.4. Being compliant: How to build the metadata model in a data privacy-friendly manner	20
3. Advanced metadata concepts	22
3.1. System-wide metadata	22
3.2. Process-related metadata	22
3.3. Document-based metadata	23
3.4. Clause-based metadata	24
3.5. Metadata on the data model.....	26
4. Inspiration: Use cases of metadata	27
4.1. Filtering, reporting and data analytics.....	27
4.2. Contract metadata-based KPIs.....	28
4.3. Benchmarking.....	28

- 5. What to expect beyond metadata 29
- 6. Appendix 1: Common data fields 30
 - 6.1. The contract..... 30
 - 6.1.1. Unique identifier (ID)..... 30
 - 6.1.2. Contract title..... 32
 - 6.1.3. Contract status 33
 - 6.1.4. Contract type..... 34
 - 6.1.5. Framework agreements vs. individual agreements 37
 - 6.1.6. Relationships..... 38
 - 6.1.7. Language 39
 - 6.2. Parties and people 40
 - 6.2.1. Contract parties..... 40
 - 6.2.2. KYC Status..... 41
 - 6.2.3. Party’s role..... 42
 - 6.2.4. Contract owner and/or owner division..... 43
 - 6.2.5. Business context/department..... 44
 - 6.2.6. Signatory 44
 - 6.2.7. Signatory role..... 45
 - 6.2.8. Other stakeholders 46
 - 6.2.9. Contract initiator 46
 - 6.3. Legal-specific data points 47
 - 6.3.1. Applicable law/jurisdiction..... 47
 - 6.3.2. CISG applicability..... 48
 - 6.3.3. Place of jurisdiction..... 48
 - 6.3.4. Liability provisions..... 49
 - 6.3.5. Confidentiality level 50
 - 6.3.6. Markers or tags for special cases or circumstances..... 51

6.4.	Coming into force	52
6.4.1.	Signature date	52
6.4.2.	Start date / effective date	53
6.5.	Contract duration.....	53
6.5.1.	Termination	53
6.5.2.	Prolongation	58
6.6.	Business-specific data points.....	59
6.6.1.	Event triggers and notifications.....	59
6.6.2.	Contract volume / amount / value.....	60
6.6.3.	Product (hardware) related data.....	61
6.6.4.	Involved countries/geographies.....	61
6.6.5.	Payment terms	62
6.6.6.	Pricing clauses (e.g. index clauses)	62
6.6.7.	Commodity	63
7.	Appendix 2: Standards and taxonomies usable for CLM metadata	64
7.1.	Contractual features	64
7.1.1.	Legal taxonomies	64
7.1.2.	Standard contractual terms.....	65
7.2.	Other lists or references	66
8.	About the authors	67

1. INTRODUCTION

1.1. WHO SHOULD READ THIS?

This whitepaper is intended to support you in designing a new contract metadata structure, or optimizing an existing one, with a focus on the legal point of view. This whitepaper is therefore particularly helpful if you are tasked with setting up, or updating, your organization's contract lifecycle management (also) from a legal viewpoint.

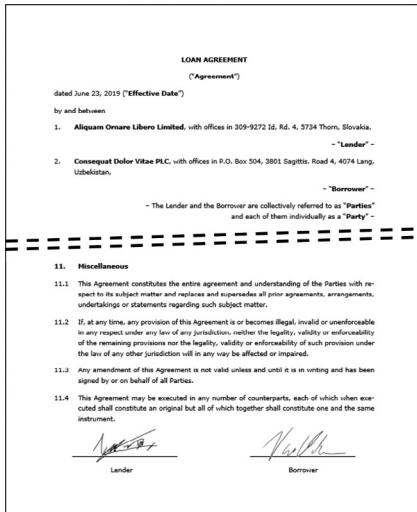
This whitepaper is designed to be accessible for beginners in the field of contract metadata—all you need is basic knowledge of what contracts are and how to manage them (i.e., contract lifecycle management).

1.2. WHAT IS METADATA AND HOW DOES IT RELATE TO CONTRACTS?

Metadata is data providing information about other data.¹ "Contract metadata", therefore, is data about a contract. In this whitepaper we apply this term both to data which can be extracted from a contract's text itself as well as data which goes beyond the mere contract text.

Looking at the image below, the contract on the left contains a lot of information, such as who the contract parties are and the fact that the contract is about one party granting a loan to the other party. The small spreadsheet on the right neatly summarizes some of this information. It contains data about the contract—contract metadata.

¹ Cf. introduction of Jenn Riley, Understanding Metadata: What is Metadata and What is it For?: A primer, National Information Standards Organization, accessible via <https://www.niso.org/publications/understanding-metadata-2017>, last accessed 20 March 2023. "Data" is anything that contains information, the value of which is in the eye of the user.



Contract ("the data")
- unstructured data -

Spreadsheet (data about "the data" = metadata)
- structured data -

	A	B	C	D
1	Party 1 Name	Party 2 Name	Agreement Type	Fully signed
2	Aliquam Omare Libero Limited	Consequat Dolor Vitae PLC	Loan Agreement	Yes
3				
4				
5				

Of course, contract metadata is not limited to summarizing the content contained in the contract text itself. The statement in the spreadsheet that the contract is "fully signed" is actually an interpretation of the signatures visible on the contract. Also, the spreadsheet could, for example, state whether the loan agreement relates to group-internal or external financing, who in the legal department was involved in drafting and negotiating it, how long it took to progress from first draft to signing, or whether it was created from the legal department's own template or started out on third party paper. Metadata does not need to be stored in a tidy spreadsheet—even the handwritten notes scribbled down by the in-house lawyer during negotiations are contract metadata. However, while all "data about data" is metadata, the practical value of metadata increases with how structured and consistent it is.

Example:

Imagine the spreadsheet above contains data about not just one contract, but thousands. Using the spreadsheet, you would be able to quickly identify all contracts with a certain party by sorting or filtering by the respective party name. Hopefully the contract managers who typed in the party names did not make any typographical errors and always spelled the party name in exactly the same way! Also, filtering might be easier if there were a rule that the party belonging to your organization always goes in the first column and the "external" party into the second.

As you can easily imagine, thousands of handwritten notes made during negotiations on pieces of paper are all but inaccessible unless someone types them into the spreadsheet, e.g. in a “comments” column. However, even if this happens, the information contained in such a free text column would be much more difficult to sort or filter because any kind of text could be placed in it. At least the comments column allows for keyword searches, which is already a vast improvement over notes on paper.

This illustrates that it is meaningful to consider not only *what* metadata you would like to track for each contract, but also *how* you want to do so, especially what restrictions on entering data you want to impose on the users to ensure consistency. As you can see from the spreadsheet columns “contract type” and “fully signed”, some data input requires interpretation—or even a legal assessment—by the user. As all lawyers know, determining the “type” of a contract is not as easy as it seems! What, for example, would you enter in the “fully signed” column if you had two copies of the contract, each signed by only one of the counterparties, but you are unsure whether signing in counterparts is actually permissible? Such difficulties can be mitigated if they are duly considered in the process of designing the data structure.

To take another piece of terminology, in the context of contract metadata you might encounter the terms “master data” and “transactional data”. Definitions vary, but master data typically describes fairly static pools of data that applications or processes tap into, while transactional data is more volatile and often used in relation to certain time or date references or iterations. Again, let’s turn to our example to illustrate the differences.

Example:

In order to better manage the consistency of the entries of party names, you configure the spreadsheet so that entries of party names cannot be typed in freely but must be selected from a pre-defined list of supplier names. This list

of supplier names is master data. Which supplier is party to a particular contract entered into at a certain point in time is transactional data. You may also consider defining a fixed list of contract types for users to choose from. Again, the list of selectable contract types is master data while the contract type chosen for a specific contract is transactional data.

Data points such as party names or contract types are obvious examples of contract metadata because they are generated by humans. But there is another dimension to consider, machine-generated data points.

Example:

The shared drive that stores the spreadsheet automatically creates a version every time the spreadsheet is updated. This new version has a timestamp and notes which user created the new version. Going through the version history of the spreadsheet, we can see how the metadata on the contract has changed over time, and who made the changes.

Even in our example using a simple spreadsheet, more metadata is generated than might be immediately apparent, some of it without any effort by the human user. Knowing when the spreadsheet was last updated, and who made the update, can be immensely helpful. Of course, a wealth of automatically generated metadata becomes available when managing contracts not through a shared drive and spreadsheet but through a contract lifecycle management platform, or if contracts are generated using a clause database and corresponding contract creation tools.

Note: *While this whitepaper is “tool agnostic”, it was written with the capabilities of current contract lifecycle management platforms (not just spreadsheets) in mind.*

1.3. WHY BOTHER ABOUT METADATA?

The extent to which you are able to reap the rewards of contract metadata depends on whether you track the right metadata, and with sufficient data quality. Tracking metadata on contracts has three major benefits, even if you start with a very basic data structure:

- I. Having structured metadata enables you to search and filter effectively through contracts and thereby make your contracts accessible and transparent in a way that just searching for words in the contract text cannot.

- II. Metadata allows for keeping track of commitments and deadlines in a systematic way—and thereby enables automated reminders—which ultimately increases contract compliance and helps avoid legal risk such as non-compliance losses.²
- III. Consistent metadata facilitates reporting on your contracts and, at a more advanced level, allows for analytics that generate additional insights into your contract landscape and contract-related processes. It allows you to, for example, track a multitude of metrics or key performance indicators (KPIs)³ across the contract lifecycle and point you towards optimization potential.

In summary, metadata is *the* starting point for the progression from data to information, knowledge and wisdom (DIKW)⁴ which ultimately provides actionable insight into your contracts and their lifecycle.⁵

² In a recent study by the audit firm Ernst & Young, “seventy-eight percent [of the legal departments asked said they] don’t systematically track contractual obligations”—a mind-bogglingly high number of departments seem to have no systematic approach to managing contracts (or rely on colleagues from business units, procurement, sales, etc.); cf. Ernst & Young, *The General Counsel Imperative: How do you turn barriers into building blocks?*, 2021 EY Law Survey, available at https://www.ey.com/en_gl/law/general-counsel-imperative-barriers-building-blocks, section “Priority 1: Transforming risk management”, last accessed 10 March 2023.

³ For more information on KPIs for the legal department, please refer to the LLI Whitepaper “Data-driven Legal Departments, An Introduction into Metrics and KPIs”, Ebersoll / Obst / Querfurth / Schichl / Schindler / Schreyvogel / Stoeckel / Waltl. DOI: 10.38023/e23ee12a-66d3-463a-bbd7-5cc374fcae9d; download for free at <https://www.liquid-legal-institute.com/library/>, last accessed 10 March 2023.

⁴ The DIKW hierarchy has been established for many decades and is commonly accepted in the field of information science (and many other knowledge-intensive disciplines). Rowley, Jennifer (2007). “The wisdom hierarchy: representations of the DIKW hierarchy”. *Journal of Information and Communication Science*. 33 (2): 163–180; doi:10.1177/0165551506070706

⁵ To mention some further benefits: identifying relevant contracts at important events (e.g. change of law, change of outer business setting, contractual triggers depending on the business); finding all contracts for a specific offering of the own business; finding all contracts with a certain other party (or group of companies); relating contracts to company standards; quickly getting information about contracts which feature certain risks and/or opportunities (probably hard to handle, but a good use case nevertheless, to see if it’s possible); optimizing financials (e.g. by using generic price structure information about the metadata).

2. WHERE TO START

Your metadata model design should start with the question:

“What does my organization need to know about its contracts?”

Resist the temptation to just incorporate whatever metadata model already exists in your organization, or what the software vendor offers as its “standard set of metadata”. Dare to start from scratch. The best way of identifying what our organization needs to know about its contracts is to talk to a diverse set of stakeholders in your organization about what type of information they need and what type of questions about the organization’s contract portfolio keeps your colleagues (and their external advisors) awake at night.

2.1. SETTING THE GOAL: WHAT METADATA IS WORTH TRACKING?

Once you have formed an idea of what your organization needs to know about its contracts, you can ask yourself the next question:

“What information about our contracts is worth tracking and maintaining on an ongoing basis?”

As you know, contract metadata provides answers to the questions your organization has about its contracts, but there are two radically different approaches to generating these answers. You can send human reviewers (often lawyers) to look through the contracts manually **after a specific question surfaces**, or you can make an upfront investment to track and maintain metadata on contracts **before the question is even asked**. The latter approach, obviously, is what we consider when setting up a contract metadata model. You anticipate certain questions your organization will have about its contracts and prepare the metadata to answer the anticipated questions ahead of time. Both approaches have their merits and demerits.

Approach	Merits	Demerits
Ad hoc human review (potentially AI-empowered)	<ul style="list-style-type: none"> • Delivers answers to any conceivable question • Effort is only spent once a question emerges • Effort can be clearly allocated to the question (your question, your budget) • You get a perfectly tailored answer to the question 	<ul style="list-style-type: none"> • Costly and time-consuming, especially if many contracts need review • Cost and effort are triggered anew each time a question arises • It can be challenging to manage consistency, especially in teams with many reviewers
Track and maintain metadata	<ul style="list-style-type: none"> • Delivers answers instantly and effortlessly, even if many questions are asked by many people every day • Data quality can be managed consistently <p>Side effect: Your metadata model may reduce the time and effort needed for ad hoc reviews to the extent that it can be used to narrow down the contracts that require human or AI-empowered review</p>	<ul style="list-style-type: none"> • Can only answer questions which were anticipated when the metadata model was set up • The effort of initial collection and ongoing maintenance of metadata is difficult to measure and allocate budget-wise • Upfront investment is required to answer questions that have not yet been asked (and may never be)

In essence, data points should be included in the metadata model for permanent tracking instead of being handled by ad hoc reviews if they:

- can be reasonably expected to be needed by your organization on a regular basis;
- are needed by a sufficiently large number of individuals (or are super-important to a smaller number of individuals); and
- can be tracked and maintained in sufficient quality with acceptable effort.

The “once in a lifetime” questions are much better dealt with ad hoc. You won’t be able to anticipate them anyway.

The skeleton model

If you are unsure about how to approach building your own contract metadata model, you should go through this checklist of core topics your model should consider:

1. **Content and context** (e.g. contract title, subject matter, contract type, applicable law) – Answer the question: What is the contract about?
2. **Organization** (e.g. business department, contract owner, region) – Answer the questions: Where does this contract belong in my organization? Who is responsible?
3. **Parties** (e.g. contract parties, or internal parties and external parties) – Answer the question: With whom have we contracted?
4. **Term** (many data points behind this one) – Answer the question: Until when will the contract be in force? When do I need to take action to terminate or prolong?
5. **Status** (the signature status, which might include ‘signed and in force’) – Answer the questions: Is the contract currently in force? If not, where are we in the contracting process?

In our experience, these five dimensions are excellent candidates for delivering answers in the form of structured data.

2.2. TAKING INVENTORY: LEVERAGING WHAT’S THERE

Now that you have an idea of what items your metadata model should contain in an ideal world, it makes sense to take a reality check and ask yourself:

“What metadata is already available in my organization?

Where can I tap into existing data sources?

What metadata is my organization used to?”

These questions have two implications, the metadata on a contract has to be entered initially, and it has to be kept up to date over the contract lifetime. If you already have a legacy system that stores contracts (or several such systems), the questions above help you consider how the existing contracts can be transferred to the new metadata model, either by changing the model of the existing system or by migrating the metadata into a new system.

Your focus at this stage should be on:

- identifying data points that exist in your organization and that your colleagues perceive as useful and effective; as well as
- existing data sources that may “fuel” your metadata model.

A few examples to illustrate our point:

- If your organization is used to clustering contracts into certain contract types, coming up with a totally different contract type list—even if you think it is more logical or advanced—may not be perceived as helpful by your colleagues and ultimately hamper adoption and data quality.
- If your organization categorizes contracts or documents into certain degrees of secrecy, your metadata model should accommodate this. Don’t invent your own categorization if you can avoid it.
- If your organization’s customer relationship management (CRM) system has a database of every customer with their full address, your metadata model should tap into it—don’t set up a separate pool of customer data if you can avoid it and stick to the address format of the CRM system.
- If you plan to link projects to certain products, organizational departments, business lines or geographies, look for systems in your organization that already contain lists of these and have such systems feed into your metadata.

As noted at the outset of this chapter, there is no value to be gained from mindlessly replicating whatever metadata model you encounter in your organization, and sometimes tough choices must be made in order to make a change for the better. However, it will pay off to understand and honour established and successful practices regarding contract metadata.

The power of standards

Relying on standards can help reporting based on standard conditions. Few standards describing contracting exist, but one can rely on generic standards for certain topics (e.g. geographical designations) and we expect further standards to develop. Where an industry uses specific contract terms or standards, referencing them in the metadata model is advantageous. For example, including the INCOTERMS™ describing delivery terms, or reliance on standard contracts for public procurement or industries (e.g. FIDIC™, German VOL/B and VOB/B including additional terms such as EVB-IT).

Where a contract relies on standard contract terms, it is advisable to refrain from duplicating their content in other fields on a per-deal basis and use standard mapping instead, i.e. the details should be prefilled based on the standard contract terms.

Another example is the well-established ISO codes for country abbreviations, languages, etc. These can save a lot of time and boost the quality of data fields.

See **Appendix 2: Standards and taxonomies usable for CLM metadata** for details.

2.3. DOING IT RIGHT: HOW TO ASSURE METADATA QUALITY

Any system storing data is useful and valuable only to the extent that the data stored fits the intended purpose and is correct and complete.⁶ This is particularly true for contract metadata stored in a contract lifecycle management (CLM) platform. Just imagine a document containing a signed contract being uploaded to a CLM platform under the wrong contract type, with a nondescript title (such as “service agreement”), without correctly stating its parties. A contract uploaded in this way is likely to end up as a ghost. Just a couple months later it will be almost impossible to find, and this will impede the quality of reports generated by the system.

⁶ Complete is not an absolute term here. The statement does expressly not imply that all data of any sort needs to be collected. What matters is that the data is not incomplete so as to create a skewed image or hinder answering questions about your contracts. Especially when used for reporting, relevant fields not being filled for a significant number of contracts can distort the distribution across the contract population.

Two practices can guide you in your endeavour to promote metadata quality: (i) rigorous automation, and (ii) respecting the human element.

2.3.1. *Rigorous automation*

The idea of rigorous automation is simple. To the extent that contract metadata can be generated automatically, without the need for human intervention, it should be. This ensures highly consistent and reliable data.

This not only applies to data fields which can be filled without human user input (e.g. the time at which the system logs the first request to draw up the contract, or the time at which the contract status is set to “in negotiation”), but also to fields that can be automatically filled based on prior human user input. For example, if the user has manually selected a contract party, the related information, such as the address of the party, should be pulled automatically from the appropriate repository.

When designing a process to automatically insert contract metadata, keep in mind that certain data fields may require periodic or event-driven updates. If possible, these should also be automated.

2.3.2. *Respecting the human element and planning to avoid or catch errors*

- If human user input cannot be avoided, the challenge is to support the human user by providing optimal input and preventing input that is clearly wrong. The following ideas may help you design a system that facilitates high-quality data input by human users:
- Make efforts to clearly explain to users **what** information they are asked to provide in a given data field. Having clear, unambiguous, concise field names, descriptions and values will go a long way. The more prior knowledge about your organization’s standards and expectations you assume your users have, the more likely somebody will misinterpret what the necessary input is.
- Help users understand **why** they are asked to fill out a specific data field. This is especially true if the purpose or benefit of tracking the data is not self-evident. Ideally, a user should be presented with a short explanation of the purpose of a specific data field and how it should be filled correctly, directly on the questionnaire.
- Try to **limit the number of data fields** that require human user input as far as possible. As you may know from personal experience, it requires significant discipline and resolve to correctly fill out a form with 50 questions. If you have many

data fields to track, consider breaking the questionnaire into parts, presented to different users or at different points in the contract lifecycle.

- **Consider the point in time** at which a human user can most easily provide the correct data for a given data field. For example, if you track the “effective date” of an agreement, it would make sense to ask for this only after the contract has been signed—because the effective date provided in the first draft may have changed during the negotiation.
- **Find the right person** to enter the data. Ideally, the person tasked with providing the data for a given field is the person best qualified to provide that data, and to whom the data matters most. A business user may, for example, be very aware of, and interested in correctly tracking, the commercial aspects of a contract. A user from the legal department, on the other hand, may be best suited to identify the applicable law or answer questions about liability limitations.
- **Embed user guidance** in close proximity to the data field, and keep it updated. Sometimes a user may struggle to make the right choice, especially when filling a data field involves interpretation that goes beyond copying and pasting information from the contract text itself: Is the contract “signed and in force”? What is the contract type? Here, unambiguous and concise guidance can help. Ideally, the system would allow users to easily log questions to be answered in an updated version of the guidance.
- Make it easy for users to provide consistent metadata by prioritizing lists of **pre-formulated choices** over asking users to manually type data into a text field.
- When offering a choice, make sure to **limit the number of possible options** as much as possible. Oftentimes this comes down to a balance between the granularity of the data and the quality of each data point. Fewer choices typically coincide with less granularity. Tracking 200 contract types would allow you to track many different types of lease agreement, while a list of 10 choices may not offer such luxury, but may produce more coherent user input. Let the purpose of the data field guide you when determining the degree of granularity required. When designing the possible options, keep in mind that offering a generic option such as “other” will become by far the most often-picked option and may defeat the whole purpose of the data field.
- When offering a choice, consider that the user interface for making that choice should facilitate **easy data entry**. As a general rule, users have an easier time making a choice if they can immediately see all available options, for example by being presented with radio buttons or checkboxes. However, this only works optimally for a small number of options. When asking users to make a choice from 10 options, a drop-down menu is often easier to work with.

- Consider **pre-filling** certain data fields to speed up data entry. Ideally the fields would not only include the most typical values but could be automatically chosen by a software algorithm based on other data fields. But beware, human users tend to leave pre-filled data fields as they are without giving them further thought. Only pre-fill if you are confident that the pre-filled value is correct. On a related note, data fields that are empty (not filled) should show this. Do not fill them with default values if they really do not have a value. This way, you still have a chance to later identify empty, i.e. not yet filled out, data fields to reassess.
- Consider creating a **dynamic user interface**, which presents only relevant mandatory and optional data fields. If the user input into a given field indicates that other fields are no longer relevant, they should become invisible or—the other way around—additional fields should appear if needed based on earlier input.
- **Use data validation**, i.e. customize the data fields as much as possible for the expected data type (string, integer, Boolean, real number, date, etc.). If you expect a number, prevent the user from typing letters! If you think a field should always be filled out, make it mandatory, but keep this to a minimum as it can lead to entering any data if asked at an inconvenient time or at an excessive volume. If only a positive amount should be entered, prohibit values below zero. Be careful, however, that you do not set up your data validation too strictly, otherwise you risk alienating users or even preventing them from making useful, high-quality input which you simply failed to anticipate when setting up the system. On a related note, you can consider setting your system to perform data normalization tasks following user input, such as removing superfluous spaces at the end of an entry. Speaking more broadly, systems should not assume that users do everything right but rather be designed to **detect and catch errors** and provide means for correction.
- If data needs to be kept up to date, consider mechanisms which **remind users** to revisit certain data fields.
- When considering mandatory data fields, accept that for certain cases **done is better than perfect**. One of the worst things that can happen is that the human user boycotts the process of entering data, either by providing answers that are incorrect (but quick) or by failing to upload contracts into the system altogether. Defining data fields as mandatory (setting the system so that it refuses to process a contract if the data field is not filled out) can seem like a hurdle to users, but sometimes it is better to get the contract into the system with just a stub of data than not at all.
- **Test your choices and guidance** with a variety of users. Nobody is able to foresee all (mis)interpretations and questions. The earlier you test, the higher the likelihood of understanding, clarifying ambiguities and being able to fix issues.

A good method is observing users using the tools (after receiving consent) and asking them about their observations. Ideally you should test: (1) whether the appropriate person is entering the data (and who else could do it alternatively); (2) whether the chosen point in time is appropriate (or whether, for example, data is still prone to change); and (3) whether they understand the guidance correctly and whether it answers their questions.

- **Check on what your users are doing.** Plan for regular quality audits, which would ideally check: (1) whether the metadata quality for a given contract is up to your expectations; and (2) whether the agreed process was observed when entering the data. This will help you understand whether you are reaching your goals in terms of quality and might help identify potential process improvements.

2.3.3. *Emerging approaches to address metadata quality*

The above covers the basics, but let's also look at a few more advanced concepts for data quality management.

2.3.3.1. **Gamification**

Gamification considers human playfulness. Children like playing, and many adults do likewise. Examples of gamification approaches include giving each user a level or rank which progresses with continued proper use of the system (e.g. moving from "beginner" to "intermediate", "advanced", "legendary", "mythic", accompanied with a similarly evolving avatar or icon). Gamification can motivate people to add (suitable) data, if implemented correctly. However, be aware that incentives can backfire. If poorly implemented, this could trigger people to add data, just to check the box (and advance in their rank) without taking care that data is actually accurate. If you intend to take this path, you will almost certainly need expert advice to get a good implementation thereof. But a simple grading system or gentle nudge, such as: "Your contract metadata for this contract has a 95% score. Would you like to get to 100%?" might change user behaviour and encourage them to enter the missing piece.

2.3.3.2. **Internal crowdsourcing of metadata to mitigate human bias and errors**

Errors in data are often the product of human error, which can be attributed to lack of interest, bias or other misjudgement. Especially where data quality is crucial, for example because high value decisions are made based on the data, mitigating such errors by unconventional means can be key. One such measure is using pro-

cedures to compensate for errors by “averaging them out”, or getting values from several people and correcting thereupon.

Especially where errors stem from bias or similar misjudgement, crowdsourcing can improve data quality. The basic concept is that data is not collected by asking a single person but several people. Where people need to enter data based on a quick judgment or estimation, their biases heavily influence the outcome. While this sounds like bad news, you can turn it to your advantage, as biases for the relevant data fields differ across people,⁷ and usually level out when you average the entries of several people (start with at least three to have a somewhat reasonable basis for correction).

Where your data is not subject to bias but other errors, you might need to adapt the concept, for example by asking two people to provide a value and a third to review and give a final judgment if the values of the first two differ. The concrete implementation strongly depends on the setting and data fields.

2.4. BEING COMPLIANT: HOW TO BUILD THE METADATA MODEL IN A DATA PRIVACY-FRIENDLY MANNER

Note: *This section is written with mainly the European data privacy framework in mind, especially the European General Data Protection Regulation (GDPR). Similar, but not necessarily identical, concepts may apply in other jurisdictions. In any case, please obtain appropriate legal advice when making data privacy-related decisions about your contract metadata model. Reading this section will not suffice! Principles such as data minimization, transparency and access rights for individuals also apply to many other privacy laws, e.g. the CPRA in California.*

Compliance is an important topic for metadata. On the downside, errors in the design or operation of your contract lifecycle management platform, including its

⁷ Bias can be personal, but might be shared across a group or even broader society. Examples include gender and race biases, which are being identified more often. For most applications relevant to the contracts field, the probability of such bias is low, as we are discussing commercial factors. While the legal profession might be biased by training, that does not usually impact outcomes. Where all lawyers share a bias, the lawyers at court most probably also share it. While this makes the approach viable in most cases, we consider it good practice to think about the potential bias involved and confirm suitability in your specific case.

metadata model, can potentially contribute to breaching data protection rules, which can cause high fines for your organization. On the upside, your metadata model can play an important role in ensuring data privacy compliance.

Data privacy laws, such as the European General Data Protection Regulation (GDPR) aim to protect personal data, such as a names, e-mail addresses, dates of birth or gender. Contracts themselves as well as contract metadata can, and very often do, contain personal data in the sense of the GDPR. An organization operating a contract lifecycle management system and the metadata contained therein is typically regarded as a “controller” by the GDPR, and therefore obligated to observe GDPR rules. Two topics are especially relevant to contract metadata:

- The GDPR states that processing personal data has to have a certain purpose with a legal justification. Some general legal bases are named in the GDPR itself, while others are contained in special laws. [Impact on contract metadata](#): Your contract metadata model should only contain personal data for which a sufficient purpose and justification can be demonstrated.⁸ Generally speaking, the less personal data your metadata model contains, the smaller the data privacy-related risk.
- Under the GDPR, a data controller can be required to delete or anonymize certain personal data. [Impact on contract metadata](#): Your metadata model should be designed to support such targeted deletions or anonymizations. For example, you could keep track of which data points in your metadata model could potentially contain personal data and educate your users on the importance of entering only such personal data into the system as is needed. Also, you could consider adding data points that track whether personal data is contained in a contract (the contract document itself and/or its metadata). This can be as simple as a checkbox for “contains personal data”.

⁸ This is also true for the text body of the contract itself, which could motivate you to consider drafting contracts devoid of personal data.

3. ADVANCED METADATA CONCEPTS

3.1. SYSTEM-WIDE METADATA

While the focus of this paper is metadata which immediately relates to a contract, it is worth considering that the IT system that holds your contracts can provide a wealth of metadata about the entirety of your contracts too—such as the number of contracts in the system, how many are associated with a specific contract type or counterparty, and the average time for progressing contracts of a certain type from initiation to signing. This system-wide contract metadata is usually used in contract reporting (see section 4.1).

3.2. PROCESS-RELATED METADATA

Many processes are involved in, or relate to, the creation, negotiation and signing of a contract, and all of them generate a wealth of information that could be captured in an appropriate metadata model.

The most common process to mention here is certainly the approval process. Most organizations allow a draft to be sent out, or a final draft to be signed, only after review by certain stakeholders from their respective angles. Relevant contract metadata regarding approvals would include who has requested, and who has given, what type of approval at what point in time and on what version of the contract. The approval processes not only applies to an individual contract but should be tracked for document templates or even clauses in a clause library. This helps keep track of who has agreed for a certain version of a template or clause to “go live”.

Another key process in the contract lifecycle is the negotiation of the draft contract. Data points to look for here can relate to duration (e.g. the duration of the negotiation phase as a whole, but also the duration of the contract being under review by one party versus the other) or individual interactions (e.g. the number of back-and-forths until completion or which clauses have seen the most redlining). It can also be revealing to track how much pushback there has been from the counterparty against each clause during negotiations, for each contract type, and whether the counterparties’ negotiation behaviour is changing over time. If a particular clause

starts seeing more pushback over time, maybe it should no longer be considered a viable starting point for efficient contracting.

Another process worth considering is the signing process. If done electronically, the e-signature solution should already track metadata on the signatories, time of signature, requester of the signature etc. In the case of ink-on-paper signatures, a contract lifecycle management system can be configured to track such data as well, but inserting it will often require some manual intervention.

Most organizations conduct know-your-client (KYC) procedures, also known as third-party risk management (TPRM), before contracting with another entity. While the metadata on KYC processes often sits in a dedicated system, for your contracts—and in your contract lifecycle management system—you might want to consider a data field summarizing the KYC process outcome, such as “KYC status”, which would allow for a choice between, for example, “pending / initiated / successful / failed”. Ideally, this status would be pulled and updated automatically from the KYC system. The contract metadata model might even allow you to drill down to any beneficial owners identified for each party.

3.3. DOCUMENT-BASED METADATA

In many cases, a contract in a CLM platform consists of more than one document—maybe it was signed in counterparts, or maybe the appendices are stored in a separate file. Of course, many CLM platforms can also store documents that do not form part of a contractual agreement but are somehow related.

For each such document, be it a PDF file, a Microsoft Word document or some other format, metadata can and should be managed on a per-document basis, on top of the metadata which applies to the contract in general. As inspiration, typical data fields in this context include:

- Document title
- Document type and/or document extension (Microsoft Word file, DOCX; PDF-File, PDF; ...)
- Document size
- Created on
- Created by

- Last modified on
- Last modified by
- Version number.

It is worth contemplating whether you would like the CLM platform to show some form of status for these documents, which could open the same menu of selections as the entire contract status, but separately for each document (see section 6.1.3 for details). Also, it is essential to flag which files constitute the actual signed contract and which are additional information (such as drafts). This can be done by adding an additional metadata field, but can also be achieved by other means, such as storing the documents that comprise the actual contract in a separate folder or document library.

Again, part of the challenge of building a good document metadata model lies in resisting the temptation to over-collect data. As always, there is not much to say against data that can be collected automatically, while extreme scrutiny is advisable for all metadata that requires human input, especially if users need to manually enter metadata for every document. There is a risk of users boycotting the system, using it less or providing incorrect data.

3.4. CLAUSE-BASED METADATA

Some CLM platforms track metadata not just for the entire contract, but for each individual clause in the contract. In most cases, the CLM platform dictates most of the metadata for technical reasons, with little room for making individual metadata design choices.

Before looking into suitable data fields, a distinction must be drawn, as clauses carry metadata in two ways. Firstly, the clause, as an abstract building block of a contract, has metadata that identifies it and describes the context in which it can be used or what dependencies it has with other clauses—you can consider this “clause master data”. Secondly, once a clause is used in an individual agreement, the version used, whether or not it has been amended during negotiations, can be tracked. This type of metadata is “clause transactional data”.

Typical data fields for clause master data are:

- the name, type and language of the clause;
- dependencies and conflicts (i.e. if using this clause requires or permits the inclusion of other clauses in the contract);
- the internal owner of the clause (i.e. the individual who has drafted the clause or is most knowledgeable about its contents);
- by whom the version was approved and when (see above process-related meta-data).

Typical data fields for clause transactional data are:

- the version and language of the clause used in the contract;
- whether, and by whom, it was amended during drafting or negotiation (redline history).

Some CLM platforms even allow for a clause to bring its own negotiation playbook, i.e. explain what deviations from the standard text are permissible and who would need to approve them.

Tracking metadata at this granular level over time allows for insights which meta-data relating to the entire contract cannot offer, such as how much effort is spent on the negotiation of a specific clause and what the outcome is. Please note that a clause staying as it was originally proposed can mean two very different things—either it is acceptable, or effort (potentially much effort) has been spent on convincing the contracting party to keep it as is. The system might also be able to measure how often the organization concedes on a specific clause, yet this requires a warning, this data may be highly biased by decisions taken upfront (how much the organization or negotiator are willing to concede).

Often, contract metadata in an actual setting implies more than the underlying contract. For example, consider the “litigiousness” of a clause (or contract). While a poorly drafted clause might be more prone to litigation, a strict limitation of liability clause might decrease litigation probability. The setting, including the industry, is also very important. Certain fields (e.g. building) are highly litigious, while others (e.g. standard software) might be less so. As another example, the limitation of li-

ability clause is not considered the most important in a contract, but it is the most negotiated.⁹

3.5. METADATA ON THE DATA MODEL

When designing a metadata model, especially the master data, it is tempting to treat the model as a static environment, but this is not the case. Metadata models can evolve dramatically over time. Existing data fields are expanded to allow for more choices, and entirely new data fields are added to the model as new requirements emerge and users provide feedback.

Example:

A CLM platform contains a list of possible contract types. Each contract in the platform is assigned to one of these contract types by the respective contract owner. The list of possible contract types is unchanged since the go-live of the platform—until one day an additional contract type is added to the list.

In the example, the list of contract types would be master data, while a contract type assigned to a contract would be transactional data. What constitutes metadata on the CLM in this example is the fact that the list of contract types was changed. It first existed in a version 1 and is now in version 2, which contains the additional contract type. This information can be important to ensure data quality. Now that a new contract type is available, it might be necessary to review certain contracts and determine whether the newly added contract type should be assigned to them. It is therefore worth considering tracking, for each contract, what version of a specific data field master data was used when selecting the value for that field.

In fact, the very amendment of the list of contract types should be captured in a process that generates process-related metadata (see above), such as by whom the updated contract list was reviewed and approved.

⁹ Cf. World Commerce & Contracting (the former International Association for Contract and Commercial Management) which publishes studies about negotiations, e.g. the Most Negotiated Terms 2020, available at https://www.worldcc.com/Portals/IACCM/Resources/9934_0_Most%20Negotiated%20Terms%202020.pdf, last accessed 10 March 2023, slide 4.

4. INSPIRATION: USE CASES OF METADATA

4.1. FILTERING, REPORTING AND DATA ANALYTICS

The previous chapters refer to scenarios where users actively search for certain information. Typically, the need for information is triggered by an external event. However, instead of waiting for the next triggering event, you might decide to proactively monitor the contract landscape.

Whatever data you include in your CLM, is available in this central repository and therefore might be analysed and reported on:

- Your risk management department is concerned about your exposure to certain jurisdictions. A standard report of “contracts by jurisdiction” or “contracts by country” might help.
- Your insurance department regularly needs up-to-date numbers on liability exposures. You might provide a report based on data points surrounding contract liability, see for example section 6.3.4.
- The HR department intends to review the headcount required in the sales or procurement department in certain countries and/or product lines. Analysing the growth/decline of corresponding contracts over the last years might be a good starting point.

In addition to such pre-defined scenarios which might be addressed by respective reports, you might also want to look for “unknown unknowns”. Combining various data fields and observing development over time might reveal certain data patterns, which could be the starting point for completely new and maybe surprising insights into your business. For example, a sudden uptick in negotiations on a certain clause may hint at the wording being out-of-touch with latest market developments. Looking carefully at clauses prone to pushback may result in a more counterparty-friendly default clause wording that shortens the negotiation time.

4.2. CONTRACT METADATA-BASED KPIS

Key performance indicators (KPIs) measure the efficiency or success of business processes. Contract metadata can be a good data source for KPIs on (but also beyond) contracting. As seen in the previous section, the complete set of data can be filtered, and datasets can be classified, categorized and analysed. This not only applies to the core metadata, but also to more advanced aspects presented in section 3, such as the time between certain status changes.

Such analysis provides specific sub-sets of data that ideally correspond to the output or input of a specific business process. For example, you might end up with the number of newly negotiated contracts handled by the Argentine procurement team in a certain month. Combining such numbers with other metrics, such as the personnel capacity of this team or its financial budget, you end up with typical input-output relations referred to as performance indicators.

From an efficiency viewpoint, data such as the time needed from the initial contract request to first draft or to signed contract, may be of interest. It can be useful to generate KPIs on the overall size of the portfolio of active contracts, total value they represent, number of potential legal risks addressed with certain clauses, or number of new contracts added to the portfolio during a certain reporting period.

Note: *For more information on metrics and KPIs in the context of a legal department, please refer to our LLI Whitepaper “Data-driven Legal Departments”.*¹⁰

4.3. BENCHMARKING

Benchmarking can include metadata. Once you enter the stage described in the previous section and start working with data analytics and the corresponding KPIs, you may want to look for the comparable metrics or KPIs of other organizations, asking how they relate to similar organizations or contracting overall, or how the own organization’s negotiations differ from others.¹¹ This can give rise to improvement hypotheses, although it is important to verify whether data from other sources is reasonably comparable.

¹⁰ Cf. footnote 3; Access the whitepaper here: <https://www.liquid-legal-institute.com/workinggroups/legal-inhouse-kpis/>; DOI: 10.38023/e23ee12a-66d3-463a-bbd7-5cc374fcae9d

¹¹ One source of information about other organizations is associations such as the previously mentioned World Commerce & Contracting and their term reports.

5. WHAT TO EXPECT BEYOND METADATA

Metadata is extremely beneficial if the design is done well, but has certain built-in flaws. The more metadata you track, the higher the maintenance effort—and even with a “full set” of metadata you may run into unforeseen events that require you to filter contracts by criteria that simply have not played a role before (and, therefore, were never tracked).

As artificial intelligence technologies, such as natural language processing or neural networks, evolve, we may very well be headed for a future in which pre-defined and pre-collected metadata plays an ever smaller role in managing contract portfolios. One day, these systems may be able to process the entirety of your organization’s contracts and connected metadata, answering even complex questions about them in seconds. They will in particular have the ability to answer questions that cannot be answered on the basis of the available metadata alone but require an analysis of the contract text. Imagine being able to ask your contract lifecycle management system, and receive a detailed and confident answer in real-time:

- Show me all contracts that directly and indirectly relate to the country of Germany and categorize this list into what the nature of that relation is.
- Due to a change in the law, we are no longer allowed to deliver the product XYZ with the specification ABC. Do any of our contracts contain a penalty or other adverse consequence for us in this case?
- Could any of our contracts with customers potentially prevent us from using a new subcontractor, located in country ABC, for producing our product XYZ? If so, how could they prevent us from doing so?

6. APPENDIX 1: COMMON DATA FIELDS

There is no “one size fits all” answer to the question of what metadata fields to choose when setting up contract management. However, certain data fields have proven useful for a majority of organizations, and these are listed below. Note that not all organizations require all fields. Furthermore, certain fields provide redundant information and should therefore, in most cases, be used as alternatives to each other, not cumulatively.

6.1. THE CONTRACT

6.1.1. Unique identifier (ID)

What it does:	Gives each contract a unique identifier for disambiguation
Example values:	<ul style="list-style-type: none">• 1656987313• 2022-2283718
Preferred data type:	Number or text field
Preferred data source:	Generated automatically
Comments:	Allocating a unique ID number, or alphanumeric combination, allows for unambiguous identification of a contract and is a vital precondition for contract automation.

Some organizations or software providers might have a certain preference for such IDs, ranging from (i) pure consecutive numbers to (ii) elaborate data schemes including, for example, dates, country codes, legal entity names or other information. While we are sure that each organization or software provider has good reasons for doing so, we need to acknowledge that approach (ii) potentially creates redundancies with other data fields and—even worse—might be misleading or create inefficiencies when these other data fields are subject to change. Please don't forget that the purpose of a unique identifier is exactly what the name suggests. All other information is better placed in separate data fields, in order to ease the CLM process, reporting, data maintenance, etc.

Unless you have very specific reasons to follow route (ii), we would recommend approach (i).

The unique identifier is beneficial for other IT systems in your organization. For example, if you have a dedicated database for disputes, legal matters, KYC, etc. you might link the datasets to the contract via the IDs.

You might also want to consider adding data fields to track the ID numbers which relate to the contract in other systems, including the ID number your counterparty has given to the contract in their system.

6.1.2. Contract title

What it does:	Gives the contract a human-readable title or name
Example values:	<ul style="list-style-type: none">• Project Penguin Cooperation Agreement• ACME Supply Agreement EMEA
Preferred data type:	Text field
Preferred data source:	User input (ideally according to a naming convention)
Comments:	<p>The contract title is a peculiar data field in that it is one of the most common and at the same time least useful from a searching, filtering or reporting perspective. Its major purpose is to give the contract a name that helps human users refer to the contract—a purpose which the unique identifier (above) serves, but in a less human-friendly form.</p> <p>In order to achieve at least some degree of coherence in how users fill in this field, it is useful to formulate naming conventions, and list some dos and don'ts (e.g. discourage users from putting the names of the contract parties into the title, or any other information which is already collected through a dedicated data field).</p> <p>Instead of free text, the field could be automatically generated as a combination of other metadata, e.g. [party 1] [party 2] [contract type]. This would be readable for the user and helpful in interactions (maybe better than a numeric ID) but of course redundant because of the other data fields.</p> <p>In scenarios with mass imports of documents, you could consider taking the file name of the contract PDF as a first place holder, which could later be replaced by something more meaningful.</p>

6.1.3. Contract status

What it does:	Informs about the current status of the contract
Example values:	<ul style="list-style-type: none">• Initiated / requested• In drafting• In negotiation• Pending review / approval• Collecting signatures• Active (signed and in force)• Terminated / cancelled• Abandoned• Deleted
Preferred data type:	Value list
Preferred data source:	User input (ideally the status value is updated automatically according to related user actions, e.g. when approval or signature is requested)
Comments:	<p>The contract status is among the most important data fields as many questions relate to the status (e.g. knowing all active contracts that are signed and in force).</p> <p>The more the CLM process is streamlined and technically supported, the more the status can be derived automatically. From an automated request procedure, via generation from a template, to integration in an e-signature tool, digital processes can offer automatic triggers to update the status.</p>

	<p>One of the most difficult status transitions to catch reliably is the transition from “in negotiation” or “pending review / approval” to “terminated / cancelled / abandoned”. When a negotiation ends without the conclusion of a contract, human users tend to forget to revisit the contract in the system, which can lead to large numbers of forgotten contracts stuck “in negotiation” or “pending review / approval”, despite the fact that they have been long abandoned. Automated reminders for contracts which have spent a certain time “in negotiation” or “pending review / approval” can increase data quality—and greatly annoy users if the reminder comes too early. You should consider combining such reminders with automated status changes, e.g. after x months in status “draft/negotiation” the status changes to “abandoned”. Of course, users can always manually reset a status if the transaction is reactivated.</p>
--	---

6.1.4. Contract type

What it does:	Informs about the general type of contract
Example values:	<ul style="list-style-type: none"> • Construction contract • Employment contract • Finance agreement • Joint venture agreement • Lease agreement • License agreement • Non-disclosure agreement (NDA) • Partnership agreement • Purchase agreement • Sales contract • Service contract
Preferred data type:	Value list

Preferred data source:	User input (ideally automatically pre-filled when generating the agreement from a template in the CLM system)
Comments:	<p>In our experience, this is a field for which it is particularly relevant to consider usability and human bias. It is also a data field which is used very differently by various organizations.</p> <p>What makes this data field especially interesting is the fact that any attempt to categorize contracts into a finite list of types often results in a combination of layers of meaning, such as:</p> <ul style="list-style-type: none">• Legal contract archetypes, e.g. lease agreements, loan agreements, purchase/sale agreements, service agreements• Business- or Industry-related categories, which are often more granular than legal categories, e.g. financial institutions differentiate between many specific forms of loan agreement or other financing agreements• Categories of agreement that vary depending on the object of the agreement, e.g. most organizations have trouble designating a share purchase agreement as a “normal” purchase/sales agreement and prefer a separate category, or categories, for M&A topics• Hierarchical aspects, e.g. a master sales agreement in contrast to an individual agreement on the sale of goods• Categorizations relating to the role of the organization in the transaction, e.g. a sale agreement versus a purchase agreement (in each case seen from the viewpoint of the organization running the CLM system).

Often, this field comes with a legacy in the organization, and in such cases it may be a balancing act to decide whether to keep the list as is, adapt it gently, or replace it with a new list. People are used to the old list and every change induces effort.

Contract type lists also have a tendency to become ever longer if not properly governed. This increases granularity but makes it more difficult for users to consistently apply the right contract type. You need to keep the list short enough so that your users can use it competently, but long enough to allow for a meaningful degree of granularity.

Note that you can reduce the number of entries in the contract type list by moving certain information to separate data fields. For example, hierarchical aspects can be shown through contract relationships (see section 6.1.6) or moved to a dedicated data field (see section 6.1.5). The role of the contracting parties can also be shown via separate data fields (see section 6.2.3). However, these additional data fields come at a cost. They add to the number of individual choices your users need to make.

Another way to shorten the list is to include a “bucket” contract type such as “other agreement” or “general agreement”. Note that this only works if your users are highly competent and motivated to look for the right contract type. Otherwise, you might find 90% of your agreements in the “other” category, which renders it practically worthless. Ideally, you would set up a process for users to go through all “other” agreements at regular intervals to reallocate them.

A certain mixture of categorizations is practically inevitable, and not necessarily a problem. As with all data fields, the goal is not dogmatic consistency but utility. The ultimate purpose is to enable your users to filter your organization’s contracts or prepare reports.

	<p>Finally, a note on governance; the list of contract types typically changes over time, as new business models are implemented and regulatory requirements change. It can be helpful to give users an easy way of suggesting a new contract type. The suggestions should be carefully reviewed and only worthy candidates should be added to the list. It is a good idea to announce new entries to the users and ask them to consider assessing their contracts for reallocation.</p>
--	--

6.1.5. Framework agreements vs. individual agreements

What it does:	Separates contracts intended to cover more than one business transaction (framework agreements) from contracts covering only an individual business transaction
Example values:	<ul style="list-style-type: none"> • Framework agreement (for all services rendered by a provider) • Individual (purchase) order (covered by a framework agreement)
Preferred data type:	Value list
Preferred data source:	Automatically pulled from other systems (e.g. in sales or procurement where the systems provide it), or alternatively via user input
Comments:	Instead of a field such as "framework agreement vs. individual agreement", you could consider the alternative structure presented in the section "related contracts", which allows you to relate individual contracts to their framework agreement via "type of relation".

6.1.6. Relationships

Contracts often relate to other contracts or other data objects such as legal matters, disputes, claims, etc. It can be useful to track such relationships in the contract metadata.

6.1.6.1. Related contract/data object

What it does:	Indicates the relations of the corresponding contract to other contracts in the CLM or data objects.
Example values:	<ul style="list-style-type: none">• A unique identifier of related contracts or other data objects, typically an ID-number
Preferred data type:	Unique (alpha)numeric identifier or URL
Preferred data source:	Generated automatically, or via user input
Comments:	<p>This field might be used if a business relationship expands over several contracts, e.g. in M&A transactions which typically involve several contracts. You might also link a certain framework agreement to its subordinate purchase orders, contract amendments, or the termination of the initial contract.</p> <p>The field could also allow you to note that the contract plays a role in a litigation case, or point to a legal matter that contains important documents (e.g. memoranda) relating to the contract.</p> <p>In some cases, you might want to qualify such relations, covered by the next data field.</p> <p>Preferably this data field provides users with a directly link to the related object.</p>

6.1.6.2. Type of relation

What it does:	Sets out the type of relation between the linked data objects, and may show the type of linked data object
Example values:	<ul style="list-style-type: none">• Amendment agreement• Termination letter• Purchase order• Dispute <p><u>Alternatively</u>, the relation can also be expressed purely in hierarchical terms:</p> <ul style="list-style-type: none">• Superordinated• Co-ordinated• Subordinated
Preferred data type:	Value list
Preferred data source:	Other systems (e.g. in sales or procurement where the systems provide it), or alternatively via user input
Comments:	This field might be difficult to use due to the variety of possible values, but can be very valuable where there is a manageable set of possible entries.

6.1.7. Language

What it does:	Informs about the language the documents are written in
Example values:	<ul style="list-style-type: none">• en• es• fr
Preferred data type:	Value list

Preferred data source:	Automatically detected or, where the available tools do not allow for this and for correction, user input
Comments:	<p>The list should be based on ISO-639 language codes (consistently use only a single coding system, i.e. the two-letter ISO-639-1 or one of the three-letter ISO-639-2 or ISO-639-3 codes).</p> <p>Whether this value is useful depends highly on the organization's setup. It can, for example, serve to automatically route to experts fluent in the relevant language.</p>

6.2. PARTIES AND PEOPLE

6.2.1. *Contract parties*

What it does:	Informs about the parties to the contract
Example values:	<ul style="list-style-type: none"> • Example GmbH • ACME Corporation
Preferred data type:	Value list
Preferred data source:	CRM, ERP
Comments:	<p>The contract party should not usually be entered as free text, but rather linked to a master data for the respective legal entity. This increases data quality by avoiding typos and easing updates (e.g. only one address change is necessary for all contracts with a party).</p> <p>Contract parties can change their name (true for both companies and natural persons). Storing the company name not as hard text but as a link to</p>

	<p>a certain entity in a database helps with keeping the party names up to date (if the database is diligently maintained), but is not always trivial to implement technologically.</p> <p>Mergers and de-mergers of companies which are contract parties create additional challenges to data quality and should ideally be covered by internal data quality control processes.</p> <p>While we present this as one data field here, it would actually have to be split into several fields:</p> <ul style="list-style-type: none"> • Party 1 and • Party 2 <p>(and maybe an option to “add another contract party”). It can be very helpful to record whether the party is internal or external. Sometimes this is done by preparing data fields for “internal parties” and “external parties”.¹² A quality assurance process is required to retain the data quality in cases of acquisitions or divestments.</p>
--	--

6.2.2. *KYC Status*

What it does:	Data field summarizing the know your customer (KYC) process outcome
Example values:	<ul style="list-style-type: none"> • Pending • Initiated • Successful • Failed
Preferred data type:	Value list
Preferred data source:	CRM, ERP or a dedicated KYC database

¹² In this model, group-internal contracts have an empty “external parties” field and several entries for “internal parties”.

Comments:	Ideally, this should not be captured on a per-contract basis but a per-entity basis. Depending on the values established, it is possible to compile KPIs and generate automatic reminders in the used and connected systems, for internal and external use, to capture the relevant data.
-----------	---

6.2.3. Party's role

What it does:	Shows the capacity in which the party has entered the contract
Example values:	<ul style="list-style-type: none"> • Seller / buyer • Principal / contractor • Lessor / lessee • Lender / borrower • Guarantor
Preferred data type:	Value list
Preferred data source:	User input
Comments:	<p>There is a difference between knowing that ABC company has entered into a loan agreement, and knowing whether it is lending or borrowing.</p> <p>This field is only required when the setup does not indicate the setting otherwise. When the scope of your business or the scope of your CLM system clearly show this information (e.g. because you use different CLM systems for purchasing and for sales), this information may, with some degree of confidence, be automatically inferred from, for example, the contract type. In such cases we suggest not tracking this as additional field.</p>

6.2.4. Contract owner and/or owner division

What it does:	Informs about the human being(s) who is/are in charge of the contract
Example values:	<ul style="list-style-type: none">• Jane Doe
Preferred data type:	<ol style="list-style-type: none">1. Role designation (choosing from a value list of roles) or2. People picker based on a directory of users, e.g. Active Directory®
Preferred data source:	User input with suggestion (e.g. based on creation of contract object)
Comments:	<p>To simplify the contract management, it helps to assign just one responsible person, who can be designated by name or role. Ownership can be declared for organizational units too, but this should happen in conjunction with naming a human being as the owner.</p> <p>The system must be able to deal with the fact that the contract owner may unexpectedly leave the company or take an extended leave. Therefore, we propose having the role of the user assigned, which is transferred to the new holder, and establishing a look-up option for the replacement. While using the role is the preferred model, we acknowledge that it provides a certain additional complexity, as there needs to be a model for the roles.</p> <p>In larger companies it can be helpful to split this field further into "contract owner business", "contract owner legal", etc.</p>

6.2.5. Business context/department

What it does:	Informs about the internal organizational unit most closely related to the contract, which is particularly helpful for larger companies
Example values:	<ul style="list-style-type: none">• Procurement• Sales• R&D
Preferred data type:	Value list
Preferred data source:	User input
Comments:	<p>In many organizations, organizational units change their names, split in two or merge into one. For these organizations, it is worth considering how the metadata model reacts to such changes.</p> <p>Given the mixed character of many contracts you may allow for multiple selections.</p> <p>You should try to reduce the number of fields in this subsection to the bare minimum, for example if you capture the own role in the contract or the owning unit of your business, you probably won't need this field.</p>

6.2.6. Signatory

What it does:	Shows who has signed the contract
Example values:	<ul style="list-style-type: none">• John Doe
Preferred data type:	<p>People picker (if only organization-internal signatories are tracked)</p> <p>Free text field (ideally auto-filled from the e-signature process)</p>

Preferred data source:	User/employee directories for internal signatories User input/e-signature process for external signatories
Comments:	Very granular systems track not only who the signatory is but also in what capacity (e.g. as managing director, proxy-holder) the signatory is acting , see section 6.2.7.

6.2.7. Signatory role

What it does:	Shows the legal capacity in which the signatory is signing
Example values:	<ul style="list-style-type: none"> • Managing director (or other legal representative organ) • Commercial proxy holder (such as the German "prokurist") • Proxy holder
Preferred data type:	Value list
Preferred data source:	User input or e-signature workflow
Comments:	<p>It can be helpful to track the legal capacity in which the signatories of the contract are acting (e.g. to assess their legal signing power).</p> <p>It is typically easier to capture this data for your organization-internal signatories, because external signatories may not fit pre-defined categories. Allowing this data to be tracked as a free text field increases flexibility but makes it difficult to report on.</p>

6.2.8. Other stakeholders

What it does:	Apart from the direct parties to a contract, there are often other involved stakeholders, either within or outside your organization, which can be documented in this data field
Example values:	<ul style="list-style-type: none">• ACME Corporation• John Doe• Federal Ministry of Finance (Germany)
Preferred data type:	Predefined list or manually typed text
Preferred data source:	As above
Comments:	n/a

6.2.9. Contract initiator

What it does:	Names the person who initially set up the contract
Example values:	<ul style="list-style-type: none">• Jane Doe
Preferred data type:	Names of individuals, ideally a people picker from an Active Directory® or other data source
Preferred data source:	Generated automatically
Comments:	You may also consider tracking (i) who asked for the contract to be set up, (ii) who provided the first draft of the contract, or (iii) who participated in negotiating it.

6.3. LEGAL-SPECIFIC DATA POINTS

6.3.1. *Applicable law/jurisdiction*

What it does:	Informs about the law that applies to the contract
Example values:	<ul style="list-style-type: none">• Brazil• Germany• England & Wales• France• USA: New York• Not stipulated in the contract text
Preferred data type:	Value list
Preferred data source:	User input (ideally a user well versed in legal matters with the knowledge to determine the applicable law)
Comments:	<p>Certain countries have multiple jurisdictions, either geographically scoped (e.g. USA) or depending on other criteria. This should be reflected in your pre-defined list.</p> <p>For cases where the applicable law is not clear, the field could either be left blank or an option such as “not stipulated in the contract” or “applicable law unclear” could be provided.</p> <p>While oftentimes a country (e.g. Germany) is sufficient to indicate the applicable law, some countries, such as the USA, may require differentiation by state (e.g. USA: Delaware).</p>
	<p>It is advisable to draw from existing formal standards such as ISO 3166-2 (available at https://www.iso.org/, last accessed 10 March 2023) to populate country lists, or rely on extensions made specifically for the legal field, for example noslegal’s taxonomy v1 “Legal Places” (available at https://www.noslegal.org/, last accessed 10 March 2023).</p>

6.3.2. CISG applicability

What it does:	Informs whether the United Nations Convention on Contracts for the International Sale of Goods (CISG) rules apply to the contract
Example values:	<ul style="list-style-type: none">• True• False
Preferred data type:	Boolean (yes/no)
Preferred data source:	User input
Comments:	n/a

6.3.3. Place of jurisdiction

What it does:	Informs about the place of jurisdiction
Example values:	<ul style="list-style-type: none">• Hamburg, Germany
Preferred data type:	Value list (fallback option, text field)
Preferred data source:	User input (ideally a user well versed in legal matters with the knowledge to determine the applicable law)
Comments:	Free text fields are often used for this value, but they do not enforce normalization or unambiguity (e.g. there are several places called Hamburg world-wide). On the other hand, it is very challenging to maintain a list of pre-defined choices that potentially has to cover all cities in all countries.

	<p>Similarly to the “applicable law” field, standards can offer some solutions. Options include SALI LMMS Dispute Venue¹³ (insofar as applicable), ISO standard location codes, such as ISO 3166-2 or UN/LOCODE. In any case decide on a consistent mode (do not mix ISO standards or an ISO standard and UN/LOCODE).</p> <p>If you decide to track this via a text field, consider asking users to at least name both city and country.</p>
--	---

6.3.4. Liability provisions

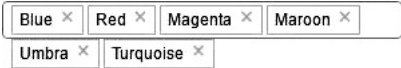
What it does:	Identifies contracts without a limited liability or limits beyond certain thresholds (which need to be set according to your company needs)
Example values:	<ul style="list-style-type: none"> • Limited • Not limited • Above [the defined threshold] [currency units]
Preferred data type:	This could be a Boolean field indicating if liability is limited or beyond the defined threshold (see comments section for alternatives)
Preferred data source:	User input
Comments:	If the options shown above are not granular enough for your purposes, you might use a numeric currency field, stating the exact amount of such limit measures in currency units.

¹³ The SALI taxonomy is available through their website <https://sali.org/>, best via WebProtégé, last accessed 7 July 2022

6.3.5. Confidentiality level

What it does:	Shows the degree to which the contract is confidential within the organization, determining who can be granted access, and clarifies the level of confidentiality or measures required, which, depending on your company needs, might be structured as a list of people or departments or pre-defined categories
Example values:	<ul style="list-style-type: none">• Public• Internal (meaning accessible to the entire staff of your company)• Confidential (to a pre-defined group of people or departments)• Highly confidential (to a closely limited group, with no access for other user groups)
Preferred data type:	Value list
Preferred data source:	User input based on contract language, especially the confidentiality clause, or internal policy
Comments:	<p>Confidentiality requirements might result from the agreement of the contracting parties (e.g. a confidentiality clause) or other regulations. Especially if you handle large numbers of contracts, you might consider an approach with pre-defined categories instead of case-specific entries in the database.</p> <p>If the access is restricted, saving authorized roles is recommended to implement restrictions technically.</p>

6.3.6. Markers or tags for special cases or circumstances

What it does:	Identifies contracts fitting to the defined criteria
Example values:	<ul style="list-style-type: none"> • Key contract / no key contract • High-risk contract / low-risk contract • Relevant / not relevant • Exception / standard • Project XYZ / project ABC • Internal audit 2022 / internal audit 2023 / internal audit 2024
Preferred data type:	<p>Multiple-choice value list (tags)</p> <p>Individual marker shown as a Boolean (yes/no) variable or checkbox</p>
Preferred data source:	User input (AI-guided suggestions based on analysis of contract text)
Comments:	<p>Depending on your company needs, you might be required to identify contracts that fit certain criteria. Often, there are several such criteria shown as pillbox fields or tags which allow for flexible data entry.</p>  <p>This allows multiple criteria to be tracked by just one data field. The available criteria or tags can either be pre-defined in a centrally managed value list or users could be allowed to create their own tags ad hoc. The latter offers a great degree of flexibility and can feel empowering for users. Ideally, it leads to the emergence of highly relevant filter criteria, in real time, without complex governance efforts. It can, however, lead to a chaos of</p>

	<p>cryptic, inconsistent, or even redundant criteria which ultimately don't add value to the system, but this risk can be mitigated by regularly reviewing the list of tags.</p> <p>Alternatively, each criterion can be shown in an individual data field as a yes/no checkbox.</p>
--	--

6.4. COMING INTO FORCE

6.4.1. *Signature date*

What it does:	Indicates the date and time, when the contract is fully signed (i.e. the date of the last required signature)
Example values:	<ul style="list-style-type: none"> • 1 January 2023
Preferred data type:	Date/time
Preferred data source:	Ideally this should be extracted from electronic signing solutions, but organizations that predominantly rely on ink-on-paper signature would have to fill the data field manually
Comments:	<p>You need to assess whether there is a need for more than one date field, for example if the contract parties signed on different dates. In cases of doubt, when you have several dates but want to limit yourself to only one data field, we recommend using the last signature date, because this is the earliest point at which the declarations of intent are duly documented for all contract parties.</p> <p>More granular models track the date of every signature on the contract and the respective signatory.</p>

6.4.2. Start date / effective date

What it does:	Indicates the date and time when the contractual obligations start
Example values:	<ul style="list-style-type: none">• 12 March 2025
Preferred data type:	Date/time
Preferred data source:	User input
Comments:	It is best to ask users to input this date only after the contract is fully signed, or at least switch the data field to mandatory only after such time.

6.5. CONTRACT DURATION

Note: *While defining a contract metadata model can be a somewhat abstract exercise, the technical boundaries of the environment dictate what you can and cannot do—and therefore what deserves your attention. This is especially true for the data model governing the duration, termination and prolongation of a contract. This part of the metadata model is often hard-wired into the CLM platform with limited room for individual preferences. The information below should therefore be seen as inspiration for what is possible or desirable—and should be closely aligned with the capabilities of your CLM platform.*

6.5.1. Termination

Simple metadata models treat the various termination rights as general reminders, tracking at minimum a reminder date and some reminder text. This simple approach can be realized even in the most limited technical environments, but requires some manual effort to maintain. The reminder dates need to be adjusted periodically as the window of opportunity to terminate the contract shifts from period to period.

Most CLM platforms offer a more complex data model. Ideally, the metadata model supported by your CLM platform should differentiate between fixed and indefinite contract duration. It should allow an automatic calculation of the next point in time at which the contract could be unilaterally terminated as well as the deadline by

which the termination would have to be declared given the notice period. Once the deadline approaches, appropriate users should be notified. If the deadline passes without action being taken, the system should automatically switch over to the next period.

6.5.1.1. Fixed end date

What it does:	Indicates the date on which the contract is set to end automatically (unless action is taken)
Example values:	<ul style="list-style-type: none"> • 2 October 2029
Preferred data type:	Date/time
Preferred data source:	User input (user friendly user interfaces allow both the direct entry of a date or a calculation involving a starting date and a duration expressed in days, weeks or months)
Comments:	<p>Contracts might explicitly state a calendar date or describe this indirectly as a term (e.g. three years) which needs to be added to the start date, signature date or another point in time. In such cases you are recommended to calculate the relevant end date and input an explicit date into the system.</p> <p>In cases without a clearly defined end date (indefinite term), the field could stay empty or state “not applicable” or “indefinite term”. Some models also track “fixed term” as a Boolean (yes/no) data field.</p> <p>For contracts with a clearly set date but an automatic renewal, you might want to consider stating “not applicable” as the factual duration is indefinite unless terminated. The possibility to cancel such automatic renewal can be captured by the “date of next termination right”.</p>

6.5.1.2. Date of next ordinary termination right

What it does:	Indicates the next date at which termination can be unilaterally (ordinarily) declared
Example values:	<ul style="list-style-type: none">• 2 October 2029
Preferred data type:	Date/time
Preferred data source:	Calculated from other data fields "termination notice period" and "termination effective date cadence"
Comments:	<p>This field helps you not miss any termination right by showing the next date by which termination action must be taken so that the notice period is observed.</p> <p>It is useful to add another data field "ordinary termination not possible before" so that an initial fixed term can be observed.</p> <p>Consider triggering a reminder ahead of this date, which should be sent early enough to allow your organization to come to a decision about whether to terminate and prepare and declare the termination.</p>

6.5.1.3. Next possible termination effective date

What it does:	Allows for the calculation of the next possible termination effective date (i.e. the date on which the contract would end if termination was duly declared)
Example values:	<ul style="list-style-type: none">• 31 December 2025
Preferred data type:	Date/time

Preferred data source:	Calculated from “termination effective date cadence”, “termination notice period” and current date
Comments:	This date informs users of the next date on which a termination can become effective if declared.

6.5.1.4. Termination effective date cadence

What it does:	Allows for the calculation of the next possible termination effective date (i.e. the date on which the contract would end if termination was duly declared)
Example values:	<ul style="list-style-type: none"> • To end-of-year • To end-of-quarter • To end-of-month • To [date] of each year • To [day] of each month
Preferred data type:	Value list
Preferred data source:	User input
Comments:	The value list of options available should be closely aligned with the termination options in your contracts.

6.5.1.5. Termination notice period

What it does:	Indicates how long in advance of the next termination effective date a termination notice must be given
Example values:	<ul style="list-style-type: none"> • 30 days • 3 months

Preferred data type:	Technically speaking, the termination notice period consists of two data fields, a number and a period (days/weeks/months/years)
Preferred data source:	User input
Comments:	The value list of options available should be closely aligned with the termination options in your contracts.

6.5.1.6. Termination comments

What it does:	Allows for human-language explanatory comments on termination
Example values:	<ul style="list-style-type: none"> Termination should be addressed to Mr. Example at Example Corporation
Preferred data type:	Text field
Preferred data source:	User input
Comments:	Even the most granular data model might not be able to capture every nuance of the agreed termination provisions. It is therefore helpful to create a space for unstructured user input.

6.5.1.7. Termination notice form

What it does:	Shows the form in which termination must be declared
Example values:	<ul style="list-style-type: none"> Written form Written form, sent by registered mail Text form
Preferred data type:	Value list

Preferred data source:	User input
Comments:	This can be combined with the termination comments (see above) and guides the user's thinking specifically to the form of termination.

6.5.1.8. Excursus: Data on actual termination

Once a termination is declared (regardless of whether by your organization or the counterparty) the metadata model should track, for each declared termination:

- when it was declared
- by whom it was declared
- when was is received by the respective recipient
- what effective date of the termination was declared
- whether it is deemed legally effective (in other words, whether the system should treat this termination as effective and calculate the end date of the contract accordingly).

A link to the respective documents should be available.

6.5.2. Prolongation

Similar to termination, simple models should simply track a date, with a comment, to show when to exercise a prolongation option. More advanced CLM platforms allow for a data model which keeps track of the next deadline for declaring a prolongation, as well as the effect of the prolongation (for how long the term is prolonged). It notifies users accordingly, tracks whether prolongation is effectively declared, reflects this in an updated contract duration, and switches the deadline for exercising the next prolongation option.

The data fields are very similar to those relating to a termination, so you may draw inspiration from the model above.

6.6. BUSINESS-SPECIFIC DATA POINTS

While this paper addresses contract metadata from a legal perspective, there are many data points with a “hybrid character”, i.e. they might not be purely legal but there might be legal implications. Some of these data points are financial in nature or relate to certain processes in productions, logistics or other areas. Therefore, these data points are often tracked in other systems, e.g. enterprise resource planning (ERP), customer relationship management (CRM), accounting, etc.

You should consider carefully whether you need to include these in your data model and whether they need to be tracked in your own database, or whether a reference/interface to other systems would be sufficient.

6.6.1. *Event triggers and notifications*

What it does:	Tracks dates of certain events relating to the contract and explains their relevance
Example values:	<ul style="list-style-type: none">• 01 May 2028 – Send special report to contract party as agreed in section 3.5 of the contract• 31 March 2024 – According to section 12 of the contract, this is the last day to request training for the following year
Preferred data type:	Date/time and text field
Preferred data source:	User input, ideally aided by AI-powered suggestions
Comments:	<p>While the combination of a date field and a free text field provides a very flexible solution, you might opt for a more structured solution, especially if the trigger events are very homogeneous or fall into certain categories, in which case you might opt for a drop-down list of pre-defined options. Depending on your needs, this might replace or supplement the free text field.</p> <p>Ideally, your CLM system would offer notifications to users when deadlines are nearing.</p>

6.6.2. Contract volume / amount / value

What it does:	Describe the absolute value of the contract in terms of money, goods or services provided there-under
Example values:	<ul style="list-style-type: none">• EUR 1,000,000.00
Preferred data type:	Currency amount
Preferred data source:	Ideally, filled automatically by a link to a supplier or customer account in the EPR/accounting system, otherwise user input
Comments:	In some scenarios you need to know how much money is involved in a contract or what the contract is worth. This information requires careful prior definition, since amounts may vary over time and are often unclear at the beginning. You should decide whether to refer to, for example, sales volume or EBIT contribution, in absolute numbers or net present value.
	While we acknowledge the need to put a price tag on a contract or scenarios where you need to separate "high value" and "low value" business relationships, we recommend pulling this data from elsewhere. Ideally the contract is linked to the company's ERP/accounting system and therefore financial data might be drawn from there, although it might not reflect the value correctly in the initial period of the contract (e.g. due to orders only being made over time).

6.6.3. Product (hardware) related data

What it does:	Describes the products to be delivered under the contract in qualitative terms (e.g. product description, product ID) and quantitative terms (e.g. number of items)
Example values:	<ul style="list-style-type: none">• 1,000 units of product ID 123456
Preferred data type:	A combination of two fields identifying (1) the product and (2) the number of products
Preferred data source:	Ideally from another IT system, such as ERP, product catalogue, etc.
Comments:	n/a

6.6.4. Involved countries/geographies

What it does:	Defines the country or countries to which the contract relates
Example values:	<ul style="list-style-type: none">• EN• DE• US
Preferred data type:	A predefined list of countries (e.g. according to ISO 3166 standard codes for countries and their subdivisions) which includes multi-select.
Preferred data source:	User input, ideally aided by AI suggestions
Comments:	Unlike the applicable law or jurisdiction, this meta-data concerns any potential relation to a country or region (e.g. shipping locations or location where a service is rendered). This can be very important to quickly identify contracts potentially impacted by trade regulations, sanctions, regional crises, specific tax regimes, etc.

6.6.5. Payment terms

What it does:	Determines when the supplier must be compensated for services
Example values:	<ul style="list-style-type: none">• Payable after delivery• Payable upfront
Preferred data type:	Value list
Preferred data source:	User input (potentially automatically suggested by the accounting system, although this would likely require manual validation)
Comments:	The value list should include common payment terms in your organization. It is prudent to include a choice for “non-standard payment terms” in case the contract contains payment terms not in your value list.

6.6.6. Pricing clauses (e.g. index clauses)

What it does:	Informs whether the contracting parties agree on price escalation clauses, refer to price indexes (CPI), fix volume discounts or other special conditions
Example values:	<ul style="list-style-type: none">• CPI-based price escalation• Volume-based discount• Non-standard pricing clause
Preferred data type:	Value list of common clauses used in the business, including a non-standard pricing clause (one not on the value list) or no pricing clause Potentially additional fields to capture common percentage values or reference values (depending on the clauses typically used)

Preferred data source:	User input (ideally AI-guided suggestions based on analysis of contract text)
Comments:	Such clauses are designed to offset extreme fluctuations and avoid unfairness, ultimately preserving the contractual relationship even in times of crisis. This is why they are often written as optional.

6.6.7. Commodity

What it does:	Tracks the commodities the (procurement) contract relates to
Example values:	<ul style="list-style-type: none"> • Hardware • Software • Services • Small moveable goods
Preferred data type:	Value list
Preferred data source:	User input (potentially automatically suggested from contract template, although this would require manual validation, especially when multiple commodities are concerned)
Comments:	The same applies to sales (e.g. direct, indirect distribution channel).

7. APPENDIX 2: STANDARDS AND TAXONOMIES USABLE FOR CLM METADATA

7.1. CONTRACTUAL FEATURES

Where reasonable, you should align your concrete field definitions with existing standards, such as:

- those already in use in your organization (e.g. for products or services purchased under contracts)
- legal taxonomies
- standard contractual terms.

We reference some relevant examples in this appendix, but please note that the list is not exhaustive and there are probably many more standards available.

7.1.1. *Legal taxonomies*

General purpose legal taxonomies	
noslegal ¹⁴	A taxonomy of legal places (geographical descriptions) which are relevant for contract classification and the further fields of legal perspectives, legal work and subjects which are probably less relevant to CLM metadata.
SALI's ¹⁵ Legal Matter Standard Specification (LMSS)	A taxonomy of legal matters, although more relevant to matter management than contract management.

¹⁴ <https://noslegal.org/>, last accessed 25 March 2022, currently available in version 1.

¹⁵ <https://sali.org/>, last accessed 25 March 2022.

Industry related	
Open Contracting Data Standard ¹⁶	A taxonomy of public procurement, used to publish public procurement data in an unambiguous, machine-readable format. While certain legal contract related fields exist, it generally addresses the content of transactions more than legal terms.
Public Contracts Ontology ¹⁷	A taxonomy used to describe contract-relevant information in public contracting

For a more comprehensive list, you can review the Liquid Legal Institute e.V.'s list of Legal ontologies.¹⁸

7.1.2. *Standard contractual terms*

Standard terms exist in various forms, often specific to a certain field.

ISDA ¹⁹	The financial derivatives industry uses a common domain model, clause library and clause taxonomy.
VOB/B, VOL/B	Standard terms for German public procurement of construction works and general services.
GDPR	Standard contractual clauses in EU privacy law, both for international transfers and commissioned data processing.
FIDIC ²⁰	Conditions of contract for the construction industry.

¹⁶ <https://standard.open-contracting.org/>, last accessed 24 March 2022.

¹⁷ https://ebrary.net/20619/_computer_science/public_contracts_ontology, last accessed 26 March 2022.

¹⁸ <https://github.com/Liquid-Legal-Institute/Legal-Ontologies>, last accessed 10 March 2023.

¹⁹ For the taxonomy see <https://www.isda.org/2019/09/04/isda-taxonomy-2-0-finalized/>, an introduction to the common domain model is available at <https://www.isda.org/a/z8AEE/ISDA-CDM-Factsheet.pdf>. These works are the basis for automated contract creation and negotiation collaboration, for which purpose the industry association has created a tool with a law firm, <https://www.isdcreate.org/>. Pierre Schammo gives an introduction to the initiatives and their relation to distributed ledger technologies and smart contracts in his paper "Of standards and technology: ISDA and technological change in the OTC derivatives market", available at <https://doi.org/10.1080/17521440.2022.2063030> and <https://ssrn.com/abstract=4101117>, all links last accessed 10 March 2023.

²⁰ <https://fidic.org/>, last accessed 10 March 2023.

Standard licenses for free and open content ²¹	The Open Source Initiative provides a list of free and open content.
OneNDA ²²	Standard NDAs.

Besides standard contracts, it might be useful to refer to the concrete standard terms covering certain aspects of deals, such as INCOTERMS®²³ regarding delivery terms (covering practical responsibility, cost and risk of transport and import/export).

Regulatory requirements can also set contractual requirements or features. In such cases the applicability of certain rules is a relevant data field. An example of such a requirement is the EU taxonomy on sustainable investment²⁴

7.2. OTHER LISTS OR REFERENCES

Where you reference topics outside the contract itself, it is advisable to use the most standardized references available. For case law within the EU, for example, we suggest using the European Case Law Identifier (ECLI).²⁵

ISO codes can help you populate lists containing languages (ISO-639, see section 6.1.7) or countries (ISO-3166, see section 6.3.3).

²¹ The Open Source Initiative keeps a list of licenses approved as “open”, available via <https://opensource.org/licenses>; for non-software content, Creative Commons Corporation, a non-for-profit, publishes a set of standard licenses aligning with various requirements, available at <https://creativecommons.org/>; links last accessed 10 March 2023. For such licenses we recommend a reference to the applicable license as one field of your CLM.

²² <https://onenda.org/>, last accessed 10 March 2023; there is a follow-up initiative to draft a standard data processing agreement for GDPR commissioned data processing, called oneDPA.

²³ <https://iccwbo.org/resources-for-business/incoterms-rules/>, last accessed 10 March 2023; the CLM should not only capture the fact that an INCOTERM applies but should include the concretely chosen term (e.g. EXW for ex works delivery).

²⁴ https://ec.europa.eu/info/business-economy-euro/banking-and-finance/sustainable-finance/eu-taxonomy-sustainable-activities_en, last accessed 10 March 2023.

²⁵ See https://e-justice.europa.eu/content_european_case_law_identifier_ecli-175-en.do for details, last accessed 10 March 2023. The ECLI enables a quick search for and reference to cases both at the EU level and from EU member states, both via https://e-justice.europa.eu/content_ecli_search_engine-430-en.do, last accessed 10 March 2023.

8. ABOUT THE AUTHORS



Baltasar Cevc

Liquid Legal Institute e.V. and fingolex

Baltasar Cevc is a lawyer and entrepreneur with ample IT experience. His key legal interests lie in the IT and health industries with a focus on IT law, AI, security and privacy. He researches methods of legal work and their improvement potential, both in terms of effectiveness and efficiency, currently in the fields of business process management and the use of data in legal services.



Dr. Maik Ebersoll, LL.M.

Head of Legal Operations Bosch Group, Liquid Legal Institute e.V., AWT Institut für ökonomische Systemtheorie e.V.

Maik studied law and economics and very much enjoys the interface between business, IT and legal. His research focuses on macroeconomics, quantitative systems theory, legal management, operations research and homomorphisms of Gibbs-Falkian thermodynamics. You can reach Maik via Maik.Ebersoll@awtinst.org



Klaus Gresbrand

Partner at Deloitte Legal Rechtsanwaltsgesellschaft mbH, Liquid Legal Institute e.V.

Klaus Gresbrand is a lawyer and partner at Deloitte Legal. He aims to combine his 10+ years of experience as a corporate/ M&A lawyer with an affinity for technology in order to enable legal departments to work more efficiently, and joyfully so. You can reach Klaus via kgresbrand@deloitte.de



Susanne Marks

Co-Head Legal Transformation Advisory, smartvokat GmbH, Liquid Legal Institute e.V., Women in eDiscovery, Digital Media Women e.V., Legal Innovation Rhein-Ruhr

Susanne is part of smartvokat's Legal Transformation Advisory team and supports industrial clients to optimise their legal & compliance processes, and legal departments & law firms on their journey of digital transformation. You can reach Susanne at susanne.marks@smartvokat.com



Kai Jacob

Co-CEO of the Liquid Legal Institute e.V. and Partner at KPMG Law

Kai Jacob is co-founder and co-chair of the Liquid Legal Institute. He has been a Partner in Legal Operations & Technology Services (LOTS) at KPMG Law since 2021. Kai is involved in various projects and initiatives dealing with agile working in the legal department, the state of health in the legal market, and the future of legal managed services. Before joining KPMG Law, Kai Jacob was a partner at Deloitte Legal and before that worked for many years at SAP SE. Kai Jacob was admitted to the bar in 2004 and studied law in Marburg, Göttingen and Osnabrück.



Jutta Löwe

Group Privacy Officer at Brenntag SE; Liquid Legal Institute e.V.

Jutta is an in-house lawyer and Group Privacy Officer at Brenntag SE and Head of the Global Data Privacy Organisation. Before joining Brenntag, she worked for various companies, in-house and in her own law firm. Since her studies, her professional focus has been on strategic and legal advice on all aspects of digitalization. Jutta is passionate about Innovation and an expert in translating legal to business to IT and back. Since 2020, she has been an active member of Liquid Legal Institute e.V. where she is involved in several projects.



Dmitri Geidel

Lawyer at Deloitte Legal Rechtsanwaltsgesellschaft mbH, Liquid Legal Institute e.V.

Dmitri Geidel enjoyed university so much, that he stayed extra-long and has now a legal and an economics degree. Currently he is deeply absorbed with designing new kinds of contracts. In his remaining time he works as a corporate lawyer in Berlin. You can reach Dmitri via dgeidel@deloitte.de