Managing Access to Language Resources in a Corpus Analysis Platform

Eliza Margaretha Illig
Department of Digital Linguistics
IDS Mannheim, Germany
margaretha@ids-mannheim.de

Nils Diewald
Department of Digital Linguistics
IDS Mannheim, Germany
diewald@ids-mannheim.de

Paweł Kamocki
Department of Digital Linguistics
IDS Mannheim, Germany
kamocki@ids-mannheim.de

Marc Kupietz
Department of Digital Linguistics
IDS Mannheim, Germany
kupietz@ids-mannheim.de

Abstract

Corpus query tools are crucial to CLARIN's mission of facilitating the sharing and use of language data for research. It is a huge challenge for online corpus platforms to manage user access rights for large corpora with complex licenses and heterogeneous restrictions on access methods and purposes. This paper presents an approach to maximize user access to corpus data while protecting rights holders' legitimate interests. Query rewriting techniques and authorization procedures allow for modeling license terms in detail, enabling broader applications. This offers an alternative to methods that only model a greatest common denominator of licenses, thereby limiting the possibilities for using the data. Our approach constitutes a flexible and extensible corpus license and user rights management component applicable for other language research environments.

1 Introduction

The core value of CLARIN is to accommodate the reuse of language data and tools for research. In pursuing this objective, CLARIN, and linguistics in general, face the challenge that its research data are typically affected by the rights of third parties. One approach to dealing with this is to use technical measures that, on the one hand, ensure that the interests of the rights holders are not infringed and, on the other hand, restrict the use of the data as little as possible. This is typically done using an online corpus query system, which only allows indirect access to the data.

Provided that uniform licenses are available, corpus concordancers, for example, allow authenticated users who have agreed to their terms of use to view keywords in context (KWICs) without allowing full-text reconstruction. The situation becomes more difficult when – as is often unavoidable – different licenses and rights exist for different parts of the data and different groups of users, or when close reading of KWICs is not the only use case. Our paper presents the KorAP (Diewald et al., 2016) approach to making very large corpora, such as the German Reference Corpus DeReKo (Kupietz & Lüngen, 2014), which is affected by more than 200 partly heterogeneous licenses and is used in very different contexts, as usable as possible, while safeguarding the legitimate interests of rights holders.

1.1 Corpus Licenses

Providers of corpus analysis platforms typically do not hold any property rights in language resources which they allow the analysis of. They usually obtain the rights to use the resources via contractual arrangements with the rights holders, which can generally be referred to as licensing agreements. These agreements allow the provider to grant limited access to the resources to the platform users, or certain groups thereof.

Language corpora represent significant economic value (which has only increased with the advent of LLMs, and is bound to increase even further), and require substantial investment (of time, qualified effort and money) to build. In order to protect this value and investment from (perceived) 'free riders' (Olson, 1965) and potential competitors, rights holders often choose to restrict access by certain categories of users, or restrict certain uses (e.g., downloading *n*-grams).

Licensing agreements define by whom, for what purposes and how a resource can be used. They can be divided into those limited to academic uses, i.e. teaching and research, sometimes only in certain

fields (such as linguistics, as opposed to journalism or media science) and those allowing commercial uses. 'Academic' licenses are generally less costly to obtain, but much more restrictive. For example, they may allow access only to users affiliated with a research organization, with an authenticated account (this also applies to automated access on behalf of the user). Further restrictions in both types of licenses may include access only via a dedicated platform or API, or even access only from a specific physical location or via a specific network.

Popular public licenses, such as Creative Commons (CC), do not discriminate between groups of users allowing resources to be available on online corpus analysis platforms without authentication. However, even CC licenses contain restrictions on re-use, ranging from simple acknowledgement of the source (BY), through the requirement to retain the original license in any modified versions (SA, or 'share-alike'), to the prohibition of making any modifications (ND, or 'no-derivatives'), to the prohibition of any commercial use (NC).

Statutory exceptions in applicable copyright legislation can also be interpreted as 'licenses' *sui generis*, i.e. permissions granted not by the rights holder, but directly by the legislator. The statutory exceptions for Text and Data Mining (TDM), harmonized at the EU level by the Digital Single Market (DSM) Directive (2019/790), deserve special attention. Article 3 of the DSM Directive allows research organisations to build corpora for TDM purposes, which (at least in certain EU jurisdictions such as Germany) can subsequently be shared with research partners.

Certain metadata and annotations can be licensed. Even though in general metadata would rarely attract copyright protection, they may still be protected by the *sui generis* database right. Moreover, some metadata may contain elements such as abstracts which typically qualify for copyright protection. Since the perceived economic value of metadata is low, licenses for metadata tend to be more liberal (oftentimes, a waiver of rights such as CC0 is used). On the other hand, many rightholders are unaware of the possibility to license metadata separately, and hence the metadata of many resources are not accompanied with any licenses.

1.2 User Rights Management

Corpus licenses determine which users have which access rights to which parts of primary data, metadata, and annotation data (the latter being determined by software licenses as well). The rights of a user therefore have to be managed in addition by a corpus platform and can be matched with the licenses after authentication and before any data can be delivered to an account. As previously introduced, these licenses determine not only whether, but also in which ways a user can access the data. This requires that access rights must not only be compared statelessly and statically, but also take into account the temporal and local contexts. For example, if licenses only allow short excerpts from texts (e.g., KWICs), there is a reasonable concern of licensors that the original full-text can be reconstructed from the search or analysis results by cleverly formulating follow-up search queries.[1] In order of prohibiting such use, it may be necessary to monitor an account's search queries over time to detect and/or prevent misuse.

In addition to static permissions of an account, further factors such as location and time can initiate dynamic restrictions. Some corpus licensors also limit the availability of their data to sites within a specific location or network. The user rights management of an online corpus platform must therefore be able to match the IP address of the account with the address space allowed for access. If licensors make licenses available to users only for a limited period of time, the user rights management system must log the initial access and check with each access whether the approved time frame has not yet been exceeded.

Users are permitted to utilize corpus data shared under TDM exceptions for text and data analysis in non-commercial research. To ensure this prerequisite is met, it is reasonable to implement a policy that restricts access to these resources on a request-only basis. In straightforward scenarios, for instance when the data is bound to a particular project, it is adequate and appropriate to host these data in a separate instance of a corpus system, ensuring that only users with approved requests can log in and access them.

author	Rax, u.a.	availability	CC-BY-SA	corpusEditor	wikipedia.org
corpusSigle	WUD17	corpusTitle	Wikipedia	creationDate	2016-11-14
docSigle	WUD17/B96	docTitle	Wikipedia, Benutzerdiskussionen mit Anfangsbuchstabe B, Teil 96	editor	wikipedia.org
externalLink	Wikipedia	foundries	• corenlp • corenlp/constituency • corenlp/morpho • corenlp/sentences	indexCreat…	2019-02-27
indexLast…	2019-02-27	language	de	pubDate	2017-07-01
pubPlace	URL:http://de.wikipedia.org	publisher	Wikipedia	reference	Benutzer Diskussion:Blurry dun, In: Wikipedia - URL:http://de.wikipedia.org/wiki/
textClass	• staat-gesellschaft • biographien-interviews	textSigle	WUD17/B96/59253	textType	Benutzerdiskussionen

Figure 1: Metadata fields of a text in DeReKo can be used to create a virtual corpus in KorAP. The availability field is particularly used to enforce access policies for DeReKo.

2 Related Work

Some corpus platforms allow users to build and work on their own corpora, for instance *Sketchengine* (Kilgarriff et al., 2014) provides corpus building tools to upload or find texts from the web. KorAP provides pre-defined virtual corpora, that are collections of texts dynamically assembled based on certain criteria such as a list of text identifiers (`textSigle`). Additionally, KorAP supports functionalities enabling users to create their own virtual corpora on the fly by filtering corpus metadata such as author name or publication year. Figure 1 presents some metadata fields of a DeReKo text. One or more virtual corpora can be used in a search by adding a reference to their identifiers, e.g. `referTo ratskorpus` illustrated in Figure 2.

In digital rights management, Open Digital Rights Language (ODRL; Iannella and Villata, 2018) is commonly used to represent policies on the use of digital content and services. It is used to define permissions, prohibitions, and obligations between parties (resource owners, users) and assets (resources), actions on assets, and constraints. It can be expressed in various serialization formats including XML and JSON-LD (Sporny et al., 2014), a lightweight Linked Data format that enables semantic definitions of the data. Comparable to an ODRL model, KorAP uses a JSON-LD based representation to describe user queries on virtual corpora (assets) on which the queries are applied, and access policies (contraints) on them. The constraints are dynamically adjusted by means of query rewriting according to authentication, authorization and access location (see Section 3). Authentication is the process of verifying a user's identity, while authorization happens after authentication to determine which resources and actions the user have access to. Furthermore, authorization scopes define access permissions that can be granted to an application. In our case, permissions on assets are not included in the representation but incorporated as authorization scopes. While ODRL focuses constraints on parties, assets or actions, our approach emphasizes constraints based on licenses required to access DeReKo.

In authentication and authorization management, Shibboleth (Cantor & Scavo, 2005) is a common identity and access management (IAM) system used in academic and research communities including CLARIN. It supports Single Sign-On (SSO) typically used to allow academic users to use their institutional logins to access corpora with academic licenses provided by corpus platforms such as OpenSoNaR (Reynaert et al., 2014). Besides, Lightweight Directory Access Protocol (LDAP; Howes and Smith, 2006) provides a central location to store information such as user details, in a hierarchical structure (directory), and defines a way to verify user credentials and determine user permissions to access systems or resources. While Shibboleth and LDAP provide foundational authentication mechanisms, they are inherently limited

[1]This is also a concern for corpus use in language models.

Figure 2: The web UI Kalamar displays the virtual corpus menu with two criteria defining a virtual corpus: the *availability* metadata field for various CC license types, and a reference to a pre-defined virtual corpus named *ratskorpus*. A pen icon indicates that query rewriting has been performed. A login form is provided to facilitate user authentication.

to static and predefined user configurations. Thus, they do not cover all access control requirements in KorAP, particularly to support access control for third-party applications. In addition to LDAP for authentication, KorAP makes use of OAuth 2.0 for authorization (see Section 3.3). OAuth 2.0 (Hardt, 2012) is an authorization framework that allows users to grant specific permissions to third-party applications, for instance to access their data in KorAP and perform a search on their behalf, without requiring them to share their credentials.

By means of authorization, KorAP is able to provide numerous linguistics resources of DeReKo to third parties. It has been integrated into the CLARIN Federated Content Search (FCS) enabling access DeReKo. However, since FCS lacks support for an authorization mechanism, that is a prerequisite to access protected DeReKo resources (see Section 3.1), only a small amount of free resources are accessible. Nonetheless, KorAP permits unauthorized requests to search the metadata of all resources, including protected ones, and can report the number of matches found. If FCS were to support the display of such results, it would significantly enhance the user experience.

In access control management, Keycloak (Thorgersen & Silva, 2021) provides a comprehensive admin console, that allows a wide range of authentication and authorization management, for example, to customize an authentication flow and manage access based on user roles. It supports login with social networks that are not limited to institutions like Shibboleth, authorization using OAuth 2.0, and user federation enabling access to external user data stores such as LDAP. The BlackLab corpus search engine (de Does et al., 2017) has been carrying an ongoing development on authentication using Keycloak. Besides, Google Zanzibar (Pang et al., 2019) is a global authorization system to store and manage access controls across a vast range of applications. It replicates authorization data to rapidly determine user access and permissions on resources over hundreds of applications with various access control policies. While Keycloak and Google Zanzibar allow managing access based on user roles and groups, KorAP also requires access control based on licenses (see Section 3.1).

3 KorAP Approach

The challenge of a corpus license and user rights management system is to find technical solutions for mapping rights and licenses, and restricting data access accordingly. The system must be performant and adaptable to changes in rights and new license forms. To maintain flexibility and independence from

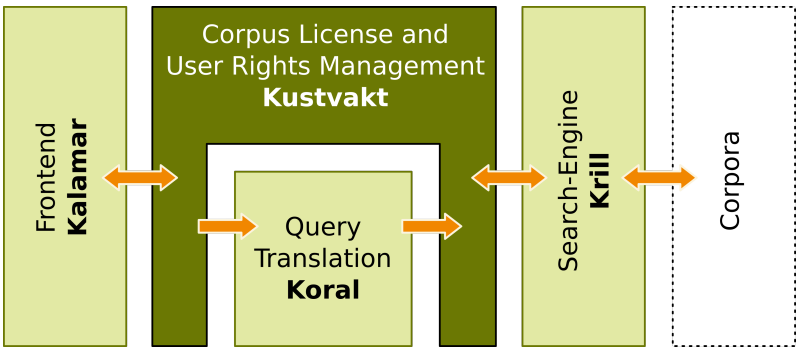

Figure 3: KorAP consists of multiple independent components. The corpus license and user rights management component Kustvakt is amiddleware broker service between API requests and the search engine.

underlying data and user interfaces, we develop a separate server-based component called Kustvakt (Illig et al., 2025)[2], that is responsible for the corpus license and user rights management in KorAP.

Figure 3 depicts the architecture of KorAP that consists of multiple independent small components. Kustvakt plays the role of middleware managing communications between the components and their tasks. It manages KorAP web service APIs that allow the web front-end Kalamar (Diewald et al., 2019) and other clients to communicate with KorAP. By administering user authentication and authorization processes, it verifies whether a request is authenticated or authorized, and applies access policies accordingly to return appropriate responses. Upon receiving (authorized) API requests, it uses Koral to translate user and corpus queries to *KoralQueries* (Bingel & Diewald, 2015). Subsequently, Kustvakt performs query rewriting (Bański et al., 2014) on KoralQueries respecting corpus license and user rights, forwards them to the search engine Krill (Diewald & Margaretha, 2017) and returns the responses to the requesting entity.

3.1 Access policies

IDS has been concluding license agreements for the use of texts for linguistics for around 50 years. Over this period, unavoidable variations have arisen beyond the classes of license conditions, mentioned above. It is typically too expensive and too risky or impossible to reopen any of the legacy agreements. This is the typical situation of institutions that offer large corpora.

To ensure compliance with licensing agreements, we have to deal with the most significant access limitations. Firstly, a great number of DeReKo resources are only available for academic or research purposes, whereas commercial use is explicitly prohibited. Therefore, users must register and agree to our terms of use to access all of DeReKo resources through KorAP. User data is stored and managed through a centralized LDAP for multiple services at IDS Mannheim including KorAP. Secondly, access to DeReKo must be restricted through a query system, such as KorAP, particularly designed to prevent the downloading and reconstruction of original texts from search results. In KorAP, we impose a limit to the size of the match context, and employ a timeout mechanism to restrict search duration thereby enhancing system responsiveness. Thirdly, users must authenticate to access protected resources.

To address all these limitations, we use the six license categories: CC (Creative Commons) variants, ACA-NC (academic, non-commercial), ACA-NC-LC (license contract also required), QAO-NC (query-analysis-only, non-commercial), QAO-NC-LOC:ids (only accessible through IDS network), QAO-NC-LOC:ids-NU:1 (only one user at a time), that are described in detail in Kupietz and Lüngen (2014). Each text of DeReKo is annotated with an `availability` metadata field representing its license category. Figure 1 presents some metadata fields of a Wikipedia text in DeReKo including the availability field with license category CC-BY-SA.

Based on license categories, login and network location, we define three types of corpus access policies

[2]https://github.com/KorAP/Kustvakt

Corpus Access	Regex Patterns of License Categories	Login Required	Access Location
Free	CC.*	no	anywhere
Public	CC.*, ACA.*, QAO-NC	yes	anywhere
All	CC.*, ACA.*, QAO.*	yes	IDS

Table 1: To comply with DeReKo license agreements, three types of access policies are defined in KorAP based on license categories, login and access location.

for KorAP: 1. *Free* access on corpora under CC licenses, that are accessible from anywhere without login, e.g., Wikipedia; 2. *Public* access on free and academic corpora, that requires login; 3. *All* corpora access, that requires login and access through IDS network or Virtual Private Network (VPN) providing a secure connection to the IDS network. By login, we mean not only user authentication but also authorization given to a third-party application (see Section 3.3). We use IP address ranges to determine the access location of requests. Table 1 summarizes the access policies and describes the regular expression patterns of license categories corresponding to each corpus access policy. Kustvakt determines and grants access to a request according to these policies.

Corpus Access	Rewrite Rules
Free	availability = CC.*
Public	availability = CC.* | ACA.* | QAO-NC
All	availability = CC.* | ACA.* | QAO.*

Table 2: Rewrite rules for corpus access are defined using *availability* metadata and regular expression patterns of license categories.

The access policies are enforced through query rewriting described in the following section. Table 2 presents the rewrite rules derived from the availability field and the regular expression patterns of the license categories. The rewrite rules define a virtual corpus (a subset of DeReKo) accessible under each specific access policy. After determining the appropriate access for a request, Kustvakt dynamically modifies the user query according to the rewrite rules. A search request from an non-authenticated user, for example, would be granted free access and executed on a virtual corpus containing all texts whose availability field specifies a CC license category variant. Figure 4 illustrates the search request and the free access granted through the virtual corpus definition at lines 31-40.

The access policies are also applied to pre-defined virtual corpora, as well as those created by users on the fly. The size of the virtual corpora accessible on request may vary according to login and access location constraints. However, metadata of all corpora is freely available regardless of access restrictions on corpus content.

3.2 Query Rewrites

To manage access to a resource in terms of both licenses and user rights while granting the user the greatest possible amount of liberty, an approach based on *query rewriting* was chosen. In this approach, a resource request (see Fig. 5) is reformulated via a central component (Kustvakt) to correspond to the access rights of the requesting entity and can be answered by the database without further knowledge of licenses and user rights (cf. Rizvi et al., 2004).

To achieve this, restrictions in the form of metadata constraints are encoded at the individual text level and can thus be excluded directly during a search or analysis on the corpora. In principle, it is possible to take any metadata into account in the rewrite process, for example the identification of a license (as in Figure 4 via the metadata field `availability`, line 33), but also corpus labels or author names. Additional rights are added to the query as additional constraints; in this sense, the approach is fundamentally

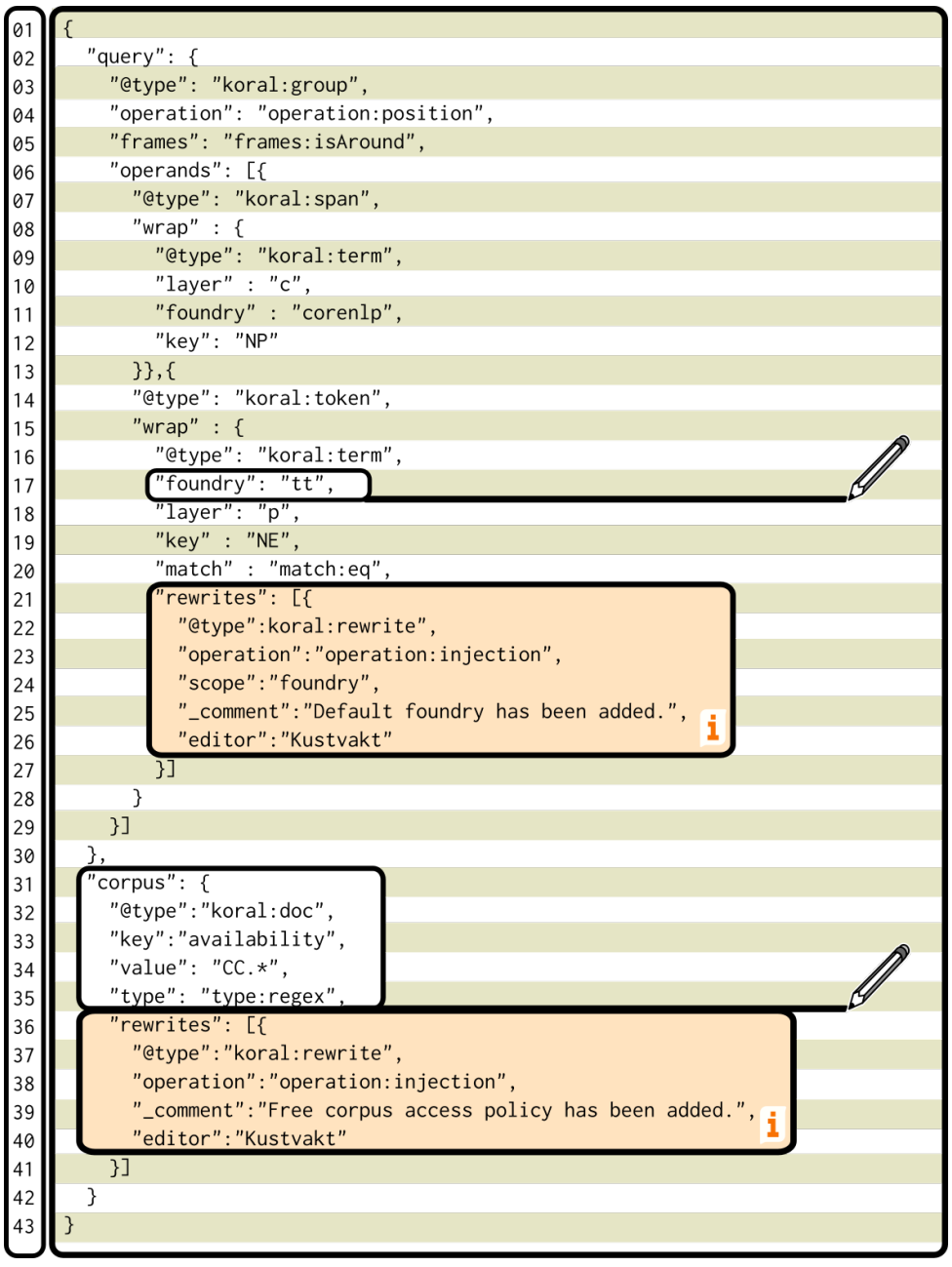

Figure 4: The corpus query ‘Return all nominal phrases `NP` annotated in the `corenlp` foundry and contain a named entity `NE`’ is rewritten by Kustvakt to use a default foundry annotation `tt` for the part-of-speech layer `p` and to restrict access to free corpora licensed under Creative Commons.

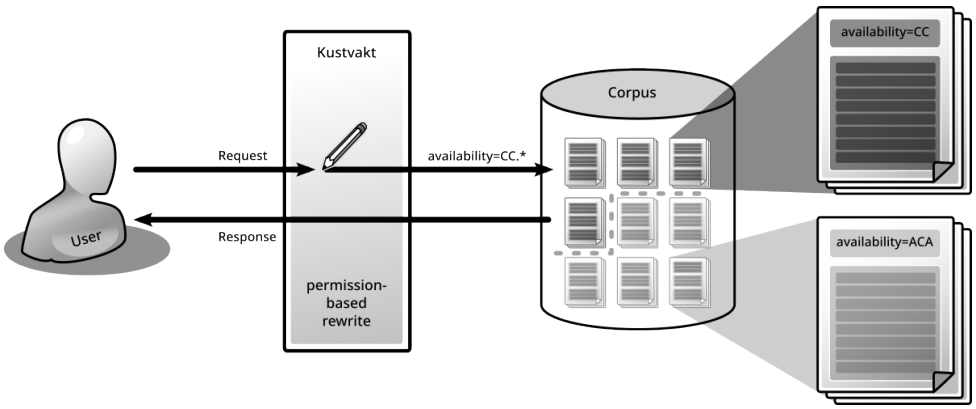

Figure 5: Resource request flow of an authenticated user: The request is rewritten to inject a constraint regarding the metadata field `availability` based on the user's access permissions.

"additive".

The basis for this query rewriting is KoralQuery (Bingel & Diewald, 2015), an implementation of CQLF (Bański et al., 2016) in form of a unified JSON-LD-based (Sporny et al., 2014) representation of an abstract corpus query. It is comparable to SQL for relational data queries (cf. Figure 4) and independent of any corpus query language such as CQP, Annis-QL, Poliqarp, and Cosmas-2-QL supported by KorAP. By adding or changing constraints, a new corpus query can be formulated that satisfies all requirements, and can be passed to a database. Figure 4, lines 31–35, illustrates a constraint that restrict a virtual corpus to all texts with a Creative Commons license.

In addition to restrictions on text access, it is also possible to exclude query options. For example, rules can be formulated to exclude the search in certain annotations or set defaults for queries on annotations. Figure 4, line 17, describes a constraint restricting the search of part-of-speech *NE* to the default source of annotation data (called *foundry*) *tt*, that is the TreeTagger annotation.

Any modification to the query is marked (see Figure 4, lines 21–26 and 36–40), and can be used by clients to inform users or to reconstruct the original query. Figure 2 presents an example where the web UI Kalamar displays a pen icon next to the virtual corpus menu, indicating query rewriting that has been performed on the corpus query level. This may be necessary, as it is the only way to ensure transparency and provide users with feedback on requested resources they actually have access to.

Query rewriting is independent of the corpus and the user size, therefore it scales and performs well with a growing database. It is only dependent on the different restrictions that need to be lifted in the case of permissions granted to the users.

3.3 Authorization

An API enables client applications to communicate with a server-based corpus platform like KorAP allowing technically skilled users to integrate web-services supported by the platform into their own applications. For instance, using RKorapClient (Kupietz et al., 2020) library, users can send search and annotation requests from R to Kustvakt, which is the API provider of KorAP, and then extract and visualize the results in R. In terms of reusability and scalability, API allows Kustvakt to be easily set up for other research environments, extended to meet specific needs, and combined with other front-ends.

As described in the previous section, only corpora with free licenses and metadata in DeReKo are accessible without user authentication. To access licensed corpora via the KorAP API, third party applications must obtain authorization, that is permission granted by users to act on their behalf. Note that location-based access restrictions still apply.

KorAP supports the authorization framework OAuth 2.0 that defines communication protocols with client applications to grant them authorizations in forms of access tokens. Access tokens are bound to

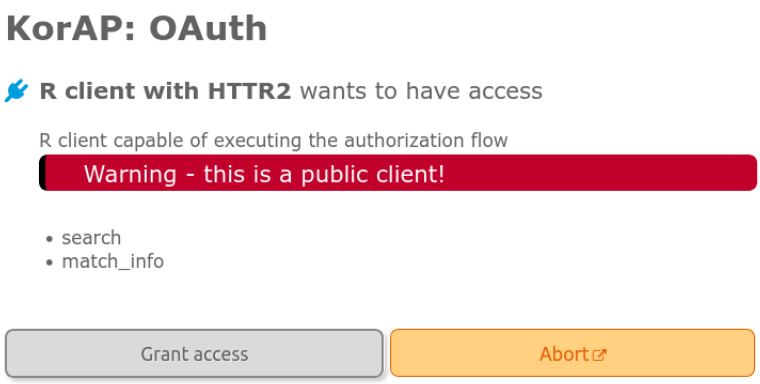

Figure 6: Kalamar displays an authorization request originating from an R client using httr2, that requests the *search* and *match_info* scopes to be able to perform search and annotation requests on behalf of the authenticated user. The user is given the option to either grant or decline access.

particular authorization scopes specifying to what extent these applications may act on user behalf or access user data. Kustvakt acts as an authorization server, that manages communications to clients via API and issues and manages access tokens (Kupietz et al., 2022). In addition to that, Kalamar acts as a front-end to the API, that provides web user interface for user authentication depicted in Figure 2.

To obtain access tokens, we support authorization code grant flow suitable for server-based client applications. This flow involves sending and processing HTTP requests multiple times to enhance security ensuring that access tokens are not leaked to intermediate user-agents such as browsers, but directly sent to clients. When a user would like to use a client to perform a search and acquire match annotations in KorAP, the client sends an authorization request including the necessary scopes (*search* and *match_info* in this example) to the KorAP authorization server. If the user is not authenticated to KorAP yet, it would ask him/her to login, and then to grant access to the client to perform search and retrieve annotation requests on behalf of him/her (see Figure 6). When granted, the KorAP authorization server would redirect the user to the client redirect URI including an authorization code. The client can subsequently exchange the authorization code with an access token by sending a token request.

For non-server-based clients such as desktop applications, that are incapable of handling HTTP requests, we provide a feature to obtain access tokens from the web UI Kalamar as shown in Figure 7. Alternatively, local web-servers or libraries can be utilized to facilitate the authorization code flow, for example RKorAPClient uses httr2 (Wickham, 2023) that also supports OAuth 2.0. To enable this workflow, the KorAP authorization server permits localhost as a client redirect URI.

Client Type	**Can store secrets**	**Access token Validity**	**Refresh Token Validity**
Confidential	Yes	Short-lived	Long-lived
Public	No	Long-lived	Not Available

Table 3: To reduce the risk of token compromise, the time validity of access tokens and the issuance of refresh tokens are adjusted depending on the ability of clients to securely store secrets.

To protect users from potential authorization abuse by malicious software, we implement the following measures. Firstly, client registration is required to use the KorAP authorization APIs. Figure 7 displays a screenshot of the web UI Kalamar illustrating a registered OAuth2 client in KorAP. Kalamar provides details about the client including its type, identifier, and active access tokens.

Secondly, the time validity of access tokens is deliberately limited depending on the ability of clients to keep credentials. It is crucial to limit the time validity of access tokens to reduce the period of time

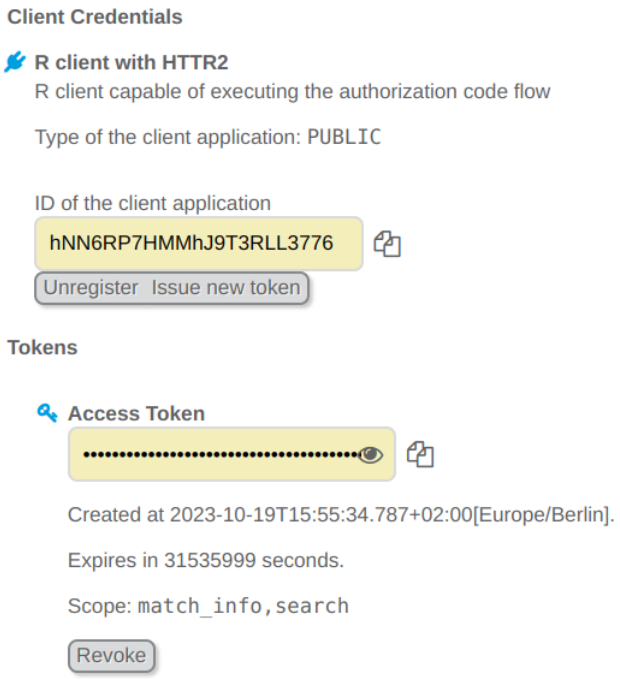

Figure 7: Kalamar provides a web user interface for managing OAuth clients and tokens. It provides client details, supports client unregistration, and shows access tokens info including expiry and authorization scopes. It also allows users to issue access tokens for public clients, particularly for non server-based client, and to revoke existing tokens.

in which a stolen access token can be exploited by an attacker. Thus, issuing short-lived access tokens is a reasonable and necessary strategy to minimize the potential impact of token theft. Since new access tokens can be issued using a special token, called refresh token, the security implications are even more significant when a refresh token is compromised. The client's ability to securely store secrets also plays a critical role in reducing the risk of token theft. According to the OAuth 2.0 specification, clients are categorized into 2 types: *confidential clients*, that can store secrets and authenticate securely, and *public clients*, that cannot. Confidential clients are issued short-lived access tokens along with a long-lived refresh token, that allows them to obtain new access tokens without requiring re-authorization. On the other hand, since public clients are more vulnerable to token compromise, they are not provided with refresh tokens. They are issued long-lived access tokens because they cannot obtain new access tokens without a refresh token. This distinction is summarized in Table 3.

Thirdly, KorAP supports token revocation (Lodderstedt & Scurtescu, 2013) to invalidate tokens before they expire. It is especially beneficial when users suspect unauthorized or malicious use of their access tokens. This functionality is available via both the API and the web UI Kalamar depicted in Figure 7.

4 Extensibility

KorAP has already covered all the access rights requirements for DeReKo. Some extensions can be profitable as follows.

Metadata is generally freely available regardless of access restrictions on corpus content. However, when certain metadata fields are restricted due to licensing terms, access to those fields can also be limited, similar to the restrictions on corpus content. Moreover, in exceptional cases, metadata such as titles (e.g., newspaper headlines) may reveal information about natural persons, which may also justify restrictions on access.

In addition to policy enforcement, the protocol-based approach enables the integration of other query rewriting methods, such as query expansion (cf. Baeza-Yates & Ribeiro-Neto, 2010, ch. 5) independent of the user and corpus base. This approach allows applying cascading rewrites to a query with policy enforcement at the end to prevent unintended expansion of permissions.

Following query rewriting, *response rewriting* can also be performed. In this case, the result set from the search engine is filtered according to certain criteria before it is returned to the account. However, since response rewriting usually requires more data to be requested from the resource than can finally be processed, this variant of rewriting is only suitable for small result sets for performance reasons. For example, it is well-suited for the individual shortening of text snippets that are displayed to the account (when certain text licenses allow longer contexts than others). Response rewriting is currently being implemented to enrich data results with external information (specifically mappings for universal dependency annotations). There are currently no efforts and no need to use this mechanism for the access-based filtering of results, which is why we do not discuss it further in this article.

Using Shibboleth, CLARIN enables distributed access to protected resources through SSO across organizational boundaries. However, this alone is not sufficient to access DeReKo's protected resources, as users must also explicitly agree to our terms of use, as described in Section 3.1. Shibboleth can be implemented in Kustvakt as an alternative to LDAP for authentication, particularly when serving academic corpora that do not require the same user agreement as DeReKo. Since KorAP is independent of specific resources, it can operate as a standalone instance serving a wide range of corpora beyond DeReKo.

5 Conclusion

Directly integrating policy enforcement at the protocol level through query rewriting and abstract authorization mechanisms allows for a great deal of transparency and flexibility for efficient and detailed access control to corpus resources with complex licenses. Our approach facilitates maximum access and usage of corpora while ensuring compliance with complex licenses. The currently applied rule set in our implementation is based on the needs of the different licenses of DeReKo, so the full flexibility is not yet exhausted. Our query rewriting approach is developed as programming APIs allowing easy integration of new rules for other applications. Simple new rewrite rules can be introduced by minor changes to the configuration, while more complex rules can be added as filters (currently between 30 and 200 lines of code). The largest application using our approach is currently a corpus query system that serves a corpus of 87 million texts for an average of 6000 queries per day. Kustvakt is open source and in conjunction with KoralQuery universally applicable for resource control in corpus analysis applications.

References

Baeza-Yates, R., & Ribeiro-Neto, B. (2010). *Modern Information Retrieval: The Concepts and Technologies behind Search* (2nd ed.). Addison-Wesley.

Bański, P., Diewald, N., Hanl, M., Kupietz, M., & Witt, A. (2014). Access control by query rewriting: The case of KorAP. *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14)*, 3817–3822. http://www.lrec-conf.org/proceedings/lrec2014/pdf/743_Paper.pdf

Bański, P., Frick, E., & Witt, A. (2016). Corpus Query Lingua Franca (CQLF). *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16)*, 2804–2809.

Bingel, J., & Diewald, N. (2015). KoralQuery - a General Corpus Query Protocol. *Proceedings of the Workshop on Innovative Corpus Query and Visualization Tools at NODALIDA 2015*.

Cantor, S., & Scavo, T. (2005). Shibboleth architecture. *Protocols and Profiles*, *10*(16), 29.

de Does, J., Niestadt, J., & Depuydt, K. (2017). Creating Research Environments with BlackLab. *CLARIN in the Low Countries*, 245–257.

Diewald, N., Barbu Mititelu, V., & Kupietz, M. (2019). The KorAP user interface. Accessing CoRoLa via KorAP. *Revue Roumaine de Linguistique. On design, creation and use of of the Reference Corpus of Contemporary Romanian and its analysis tools. CoRoLa, KorAP, DRuKoLA and EuReCo*,

64(3), 265–277. http://www.lingv.ro/images/RRL%5C%203%5C%202019%5C%2006-%5C%20Diewald.pdf

Diewald, N., Hanl, M., Margaretha, E., Bingel, J., Kupietz, M., Bański, P., & Witt, A. (2016). KorAP Architecture - Diving in the Deep Sea of Corpus Data. *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16)*, 3586–3591.

Diewald, N., & Margaretha, E. (2017). Krill: KorAP search and analysis engine (M. Kupietz & A. Geyken, Eds.). *Journal for language technology and computational linguistics (JLCL)*, *31*(1), 73–90. http://www.jlcl.org/2016_Heft1/Heft1-2016.pdf

Hardt, D. (2012, October). The OAuth 2.0 Authorization Framework. https://doi.org/10.17487/RFC6749

Howes, T., & Smith, M. C. (2006, June). Lightweight Directory Access Protocol (LDAP): Uniform Resource Locator. https://doi.org/10.17487/RFC4516

Iannella, R., & Villata, S. (2018). ODRL Information Model 2.2. *W3C Recommendation*, *15*.

Illig, E. M., Diewald, N., Kupietz, M., Hanl, M., & Bodmer, F. (2025, March). *Kustvakt* (Version 0.76). Zenodo. https://doi.org/10.5281/zenodo.15044768

Sporny, M., Longley, D., Kellogg, G., Lanthaler, M., & Lindström, N. (2014). JSON-LD 1.0. A JSON-based Serialization for Linked Data. http://www.w3.org/TR/json-ld/

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., & Suchomel, V. (2014). The Sketch Engine: ten years on [Communicated by Yukio Tono.]. *Lexicography*, *1*(1), 7–36. https://doi.org/10.1007/s40607-014-0009-9

Kupietz, M., Diewald, N., & Margaretha, E. (2020). RKorAPClient: An R Package for Accessing the German Reference Corpus DeReKo via KorAP. *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC '20)*, *12*, 7015–7021.

Kupietz, M., Diewald, N., & Margaretha, E. (2022). Building paths to corpus data. A multi-level least effort and maximum return approach. In D. Fišer & A. Witt (Eds.), *CLARIN. The Infrastructure for language resources* (pp. 163–189). de Gruyter. https://doi.org/10.1515/9783110767377-007

Kupietz, M., & Lüngen, H. (2014). Recent developments in DeReKo. *Proceedings of the 9th conference on international language resources and evaluation (LREC'14)*, 2385.

Lodderstedt, T., & Scurtescu, M. (2013, August). *OAuth 2.0 Token Revocation* (RFC No. 7009). RFC Editor. https://tools.ietf.org/html/rfc7009

Olson, M. (1965). *The Logic of Collective Action: Public Goods and the Theory of Groups*. Harvard University Press.

Pang, R., Caceres, R., Burrows, M., Chen, Z., Dave, P., Germer, N., Golynski, A., Graney, K., Kang, N., Kissner, L., Korn, J. L., Parmar, A., Richards, C. D., & Wang, M. (2019). Zanzibar: Google's Consistent, Global Authorization System. *2019 USENIX Annual Technical Conference*.

Reynaert, M., van de Camp, M., & van Zaanen, M. (2014). OpenSoNaR: User-Driven Development of the SoNaR Corpus Interfaces. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, 124–128.

Rizvi, S., Mendelzon, A., Sudarshan, S., & Roy, P. (2004). Extending Query Rewriting Techniques for Fine-Grained Access Control. *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, 551–562. https://doi.org/10.1145/1007568.1007631

Thorgersen, S., & Silva, P. I. (2021). *Keycloak-identity and access management for modern applications: harness the power of Keycloak, OpenID Connect, and OAuth 2.0 protocols to secure applications*. Packt Publishing Ltd.

Wickham, H. (2023). *httr2: Perform HTTP Requests and Process the Responses* [https://httr2.r-lib.org, https://github.com/r-lib/httr2].