The Kosh Suite:
A Framework for Searching and Retrieving Lexical Data Using APIs

Francisco Mondaca, Philip Schildkamp, Felix Rau, Luke Günther
University of Cologne

Overview

- What is Kosh?
- The rationale behind Kosh
- How Kosh works
  - Preparing your data
  - Deployment options
  - Building search queries
- Complementary tools
- Use cases

Kosh

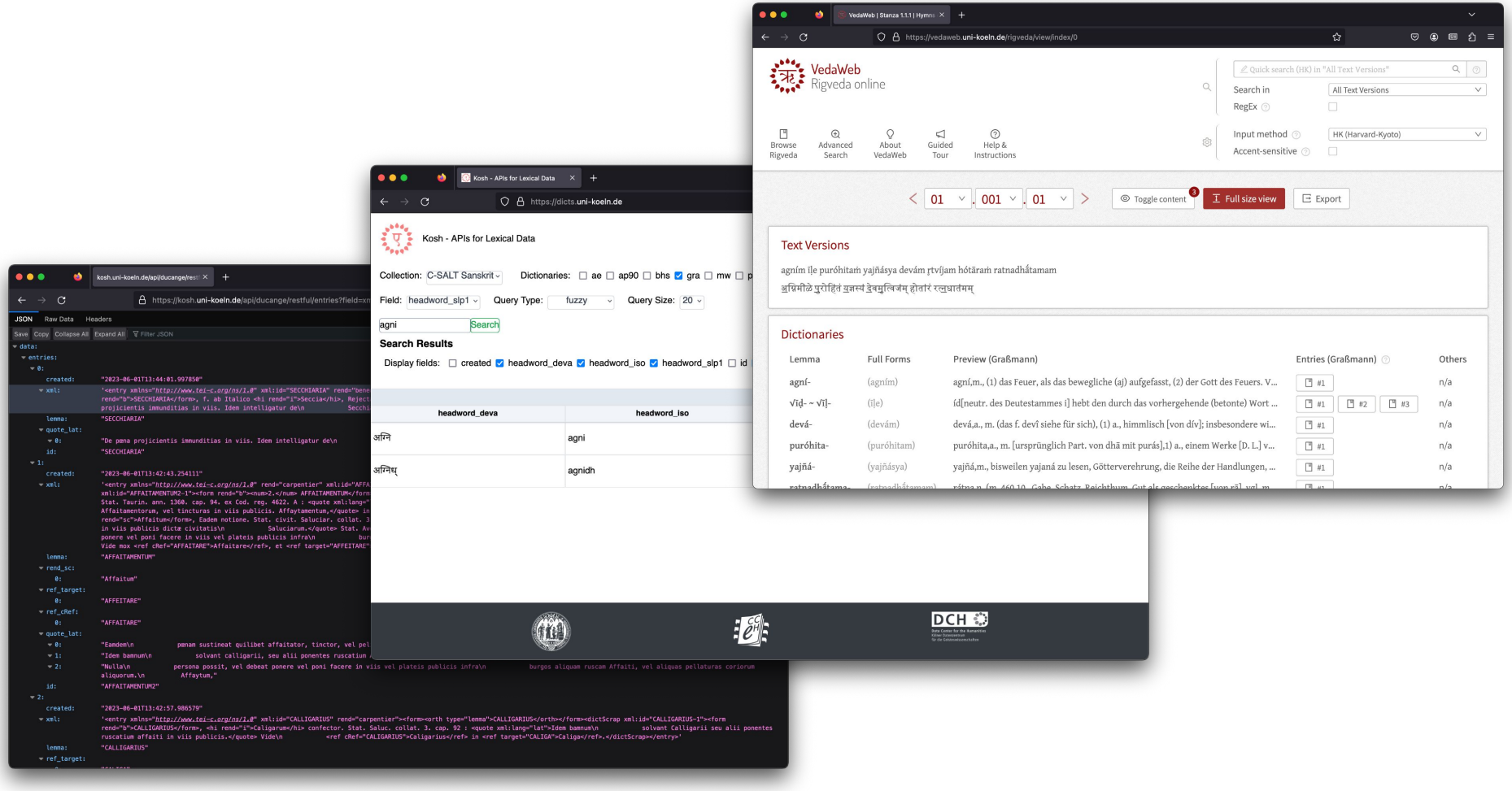
… is a framework to serve diverse lexicographic data

- flexibly
- sustainably
- with minimal configuration

Kosh provides APIs for XML-encoded dictionaries

- format agnostic
- serves REST and GraphQL APIs

… but also comes with a default user interface



The Kosh Suite – eLex 27-29 June 2023, Brno, Czech Republic

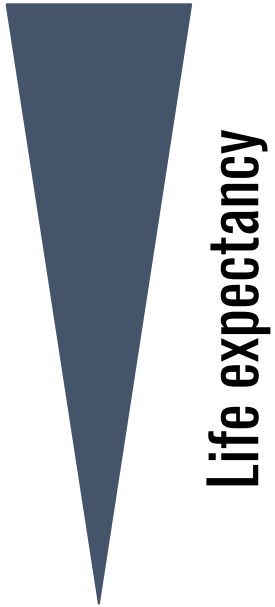
Motivation

- We are supporting a variety of projects at the University of Cologne and beyond
- We might have limited influence on data models and serialization
- We are responsible for sustainably providing access to lexicographic data, even after the project has ended
- Project require different types of data integration and data presentation

➔ Focus on generic and long lasting parts of the infrastructure

Infrastructure Life Expectancy

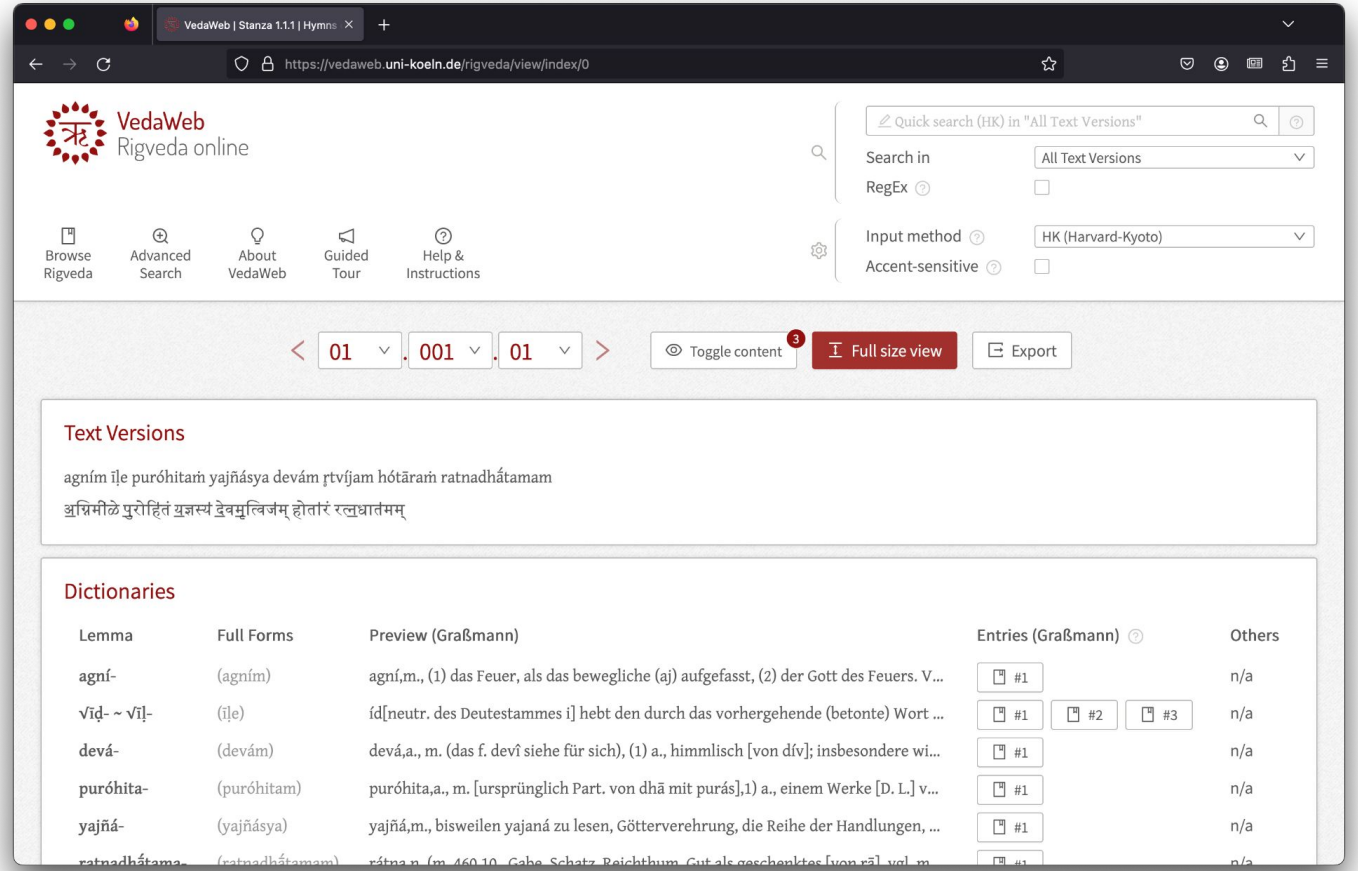
- Data models
- Data serialization
- API definitions
- Backend software
- Frontend software



Focus: Providing APIs

- Actively serve data in an easily usable way (not just preserve it in a repository)
- Accommodate different data models and data serializations
- The backend software can be refactored or rewritten
- Data can be integrated in project specific presentations and applications
- Data can be used and viewed even if project specific solutions can no longer be maintained

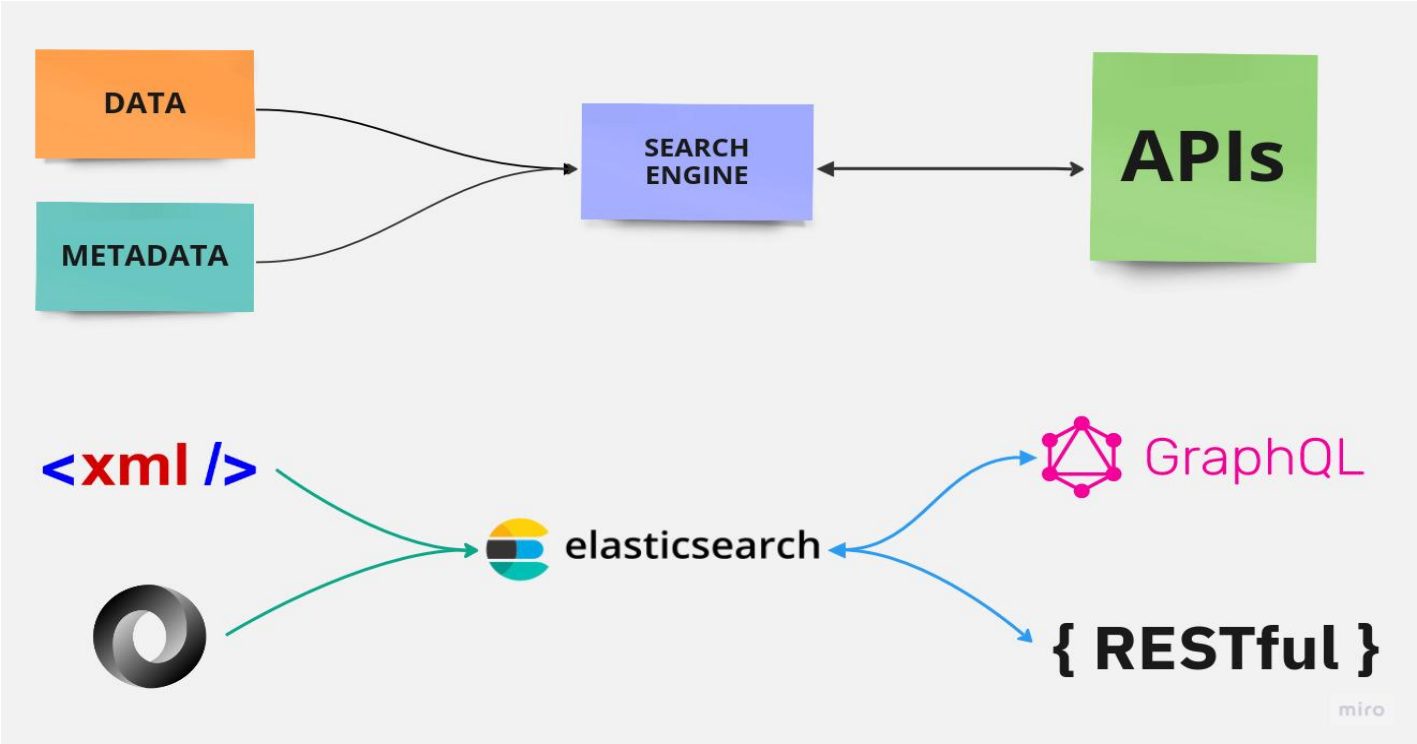
Most user-friendly, sustainable investment of work for our situation



https://vedaweb.uni-koeln.de/

Kosh Suite: How it works

Kosh Suite: How it works



Kosh Backend: Building blocks

- Kosh can be used as is with any data already existing in XML format
- All you need are two simple configuration files

First, you specify how your data should be **indexed**.

➔ Mapping XML nodes to search fields

Second, you add any **metadata** fields you want to be available.

➔ Represented languages, authors/contributors, timeframe, licenses

Mapping XML Data

```xml
<entry id="13">
  <form>
    <orth>abadetasun</orth>
  </form>
  <sense n="1">
    <gramGrp>
      <pos>
        <q>iz.</q>
      </pos>
    </gramGrp>
    <def>monasterioko buruaren kargua eta egitekoa</def>
  </sense>
  <sense n="2">
    <gramGrp>
      <pos>
        <q>iz.</q>
      </pos>
    </gramGrp>
    <def>apaizgoa</def>
    <usg type="geo">
      <q>Bizk.</q>
    </usg>
  </sense>
</entry>
```

```json
{
  "mappings": {
    "_meta": {
      "_xpaths": {
        "id": "./@id",
        "root": "//entry",
        "fields": {
          "lemma": "./form/orth",
          "[sense_def]": "./sense/def",
          "[sense_pos]": "./sense/gramGrp/pos/q",
          "[dicteg]": "./sense/dicteg/q"
        }
      }
    },
    "properties": {
      "lemma": {
        "type": "keyword"
      },
      "sense_def": {
        "type": "text"
      },
      "sense_pos": {
        "type": "text"
      },
      "dicteg": {
        "type": "text"
      }
    }
  }
}
```

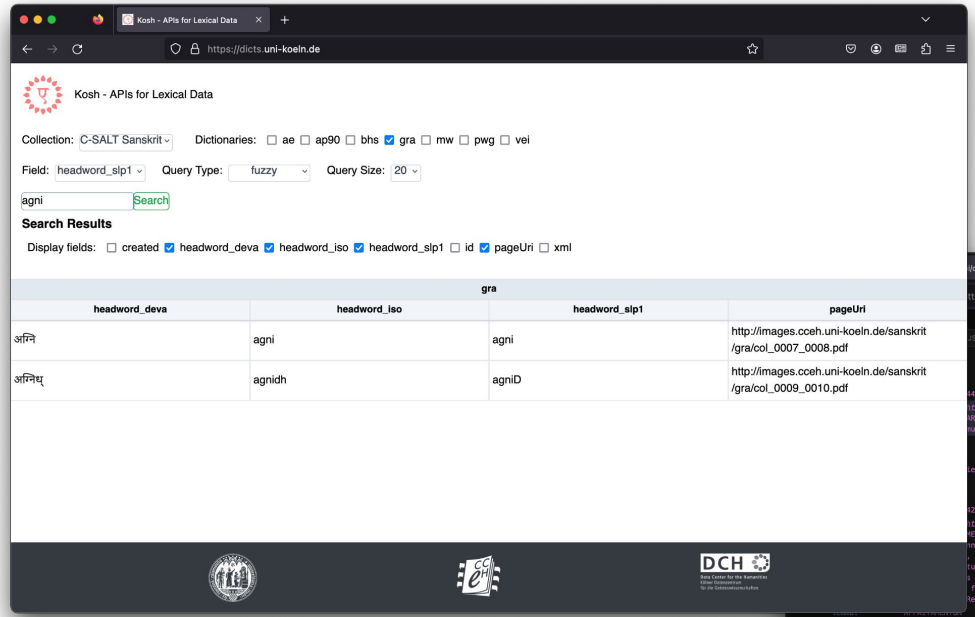
How to serve data with Kosh

Currently, there are three ways you can use Kosh:

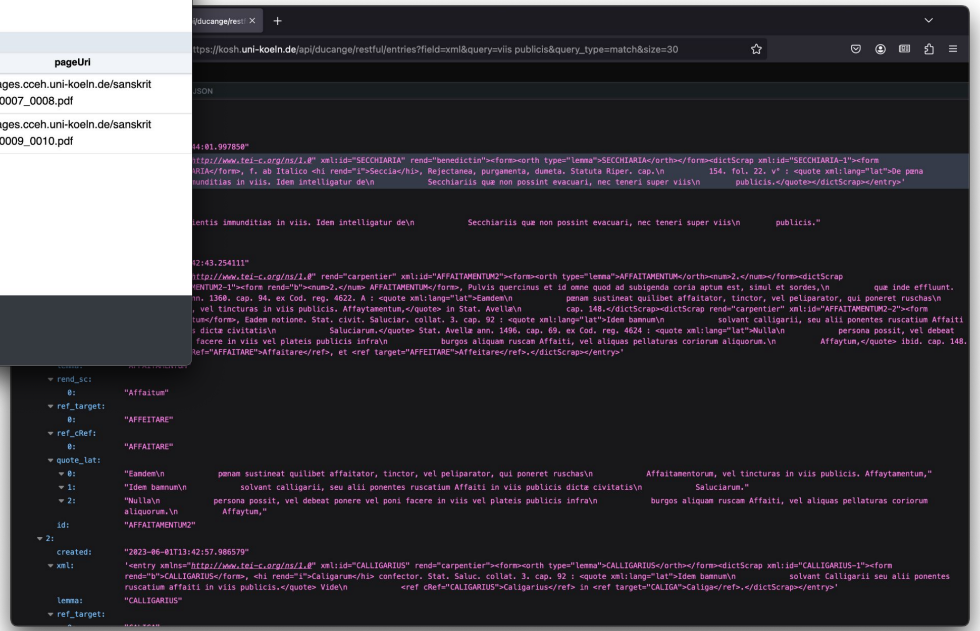
- Our demo instances
- Docker container
- Native support on Unix-like systems → Just grab the code from GitHub!

Real-time updates: Once Kosh has been started, it will watch your data directory for changes and will continuously index new items.

Kosh sample data: Explore Kosh by looking at both external historical but also current project data from our institutes in Cologne.



https://kosh.uni-koeln.de/



https://dicts.uni-koeln.de

Querying Data

- Search strings are matched against the content of a specific property for each lexical item in your database
- There are various search strategies, e.g. fuzzy matching or wildcard search

You can inspect and test the two API endpoints through graphical interfaces!

A typical query for the **REST API** consists of the **query string**, the **field to be queried** and the **search strategy** and, optionally, the **query size**:

```
https://<endpoint>/<dictionary_id>/restful/entries?query=search%20query&field=xml&query_type=match&size=30
```

Complementary tools: Client and Sync

The **Kosh Client** enables you to:

- Compare dictionaries with each other in a familiar table layout
- Search multiple different collections of dictionaries
- Create granular queries and customize the way results are displayed

➔ Can be used locally but the Kosh backend has to be reachable first.

The **Kosh Sync** service can be run additionally if you want continuous integration and deployment of your data from a Git repository.

Summary: Use cases for the Kosh Suite

Kosh can help you:

- Get an overview of your current work
- Gage how far you have come in your data selection and curation process
- Check for errors in your data
- Provide easy access for fellow researchers
- Present your research data to the public, e.g. on a project website

Getting help

For further information, please consult the documentation at:

https://kosh.uni-koeln.de

Or send an email to:

info-kosh@uni-koeln.de

We’re happy to help!
Thank you for your attention!