

# Agile research data management with LinkAhead

IndiScale GmbH



## History

- LinkAhead started at MPI-DS (Göttingen) around 2011 (name: CaosDB)
- Running stable since ca. 2016, released as open-source (AGPLv3)[1] in 2018
- Increasing adoption since 2020
- Commercial support by IndiScale GmbH
  - distribution branded as LinkAhead



## History

- LinkAhead started at MPI-DS (Göttingen) around 2011 (name: CaosDB)
- Running stable since ca. 2016, released as open-source (AGPLv3)[1] in 2018
- Increasing adoption since 2020
- Commercial support by IndiScale GmbH
  - distribution branded as LinkAhead



[1] <https://gitlab.com/caosdb>

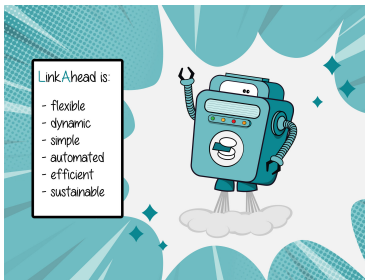
## History

- LinkAhead started at MPI-DS (Göttingen) around 2011 (name: CaosDB)
- Running stable since ca. 2016, released as open-source (AGPLv3)[1] in 2018
- Increasing adoption since 2020
- Commercial support by IndiScale GmbH
  - distribution branded as LinkAhead



[1] <https://gitlab.com/caosdb>

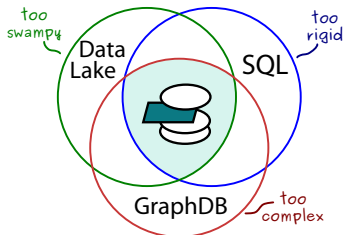
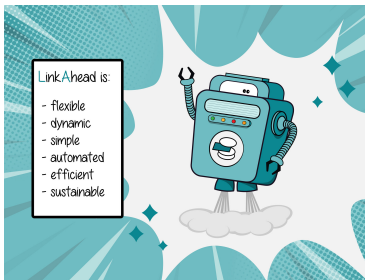
# Challenges: Findability, Linkage, Dynamic Environments



 LinkAhead has:

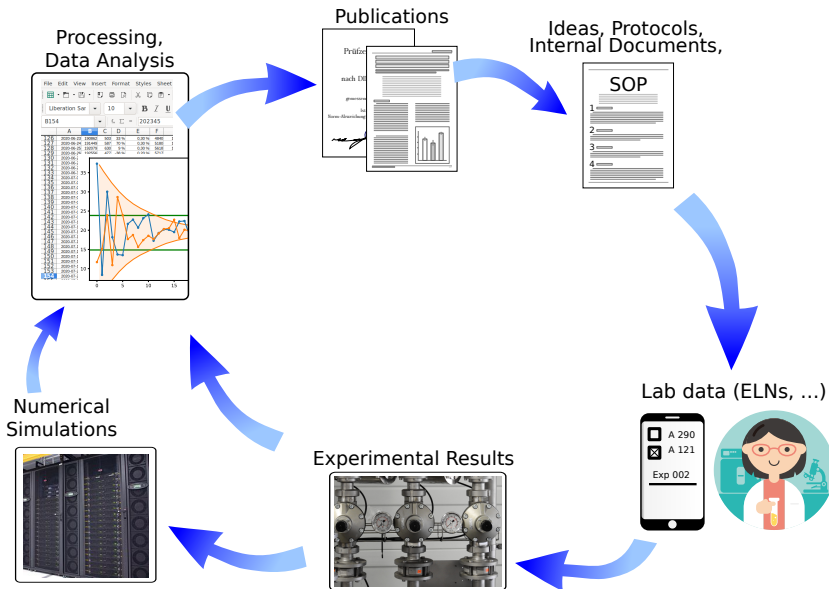
- Linked data files and meta data.
- Data embedded into original context.
- Easy, yet powerful semantic search.
- Semantic core: Fast and easy tuning of the data model.

# Challenges: Findability, Linkage, Dynamic Environments

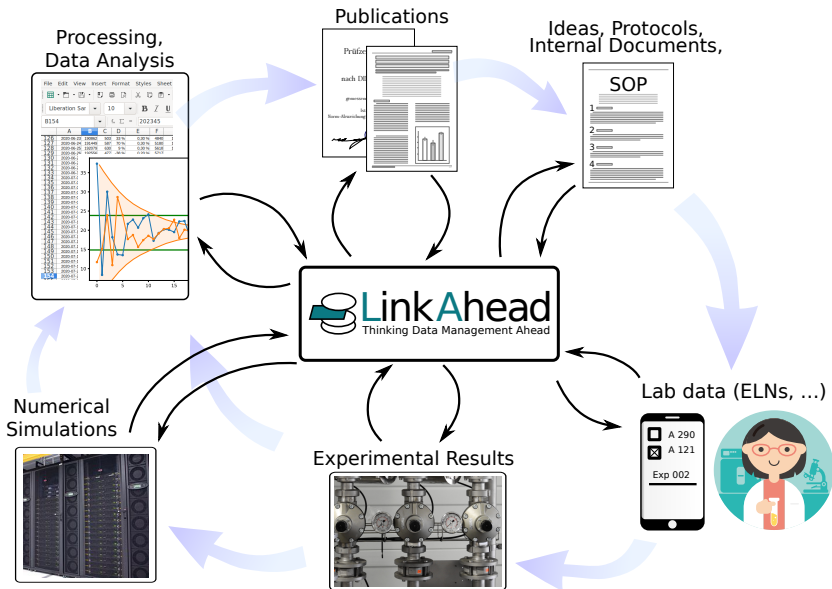


**LinkAhead**  
Thinking Data Management Ahead

# Data Life Cycle

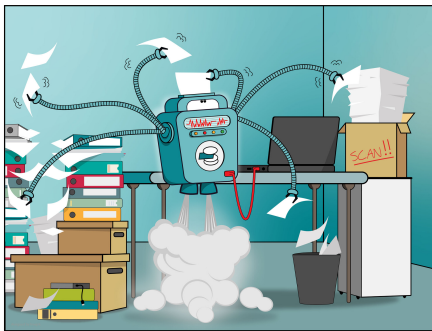




# Data Life Cycle



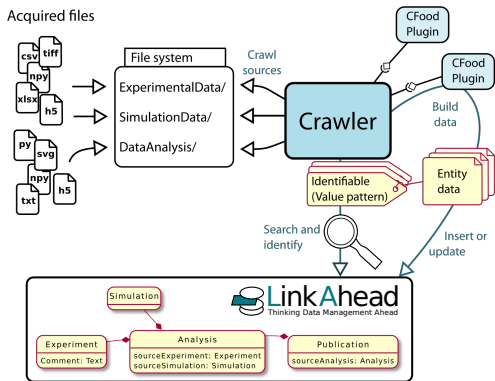
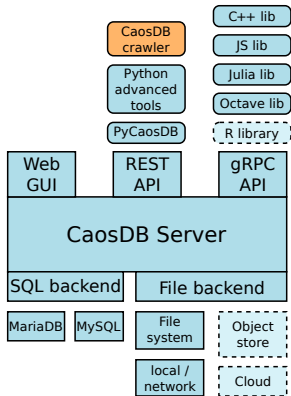


# Usability: Automation



-  LinkAhead can be automated to free users from the repetitive, error-prone task of inserting data manually.
- Common searches can be stored as templates in  LinkAhead.
- We at IndiScale are happy to offer our experience to reduce your work!

# Architecture & Crawler LinkAhead

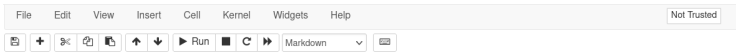


## Live-Demo

<https://demo.indiscale.com>

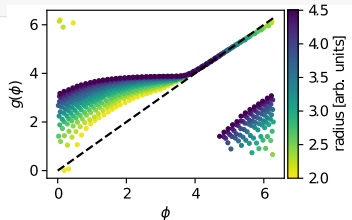
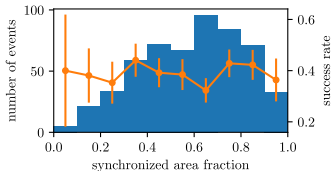


Jupyter Introduction to the CaosDB-Python Client (autosaved)



## Data Analysis with Jupyter

```
In [ ]: import caosdb as db
data = db.execute_query("SELECT quality_factor FROM RECORD Analysis with quality_factor")
table = to_table(data)
plt.plot(table.x, table.y)
```





- Online demo:
  - <https://demo.indiscale.com>
- Source code and development: <https://gitlab.com/caosdb>
- Documentation: <https://docs.indiscale.com>
- Scientific article, published in ***Data***:  
<https://doi.org/10.3390/data4020083> 



- Fully open-source software (AGPLv3)



- Fully open-source software (AGPLv3)
- Powerful search options



- Fully open-source software (AGPLv3)
- Powerful search options
- LinkAhead can be easily extended or modified





- Fully open-source software (AGPLv3)
- Powerful search options
- LinkAhead can be easily extended or modified
- Integrates into existing workflows



- Fully open-source software (AGPLv3)
- Powerful search options
- LinkAhead can be easily extended or modified
- Integrates into existing workflows
- Stores semantic links together with the data

Thank you for your interest!