

SLICES-DE GINO Platform

A platform for easy Internet measurement data analysis for the **Global IN**ternet **Ob**servatory

Project GINO – Global Internet Observatory

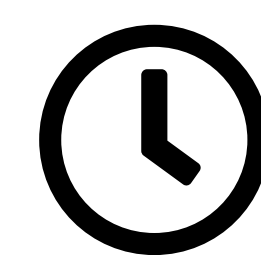
- Our chair performs regular Internet measurements in order to better understand the Internet
- We conduct Internet-wide DNS, TLS, QUIC and port scans for common protocols
- We publish an IPv6 hitlist service
- Downloads of Internet-related data like
 - Zone files of many gTLDs and some ccTLDs
 - Top and block lists
 - CT logs
 - BGP data
 - ...
- Huge amount of data available for analysis but
 - as files on our storage servers
 - in lots of different file formats and complex folder structures

More information on project GINO can be found under <https://net.in.tum.de/projects/gino/>



Challenges of the Current Setup and Proposed Solutions

Challenges of the current Setup:



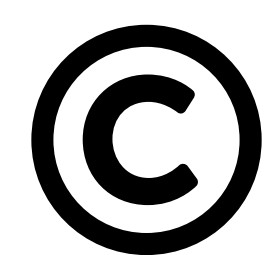
Time

Long setup times until one can start with research



Privacy

Data contains sensitive, private information



Restrictions

License restrictions may restrict sharing

Fine-grained access control of the data is essential for data sharing

Proposed Solutions:

1. ASQ-GINO

Answering pre-defined queries

2. General purpose data access

Authorized data access on general-purpose infrastructure



Goals of the GINO platform

High-level goals:

- Allow external researchers and students to quickly start working with the available data they are allowed to access
- Create a platform for easy and quick analysis of the available data

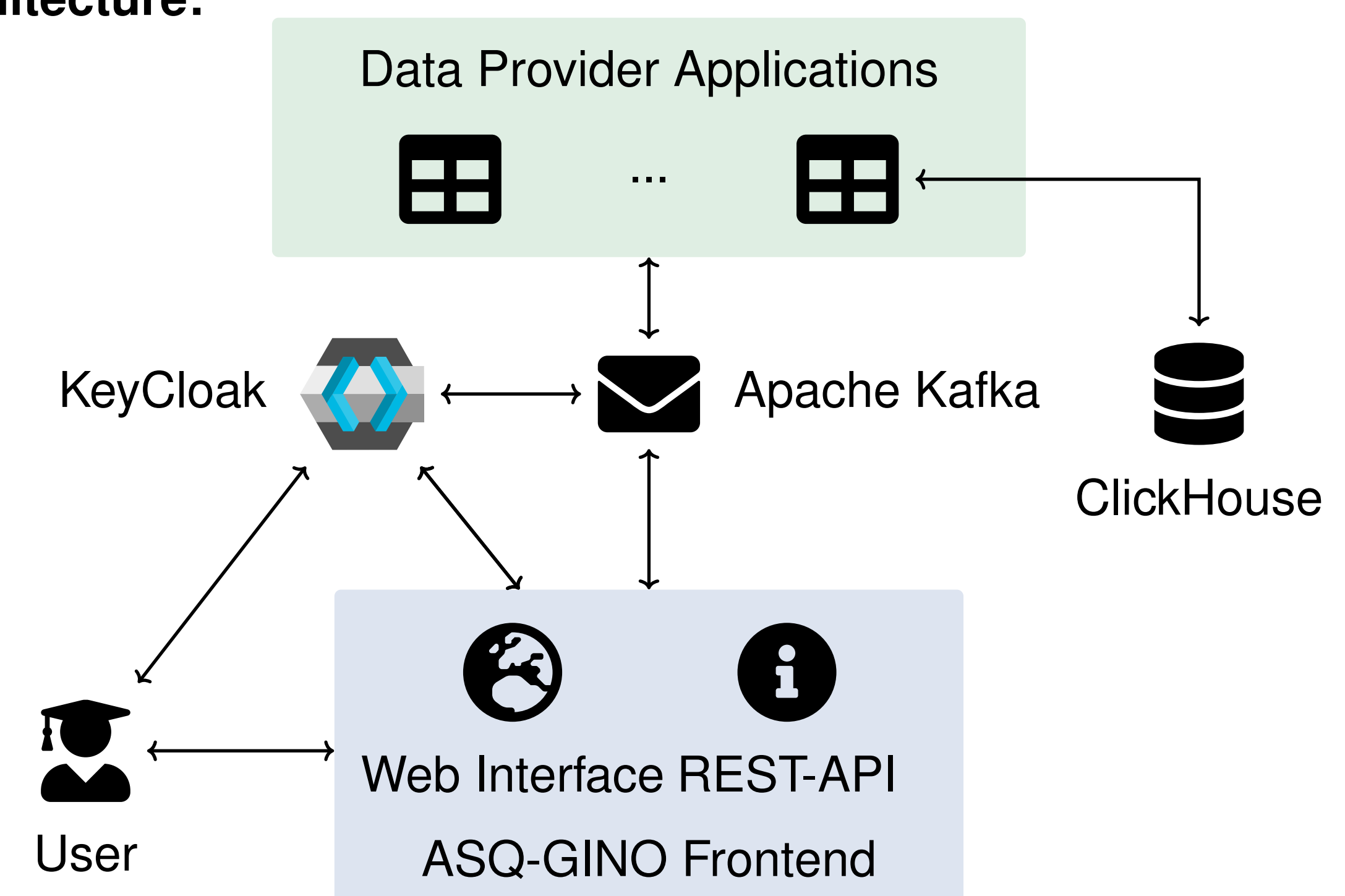
Base goals:

- Have commonly used data sources already loaded in a database ready for analysis
- Allow quick importation of non-imported datasets on demand
- Enable fine-grained access control of
 - the available data
 - the computation resources
- User authentication for researchers using ORCID
- Offer separate computation resources for analysis
- Offer services that answer pre-defined queries and analyses upon user request

ASQ-GINO: Answering Pre-Defined Queries

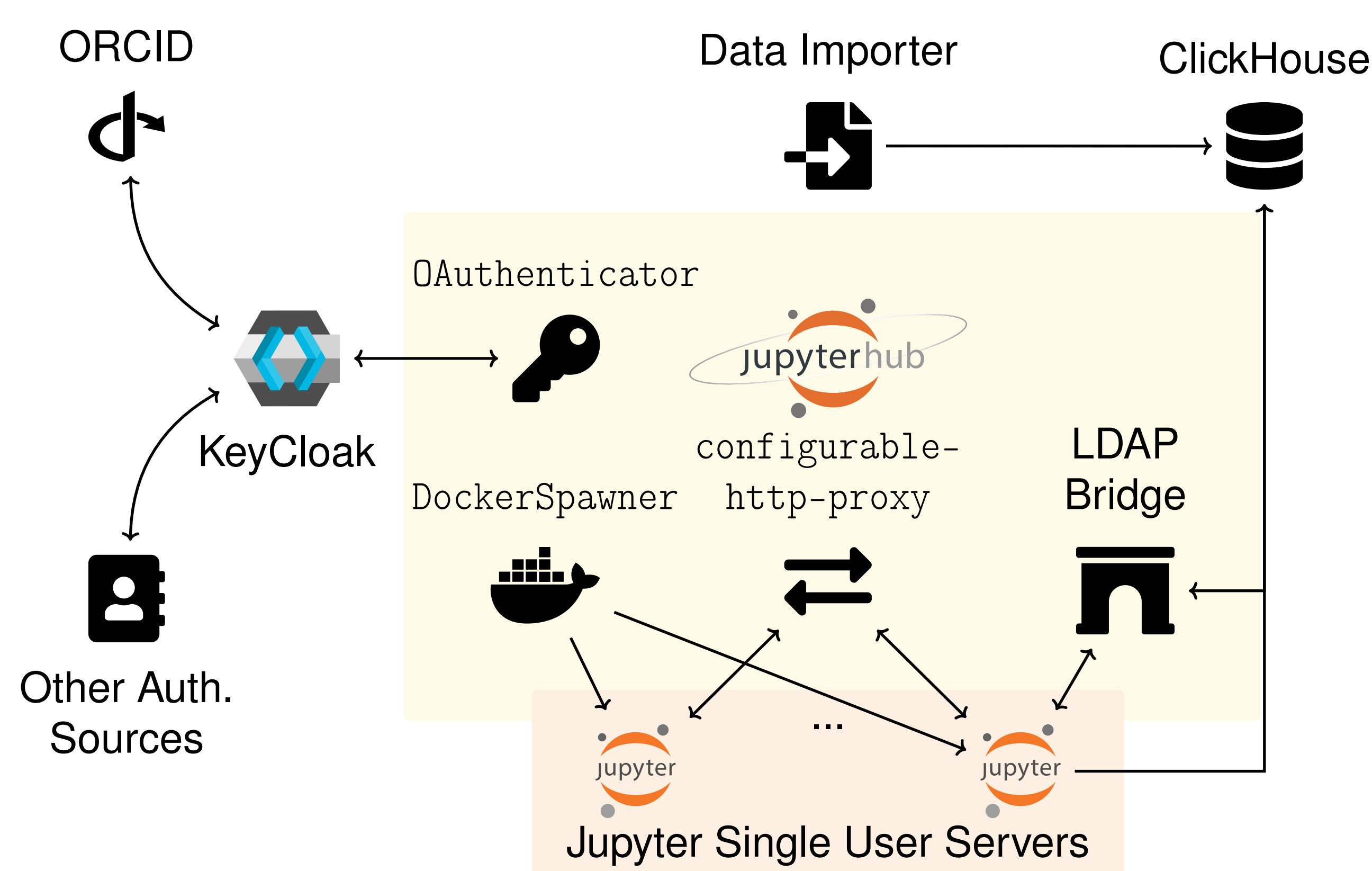
Goal: Users should be able to get answers to pre-defined queries

Architecture:



ASQ-GINO was implemented by Tobias Wothge in his Master's Thesis.

Overview over System Components



Next Steps

- Evaluate methods for multi node deployments for scalability
 - **Kubernetes-based:** KubeSpawner to spawn containers in a Kubernetes cluster
 - **pos-based:** Custom spawner to boot a live image with preconfigured JupyterLab on a testbed node
- Design and implement the data importer^a
- Implement data provider applications for all major data sets^b
- Improve LDAP-bridge and write helper tooling for easy usage
- Make ASQ-GINO production ready: e.g. rate limiting
- Implement monitoring system
- Deploy to production servers

^aThis is currently being developed as part of a Bachelor's Thesis.

^bThis is currently being developed by students as part of their Theses.