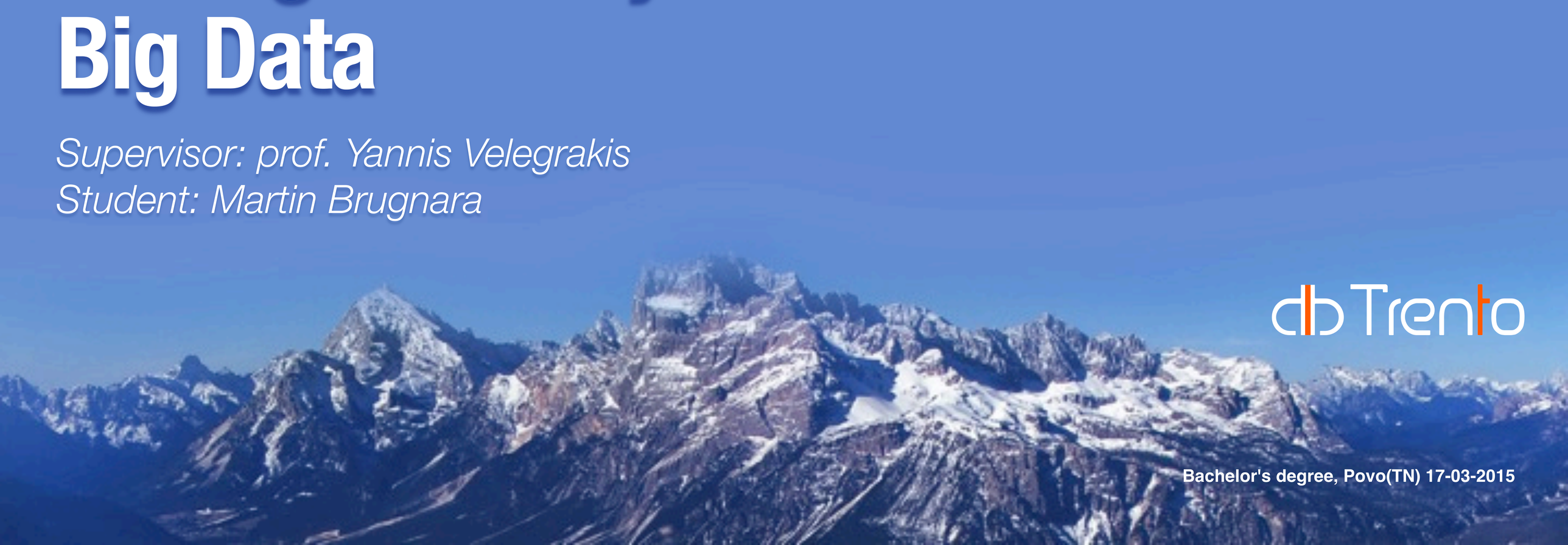
DIP: A Dataset and Process Management System for Big Data

Supervisor: prof. Yannis Velegrakis
Student: Martin Brugnara

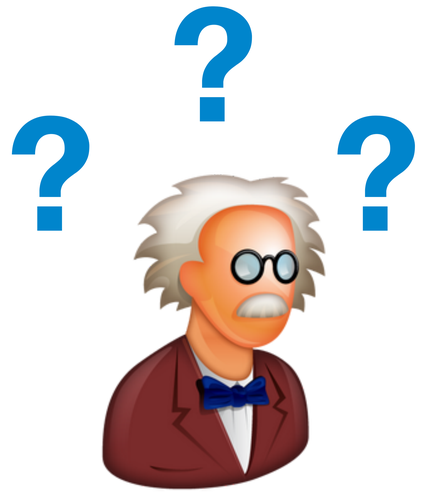


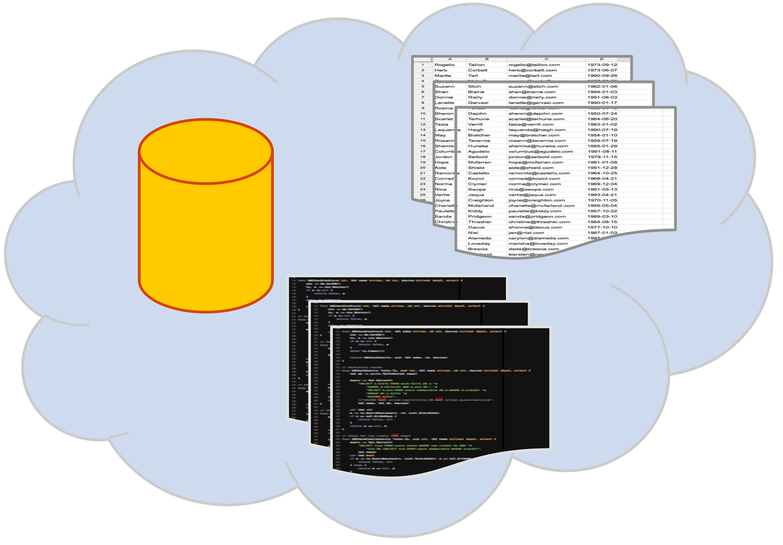
db Trento

Bachelor's degree, Povo(TN) 17-03-2015

Motivating example

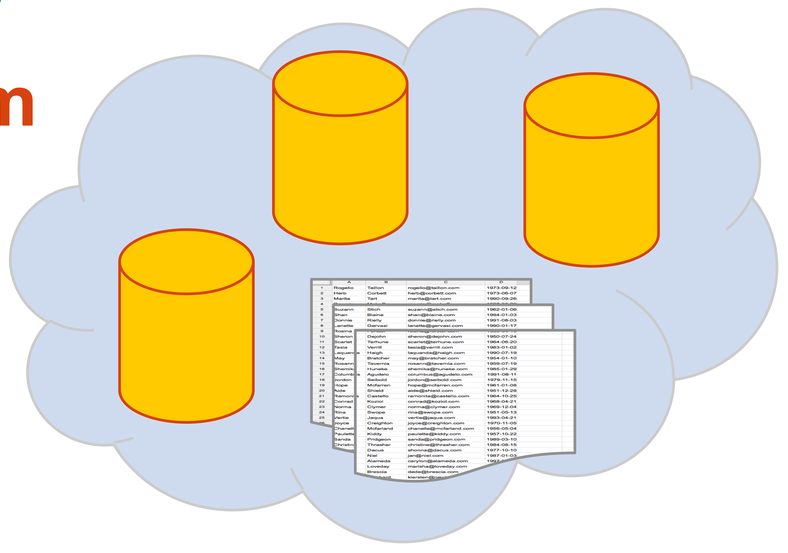
Private Datacenter





Amazon Cloud

Gov. open data portal



Researcher Workstation

? Where is that dataset

? Where does it come from

INDEX

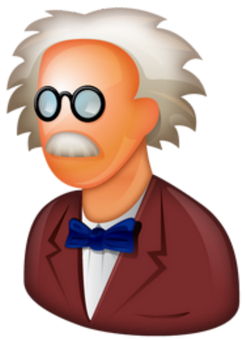
PROVENANCE

dbTrento

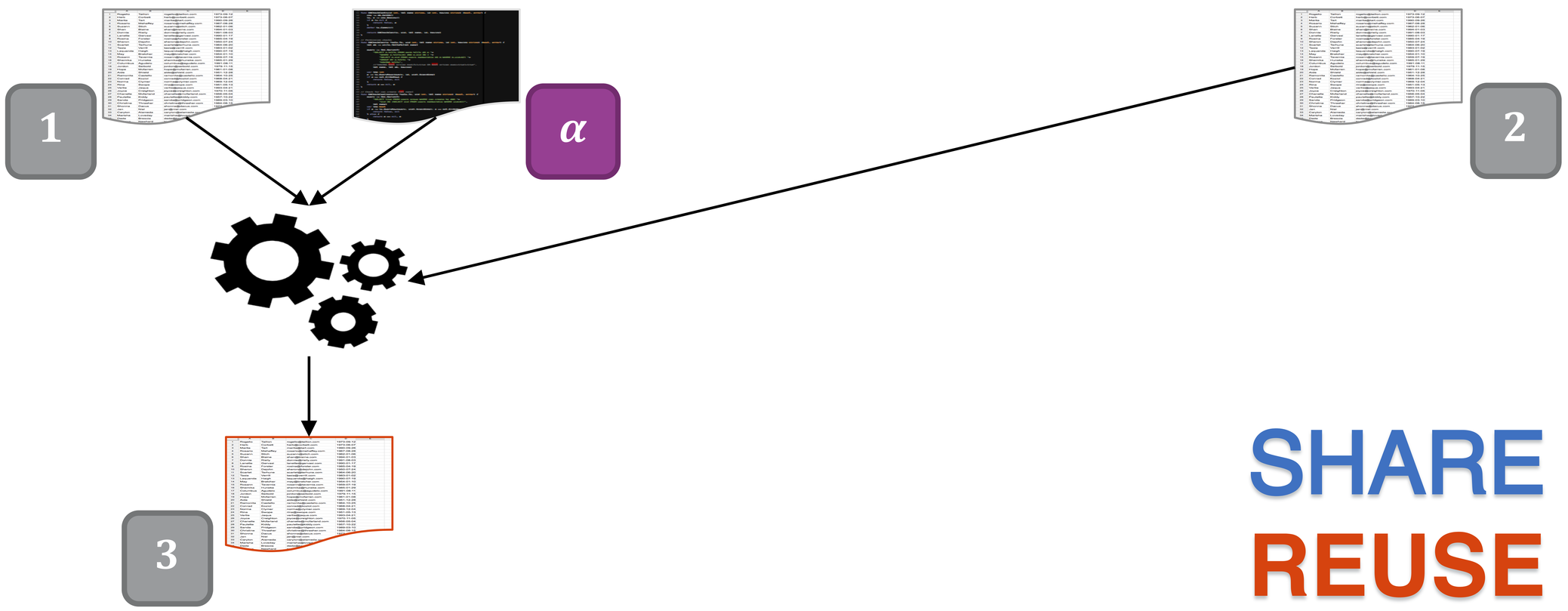
Motivating example

Alice

Computation with 3rd party datasets



Bob



SHARE

REUSE

dbTrento

Challenges

Storage

- Data
- Metadata

Accessing

- Technical
- Legal

Processing

- Scheduling
- Reproducibility
- Validation

Existing systems

Data Stores

- CKAN
- Amazon Public Dataset repository

NO execution

Workflow Managers

- Azkaban
- Oozie

NO provenance

NO index

Mixed Solutions

- Vistrails
- Hue

Desktop

Only Hadoop

Goal

Data Management

- Data portal
- Metadata managing
- Access Control

Workflow Design & Processing

- Operation scheduling
- Process Execution
- Support for different environments

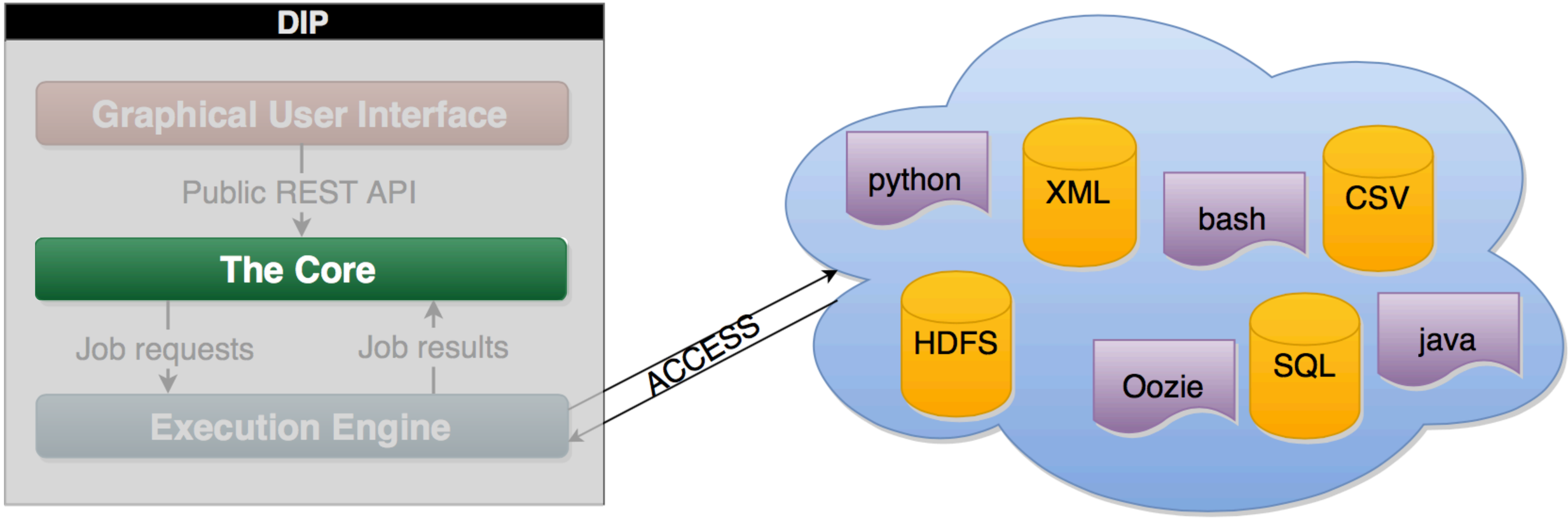
Sharing & Provenance

- Workflow reutilization
- Data Provenance

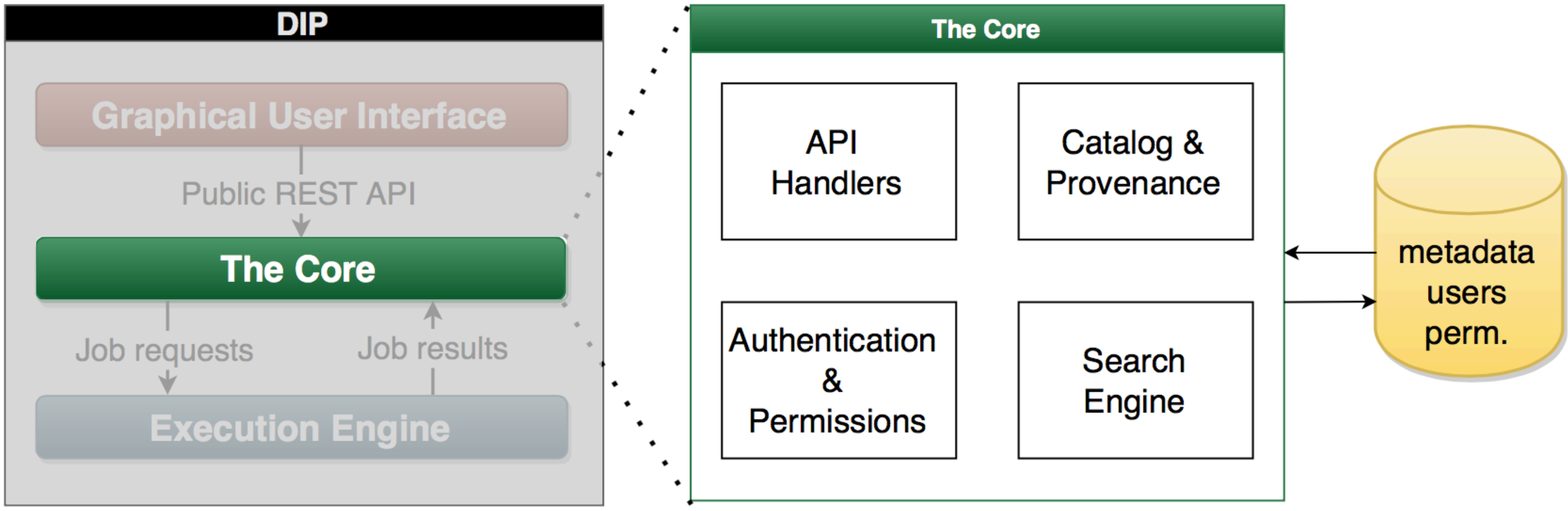
Our Solution: DIP

Built bottom up from scratch.

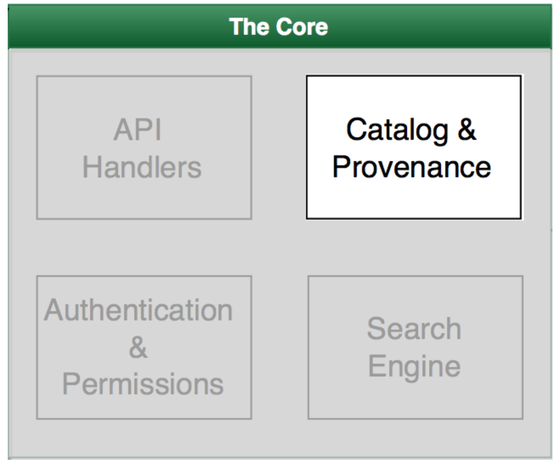
Architecture



The Core


DIP
Graphical User Interface
Public REST API
The Core
Job requests
Job results
Execution Engine
The Core
API Handlers
Catalog & Provenance
Authentication & Permissions
Search Engine
metadata
users
perm.

Catalog & Provenance



Orchestrator for the concept of:

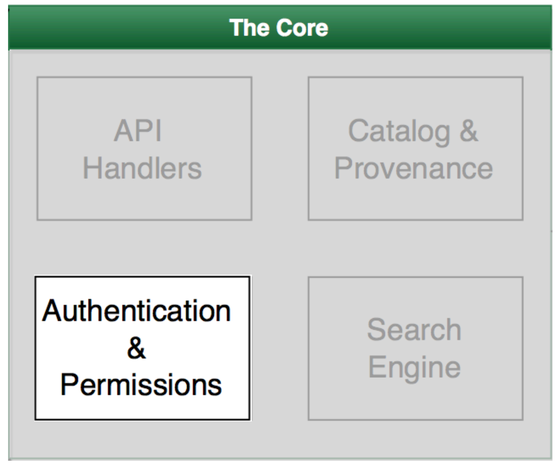
Metadata

Job Serialization

Provenance Graph

dbTrento

Authentication & Permissions



Request Authentication

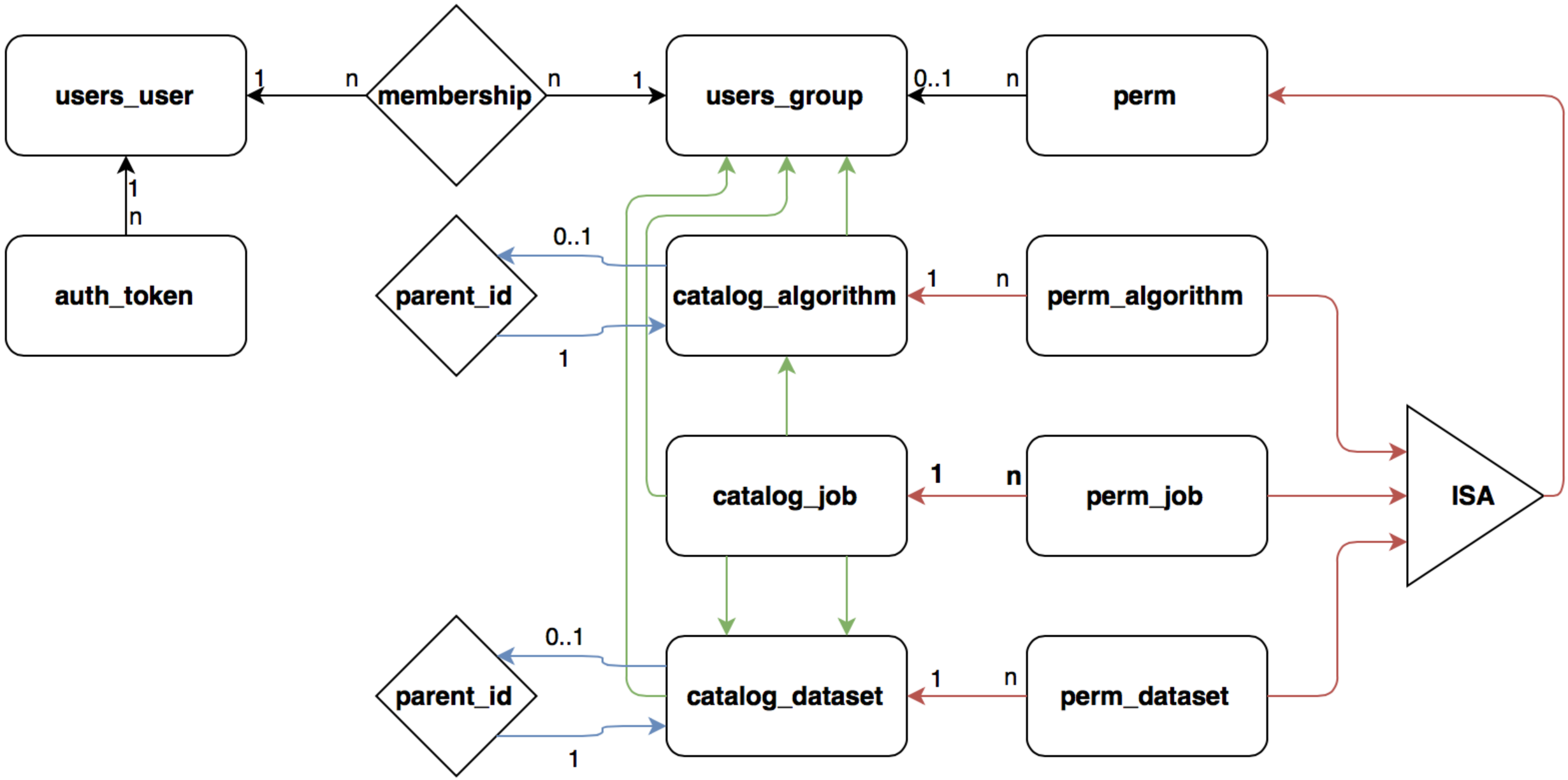
- HTTP Basic Authentication
- Token
- Cookie

Users & Groups

Superuser

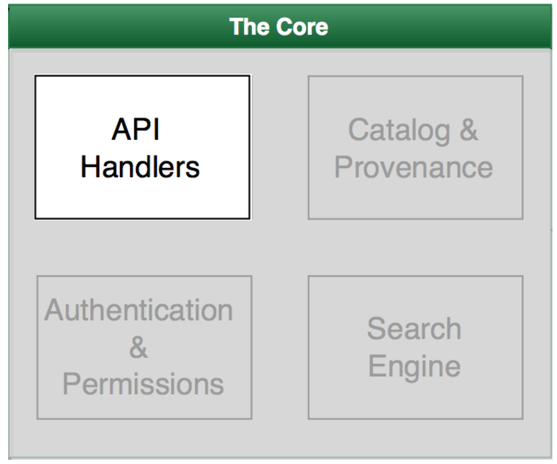
dbTrento

ER diagram



dbTrento

API Handlers



RESTful

JSON

HTTP Status Code

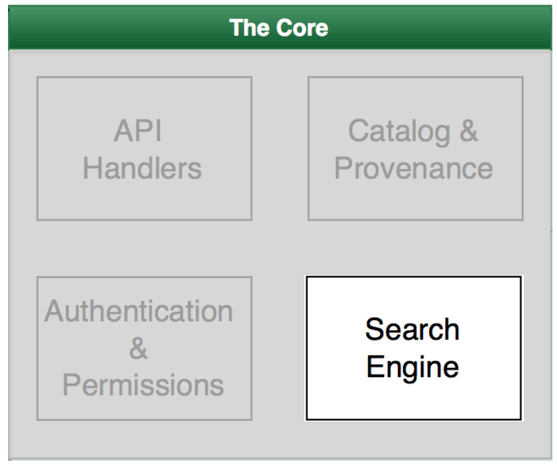
```
GET /dataset/42 HTTP/1.1

[{
    "jid": 23,
    "owner_id": 4,
    "inputs": [2,4,5,12],
    ...
},
...]

HTTP/1.1 200 OK
```


dbTrento

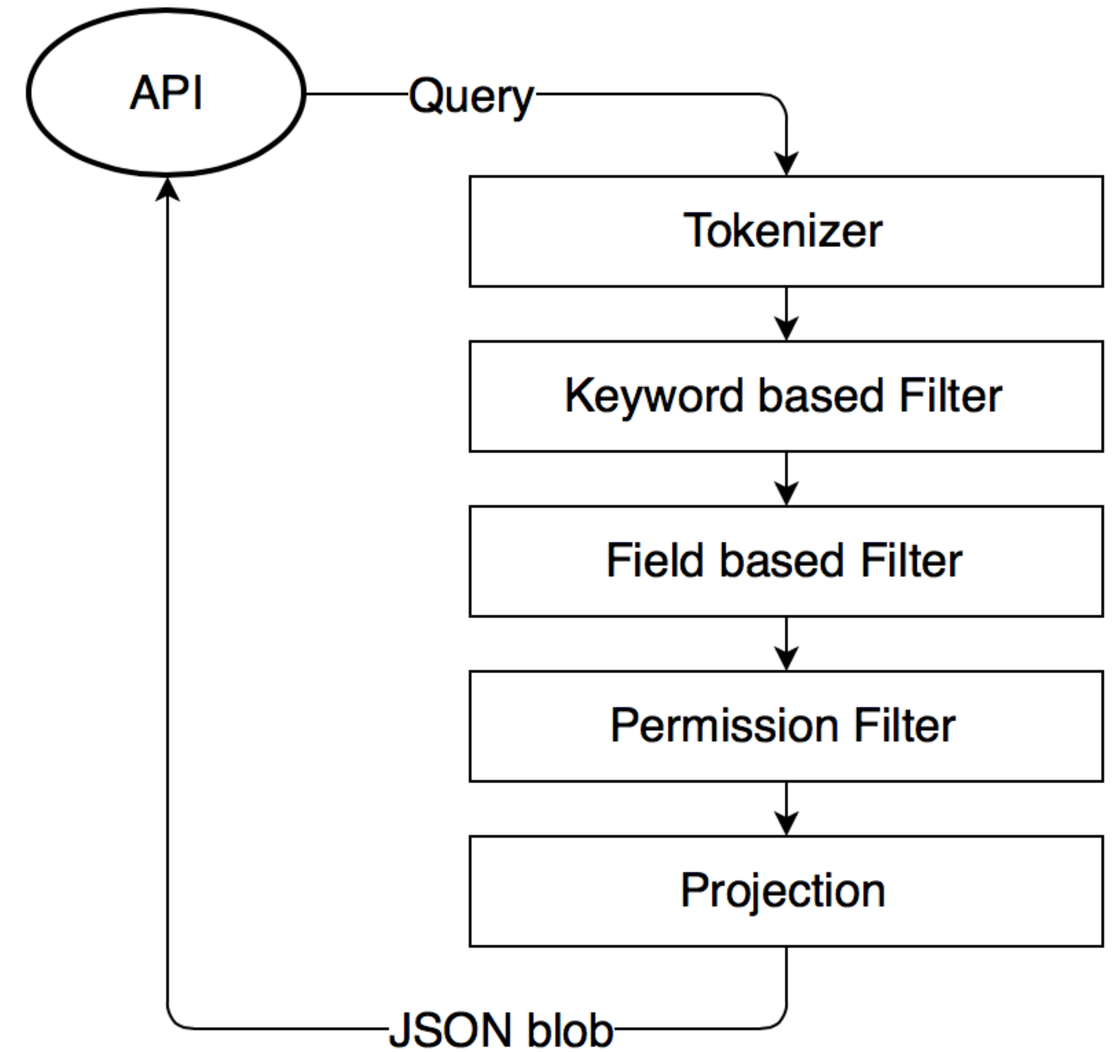
Search Engine



Indexes

Query Parsing

Record Filtering



dbTrento

Used Technologies

- GO 1.4.2
  - absoludity/goforms
  - jmoiron/sqlx
  - drone/routes
- PostgreSQL 9.4.1
  - JSON Type

Conclusion

Data Storage

Workflow Management

Sharing (Access Control)

Provenance Tracking

Thank you!

Questions?

Back-up slides

Why GO?

Concurrency primitives

Fast compilation and deploy

Built-in features

Standard library

PostgreSQL JSON type

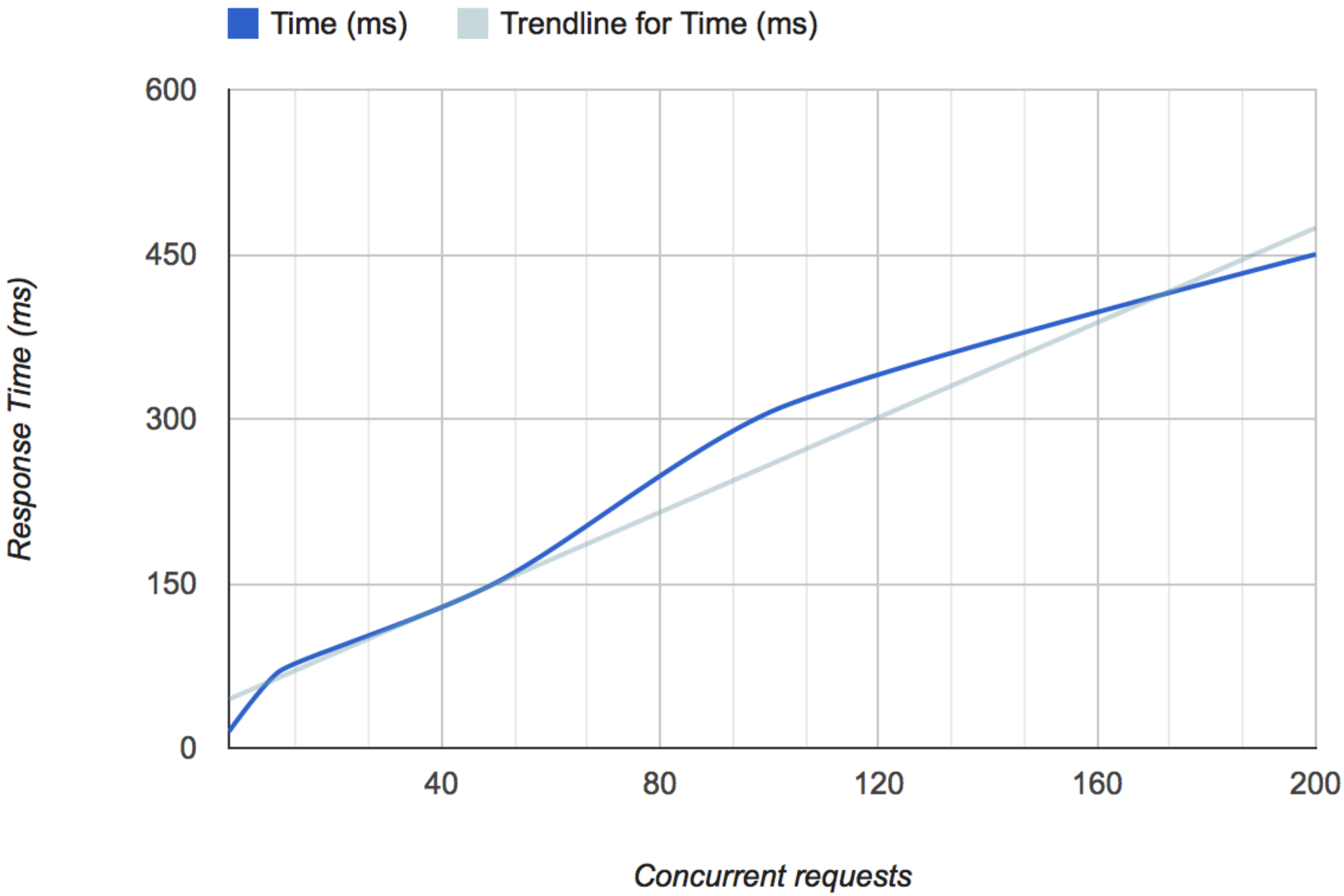
Native support

JSON validity verification

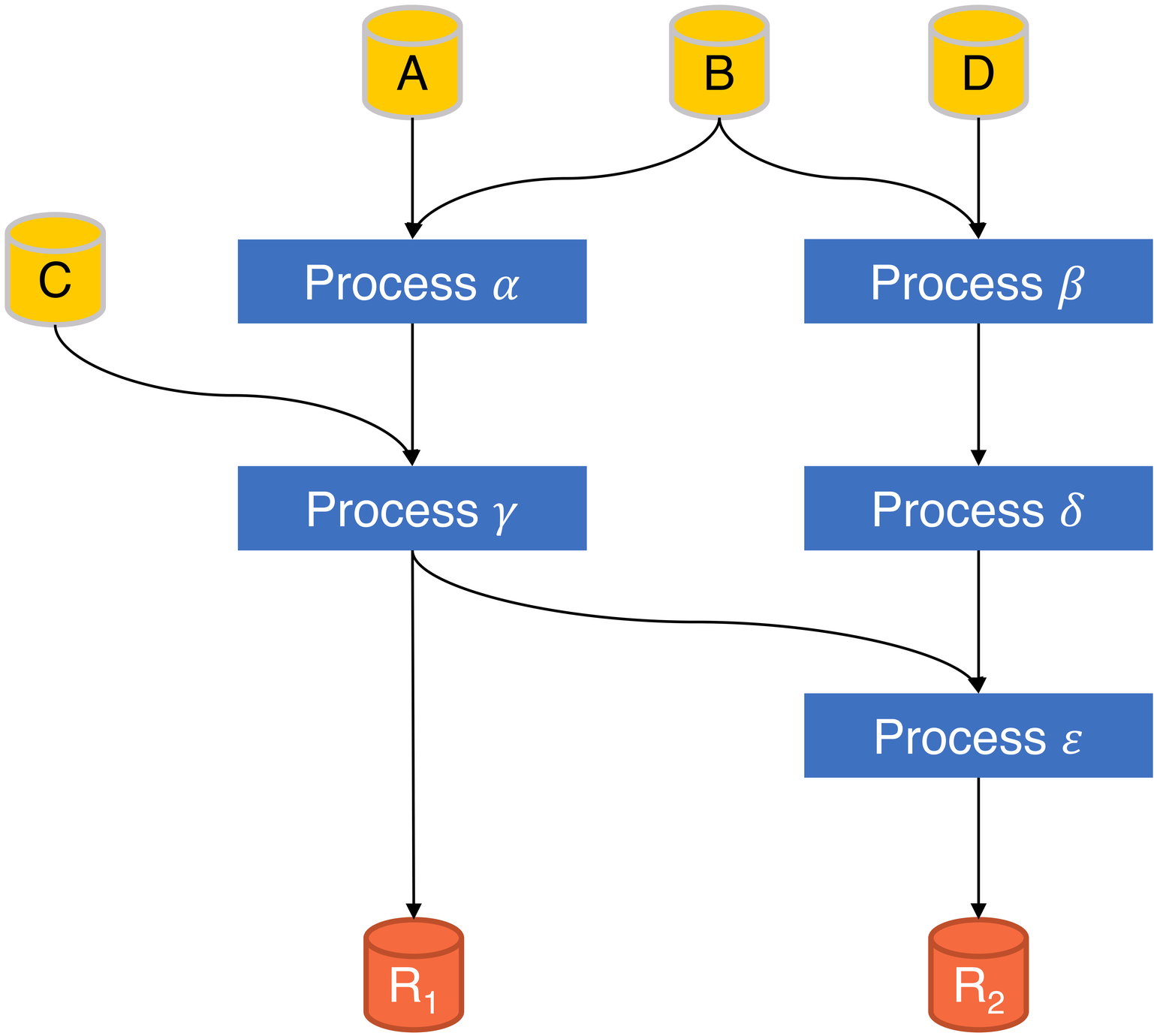
Indexes

Built-in management functionalities

Performance :: Load Test



Workflow


A
B
D
C
Process α
Process β
Process γ
Process δ
Process ε
R_1
R_2