Human Brain Project

Co-funded by
the European Union

Data Management & Data Integration - Analytical Methods & Workflow Engine

Federating Hospital Data

Thanasis Karampatsis — University of Athens

Kostis Karozos — Athens University of Economics & Business

Medical Informatics Platform (MIP)

Executing analyses on data from several hospitals.

Co-funded by
the European Union

Federated Data Processing Engine

Problem:

Patient data should not leave the hospital.

Solution:

Break the algorithm execution in local – global steps.

a) Local steps are executed in the hospitals. They touch the patient data but return aggregate information.

b) The engine aggregates the results of the local nodes and provides them as input to the master node.

c) Global step is run on a master node. Master node performs further computations on the local aggregates.

Federated Data Processing Engine

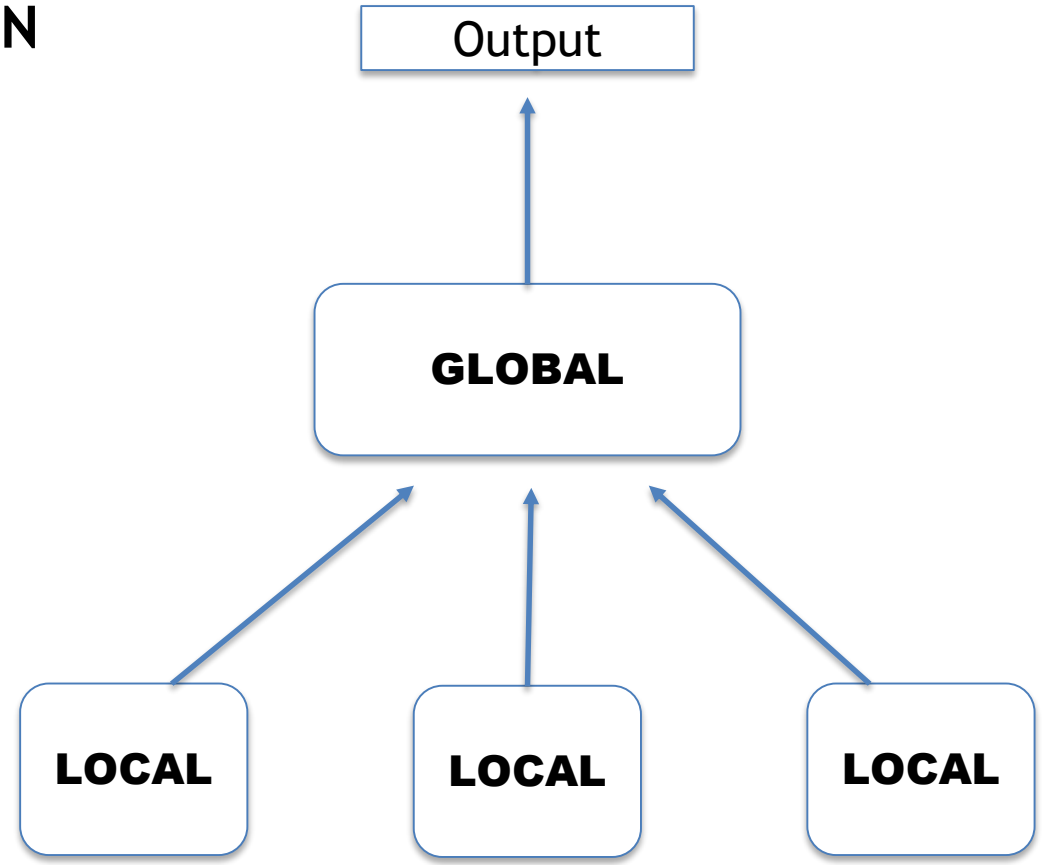
Federation Algorithm Types:

1) Local – Global
2) Multiple Local – Global
3) Iterative Local – Global

Co-funded by
the European Union

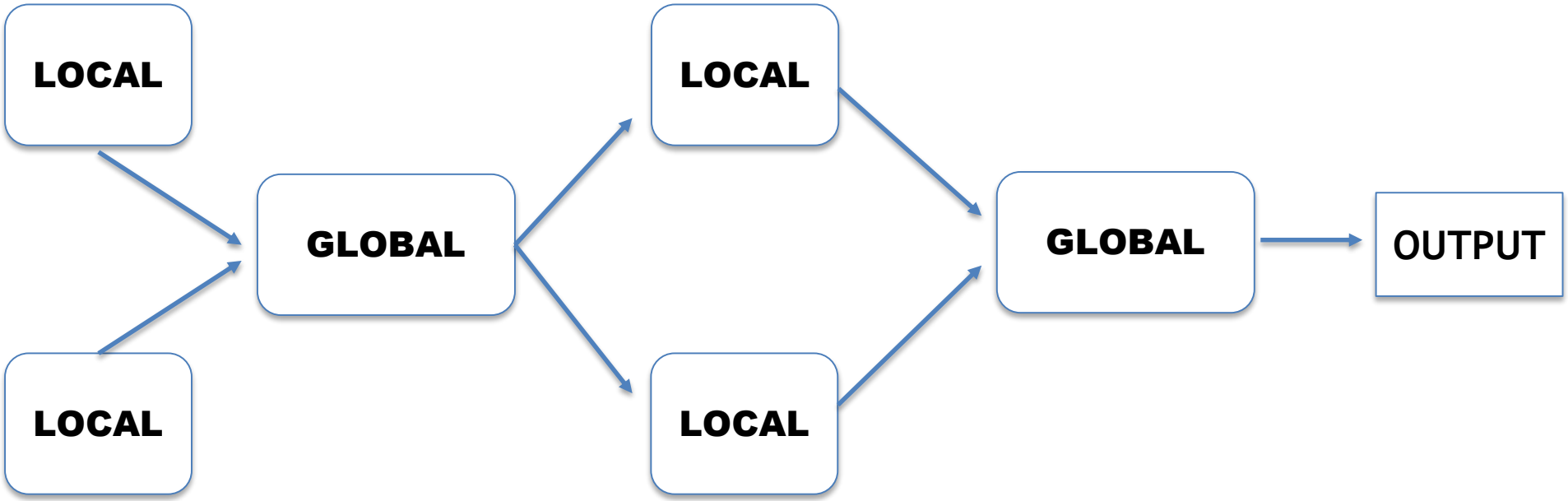
Federated Data Processing Engine

LOCAL – GLOBAL EXECUTION



Federated Data Processing Engine

MULTIPLE LOCAL – GLOBAL EXECUTION



Co-funded by the European Union

Federated Data Processing Engine

How do we create a federated algorithm?

- Implementation in Local and Global scripts (Python).
- A common library with useful functions, e.g. a function for querying the database in a privacy compliant manner.
- Code defining how the aggregation is done on the master node.
- Properties file describing the algorithm parameters/properties.

Co-funded by the European Union

Workflow Engine

Allow users to combine existing algorithms as building blocks in order to create more complex, composite algorithms.

Co-funded by the European Union

Workflow Engine

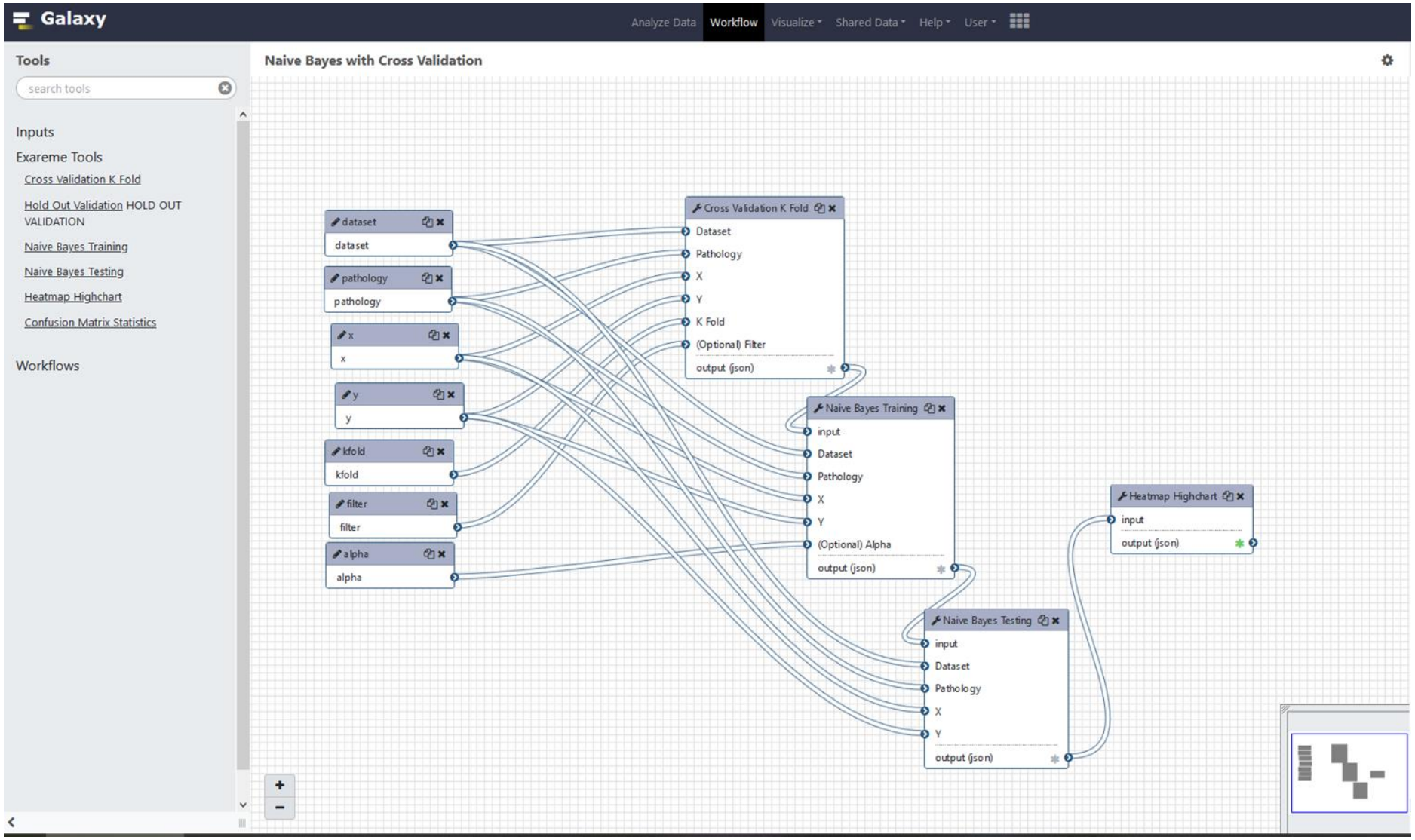
1. Create federated algorithms and integrate them into the WE.

2. Create different visualization techniques.

3. Allow the end user to combine them and create a new workflow.

Co-funded by the European Union

Scientific Workflow Engine



Medical Informatics Platform (MIP)

- Down and Up Streams

/ Downstream (data acquisition, preparation)

  - Inside hospital of origin: Data Factory

/ Upstream (data analysis, modeling, learning)

  - Inside hospital of origin for common diseases
  - Across hospitals for rare diseases

Hospital Data Integration

Problem:

Transform heterogeneous hospital data into a global schema.

One global schema per Medical Condition.

Co-funded by
the European Union

Hospital Data Integration

Solution:

a) Data Capture: Map local hospital data to an I2B2 relational database schema.

b) Data Harmonization: Transform the clinical variables' values according to MIP's rules (ex. brain volumetric variables measured in $cm3^2$)

c) Anonymize

Co-funded by the European Union

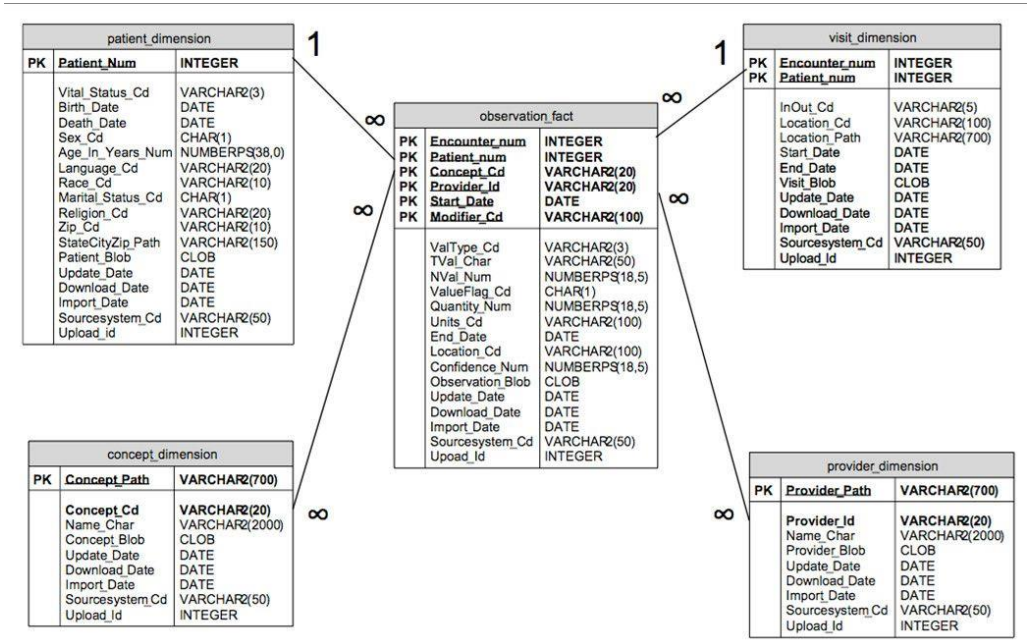
I2B2: Informatics for Integrating Biology and the Bedside

Q: Why use an I2B2-schema db?

A: It can store data with varying schemas, as long as its main Entities are Patient-Visit-Observation.

How?

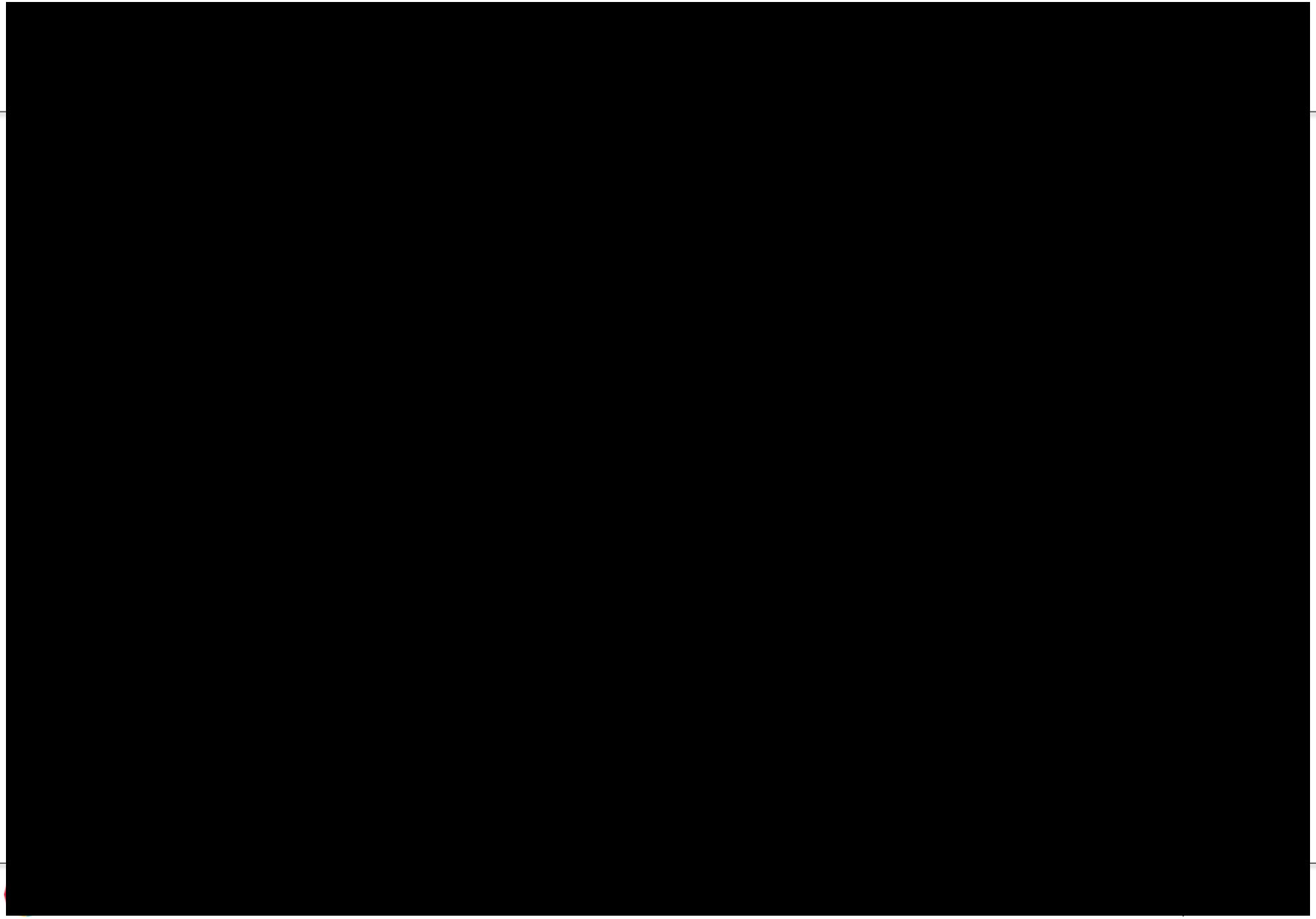
- ➢ Store all information in I2B2 fact table and dimensions
- ➢ Unpivot clinical observation variables
  - store them in attribute-type-value tuples
  - downside: much more tuples



MIPMap

MIPMap: schema mapping and data exchange tool

- Define source-target schema correspondences via MIPMap's GUI → produce mapping tasks.
- A mapping task is a configuration XML file.
- MIPMap's engine translates this XML file into SQL statements and generates the target dataset.

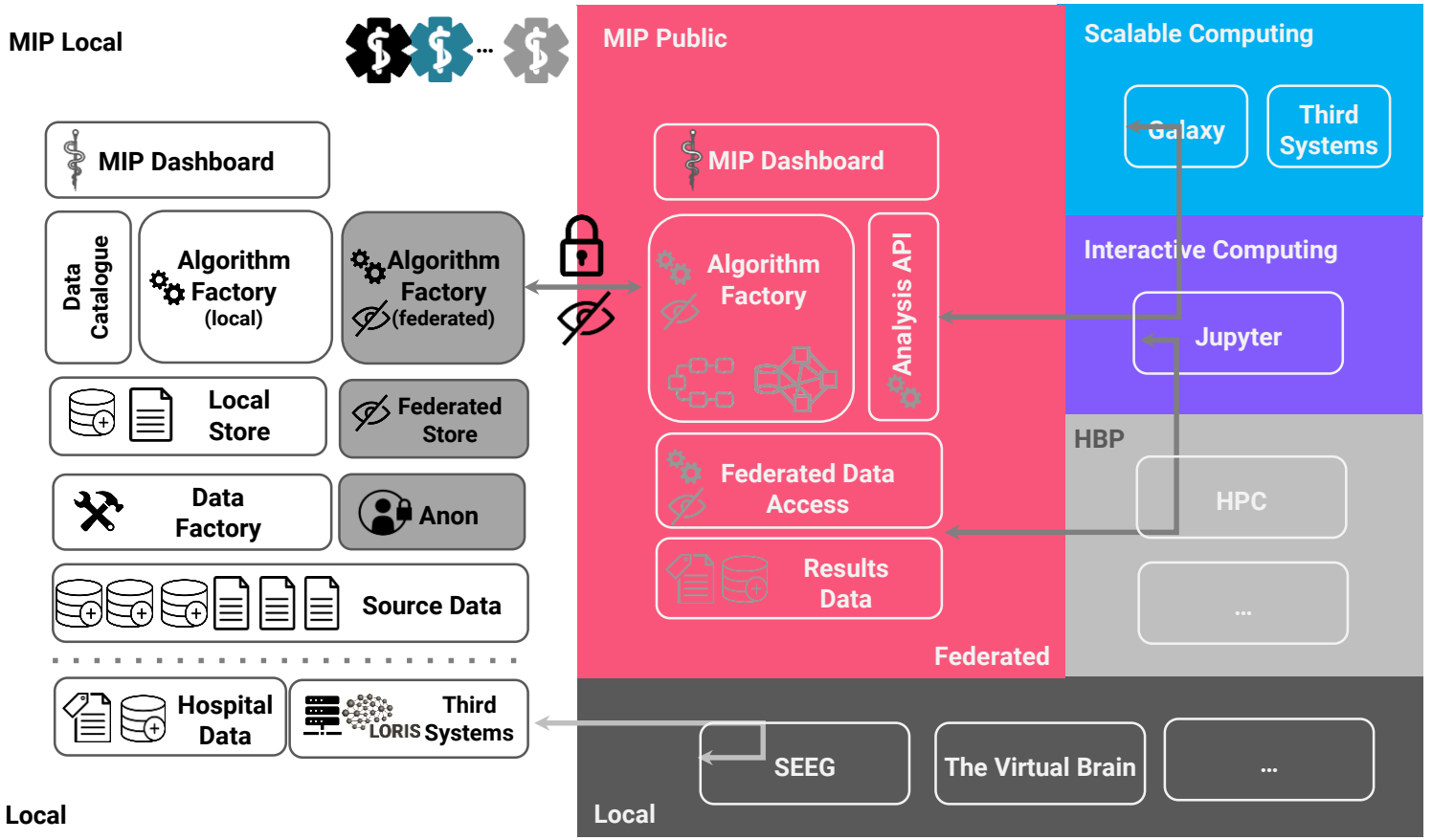


Data Catalogue

- What data has each hospital?
- What are the data descriptions and semantics (metadata)?
- How the local data are mapped to the global schema?

Co-funded by
the European Union

Medical Informatics Platform (MIP)



Data Factory