FET FLAGSHIPS

Human Brain Project
Unifying our understanding of the human brain.

European Commission

Human Brain Project

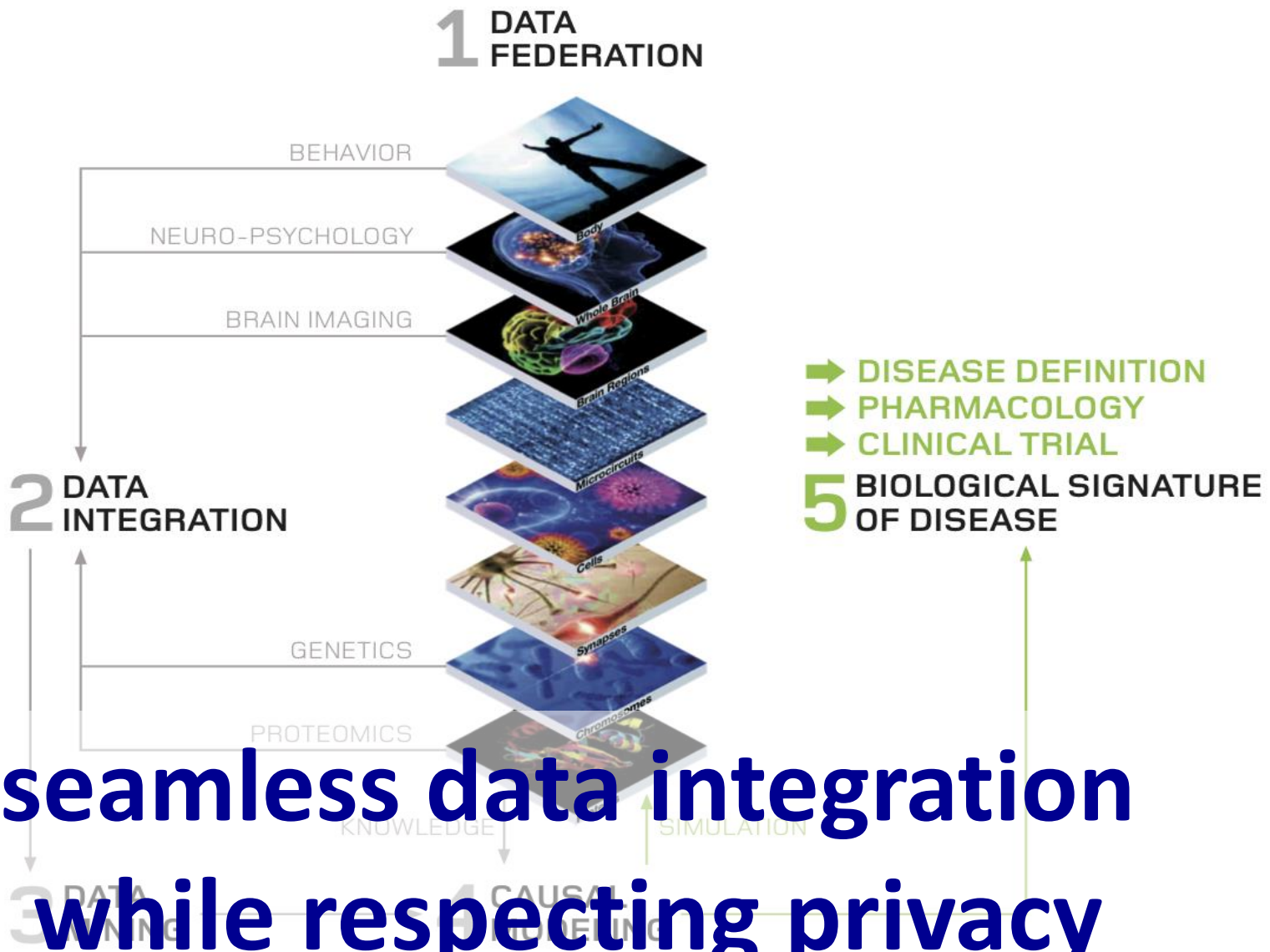
Vasilis Vassalos
vassalos@aueb.gr

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Human Brain Project

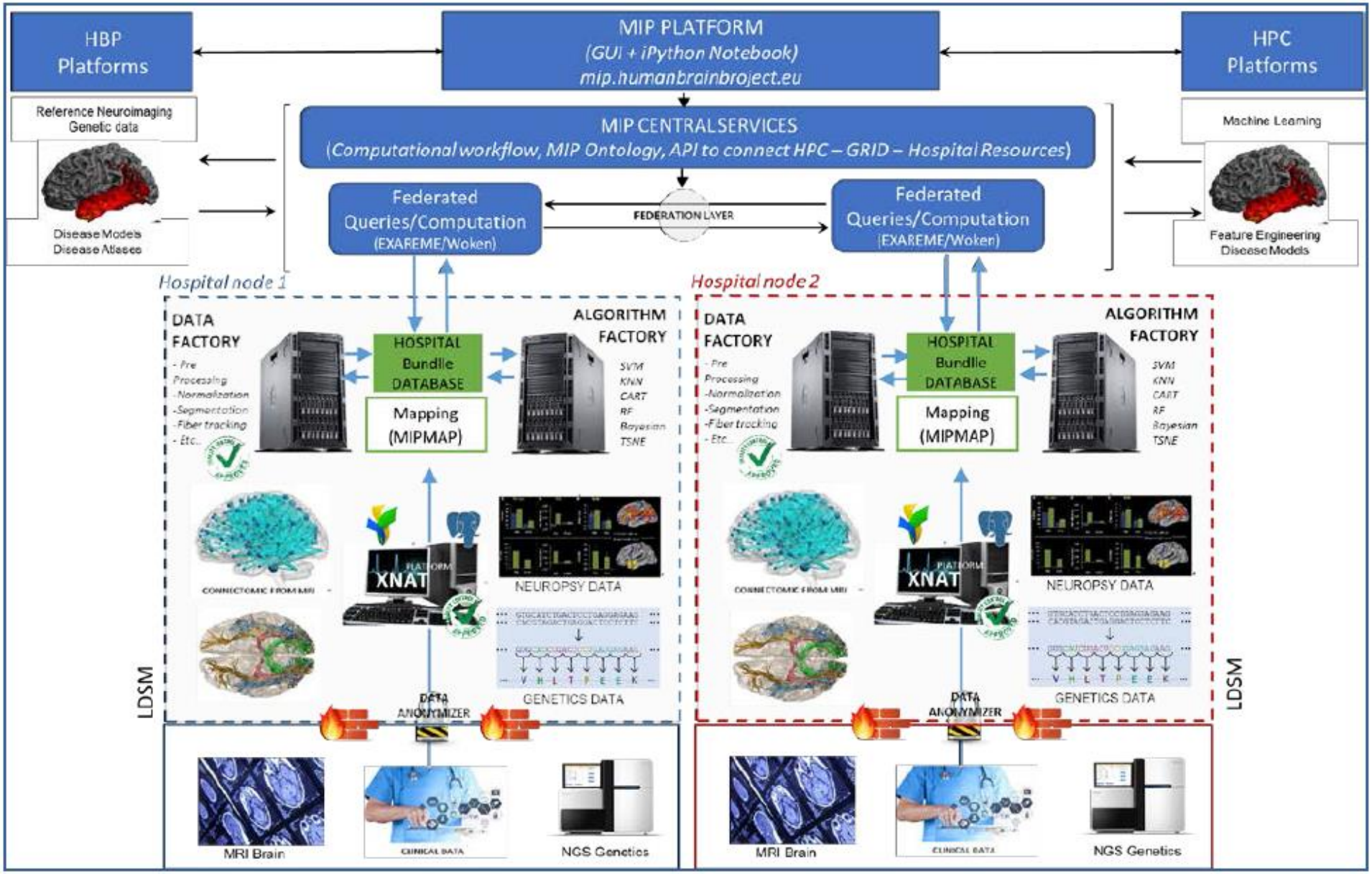
- **Goal**
  - **What makes us human?**
  - **Brain Disease Treatments**
  - **Computation Technologies**
- **Six Platforms**
  - Neuroinformatics
    - organizes neuroscience data from the HBP and beyond
  - Brain Simulation
    - internet-accessible collaborative Platform designed for reconstruction and simulation of brain models
  - High Performance Computing
    - developes and provides supercomputing, storage, visualisation and simulation technology that can run on supercomputers.
  - Neuromorphic Computing
    - implements aspects of biological neural networks as analogue or digital copies on electronic circuits.
  - Neurorobotics
    - Experiment with virtual robots connected to simulated brains.
  - **Medical Informatics Platform**

Medical Informatics Platform



seamless data integration while respecting privacy

MIP Architecture

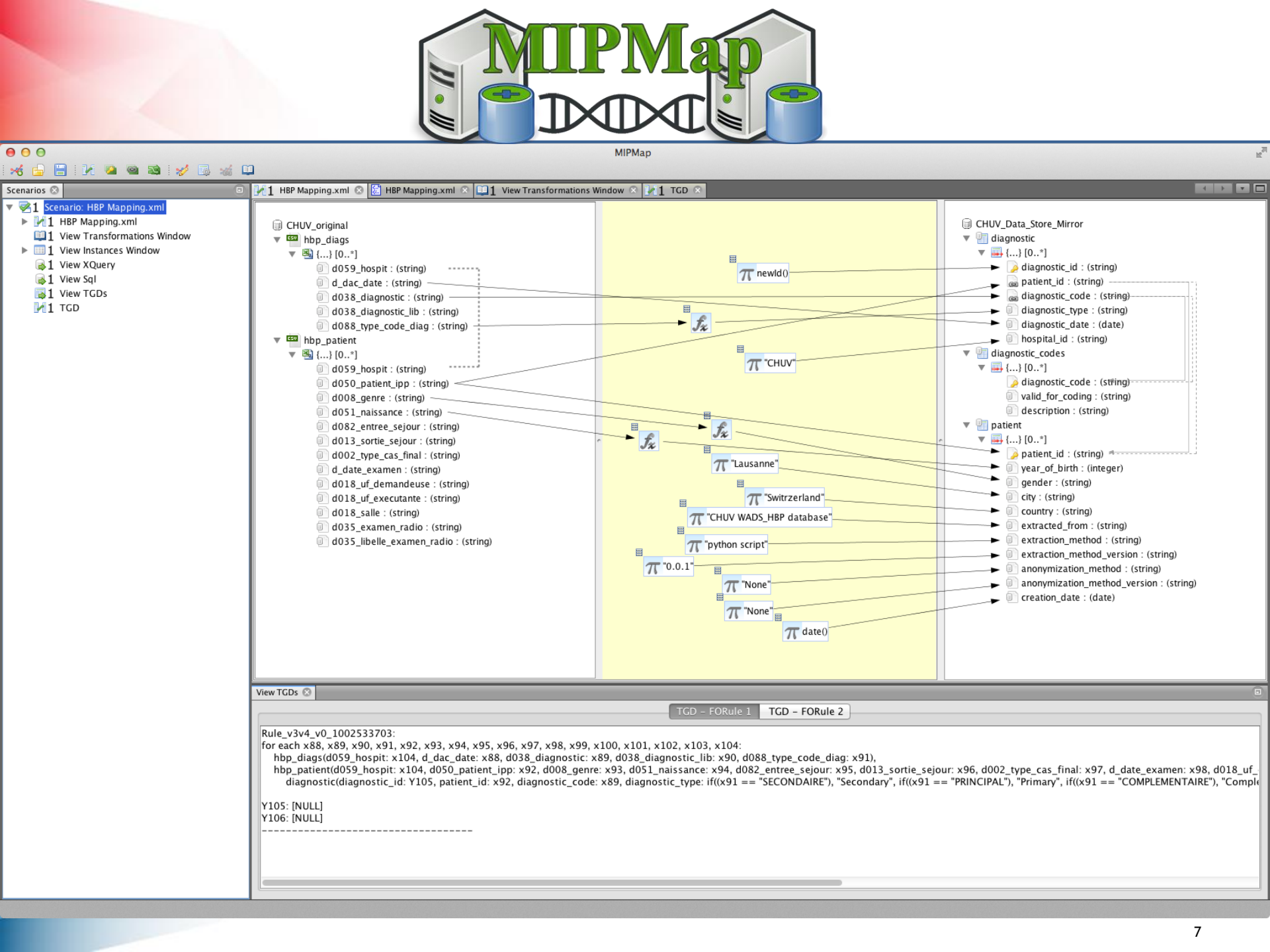


Data Integration Challenges

- **Web Portal**
  - **Variable extraction**
  - **Standardization**
- **Federation Layer**
  - **Query rewriting and Query planning**
  - **Identify pieces of answers, retrieve them, combine them *and* maintain privacy/anonymity**
    - Mapping Expressivity vs Computational Complexity
    - Correct and efficient distributed execution
- **Local Layer**
  - **Schema Creation for Data Store Mirrors**
  - **Schema mapping**
    - Mapping Definition: What maps to what, under what conditions
    - Mapping Execution/Data Exchange/Conversion
      - » Expressivity vs Computational Complexity
      - » Efficient execution
      - » Anonymity/privacy

Data Integration Challenges

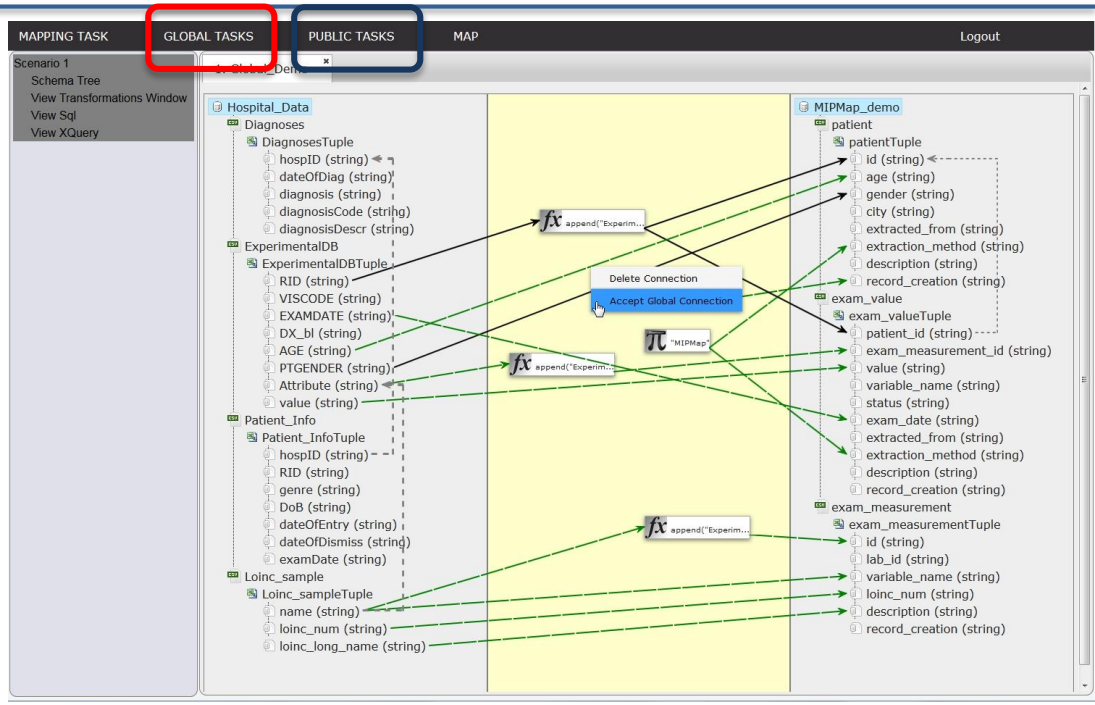
- **Web Portal**
  - Variable extraction
  - Standardization
- **Federation Layer**
  - Query rewriting and Query planning
  - Identify pieces of answers, retrieve them, combine them *and* maintain privacy/anonymity
  - Perform complex mining tasks maintaining privacy/anonymity
- **Local Layer**
  - Variable extraction
  - Standardization
  - Schema Creation for Data Store Mirrors
  - Population of Local Data Store Mirrors

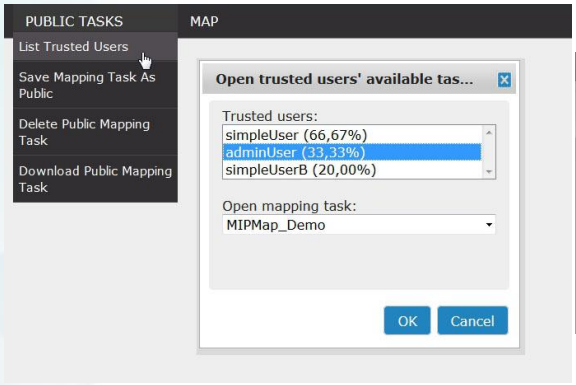

MIPMap
MIPMap
Scenarios
Scenario: HBP Mapping.xml
HBP Mapping.xml
View Transformations Window
View Instances Window
View XQuery
View Sql
View TGDs
TGD
HBP Mapping.xml
HBP Mapping.xml
View Transformations Window
TGD
CHUV_original
hbp_diags
{...} [0..*]
d059_hospit : (string)
d_dac_date : (string)
d038_diagnostic : (string)
d038_diagnostic_lib : (string)
d088_type_code_diag : (string)
hbp_patient
{...} [0..*]
d059_hospit : (string)
d050_patient_ipp : (string)
d008_genre : (string)
d051_naissance : (string)
d082_entree_sejour : (string)
d013_sortie_sejour : (string)
d002_type_cas_final : (string)
d_date_examen : (string)
d018_uf_demandeuse : (string)
d018_uf_executante : (string)
d018_salle : (string)
d035_examen_radio : (string)
d035_libelle_examen_radio : (string)
newId()
"CHUV"
"Lausanne"
"Switzerland"
"CHUV WADS_HBP database"
"python script"
"0.0.1"
"None"
"None"
date()
CHUV_Data_Store_Mirror
diagnostic
{...} [0..*]
diagnostic_id : (string)
patient_id : (string)
diagnostic_code : (string)
diagnostic_type : (string)
diagnostic_date : (date)
hospital_id : (string)
diagnostic_codes
{...} [0..*]
diagnostic_code : (string)
valid_for_coding : (string)
description : (string)
patient
{...} [0..*]
patient_id : (string)
year_of_birth : (integer)
gender : (string)
city : (string)
country : (string)
extracted_from : (string)
extraction_method : (string)
extraction_method_version : (string)
anonymization_method : (string)
anonymization_method_version : (string)
creation_date : (date)
View TGDs
TGD - FORule 1
TGD - FORule 2
Rule_v3v4_v0_1002533703:
for each x88, x89, x90, x91, x92, x93, x94, x95, x96, x97, x98, x99, x100, x101, x102, x103, x104:
hbp_diags(d059_hospit: x104, d_dac_date: x88, d038_diagnostic: x89, d038_diagnostic_lib: x90, d088_type_code_diag: x91),
hbp_patient(d059_hospit: x104, d050_patient_ipp: x92, d008_genre: x93, d051_naissance: x94, d082_entree_sejour: x95, d013_sortie_sejour: x96, d002_type_cas_final: x97, d_date_examen: x98, d018_uf_
diagnostic(diagnostic_id: Y105, patient_id: x92, diagnostic_code: x89, diagnostic_type: if((x91 == "SECONDAIRE"), "Secondary", if((x91 == "PRINCIPAL"), "Primary", if((x91 == "COMPLEMENTAIRE"), "Compl
Y105: [NULL]
Y106: [NULL]

Community Schema Curation Module

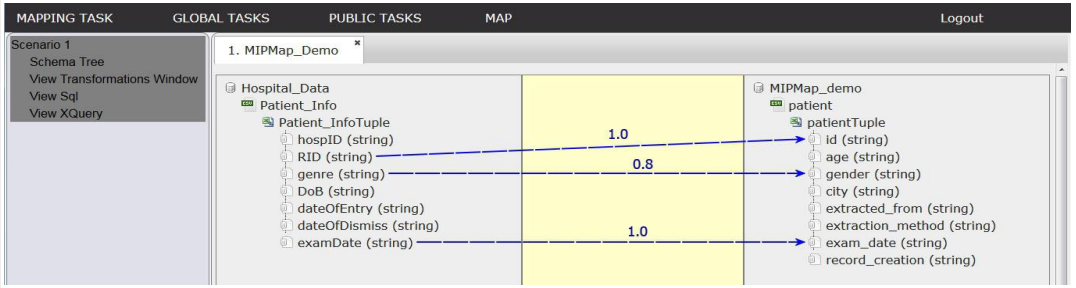
Share Global Mappings

- Share Public Mappings
- Endorse (partial) Global/Public Mappings
- Merge and extend existing schemata

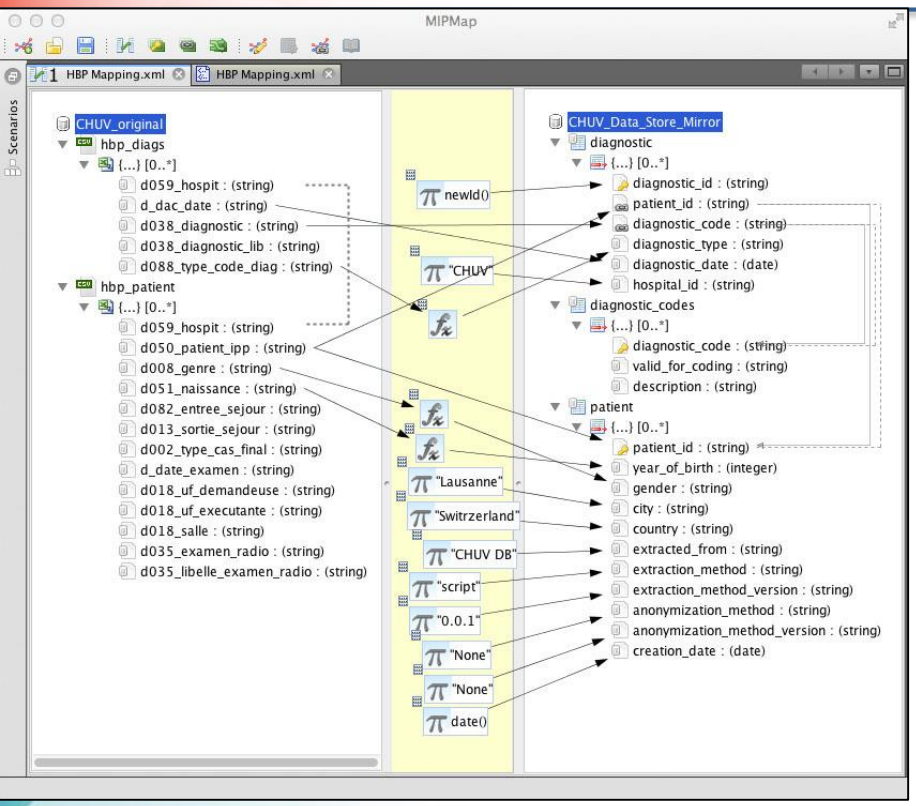




Aggregate Mappings from Trusted Users



Online Data Integration Module



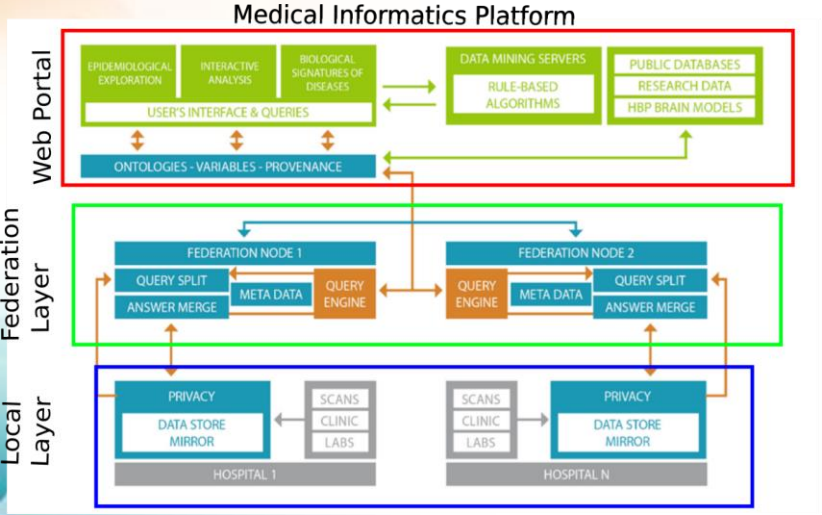
- 1st batch of Data
  - patient A has no diagnostics
  - rule not activated for A
- 2nd batch of Data
  - diagnostics are received for patient A
  - rule not activated for A
- Need to re-run for all data
- Incremental Data Exchange
  - avoid recomputing

- Rule
  - patients with diagnostics populate the Target diagnostic table

Ontology-Based Data Access Module

I want to know the number of cases suffering from some form of Dementia.

So many possibilities:
Dementia can be caused by diseases with ICD-10 codes G30.9 (for Alzheimers), or G20 (for Parkinsons), or G31 (for Pick's dementia), or G10 (for Huntington's), or ...



– Translate to low-level SQL using the mappings and evaluate it over the data

```
SELECT d.pid
FROM diagnostic AS d
WHERE d.code="G30.9"
  UNION
SELECT d.pid
FROM diagnostic AS d
WHERE d.code="G20"
  UNION
...
```

Ontologies

– Represent domain knowledge in an abstract way
e.g., SNOMED CT

$$\mathsf{AlzheimersDementia} \sqsubseteq \mathsf{Dementia}$$
$$\mathsf{ParkinsonsDementia} \sqsubseteq \mathsf{Dementia}$$
$$\mathsf{PicksDementia} \sqsubseteq \mathsf{Dementia}$$
$$\mathsf{ClostridiumNovyi} \sqsubseteq \exists \mathsf{hasActiveIngredient}.\mathsf{Toxoid}$$
$$\ldots$$

Ontology-Based Query Answering

– Issue queries using ontology vocabulary and not low data-level details (ICD codes, etc)

$$\mathcal{Q} = Q(x) \leftarrow \mathsf{Pat}(x) \wedge \mathsf{hasEff}(x, y) \wedge \mathsf{Dem}(y)$$

– Translate/Rewrite/Expand User Query Using a **Reasoning System**

Compute so-called rewriting $\mathcal{R}$

$$\mathcal{Q}_1 = Q(x) \leftarrow \ldots \wedge \mathsf{AlzheimersDementia}(y)$$
$$\mathcal{Q}_2 = Q(x) \leftarrow \ldots \wedge \mathsf{ParkinsonsDementia}(y)$$
$$\mathcal{Q}_3 = Q(x) \leftarrow \ldots \wedge \mathsf{PicksDementia}(y)$$
$$\ldots$$
$$\mathcal{R} = \{\mathcal{Q}, \mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3, \ldots\}$$

Retrieving the Answers

– Low level mappings dictate how data are mapped to ontology terms

$$\mathsf{diagnostic}(\mathbf{pid}, G30.9, \mathbf{date}, \text{“CHUV”}) \rightsquigarrow_{m_1} \mathsf{Patient}(\mathbf{pid}) \wedge \mathsf{hasEffect}(\mathbf{pid}, ad) \wedge \mathsf{AlzheimersDementia}(ad)$$

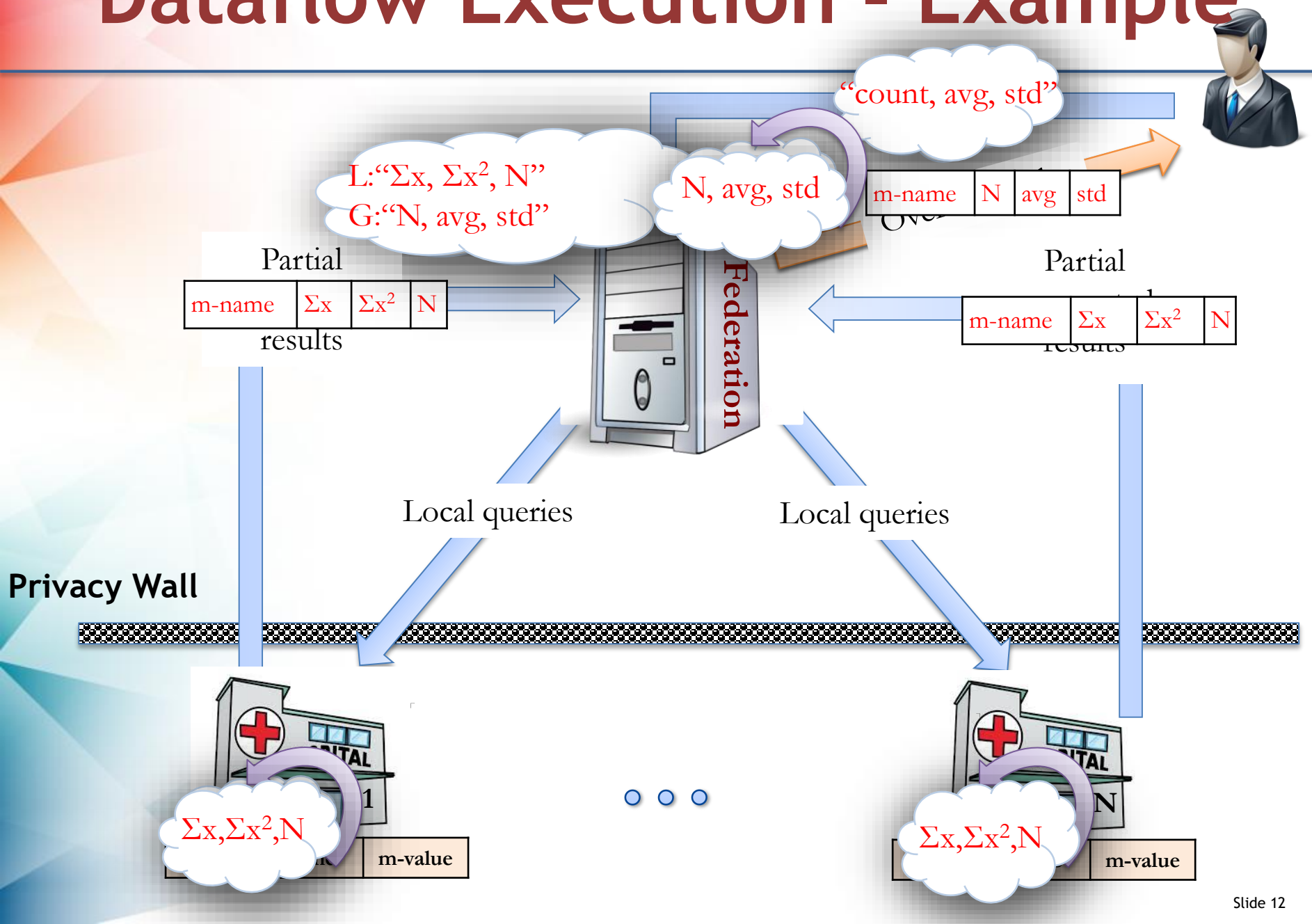
$$\mathsf{diagnostic}(\mathbf{pid}, G20, \mathbf{date}, \text{“CHUV”}) \rightsquigarrow_{m_2} \mathsf{Patient}(\mathbf{pid}) \wedge \mathsf{hasEffect}(\mathbf{pid}, pd) \wedge \mathsf{ParkinsonsDementia}(pd)$$

$$\ldots$$

Complex Distributed Workflows

- Characteristics
  - Complex – nontrivial algorithms, e.g., data mining
  - Distributed – executed over all relevant hospitals
- Privacy preservation
  - Moving only aggregates (sufficient statistics) of hospital data
- Data operator support
  - User defined Functions (UDFs)
  - Iteration

Dataflow Execution - Example



Human Brain Project

Unifying our understanding of the human brain.

Thank you!