

ViroLab Virtual Laboratory

Marian Bubak
AGH Krakow, UvA Amsterdam
bubak@agh.edu.pl

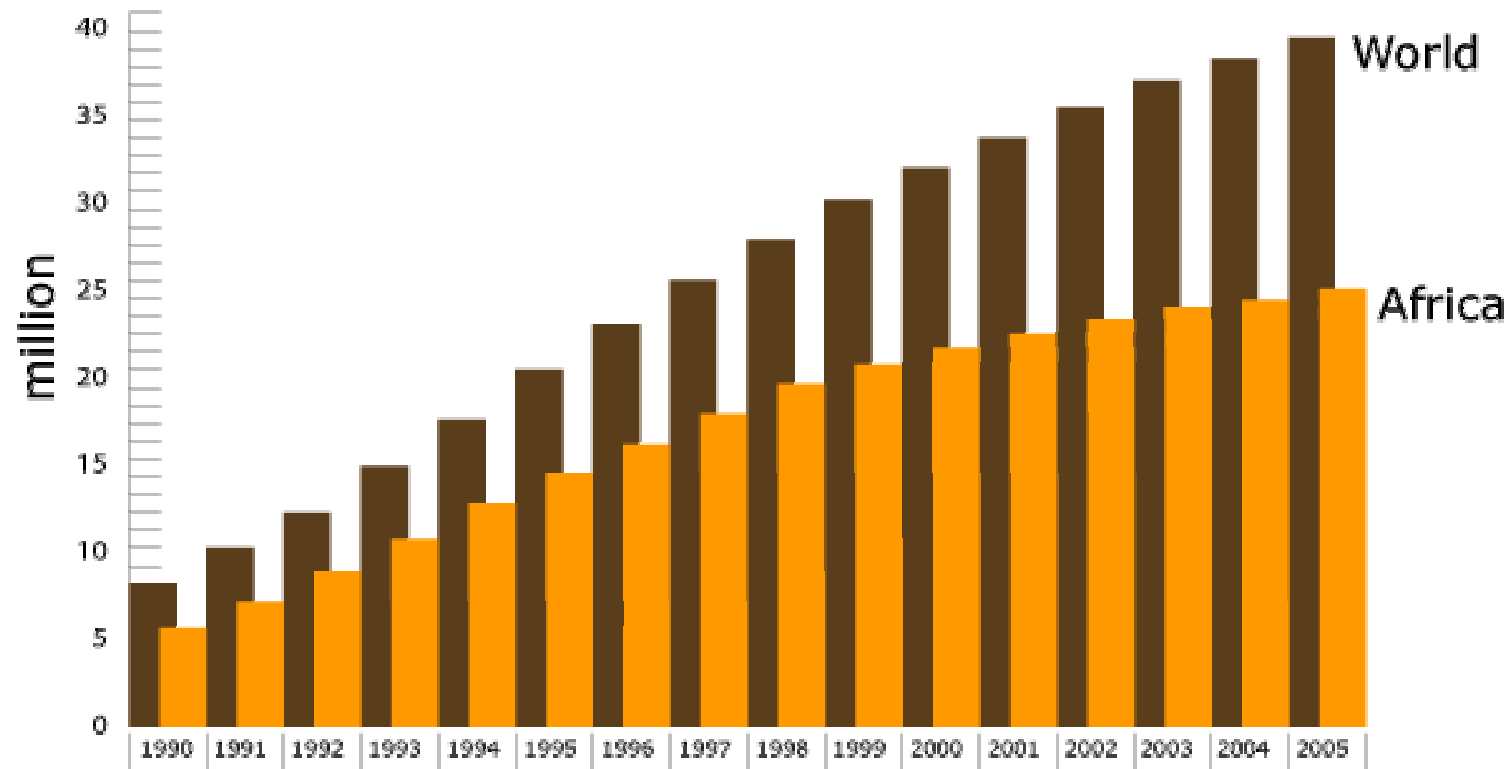
www.virolab.org

virolab.cyfronet.pl

Outline

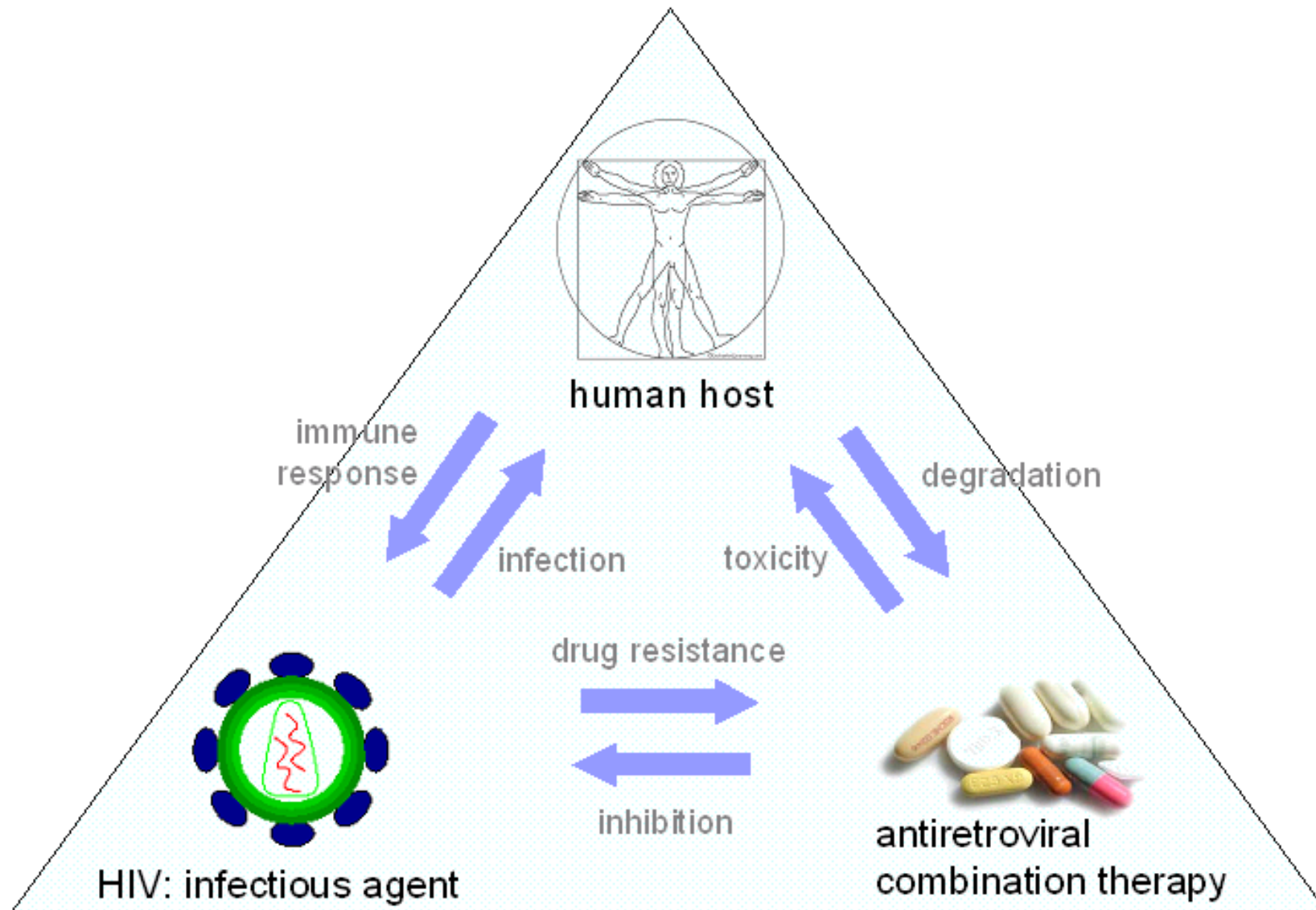
- Objectives of the ViroLab project
- ViroLab applications and users
- Structure of the virtual laboratory
- Virtual laboratory in action
- Challenges
- Concluding remarks

HIV Epidemics

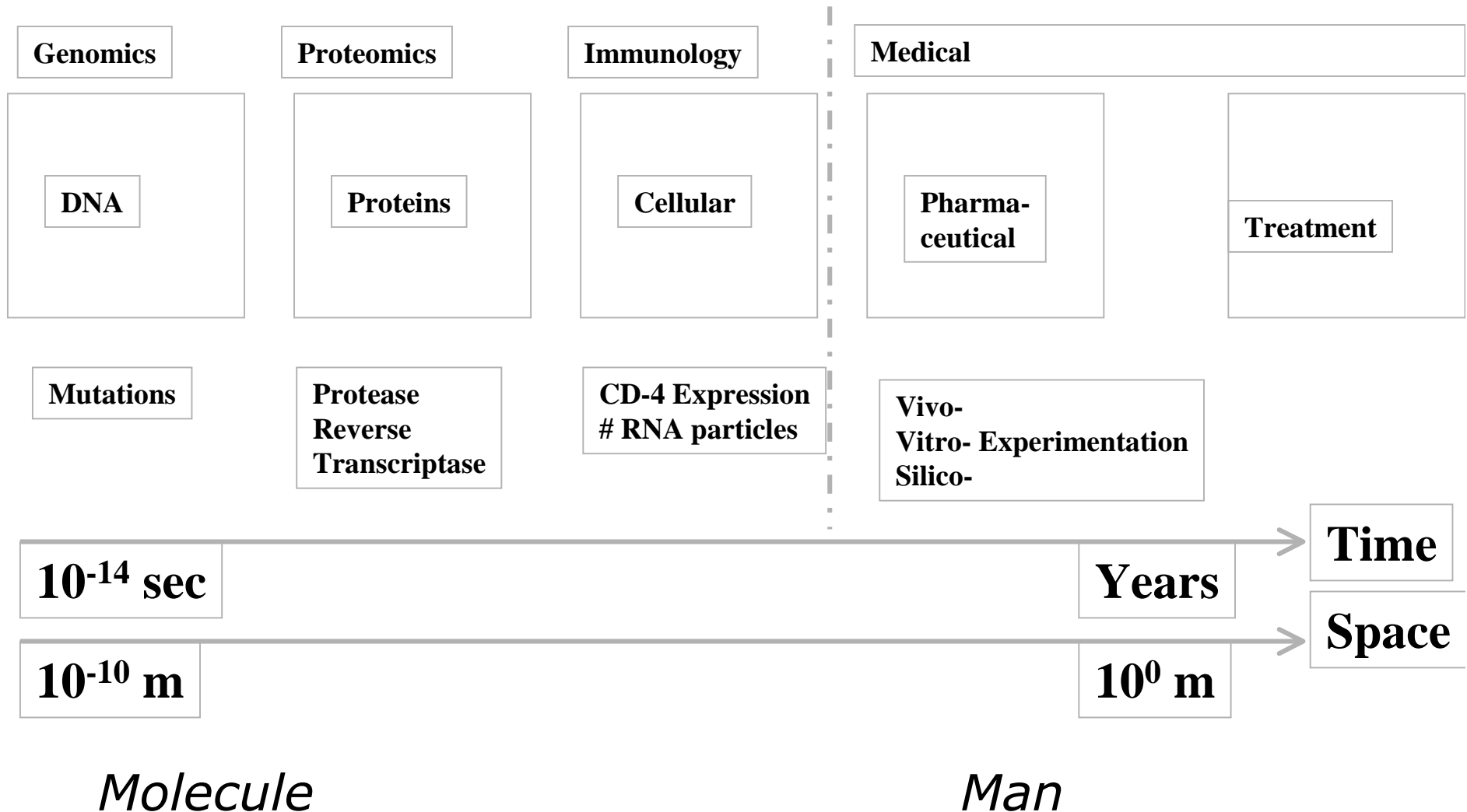


The number of people living with HIV has risen from around 8 million in 1990 to nearly 40 million today.

The complex interplay in HiV



From Molecule to Man



ViroLab Applications

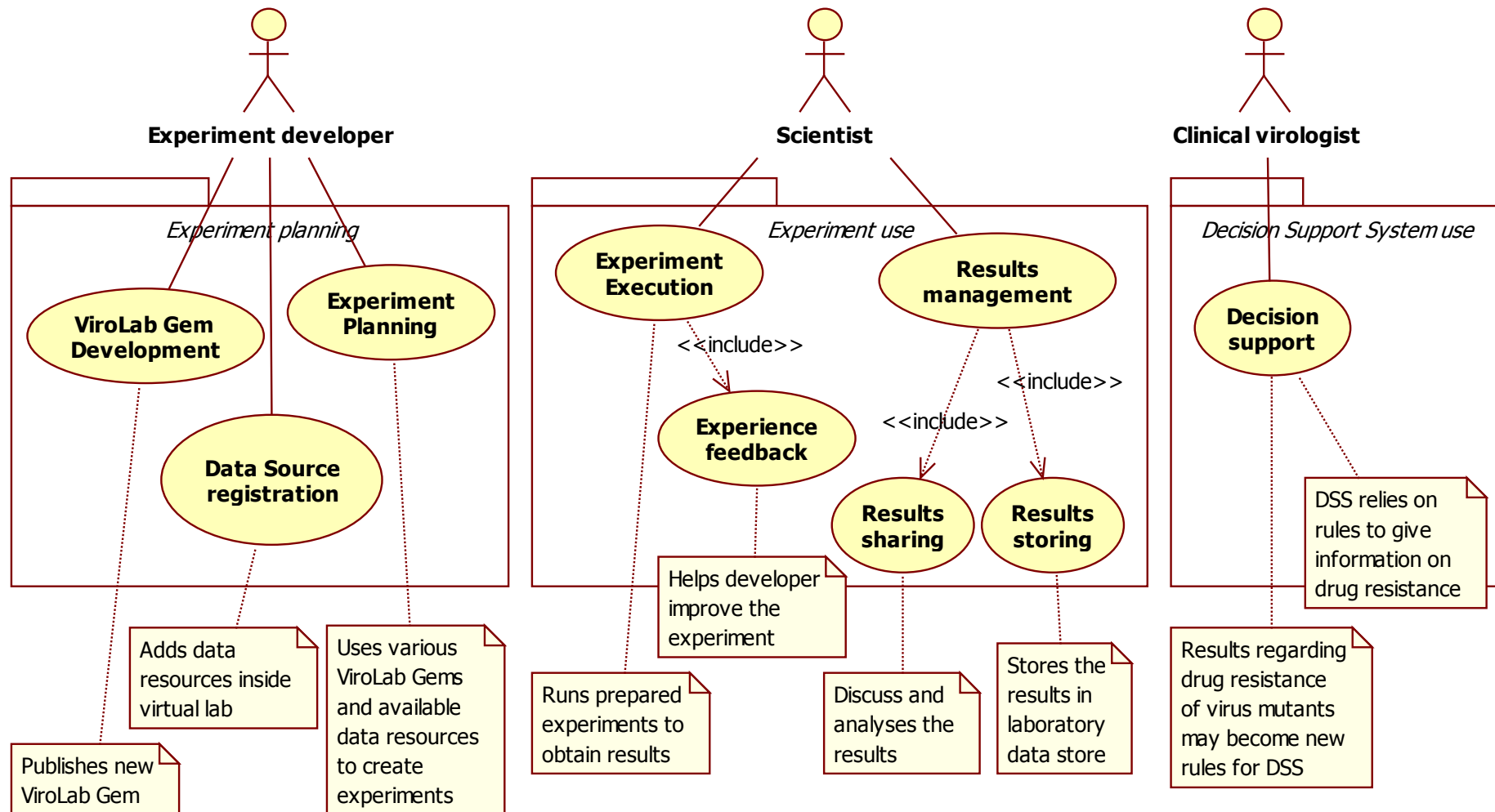
The following ViroLab applications have been analyzed:

- Drug ranking system
- From genotype to drug resistance
- Establishing large databases of HIV sequences
- Data retriever
- Predictor evaluator
- Molecular dynamics and CA simulations

Classes of Users

- **Experiment developer (scientific programmer)**
 - Plans a ViroLab experiment using the VLvl development tools
 - Knows both how to script an experiment and the modelled domain
 - May develop dedicated UI for an experiment user
- **Experiment user (research-oriented user, scientist)**
 - Uses prepared experiment to gather scientific results (via Portal)
 - Results may be shared with others using collaboration tools
 - Results provenance could be tracked and recorded
 - Results may be stored in ViroLab Laboratory Database
- **Medical User (clinical virologists, healthcare provider)**
 - Interested mainly in meta DSS drug ranking system
 - Uses dedicated web GUI (inside ViroLab Portal)
 - The system may use some ViroLab applications (Binding Affinity Calculator, rule mining) to provide better support over time, but not in automatic way

Generic Use Cases



Requirements

The following requirements have been established:

- Mechanisms for user-friendly experiment creation and execution – an integrated development/execution environment
- Possibility of reusing existing libraries, tools etc.
- Gathering and exposing provenance information
- Integration of geographically-distributed data resources
- Access to WS, WSRF, MOCCA components and jobs
- Secure access to data and applications

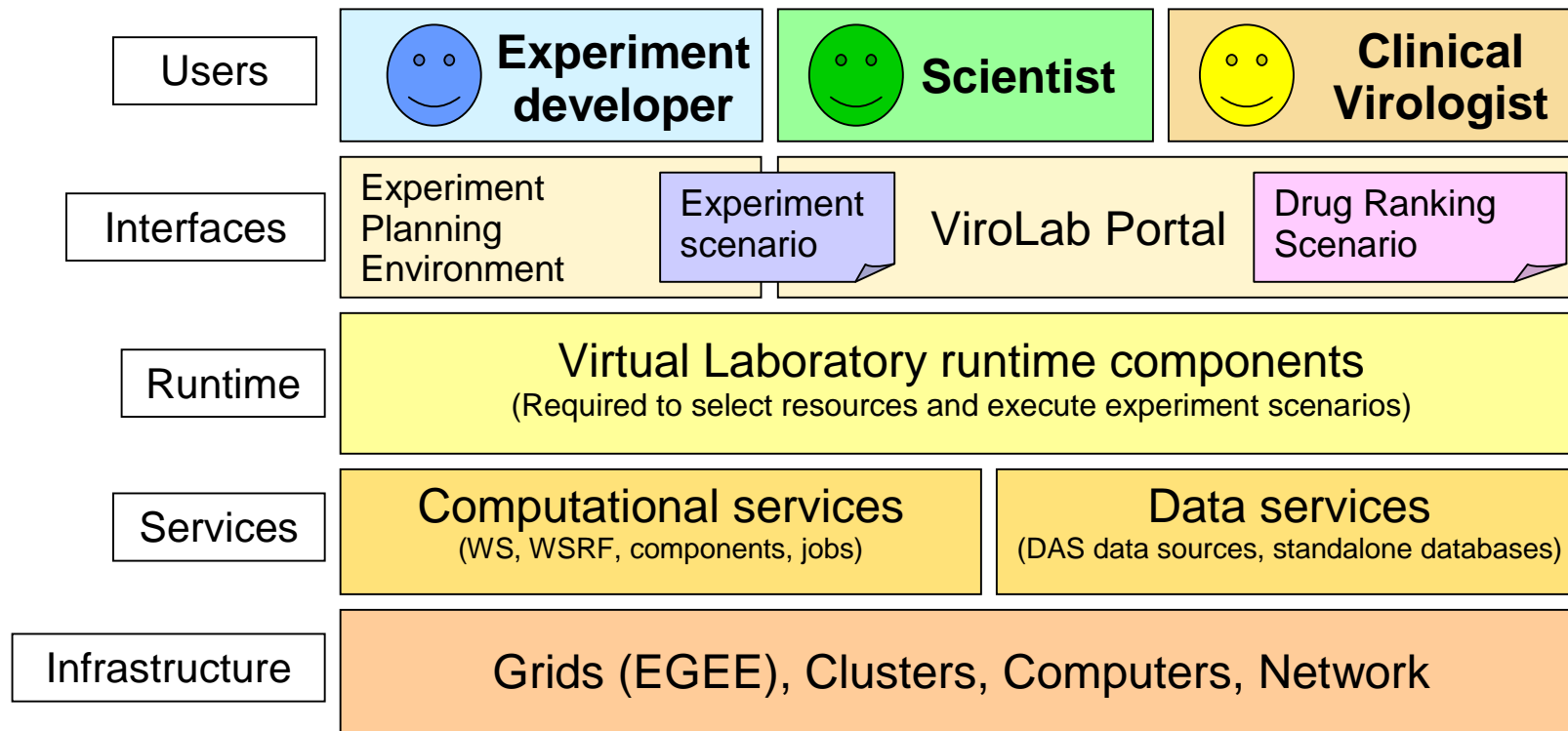
Overview of Workflow Systems and Virtual Laboratories

Virtual Laboratory	App construction	Middleware
Kepler	Drag&Drop	Ptolemy II, Globus Toolkit, WS
Triana	Drag&Drop	Globus Toolkit, GAT
myGrid, Taverna	Drag&Drop	Globus Toolkit, WS
Geodise	Matlab scripts	Computational Toolbox, Jython, WS, Java CoG (1.1), GT2
NESSgrid	webpage with tools	Globus Toolkit, WS
VL-e	Drag&Drop	Globus Toolkit, SoapLab
VL PSNC	JWS apps, batch jobs	Globus Toolkit, GAT

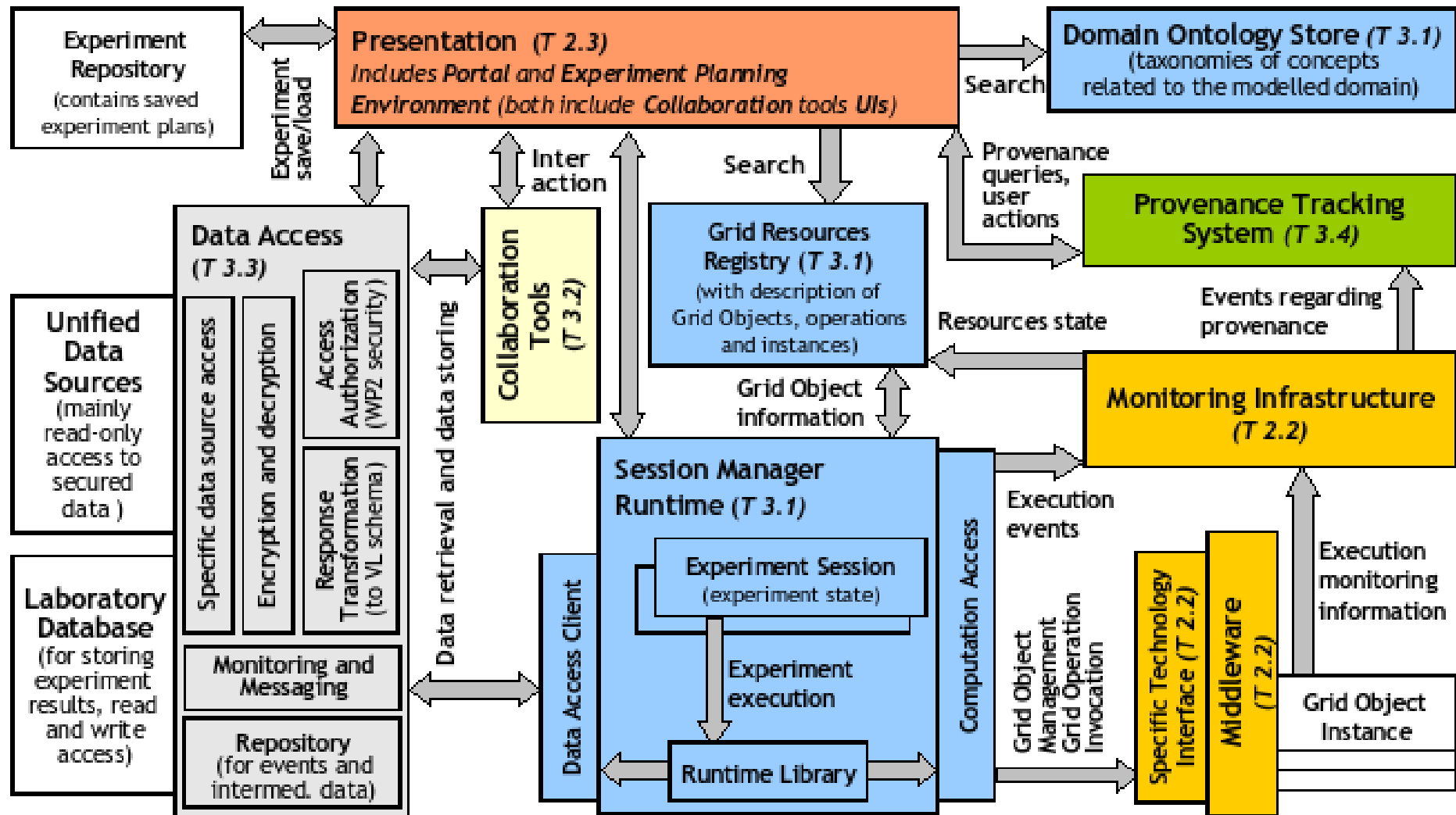
Conclusions:

- There is no solution that fulfills all ViroLab requirements
- Many useful ideas have already been implemented and ViroLab vl reuses them (in semantic modeling, tool registry, provenance tracking etc.)
- Using a scripting language for creating scripts has turned out to be useful (Geodise) and it is the best solution for ViroLab users

Virtual Laboratory layers



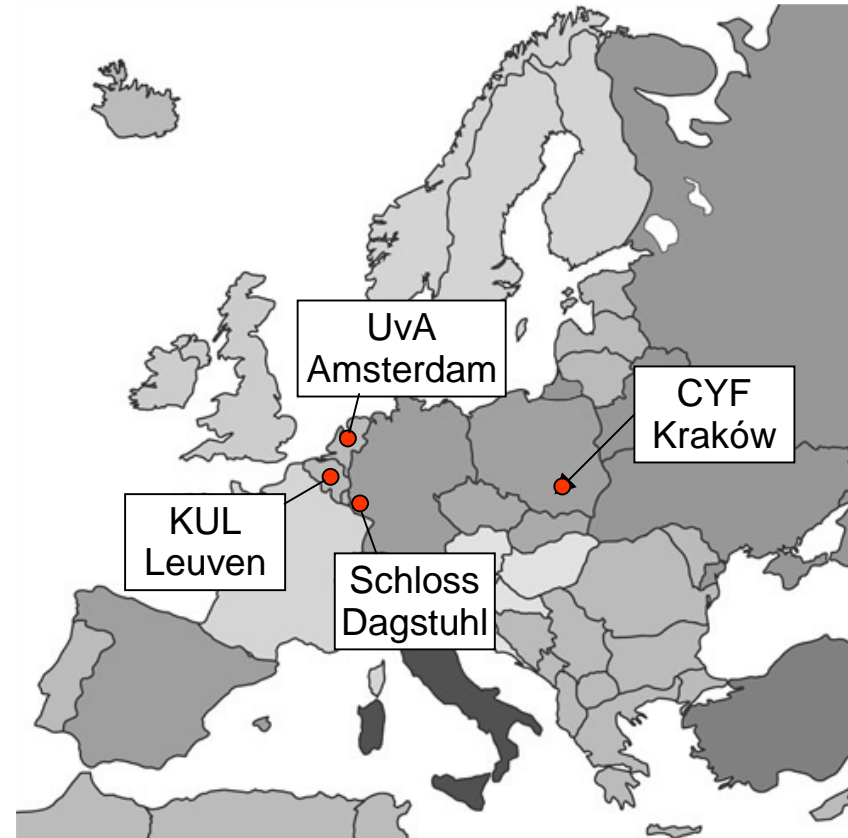
Structure of ViroLab virtual lab



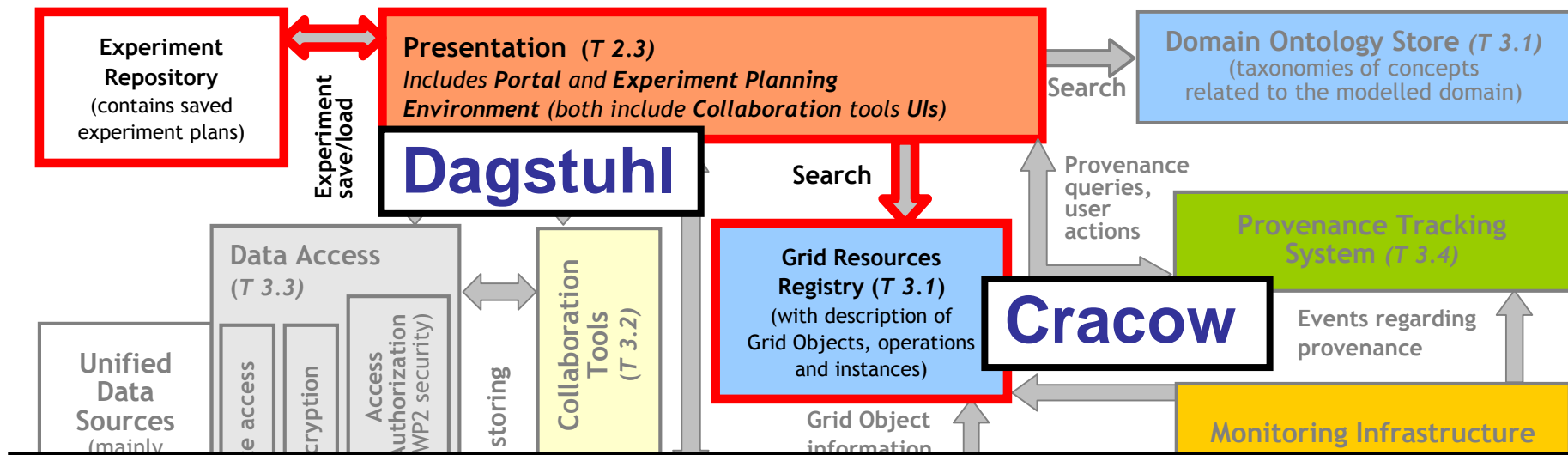
Demo: experiment planning and execution

Experiment steps

- Retrieve nucleotide sequences of virus detected in certain patient
- Align the sequences to a reference strain
- Detect the mutations in the virus
- Detect the virus subtype
- Ask for drug ranking

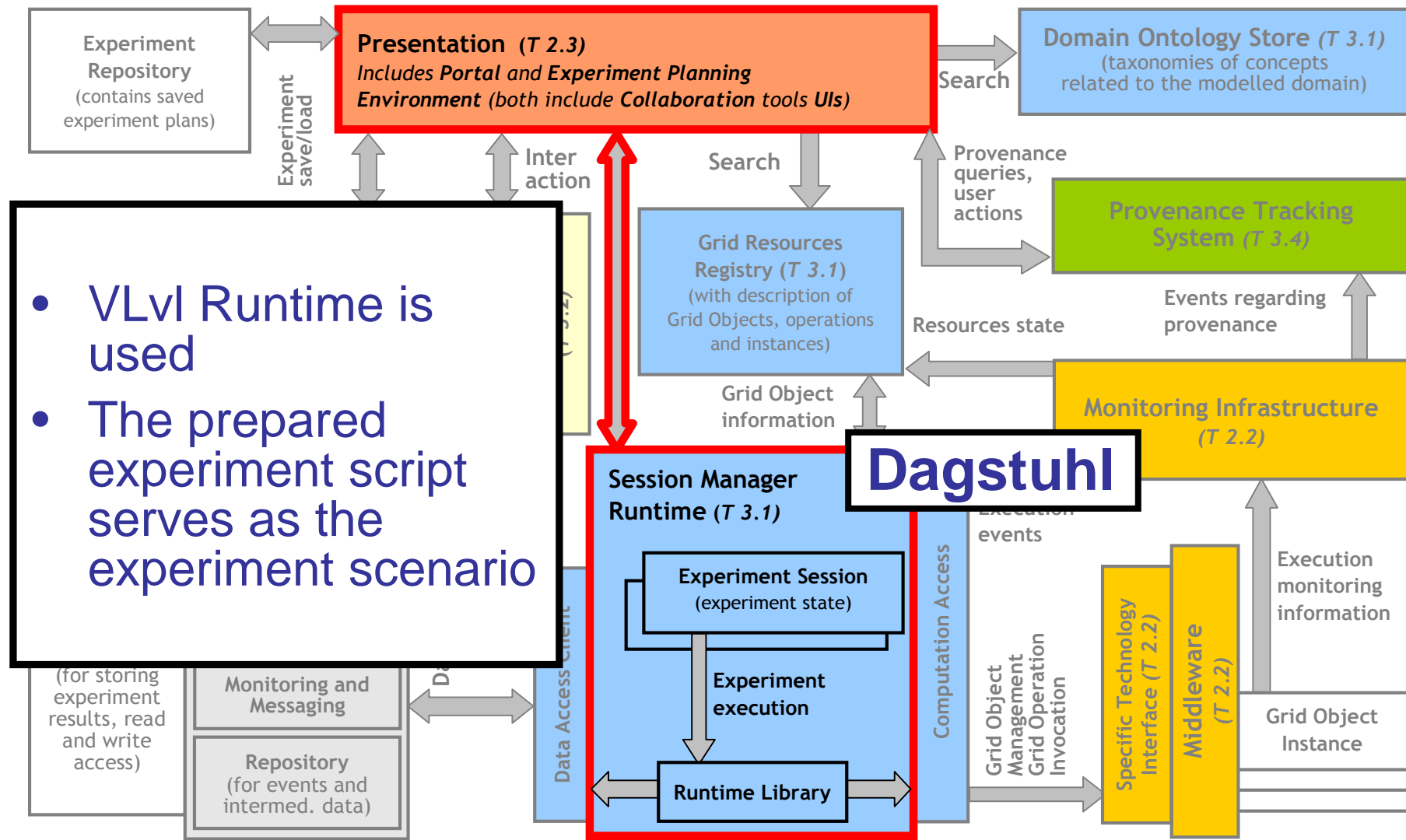


(1) Developer Prepares the Experiment

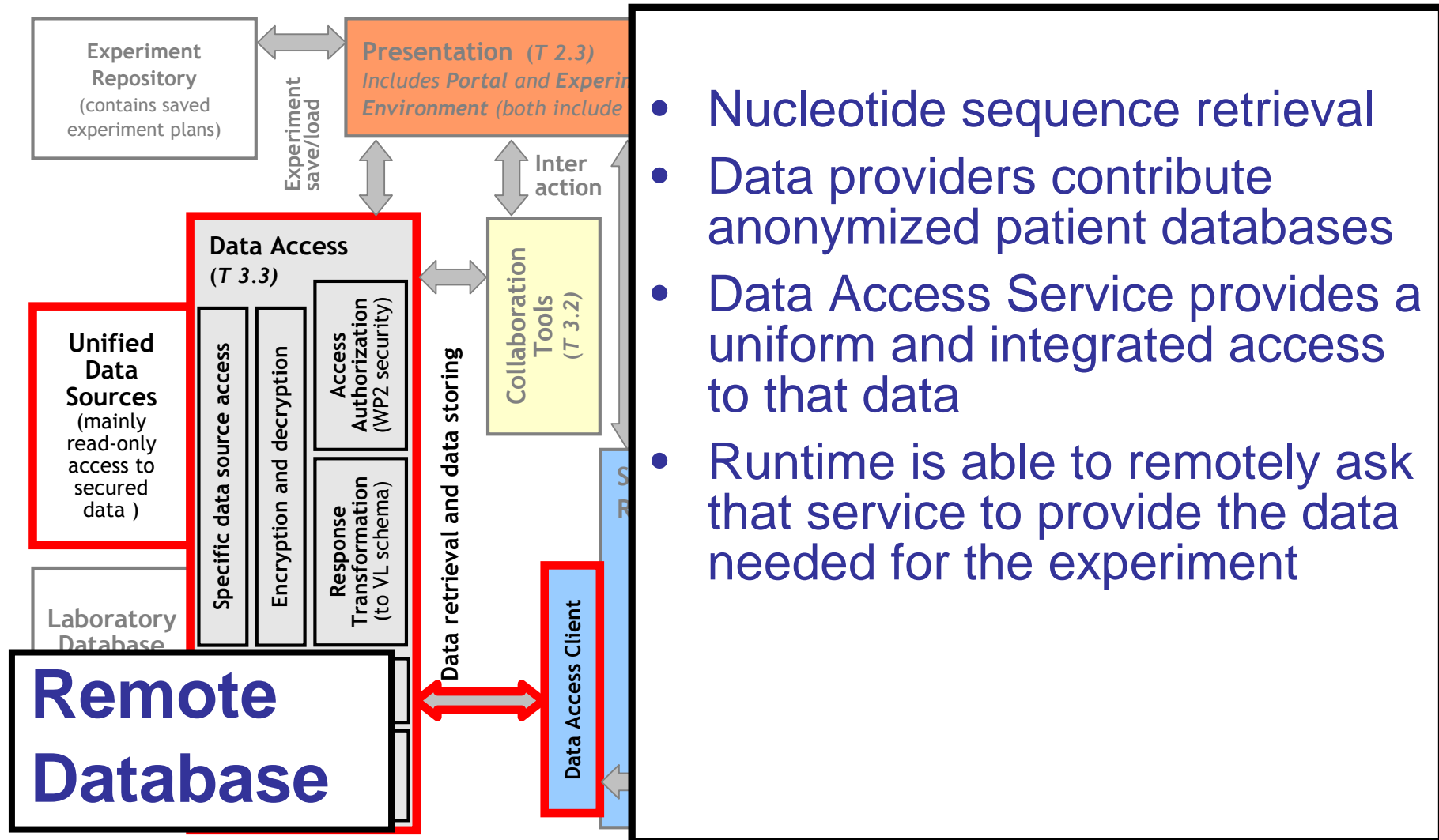


- Developer uses Experiment Planning Environment
- The Registry enables planning the experiment
- The experiment is stored inside the Repository, ready for use by scientists

(2) The Experiment is Executed



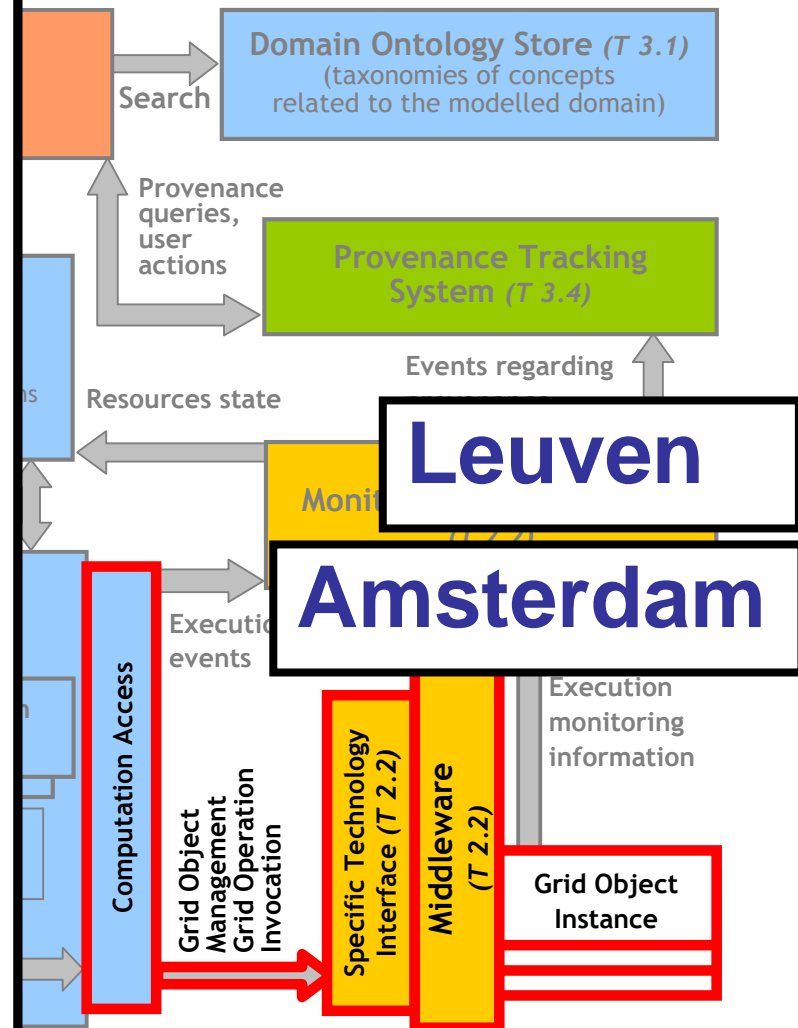
(3) Remote Data Retrieval



- Nucleotide sequence retrieval
- Data providers contribute anonymized patient databases
- Data Access Service provides a uniform and integrated access to that data
- Runtime is able to remotely ask that service to provide the data needed for the experiment

(4) Remote Tool Invocation

- Tools
 - Alignment (KUL)
 - Mutation detection (KUL)
 - Subtype detection (KUL)
 - Drug ranking (UvA)
- ViroLab developers make their tools accessible remotely
- Runtime is able to call the appropriate tools when required by the experiment



Experiment Plan

patientID = 6 region = "rt"

Parameters

```
remoteDB = DACConnector.new(  
    "mysql", "virolab.cyfronet.pl", "test", "testuser", "")  
sequences = remoteDB.executeQuery(  
    "select nucleotides from nt_sequence where  
    patient_id=#{patientID.to_s};")
```

NT retrieval

```
regaDBMutationsTool = GObj.create('regadb.RegadbMutationsTool')  
regaDBMutationsTool.align(sequences, re  
mutations = regaDBMutationsTool.getResult
```

**Alignment + Mutation
detection**

```
regaDBSubtypingTool = GObj.create('regadb.RegadbSubtypingTool')  
regaDBSubtypingTool.subtype(sequences[0])  
puts regaDBSubtypingTool.getResult
```

Subtyping

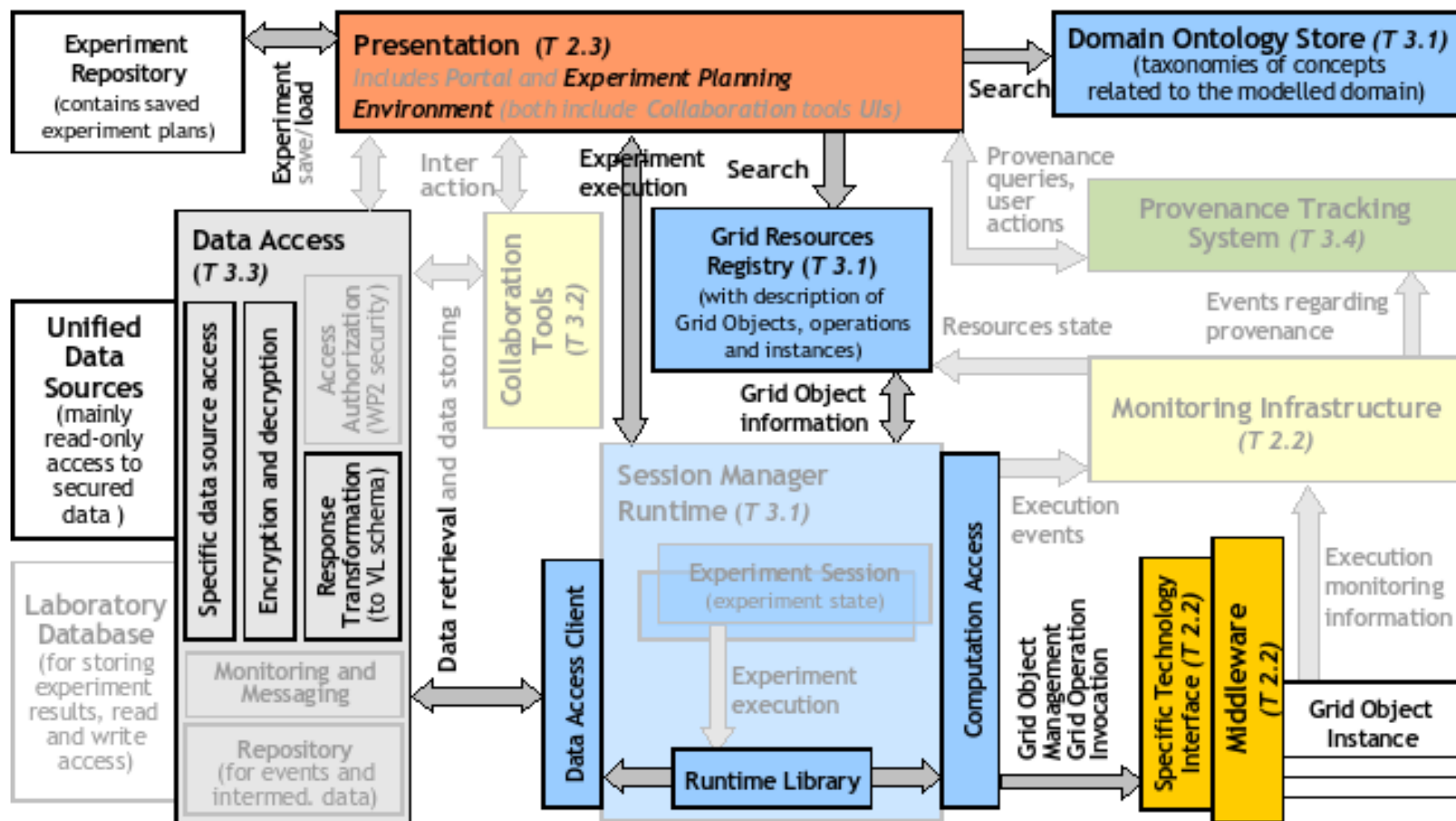
```
puts drs.drs('retrogram', region, 100, mutations)
```

Drug ranking

Demo: experiment planning and execution

- *Switch to the demo*

Components Presented in Demo



PROvenance Tracking System (PROToS)

- **Process-oriented provenance**

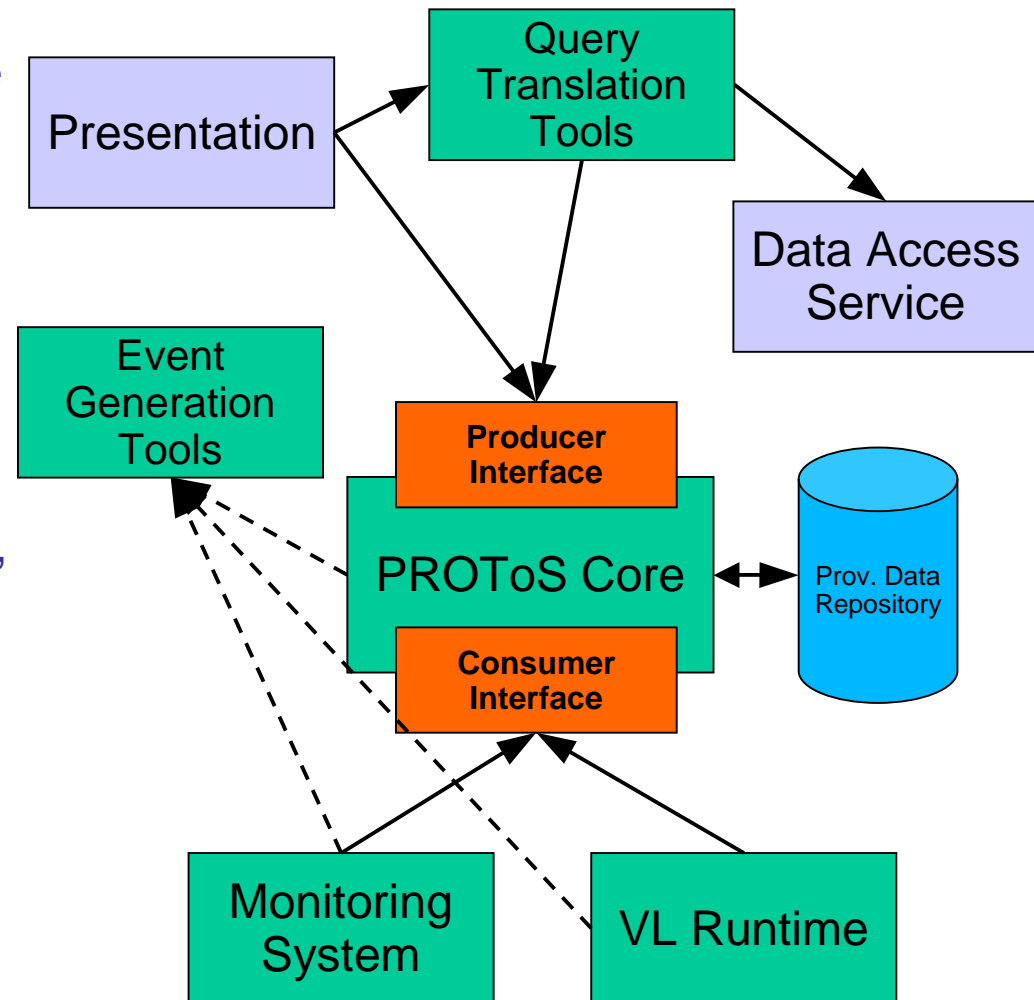
- ‘Experiment provenance’
- Full, information-rich log (trace) of an actual experiment execution
- Based on generic and domain ontologies

- **How provenance be used**

- Advanced experiment handling, such as saving and replay
- Search over experiment repositories, e.g. search for users executing similar experiments

- **Query Translation Tools**

- Scientist-oriented tools to help in construction of complex queries



Available Experiments

- Developing (planning) a new experiment
- From virus genotype to drug resistance
- Querying the historical and provenance information about experiments
- Drug Resistance System based on the Retrogram set of rules
- Data mining and classification with Weka

Scientific Challenges

- Development of collaborative, multidisciplinary applications
 - Abstract layer to hide technological changes
 - Semantic description of applications
 - Integration of provenance recording and tracking
 - A new method of collaborative application development
- Constructing the virtual laboratory
 - Semantic description of resources
 - Deployment on available Grid systems, clusters, and single CEs
 - Integration of technologies: WS, WSRF, components, jobs
 - Secure data access and integration
 - Virtual laboratory as a complex system (formal description, modeling, analysis, ...)
 - Software engineering aspects
 - Exploitation of modern IT technologies

Technologies 1/2

- GridSphere as the user portal
 - Compliant with widely accepted portlet standards
 - Numerous portlets already available
- Eclipse RCP as the developer UI
 - Tool of choice of many software developers (including ViroLab consortium)
 - Industry standard with a large number of available plug-ins, extensions
- JRuby for experiment planning
 - Powerful standard library combined with simple, easy to learn syntax
 - Enables smooth integration with existing Java software (including Grid software)

Technologies 2/2

- Middleware
 - WS (for simple, stateless services with blocking calls)
 - MOCCA (for stateful components and asynchronous calls)
 - Planned in the future: WSRF and Grid job submission support
- OGSA-DAI for data sources integration
 - Well-developed, quite stable and under standardization process by OGF
- OWL, Jena, eXist for ontology storage and processing of semantically-rich provenance data
 - Popular tools for ontology modelling
- GEMINI, Ganglia and JMX for experiment and infrastructure monitoring
 - Standard, generic and extendable solutions

The Team

- WP3 leader: Marian Bubak
- Task 3.1 Runtime
 - Tomasz Gubala, Marek Kasztelnik, Piotr Nowakowski
- Task 3.2 Collaboration Tools
 - Alfredo Tirado, Derek Groen
- Task 3.3 Data Access
 - Stefan Wesner, Matthias Assel, Aenne Loehden, Bettina Krammer
- Task 3.4 Provenance
 - Bartosz Balis, Jakub Wach, Michal Pelczar
- Task 2.2 Middleware
 - Maciej Malawski, Tomasz Bartynski, Eryk Ciepiela, Joanna Kocot, Iwona Ryszka, Katarzyna Rycerz
- Task 2.3 Presentation (EPE)
 - Wlodzimierz Funika, Piotr Pegiel, Dariusz Krol

More about the ViroLab Project

- Coordinator: Prof. Peter M.A. Sloot – UvA
www.virolab.org
- ViroLab Virtual Laboratory description, demos, downloads, ... at
virolab.cyfronet.pl
- ViroLab Session at CGW'07
www.cyfronet.krakow.pl/cgw07