# Towards better experiments on Grid'5000

Lucas Nussbaum
lucas.nussbaum@loria.fr

ALGORILLE team
LORIA / Nancy-Université

# Introduction

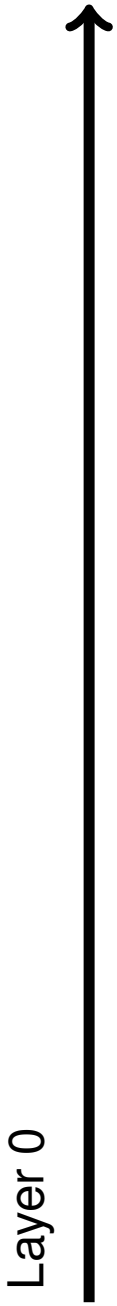
- Standard experiment on Grid'5000 today:
  - Quick and dirty shell scripts
    Best case: quick and dirty Ruby scripts
  - Many manual steps
- We agree that we need to improve the quality of our experiments
  - To push back the limits of what is currently doable
    $\rightarrow$ Complex experiments at a large scale
  - To increase the quality of our experiments
    - Get more confidence in the obtained results
    - Be able to repeat experiments and reproduce results

# Introduction

- Standard experiment on Grid'5000 today:
  - Quick and dirty shell scripts
    Best case: quick and dirty Ruby scripts
  - Many manual steps
- We agree that we need to improve the quality of our experiments
  - To push back the limits of what is currently doable
    $\rightarrow$ Complex experiments at a large scale
  - To increase the quality of our experiments
    - Get more confidence in the obtained results
    - Be able to repeat experiments and reproduce results

**But how can we do that on Grid'5000?**

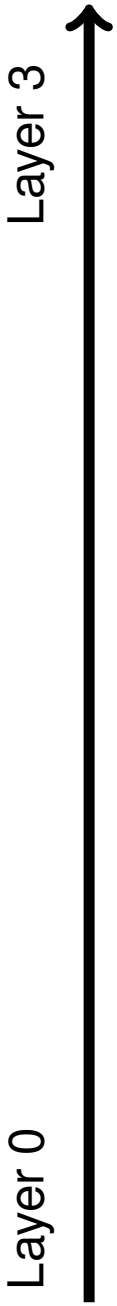
# What's needed?




**Experimental testbed (Grid'5000):** provides reconfigurable hardware and network, isolation, some instrumentation and monitoring

David Margery's talk

# What's needed?



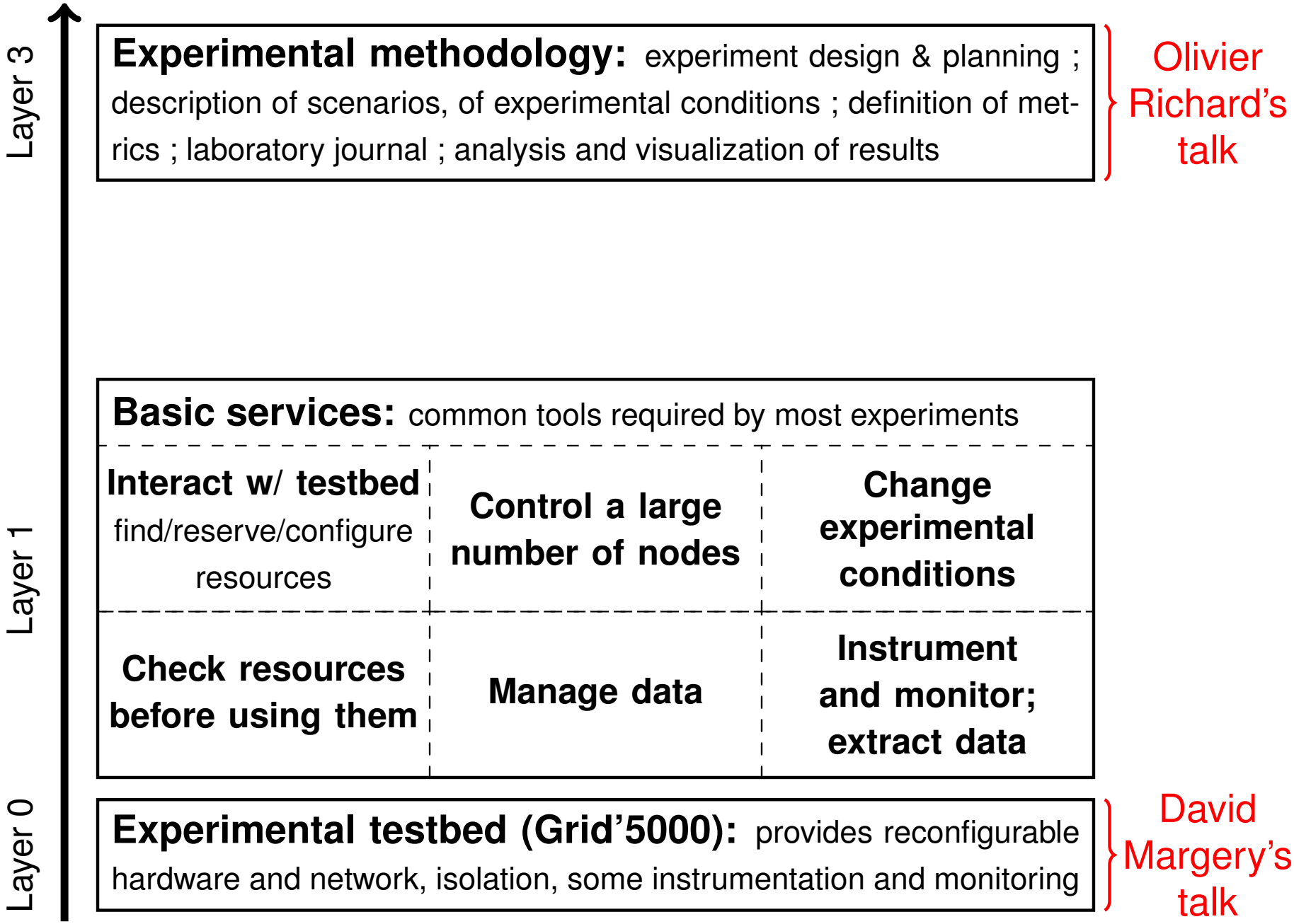

**Experimental methodology:** experiment design & planning ; description of scenarios, of experimental conditions ; definition of metrics ; laboratory journal ; analysis and visualization of results

Olivier Richard's talk

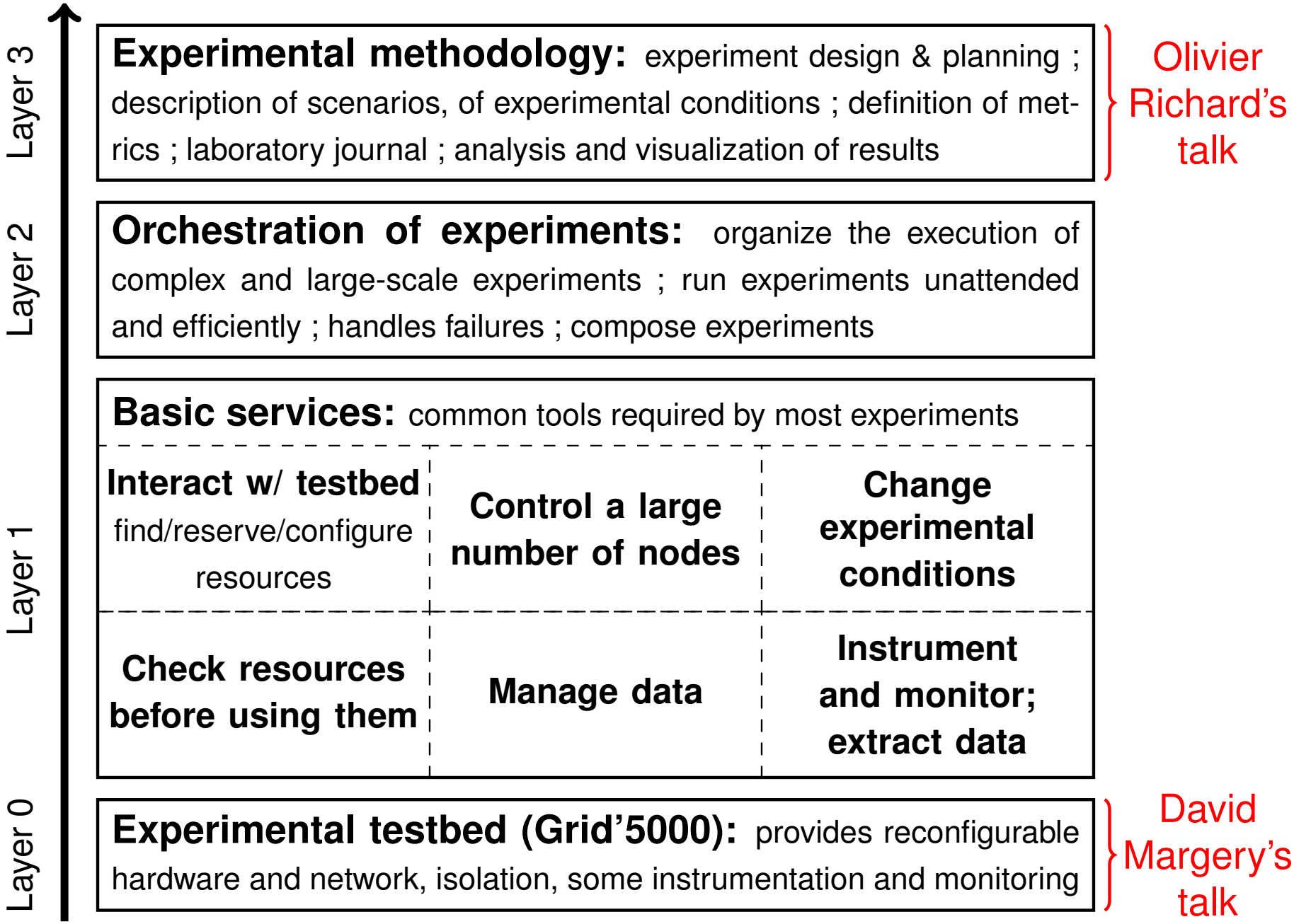
**Experimental testbed (Grid'5000):** provides reconfigurable hardware and network, isolation, some instrumentation and monitoring

David Margery's talk

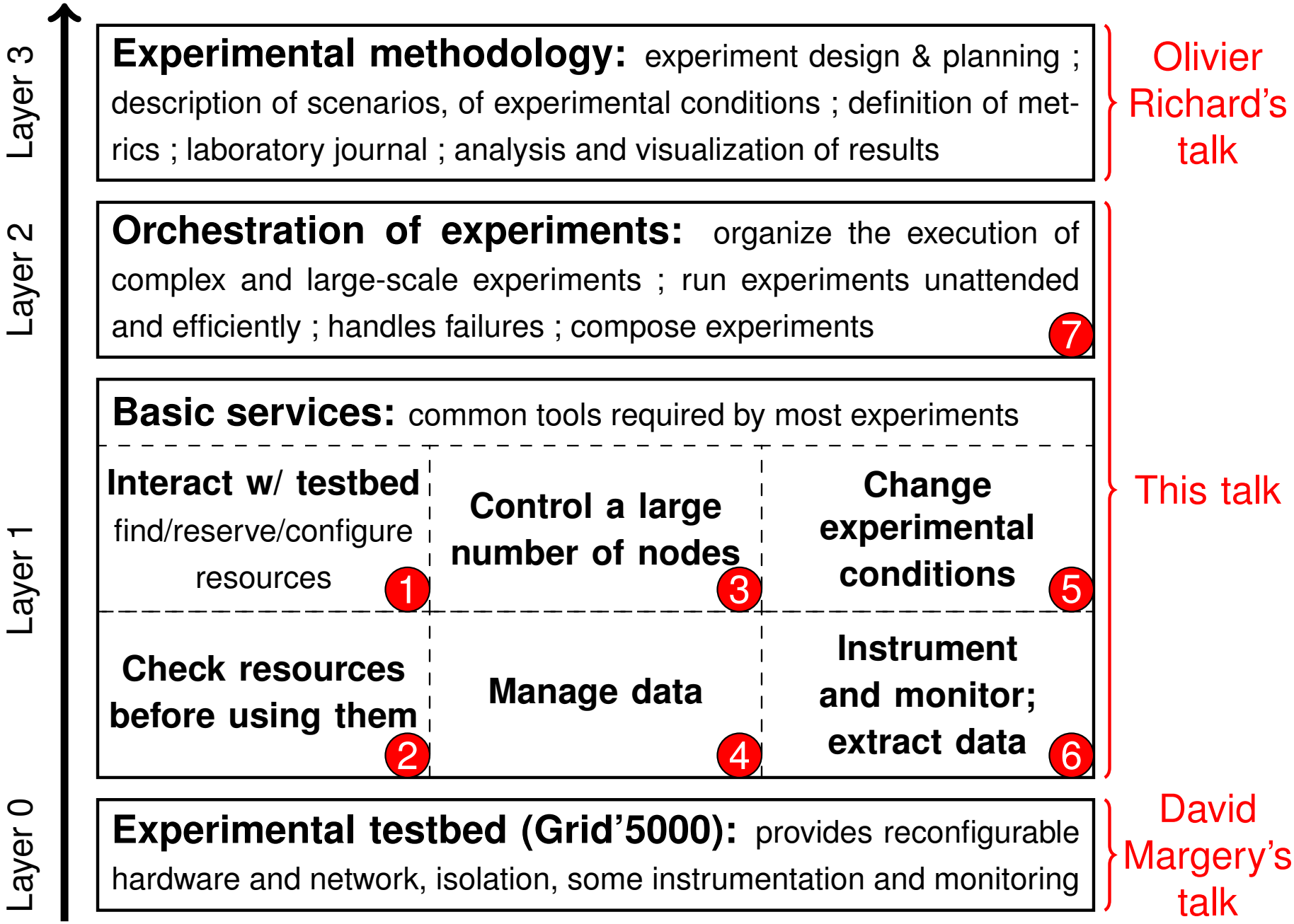
# What's needed?



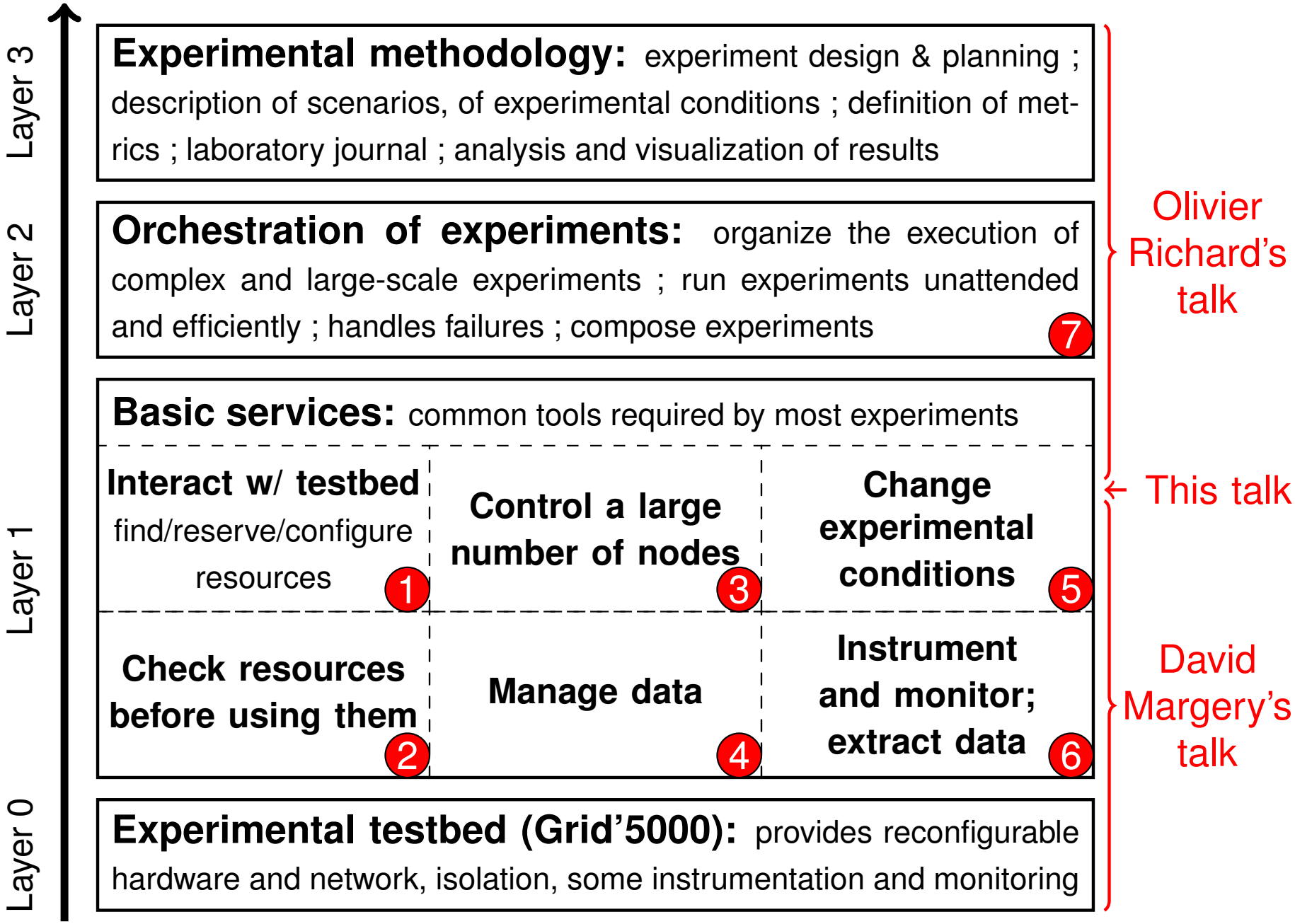
# What's needed?



# What's needed?



# What's needed?



# Interacting with the testbed infrastructure

Past:

- Command-line tools: `oarsub`, `kadeploy3`, ...
- Hard to interact with in an automated way

# Interacting with the testbed infrastructure

Past:

- Command-line tools: `oarsub`, `kadeploy3`, `...`
- Hard to interact with in an automated way

Now:

- Grid'5000 API – `https://api.grid5000.fr/`
  - Reference API: description of the testbed
  - Jobs API: interact with OAR
  - Deployment API: interact with Kadeploy
  - Monitoring API: data about availability of resources
  - Metrology API: interact with Ganglia

**It's (quite[1]) ready, just use it!**

(Already 160 users, including 50 very active ; 2 millions requests in April 2011)

[1] description of resources not uniform across sites, and sometimes incorrect. Doesn't contain everything you might need.

# Checking resources before using them

*Problem:* **What if the resources you got are not perfect?**

- Slow or badly partionned disks
- Network cards with wrong auto-negociation (100 Mb/s vs 1 Gb/s)
- Wrong BIOS settings (e.g hardware prefetching; frequency scaling)

# Checking resources before using them

*Problem:* **What if the resources you got are not perfect?**

- Slow or badly partionned disks
- Network cards with wrong auto-negociation (100 Mb/s vs 1 Gb/s)
- Wrong BIOS settings (e.g hardware prefetching; frequency scaling)

*Current status:*

- Up to the user to detect problems
- In the production environment: **G5K-checks**
  - Will not catch all problems
  - Not usable from deployed environments
- The problem can be mitigated using statistics
  (e.g detect and exclude outliers)

*Possible future:*

- G5K-checks runnable from deployed envs, including user-provided tests

# Control of a large number of nodes

*What most users (still) do:*

```
for n in $(<nodes); do
    ssh $n command
done
```

# Control of a large number of nodes

*What most users (still) do:*

```
for n in $(<nodes); do
    ssh $n command
done
```

*Other (better) solutions:*

- TakTuk + kanif or Taktuk::Pilot (Perl) – good performance ☺, but UI. . . ☹
- Ruby's net-ssh-multi – UI ☺☺, but performance ☹
  - Work in progress: ruby libssh bindings: `https://github.com/leehambley/libssh.rb`
- Clustershell (Python library, CEA) – UI ☺☺, but performance ☹
- pdsh, dsh, dish, clusterssh
- GNU parallel, xargs

# Managing data

*Different use cases:*

- **Broadcast:** push identical data to all nodes
  - Asynchronous or synchronous start
- **Scatter:** push different data to each nodes
- **Gather:** gather data from nodes
- **Share** data between nodes

# Managing data

*Different use cases:*

- **Broadcast:** push identical data to all nodes
  - Asynchronous or synchronous start
- **Scatter:** push different data to each nodes
- **Gather:** gather data from nodes
- **Share** data between nodes

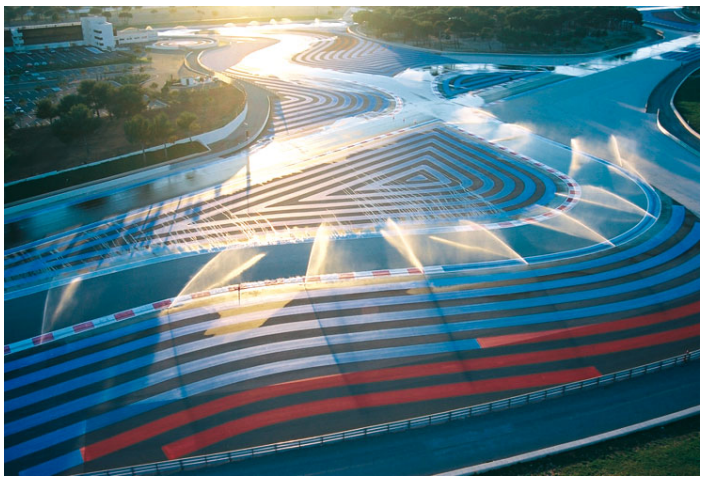
*No silver bullet, but several solutions:*

- standard tools: Rsync, SSHFS, NFS
- Kastafior (chain-based data broadcast)

# Changing experimental conditions

*Sometimes Grid'5000 is too perfect:*

- Need to degrade experimental conditions
  - Emulate different CPU or network performance
    - Some standard tools (Linux TC, etc.)
    - **Modelnet**: used by Davide Frey in Rennes
    - **Wrekavoc**: emulate different CPU speeds & network topology
      *New release expected fall 2011!*
  - Inject load and faults
    - only ad-hoc tools?



# Instrumentation and monitoring

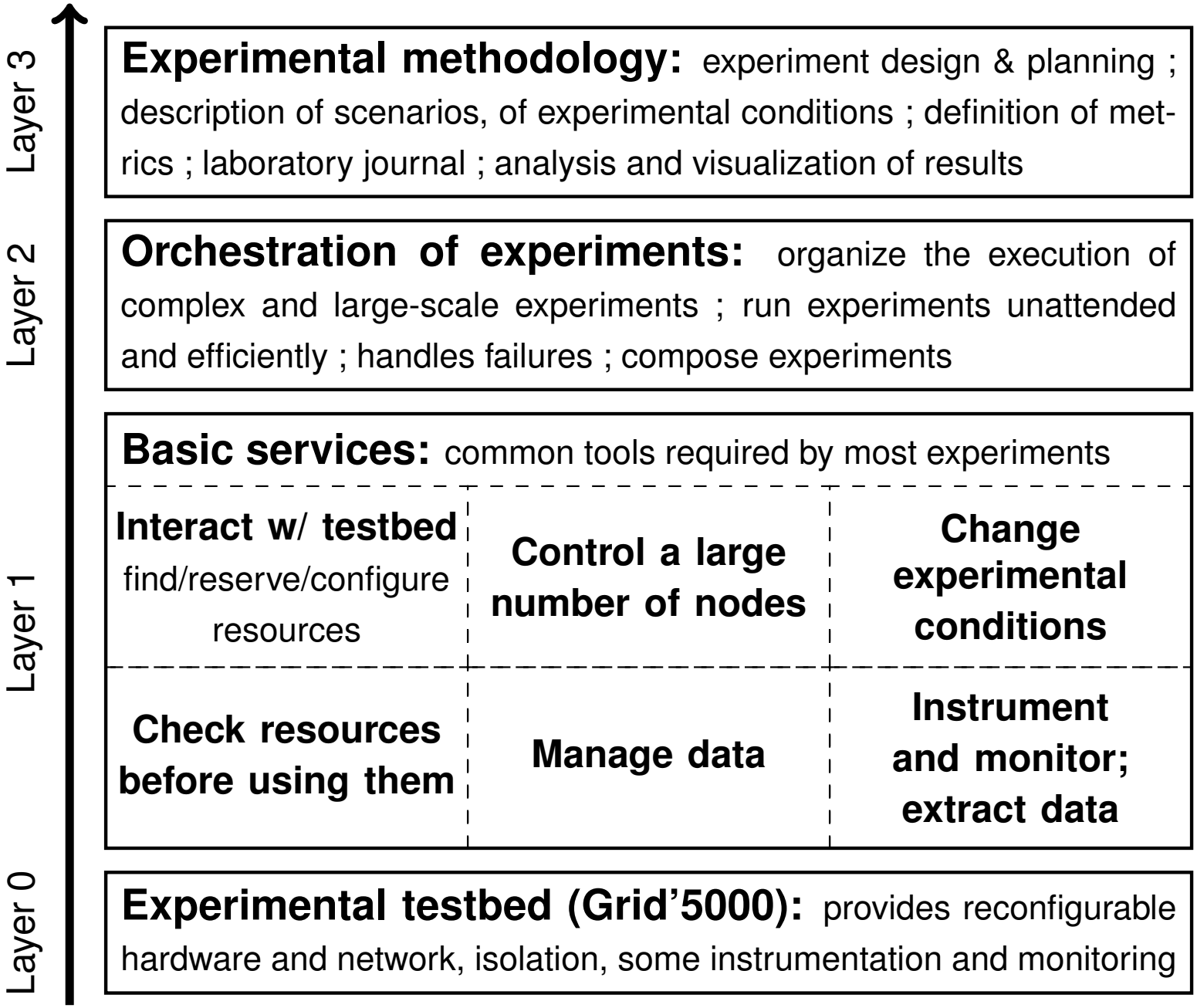
*Tools provided by the testbed:*

- Ganglia, through the monitoring API
  - Can be customized to push your own data
- Metroflux
  - Monitor inter-site traffic with high accuracy
  - Only in Lyon and Lille

*Standard tools:*

- Systemtap, perf, PAPI, etc.
- ...

# What's needed?



# Orchestrating experiments

*Not a new problem:*

- Emulab
- PlanetLab & GENI
- Computational sciences
- Other sciences
- some Grid'5000 attempts

# Emulab

Experiment management integrated in the Emulab framework

- Eric Eide, Leigh Stoller, Tim Stack, Juliana Freire, and Jay Lepreau. Integrated scientific workflow management for the emulab network testbed. USENIX'06
- Eric Eide, Leigh Stoller, and Jay Lepreau. An experimentation workbench for replayable networking research. NSDI'2007

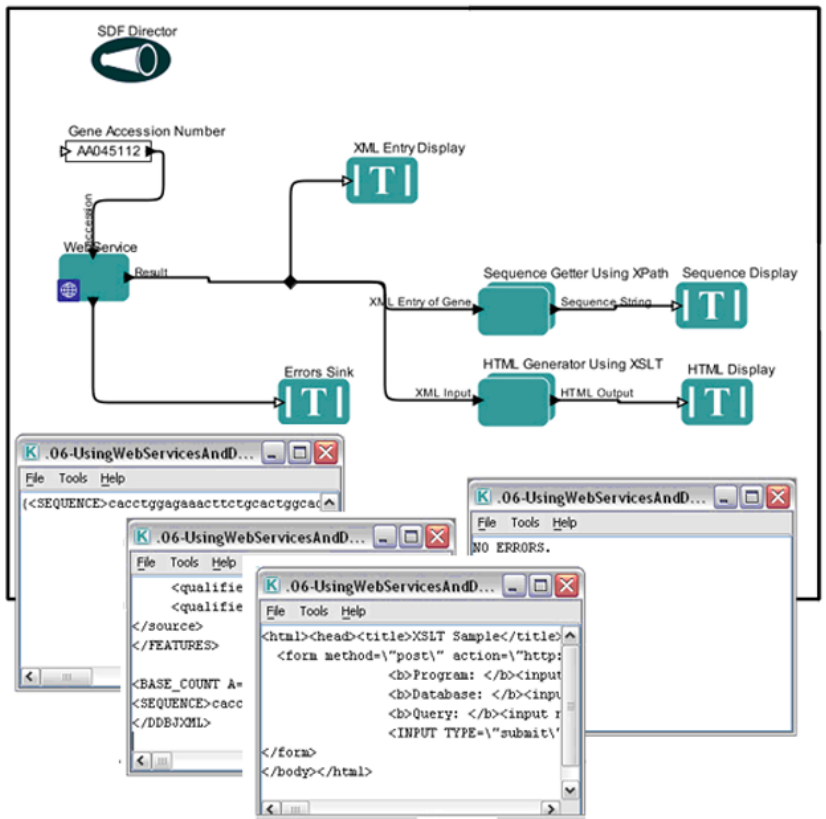
`http://www.cs.utah.edu/flux/workbench/`

# PlanetLab & GENI

- Jeannie Albrecht, Christopher Tuttle, Alex C. Snoeren, and Amin Vahdat. PlanetLab Application Management Using **Plush**. ACM Operating Systems Review (SIGOPS-OSR), 40(1), January 2006
- **Gush**: GENI User Shell – `http://gush.cs.williams.edu/trac/gush` Jeannie Albrecht and Danny Yuxing Huang. Managing Distributed Applications using Gush. TridentCom 2010.
- Inactive GENI WG: *GENI Experiment Workflow and Services*
  - Scope: What do experimenter-users need from GENI? Consider planning, scheduling, running, debugging, analyzing experiments; long running experiments & how they grow; archiving data.

# Computational sciences

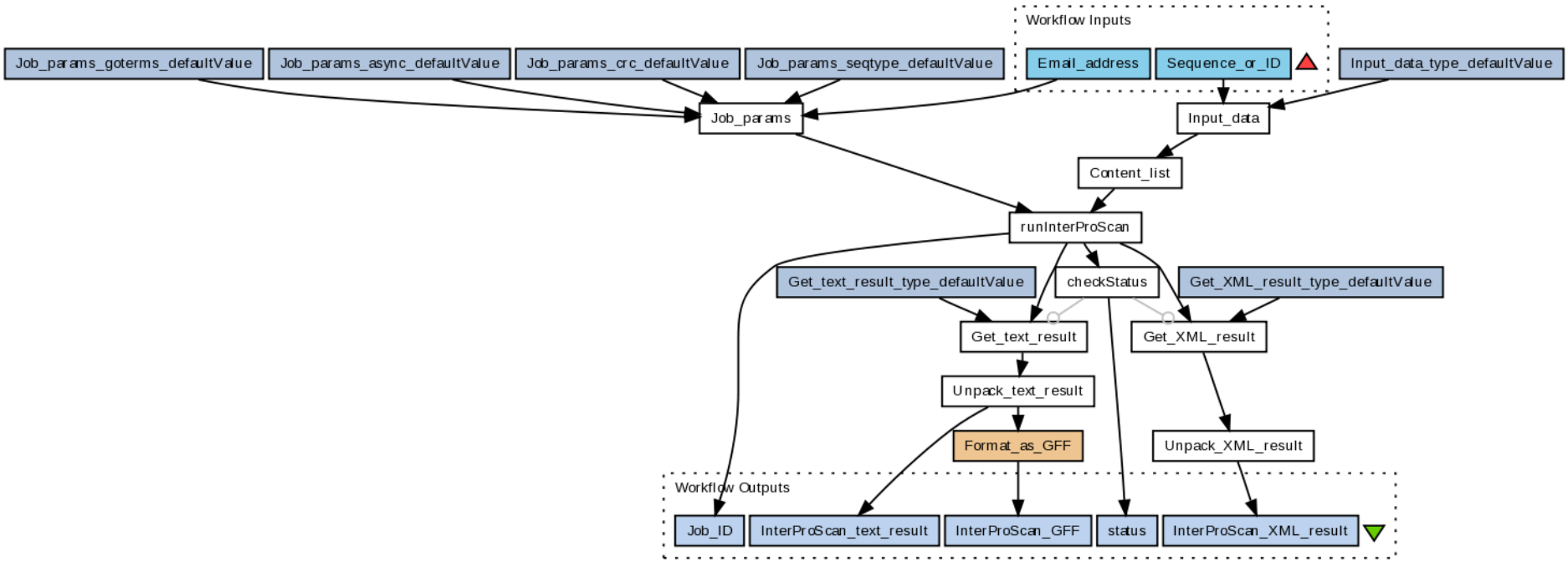
- Many scientific workflow management systems for computational sciences
- Kepler, Taverna, Triana, VisTrails, ...




https://kepler-project.org/

# Computational sciences

- Many scientific workflow management systems for computational sciences
- Kepler, Taverna, Triana, VisTrails, . . .




`http://www.taverna.org.uk/`

Integrated with `http://www.myexperiment.org/`

# Other sciences

NATIONAL INSTRUMENTS

Cart | Help Search

Improve your ni.com experience. Login or Create a user profile.

MyNI | Contact NI | Products & Services | Solutions | Support | NI Developer Zone | Academic | Events | Company

All Products and Services > NI LabVIEW

## The LabVIEW Environment

### The LabVIEW Environment

Engineers and scientists can rapidly and cost-effectively interface with measurement and control hardware, analyze data, share results, and distribute systems through intuitive graphical programming.

» What is LabVIEW?

### How Can I Use LabVIEW?

Applications are as varied as the engineers who create them. Fortunately, LabVIEW combines the flexibility of a programming language with the power of an advanced engineering tool so users can complete their projects regardless of their unique, custom requirements.

Browse all applications

### LabVIEW Product Options

Compare LabVIEW development systems, explore add-ons, and see pricing.

Shop for LabVIEW Products

# On Grid'5000

Several attempts already:

- GRUDU (Lyon – GRAAL)
- NXE (Romaric Guillier, Lyon – RESO)
- Expo (Brice Videau & Olivier Richard, Grenoble)
- Execo (Matthieu Imbert, Lyon)
  `https://gforge.inria.fr/projects/execo/`
- g5k-campaign (Cyril Rohr, Rennes)
  `http://g5k-campaign.gforge.inria.fr/`

But:

- Not clear where we should go
- More attempts are probably needed

# Conclusions

- Grid'5000 is a great experimental platform
- But it requires software to support a better experimental process
- Many tools already exist
- More work is needed