

# Mariotel: A Virtual Remote Computer Science Lab

## Abstract

This paper introduces Mariotel: a free-software project for deploying virtual remote computer science labs that are easily accessible from ordinary web browsers. The Mariotel platform was developed during the first generalized lockdown period due to the Covid-19 pandemic situation. The system has been successfully used at USPN since early 2020.

## Index Terms

Online learning, interactive learning, Remote computer science Lab

## 1 Introduction

This work presents a new web-based virtual remote lab, called Mariotel, which has been developed at USPN as a tool to ensure online distance learning to support online labs during lockdown periods dictated by the pandemic situation due to Covid-19.

Two main features of the platform developed are the following:

- Allowing access to a customized Linux-based virtual machine through simple web browsers.
- providing a simple shared control facility that allows teachers to efficiently help students during the practical lab sessions.

A quick user guide to the platform is described in section

The general architecture of the platform is summarized in section III. Finally, a short usage report is given in section IV.

## 2 Mariotel: Usage Guide


Two main users' roles are defined in the Mariotel application: the *teacher* role and *student* one. The latter role is automatically assigned to users who receive a *student access link*. The teacher role allows working sessions to be

Bonjour **kanawati**, bienvenue sur la page de  
réservation d'une salle Mariotel<sup>i</sup>

### Votre réservation



**Date et heure**

2021-09-24 16:39





**Durée**

3



**Nombre d'étudiants**

25



**Email de notification du lien**

kanawati@univ-paris13.fr

Réserver

Déconnexion

Inscrire

Profil

Planning

Fig. 1. Mariotel: Booking interface

booked and, hence, student access links to be generated. Three levels of teacher role are defined:

- Level 0: this allows limited working sessions to be booked. Limitations cover a limited session duration (up to 5 hours in the current implementation) as well a limit on the number of students to be invited (up to 32 students).
- Level 1: in addition to the privileges of level 0, level-1 teachers can also register new teachers (of level 0 or 1). This level also allows long working sessions to be booked (up to 10 hours per session)
- Level 2: this level corresponds to the *super use* of the platform, as it allows involvement in sessions booked by other teachers (of any level). This level also allows large-scale working sessions to be booked, which can also be used for performance testing (up to 320 students per session).

Figure 1 shows the booking interface that allows the main session parameters to be specified: session date, number of students, and the length of the session.

The booking interface also allows different services to be accessed, such as changing the current user profile (updating e-mail addresses, and/or the user password), accessing the booking schedule of the platform, and registering new users (if the current user has teacher level 1 or 2).

Once a teacher has booked a session, reservation information is displayed, including two types of platform access links (see Figure 2).

## Réservation de salles Mariotel<sup>3</sup>

### Détails de la réservation

ioddo-202010090900-NhfiC8cRK3HoWCLw

#### Réservation

Responsable : **ioddo**, date : 2020-10-09 09:00:00, durée : 4 heures, nombre d'étudiants : 30

#### Lien d'accès étudiant

<https://tel.marionnet.org/session.php?link=ioddo-202010090900-NhfiC8cRK3HoWCLw>

Mot de passe VNC/Unix : **WCLw** (4 derniers caractères du lien)

Note : après la saisie de votre NOM-DE-FAMILLE, vous serez automatiquement connectés au système GNU/Linux

#### Lien d'accès enseignant

<https://tel.marionnet.org/session.php?teacher=5d91c62d61&link=ioddo-202010090900-NhfiC8cRK3HoWCLw>

Mot de passe VNC/Unix : **HoWCLw** (6 derniers caractères du lien)

Note : vous serez l'utilisateur Unix **teacher** seulement sur votre poste, vous partagerez l'écran et l'identité Unix **student** sur tous les postes étudiants

### Envoyer un email

mailto: students

mailto: teacher

Déconnexion

Inscrire

Profil

Réservation

Planning

Fig. 2. Booking confirmation window with both student and teacher access links

- **Student access link:** this is the link to send to students in order for them to access the platform. This link allows one free virtual machine to be accessed in the context of the booked session. The invited student has to give their name (as a label for the accessed virtual machine) and should provide a predefined password composed of the last four characters of the access URL. In the first implementation of the platform, all students have the same password to access virtual machines activated in the working sessions. Naturally, once a virtual machine is accessed by

Accès à la salle virtuelle Mariotel

kanawati-202107271555-zJgVioRaK5XSr9O  
Réservée par kanawati, 24 postes, en cours  
Pour la semaine 30, mardi 27-07-2021 de 15:55 à 18:55

Affectation des stations de travail (2/24)


























Étudiant	Station	Étudiant	Station	Étudiant	Station	Étudiant	Station	Étudiant	Station
1 Alain		2		3 Demariot		4		5	
6		7		8		9		10	
11		12		13		14		15	
16		17		18		19		20	
21		22		23		24		25	

Fig. 3. Mariotel: Teacher’s view of the online lab

one student, it cannot be accessed by another. This simple method of authentication relies heavily on a *social protocol* which assumes that students will not forward the student access link to unauthorized persons. While this approach may seem to be a light authentication approach, it has its roots in the simplicity design rule adopted in this project and the fact that the virtual machines accessed contain no personal information and they have a limited life span.

- *Teacher access link*: this link allows the teacher to access the whole booked session. Figure 3 shows the teacher’s view of the ongoing working sessions. The teacher can see all the virtual machines included in the session. The name of the student using a machine is displayed next to the machine icon. The teacher can access students’ machines with a simple click on the icon. The teacher then shares the control of the machine with the student and can help the student to correct or accomplish the required task.

A simple planning display module is provided, allowing *teachers* to display all scheduled sessions booked by themselves or other teachers (see Figure 4). Naturally, a teacher can edit their own booked sessions: change their different parameters, or even cancel a session.

In order to run a remote practical lab, the teacher has to book a session as well as a video-conferencing session using their favorite conferencing solutions (e.g. Microsoft Teams, Zoom, Discord, etc.). The conferencing tools serve to establish a real-time interaction channel between the students and the teacher. Any participant can share the display of their own

### Planning des salles Mariotel<sup>1</sup>

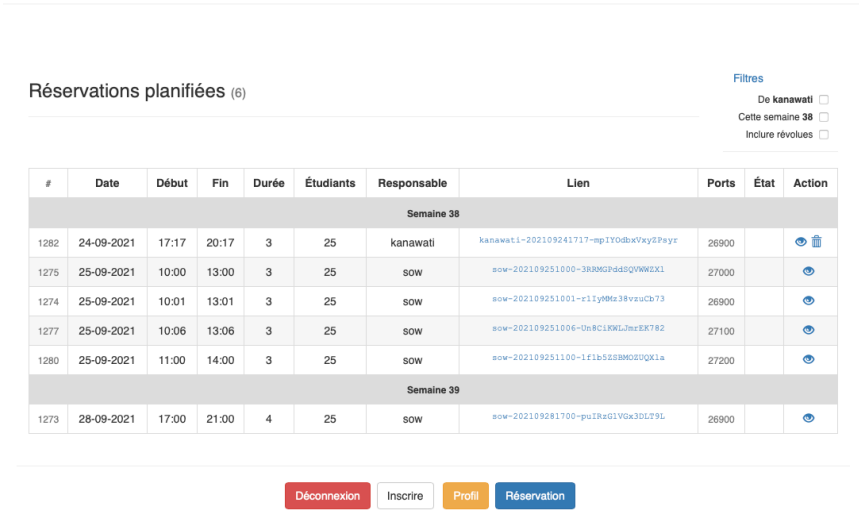


Fig. 4. Teacher's view of Mariotel planning

(virtual) machine to others. However, only the teacher can access any machine ("workstation" hereafter) included in the session. Observations show that students prefer using classical chatting tools provided by conferencing systems to interact with one another and with the teacher.

### 3 Mariotel: Architecture

The internal behavior of the Mariotel system is based on two fundamental software components, Docker [1] and noVNC [2], which are both, in turn, based on the GNU/Linux platform. In addition, the assembly of the Docker and noVNC technologies into a virtual GNU/Linux distribution system (precisely, the popular Ubuntu 18.04) was realized from a third relevant software component provided by the Accetto project [3].

- 1) *Containers*: Docker allows a group of processes, named “containers”, to run within a single Linux instance, sharing the same host kernel but running local processes in full isolation (local process trees, local network, local user IDs, local mounted file systems, local resource limiting for memory and CPU, etc.). Avoiding kernel replications, Docker containers are extremely *lightweight*, in such a way that a single physical server (or even a virtual machine) can run several containers simultaneously.
- 2) *noVNC*: Our Docker containers are structured, except for the kernel, like an ordinary GNU/Linux distribution system, both for the file system and the initial part of the system process tree. This way the user, i.e. the student, will have the impression of using a real workstation with a graphical desktop environment. The noVNC component is required at this point: we want a comfortable GNU/Linux distribution system, like the popular Ubuntu 18.04, where the standard X11 graphical server would be replaced by noVNC. Actually, by means of a companion tool on the server side (websockify.py) and a second companion tool on the client side (a JavaScript application), this special X11 server will be able to display the graphical elements and control X11 events in an HTML canvas in a tab of any web browser.
- 3) *Accetto*: The third free software component we used is the branch named *ubuntu-vnc-xfce* from the Accetto project. This branch provides a working version of a setup program, called “dockerfile”, able to build an Ubuntu 18.04 image integrating noVNC into a Docker container. Even if the adaptation work

has been substantial, this initial dockerfile has proved to be an essential working base to get an image suitable for our USPN students in a very reasonable time. One of the most important modifications made concerns the installation of the Marionnet software [4], allowing the simulation of TCP/IP networks, thus allowing labs for several courses about computer networks learned at USPN. It is interesting to note the impressive number of nested virtualization layers this implies, and it is satisfying and quite fun to watch as it runs smoothly, demonstrating the robustness of each

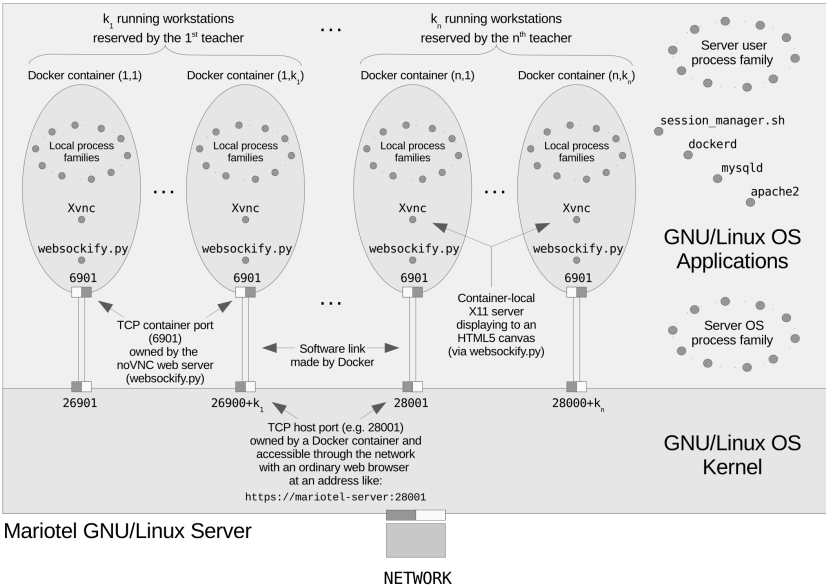


Fig. 5. Mariotel: Server architecture

layer. Actually, the server used is a QEMU [5] virtual machine (equipped with a GNU/Linux Debian 10), supporting Docker containers that run, in their turn, the UML (User Mode Linux [6]) virtual machines launched by Marionnet.

The Mariotel project integrates these first three components, adapting them with two new software components: the booking manager and the session execution manager. Mariotel code is released under a GNU-GPL license, while all other components used are redistributed according to the original free license. The whole project is available from the LIPN git repository.<sup>1</sup>

- 4) *Booking manager:* A web interface allows teachers registered on the platform to book and manage lab sessions. The website has been developed with the PHP framework, using the Bootstrap software library [7]. When a classroom is reserved, the record defining the scheduled session is stored in a MariaDB database [8]. The two HTML links generated for

<sup>1</sup> <https://depot.lipn.univ-paris13.fr/marionnet/mariotel/>

the students and teacher are displayed on a page that summarizes the booking (see figure). Through the same page, these links can be transmitted by email to the group of students or to the teacher themselves. At running time, when the session is active and every student is assigned to their workstation, the teacher will access a web page representing the room (see Figure 3) via the link reserved for them and is so able to access all the stations and help the students. There are two important technical points to this part. The first point concerns the choice of a unique range of TCP ports for the set of workstations in a given session, avoiding all possible intersections with other interlacing sessions, even if only for a few minutes. The second point concerns the assignment of stations to students in a quasi-alphabetical order, so that the teacher can easily find the place occupied by a student who needs help.

- 5) *Session manager*: The goal of this component is to manage the lab sessions, i.e. to launch all the workstations booked for the students and to stop them. These two kinds of actions should happen on the two dates of the beginning and end of the project, as foreseen by the teacher during the reservation. To perform this task, the session manager, scheduled as a Unix *cron* task, is launched regularly, every two minutes, on the server to query the booking database and then to activate or stop the sessions involved. Figure 5 gives an outline of the running processes and their dependencies, assuming that  $n$  teachers have each booked a class and are using  $k_1, \dots, k_n$  workstations for their students respectively.

A secondary task of the manager is to protect the active sessions from possible unauthorized users, whom we call “disturbers”. A disturber is someone, identified by their IP address, that has attempted to connect to at least two distinct workstations without being recognized as a possible teacher (because they are not in the database of IP addresses associated with a teacher). Actually, the only ones naturally allowed to connect to several machines in the same session are teachers. So, others are disturbers. When a disturber is detected, an *iptables* command banning the access of the disturber is sent and executed on all workstations of the session in question.

The session manager has been developed with the Bash language, which is sufficient, even very efficient, for this kind of task, which consists, in a caricatural way, only of automating the launching of complex Docker commands, so complex that it would be difficult to launch them manually. With algorithmic problems having been confined to the booking manager, and thus easily solved in PHP, the Bash language is perfectly adapted to this simple mission of building and launching complex Unix shell commands.

## 4 Mariotel: Usage Report

The Mariotel platform was first implemented on a virtual server with 32 cores and 96 GB of Ram memory. The server was put into effect in order to host different programming, system administration, as well as computer network labs scheduled during the first lockdown period due to the Covid-19 pandemic situation in early 2020.

The simple user interface proved to be very effective since both teachers and students are able to use the system without any training. The fact that users are free to use their favorite video conferencing tool in parallel to the Mariotel-based remote practical session was also an important factor in the early adoption of the system. The computation power of the deployed server was enough to support up to 150 simultaneous student connections involved in ten different lab sessions (not all using Marionnet networking virtual lab applications). However, the server proved to be very limited in running labs using greedy CPU-power applications such as GNS3.<sup>2</sup>

On the negative side, the social protocol underlying the platform access scheme was rapidly revealed to be very limited. Some teachers sent students the teacher access link instead of the student access one, allowing students to access any machine in the session in question. This problem can be quickly solved since teachers can cancel a scheduled session and reschedule a new one, which can be ready to run in less than 3 minutes.

More seriously, a few sessions were targeted by some unauthorized disturbers, as described in section III-5. In order to limit these perturbations, simple patches were applied to the system's access module. Access from devices assigned to foreign IP addresses was banned. In addition, a student's IP address is allowed to be connected only once to a given session. These rapidly deployed patches drastically reduced the sessions' perturbations.

Except for these very few perturbations (concerning less than 1% of sessions), the platform was successfully used during the first lockdown period by 32 distinct teachers operating more than 4500 student connections whose working duration exceeded one hour. The platform is still being used for in-person computer science labs since students widely appreciate the shared control features the platform offers to teachers during students' sessions. Another kind of use of the platform consists in scheduling sessions where students can work in an autonomous way in order to review the principles and topics they have studied.

---

2 <https://www.gns3.com/>

## 5 Conclusions

The generalized Covid-rooted lockdown periods have increased the awareness of academic authorities of the absolute necessity of deploying virtual remote labs in higher education institutes. The Mariotel platform we have presented in this paper provides a simple but powerful solution for rapidly deploying GNU/Linux-based computer science VRL. The main advantage of the platform is its ability to export the displays of virtual machines running on a central server in the students' browsers, allowing students to have access to low computation power as well as a low bandwidth network connection to join a practical lab efficiently. The shared control between the teacher, on the one hand, and each student, on the other, is another main feature of this platform. Mariotel was deployed during the first lockdown period in France (Spring 2020) and was extensively used with very few incidents despite the use of a simple social access protocol.

Short-term extensions of the current work include implementing a robust access and authentication process and implementing a persistent virtual machine allowing each student to have her/his own virtual machine that can be accessed at any time and on which they can save both data and modified configurations. This is particularly interesting for system administration-oriented labs.

## Acknowledgment

We would like to thank Xavier Monnin, the IT System administrator at LIPN Computer science laboratory CNRS UMR 7030, for his valuable support in providing the first infrastructure for hosting the Mariotel platform. Thanks goes also to LIPN for allowing us to use the lab IT infrastructure to host the first operational Mariotel version and git repository. We would also like to thank the Technical University Institute of Villeteuse (IUTV) for its financial support.

## References

- D. Merkel, "Docker: lightweight Linux containers for consistent development and deployment," *Linux journal*, vol. 2014, no. 239, p. 2, 2014.
- noVNC, 2010-2021. [Online]. Available: <https://github.com/novnc>
- Accetto, 2017-2021. [Online]. Available: <https://github.com/accetto/ubuntu-vnc-xfce>

Marionnet, 2007-2021. [Online]. Available: <http://www.marionnet.org/>  
QEMU, 2007-2021. [Online]. Available: <https://www.qemu.org/>  
UML, 2004-2021. [Online]. Available: <http://user-mode-linux.sourceforge.net/>  
Bootstrap, 2011-2021. [Online]. Available: <https://getbootstrap.com/>  
MariaDB, 2009-2021. [Online]. Available: <https://mariadb.org/>

## Authors



Jean-Vincent Loddo, PhD  
University Sorbonne Paris Nord  
LIPN, CNRS UMR 7030  
99 Av. J-B. Clément  
93430 Villetaneuse, France  
<https://lipn.univ-paris13.fr/~loddo/>  
[loddo@univ-paris13.fr](mailto:loddo@univ-paris13.fr)



Rushed Kanawati, PhD  
University Sorbonne Paris Nord  
LIPN, CNRS UMR 7030  
99 Av. J-B. Clément  
93430 Villetaneuse, France  
<https://orcid.org/0000-0002-0149-6197>  
[kanawati@univ-paris13.fr](mailto:kanawati@univ-paris13.fr)