No. 99

UNIVERSITY OF COLOGNE

WORKING PAPER SERIES IN ECONOMICS

Z-TREE UNLEASHED: A NOVEL CLIENT-INTEGRATING ARCHITECTURE FOR CONDUCTING Z-TREE EXPERIMENTS OVER THE INTERNET

MATTHIAS L. DUCH
MAX R. P. GROSSMANN
THOMAS LAUER

Department of Economics
University of Cologne
Albertus-Magnus-Platz
D-50923 Köln
Germany

http://www.wiso.uni-koeln.de

z-Tree unleashed: A novel client-integrating architecture for conducting z-Tree experiments over the Internet

Matthias L. Duch
University of Cologne

Max R. P. Grossmann*
University of Cologne

Thomas Lauer
University of Cologne

March 30, 2020 (version 2)

Abstract

We present *z-Tree unleashed*, a novel approach and set of scripts to aid the implementation of experimental economics laboratory experiments outside of the laboratory. *z-Tree unleashed* enables subjects to join the experiment using a web portal that requires no additional software. Experimenters are likewise enabled to administer their experiments from anywhere in the world. Except for z-Tree itself, *z-Tree unleashed* is entirely based on free and open source software. In this paper, we firstly outline the motivation. Secondly, we give a high-level overview of *z-Tree unleashed*'s advantages and its design. We then proceed to show how to set up the server and demonstrate the steps required for conducting an entire experiment. We subsequently explain how to leverage the security and routing features of a virtual private network with *z-Tree unleashed*, enabling servers to securely run behind routers.

1 Introduction

Since its inception in 1998, many researchers have come to rely on the Zurich Toolbox for Readymade Economic Experiments (z-Tree), cf. Fischbacher

*Corresponding author: University of Cologne, Cologne Laboratory for Economic Research, Albertus-Magnus-Platz, 50923 Cologne, Germany. Email: m.grossmann@uni-koeln.de.

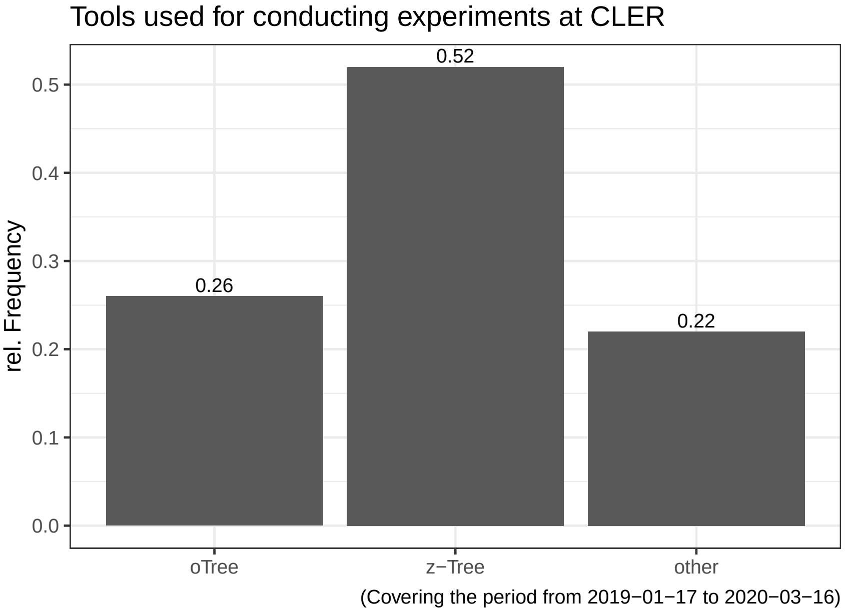


Figure 1: z-Tree usage at the Cologne Laboratory for Economic Research (CLER).

(2007). Despite the fact that several new tools have become available in recent years, a majority of economic experiments are still conducted using z-Tree (see Figure 1 for an example). There are compelling reasons for this persistence: z-Tree is easy to set up and can be learned in short order. One of the most developed and widespread alternatives is oTree by Chen et al. (2016). This Python based platform provides a flexible and powerful programming environment for interactive experiments. This flexibility, however, comes at a cost. oTree also increases the complexity of the programming environment that researchers have to handle in order to prepare their experiments.[1] The productive use of oTree or similar platforms requires considerable programming skills, and involves substantial installation time.[2]

z-Tree is based on a classic client-server architecture. The clients, i.e. the digital computers which are used by subjects during the experiment, are typically Personal Computers with Microsoft Windows. These PCs run z-Leaf, the client software. z-Leaf connects to z-Tree over the Local Area Network (LAN) and z-Tree streams the experiment over the network, while z-Leaf returns the subjects' input using a proprietary network protocol based

[1] The code for any oTree experiment is spread across several files, while zTree allows researchers to use a single file for programming and conducting the experiment.

[2] There are several other software platforms in different stages of development, e.g. BreadBoard, ConG (Pettit et al. (2014)), MobLab, NodeGame (Balietti (2017)), Psynteract (Henninger et al. (2017)) and UbiquityLab.

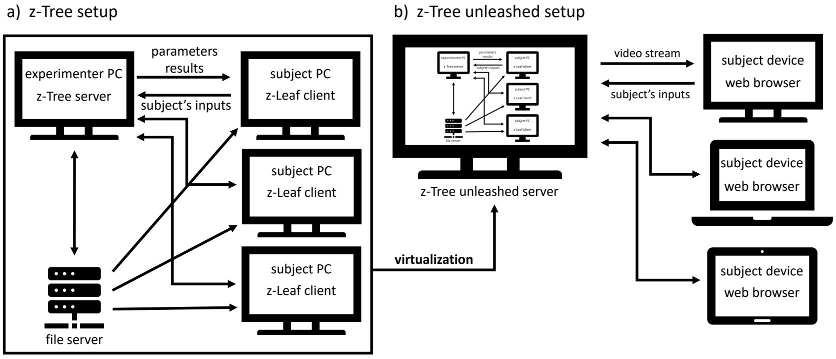


Figure 2: a) The client-server architecture of z-Tree; b) the boxed and simplified architecture of *z-Tree unleashed*.

on TCP/IP (see Figure 2, panel a)). Given that z-Leaf is connected to z-Tree over the LAN, a question that has frequently arisen is whether z-Tree experiments could also be played over the Internet, allowing subjects to dial in from anywhere in the world. However, it is generally understood that one cannot expect human subjects to use a PC, use Microsoft Windows, install foreign software on their computer, do not use Wifi or mobile Internet to connect to z-Tree to ensure network stability and to then uninstall that software after the experiment has completed. Hence, the answer to this question used to be a resounding “no”.

The global pandemic known as COVID-19 has renewed interest in the question. As scientists struggle to continue their research activities while many experimental economics laboratories are closed, many experimental researchers are looking for a method to allow subjects to participate in z-Tree experiments remotely. This reemergence has prompted us to develop *z-Tree unleashed*, which is a set of techniques and scripts to address several of the issues that arise in the discussion.

One of the most important features of *z-Tree unleashed* is that subjects are able to participate in the experiment using just a web browser. No installation of additional software is required and there is no difficulty when using an Internet connection that uses electromagnetic radiation instead of electrical or fiber-optical signals. Indeed, no difficulty arises even if subjects switch between networks during the experiment.[3] Subjects may use all operating systems and system architectures as long as they can access a graphical web browser with JavaScript enabled. Subjects can participate from anywhere in the world.

z-Tree unleashed can be run as a virtual machine, i.e. as a completely isolated system. It does not pollute any existing installation of z-Tree. Sub-

[3] Even if our web portal does not reconnect immediately, subjects can reload the portal without any loss of data.

jects are likewise completely isolated from each other; they cannot interfere with other subjects or the overall administration of the experiment. *z-Tree unleashed* upholds the privacy of its users and encrypts all traffic using HTTPS or a VPN or sends it over trusted internal interfaces (such as the loopback device).

Our entire package is permissively licensed, based on free software and can be freely redistributed (except for z-Tree itself). We provide a set of scripts as well as a user-friendly virtual machine image.

2 Advantages of *z-Tree unleashed*

In this section we briefly describe the most important features of *z-Tree unleashed* and resulting advantages for experimenters and lab administrators. While the installation of z-Tree itself has always been as simple as possible, the setup of a productive laboratory environment requires some additional steps like providing a local network and a shared file storage.

With *z-Tree unleashed* we simplify this setup to the absolute minimum (see Figure 0, panel b)). Every modern desktop computer or laptop can be used to set up an entire laboratory. All z-Tree related network communication and file transfers can be handled in a single machine and all necessary software is provided in a single virtual machine image file that can easily be mounted in a hypervisor.

Generations of experimental economics have been trained to use z-Tree and over time their experience and knowledge has become a valuable resource in many behavioral disciplines. Until now this expertise is only slowly spreading to other software tools and there is no easy way to implement existing z-Tree code in alternative software tools.

With *z-Tree unleashed* it is possible to use existing z-Tree programs without any adjustments. Not only can experimenters use their already designed experiments in a remote environment, but new experiments can still be developed on the foundation of established code and procedures. Another advantage is that all media files, predefined tables or external programs that are embedded into a z-Tree file can be stored on one device. The integrated file server obviates the need for a shared network storage that is usually used to grant file access to the z-Leaf clients.

The second group that eminently benefits from the simplified architecture of *z-Tree unleashed* are the administrators and managers of behavioral laboratories around the world. Not only does the compact architecture of our tool allow for low performance, and therefore also low cost, client computer hardware, it also minimizes maintenance and update efforts for lab administrators. Since participant devices only need to handle the z-Tree video stream in a web browser, more expensive office-type computers can be replaced by inexpensive and even single board computers. The same is true for the software that is needed for the client computers. With *z-Tree unleashed* it is possible to set up the clients without any licensing costs for operating systems or other software. These ultra compact devices only need a minimal setup to receive and display the video stream, and can easily be replaced if necessary. Moreover, these lean single board computers have about twice the service life of standard desktop computers and are therefore the more environmentally friendly choice (cf. Pattinson et al. (2015)).

An additional benefit for laboratory administrators comes from the fact that all upgrades, backups and other maintenance measures can be done on a single device. Different versions of z-Tree and z-Leaf can easily be managed in a single folder without the need to distribute the z-Leafs to dozens of client computers. Without the additional (external) file server used in classic z-Tree setups all components of an experiment (e.g. instructions, media files, data files and program files) can also be stored in a single folder. Hence, organizing a laboratory with a large number of independent experimenters becomes much easier.

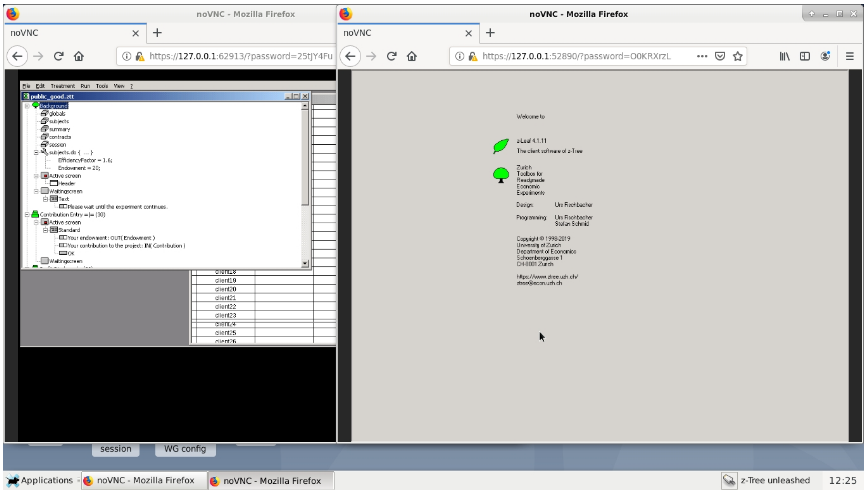
3 Use cases

Our tool not only helps reduce the efforts and costs for setting up and maintaining a traditional computerized laboratory, but at the same time allows for a tremendous extension of z-Tree's field of application. In this section we outline four important use cases with different levels of mobility for organizing, conducting and participating in experimental sessions. We start with the most common case of a (i) local laboratory setup, followed by a (ii) mobile laboratory setup, a (iii) remote setup and a (iv) tremendously mobile setup.

3.1 Local laboratory setup

Up until now most economic laboratories rely on a closed network with a dedicated file server to simplify the implementation of z-Tree's classic client-server-architecture. The virtualized architecture of *z-Tree unleashed* (see Figures 2 and 3) renders this closed setup redundant. That does not mean that existing hardware can no longer be used. The opposite is true; all pre-existing hardware components can still be used without major changes.

The only step that is necessary to switch from the existing setup to our tool is the installation of the *z-Tree unleashed* server on a computer that can be used as an experimenter's device (e.g. an existing z-Tree server). On account of the virtualization it is not even necessary to choose between the existing setup and *z-Tree unleashed*. By using a virtual machine that hosts the *z-Tree unleashed* server on the existing computer, both setups can be used in parallel.



```
Background
    globals
    subjects
    summary
    contracts
    session
    subjects.do { ... }
        EfficiencyFactor = 1.6;
        Endowment = 20;
    Active screen
        Header
    Waitingscreen
        Text
            Please wait until the experiment continues.
Contribution Entry =|= (30)
    Active screen
        Standard
            Your endowment: OUT( Endowment )
            Your contribution to the project: IN( Contribution )
            OK
    Waitingscreen
```

Welcome to

z-Leaf 4.1.11
The client software of z-Tree

Zurich
Toolbox for
Readymade
Economic
Experiments

Design: Urs Fischbacher

Programming: Urs Fischbacher
Stefan Schmid

Copyright © 1998-2019
University of Zurich
Department of Economics
Schoenberggasse 1
CH-8001 Zurich

https://www.ztree.uzh.ch/
ztree@econ.uzh.ch

Figure 3: A screenshot of a live session in *z-Tree unleashed*.

Since probably all client computers use an operating system that comes with a web browser, changes to the clients are not necessary. At the same time the pre-existing file server can still be used as a backup device for documentation, z-Tree files and raw data.

3.2 Mobile laboratory setup

The benefits of virtualization become even clearer in a mobile laboratory setup. The use of a wired local network in a portable laboratory with 20 or more clients is logistically challenging to say the least. It also puts anachronistically high requirements on the used client hardware, e.g. RJ-45 jacks for ethernet cables.

One alternative could be the use of a private Wi-Fi network; but even then additional hardware would be required. Moreover, many users report issues using a wireless connection to connect the z-Leaf clients to the server.

In contrast to that *z-Tree unleashed* does not need a dedicated network, neither in a wired setup nor in a wireless one. It can be used in any public, private or university Wi-Fi. The only requirement is that the *z-Tree unleashed* server is reachable via its IP address. Just as for the local laboratory setup our architecture allows for minimal requirements on the part of the client devices. It is even possible that participants use their own devices, as long as these devices can connect to the Internet.

3.3 Remote setup

Given that participants can use their own devices, *z-Tree unleashed* is also ready to be used remotely. That means the server can be set up in a local laboratory while participants are invited to join the experiment from wherever they are.

This of course brings up different challenges for the timing of interactive experiments, i.e. all participants have to be connected to the experiment server at a particular time. That, of course, is not only true for our setup, but for all online experiments.

In contrast to other online experiment platforms our setup does not require an explicit routine to handle temporary dropouts. z-Trees original ability to reconnect lost clients and restore the stage where they dropped out allows participants to simply re-establish the temporarily lost connection. Even if participants accidentally close the browser window, they just have to click the link from the invitation mail again in order to continue the experiment at the exact screen they stopped at.

3.4 Tremendously mobile setup

The idea of a maximally mobile laboratory can be pushed even further, by also virtualizing the used network. This way the whole laboratory setup can be boxed and used on the go. Even a cellular network would be sufficient to conduct an experiment, and a computerized data collection with z-Tree can take place almost anywhere in the field.

The tremendously mobile setup is especially useful for cross-cultural experiments, where the same z-Tree program is used in many different regions of the world. Instead of setting up an identical laboratory in every country, the researcher can simply bring her own lab with her in a briefcase. The detailed requirements and architecture of such a setup with a virtual private network (VPN) are described in Section 6.

4 Architecture

z-Tree unleashed integrates several z-Leaf clients into a single digital computer (the "server"). These z-Leaf clients can then be securely streamed over the network to subjects who participate in the experiment. The basic architecture is depicted in Figure 4.

The *z-Tree unleashed* server runs on GNU/Linux. For each individual z-Leaf client, a separate Unix user account on that server is created.[4] This segregates different users from each other and makes any unintended interference impossible. Even if a subject were able to break out of the confines of the experiment, she would not be able to disturb the overall administration

[4] In Figure 4, K users were created, and up to K z-Leaf clients can hence be run simultaneously. Currently, *z-Tree unleashed* requires that all such users share a common prefix which is appended with 1 to K. A zeroth user is used to administer the experiment and run the z-Tree graphical user interface.

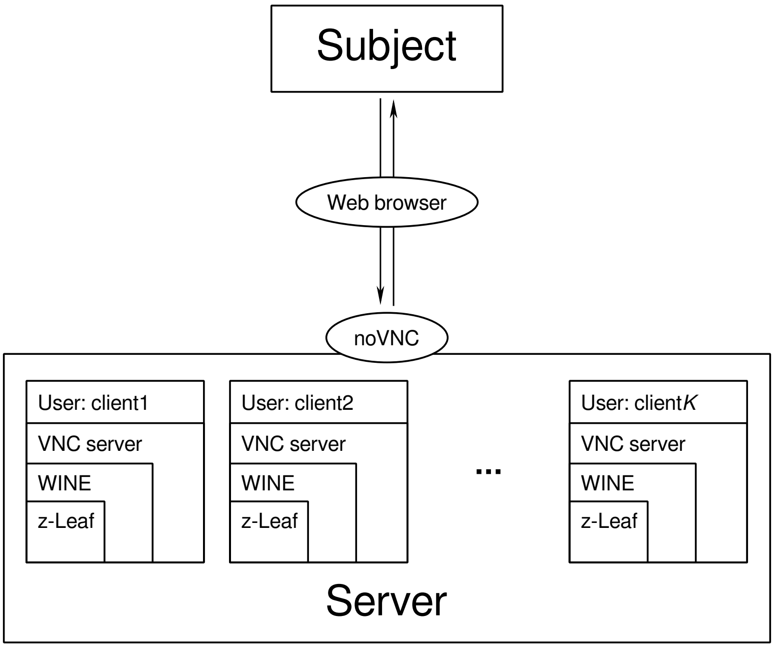


Figure 4: The *z-Tree unleashed* architecture. The reverse proxy is not shown.

of the experiment. These individual Unix users are the analogue to separate computers in a stationary experimental economics laboratory.

Upon commencement of an experiment, each Unix user executes a chain of commands. This is done automatically by the scripts that we provide. It is essential that z-Tree and each z-Leaf are emulated on GNU/Linux through Wine (cf. Amstadt and Johnson (1994)), an open source compatibility layer that allows running Microsoft Windows executables on Unix-like operating systems. Each z-Leaf automatically connects to z-Tree over the loopback interface.[5] For each user, we initialize a headless X Window System session and emulate a physical screen. Secondly, a VNC server is used to continually photograph the (emulated) screen and allow the streaming thereof over the network. However, the VNC server accepts connections only on the loopback device to preclude unwanted access from outside.

noVNC is an open sourced programming library and application that allows web browsers to connect to a VNC server. For each user, a separate instance of noVNC is run; noVNC listens for HTTP connections on a user-defined network address and port and serves a simple portal that allows dialing in to the VNC session. Ordinarily, interacting with a VNC session requires the installation of separate software, but noVNC obviates that requirement. Hence, *z-Tree unleashed* enables subjects to participate

[5] This eliminates all networking issues that could possibly arise and enables the experiment to run fluidly with the highest performance possible given the hardware layer.

by using only a web browser.

To ensure privacy and security, an additional layer of networking software is used that is not shown in Figure 4. Firstly, noVNC's HTTP server only listens on the loopback interface. Secondly, we reverse proxy requests on a different, but globally reachable port to noVNC's HTTP server. This is achieved through the use of a web server such as nginx that acts as the reverse proxy combined with a properly issued X.509 certificate for HTTPS (see Tanenbaum and Wetherall (2011), section 8.9.3).[6] Simply put, subjects access a HTTPS protected website using a web browser. On that website, they can login to the experiment and interact with z-Leaf. Their inputs are internally forwarded from the HTTPS server/reverse proxy (nginx) to noVNC's HTTP server, which is listening on the loopback device.[7] noVNC in turn converts and transmits the inputs to the VNC server, which performs the desired action over the X Window System. New frames that appear on the emulated screen (such as a dialog that appears in z-Leaf) are likewise transmitted from the X Window System to the VNC server. noVNC reads these inputs and converts and transmits them to the reverse proxy. The reverse proxy performs cryptographic primitives and sends out TCP/IP packets that are sent over the Internet stack to the subject's digital computer. The subject's digital computer ultimately displays the new frame, allowing the (human) subject to react.

This setup requires that the server on which z-Leaf unleashed is running is globally reachable via its IP address. In most cases, this will require a full-fledged setup in a datacenter, such as a Virtual Private Server (VPS) with sufficient bandwidth and computing power. However, many Internet providers offer residential customers an (ephemeral) public IPv4 address and port forwardings to devices within the LAN. In these cases, *z-Tree unleashed* can be installed and used from home networks. Additionally, it is possible to direct subjects to a user-friendly hostname by using Dynamic DNS. In instances where this is not feasible, we provide a method using a VPN and a globally reachable "Thin" server in Section 6.

[6] X.509 certificates can be procured commercially, but research institutes and services like Let's Encrypt frequently permit the issuance of universally recognized certificates free of charge. X.509 certificates can also be self-signed, but these certificates create a warning in modern web browsers as the identity of the site using the certificate cannot be verified.

[7] Since sensitive data is only ever transferred over the (internal) loopback device or encrypted and transferred over authenticated channels, *z-Tree unleashed* reachest the highest standards of data safety. Indeed, *z-Tree unleashed* is more secure than z-Tree over switched Ethernet (the standard lab configuration) since the TCP/IP packets sent and received by z-Tree can be eavesdropped at any point between the z-Leafs and the z-Tree server. This is impossible in *z-Tree unleashed*, which does not require a secure network.

5 Conducting an experiment

In both the model of Section 4 as well as that of Section 6, the basic steps are identical. A *z-Tree unleashed* server provides facilities to start, run, control and terminate an experiment. One instance of an experiment is known as a session. As soon as a session is initiated, URLs are generated that allow both the experimenter as well as subjects to access the experiment. Internally, all the software described in Sections 4 and/or 6 is initialized. That is, the experimenter receives a set of $K + 1$ URLs, the first of which streams the graphical user interface of z-Tree. All remaining K URLs may be sent out to subjects, who can subsequently access the z-Leaf interface and hence, the experiment. No password entry is required because passwords are included as part of the URL. We use port hopping (see Lee and Thing (2004)) to renew all assigned ports, internal as well as external, from session to session. This makes guessing or predicting ports unlikely. All ports lie in the range 49152–65535, which is reserved for private or temporary usage. This range of ports must be kept open on the firewall if the approach in Section 4 is to be used. The necessary z-Tree program files can be transferred into a shared directory (`/share`) using SFTP and the graphical interface of the virtual machine.

Our scripts start a predefined number of virtual clients and sets the names with which they identify themselves to z-Tree (e.g. in the clients table) accordingly. At this stage, all user passwords are changed in order to ensure that only intended subjects participate in this specific study. The resolution of the virtual client screens can be set during the initial setup or at a later point by editing the settings. As z-Leafs are sensitive to changes in screen resolution (e.g. overlapping boxes) the experimenter might define it to fit her needs. Furthermore it is possible to set the font size of the z-Leafs globally (i.e. for all radio button labels and OK buttons). This feature significantly improves the predictability of the screen output as it will not depend on the resolution of the client computer. After a short period all clients appear in the clients table of z-Tree.

The next steps depend on the intended use case. Note that the location and the type of the *z-Tree unleashed* server does not matter. What matters is that the server can be accessed by subjects and the experimenter is reachable through one of the methods outlined in Sections (4) or (6). The first use case is the local setup in an existing lab. In this case the experimenter sends each stationary client computer an individual link which will be opened in a full screen browser window by using the lab's administration software like Labcontrol.[8] It is trivial to achieve that the shown usernames of z-Leafs match the number of each seat or cabin booth. In this sense the experimenter

[8] For more information, see `https://www.kirchkamp.de//lab/zTree.html`, accessed March 30, 2020.

might give special attendance to a certain subject should they encounter an issue and the subjects' experience does not deviate from running z-Tree directly and without our scripts in the lab.

In times where personal attendance to lab experiments is impossible the second use case is vital for still conducting experiments with zTree. In this case subjects use their own computers to partake in the experiment. The unique and individual URL to access the z-Leaf is sent to the participants via email. It is merely necessary to follow a link and after connecting they are automatically logged in and the experiment might be started. Sending these personalized links gives an opportunity to accurately track and support specific subjects in the case of questions or problems.

Starting the experiment itself is straightforward as the experimenter just needs to press 'Start Treatment' inside z-Tree in his administrator browser window. During the experiment he might use z-Tree in the same way as in the lab and view available tables and submitted input. At this point z-Tree produces its output in a specified folder.[9] After the experiment is finished the experimenter might retrieve her resulting Excel file and other generated files via SFTP from inside the virtual machine.

6 The tremendously mobile laboratory

As stated earlier, the *z-Tree unleashed* server requires a globally reachable IP address to act as an endpoint for the interconnection with subjects. However, with little additional overhead, *z-Tree unleashed* permits the use of ordinary hardware to achieve its ends. Networking professionals are well aware of the challenges implied by the use of consumer-grade internetworking links, such as the proliferation of IPv4 network address translation (NAT, cf. Tanenbaum and Wetherall (2011), pp. 452-455). The lack of sufficient IPv4 address space makes giving each Internet-capable device its own IPv4 address impossible. Hence, in a normal setup as in Figure 4, a server that is run outside a institutionally provided network infrastructure could not be accessed by subjects as it is usually behind a router that provides NAT. On the other hand, IPv6, where available, was designed such that each device receives a globally unique and globally reachable IPv6 address. But router firewalls frequently restrict IPv6 traffic and shield devices from ostensibly unwanted traffic. Even where "port forwardings" are possible, setting them up may expose the server to unwanted and dangerous traffic. Finally, setting up Dynamic DNS or relying on port forwardings at all may be infeasible in scenarios where the *z-Tree unleashed* server is used in a mobile configuration.

If consumer-grade hardware and network links are to be used with *z-*

[9] For each session, a unique folder in `/share/scratch` is created. This alleviates the well-known limitation that made opening z-Tree more than once per minute impossible; since z-Tree's output is redirected to the unique folder, the limitation does not apply.

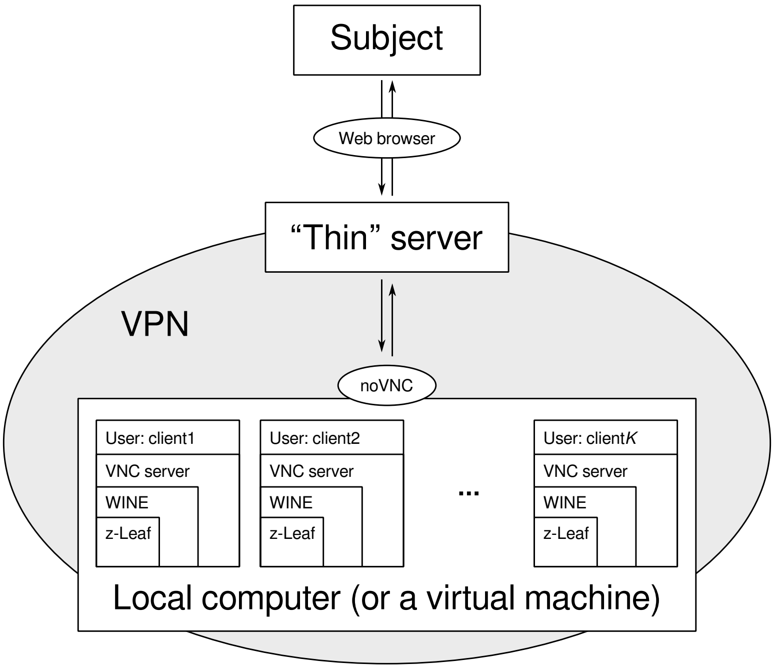


Figure 5: The *z-Tree unleashed* architecture with tunneling. The reverse proxy is not shown.

Tree unleashed, we must make what subjects connect to and participate on globally reachable, although within parameters and perhaps only indirectly. We also must provide a secure and authenticated method to allow subjects to connect with the server. All of these ends are accomplished through the use of a virtual private network (VPN), see Figure 5.

In this model, subjects do not establish a direct connection with the *z-Tree unleashed* server. Instead, they establish a connection with a "Thin" server, which in turn routes the traffic to the *z-Tree unleashed* server over the VPN. Conceptually, the "Thin" server has only three requirements: to be globally reachable by subjects; to provide an entry and exit point for the VPN; to route traffic between the subjects and the *z-Tree unleashed* server.[10] Essentially, we offer a private port forwarding that is independent of the *z-Tree unleashed* server's reachability and never exposes the server to insecure or unwanted traffic.

WireGuard (see Donenfeld (2017)) is a recent protocol and software tool for the establishment of virtual private networks. WireGuard enables users to establish secure connections between at least two peers even over insecure

[10] In our investigation, we found that "Thin" servers with sufficient functionality and bandwidth are available for less than €3/month. This includes one static IPv4 address, making the "Thin" server globally reachable. A "Thin" server is not just "Thin", it is also "Cheap".

networks. The connection between two peers is always authenticated and end-to-end-encrypted using the well-known Elliptic Curve Diffie-Hellman key exchange and related cryptographic protocols. We recommend the use of WireGuard because of its simplicity. A working configuration can be established with 9 lines on the "Thin" server and 8 lines on the *z-Tree unleashed* server. WireGuard assigns to each peer a private static IP address. Indeed, this setup allows experimenters to run a virtual machine that handles all details of *z-Tree unleashed* on their own digital computers at home. Since this virtual machine and the "Thin" server are interconnected through WireGuard, subjects only interact with the "Thin" server, which can have a user-friendly domain name that points to the "Thin" server's public static IP. WireGuard is seamlessly integrated into an existing networking infrastructure and it straightforwardly makes *z-Tree unleashed* available through NAT and similar complications. This setup does not require dynamic DNS. In our set of scripts, we provide a facility to generate WireGuard configuration files for both the "Thin" server as well as the *z-Tree unleashed* server.

As mentioned previously, the *z-Tree unleashed* server's noVNC instances create a HTTP server. We now configure each instance of noVNC such that the HTTP server binds to the *z-Tree unleashed* server's static private IP address instead of the loopback device. As soon as the "Thin" server and the *z-Tree unleashed* server are connected through WireGuard, internal traffic can flow between the two servers using these internally assigned IP addresses. Binding to the static private IP address ensures that only other devices in the VPN can access the service, such as the "Thin" server.[11] By reverse proxying requests to the "Thin" proxy over the WireGuard interface to the *z-Tree unleashed* server, we enable subjects to securely access the interface over HTTPS even as the actual traffic is redirected from the "Thin" server to the *z-Tree unleashed* server and vice versa. Subjects do not notice that the actual server is not the "Thin" server, and hence the forwarding is transparent. The strong encryption used by WireGuard makes reverse proxying HTTP over the VPN completely secure and even enables experimenters to roam between networks without interruptions, cf. Donenfeld (2017). In other words, this mode of operation views the reverse proxy and the *z-Tree unleashed* server as distinct physical devices and conceptually moves the *z-Tree unleashed* server away from the subject.

The WireGuard VPN enables any computer with sufficient hardware and uplink to act as a *z-Tree unleashed* server, not just globally reachable servers. This special derivative architecture makes the lab tremendously mobile. It only has one stationary component: The "Thin" server, which can simultaneously be used for many other purposes. However, in contrast to other use cases, neither the subjects' computers nor the experimenter's

[11] This implies that the actual raw HTTP or VNC services are never directly made publicly available.

computer must meet demanding requirements such as global reachability. This laboratory can be put in a briefcase.

7 Performance, requirements and network usage

Running many clients on a single machine imposes significant strain on the hardware. We reduce each processor's load by allocating subsets of z-Leaf instances to different processor cores, but withhold one solely for the central z-Tree instance.[12] The resulting performance improvements facilitate the productive use of *z-Tree unleashed*. Using a minimal graphical interface and specific commands to start the clients' graphical user interfaces we were able to reduce the memory load to a very low level. As can be seen in Figure 6, a full setup of *z-Tree unleashed* using 32 connected clients running a resource demanding experiment did not use more than four gigabytes of RAM. Moreover, using our environment for conducting a regular experiment we found a steady memory consumption which is even available in most personal computers. But in order to provide subjects a reasonably smooth user experience, we recommend, given these observations, the use of highly parallelizable hardware and at least 4 processor cores for running complex experiments. While this appears to make the use of dedicated hardware for the single purpose of running *z-Tree unleashed* more attractive, the cost of such hardware in datacenters may be substantial. In most cases, it will also not be strictly necessary to run the *z-Tree unleashed* server for 24 hours a day.

Another astonishing fact was that the uplink bandwidth available in our tests was low and at the same time we experienced a latency when transmitting inputs of almost zero. We expected to need a very fast Internet connection, but we were surprised to find that *z-Tree unleashed* was highly responsive. During testing we used a consumer-grade Internet link and did not notice any latency or interruptions. As depicted in Figure 6 our test session created only a total traffic of a few dozen megabytes. Therefore it is well possible to use *z-Tree unleashed* over a cellular data link.

8 Conclusion

In this paper, we presented *z-Tree unleashed*, a novel tool with a set of scripts to enable the administration of economic laboratory experiments created with z-Tree over the Internet. *z-Tree unleashed* enables experimenters and lab administrators to "put the lab in a briefcase", and it grants human subjects the opportunity to rely on secure and resilient communication channels while participating from anywhere in the world.

[12] Basically we customized a command described by Oliver Kirchkamp, see `https://www.kirchkamp.de//lab/zTree.html`, retrieved March 30, 2020.

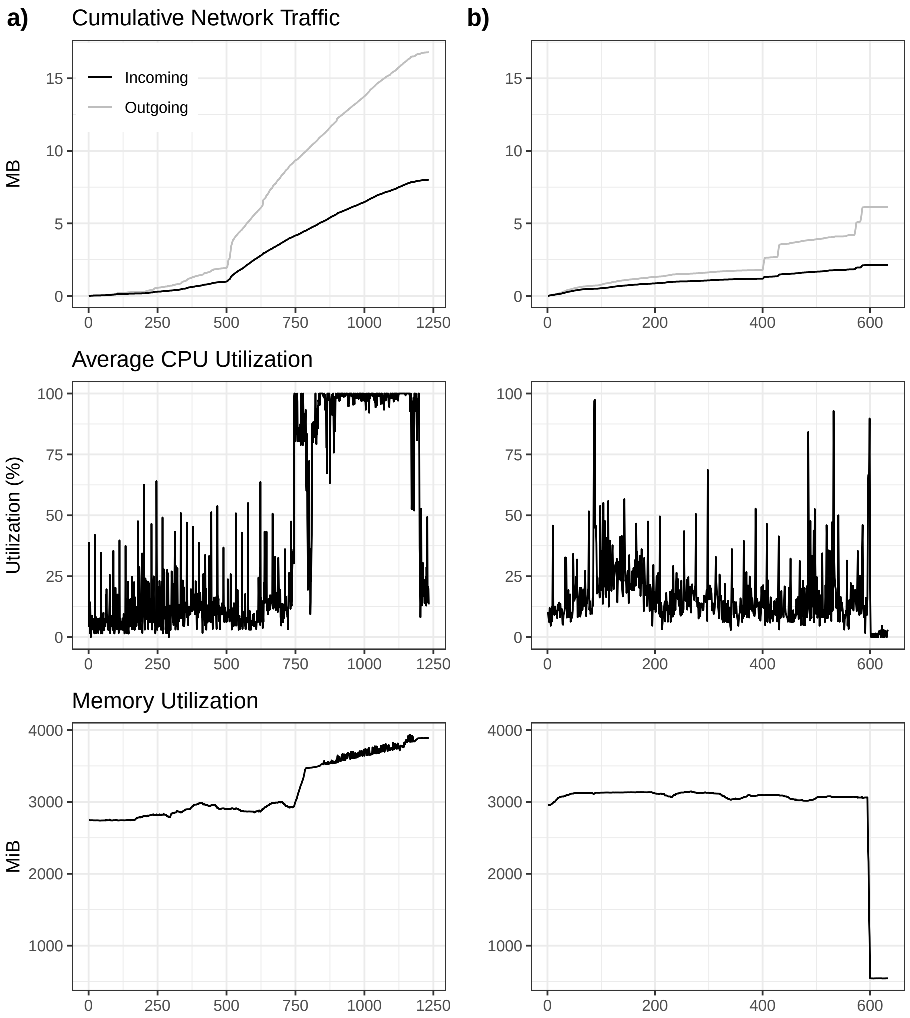


Figure 6: On the left: system utilization while running GIMS, a heavy load market simulation. On the right: a regular public goods game. Usage was recorded every half second.

The two main challenges involved with *z-Tree unleashed* are the hardware and the network bottleneck. But as outlined in the previous sections, *z-Tree unleashed* has moderate processing power and bandwidth requirements. Other challenges that were traditionally salient with the implementation of online experiments, such as the global reachability of the server itself, have been overcome through the use of a VPN. We have also addressed data privacy concerns through the use of authenticated and end-to-end-encrypted networking channels.

Our tool can not only be used to simplify existing laboratory setups and take them to the Internet, but also to build low budget laboratories that can be used in every place with any network connection, cellular, Wi-Fi or LAN. *z-Tree unleashed* can also help to reduce downtime by allowing for a simple replacement of server and client hardware, and by being able to switch to another network easily. Finally our tool can also help to reduce the carbon footprint of your lab by encouraging the use of "thin" hardware with low power usage and longer service times.

z-Tree unleashed has been released under a permissive free software license, allowing anyone to use, adapt and redistribute the software package. Our license is a derivative of the Mozilla Public License 2.0 with an additional clause requiring the citation of this paper in all academic publications that make use of the *z-Tree unleashed* scripts or derivatives thereof.

In principle, the methodology outlined here applies to any stationary lab software. With little effort, our scripts could be adopted such that other software can be "streamed" over the Internet. We invite discussion and contributions to our scripts on the GitLab repository and the Google Group of *z-Tree unleashed*.[13]

On our project website, we provide a royalty-free virtual machine image that can be used on dedicated hardware as well as for the occasional experiment. This virtual image contains all *z-Tree unleashed* scripts and allows lab managers and researchers the flexibility of risklessly experimenting with different setups. An additional advantage is that firstly, the virtual machine image is an isolated container that can be trivially transferred between digital computers; and that secondly, many adjustments can be made such as an increase of the number of processor cores that are available to the virtual machine. Our virtual machine has been created in and can be imported into Oracle VM VirtualBox, a free software hosted hypervisor. It is essentially "plug and play".

[13] See `https://gitlab.com/cler1/zTree-unleashed` as well as `https://groups.google.com/d/forum/z-tree-unleashed`.

9 Acknowledgments

We acknowledge funding by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under grant agreement No. 741409, as well as by the German Research Foundation (DFG) under Germany's Excellence Strategy (EXC 2126/1—390838866).

We thank Urs Fischbacher for supporting our idea and allowing us to use the name z-Tree.

References

Amstadt, B. and Johnson, M. K. (1994). "Wine". *Linux Journal*, 1994(4es), p. 3 (cited on p. 8).

Balietti, S. (2017). "nodeGame: Real-time, synchronous, online experiments in the browser". *Behavior research methods*, 49(5), pp. 1696–1715 (cited on p. 2).

Chen, D. L., Schonger, M., and Wickens, C. (2016). "oTree—An open-source platform for laboratory, online, and field experiments". *Journal of Behavioral and Experimental Finance*, 9, pp. 88–97 (cited on p. 2).

Donenfeld, J. A. (2017). "WireGuard: Next Generation Kernel Network Tunnel." In: *NDSS* (cited on pp. 12, 13).

Fischbacher, U. (2007). "z-Tree: Zurich toolbox for ready-made economic experiments". *Experimental economics*, 10(2), pp. 171–178 (cited on p. 1).

Henninger, F., Kieslich, P. J., and Hilbig, B. E. (2017). "Psynteract: A flexible, cross-platform, open framework for interactive experiments". *Behavior research methods*, 49(5), pp. 1605–1614 (cited on p. 2).

Lee, H. C. and Thing, V. L. (2004). "Port hopping for resilient networks". In: *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004*. Vol. 5. IEEE, pp. 3291–3295 (cited on p. 10).

Pattinson, C., Cross, R., and Kor, A.-L. (2015). "Chapter 14 - Thin-Client and Energy Efficiency". In: *Green Information Technology.* Ed. by M. Dastbaz, C. Pattinson, and B. Akhgar. Boston: Morgan Kaufmann, pp. 279–294 (cited on p. 5).

Pettit, J., Friedman, D., Kephart, C., and Oprea, R. (2014). "Software for continuous game experiments". *Experimental Economics*, 17(4), pp. 631–648 (cited on p. 2).

Tanenbaum, A. S. and Wetherall, D. J. (2011). *Computer Networks.* 5th ed. Prentice Hall (cited on pp. 9, 11).