ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-M02A006

# NEUTRON FROM A DISTANCE: REMOTE ACCESS TO EXPERIMENTS

P. Mutti*, F. Cecillon, C. Cocho, A. Elaazzouzi, Y. Le Goc, J. Locatelli, H. Ortiz
Institut Laue-Langevin, Grenoble, France

*Abstract*

Large-scale experimental facilities such as the Institute Laue-Langevin (ILL) are designed to accommodate thousands of international visitors each year. Despite the annual influx of visitors, there has always been interest in options that don't require users to travel to ILL. Remote access to instruments and datasets would unlock scientific opportunities for those less able to travel and contribute to global challenges like pandemics and global warming. Remote access systems can also increase the efficiency of experiments. For measurements that last a long time scientists can check regularly on the progress of the data taking from a distance, adjusting the instrument remotely if needed. Based on the Virtual Infrastructure for Scientific Analysis (VISA) platform, the remote access becomes a cloud-based application which requires only a web browser and an internet connection. NOMAD Remote provides the same experience for users at home as though they were carrying out their experiment at the facility. VISA makes it easy for the experimental team to collaborate by allowing users and instrument scientists to share the same environment in real time. NOMAD Remote, an extension of the ILL instrument control software, enables researchers to take control of all instruments with continued hands-on support from local experts. Developed in-house, NOMAD Remote is a ground-breaking advance in remote access to neutron techniques. It allows full control of the extensive range of experimental environments with the highest security standards for data, and access to the instrument is carefully prioritised and authenticated.

## SETTING THE SCENE

Neutron experiments play a pivotal role in various scientific domains, ranging from fundamental physics to materials science and biology. The ability to control and manipulate the measurements remotely has become increasingly important, especially in the wake of global challenges such as the COVID-19 crisis which has dramatically transformed our work practices. The ILL, as many other research centres around the world, has played a crucial role during the pandemic. To allow researchers to dig into the secrets of the virus, the beam lines had to stay operational even during the confinement periods. We had to find quickly solutions to allow scientists, both internal as well as external users, to perform their experiment without being on-site. After the pandemic crisis was over, remote working and the increasing awareness on the climate impact of long air-plane travels, has strengthened those needs for remote access to experiments. Moreover, this possibility has gained prominence due to its potential to overcome geographical, logistical, and safety constraints. NOMAD [1] is the ILL control software deployed on all 50 instruments in operation at the institute. Its client-server architecture was from the very beginning compliant with the model-view-controller software pattern principle, but it was necessary to confront our architecture with these new use-cases. The adopted solution is depicted in Fig. 1. The CAMEO [2] application manager is the core of the NOMAD ecosystem. It is handling the life-cycle of all the other applications, including NOMAD-Server and Nomad-GUI. It is also responsible for the execution of the real-time plot and, if any, of all third party applications as, for example, the automatic data reduction. Nomad-Server is handling the connection with the central data storage, the electronic logbook and the parameters survey writing into the database as well as the alarms management.

### *A Multi-Client Solution*

NOMAD has natively multi-client capabilities. Within the ILL we had so far three different clients: the one running on the instrument computer, those installed in the instrument cabins or the offices and a remote Android client running on tablets. All these GUI instances require a specific local software installation and configuration. To extend this concept to a remote client we had, first of all, to find a solution that did not require any intervention on the distant machine. The natural answer came with a web-based remote desktop of a virtual machine (VM). Using OpenStack [3] virtualization infrastructure coupled with Apache Guacamole [4] client-less remote desktop gateway, we could find an optimal compromise accounting for all our needs. First, the security aspects could be totally handled by the web connection, including double factor authentication. Second, the GUI application could be included directly in the image used to generate the VM and kept up-to-date by an automatic synchronization with the software repository. In such a way, we could preserve the data policy and prevent undesired connections thanks to a web-service which connect in a unique way the user demanding a connection with a specific instrument running a well defined experiment. The network integrity is guaranteed by the assignment of specific communication ports between server and client to a user for a specific experiment. The look-&-feel of the GUI is preserved since the user is running on the VM exactly the same application as on instrument.

In our use-case we can distinguish three different user's types:

- *superusers* who are members of the instrument control department and have permission to configure all devices,
- *instrument managers* who drive and supervise the experiment and who can change the instrument configu-

* mutti@ill.fr

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-M02A006

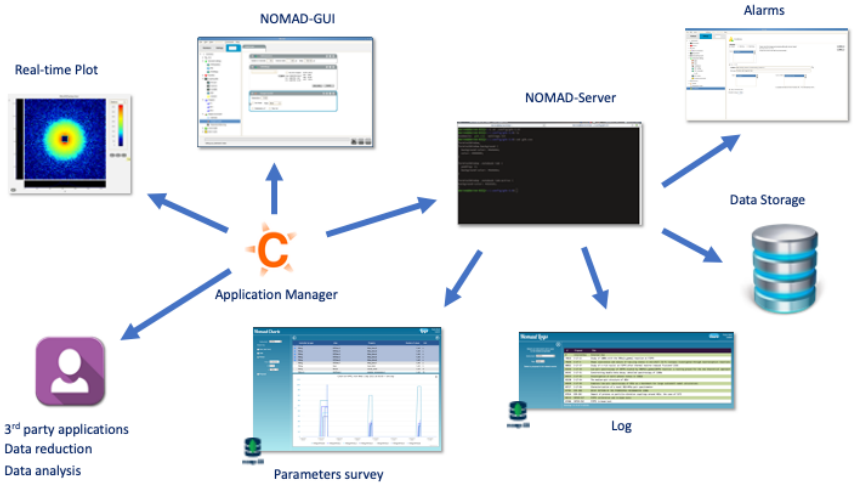


Figure 1: The NOMAD ecosystem showing the link between the different applications.

ration,

- *external users* who monitor the parameters of the instrument during the measurements but also program the sequence of actions to be performed during the experiment.

Each user category needs an interface which has different rights and modus-operandi. The initial authentication, as described above, allows us to master these different roles. To prevent the concurrent interaction between the various clients and the single server instance, we have introduced a *token*. This can be requested by the client which desires to interact with the server while all the other remain pure observers. A given client owns the token as long as needed or until a time-out occurs if no activity is detected. The GUI is user-centric and will restrict capabilities according to the user's role and the current instrument status.

The *main* client is located geographically close to the instrument and it fulfils the needs of superusers and instrument managers. Thanks to a specific configuration it can at any time reacquire the token and become the main cockpit driving the experiment, changing parameters or adjusting the configuration of the instrument. All other remote clients are running from a VM instantiated from our web-portal Virtual Infrastructure for Scientific Analysis (VISA) developed within the Photon and Neutron Open Science Cloud (PaNOSC) [5] European initiative. They can also acquire the token to directly interact with the instrument and this last will be assigned, when free, according to a request list. Every remote client receives all the status updates from the server but it can freely customise the interface layout without interfering with the other clients. Only the execution workflow is unique and common to all clients. Therefore, a user can open a plot, edit a new work-flow, look at the electronic logbook or run a simulation in a completely independent way. All remote clients are uniquely identified and their activity is traceable and recorded in the electronic logbook.

## CAMEO APPLICATION MANAGER

Most of the ILL instruments have a server computer and one or two main client. A *real* NOMAD-server application is running on this machine, meaning that it has all the physical devices of the instrument connected and communicating. The NOMAD-GUI client application is started on user demand and executes either on the client computers or on a VISA instance, making it the basis for NOMAD Remote. Since one can have as many VISA instances as possible, the architecture of NOMAD is one *real* server for N clients as shown in Fig. 2.

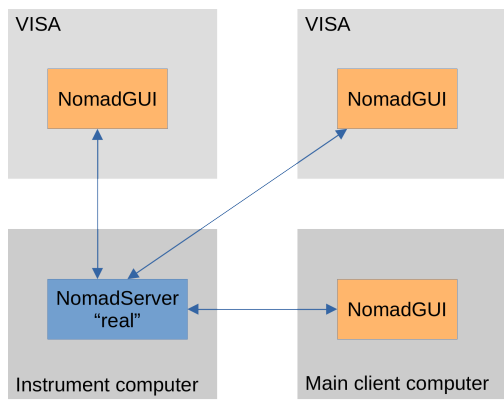


Figure 2: The NOMAD architecture with one single server and many distributed clients.

The interaction between the clients and the *real* server uses a Request/Response as well as a Publish/Subscribe communication. However it is not limited to that. The life cycle management of the applications is also included in the interaction. The CAMEO middle-ware that we have developed at the ILL provides all the mechanisms to start, stop and make applications communicate in different ways. It provides services to easily write distributed applications running on any platform and language, making it available for any non-network expert.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-M02A006

To use the CAMEO middle-ware, it is necessary to install a CAMEO server on each computer as shown in Fig. 3. The CAMEO server is configured with a list of runnable applications. Such a list may differ on each computer because some instruments require extra specific applications. However all server computers have at minimum the application NOMAD-server available.

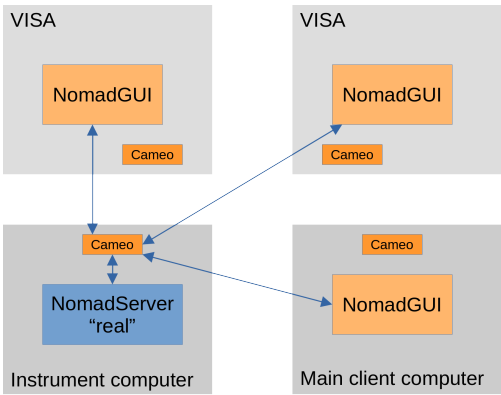


Figure 3: Schematic of the different CAMEO installations.

For different reasons, a *real* NOMAD-server may not be running when a client is started by a user. The NOMAD-GUI client application first connects to the CAMEO server of the instrument computer and requests the connection to the NOMAD-server application. If this last is not already running, the client application requests its start. However for security reasons, a client on a VISA instance cannot start the NOMAD-server application. The start of the server is done by the CAMEO server which owns the main process. Later the NOMAD-server running application can be stopped by contacting the CAMEO server. Once both client and server have been started, the communication can be established. It is done using high-level patterns provided by the CAMEO APIs in C++, Java and Python. Besides CAMEO is using the high-performance ZeroMQ [6] sockets for all the communications – server and APIs. CAMEO patterns offer very useful properties like synchronization of the Requester and Responder so that, if the Requester started before, the Responder is waiting for it. Time-out in communications is also managed even if it is not the preferred way of using ZeroMQ. The NOMAD-GUI application is sending request messages to the server application e.g. toggling a check button or starting the sequence of programmed commands. A Requester is used to send request messages and a Responder on the NOMAD-server side is responding to them. Some event messages are also sent from the NOMAD-server using a Publisher to which all the clients subscribe. The events are for instance the state of the execution of the previous sequence or the actual positions of a motors. However in a multi-client environment, a button that has been toggled on one client must also be toggled on all the other clients. For that, every request identified as requiring a notification is transformed into an event message that is published by the NOMAD-server to all the clients. We call them synchronization events. It means that if the main client is toggling a check button, all the clients on the VISA instances will receive a message to toggle the same button on the GUI interface. We presented the execution of a single *real* NOMAD-server application however there are some cases where we run *simulated* servers i.e. without any real connected devices but replaced with their *virtual* version. This is typically the case when a user needs to verify the execution of a sequence of commands. Very frequently this happens on Three-Axis Spectrometer (TAS) where a complex configuration of the instrument might drive some elements in a forbidden status (collision with walls, etc.).

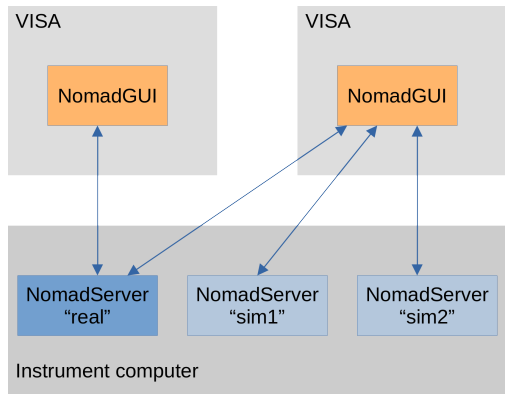


Figure 4: Schematic of the execution of *simulated* NOMAD-server.

Figure 4 shows how the *simulated* NOMAD-servers are executed on the instrument computer. However they are started from NOMAD-GUI client requests within a VISA instance. A second use-case is when a user wants to prepare its future experiment and needs to have a *simulated* NOMAD-server to prepare some work-flows that will be executed during the future measurement. In that case, the user can start a *simulated* NOMAD-server that we define as *virtual* in that case. Figure 5 shows the simple execution of the NOMAD-server applications directly on a VISA instances.

We have previously seen that for an instrument, an indefinite number of NOMAD applications can be created on demand by the users. Each VISA instance can launch a NOMAD-GUI application and many *simulated* NOMAD-servers. All that requires a minimum of control over the application instances to prevent from keeping orphaned processes. The CAMEO middle-ware focuses on applications called *Apps* and provides the necessary services to do it. Four use-cases are taken into account:

- The NOMAD server is stopped: It can be stopped intentionally or unexpectedly with an exception. Then, all the NOMAD-GUI clients connected to it are shutdown. To realize that, the NOMAD-GUI application has a reference to the NOMAD-server application and follows its state. If its state becomes a terminal state, the client stops with an appropriate message for the user.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-MO2A006

- A NOMAD-GUI client is stopped: The server registers all the connected clients and needs to track their states to keep the list up-to-date. For that NOMAD-server has a list of references to each NOMAD-GUI application and follows their state. If the state becomes a terminal state then it is removed from the list. Moreover, a VISA instance can be shut-down meaning that it becomes unreachable. Then the client is removed from the list.
- A NOMAD-GUI client is sleeping: If a VISA instance is sleeping because it is not used for a while, a client can become unreachable from the server. It is removed from the list but can come back later when it awakes. In that case the server tells the client to stop with an appropriate message to the user.
- Clean the orphaned *simulated* NOMAD-server applications: A *simulated* NOMAD- server is executed on the instrument computer but is launched from a VISA instance or from the main client. It is only seen by the client that started it. If the client stops intentionally, it requests the stop of all the associated *simulated* servers. However if the client is stopped unexpectedly i.e. with an exception or with the shut-down of the VISA instance, a CAMEO feature is used to automatically stop the servers. They are started with the linked option so that each server is tracking the state of their starter *Apps*.

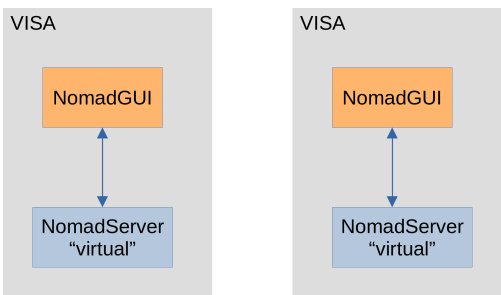


Figure 5: Schematic of the execution of a *virtual* NOMAD-server directly within a VISA instance.

All the communication between the NOMAD applications is done by using the CAMEO middle-ware which is based on ZeroMQ. The communication patterns Responder and Publisher use a main ZeroMQ socket bound to a TCP endpoint which uses a network port. An instance of NOMAD-server defines many Responder and Publisher objects which leads the number of opened ports around 10. Moreover, we have seen previously that many *simulated* servers can be executed on the instrument computer. This lead to a possible increase of the number of opened ports up to 50 in the case of 4 *simulated* servers.

This variable number of ports is not acceptable by the network infrastructure as the instrument computers and the VISA instances are on different networks. We solved this problem by using the proxy feature of CAMEO. All the client request messages are sent to the CAMEO proxy router socket which forwards the messages to the wanted socket.

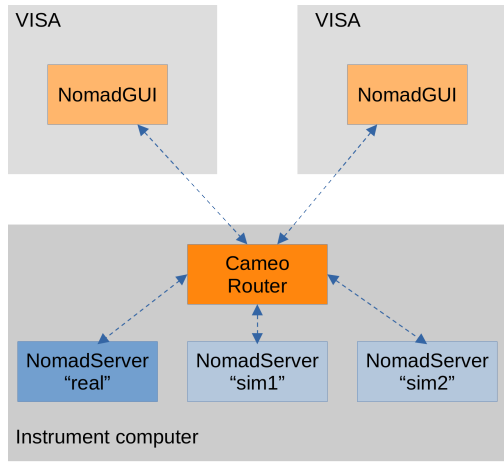


Figure 6: Schematic of the router with the NOMAD environment.

On the other side, all the opened sockets in the instrument computer send their response messages to the proxy router which forwards them to the appropriate request socket. The same principle is used for all the event messages sent by the Publisher sockets. Figure 6 shows the router in the NOMAD environment. In fact there is also a CAMEO router running on each VISA instance but it is not displayed for readability reasons. With the router solution, the number of opened ports in the firewall is limited to two which is acceptable by the network infrastructure.

## NOMAD REMOTE UTILITIES

Since the deployment of the NOMAD Remote to the various instruments, the demand from the users of increase connectivity has dramatically increased. The distant access allows a greater number of scientists to contribute to the measurements and therefore to improve the team work. As a consequence we have embedded within the NOMAD-GUI a number of tools to facilitate the exchanges amongst the experimental team and the local experts. Figure 7 shows the NOMAD cockpit with the different icons reporting the status of the token (*a*), the activation of the remote connection (*b*). Figure 7(c) allows to start a chat amongst the participants to the experiment so that they can exchange messages and share documents.

Figure 7(d) opens a video-chat application based on Jitsi [7] free video conferencing. At the moment when a new experiment is created, NOMAD-server generates an encrypted key which is used as unique identifier to create a new video chat-room and which is made available to all participants. Figure 7(e) contains the list of all connected remote sessions and allows a given user to request the token if he does not own it.

## INFRASTRUCTURE

As already mentioned earlier in the text, NOMAD Remote runs on a VISA instance. To support the load of the

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-M02A006


Figure 7: The NOMAD cockpit gives access to the different tools to facilitate the exchange between the experimental team and the local experts.

virtualization infrastructure the IT department has deployed dedicated hardware which is able to account for more than 500 simultaneous sessions. This consists of a servers rack with 216 cores at 3.1 GHz, equivalent to 432 virtual CPUs. The total amount of memory is 6 TB and the system is linked to the central storage with a 20 Gbps network access.

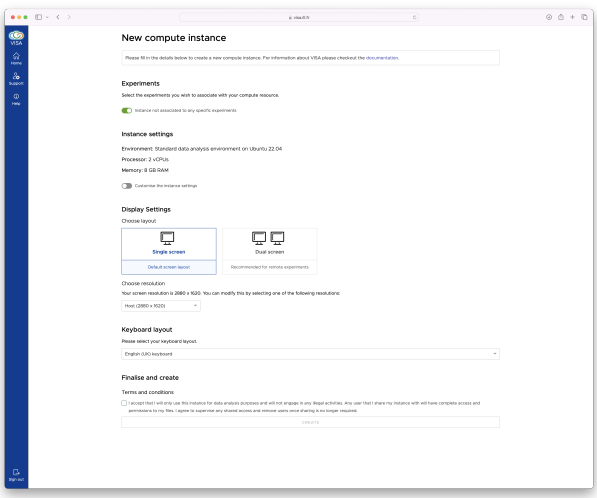


Figure 8: VISA web interface for the creation of a new instance.

At the moment of the creation, the user has the option to customize the resources of the new VISA instance (see Fig. 8). As a default, each VM used for remote experiment control consists of 2 vCPUs and 8 GB of memory. Those limited resources are more then enough to run the NOMAD client which is very light-weight. For more computational intense tasks as, for example, data reduction or data analysis more resources can be allocated on demand. To avoid ghost instances remaining inactive for long time, the lifetime of a given instance has been limited to 4 days when no activity is detected.

## CONCLUSION

Remote access to experiments is an increasingly vital aspect in many research fields, offering numerous benefits and addressing various challenges. We have demonstrated how the solid client-server architecture of the instrument control software NOMAD is perfectly adapted to this novel use-case. Moreover, the CAMEO middle-ware that we developed at the ILL has proven to be an essential tool to provide a robust and reliable NOMAD multi-client environment to the users. The strong link between NOMAD Remote and the VISA infrastructure provides a great environment for instrument control as well as for data analysis on the same platform. Thanks to the users feedback the NOMAD Remote ecosystem has grown, including numerous tools to improve the quality and the efficiency of the distant interaction between external users and local experts. During the last reactor cycle of 60 days, our beam lines were performing a total of 441 experiments both in presence and remotely. During this period we could count 1116 different users creating a total of 7776 VISA instances for more then 65 thousands remote connections.

## REFERENCES

[1] P. Mutti *et al.*, "NOMAD - more than a simple sequencer", in *Proc. ICALEPCS'11*, Grenoble, France, Oct. 2011, paper WEPKS014, pp. 808–811.

[2] CAMEO, `https://code.ill.fr/cameo/cameo`

[3] OpenStack, `https://www.openstack.org/use-cases/containers/leveraging-containers-and-openstack`

[4] Guacamole, `https://guacamole.apache.org/doc/gug/guacamole-protocol.html`

[5] PaNOSC, `https://www.panosc.eu`

[6] ZeroMQ, `https://zeromq.org`

[7] Jitsi, `https://jitsi.org`