

Geppetto [1] is an open-source platform for building web applications for visualizing neuroscience models and data, as well as managing simulations. Geppetto underpins a number of neuroscience applications available to the research community, including Open Source Brain (OSB) [2], Virtual Fly Brain (VFB) [3], NetPyNE-UI [4] and a new web user interface for the Human Neocortical Neurosolver [5].

While Geppetto was originally created with a JavaScript based frontend for user interactions on a browser and a Java based backend hosted on a server we have now extended it to also use a Python based backend. This means that applications built with Geppetto now also offer their users the ability to interact directly with any underlying Python APIs, while seamlessly keeping the user interface synchronised. Python Geppetto applications can be deployed locally, installed using standard Python packages (accessible from PyPI) or Docker, or they can be deployed remotely on the web using Kubernetes and JupyterHub.

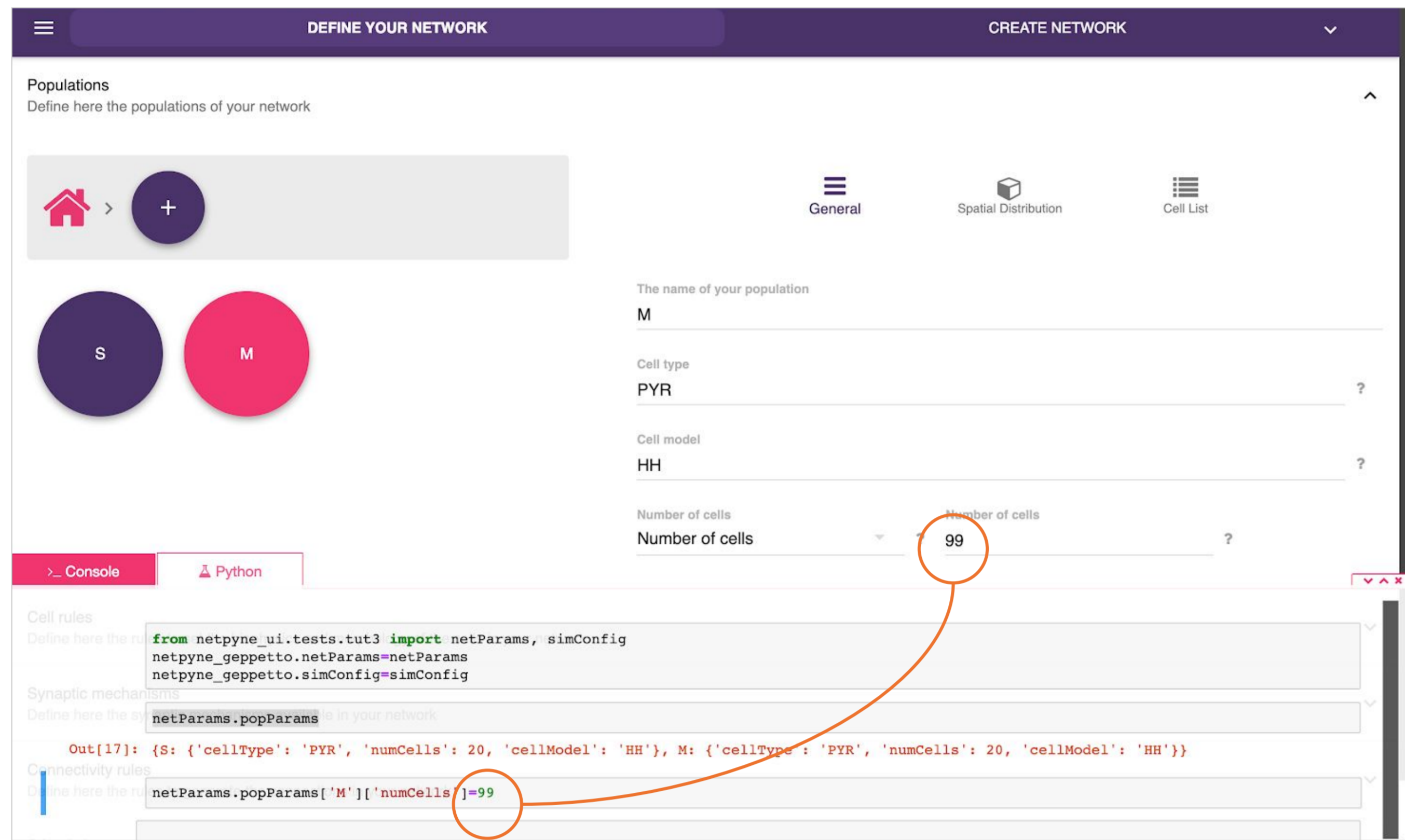


Figure 1. NetPyNE-UI [4] as an example of an application built with Python Geppetto. In the screenshot the number of cells for population M was programmatically changed via an integrated Jupyter notebook (bottom tabbed panel), causing the Graphical User Interface (GUI, top) to automatically update.

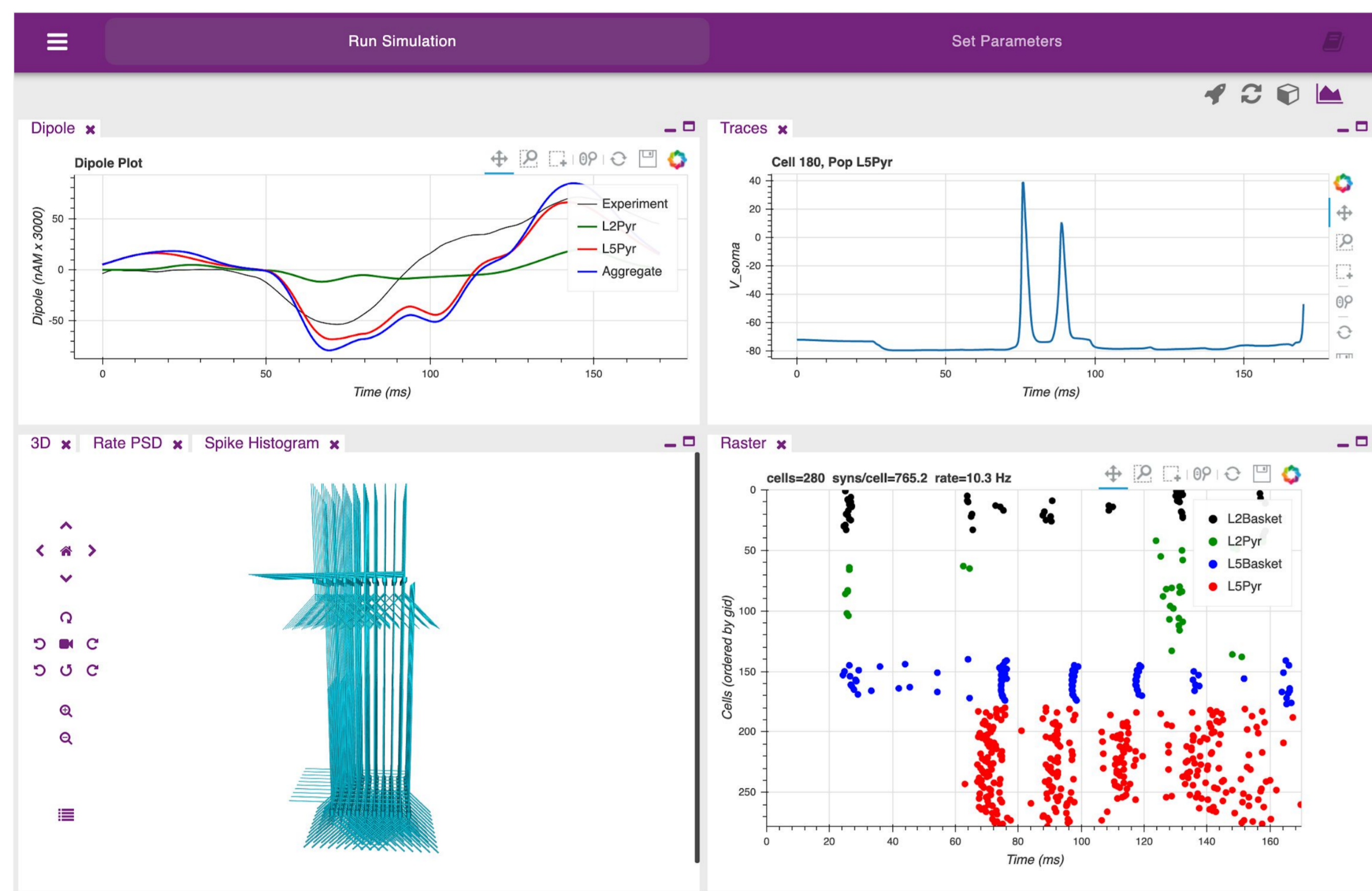


Figure 2. HNN-UI [5] as an example of an application built with Python Geppetto. The simulation was executed in Python using NetPyNE. The analysis plots are generated also in Python while the 3D canvas is reusing the JavaScript Geppetto component. Geppetto makes it possible to seamlessly combine the different visualizations.

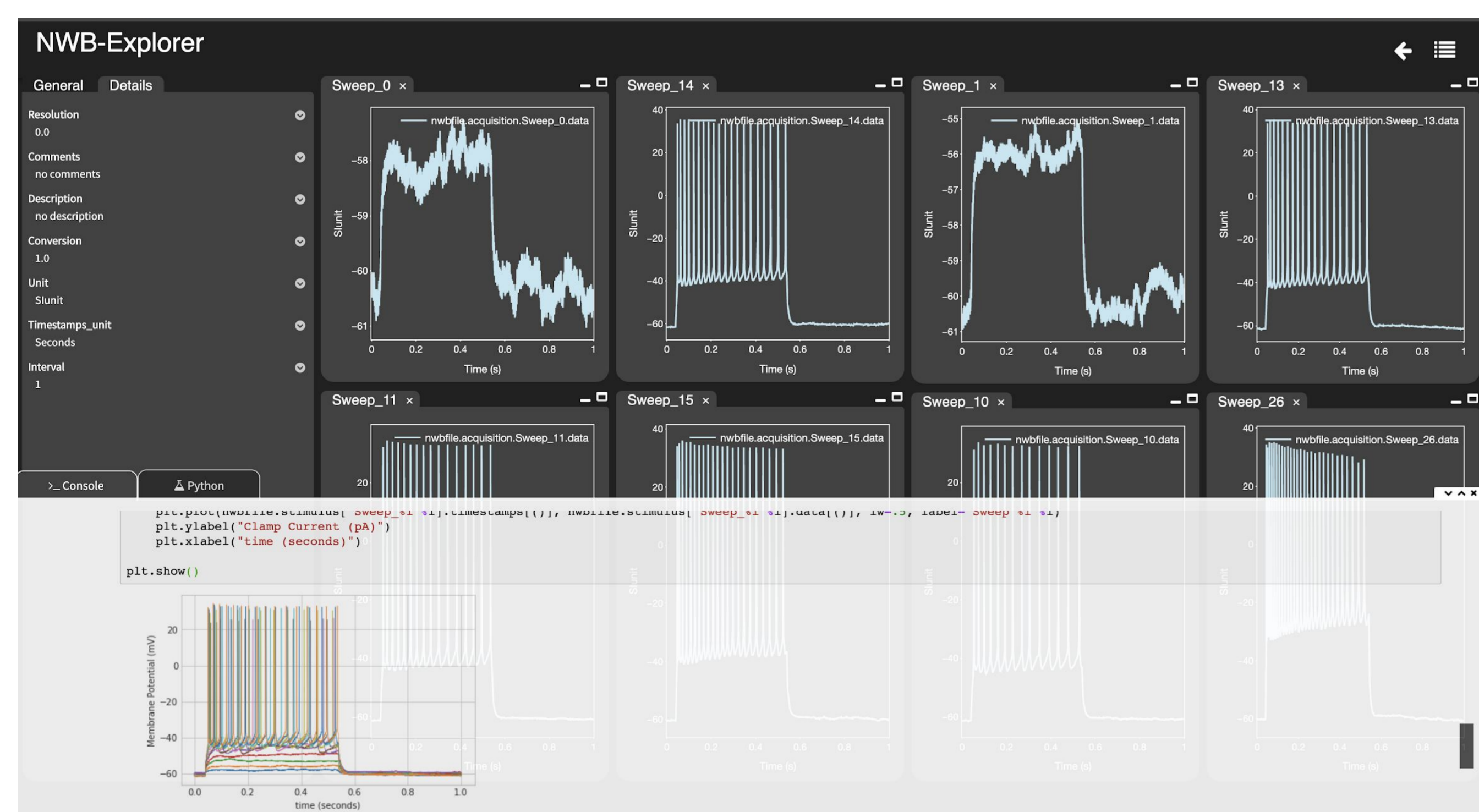


Figure 3. NWB Explorer as an example of an application currently being developed using Python Geppetto. The Neurodata Without Borders (NWB) [6] file is loaded using PyNWB and it is represented in memory via a Geppetto model which enables reusing all Geppetto components to represent the data within the files. The NWB file that was loaded by the graphical interface is also accessible programmatically through an integrated Jupyter notebook allowing advanced users to perform arbitrary analysis.

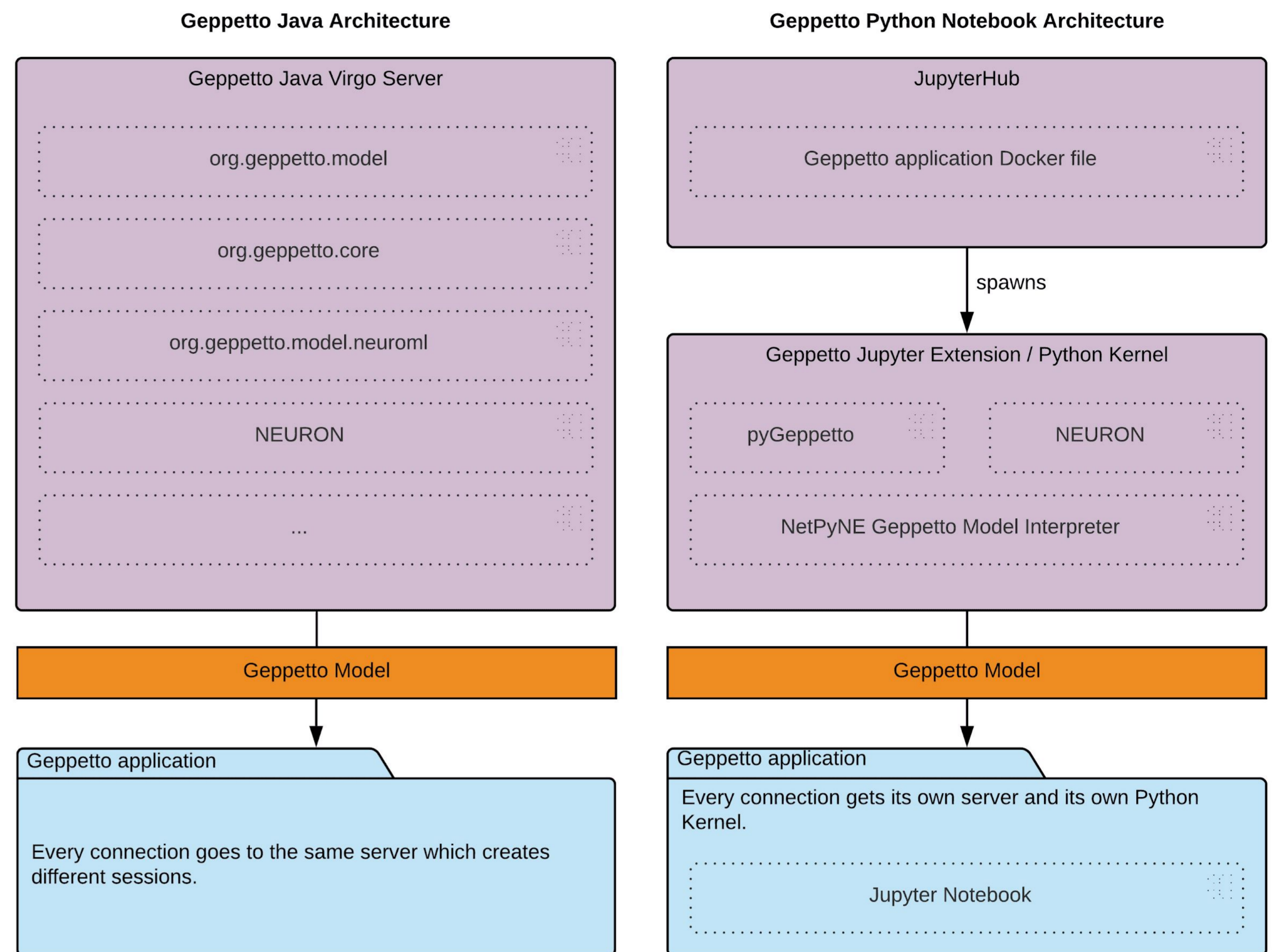


Figure 4. Comparison of the Geppetto Java architecture and the Geppetto Jupyter Python one. In Java a Tomcat server hosts all the backend bundles and accepts connections from multiple clients. In Python JupyterHub is responsible for spawning (e.g. creating a new Kubernetes pod) a new Python Kernel from a Geppetto application Docker file. Each client will have dedicated resources and direct access to the memory of the Kernel. In both scenarios a Geppetto Model is exchanged between the server and the client. Additionally, the application code is mostly independent of the architecture, which makes migrating an application from one stack to the other a fairly easy task.

User interface Synchronization

We developed a series of *JavaScript-Python Connectors* that let developers easily build a user interface, whose state can be controlled from a Python model and vice versa. These connectors are Higher-Order React components (<https://reactjs.org/>) capable of extending any standard React control (e.g. TextBox, CheckBox, Dropdown, etc.) to link the underlying value of the control to a Python model of choice. Once these components will connect to the Python Kernel any changes made on the user interface (e.g. ticking a checkbox) will automatically affect the connected Python variable synchronising the value from the UI. In the same way any programmatic changes made to the variable using the embedded Jupyter Notebook will automatically be reflected on the connected component.

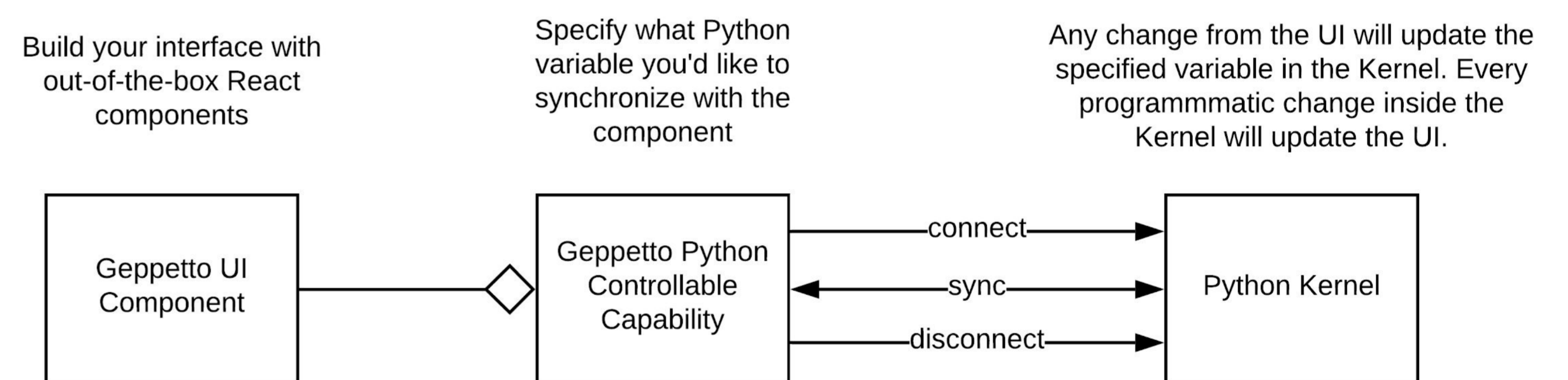


Figure 5. Component diagram behind Geppetto Python user interface synchronisation. A React higher-order component (middle) extends a Geppetto UI component. The controllable capability adds to the component the ability to connect, disconnect and be in sync with a specified Python variable.

Neuroscience applications built with Python Geppetto have the advantage of bridging the beginner/advanced user usability gap. Beginners will be able to interact with a user interface that will simplify the accessibility of the underlying APIs. Expert users will be able, from the same GUI, to programmatically interact with the underlying data models and Python APIs while the user interface will be kept updated graphically reflecting any programmatic changes. Three neuroscience applications are already being built using Python Geppetto: NetPyNE-UI (Figure 1), Human Neocortical Neurosolver UI (Figure 2) and NWB Explorer (Figure 3). All three applications use jupyterGeppetto, a Jupyter Notebook extension we developed to enable fully custom web applications to run inside a notebook environment.

Applications built using jupyterGeppetto have, on top also of all the advantages of a standard Geppetto application, a ready to use set of components (e.g. 3D Canvas, Control Panel, etc.), a predefined Meta-model capable of representing arbitrary neuroscience data and models which are instantiated at runtime and a backend capable of running asynchronous simulations using external simulators (e.g. NEURON), connecting to external data sources (e.g. Neo4j) and persist user data. **For more information about Geppetto please refer to the 2018 publication in Philosophical Transactions of the Royal Society B, available at <http://paper.geppetto.org>.**

References

- Cantarelli M et al. "Geppetto: a reusable modular open platform for exploring neuroscience data and models." *Philosophical Transactions of the Royal Society B: Biological Sciences* 373.1758 (2018): 20170380.
- Gleeson, P. et al. (2019). "Open Source Brain: A Collaborative Resource for Visualizing, Analyzing, Simulating, and Developing Standardized Models of Neurons and Circuits." *Neuron* 103, 1-17.
- Armstrong JD et al. "Towards a virtual fly brain." *Philos. Trans. R. Soc. A* 367, 2387-2397. (doi:10.1098/rsta.2008.0308)
- Dura-Bernal, S. et al. (2019). NetPyNE, a tool for data-driven multiscale modeling of brain circuits. *eLife* 8, e44494.
- Neymotin S et al. "Human Neocortical Neurosolver (2018)" (doi:10.5281/zenodo.1446517)
- Jeffery L. Teeters et al. "Neurodata Without Borders: Creating a Common Data Format for Neurophysiology" *Neuron*, 88, Issue 4, 2015, 629-634, (doi:10.1016/j.neuron.2015.10.025)

Acknowledgements: This work was supported by Wellcome Trust 101445, NIH U01EB017695, NIH R01EB022889, NYS DOH 01-C32250GG-3450000

 <http://geppetto.org>

 <http://git.geppetto.org>

 @geppettoengine

Join Us!