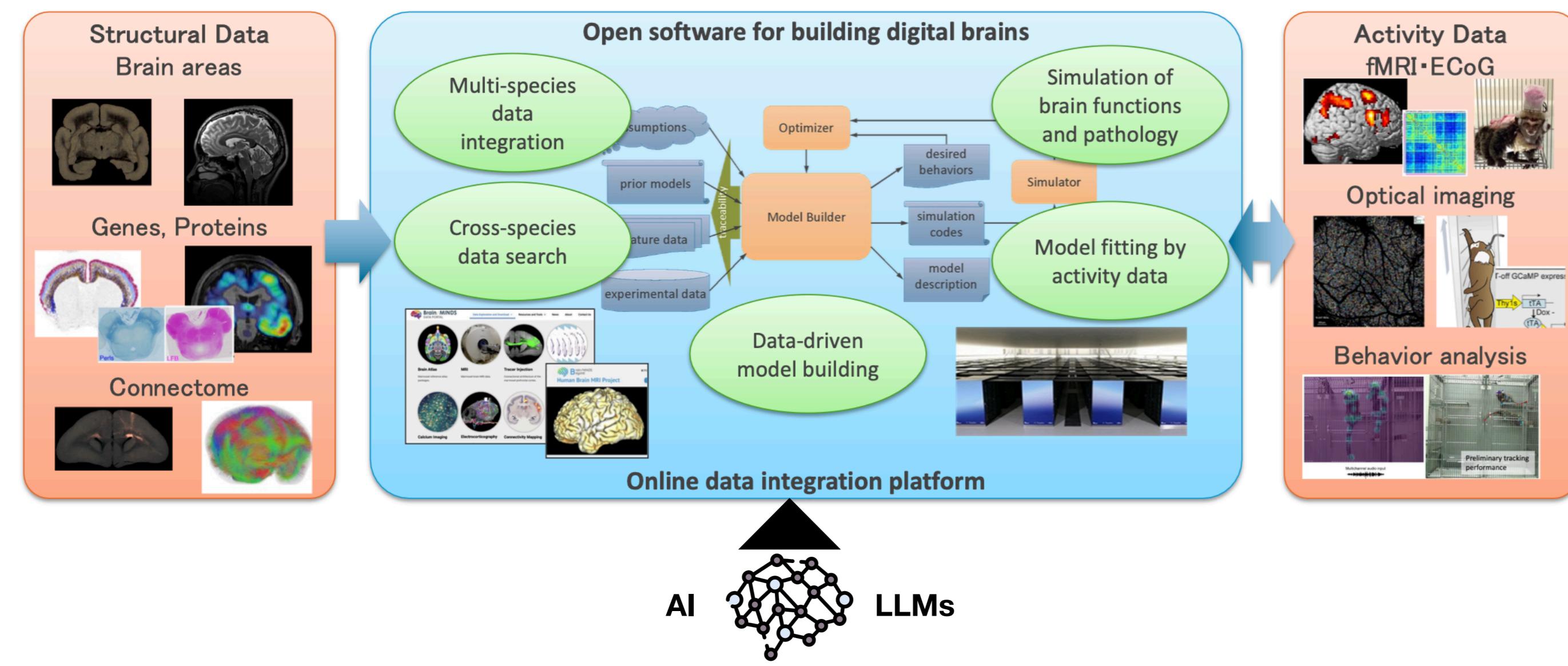


Carlos Enrique Gutierrez(1), Henrik Skibbe(2), Gaganpreet Singh Jhajj(3), Yukako Yamane(1), Kenji Doya(1).

(1) Okinawa Institute of Science and Technology Graduate University, Japan. (2) RIKEN CBS, Japan. (3) Athabasca University, Canada

Brain/MINDS 2.0 - The Digital Brain Project

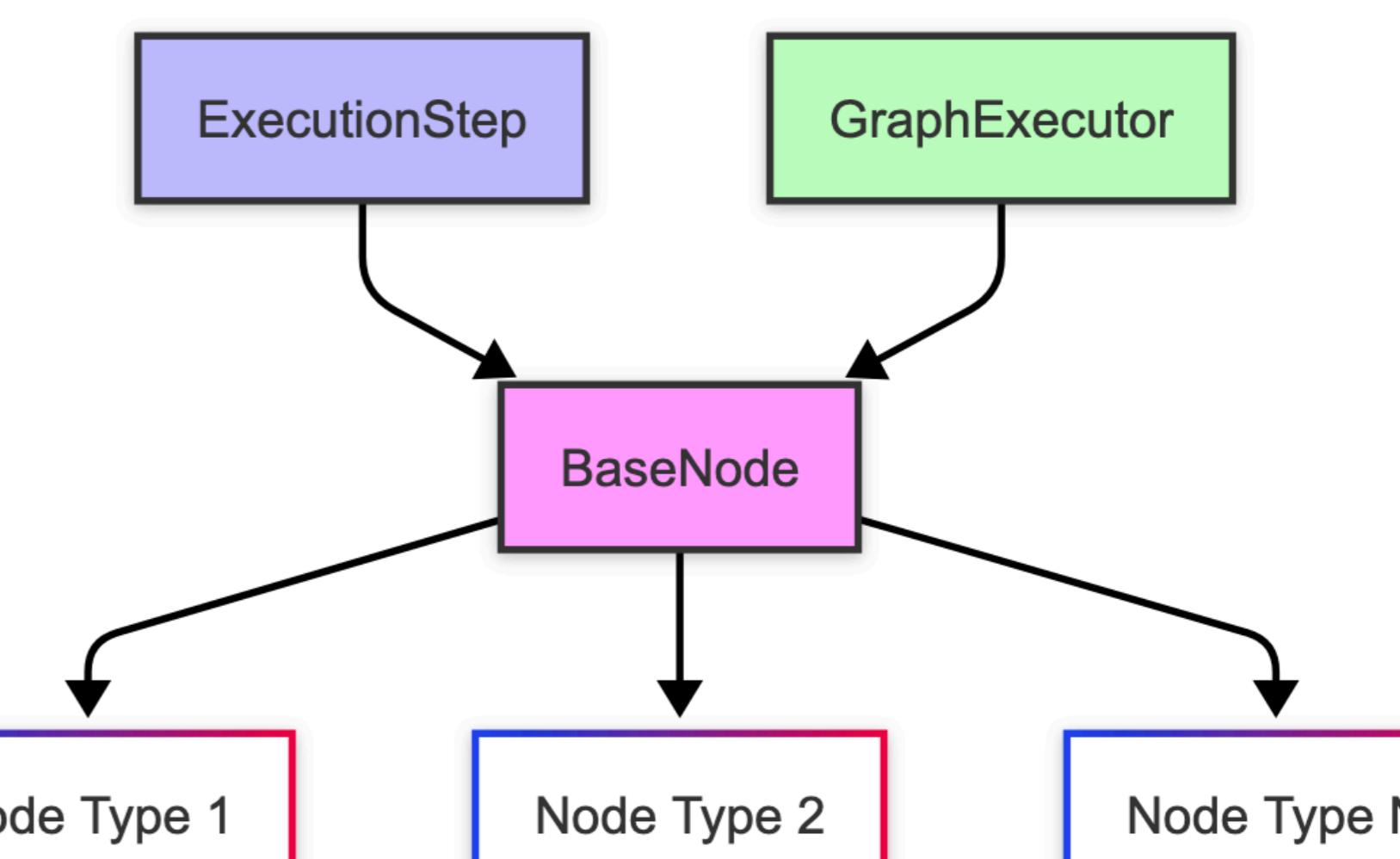
Goal: open software for building digital brains
-online platform for data integration, model building and simulations-



The Digital Brain as a graph-based framework

- ▶ The Brain/MINDS 2.0 Digital Brain aims to create a framework that supports interoperability with popular neuroscience tools such as TVB, NEST, NEURON, and others, leveraging their Python interfaces.
- ▶ Modern brain modeling and simulation workflows often require complex sequences of data processing, model configuration, and analysis steps. While powerful tools exist, they frequently demand advanced programming knowledge, limiting accessibility and collaboration potential.
- ▶ We present a graph-based framework that transforms complex scientific workflows into modular, reusable components through a node-edge architecture.

Framework Architecture



Core Components:

ExecutionStep Class:

defines individual execution steps within nodes

BaseNode Class:

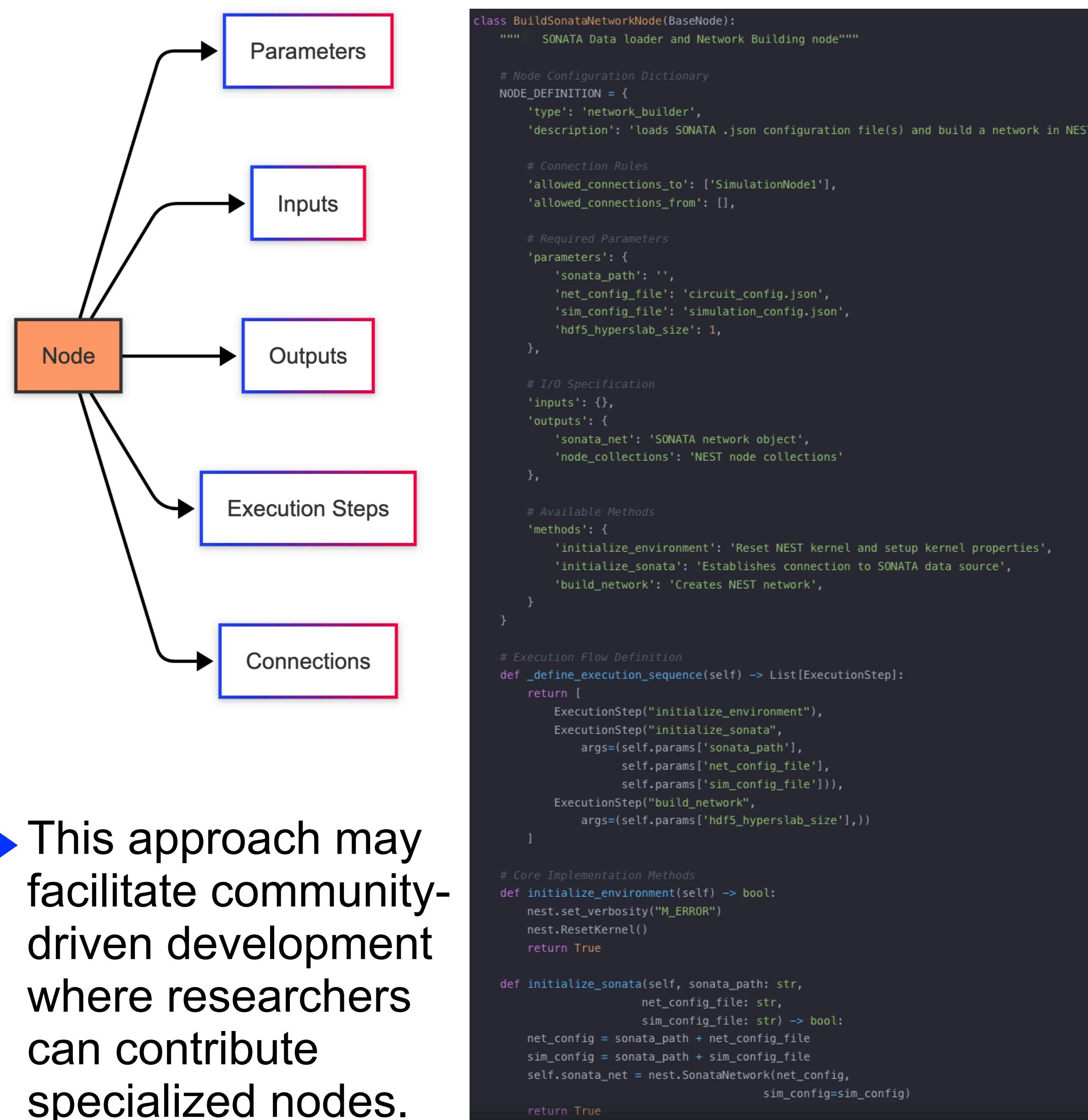
template for all workflow nodes

GraphExecutor Class:

manages workflow execution

Node Structure

- ▶ Our framework enables researchers to encapsulate scientific procedures as self-documenting nodes with standardized interfaces, clear input/output specifications, and automated data flow management.

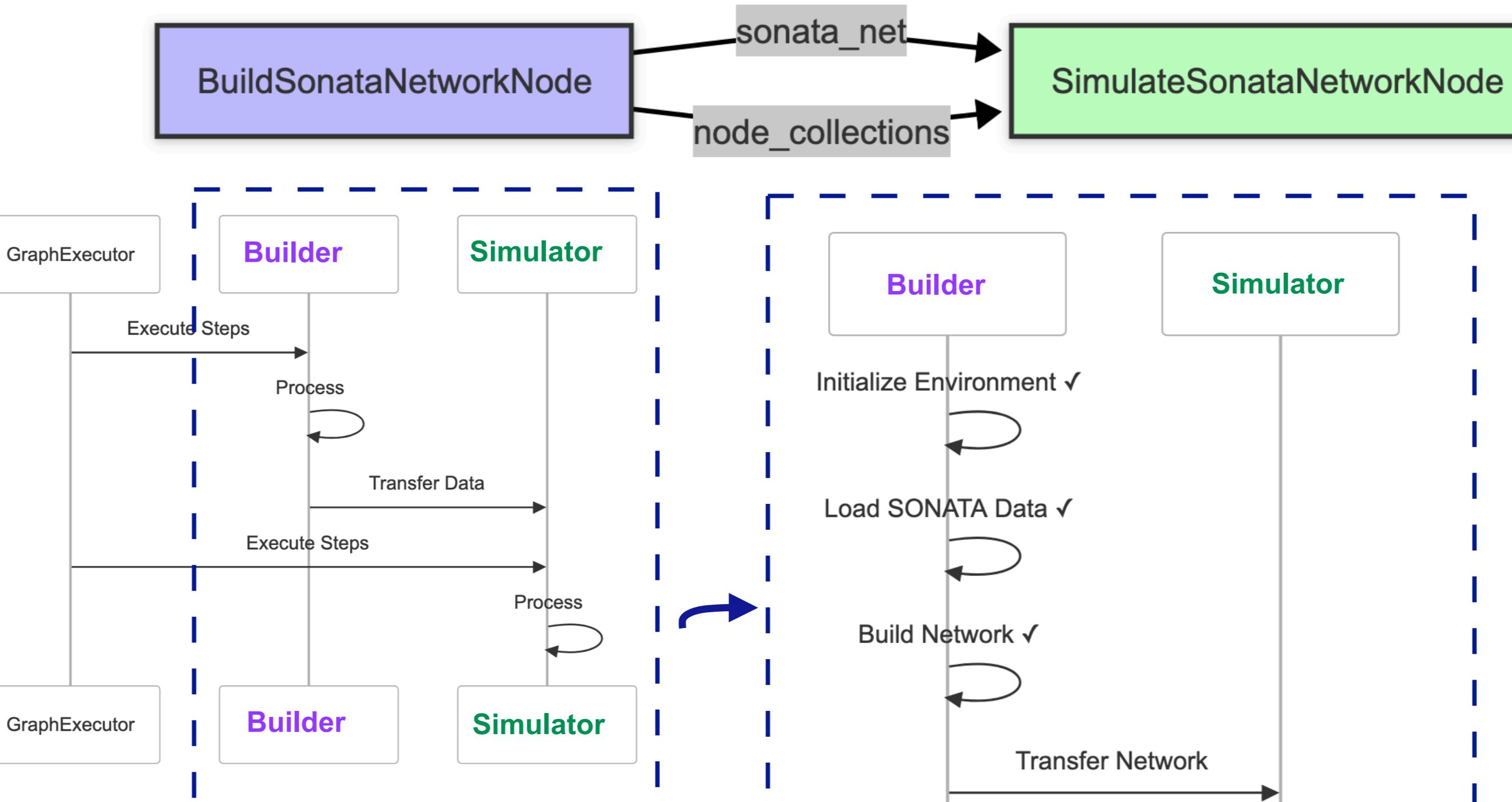


- ▶ This approach may facilitate community-driven development where researchers can contribute specialized nodes.

WorkFlow Example

- ▶ This proof-of-concept demonstrates the framework's potential through a **SONATA-NEST** workflow implementation

BuildSonataNetworkNode: Network construction from SONATA files.



BuildSonataNetworkNode

Parameters:

- SONATA path: .../300_neurons/
- Config files: circuit & sim
- hdf5 hyperslab size: 1M

SimulateSonataNetworkNode

Parameters:

- Simulation time: 1000.0 ms
- Recording: 200 neurons

Example of a Data Analysis Node

```

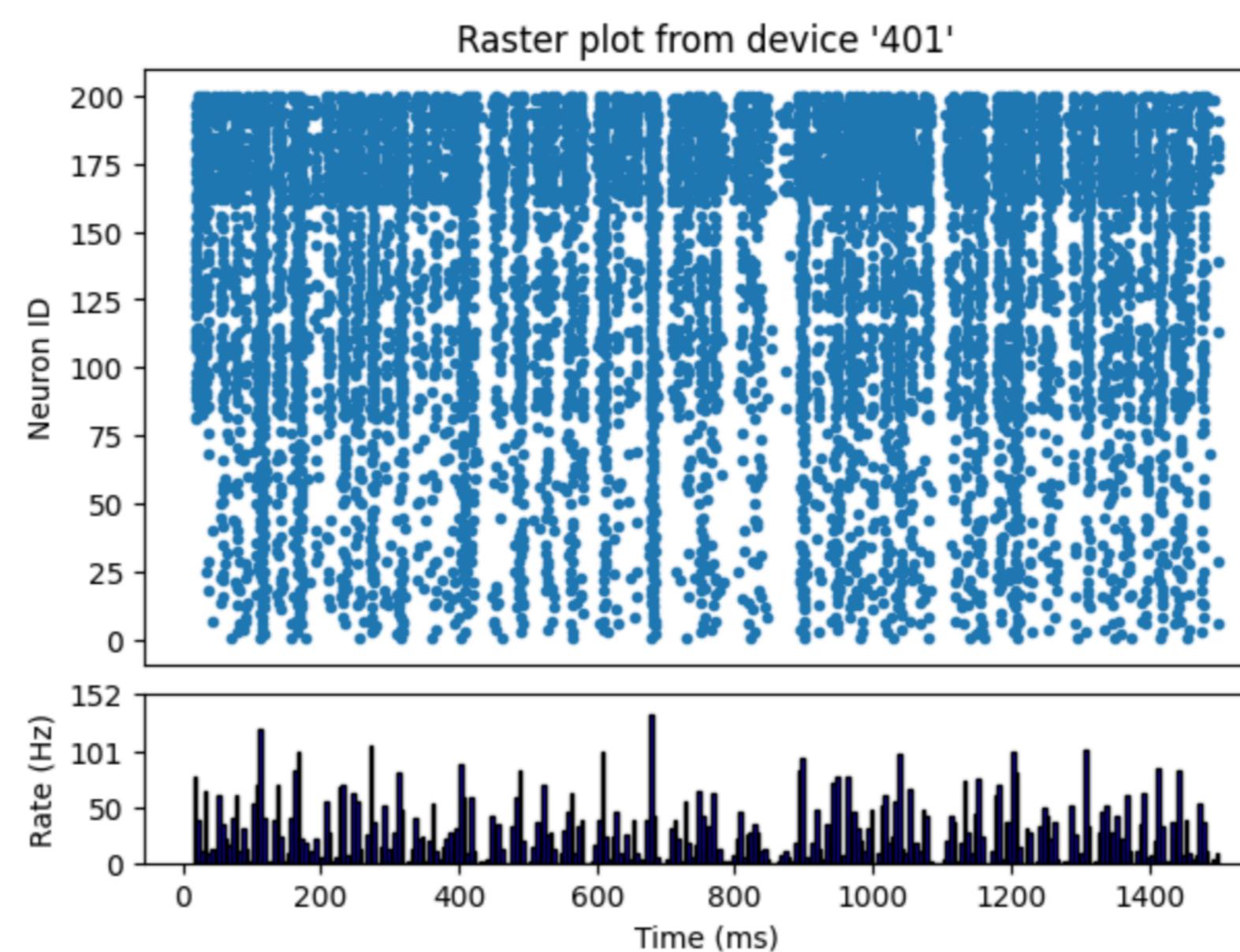
class DataAnalysisNode(BaseNode):
    """ Statistical Analysis Node"""

    # Node Configuration Dictionary
    NODE_DEFINITION = {
        'type': 'analysis',
        'description': 'Performs statistical analysis',
        'parameters': {
            'analysis_type': 'basic',           # Type of statistical analysis
            'confidence_level': 0.95          # Statistical confidence level
        },
        # I/O Specification
        'inputs': {
            'data': 'Input dataset'
        },
        'outputs': {
            'results': 'Analysis results'
        }
    }

    # Analysis Pipeline Definition
    def _define_execution_sequence(self):
        return [
            ExecutionStep("prepare_data"),
            ExecutionStep("run_analysis"),
            ExecutionStep("generate_report")
        ]

```

Simulation Output (200 neurons)



This framework may enable:

- Rapid development of scientific workflows
- Easy sharing of analysis components
- Reproducible research practices
- Collaborative development
- Clear documentation
- Quality control

Conclusions and Future Work

This modular node-based architecture provides:

- ◆ Flexibility: Each node is a self-contained unit with clear I/O specifications
- ◆ Reproducibility: Standardized execution sequences ensure consistent workflow processing
- ◆ Extensibility: New nodes can be easily added by inheriting from `BaseNode`
- ◆ Interoperability: Structured data flow between brain modeling tools
- ◆ Maintainability: Clear separation between node definition and implementation

Future development will include a web-based graphical interface, enabling scientists to create and modify workflows through drag-and-drop interactions.

Acknowledgments: This work was supported by Brain/MINDS 2.0 project (AMED), Development of the "digital brain" and related research platforms utilizing mathematical models.