

# MANAGING JUPYTER KERNELS ON RCAC CLUSTERS

Geoffrey Lentner

Lead Research Data Scientist

# About Me

## Data Scientist, Astrophysicist, Research Software Engineer

Things I do at the Rosen Center for Advanced Computing (RCAC)

- **System Support**
- **Scientific Software**
- **Consulting**
- **Training and Teaching**
- **Outreach and Engagement**
- **Innovation**

## Things we assume you already know

- Linux command-line basics ([Unix 101](#) and [Unix 102](#))
- Cluster basics ([Clusters 101](#))
- Python basics (not critical)

## Topics covered

- What is Jupyter
- What is a kernel and where does Jupyter look for them
- Anatomy of a kernel and how to customize them
- Extra topics

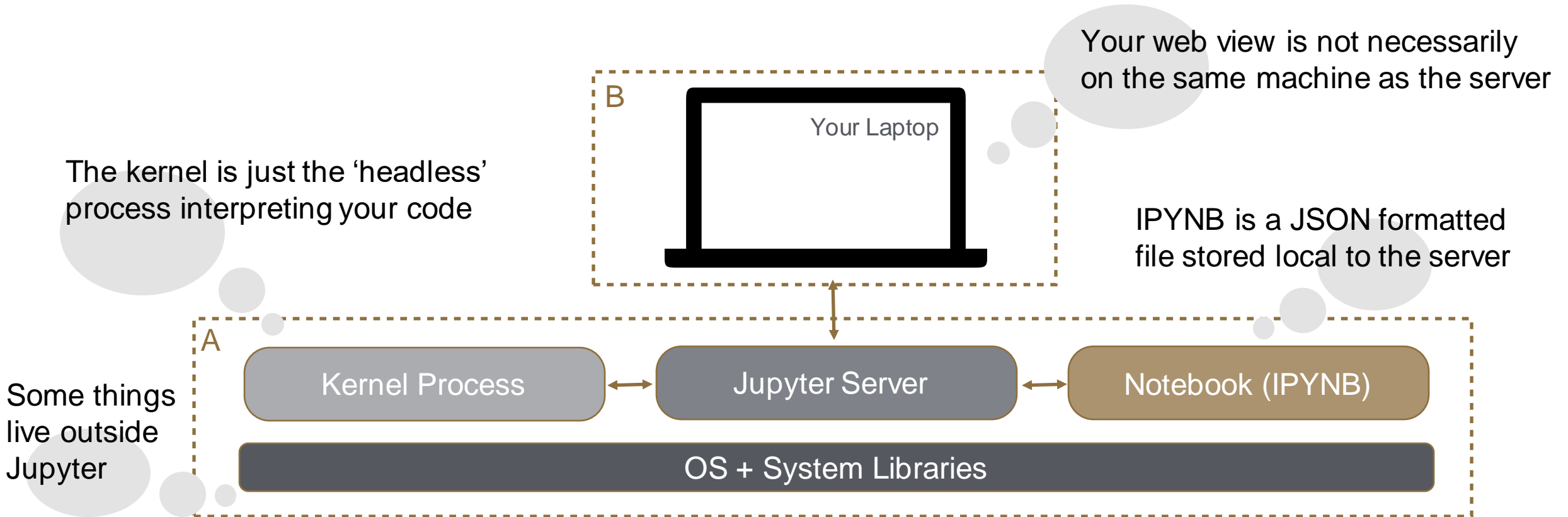
# 1

## What is Jupyter

What are the components that make up the Jupyter ecosystem?

# What is Jupyter

## Elements of the Jupyter system



# 2

## What is a Kernel

The runtime environment specification that underpins Jupyter notebooks.

# What is a kernel?

A Jupyter “kernel” can refer to two things

- The “kernel spec” or **configuration file**
- The **runtime** environment it refers to (e.g., Anaconda)



# What is a kernel?

## Where does Jupyter look for kernels?

```
jane@login00.cluster [~] $ tree ~/.local
.
├── bin/
├── etc/
├── include/
├── lib/
├── var/
├── share/
├── jupyter/
├── kernels/
├── py39-foo/
├── ...
└── kernel.json
```

- System: /usr/local/share/jupyter/kernels
- User: ~/.local/share/jupyter/kernels
- Local: ../share/jupyter/kernels

Within the same installation  
**prefix** as Jupyter itself

# 3

## Anatomy of a Kernel

What's inside a kernel specification?

# What's inside a kernel specification

Take a look inside the *kernel.json* file

```
jane@login00.cluster [~] $ cat ~/.local/share/jupyter/kernels/py39-foo/kernel.json
{
  "argv": [
    "/home/jane/.conda/envs/cent7/2020.11-py38/py39-foo/bin/python",
    "-m",
    "ipykernel_launcher",
    "-f",
    "{connection_file}"
  ],
  "display_name": "Python 3.9 (Foo)",
  "language": "python"
}
```

The “**argv**” section is literally the command-line arguments that will be *invoked on your behalf*.

# What's inside a kernel specification

Take a look inside the *kernel.json* file

```
jane@login00.cluster [~] $ cat ~/.local/share/jupyter/kernels/py39-foo/kernel.json
{
  "argv": [
    "/home/jane/.conda/envs/cent7/2020.11-py38/py39-foo/bin/python",
    "-m",
    "ipykernel_launcher",
    "-f",
    "{connection_file}"
  ],
  "display_name": "Python 3.9 (Foo)",
  "language": "python",
  "env": {
    "PROJ_HOME": "/home/jane/.conda/envs/cent7/5.3.1-py37/my_env/share/proj"
  }
}
```

You can do things like customize environment variables within the kernel specification

# 4

## Special Topics

What other things can you do with Jupyter?

Alternative configurations?

How to trouble shoot when things goes wrong?

# Troubleshooting

## How to find issues when Jupyter/kernels aren't working

```
jane@login00.cluster [~] $ module load jupyterhub  
jane@login00.cluster [~] $ jupyter notebook --port 8787 --no-browser  
...
```

Errors from Jupyter itself as well as anything to do with the running kernel will appear in the logs

## We can discuss many other things

- Non-Python kernels
- Integrating Modules with Jupyter Notebooks
- Distributed Computing within notebooks
- Notebook extensions
- Debugging Jupyter, Notebooks, Kernels
- OnDemand and Jupyter
- Running custom Jupyter on compute nodes
- Git and Jupyter
- Jupyter Lab
- Notebook size and visualizations

# ***THANK YOU***

Please reach out to [rcac-help@purdue.edu](mailto:rcac-help@purdue.edu) for questions.