

The OOD App Ecosystem

Ideas for Enhancement and Collaboration

The WFU HPC Team[†]:

Sean M. Anderson¹, Adam Carlson¹, Cody Stevens^{1,2}

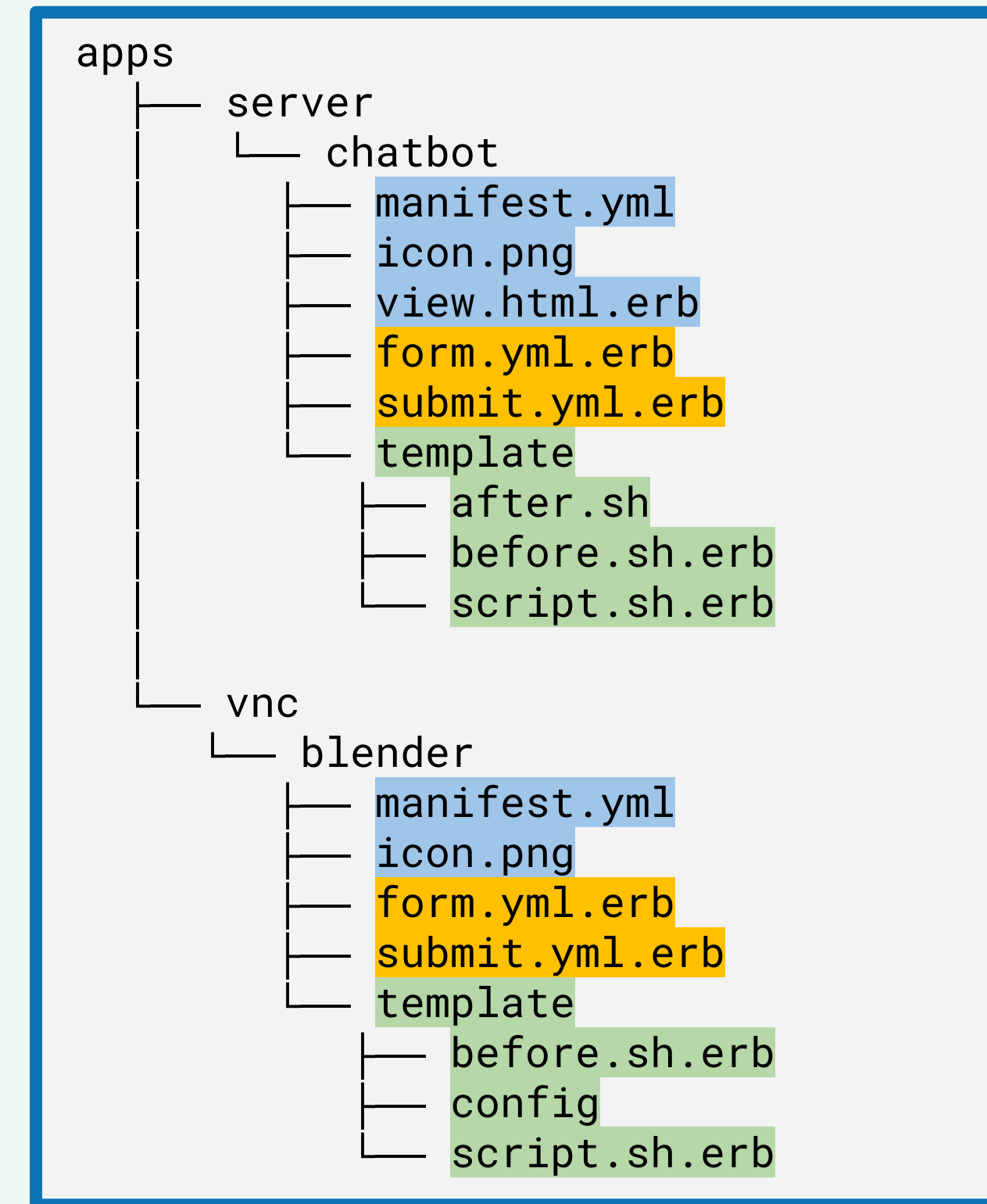
¹ Information Systems, Wake Forest University

² Department of Computer Science, Wake Forest University

[†] Correspondence: anderss@wfu.edu



Idea 1: App Centralization



Most apps are comprised of a similar set of files that can be categorized as follows:

- **manifest** and **icon** files are for organizational and aesthetic purposes. They do not change the functionality of the application – only how they appear in the dashboard. For server applications, the **view** file contains the connection specifics so the user can connect; It rarely changes once configured.
- The **form** and **submit** files are **site-specific** to your infrastructure. In most installations, they will be very similar across all applications. These two files determine how your user interacts with your application, and in turn, how it interacts with your infrastructure.
- The files in the **template** directory are **application-specific** and contain the scripts and configuration framework to run the application. These will be dependent on how the underlying software is delivered (system packages, environment modules, etc.) and on the base operating system.

If sites provide their own **form** and **submit** files, apps could be as easy to distribute as this:

```

$ ood_app remote list

matlab      osc      vnc      rocky9
vmd          osc      vnc      rocky9
visit       osc      server   rocky9
matlab_server osc      server   rocky9
jupyter_spark osc      server   rocky9
jupyter     osc      server   rocky9
ansys_workbench osc      vnc      rocky9
rstudio_server osc      server   rocky9
pymol       osc      vnc      rocky9

$ ood_app remote install jupyter
    
```

Idea 2: Per-profile App Customization

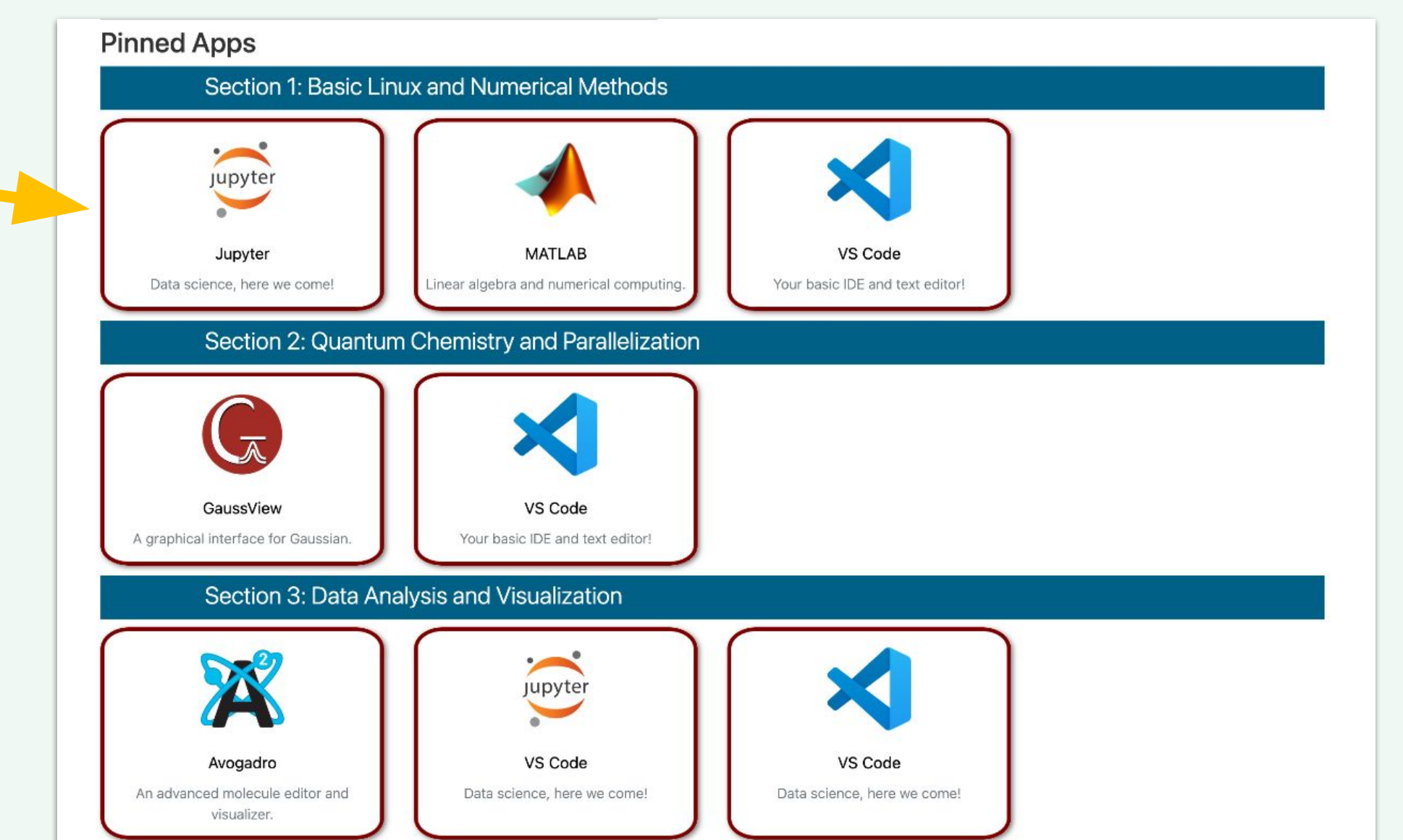
Profiles offer the perfect framework for creating custom portals for classes, user groups, departments, or even for development environments. Customizing individual applications for different profiles requires making a complete copy of each app – even if you are only changing a few options in the form – which can add extra administrative overhead.

What we propose here would allow you to customize app settings at the **profile level** without needing to manage individual files.

As shown in the mockup, we can add a system app to a profile, and change specific items in the form and manifest. This would allow for unlimited customization of both the appearance and launch settings, without the need to copy or modify existing apps.

```

apps:
  jupyter1:
    - path: "sys/jupyter"
    - attributes:
        accounts: "hpc472"
        num_cores: 1
        memory: 8
        bc_num_slots: 1
        bc_num_hours: 2
        queues: "login"
        node_type: "Any"
    - manifest:
        category: "Section 1: Basic Linux and Numerical Methods"
        title: "Jupyter"
        border_color: "maroon"
        sub_caption: |
          Data science, here we come!
    
```



Idea 3: Smarter App Metadata

If applications can be distributed and customized more readily following the ideas above, then we should also consider how to add meaningful and rich **metadata** that can be leveraged for organization and management purposes.

We envision that the metadata fields could contain different data structures (lists, key-value pairs) that can be iterated over or keyed-off of. Some of the metadata fields be **organizational**, such as:

- Fields of science with various entries
- Keywords
- Descriptors for software environments (compilers, math libraries, provenance)
- Descriptors for operating system or window managers

```

metadata:
  testing: False
  hidden: False
  type: server
  field_of_science:
    - "statistics"
    - "computer science"
    - "physics"
  groups:
    - csc191
    - sta379
    - phy262
  profiles:
    - class_sta379:
        - "Section 1"
        - "Homework 3"
  keywords:
    - jupyter
    - python
    - "numerical methods"
    
```

Other metadata field could be **managerial**, such as:

- Testing: enabling an app to be only available for administrators
- Hidden: hiding an app from all views, but enabled in certain menus or contexts
- Groups: define which POSIX groups can use the app
- Profiles: define which profiles can show the app

If your cluster has hundreds of applications, libraries, languages, and tools, think about how you can make them more accessible for your users to find and use!

About Us

The **DEAC Cluster** is Wake Forest University's supercomputer, which prioritizes equity, economy, and innovation with big processing power for students and faculty in all departments. The cluster is:

- a supercomputer composed of 100 servers;
- a free resource for all WFU researchers, including students, faculty, and staff;
- managed by a dedicated team of research computing engineers;
- an ever evolving environment to meet the needs of researchers across the WFU campus.



500
Active Accounts



12 Million
Tasks (2020-Present)



100 Million
Core-Hours (2020-Present)



17
Departments

Follow-up

Interested in discussing more?

While still early in our journey with Open OnDemand, we have already begun integrating it into classrooms and into more general research workloads. Unsurprisingly, our users have embraced the many new capabilities OOD has to offer.

We are always looking for ways to enhance our services and infrastructure, and share our work with the broader OOD community. **Please reach out if any of these ideas resonate with you!**

