

# Introduction to pyProcserv



Canadian  
Light  
Source    Centre canadien  
de rayonnement  
synchrotron

THE BRIGHTEST LIGHT IN CANADA | [lightsource.ca](http://lightsource.ca)

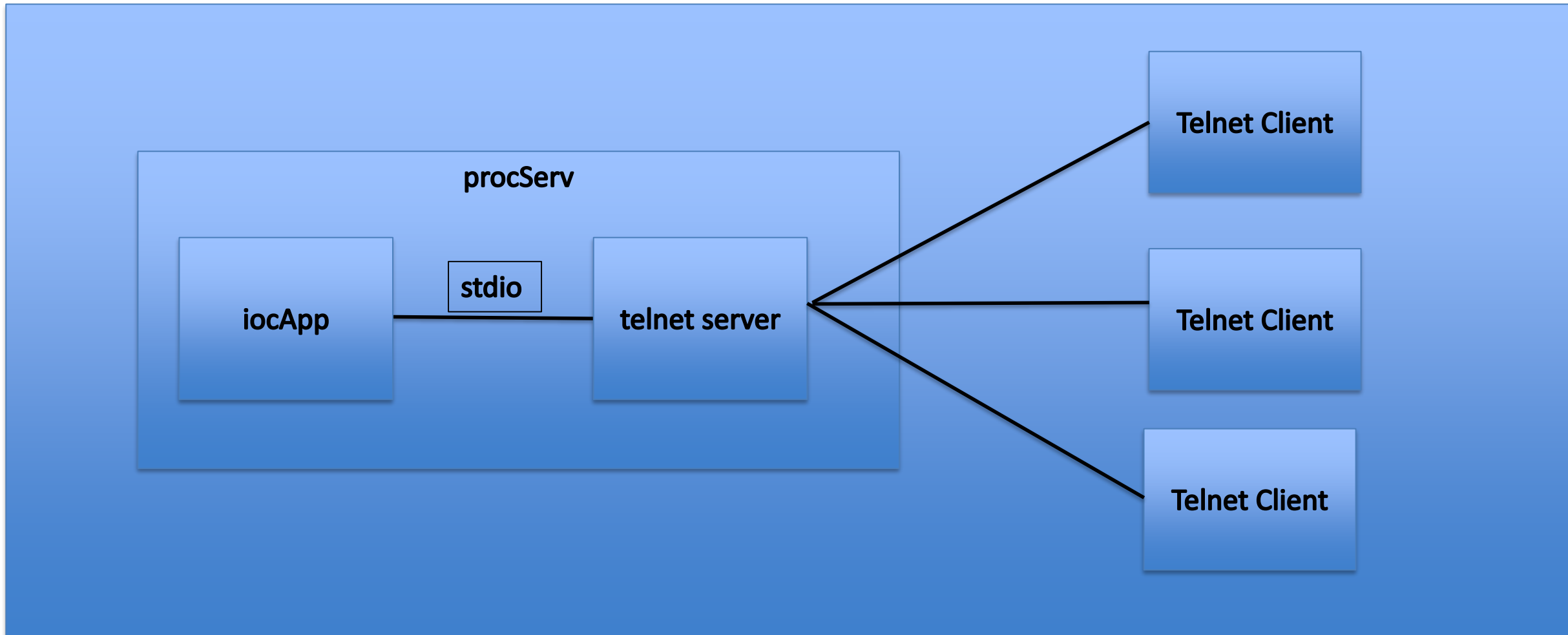


# pyProcServ

- Python-based
- Platform-Independent
- Plug-in replacement for existing procServ



# procServ (Recap)



# Opportunities

- Auto-discover (respond to broadcasts)
- Broadcast server presence (beacons)
- Authentication
- Different protocols
  - SSH
  - SSL/TLS
  - Websockets (a la Jupyterlab/Terminado)

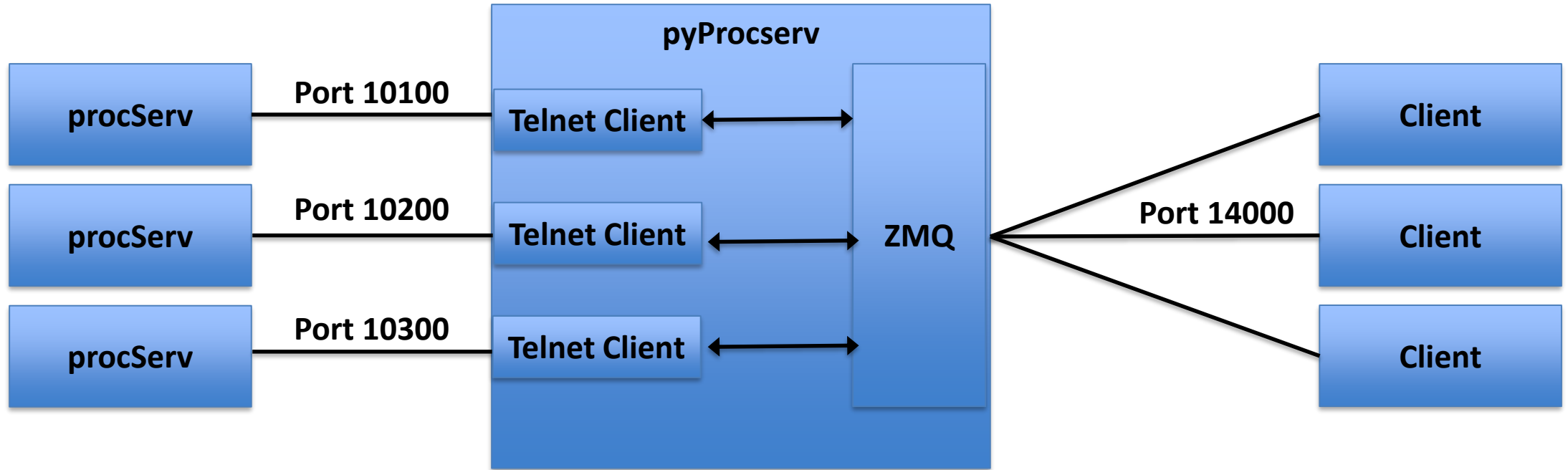


# Opportunities

- procServ “concentrator”
  - Task to watch for existing procServ installations
  - Connect to them via a telnet client
  - Serve those through a single port via ZMQ or other means
  - High-level menu:
    - Allow client to select one or more procSers
    - Dynamically add/delete subprocesses to serve (remotely)



# procServ Concentrator



# Status of ~~Introduction to~~ pyProcserv



Canadian  
Light  
Source    Centre canadien  
de rayonnement  
synchrotron

THE BRIGHTEST LIGHT IN CANADA | [lightsource.ca](http://lightsource.ca)



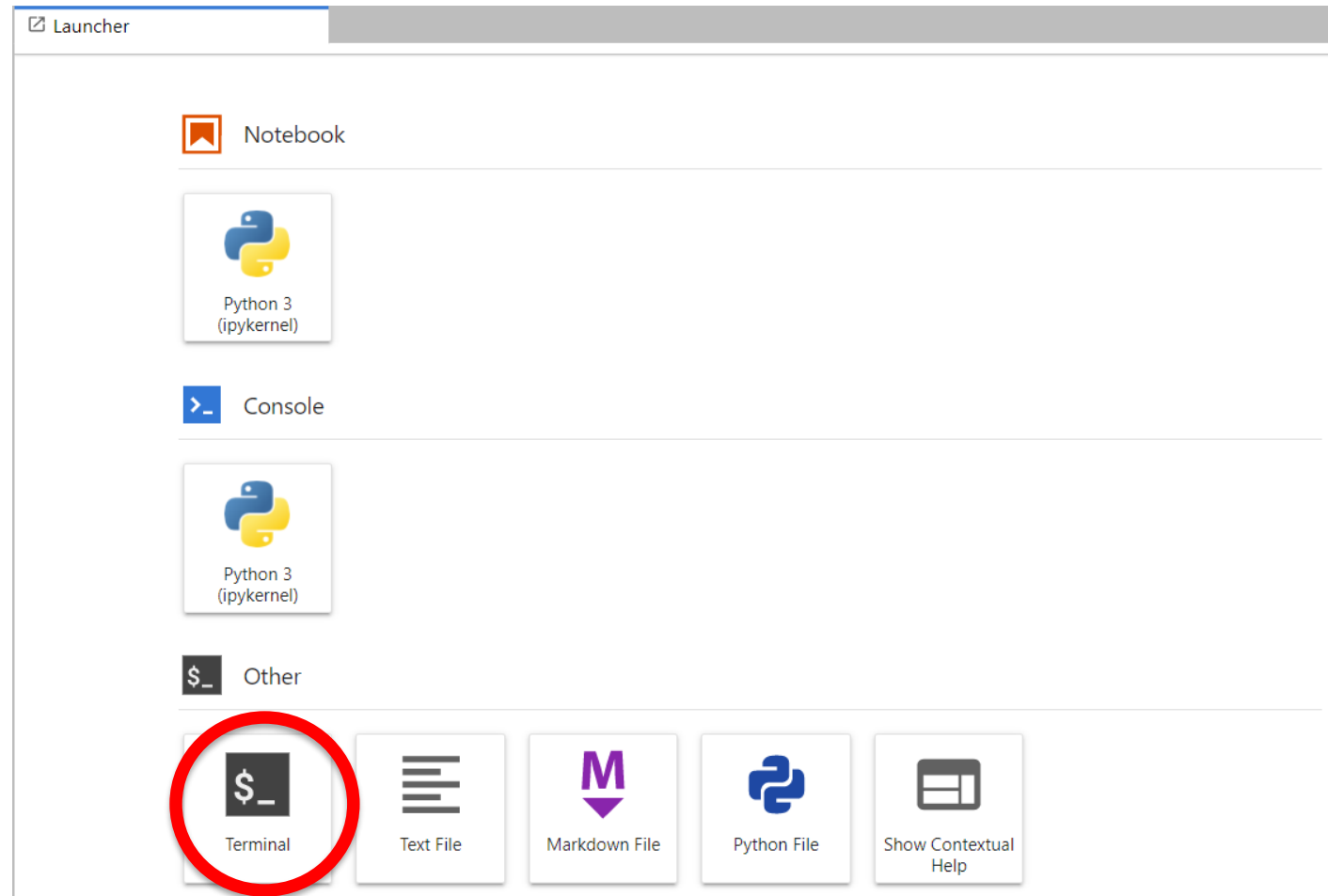
# Jupyter/Terminado

- Terminado does exactly what I want (and more) over websockets.
  - Uses Tornado/Asyncio (Server)
    - The Tornado 6.0 event loop is now a wrapper around asyncio's
  - Xterm.js (client)
  - Seemed a good fit

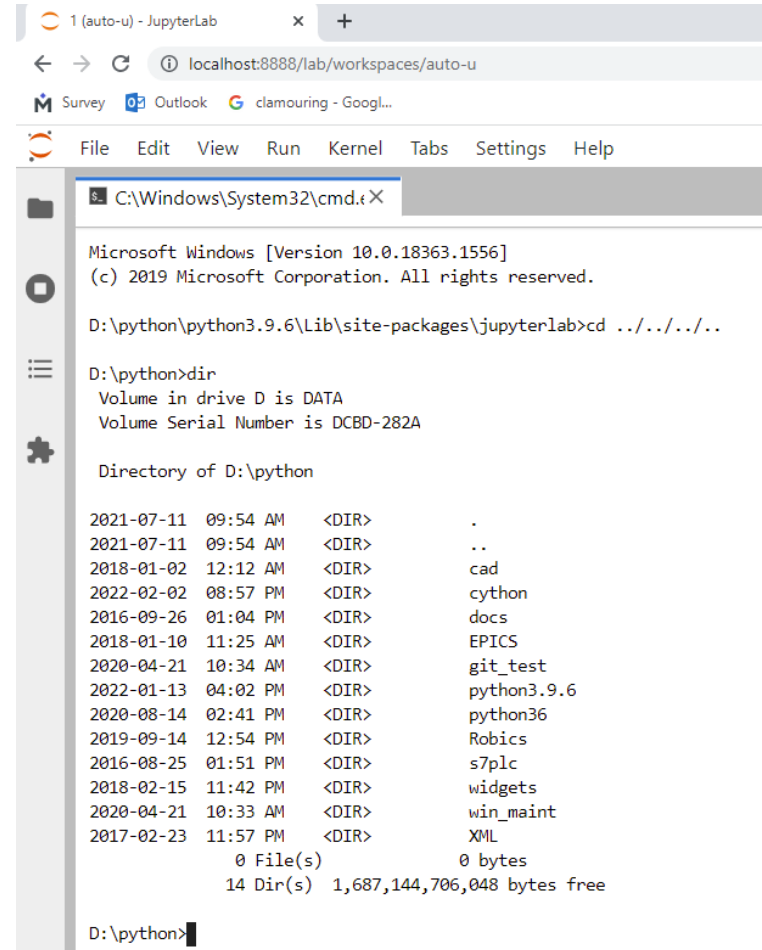




# Jupyter/Terminado (Main Screen)



# Jupyter/Terminado (Win Terminal)



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\python\python3.9.6\Lib\site-packages\jupyterlab>cd ../../../../

D:\python>dir
Volume in drive D is DATA
Volume Serial Number is DCBD-282A

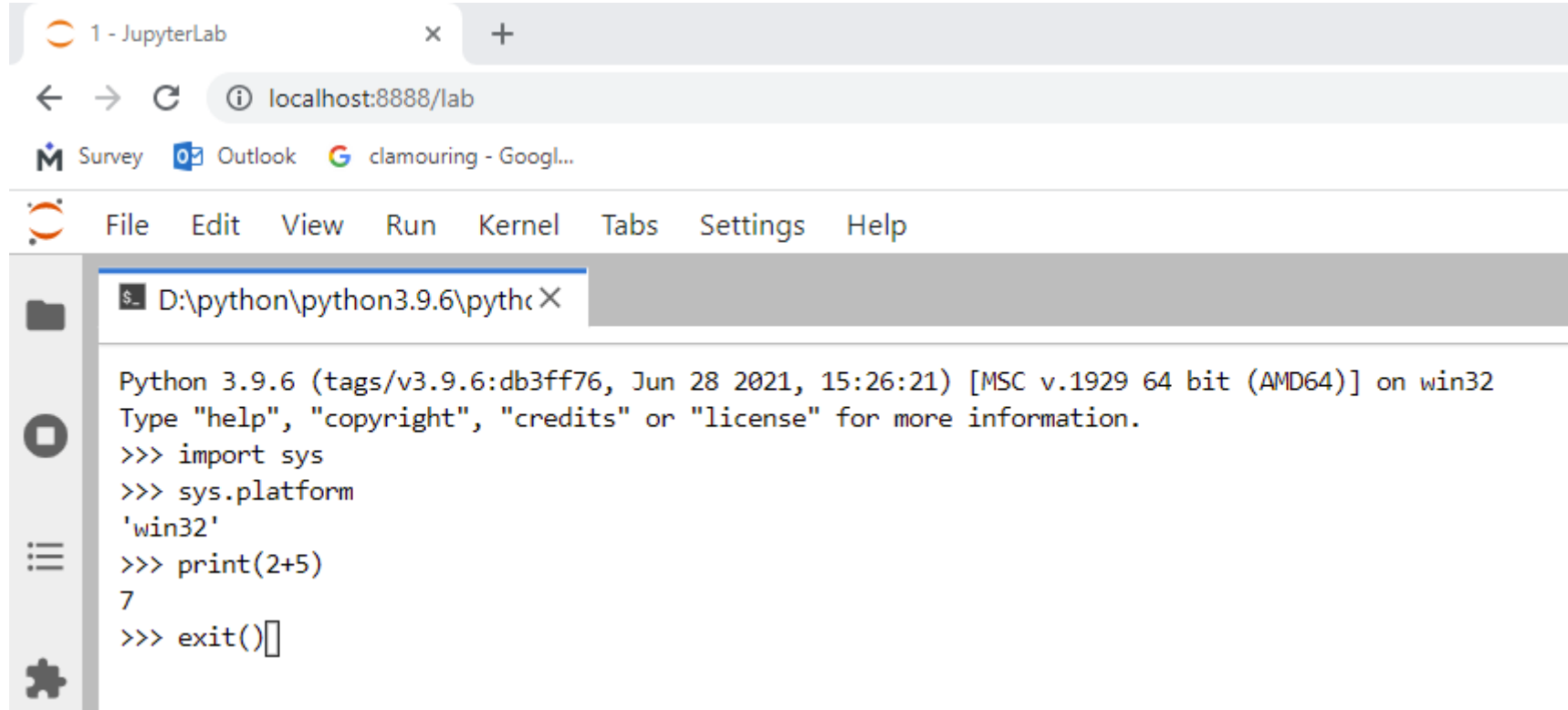
Directory of D:\python

2021-07-11 09:54 AM <DIR>      .
2021-07-11 09:54 AM <DIR>      ..
2018-01-02 12:12 AM <DIR>      cad
2022-02-02 08:57 PM <DIR>      cython
2016-09-26 01:04 PM <DIR>      docs
2018-01-10 11:25 AM <DIR>      EPICS
2020-04-21 10:34 AM <DIR>      git_test
2022-01-13 04:02 PM <DIR>      python3.9.6
2020-08-14 02:41 PM <DIR>      python36
2019-09-14 12:54 PM <DIR>      Robics
2016-08-25 01:51 PM <DIR>      s7plc
2018-02-15 11:42 PM <DIR>      widgets
2020-04-21 10:33 AM <DIR>      win_maint
2017-02-23 11:57 PM <DIR>      XML
                0 File(s)          0 bytes
                14 Dir(s)  1,687,144,706,048 bytes free

D:\python>
```



# Jupyter/Terminado (Python Console)



The screenshot shows a web browser window with a single tab titled "1 - JupyterLab". The address bar displays "localhost:8888/lab". Below the browser window is the JupyterLab interface, which includes a menu bar with "File", "Edit", "View", "Run", "Kernel", "Tabs", "Settings", and "Help". The main area is a terminal window with a title bar that reads "D:\python\python3.9.6\pythc X". The terminal content is as follows:

```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> sys.platform
'win32'
>>> print(2+5)
7
>>> exit()[]
```



# Jupyter/Terminado (iocApp Console)

```
C:\WINDOWS\system32\cmd X  
  
## Register all support components, I say  
dbLoadDatabase "./dbd/sr_ioc.dbd"  
sr_ioc_registerRecordDeviceDriver pdbbase  
Warning: IOC is booting with TOP = "D:/sandbox/2021_EPICS_Collaboration_Meeting/presentations/Tanner/demos/testIoc"  
        but was built with TOP = "D:/sandbox/AARMS/AARMS_save_restore_local_delete_after_deploy/testIoc"  
## Load record instances  
dbLoadRecords("./db/sr_test.db", "user=tannerr")  
iocInit  
Starting iocInit  
#####  
## EPICS R3.14.12.8  
## EPICS Base built Apr 24 2020  
#####  
cas warning: Configured TCP port was unavailable.  
cas warning: Using dynamically assigned TCP port 52635,  
cas warning: but now two or more servers share the same UDP port.  
cas warning: Depending on your IP kernel this server may not be  
cas warning: reachable with UDP unicast (a host's IP in EPICS_CA_ADDR_LIST)  
iocRun: All initialization complete  
dbpf ioc:app:start "2022-09-13 14:40:12.65"  
DBR_STRING:          "2022-09-13 14:40:12.65"  
AARMS_IOC_SIM > █
```



# pyProcServ

- Multiple options for wrapping a subprocess in Python:
  - subprocess
  - multiprocessing
  - pywinpty
  - Asyncio subprocess functions
- ...as well as for networking
  - socketserver
  - asyncio
  - Telnetlib3 (asyncio-compatible)

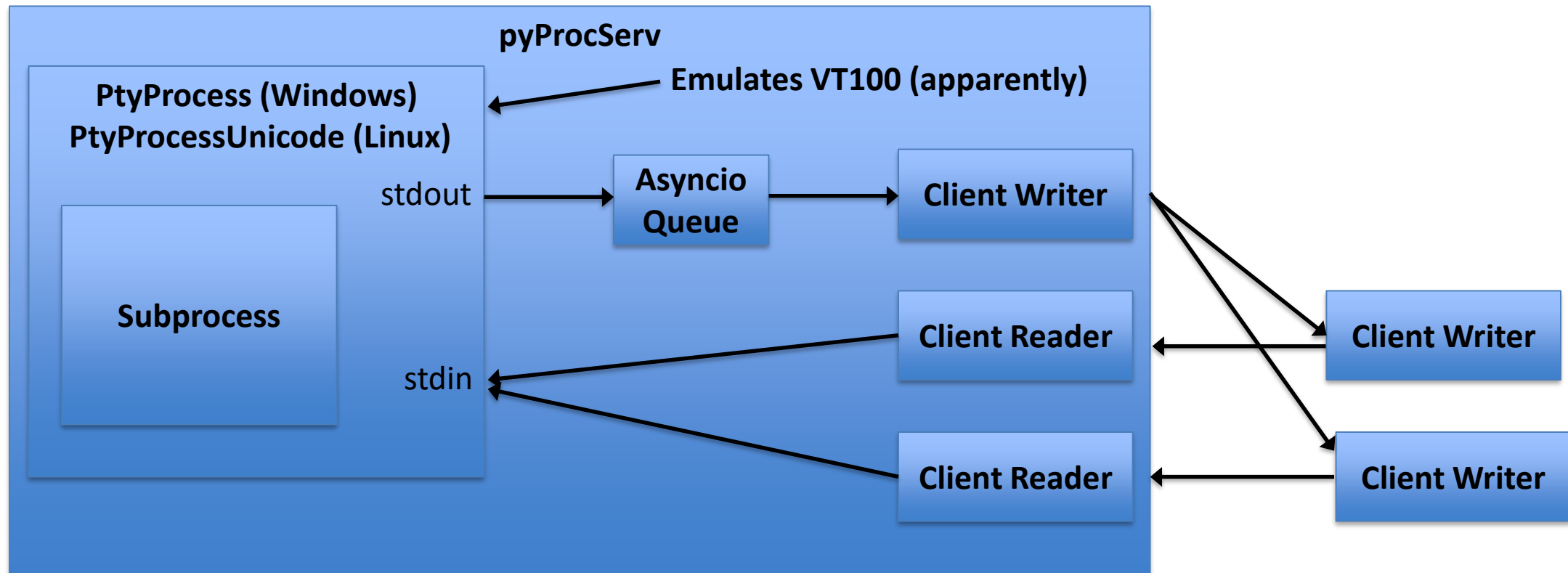


# Asyncio

- Concurrency in a single thread
- Cooperative
- Used by PythonIOCC
- Used by Terminado (Tornado loop v6 wraps asyncio loop)
- Telnetlib3 uses asyncio
- Great separation between transport and protocol
- There is an asyncSSH module



# Basic Layout



# Telnet Output (Sockets)

C:\> Telnet localhost

```
Microsoft Windows [Version 10.0.18363.1556] [9C  
(c) 2019 Microsoft Corporation. All rights reserved. [52C  
D:\sandbox\epics\pyProcServ\area_51> [16C
```





# Telnet Output (Socket)

```
Microsoft Windows [Version 10.0.18363.1556][9C
(c) 2019 Microsoft Corporation. All rights reserved.
[52C
D:\sandbox\epics\pyProcServ\area_51>[16C
dir

Volume in drive D is DATA[35C
Volume Serial Number is DCBD-282A[27C
[61C
Directory of D:\sandbox\epics\pyProcServ\area_51[12C
[61C
2022-09-13 04:13 PM <DIR>      .[21C
2022-09-13 04:13 PM <DIR>      ..[20C
2022-09-13 11:51 AM          10,091 pass_through.py[26C
2022-08-30 07:24 PM          22,964 testing_telnet.py[24C
2022-01-27 08:09 PM          11,979 test_subprocess.py[23C
2022-01-13 08:52 PM           2,498 web_term.py[30C
2022-08-10 05:12 PM           4,538 winsuedoterm.py[25C
2022-08-15 10:07 AM <DIR>      __pycache__[30C
          36 File(s)    211,074 bytes[35C
          4 Dir(s) 1,687,143,002,112 bytes free[27C
0;C:\Windows\System32\cmd.exe
          D:\sandbox\epics\pyProcServ\area_51>[44C
```



# Telnet Output (Telnetlib3)

- Using telnetlib3
- Same output
- Noticed a lot more going back and forth (IAC) 0xFF



# Under the Hood

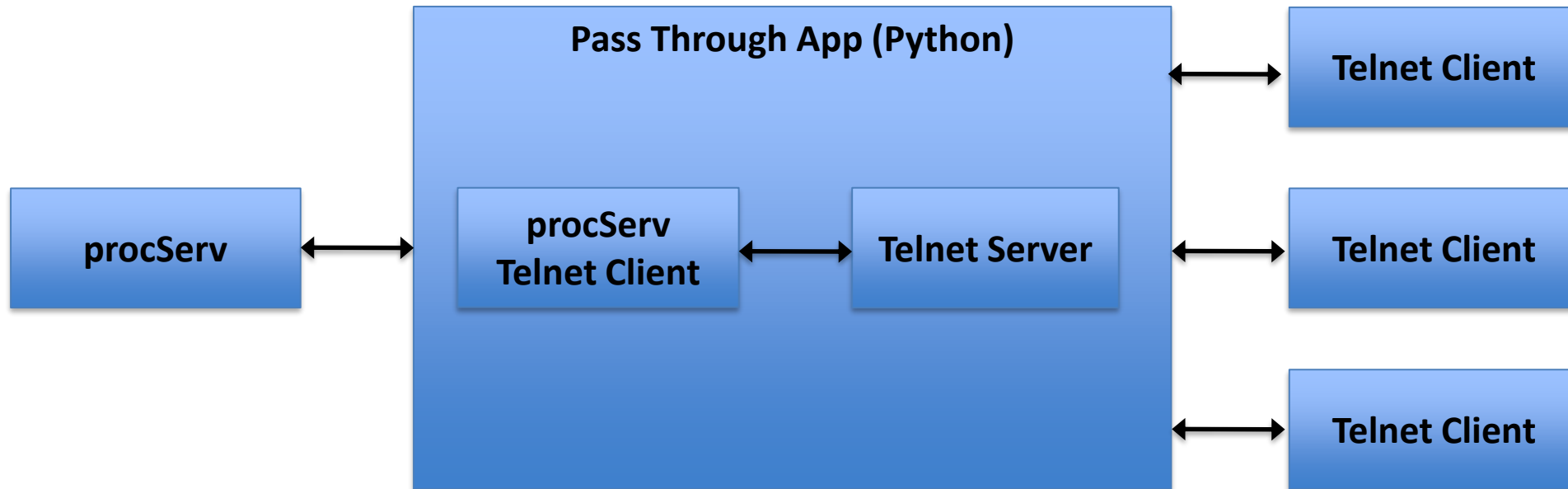
x1b[?25|Microsoft Windows [Version 10.0.18363.1556]  
\x1b[9X\x1b[9C\r\n

(c) 2019 Microsoft Corporation. All rights reserved.\r\n  
\x1b[52X\x1b[52C\r\n

D:\\sandbox\\epics\\pyProcServ\\area\_51>\x1b[16X\x1b[16C\x1  
b[4;37H\x1b[?25h



# Pass-Through Test



# Conclusion

- Any help GREATLY appreciated
  - Further reading
  - Links
  - Other resources
  - Guidance
- All is not lost:
  - Concentrator proof-of-concept

