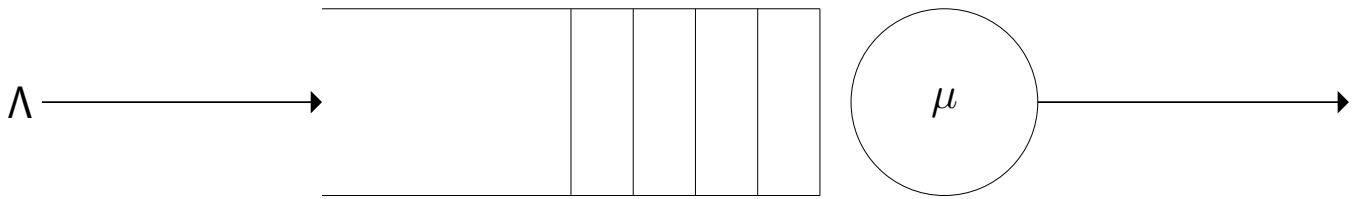
ASQ Simulates Queues

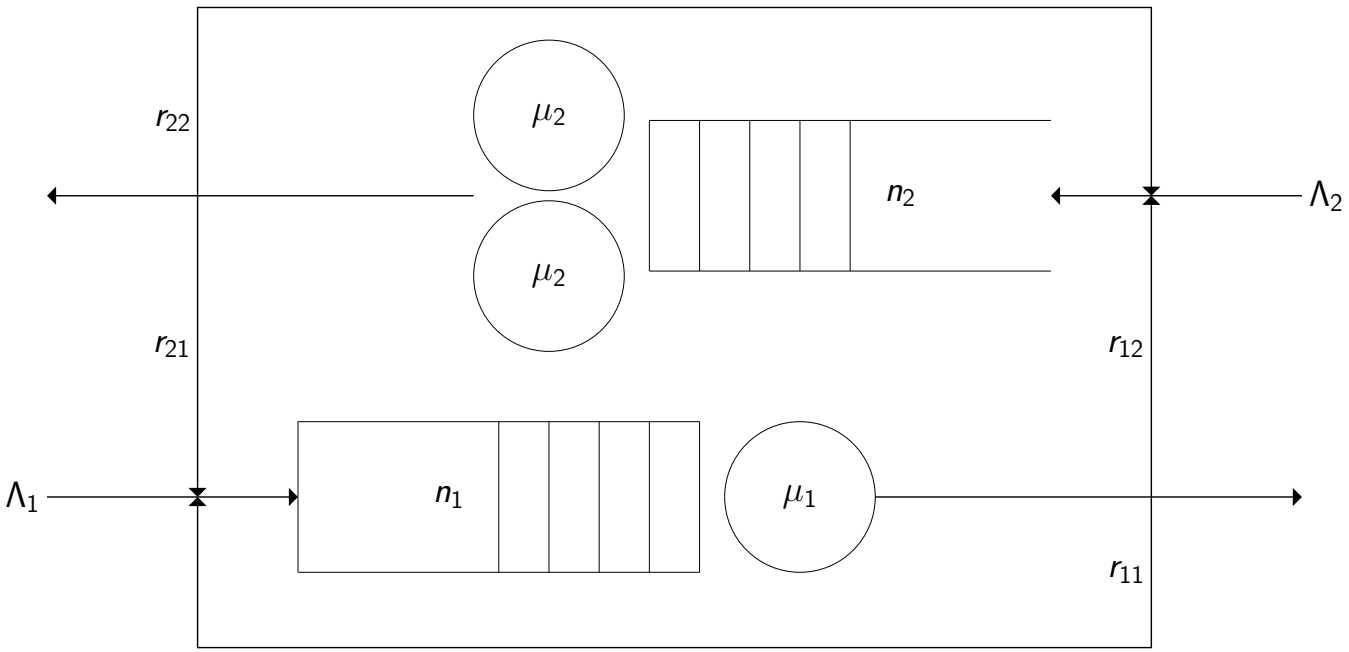
Geraint Palmer

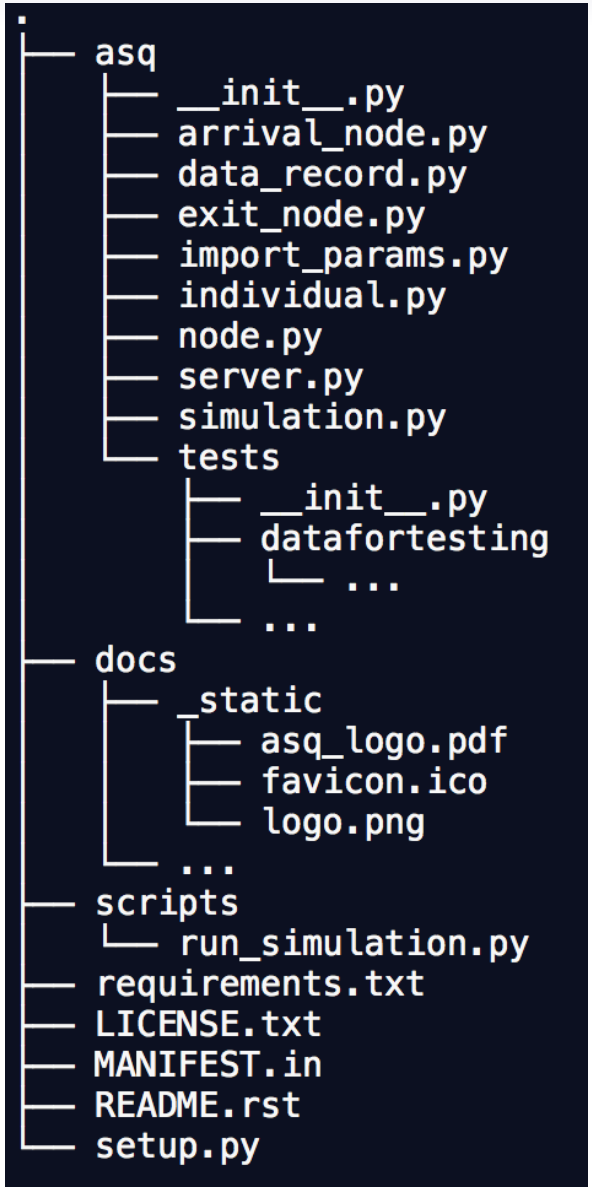
PyCon Namibia 2016

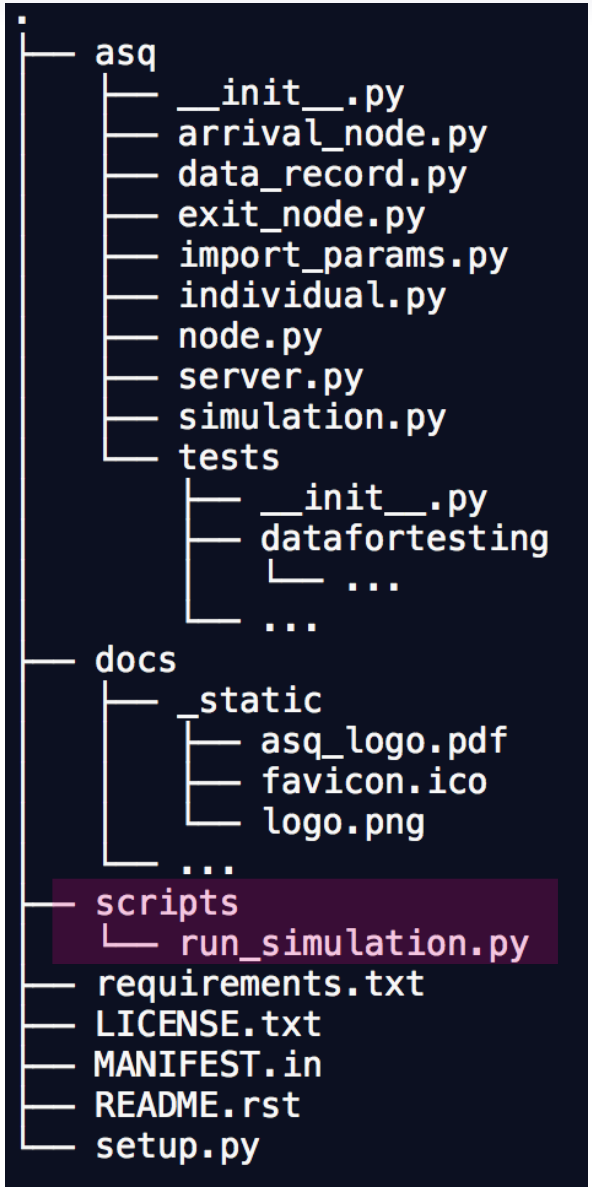
What is a Queue?



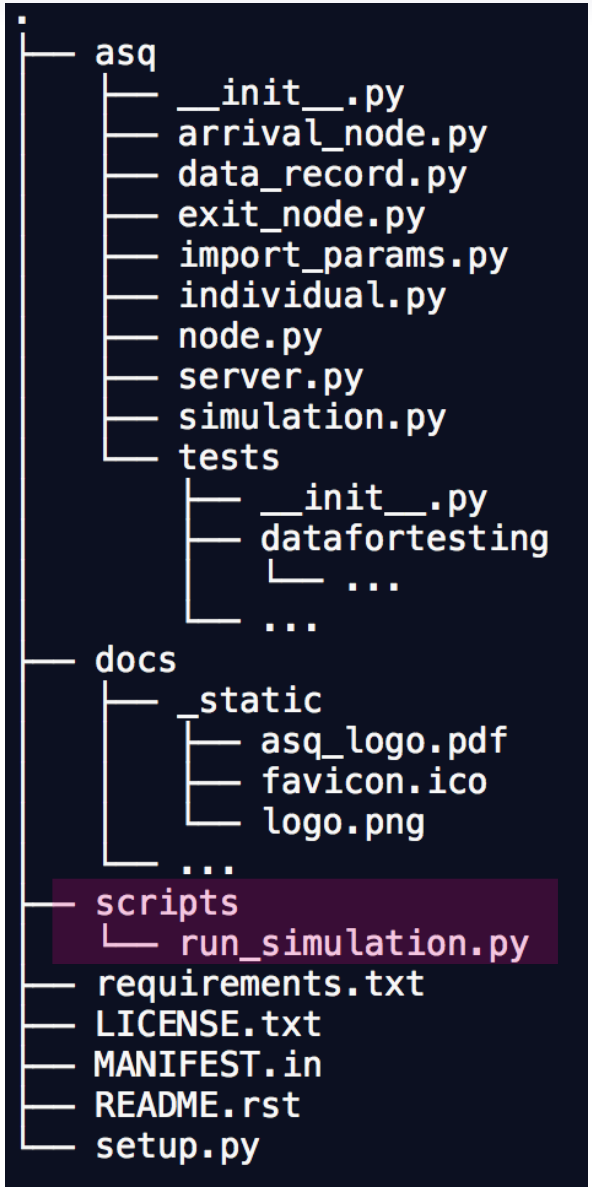
What is a Queue?

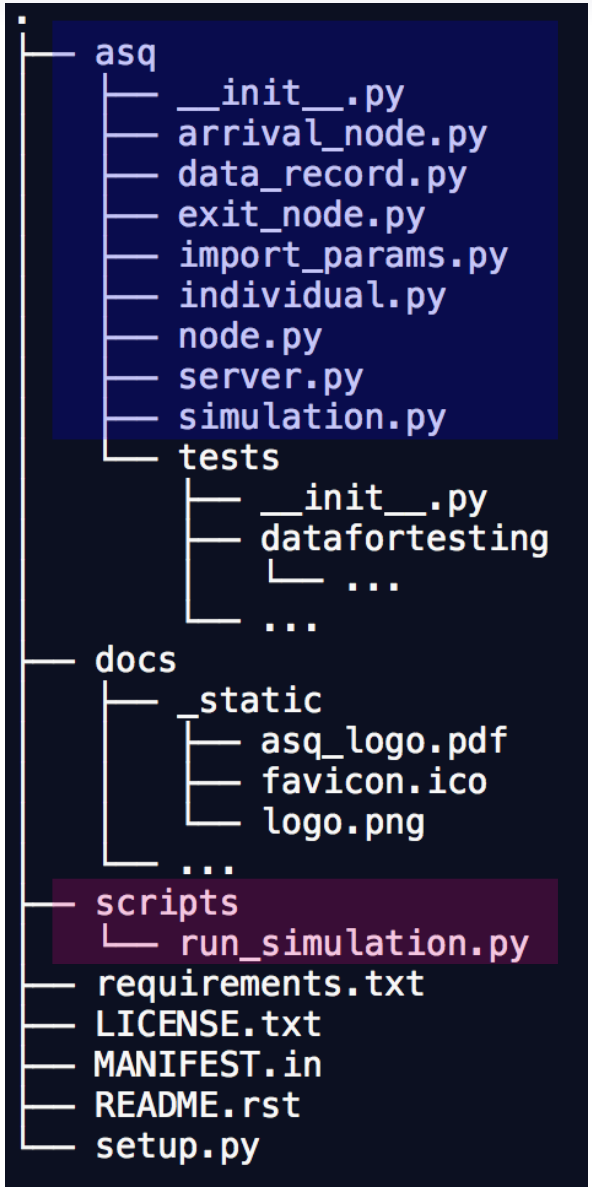



.
├── asq
│   ├── __init__.py
│   ├── arrival_node.py
│   ├── data_record.py
│   ├── exit_node.py
│   ├── import_params.py
│   ├── individual.py
│   ├── node.py
│   ├── server.py
│   ├── simulation.py
│   └── tests
│   ├── __init__.py
│   ├── datafortesting
│   │   └── ...
│   └── ...
├── docs
│   ├── _static
│   │   ├── asq_logo.pdf
│   │   ├── favicon.ico
│   │   └── logo.png
│   └── ...
├── scripts
│   └── run_simulation.py
├── requirements.txt
├── LICENSE.txt
├── MANIFEST.in
├── README.rst
└── setup.py

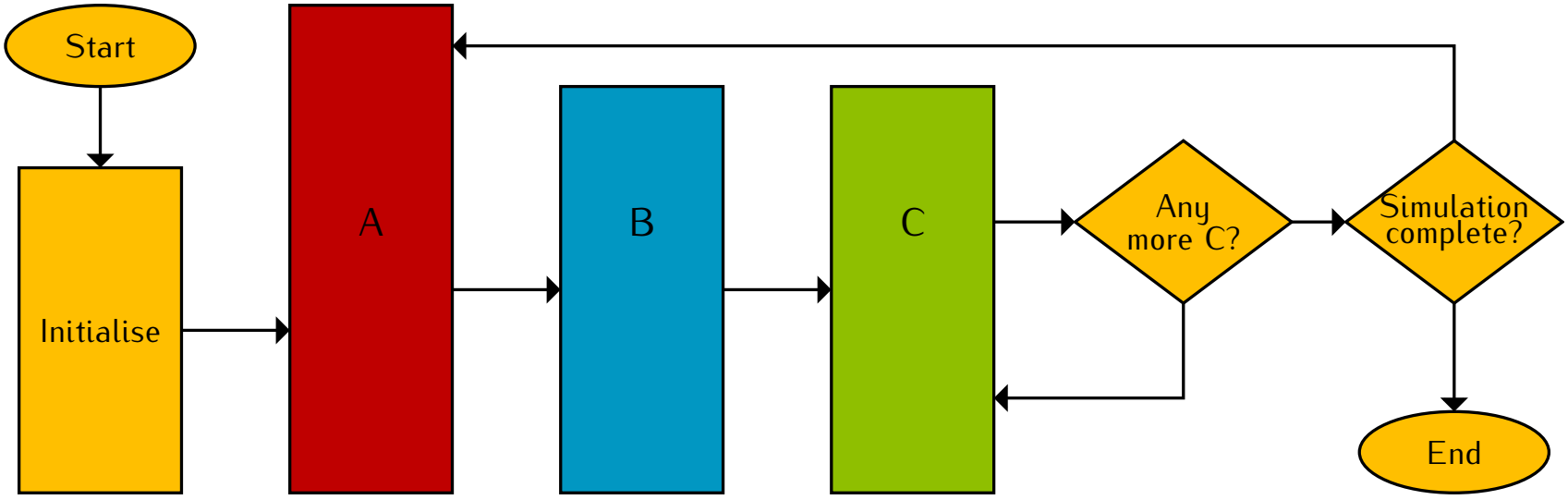

.
asq
__init__.py
arrival_node.py
data_record.py
exit_node.py
import_params.py
individual.py
node.py
server.py
simulation.py
tests
__init__.py
datafortesting
...
...
docs
_static
asq_logo.pdf
favicon.ico
logo.png
...
scripts
run_simulation.py
requirements.txt
LICENSE.txt
MANIFEST.in
README.rst
setup.py

Demo


.
asq
__init__.py
arrival_node.py
data_record.py
exit_node.py
import_params.py
individual.py
node.py
server.py
simulation.py
tests
__init__.py
datafortesting
...
...
docs
_static
asq_logo.pdf
favicon.ico
logo.png
...
scripts
run_simulation.py
requirements.txt
LICENSE.txt
MANIFEST.in
README.rst
setup.py


.
asq
__init__.py
arrival_node.py
data_record.py
exit_node.py
import_params.py
individual.py
node.py
server.py
simulation.py
tests
__init__.py
datafortesting
...
...
docs
_static
asq_logo.pdf
favicon.ico
logo.png
...
scripts
run_simulation.py
requirements.txt
LICENSE.txt
MANIFEST.in
README.rst
setup.py

Three-Phase Simulation Approach



Event Types

External Arrival

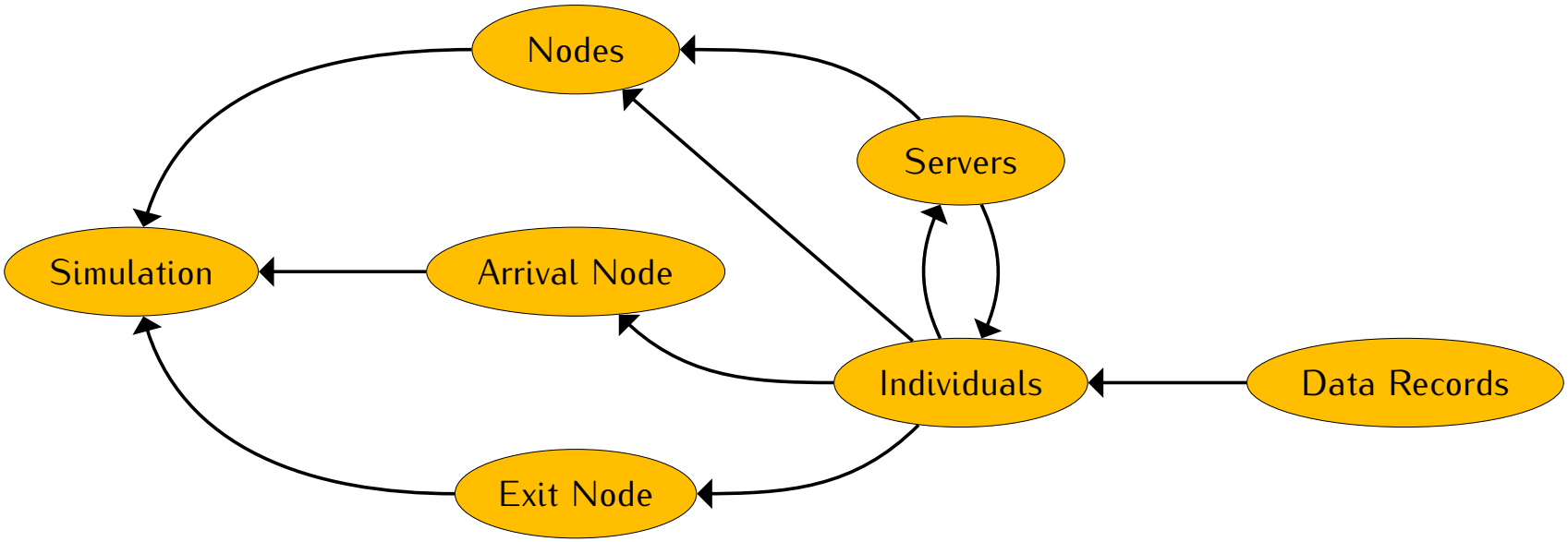
Finish Service

Start Service

Block Customer

Release / Move Customer

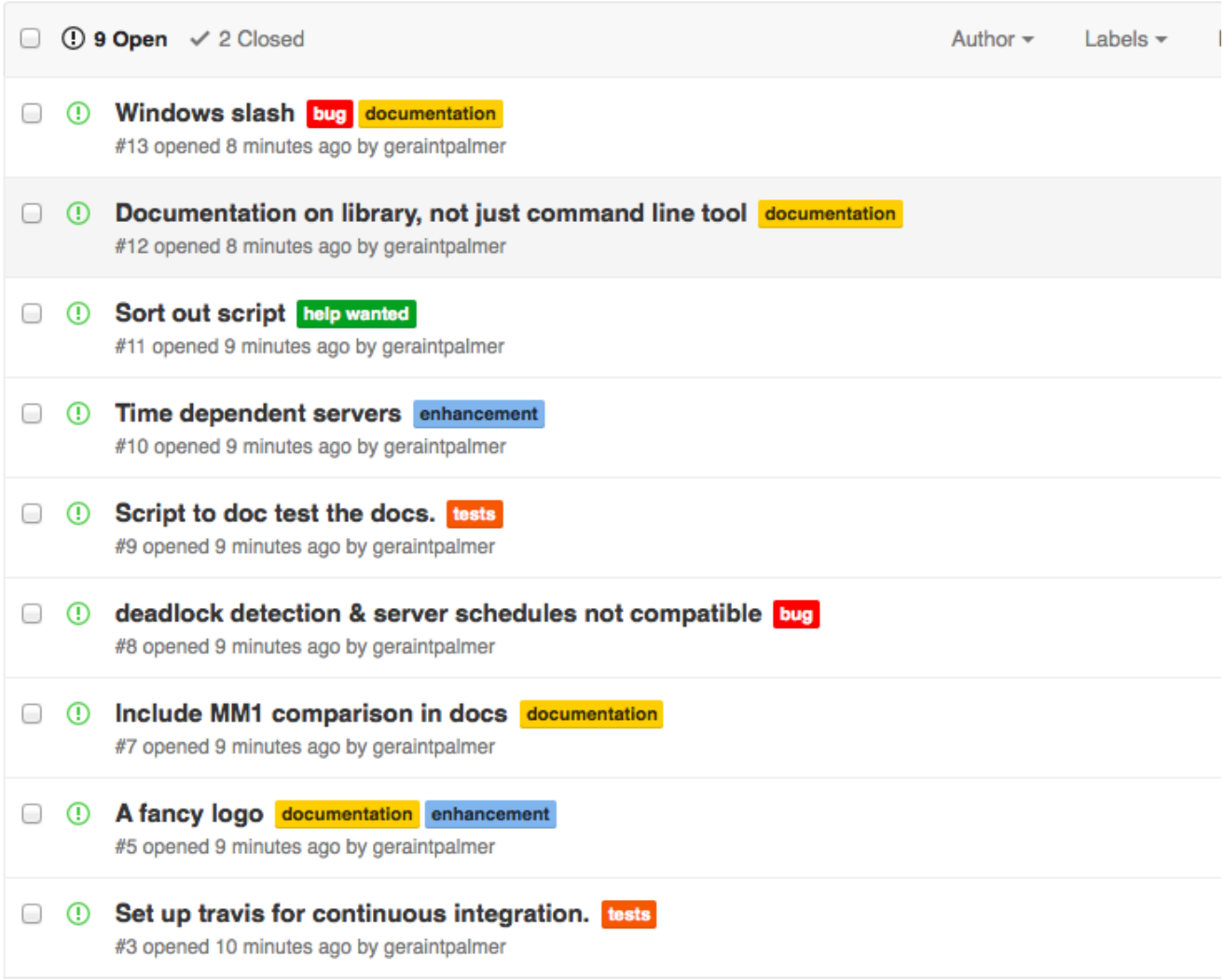
Code Structure

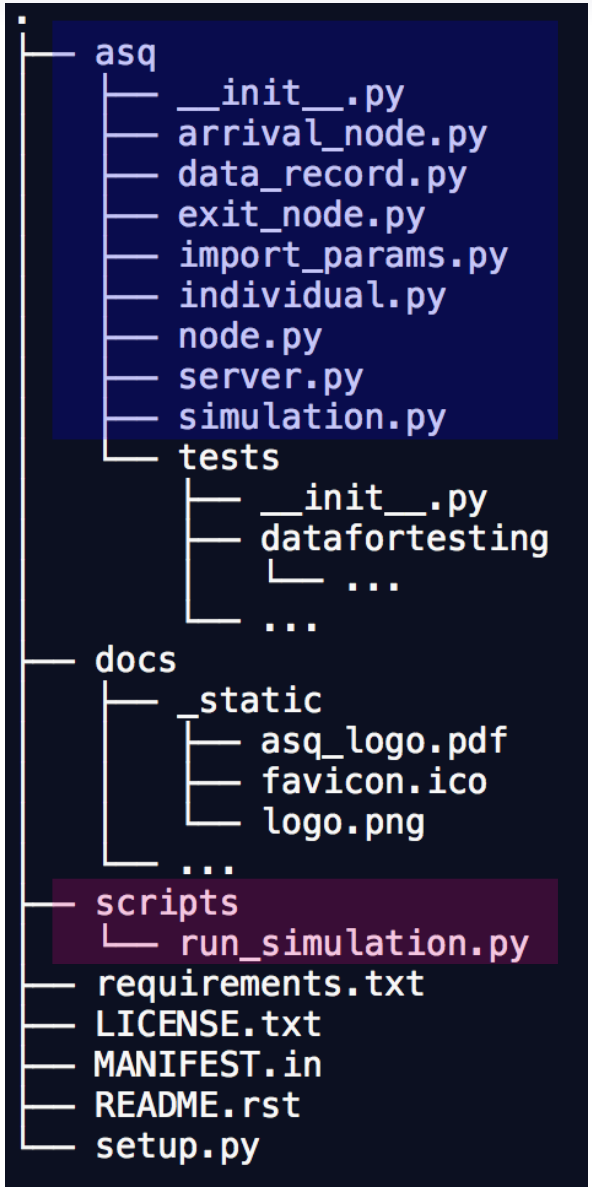


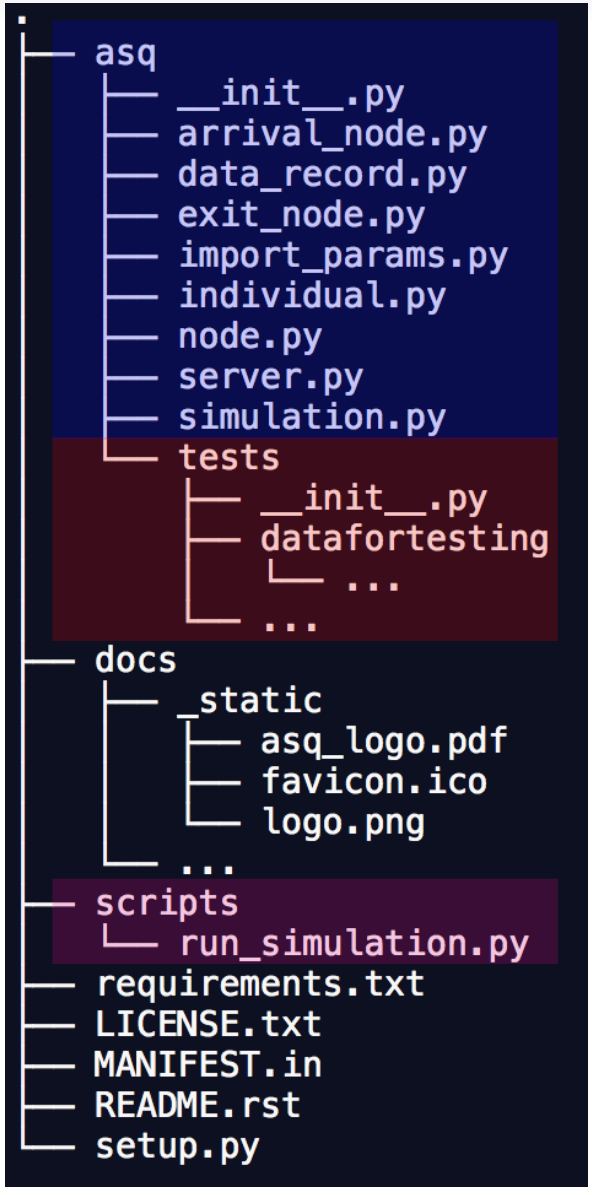
Pair Programing / Collaborative Work

Git & GitHub

GitHub Issues




.
asq
__init__.py
arrival_node.py
data_record.py
exit_node.py
import_params.py
individual.py
node.py
server.py
simulation.py
tests
__init__.py
datafortesting
...
...
docs
_static
asq_logo.pdf
favicon.ico
logo.png
...
scripts
run_simulation.py
requirements.txt
LICENSE.txt
MANIFEST.in
README.rst
setup.py


.
asq
__init__.py
arrival_node.py
data_record.py
exit_node.py
import_params.py
individual.py
node.py
server.py
simulation.py
tests
__init__.py
datafortesting
...
...
docs
_static
asq_logo.pdf
favicon.ico
logo.png
...
scripts
run_simulation.py
requirements.txt
LICENSE.txt
MANIFEST.in
README.rst
setup.py

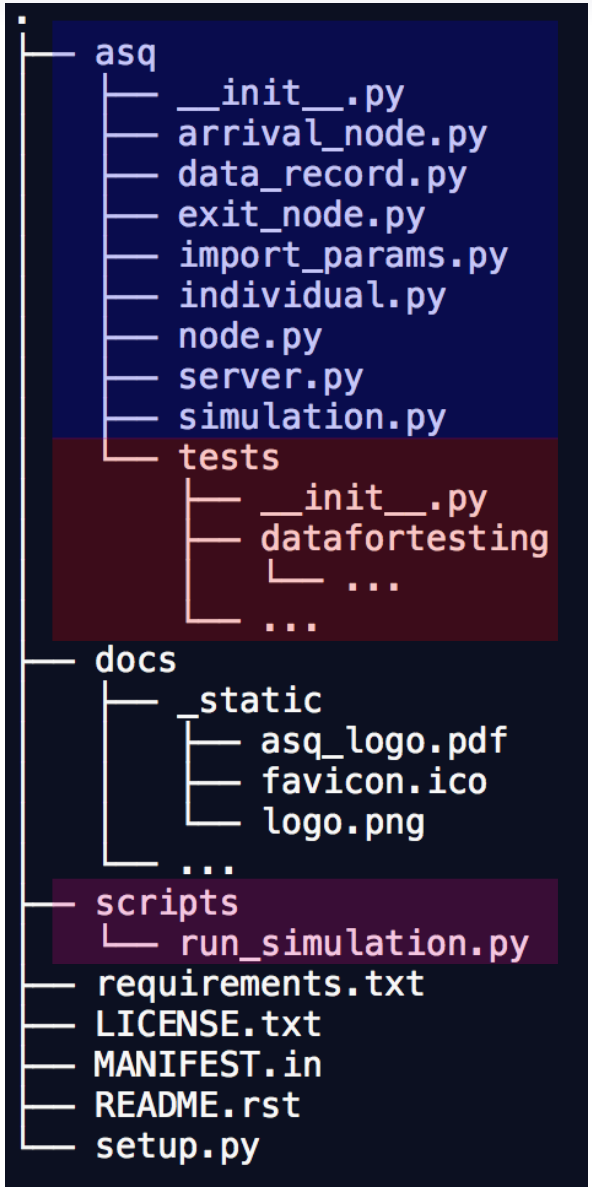
Doctests

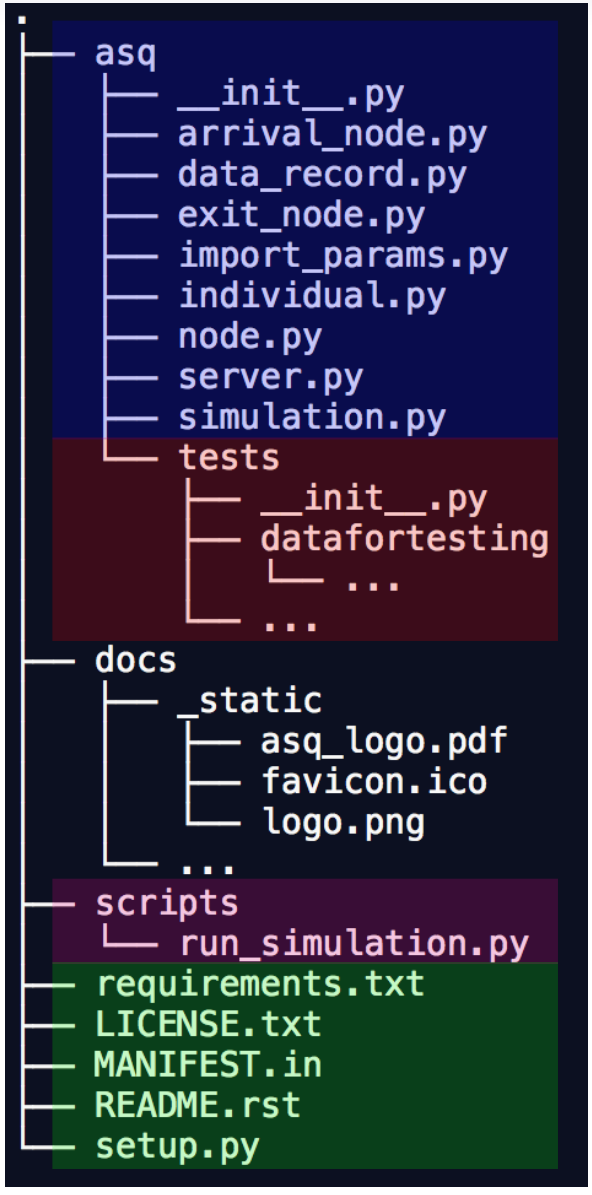
```
def geraints_function(a, b):
    """
    Returns the absolute difference between a and b

        >>> my_amazing_function(7, 9)
        2
        >>> round(geraints_function(20.4, 3.1), 1)
        17.3
    """
    return max(b-a, a-b)
```

Unittests

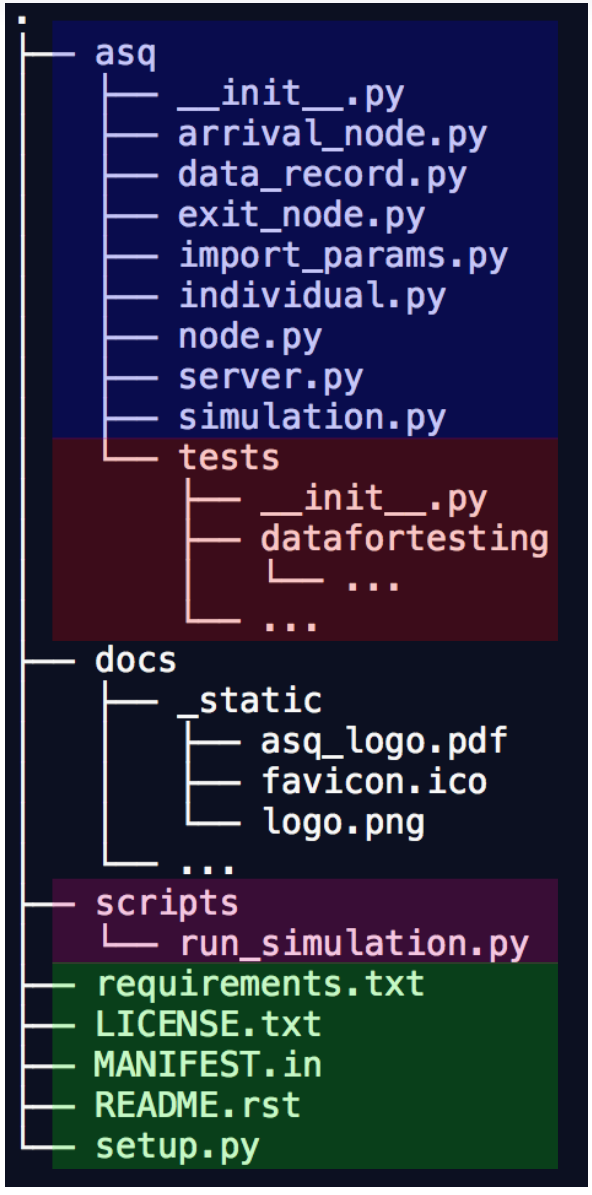
```
import unittest
import geraints_function

class TestGeraintsFunction(unittest.TestCase):

    def test_geraints_function(self):
        self.assertEqual(geraints_function(7, 9), 2)
        self.assertEqual(geraints_function(20.4, 3.1), abs(20.4-3.1))
```

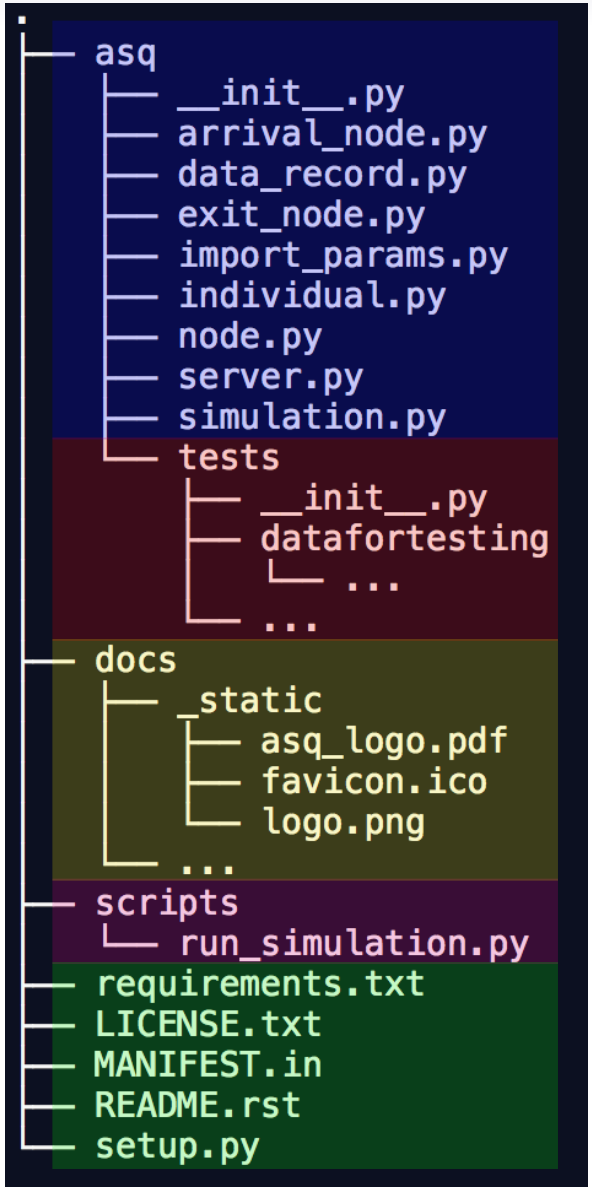
Travis


.
asq
__init__.py
arrival_node.py
data_record.py
exit_node.py
import_params.py
individual.py
node.py
server.py
simulation.py
tests
__init__.py
datafortesting
...
...
docs
_static
asq_logo.pdf
favicon.ico
logo.png
...
scripts
run_simulation.py
requirements.txt
LICENSE.txt
MANIFEST.in
README.rst
setup.py


.
asq
__init__.py
arrival_node.py
data_record.py
exit_node.py
import_params.py
individual.py
node.py
server.py
simulation.py
tests
__init__.py
datafortesting
...
...
docs
_static
asq_logo.pdf
favicon.ico
logo.png
...
scripts
run_simulation.py
requirements.txt
LICENSE.txt
MANIFEST.in
README.rst
setup.py

Packaging


.
asq
__init__.py
arrival_node.py
data_record.py
exit_node.py
import_params.py
individual.py
node.py
server.py
simulation.py
tests
__init__.py
datafortesting
...
...
docs
_static
asq_logo.pdf
favicon.ico
logo.png
...
scripts
run_simulation.py
requirements.txt
LICENSE.txt
MANIFEST.in
README.rst
setup.py


.
asq
__init__.py
arrival_node.py
data_record.py
exit_node.py
import_params.py
individual.py
node.py
server.py
simulation.py
tests
__init__.py
datafortesting
...
...
docs
_static
asq_logo.pdf
favicon.ico
logo.png
...
scripts
run_simulation.py
requirements.txt
LICENSE.txt
MANIFEST.in
README.rst
setup.py

Documentation

Academic Uses

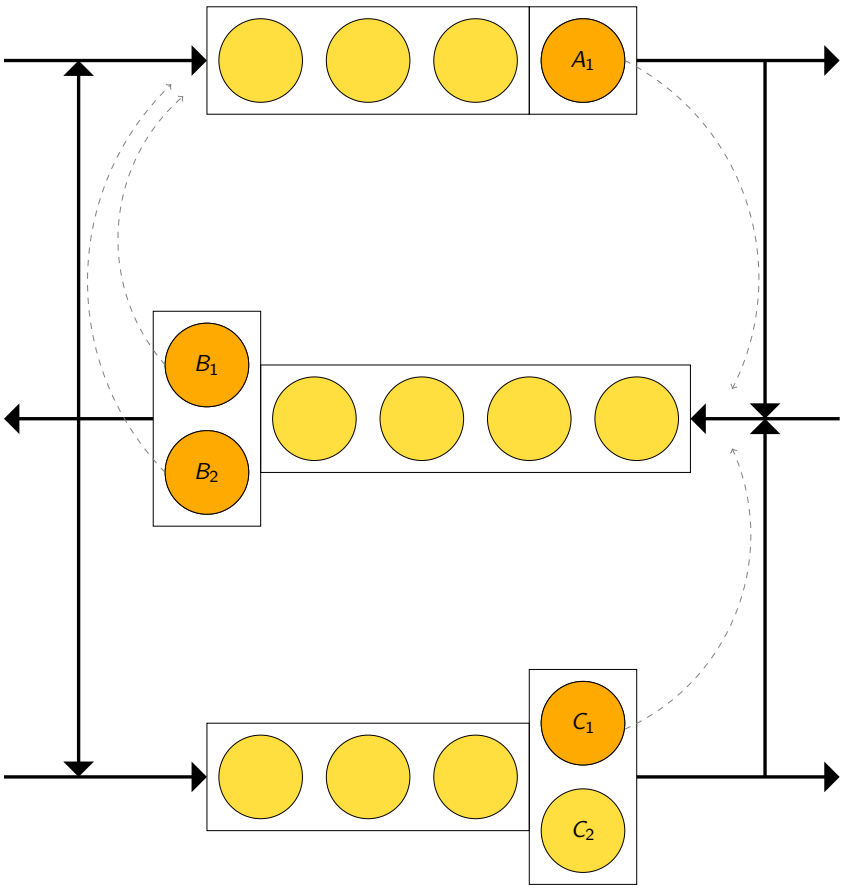
Theoretical Work

Investigating deadlock in queueing networks.
(Geraint Palmer, Prof. Paul Harper, Dr. Vincent Knight)

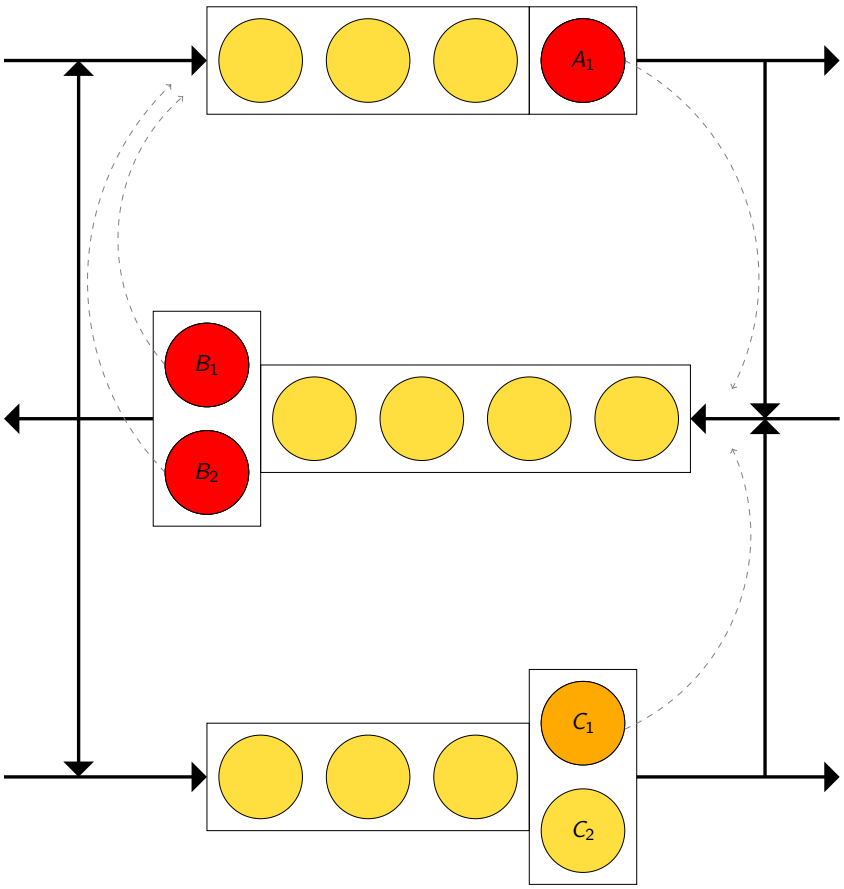
Practical Work

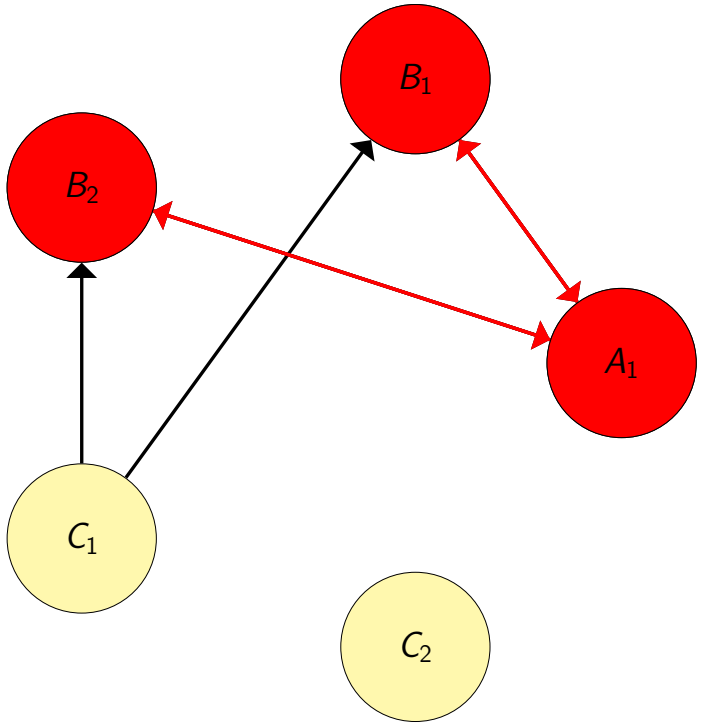
Modelling an ophthalmology clinic to strategise scheduling.
(Lieke Hölscher, Dr. Jennifer Morgan)

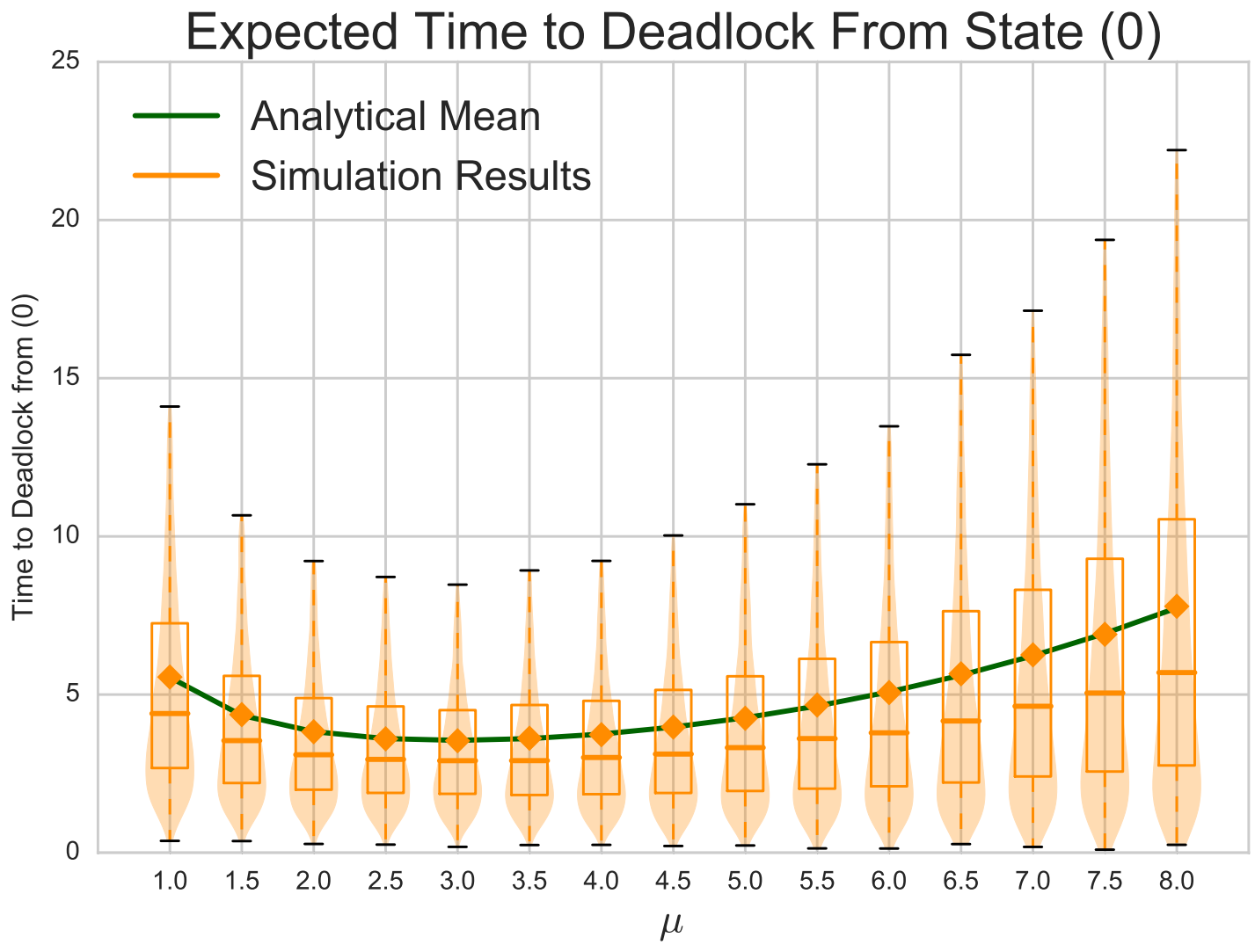
Investigating Deadlock



Investigating Deadlock






Expected Time to Deadlock From State (0)
Analytical Mean
Simulation Results
Time to Deadlock from (0)
0
5
10
15
20
25
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
μ

Modelling Ophthalmology Clinic