Fault Injection

Agenda

- Chaos Engineering
- Fault injection 101

Netflix and Chaos Monkeys
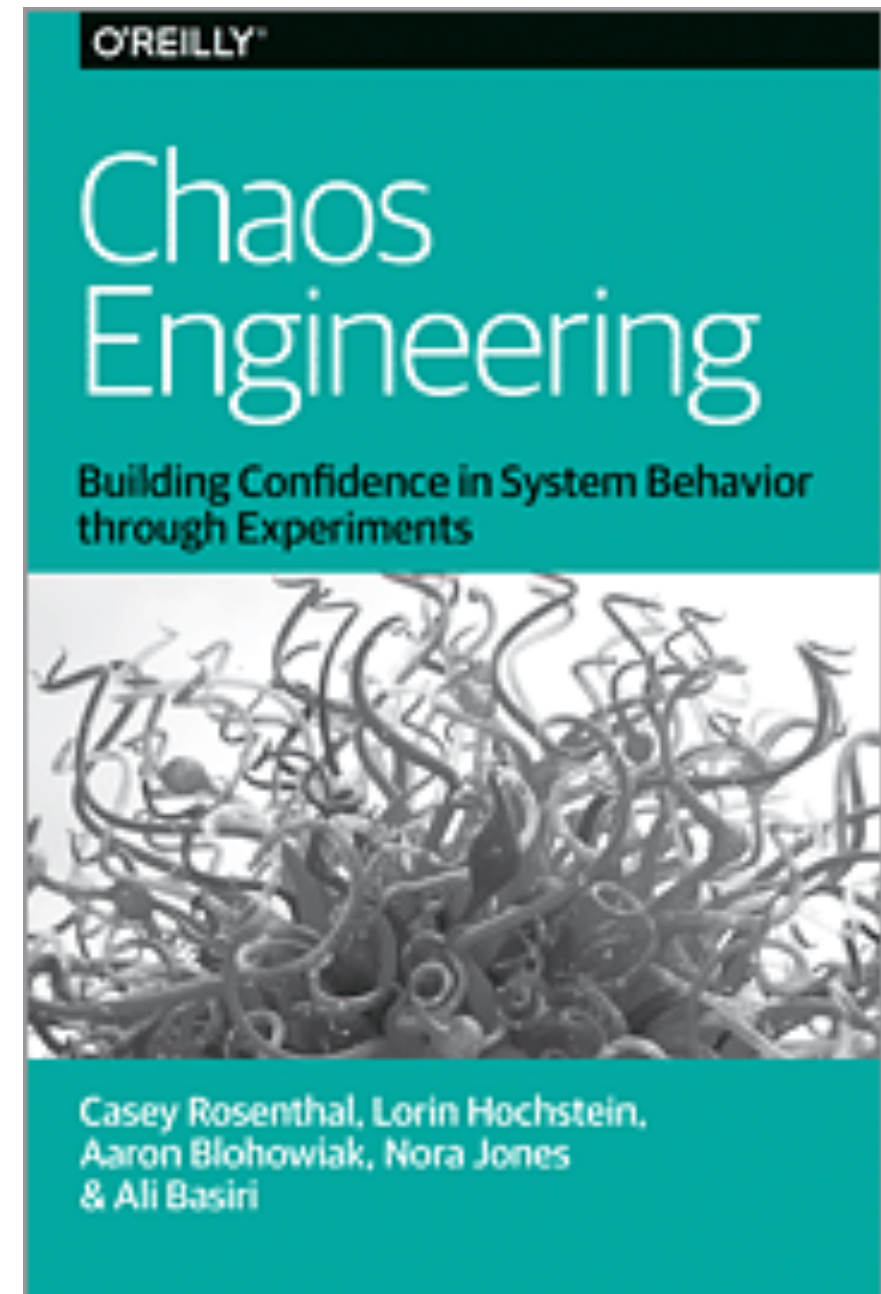
Simian Army Projects

- Chaos Monkey
- Chaos Gorilla
- Chaos Kong
- Janitor Monkey
- Doctor Monkey
- Compliance Monkey
- Latency Monkey
- Security Monkey

Chaos Engineering

"Chaos Engineering is the discipline of experimenting on a distributed system in order to build confidence in the system's capability to withstand turbulent conditions in production"

- principlesofchaos.org

O'REILLY®

Chaos Engineering

Building Confidence in System Behavior through Experiments

Casey Rosenthal, Lorin Hochstein,
Aaron Blohowiak, Nora Jones
& Ali Basiri

# Chaos Engineering and Faults

- Most chaos engineering is done in production

- But it can also be beneficial to test in dev and pre-prod

- Manually terminating instances and modifying the network stack is tedious and error prone

- "Mocks bleed into the architectural boundaries"

Hoverfly Fault Injection

- Hoverfly can help automate fault injection

- Primarily two methods:

- Delays

- Middleware
  - Response/Request consumed via stdin/stdout
  - Modify data with your language of choice
  - Return HTTP status codes, bad data, etc…

Fault Injection

Consumer

External Producer

Middleware

Fault Injection
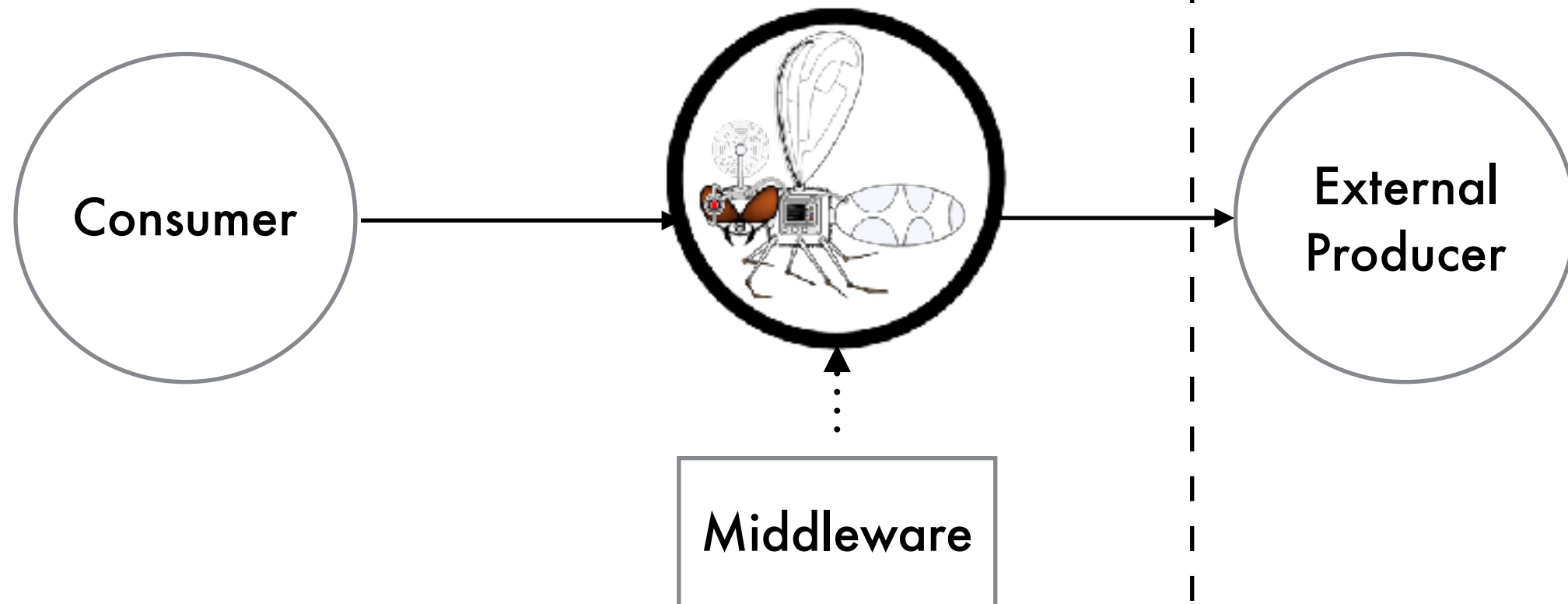
```python
def main():

    data = sys.stdin.readlines()
    # this is a json string in one line so we are interested in that one line
    payload = data[0]
    logging.debug(payload)


    payload_dict = json.loads(payload)


    payload_dict['response']['status'] = 201
    payload_dict['response']['body'] = "body was replaced by middleware"


    # now let' sleep for 2 seconds
    sleep(2)


    # returning new payload
    print(json.dumps(payload_dict))
```