STAR TREK

THE NEXT GENERATION

RailsConf

THE Q CONTINUUM
Applied Queueing Theory

Queue? Q Who?

Justin Bowen

@TonsOfFun111

@TonsOfFun

CTO Consultant @ SVSG.co

Director of Engineering @ nSightSurgical.ai

Parallelism Examples: GitHub.com/TonsOfFun/RailsConf2022

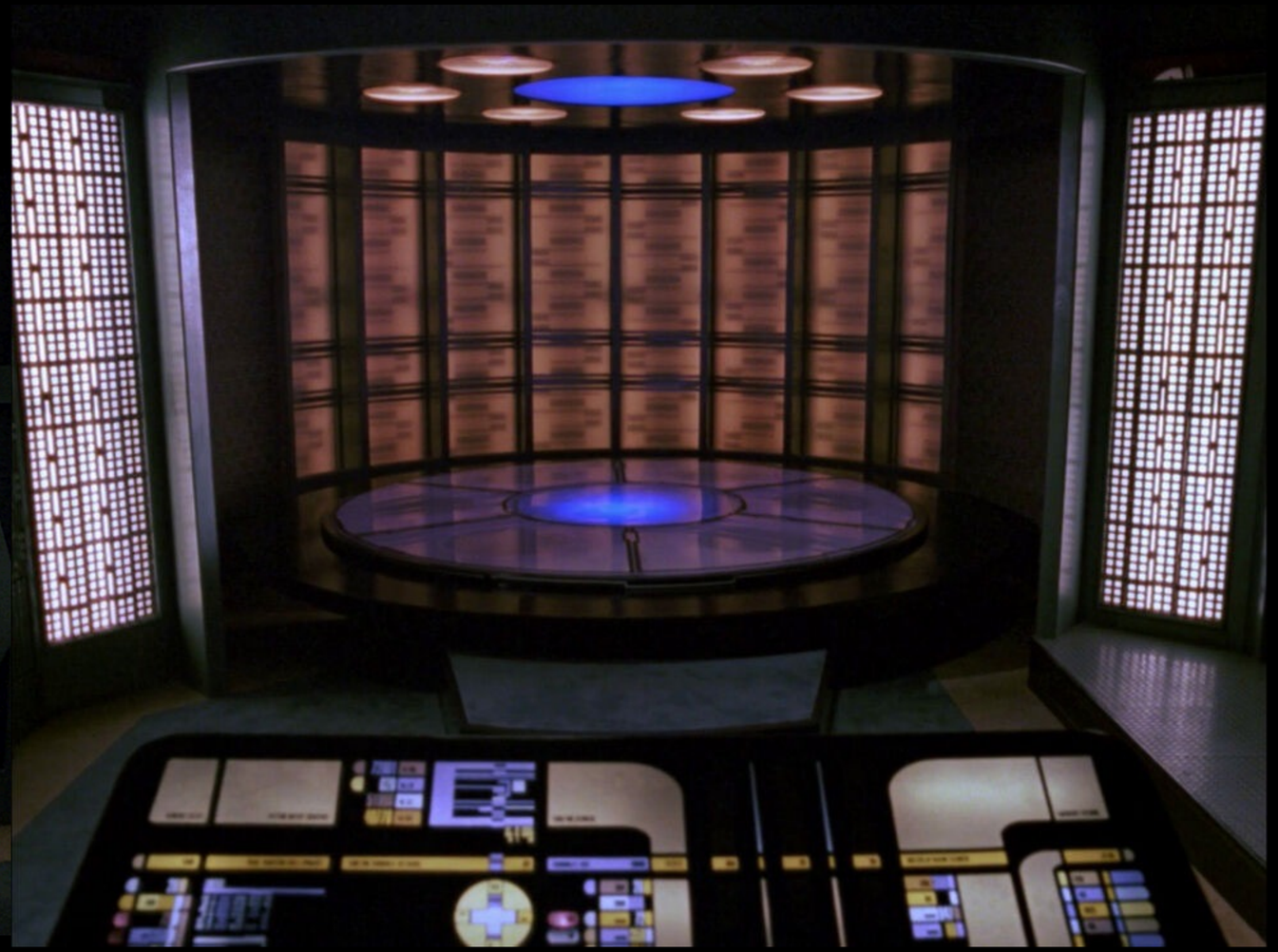
Queue? Q Who?

Star Trek

Make It So

Transporters

Beam me up Scotty!

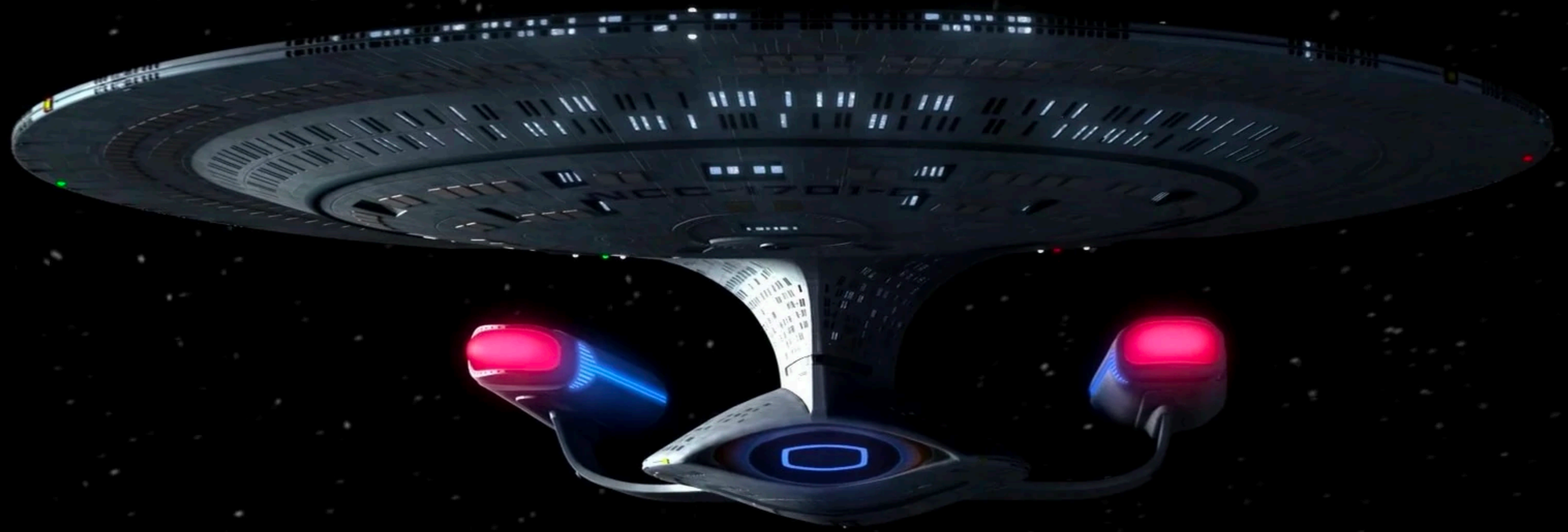


Queue? Q Who?

The Q Continuum

Transporters

Capacity Planning



The Q Continuum

The Power of the Q

The Queue Continuum

Queueing for Humans

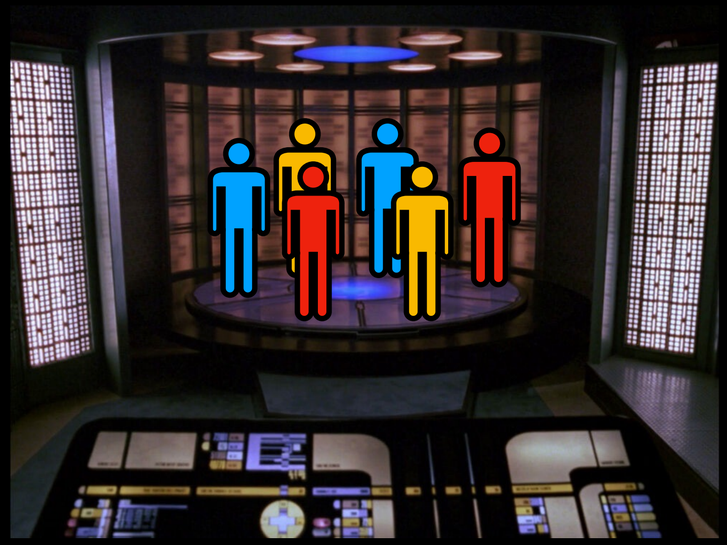
Basic Terminology

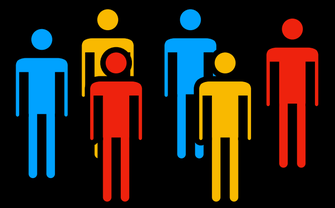
Queueing Things

- Queue - Waiting in line
- Unit of Work (Jobs or Requests)
- Servers
- Service Time

Unit of Work

Transports



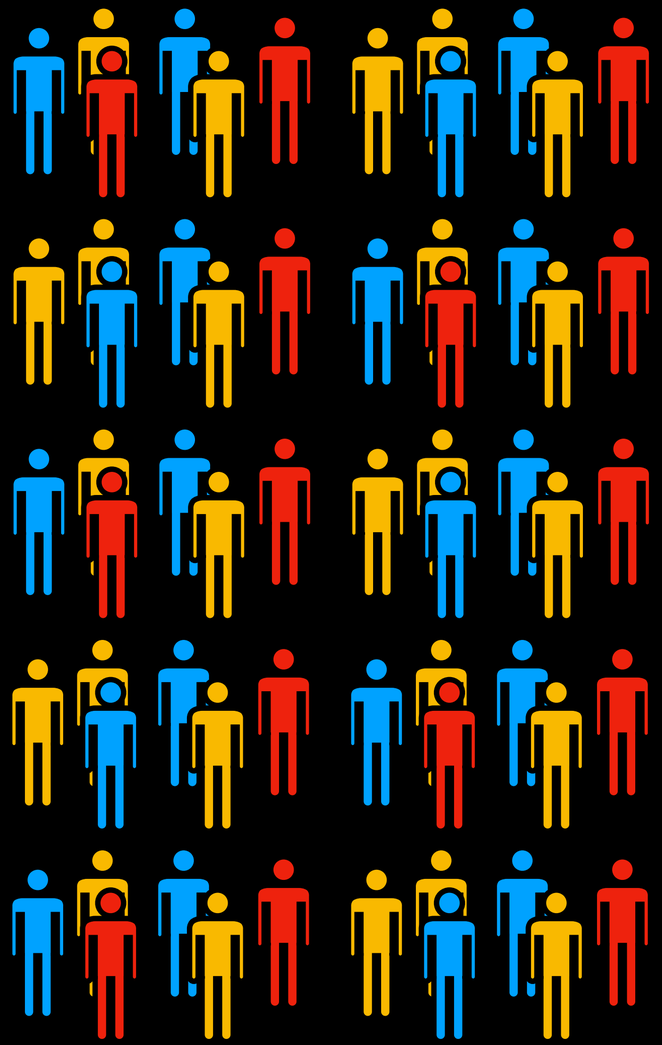


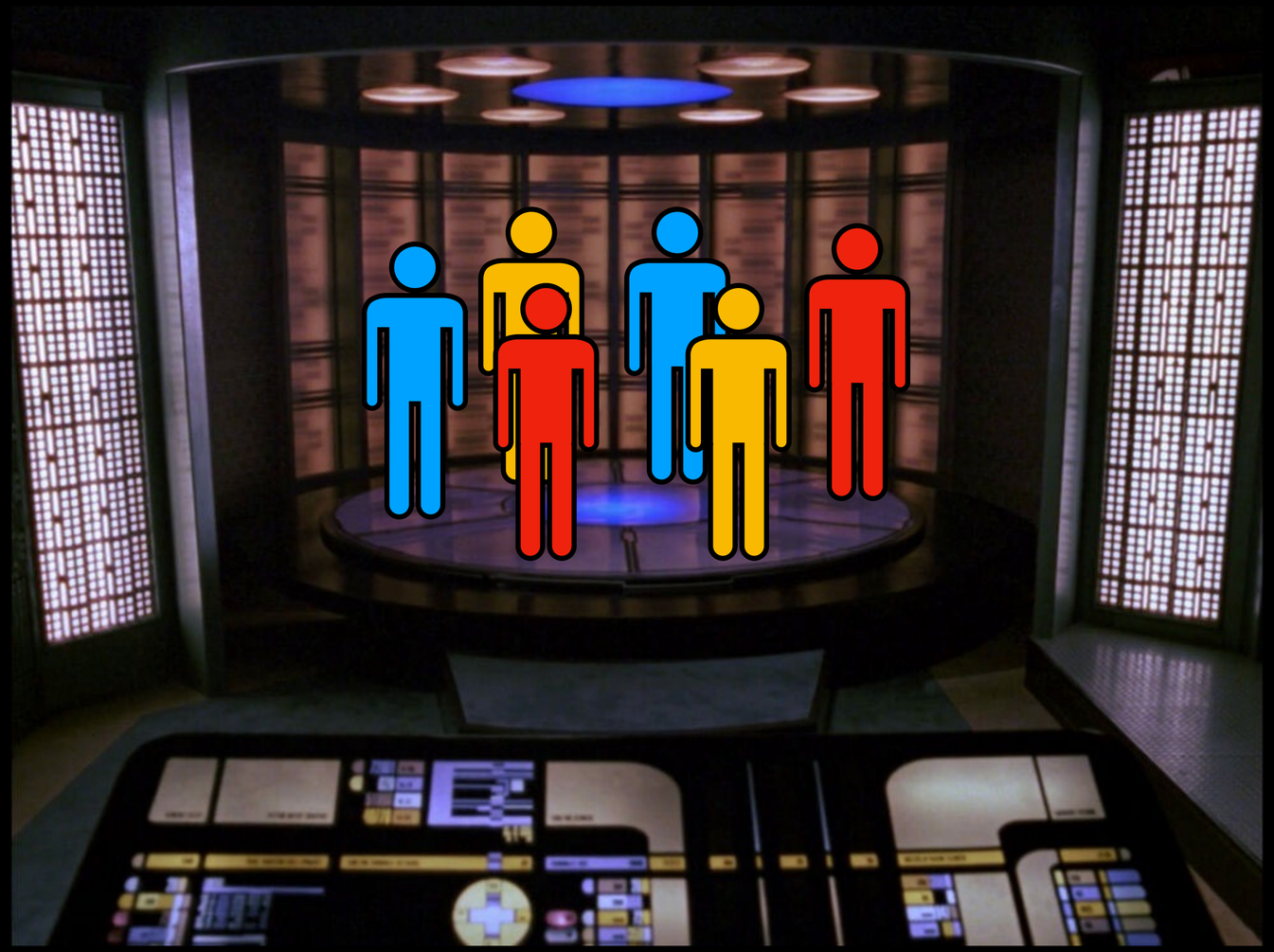
Server with Processors

Ships with Transporters

Little’s Law

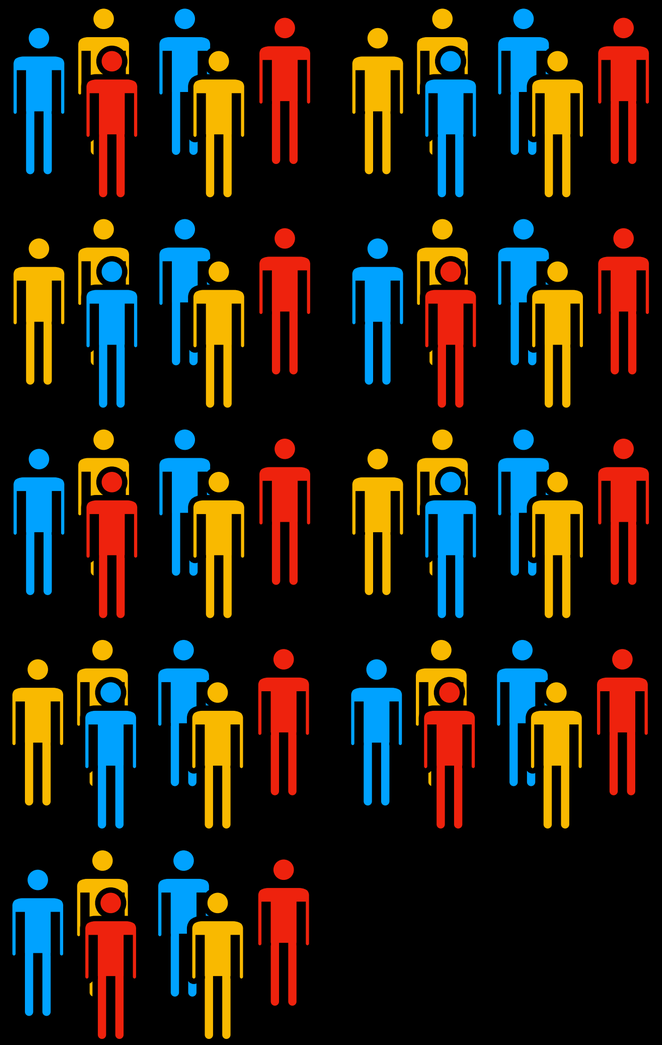
Offered Traffic

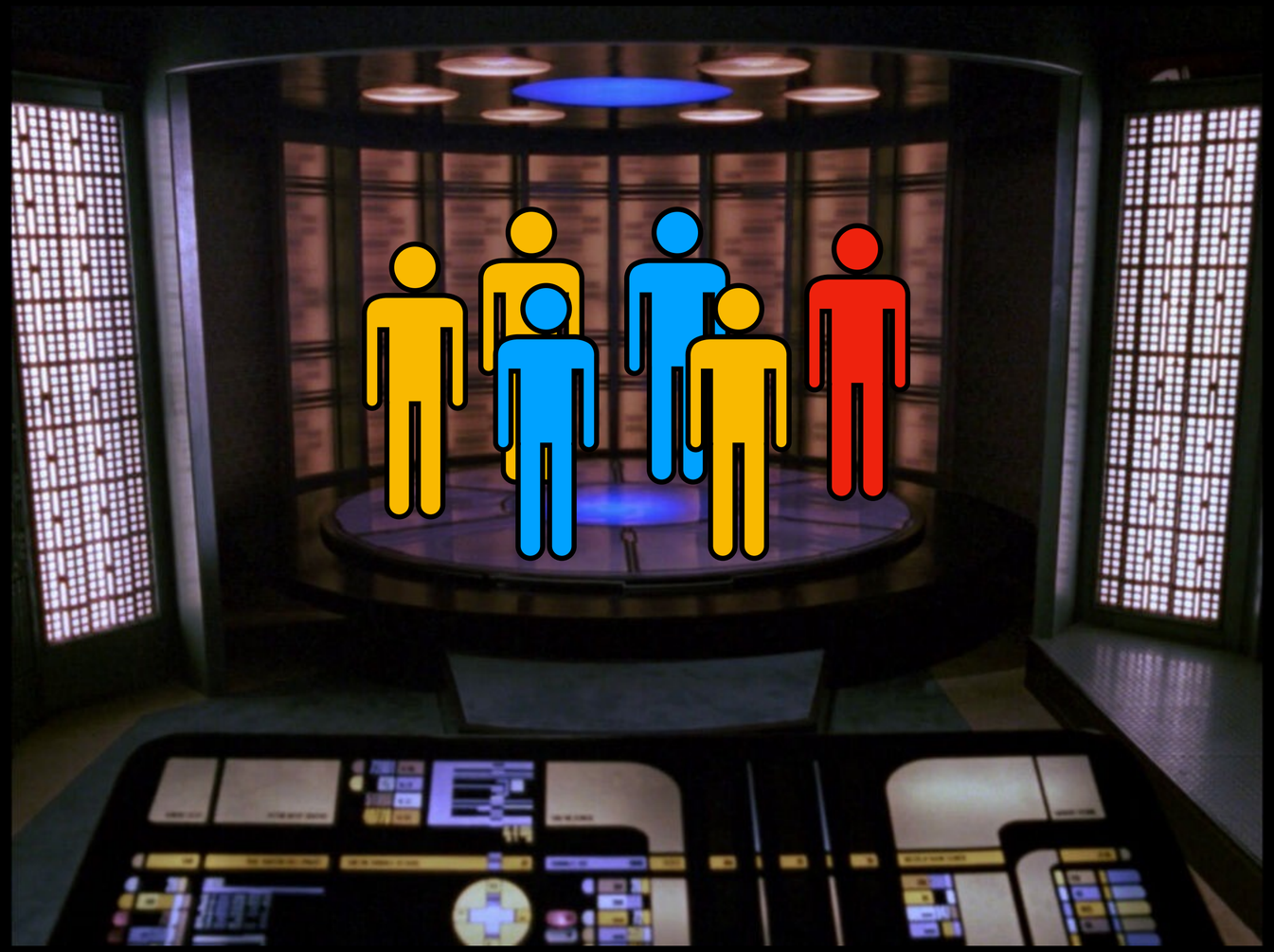




Little's Law

Offered Traffic





Offered Traffic

Service Time * Rate of Jobs

5 seconds

Service Time per Transport

8.3 job/sec

1000 jobs / 2 minutes / 60 seconds = 8.333

Offered Traffic

41.5

5 seconds * 8.3

USE Method - Intel Engineer Brendan Gregg

- Utilization - services used divided by services available
- Saturation - when utilization is at 100%
- Errors - wasted resources

Utilization

Offered Traffic / Parallelism

41.5 crew/second
1 transporter

4150%!!!

41.5 crew/second
6 transporter sites

690%!

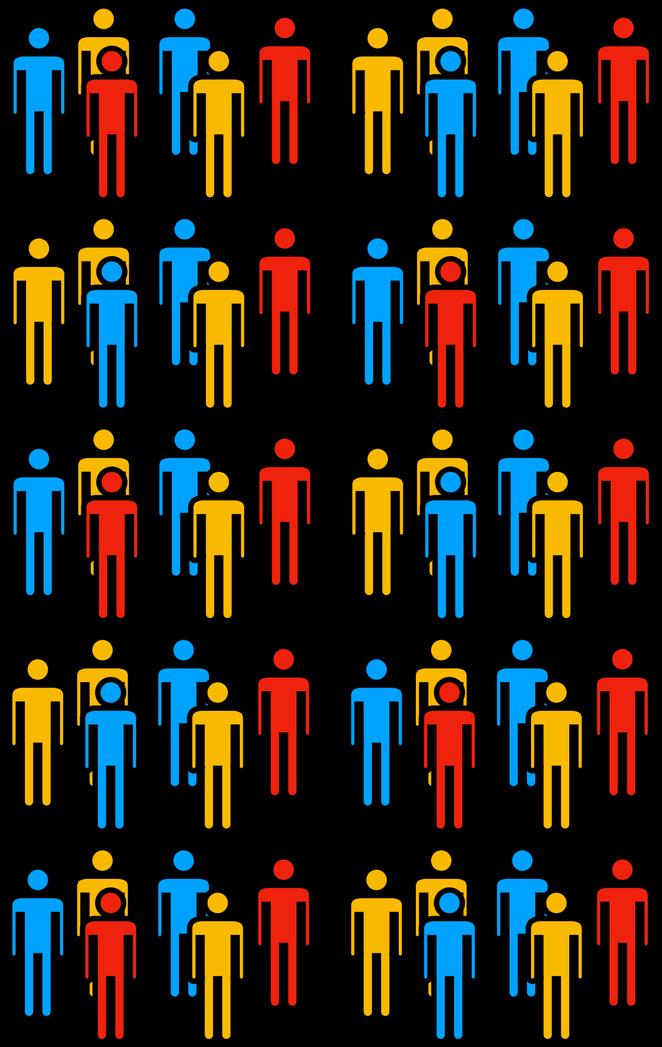
Saturation

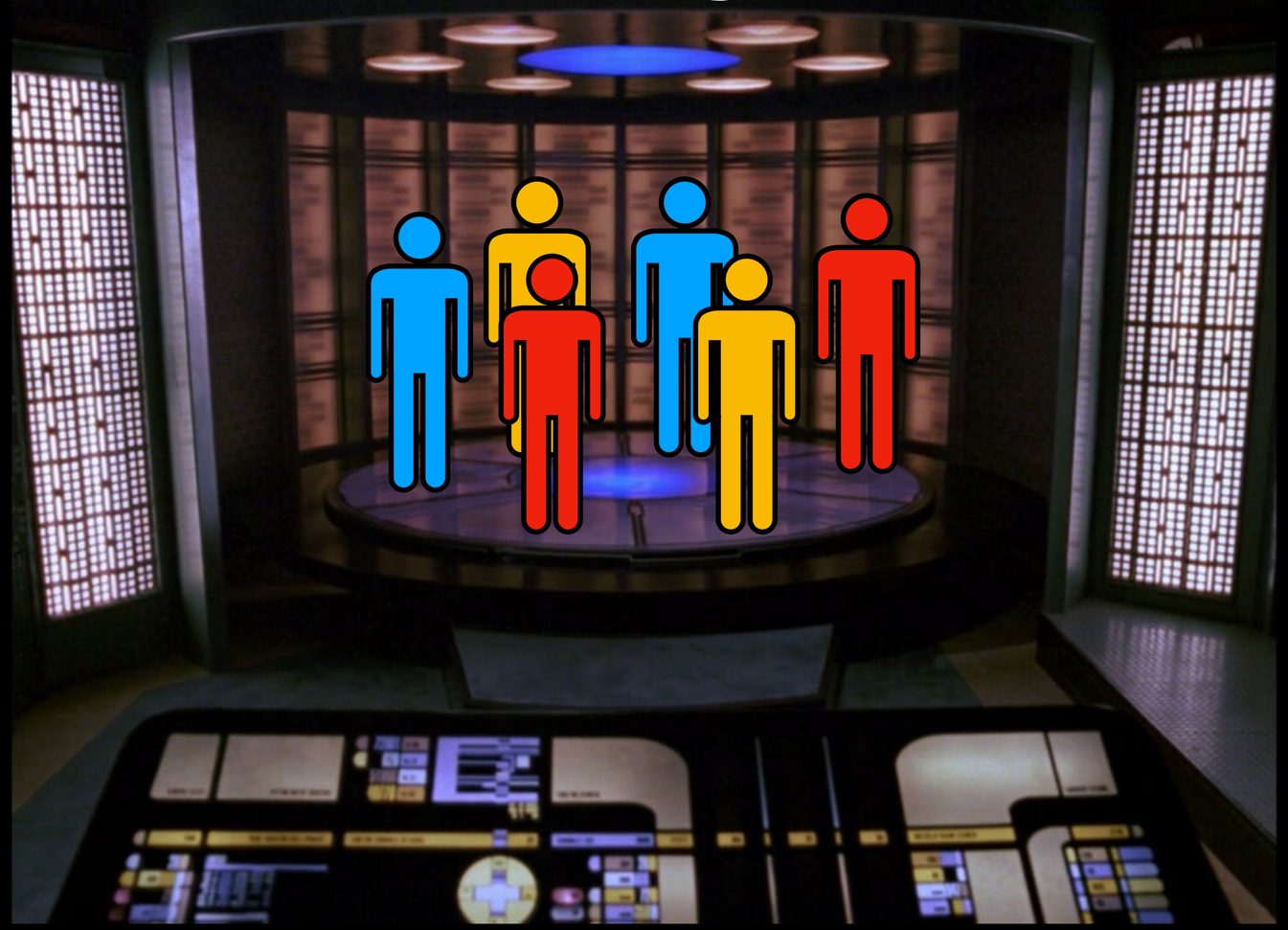
Utilization Near 100% == Resources Overloaded

Queue Latency

Saturation Causes Queue Latency

Latency of 5 seconds

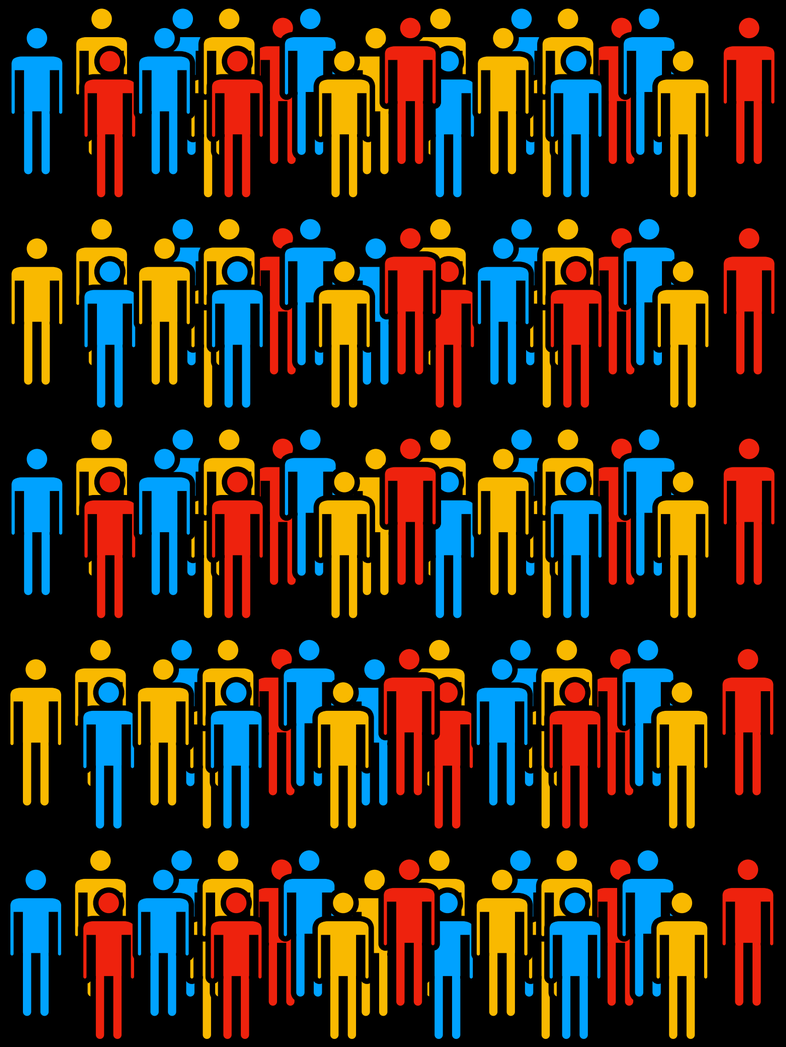


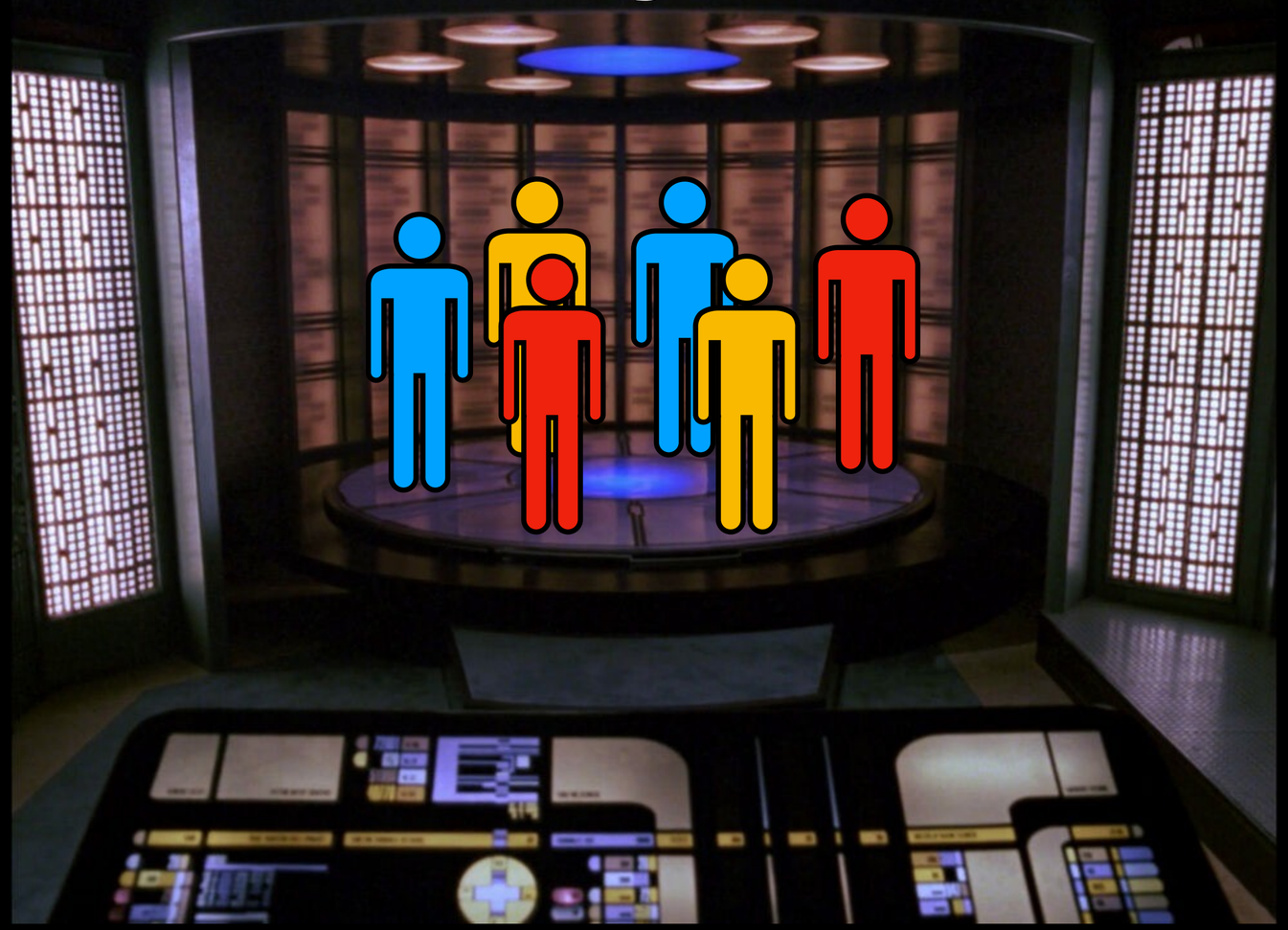


Queue Delay

Saturation Causes Service Delays

Delay of 833 seconds





690%

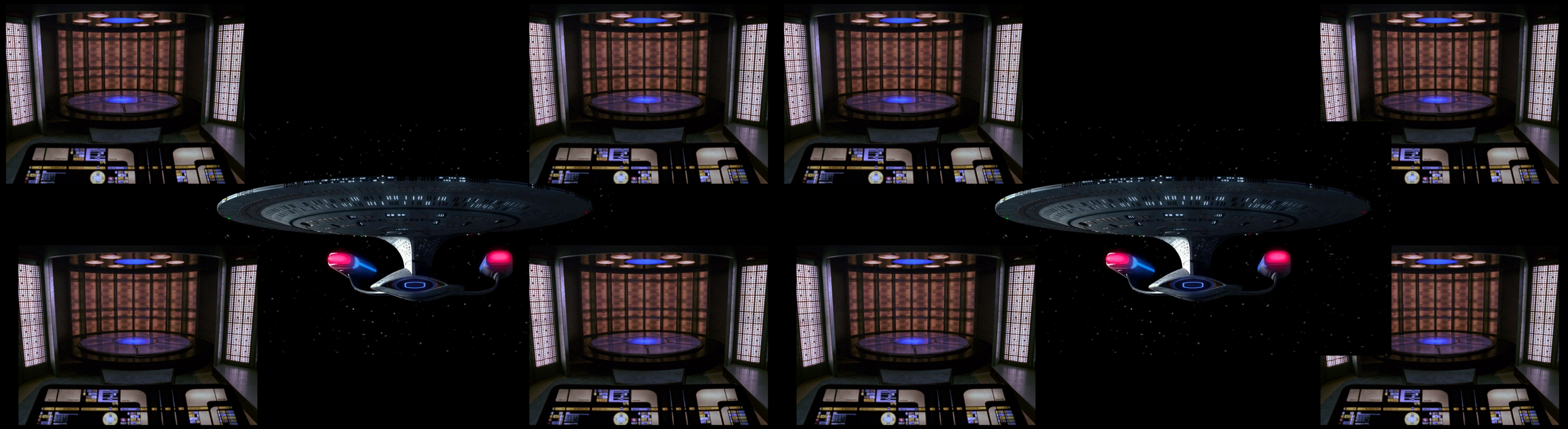
41.5 crew/sec
6 transporter sites

41.5 crew/sec
7 rooms x 6 sites

41.5 offer traffic
42 parallelism

98.8%

41.5 crew/sec
42 transporters



41.5 crew/sec
8 rooms x 6 transporter sites

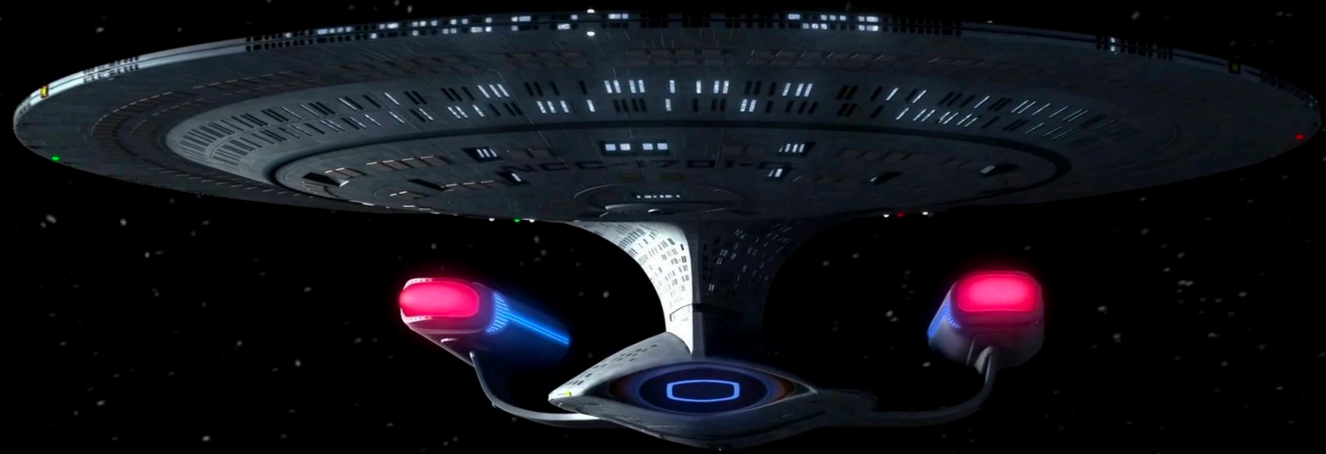
41.5 offer traffic
48 Parallelism

86.46%

41.5 crew/sec
48 Parallelism

Vertical Scale

Bigger is Better?!



Vertical scaling makes horizontal scaling more expensive

Scaling Quantum

50 crew/sec
48 Parallelism x 1 Ship

Utilization at 104%
With 48 transporters in one ship

50 offer traffic
48 Parallelism x 2 Ships

Utilization at 52%
96 transporters in two ships

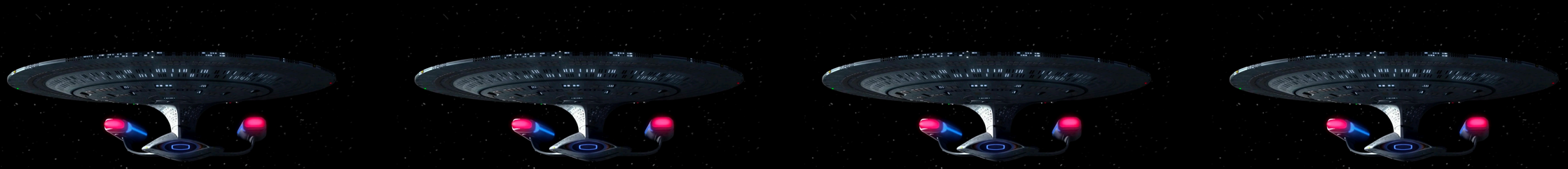
52%

50 crew/sec
96 Parallelism

Horizontal Scale

More as Needed

Horizontal scaling requires you to be mindful as well



50 crew/sec
24 Parallelism x 2 Ships

Utilization at 104%
With 48 transporters in two ships

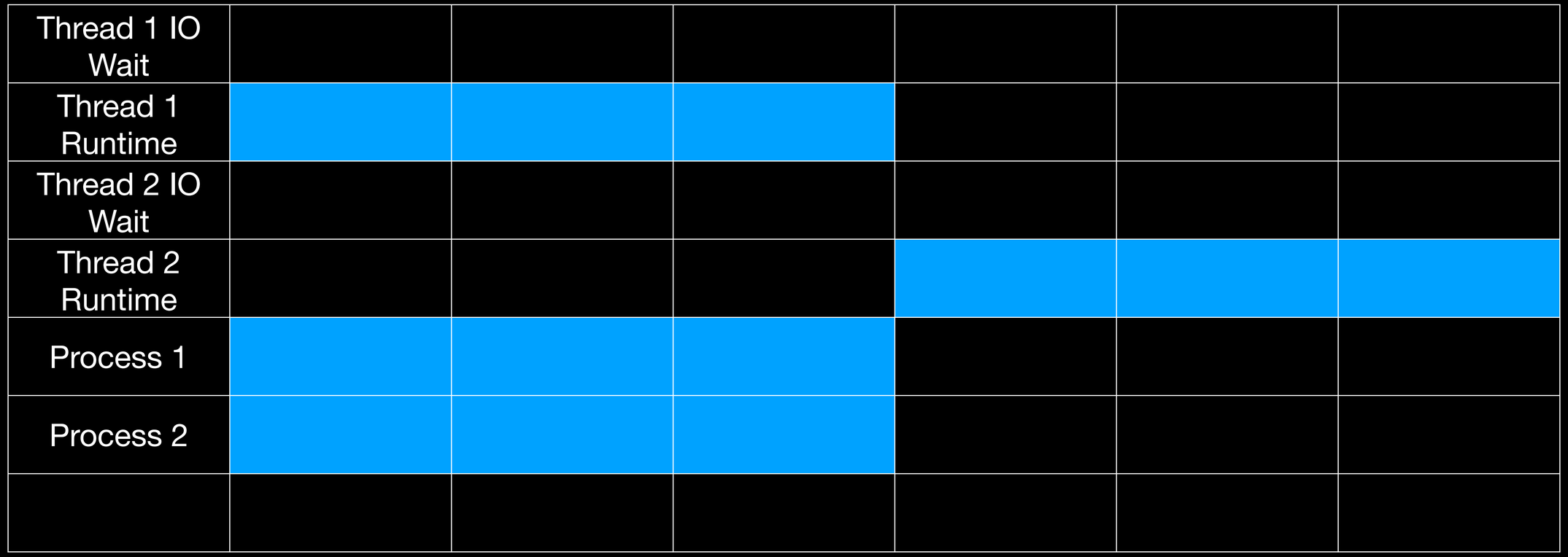
50 offer traffic
24 Parallelism x 3 Ships

Utilization at 69%
72 transporters in two ships

Parallelism vs Concurrency

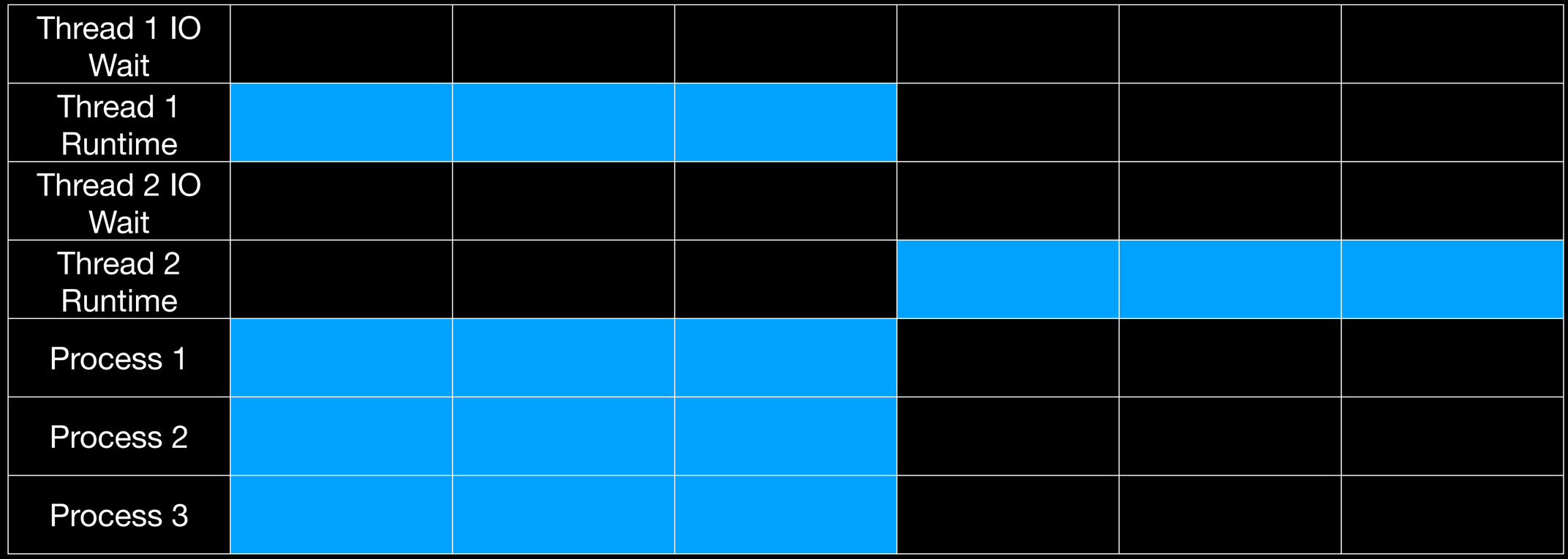
Parallelism is Concurrency

Concurrency is not exactly the same as parallelism



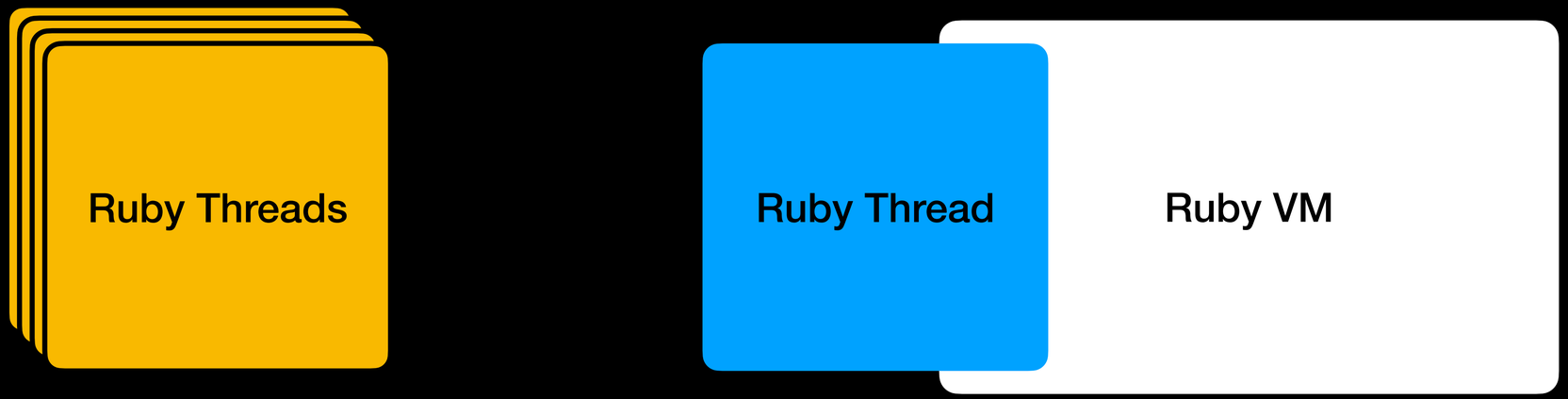
Parallelism is Concurrency

Concurrency is not exactly the same as parallelism



Benefits of Threaded Concurrency

Sharing Available Resources



Parallelism is when at least two tasks are being worked on simultaneously

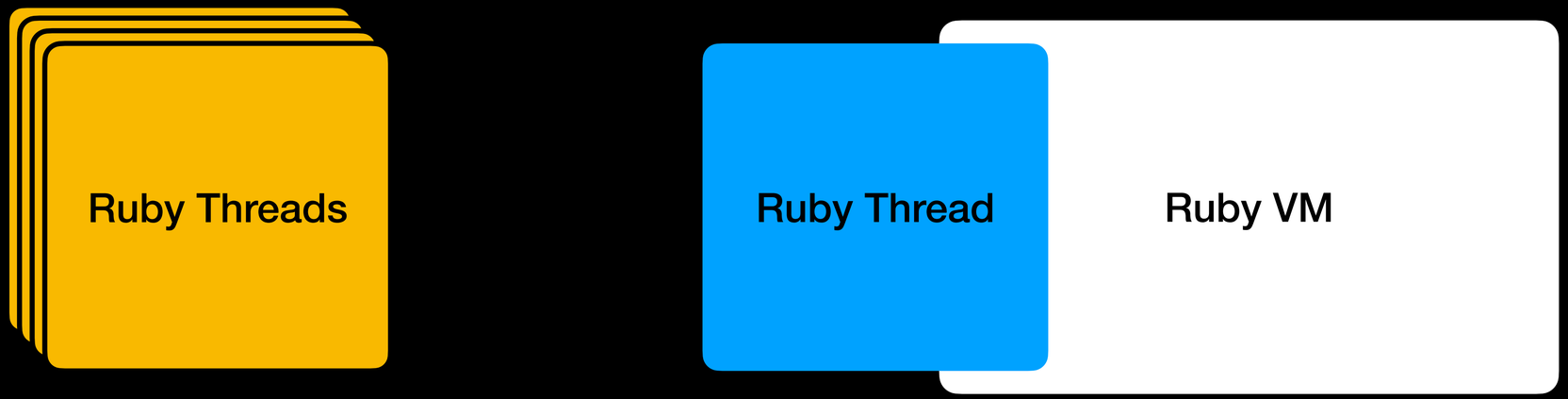
Concurrency is when multiple tasks are done in overlapping time periods

Parallelism is Concurrency

Concurrency can provide parallelism, but not always

Queueing for the GVL

Threads Lock On to the Ruby VM

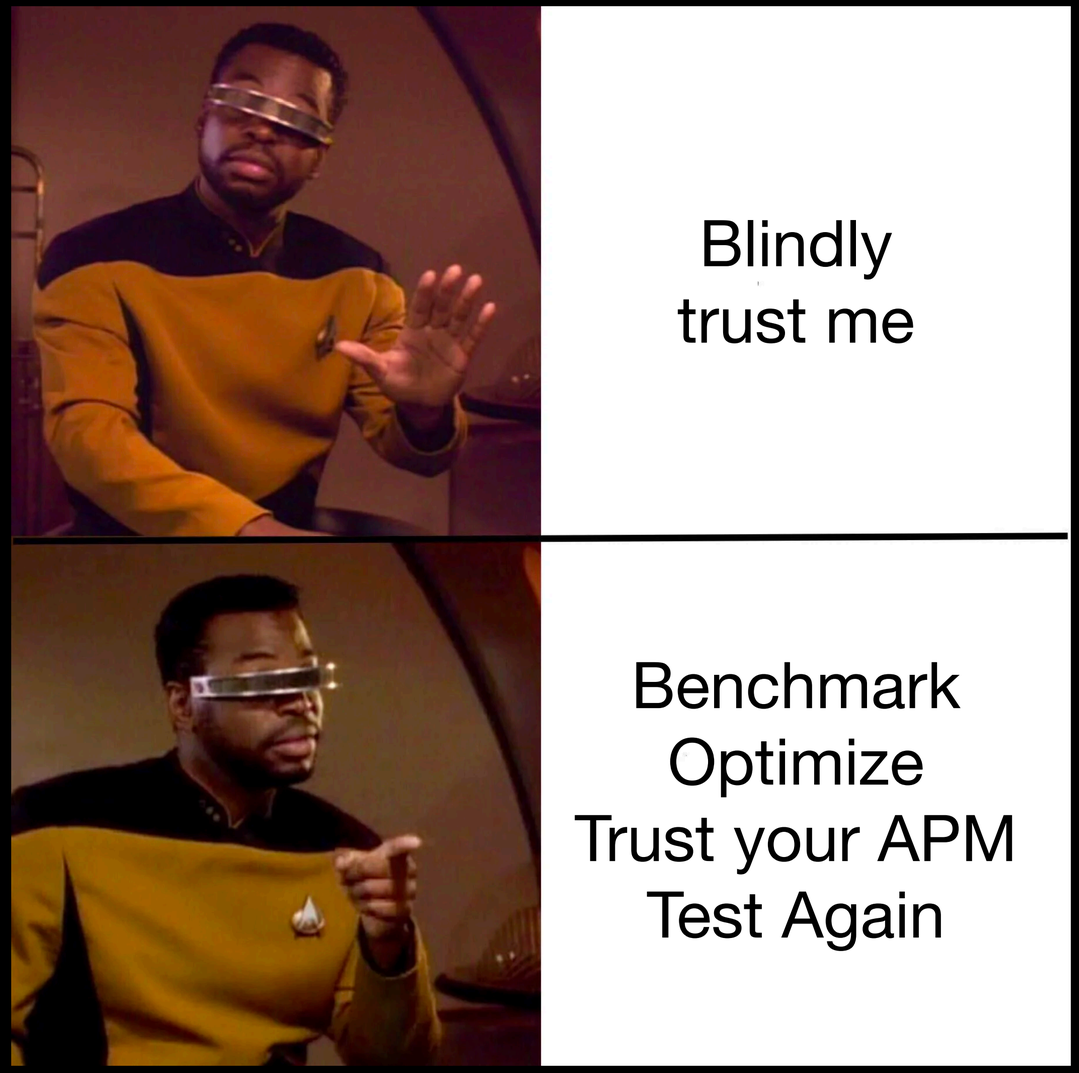


Where can our apps benefit?

- Networking I/O in CRuby is implemented in C
  - So when Ruby is waiting on I/O to come back from a
  - 3rd party API
  - database queries
  - elastic search

How much concurrency?

But you don’t have to take my word for it



Amdahl's Law

1/(1-wait_percentage+wait_percentage/Concurrency)

- Satisfying table of the parallelism of a CRuby process:

I/O Wait	Concurrency	Parallelism
< 5%	1	1
25%	5	1.25
50%	10	2
75%	16	3
90%	32	8
95%	64	16

    - Sidekiq in Practice - Nike Berkopec

Sidekiq in Practice

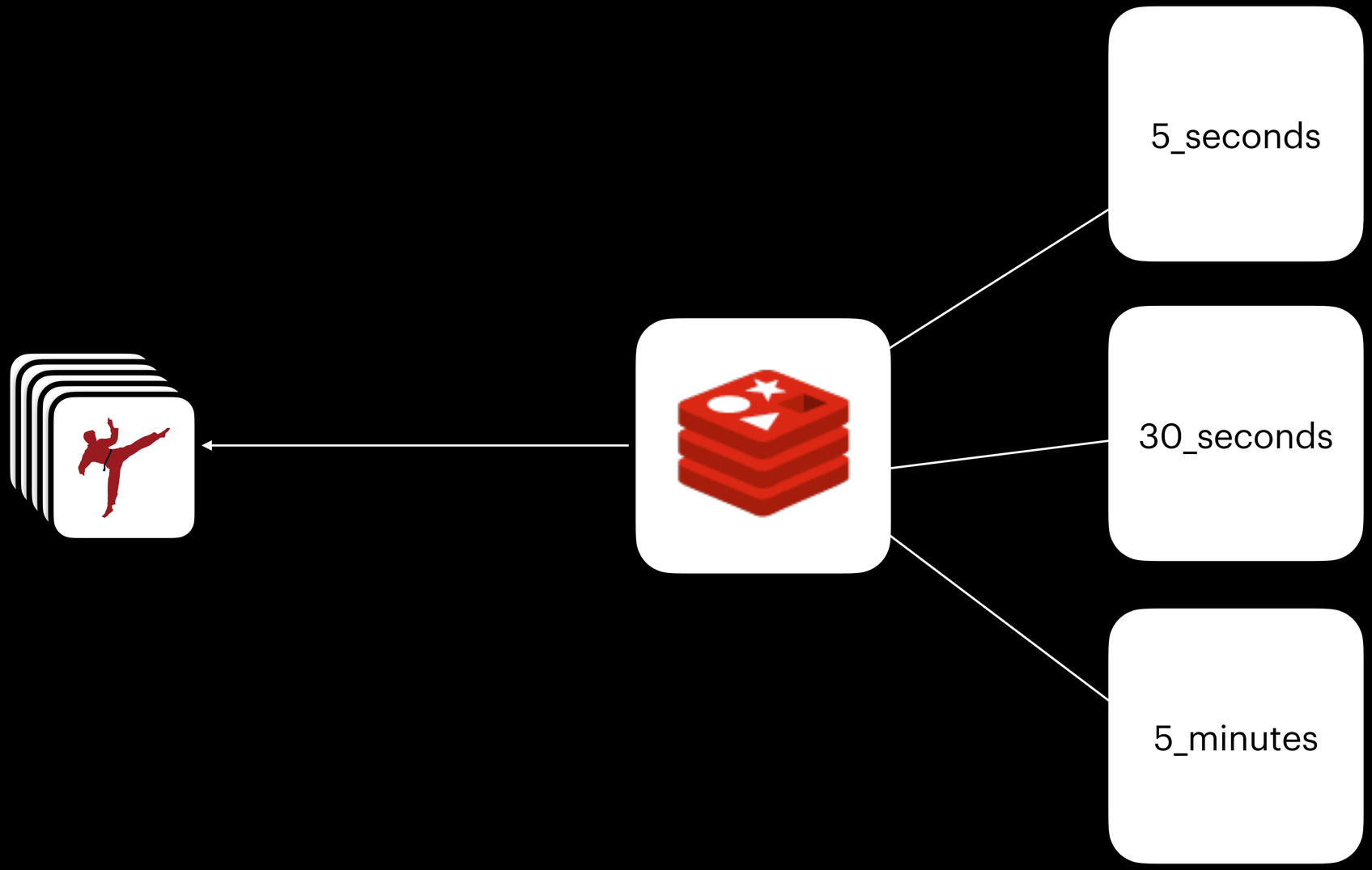
Nate Berkopec

Job Queueing: Sidekiq

Distributed Processing of a Centralized Queue

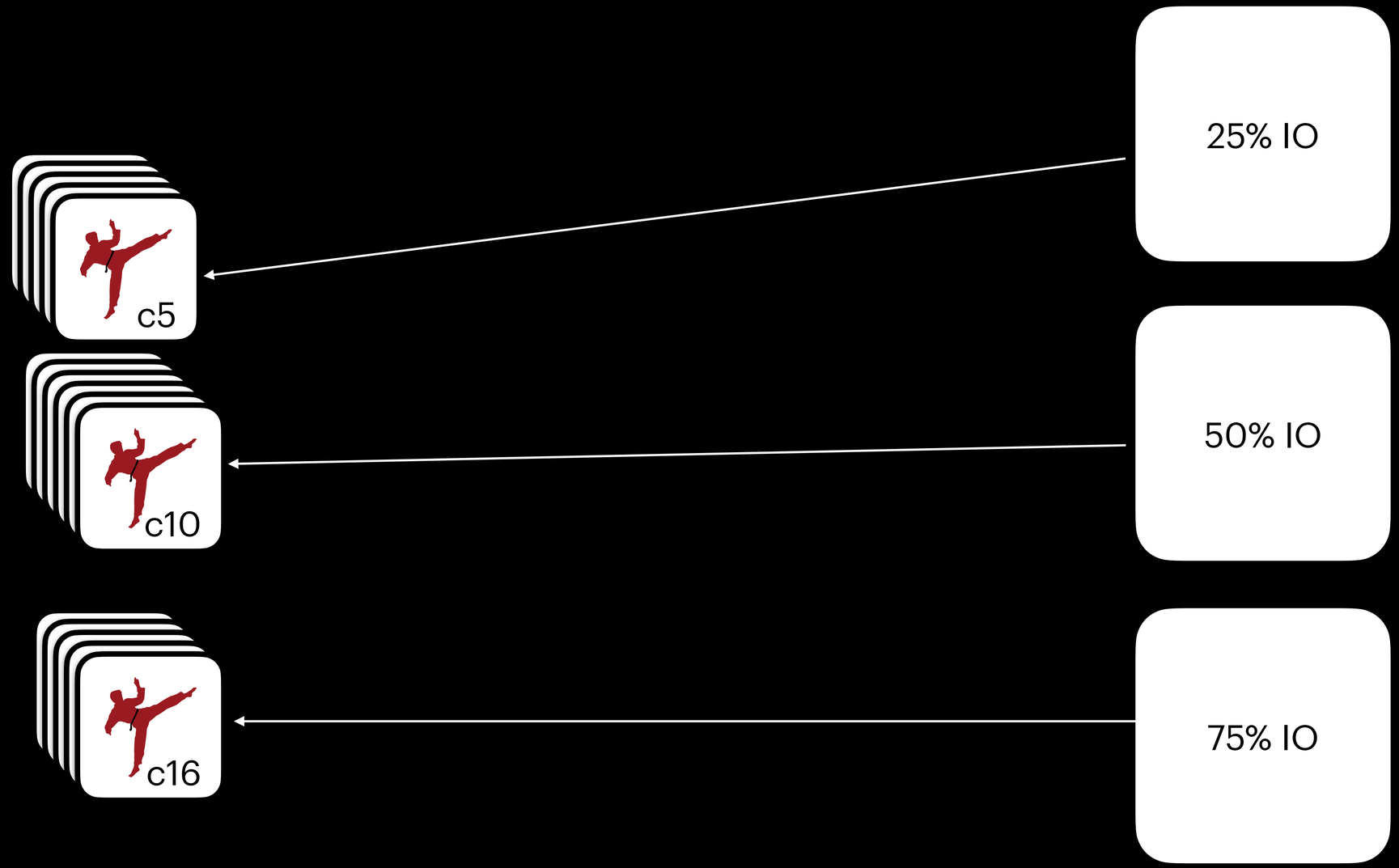
Latency Based Queueing

Latency Based Scaling



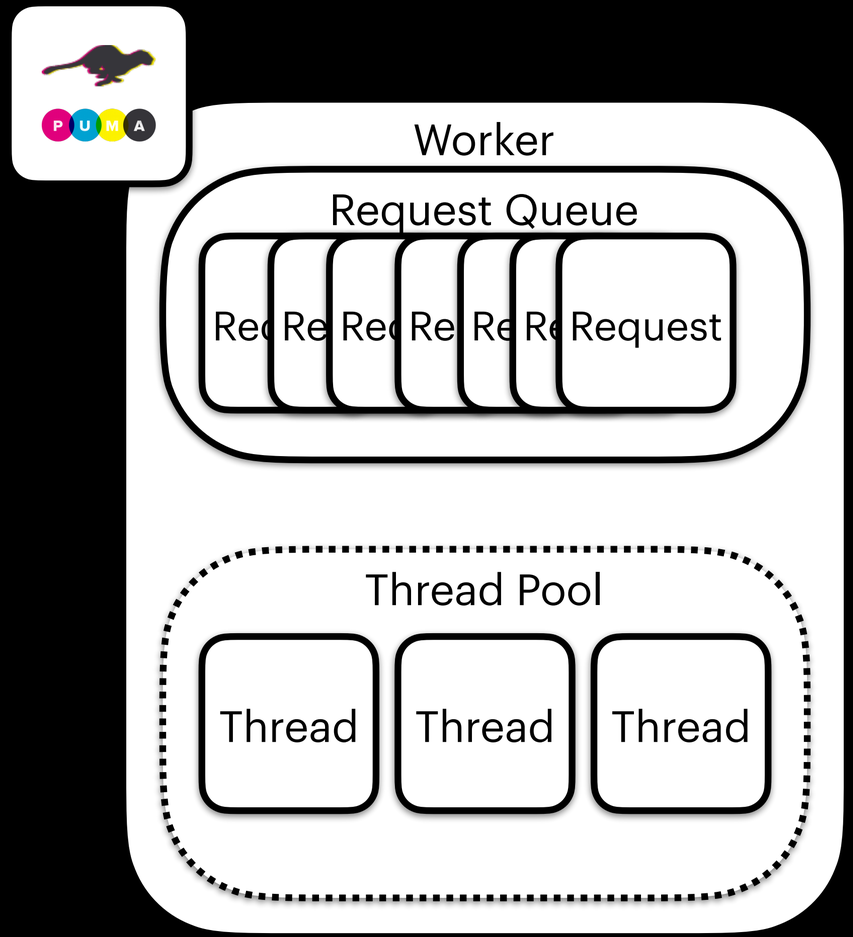
I/O Wait Based Queueing

Similar I/O Means Similar Concurrency



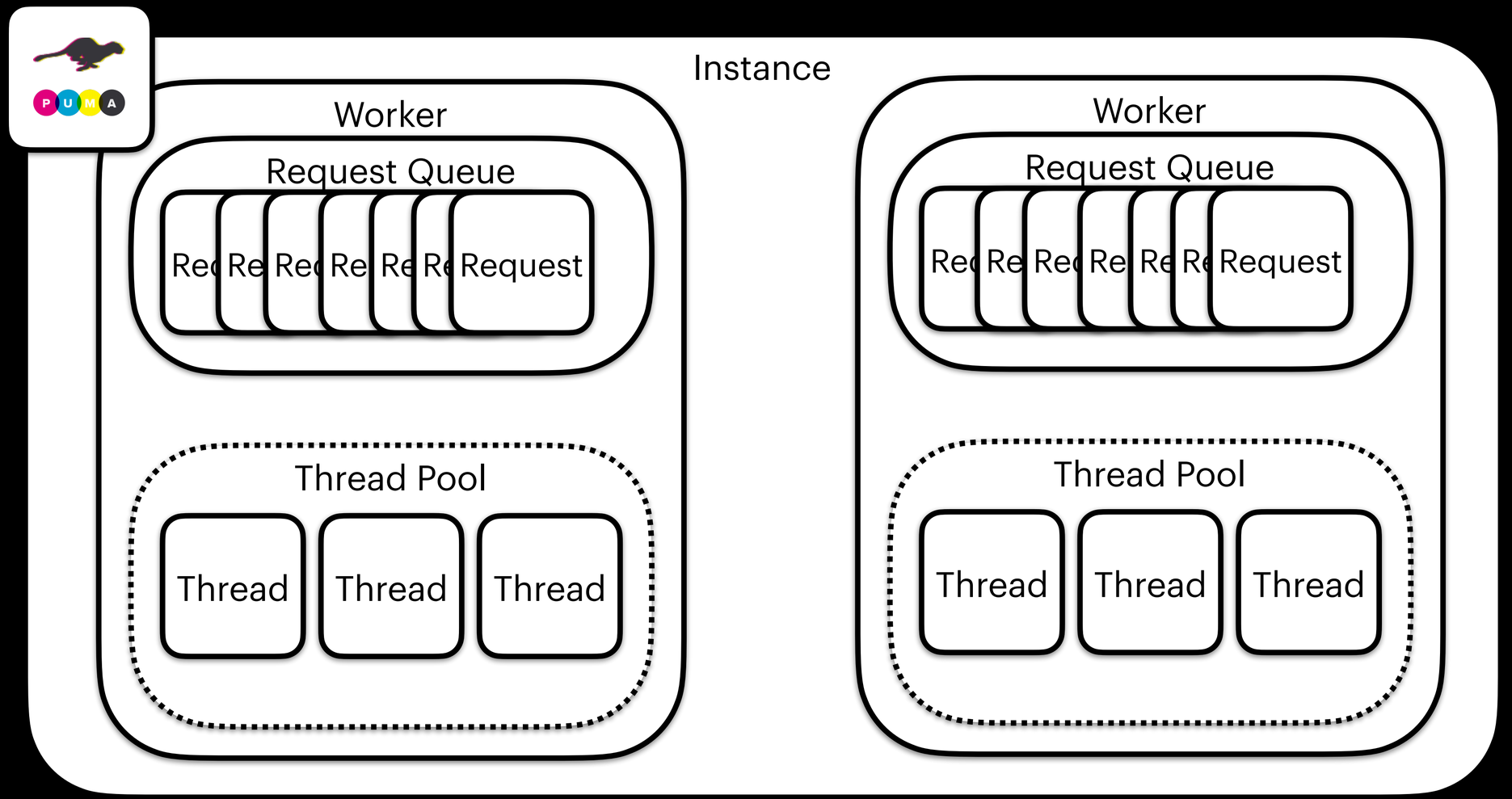
Request Queueing: Puma

Per Process Queueing



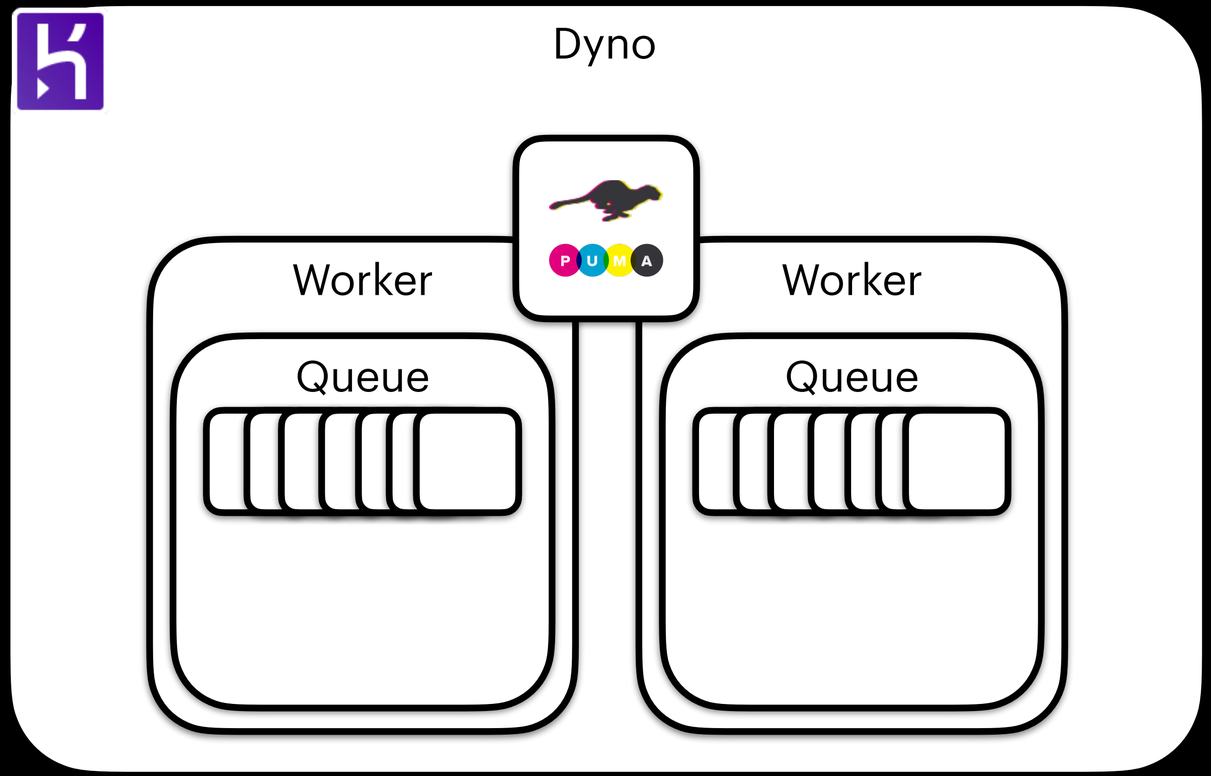
Request Queueing: Puma

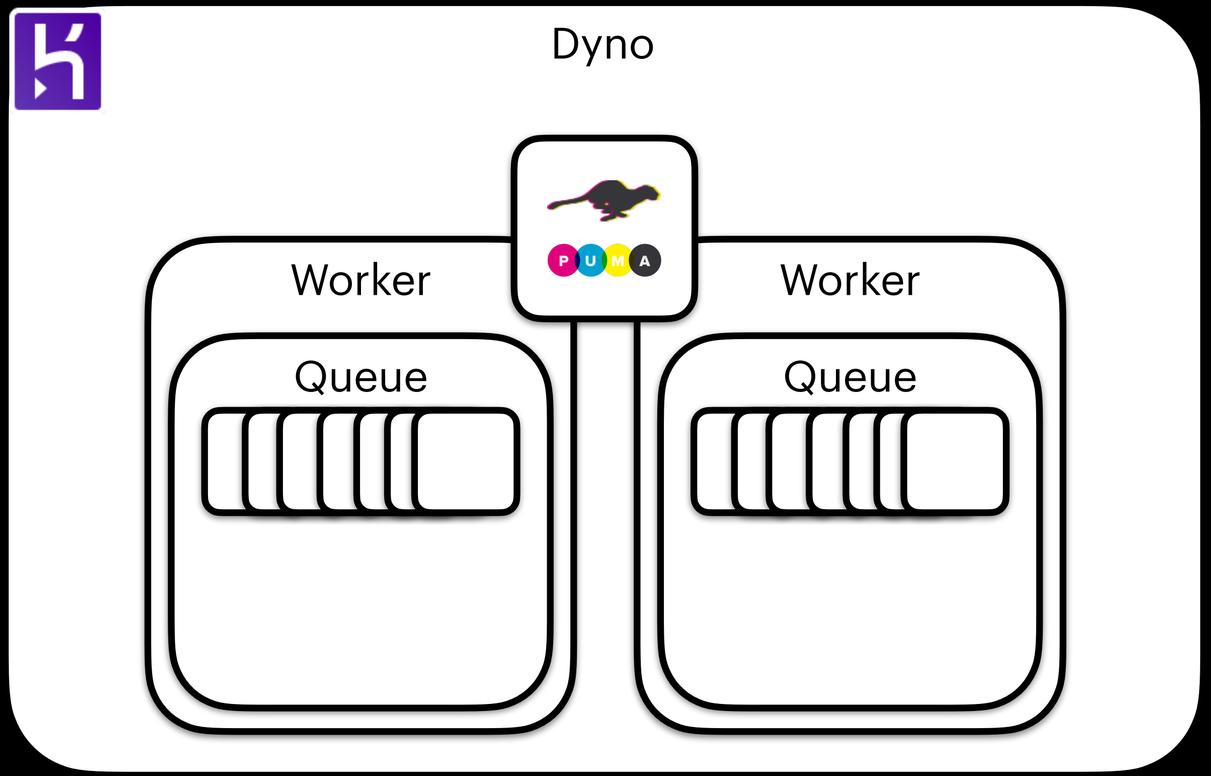
Per Process Queueing



Heroku Intelligent Routing

Random Routing





Remember Why We Queue

Profit!

Customer Satisfaction

Timeouts and Rejection

Waiting too long causes frustration

- Queueing delay can make users feel like things aren’t working

Other Queues

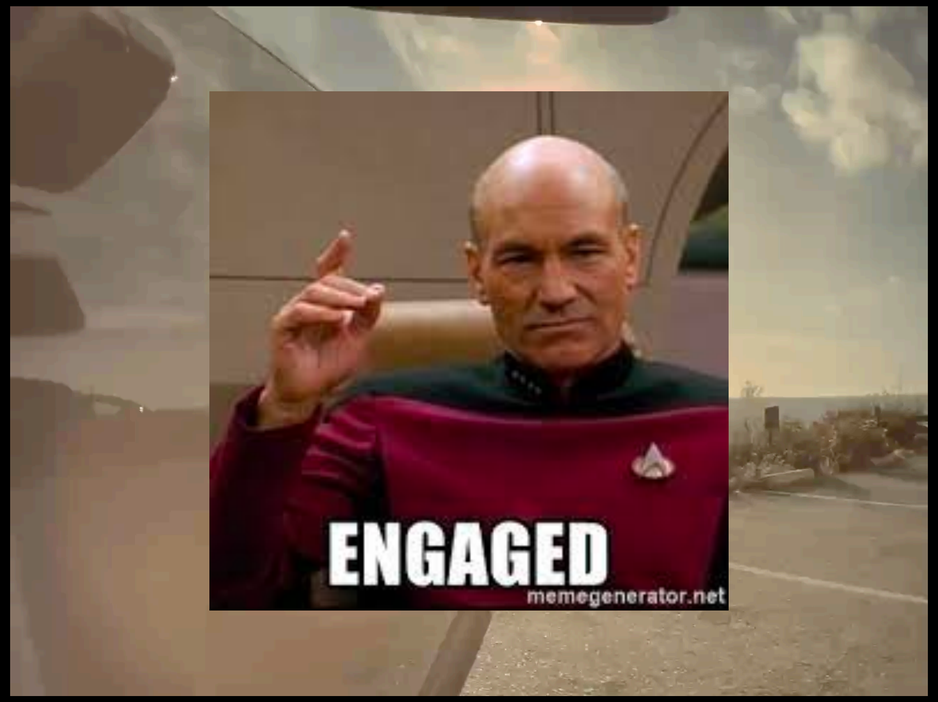
Queue are everywhere

- System Resource Queueing
- Database Query Queueing

Other Queues

Queue are everywhere

- System Resource Queueing
- Database Query Queueing



Finding Balance

- Scale to reduce utilization to avoid saturation and queueing
- Be mindful of your system resources and scaling quantum
- Per process application footprint
- Per thread memory fragmentation

All Good Things…

GENE
RODDENBERRY
1921-1991

All Good Things…

EMLYN THOMAS
BOWEN III
1953-2022

Justin Bowen

Tons Of Fun

@TonsOfFun111

TonsOfFun

CTO Consultant @ SVSG.co

Director of Engineering @ nSightSurgical.ai

Parallelism Examples: GitHub.com/TonsOfFun/RailsConf2022