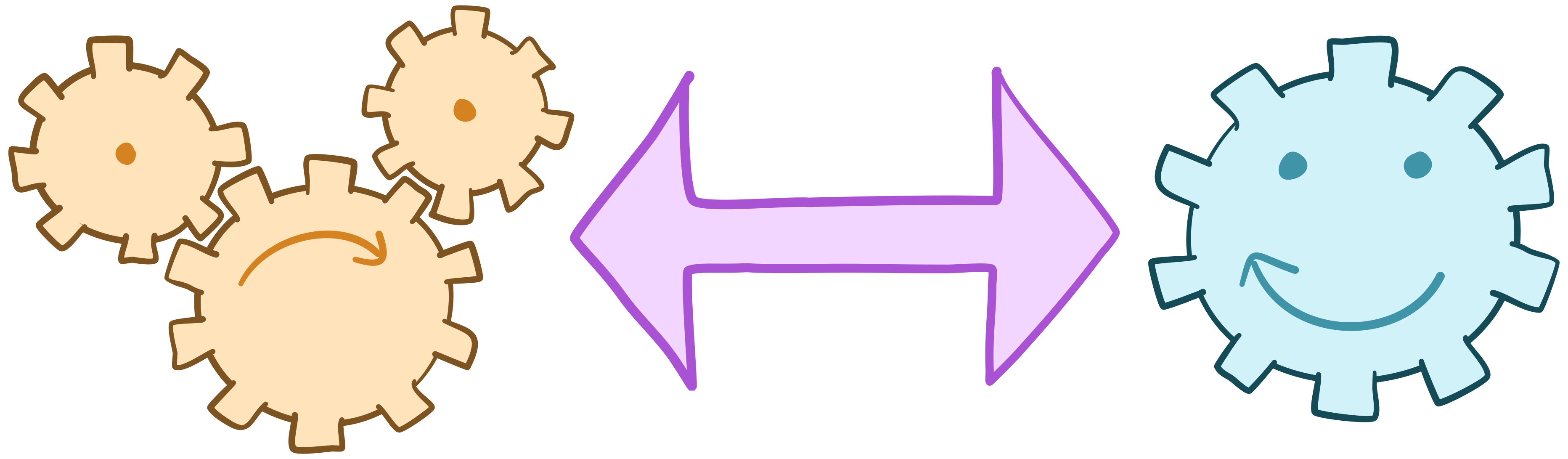
Coupling Techniques

for Complex Control Problems

Ziv Scully
Carnegie Mellon University

Sid Banerjee
Cornell University



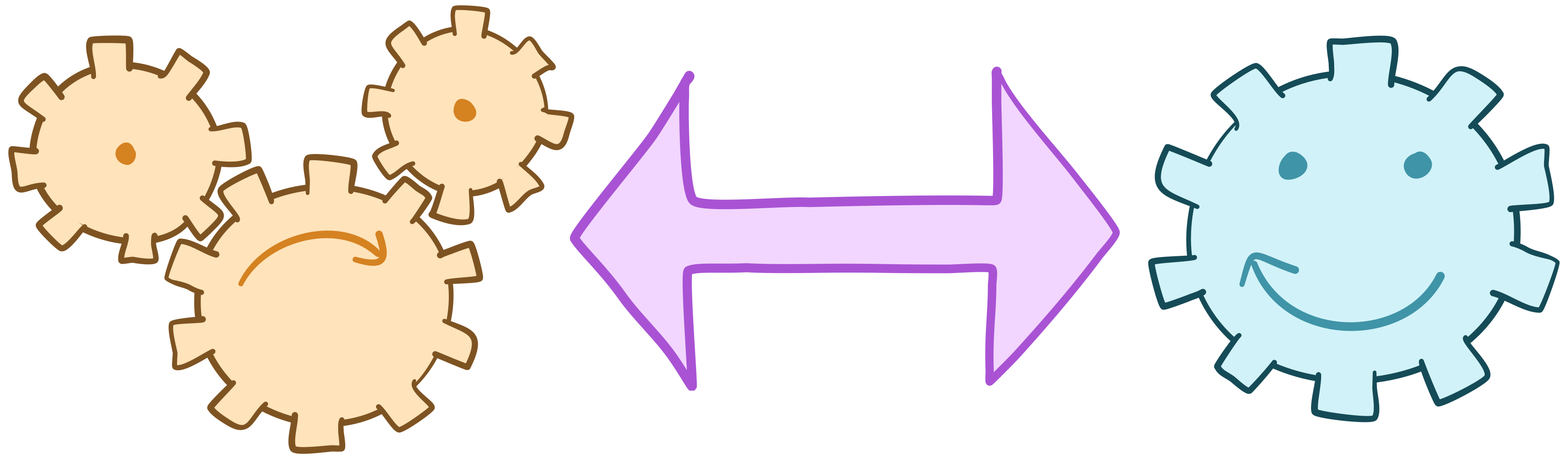
Part 2 of 2

Coupling Techniques

for Complex Control Problems

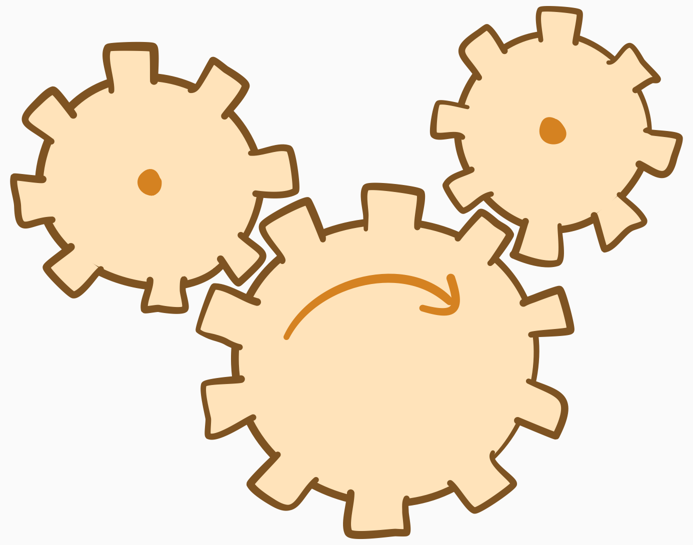
Ziv Scully
Carnegie Mellon University

Sid Banerjee
Cornell University



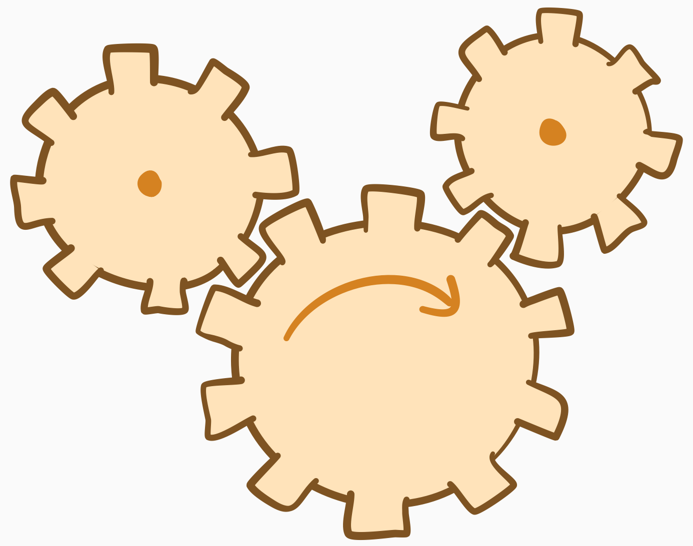
What is coupling?

complex system *X*



What is coupling?

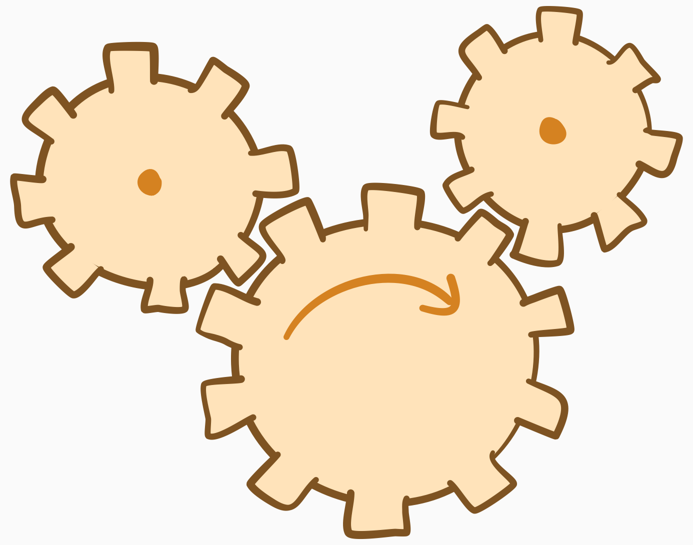
complex system *X*

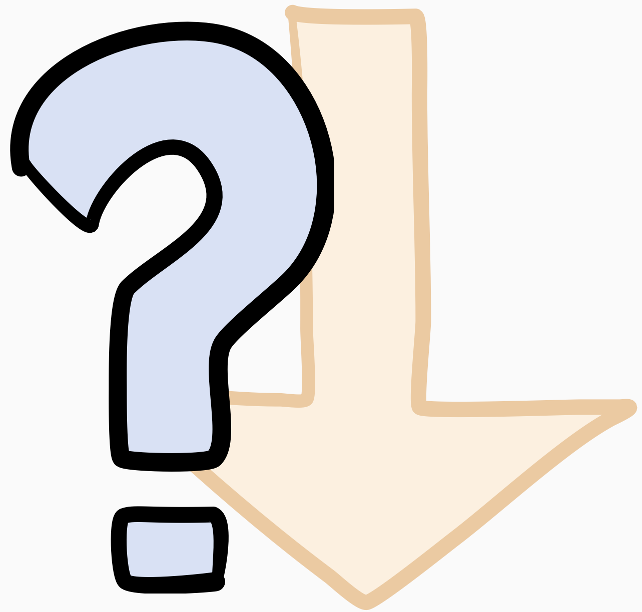


Goal: answer a question about *X*
(approximate is okay)

What is coupling?

complex system *X*

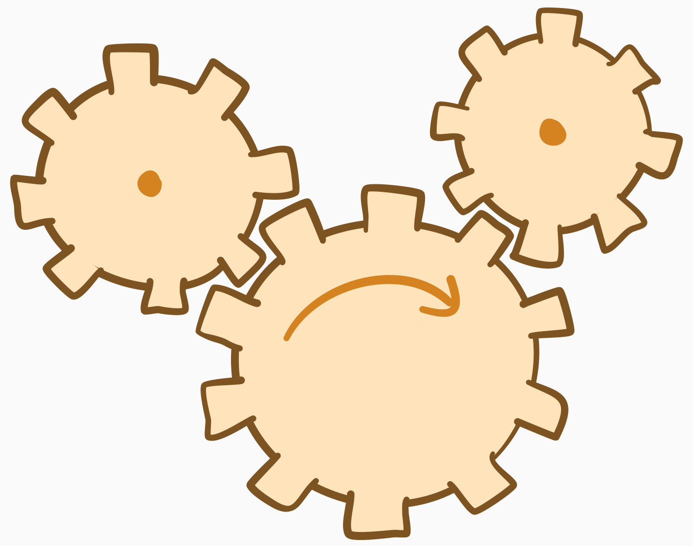


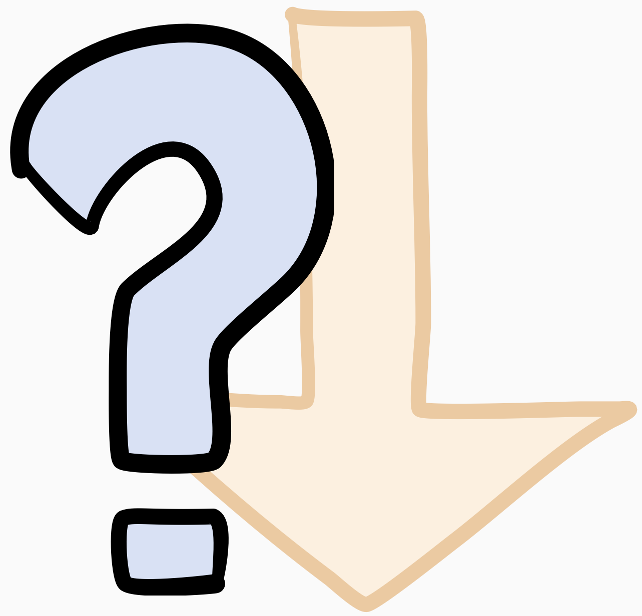


Goal: answer a question about *X*
(approximate is okay)

What is coupling?

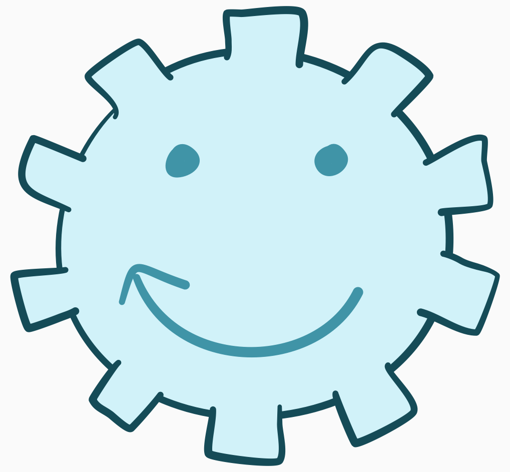
complex system X





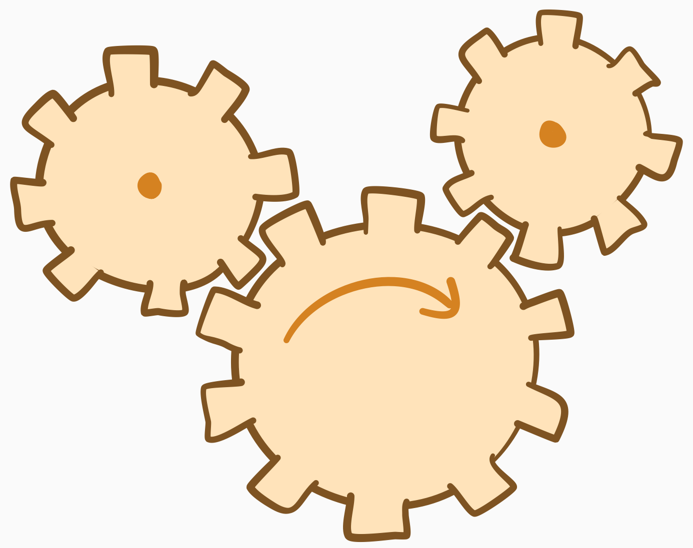
Goal: answer a question about X
(approximate is okay)

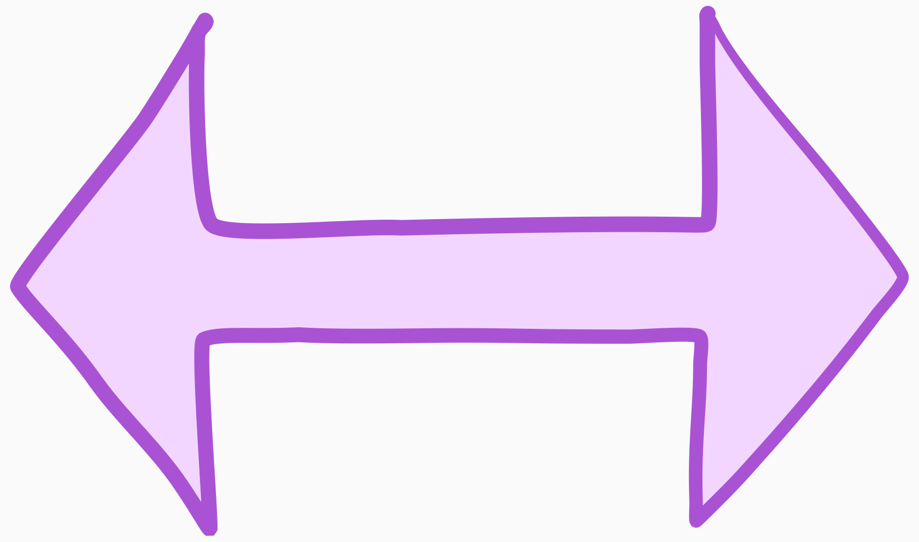
easy system Y



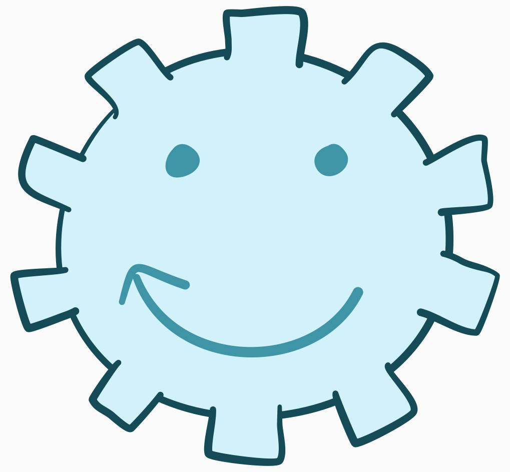
What is coupling?

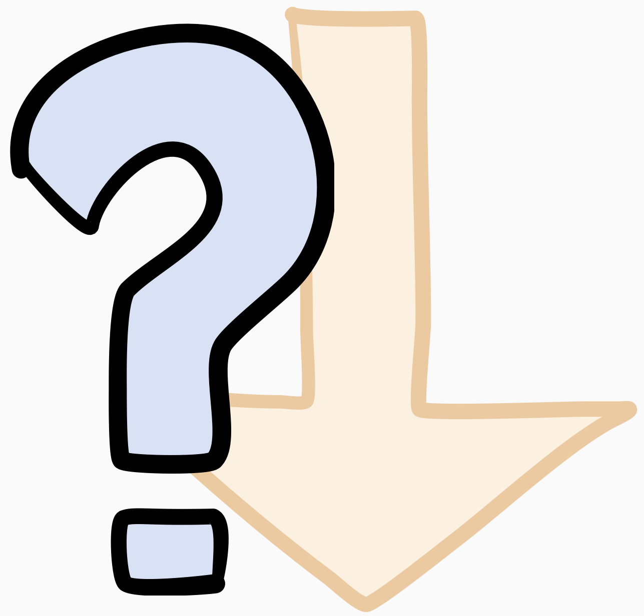
complex system X





easy system Y

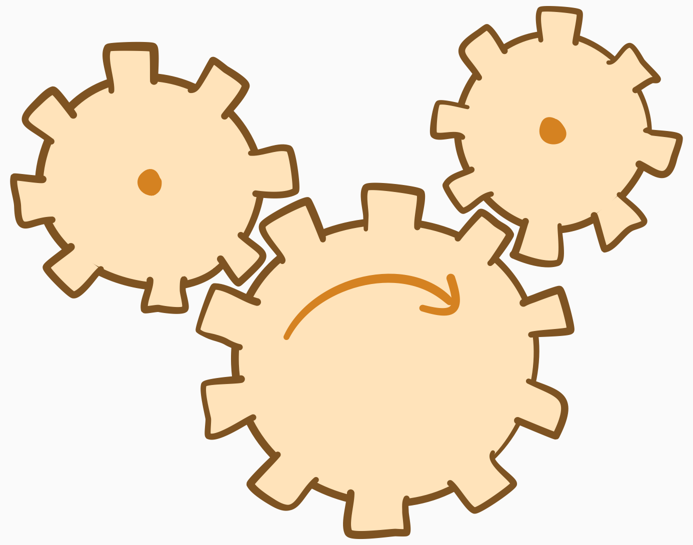


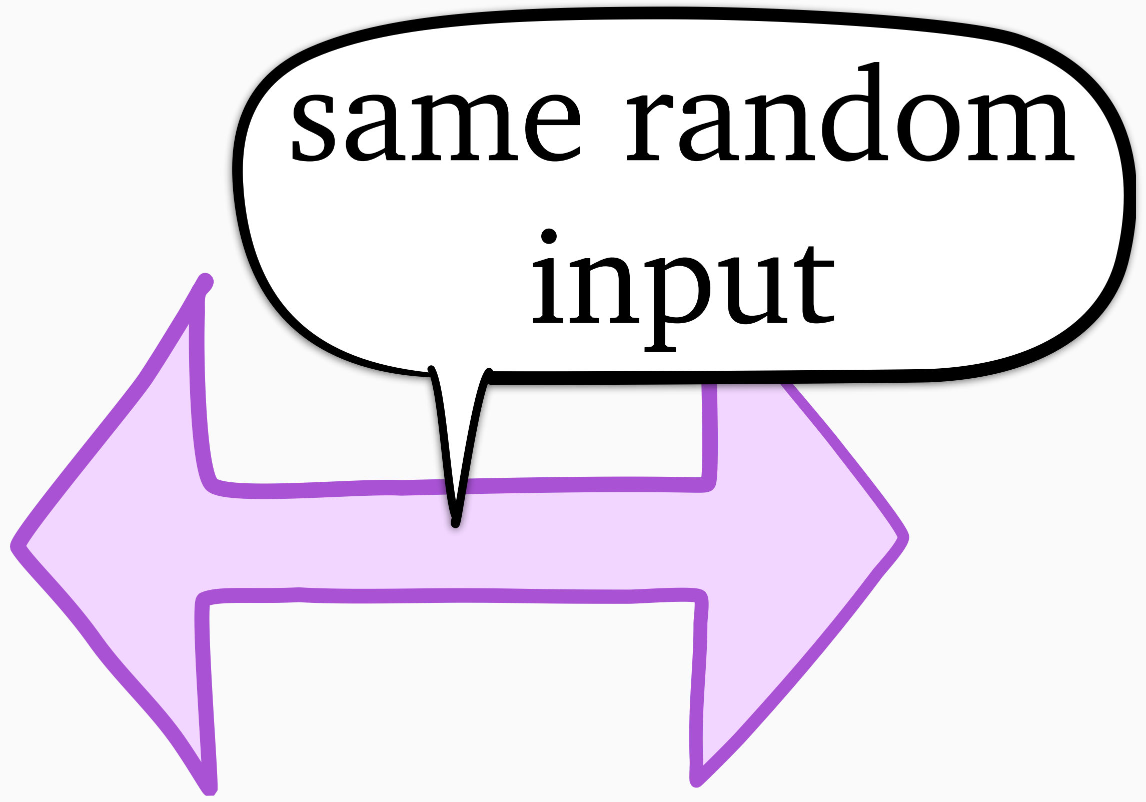


Goal: answer a question about X
(approximate is okay)

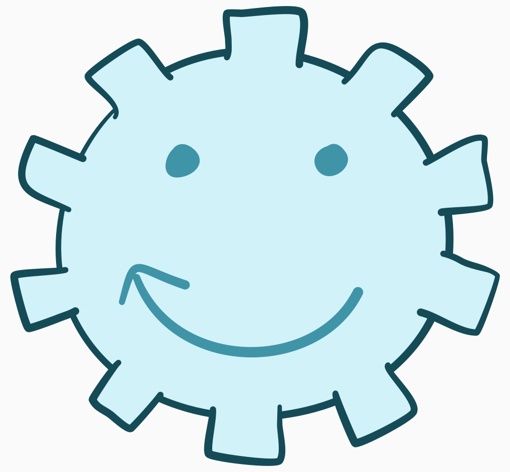
What is coupling?

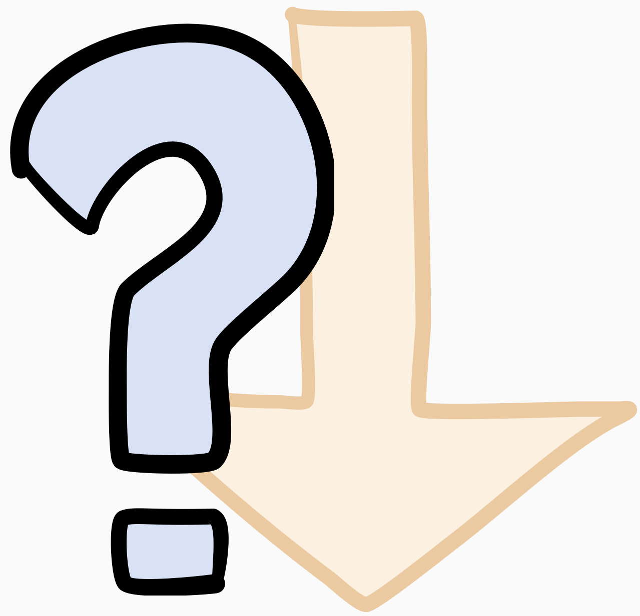
complex system X





easy system Y



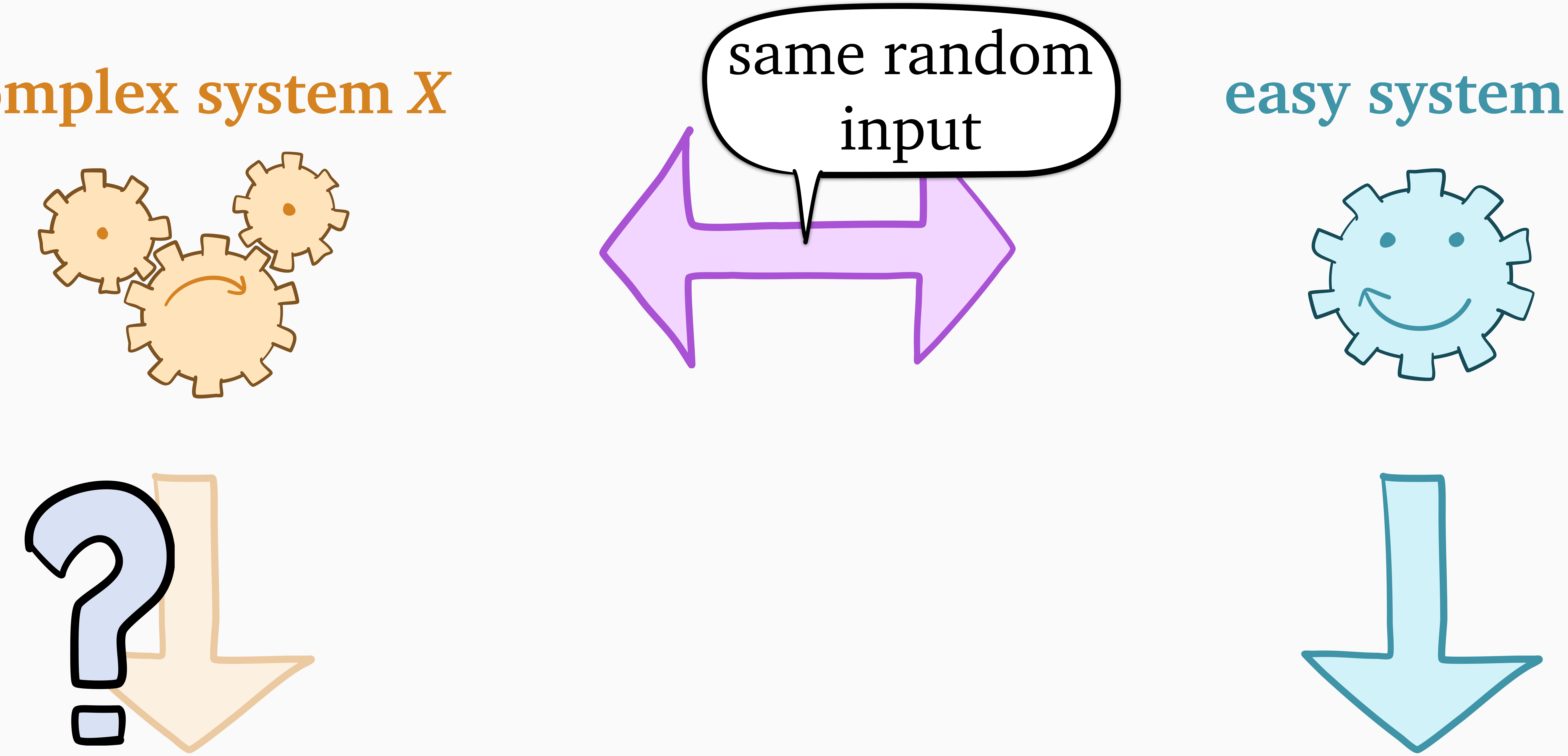


Goal: answer a question about X
(approximate is okay)

What is coupling?

complex system X

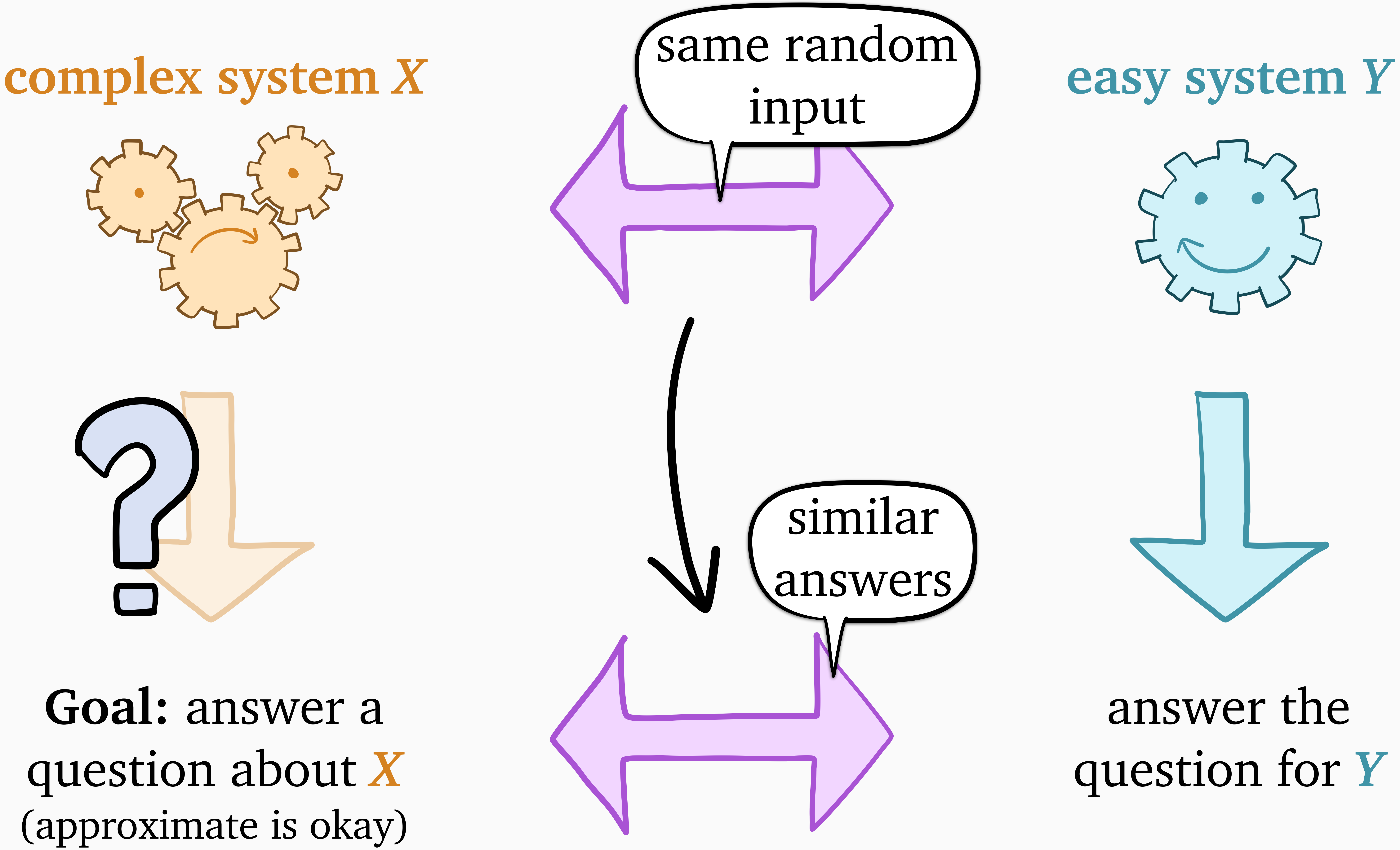
easy system Y



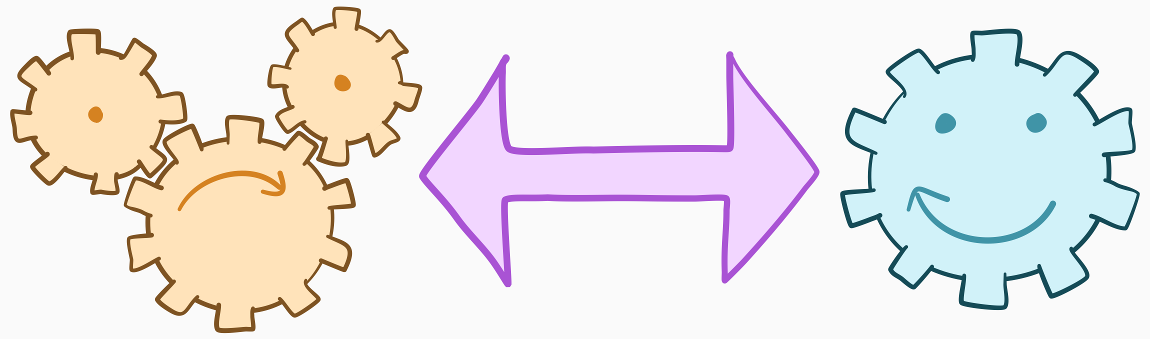
Goal: answer a question about X
(approximate is okay)

answer the question for Y

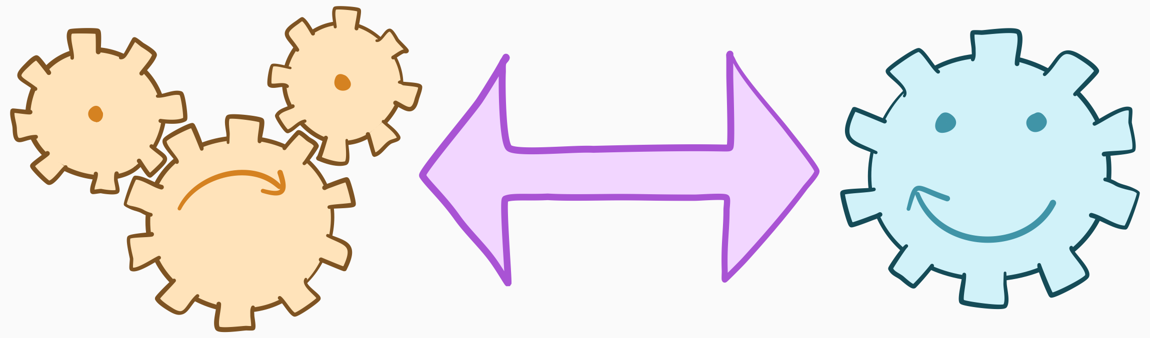
What is coupling?



Classifying coupling techniques

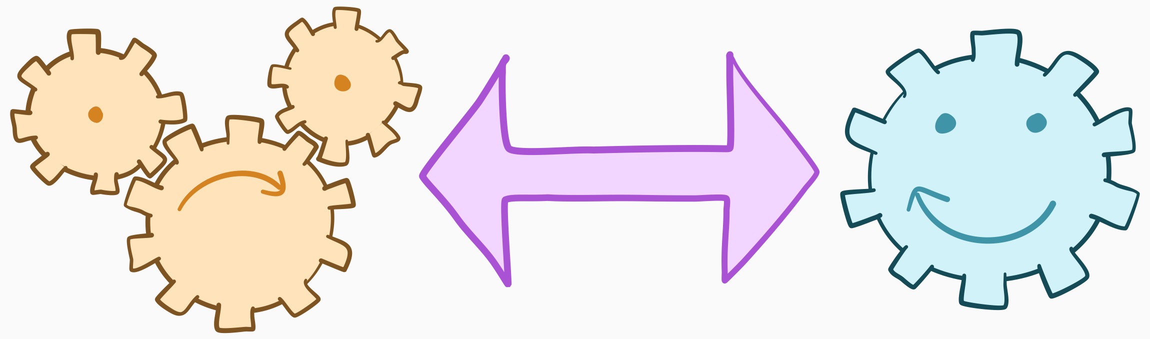


Classifying coupling techniques



How does ***Y*** make ***X*** easier?

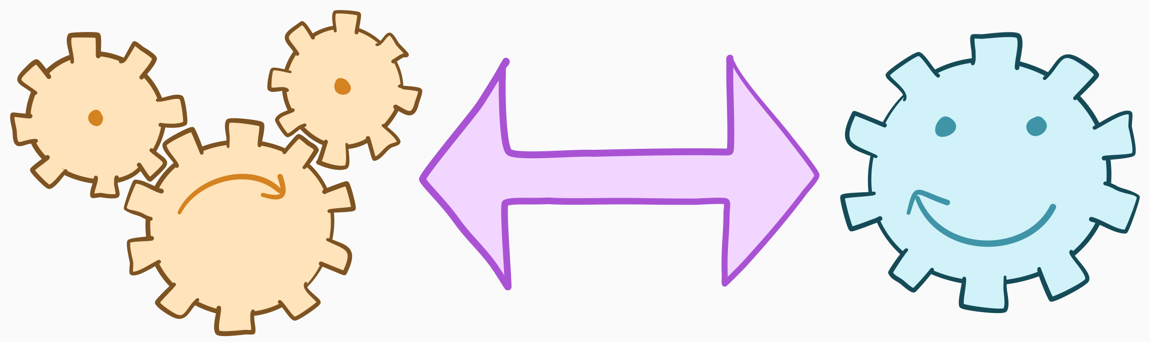
Classifying coupling techniques



In what sense are X and Y close?

How does Y make X easier?

Classifying coupling techniques

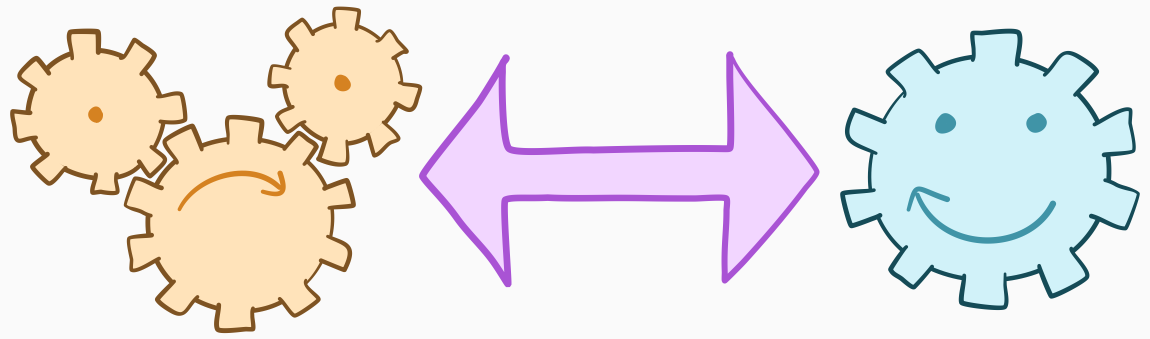


1. More information

In what sense are X and Y close?

How does Y make X easier?

Classifying coupling techniques



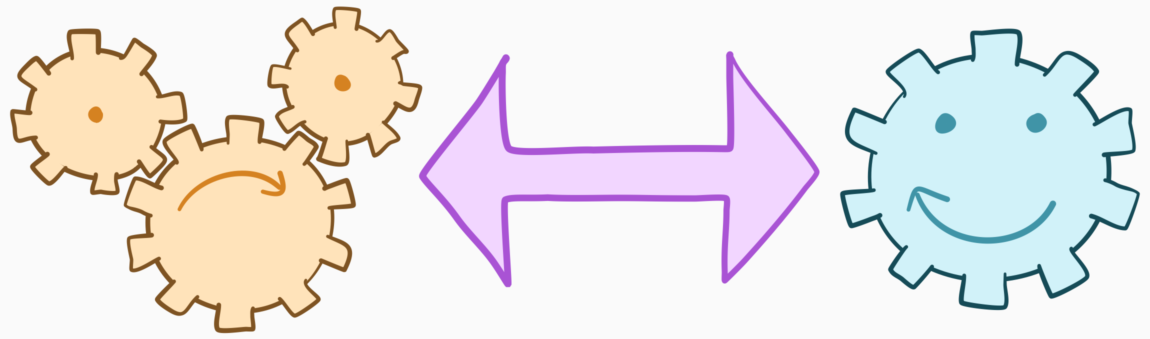
1. More information

2. Fewer constraints

In what sense are $\textcolor{orange}{X}$ and Y close?

How does Y make X easier?

Classifying coupling techniques



1. More information

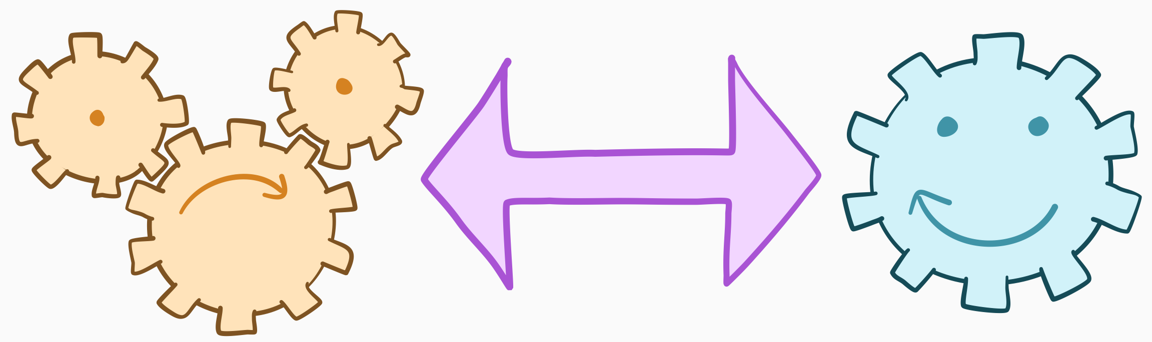
2. Fewer constraints

3. Simpler dynamics

In what sense are *X* and *Y* close?

How does *Y* make *X* easier?

Classifying coupling techniques



A. Every sample path

1. More information

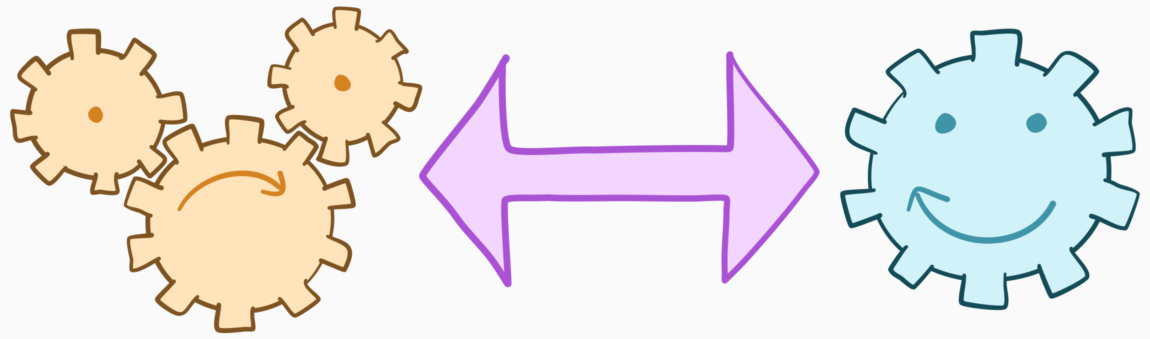
In what sense are *X* and *Y* close?

2. Fewer constraints

How does *Y* make *X* easier?

3. Simpler dynamics

Classifying coupling techniques

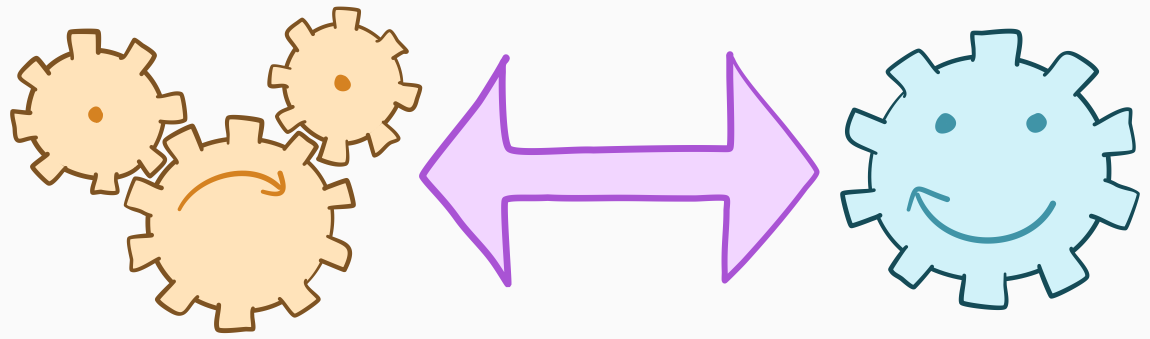


	A. Every sample path	B. Steady-state distribution
1. More information		
2. Fewer constraints		
3. Simpler dynamics		

In what sense are X and Y close?

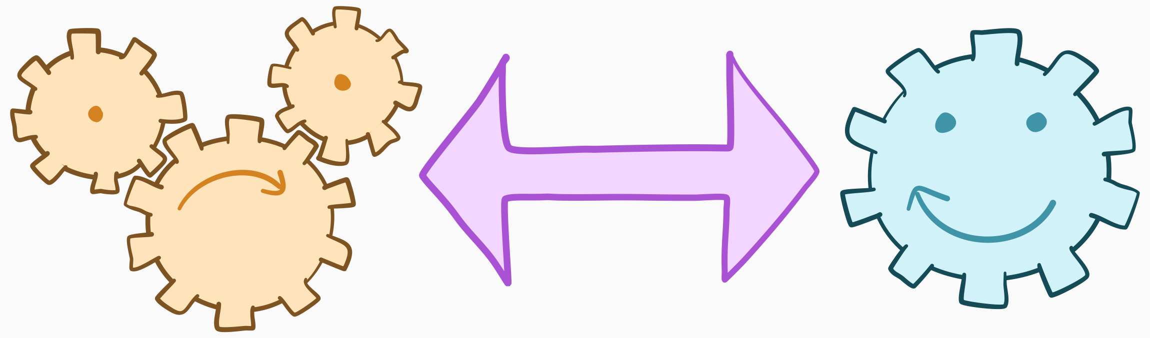
How does Y make X easier?

Classifying coupling techniques



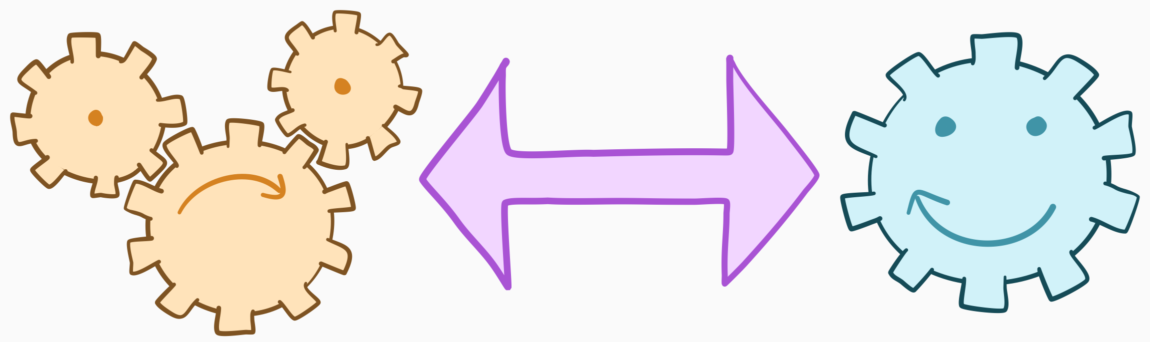
	A. Every sample path	B. Steady-state distribution
1. More information	A1	B1
2. Fewer constraints	A2	B2
3. Simpler dynamics	A3	B3

Classifying coupling techniques



	A. Every sample path	B. Steady-state distribution
1. More information	A1 BIG online knapsack (via compensated coupling)	B1
2. Fewer constraints	A2 M/M/k vs. M/M/1 Online knapsack (via constraints-to-costs)	B2
3. Simpler dynamics	A3 SIS epidemics Queues with redundancy	B3

Classifying coupling techniques



	A. Every sample path	B. Steady-state distribution
1. More information	A1 BIG online knapsack (via compensated coupling)	B1
2. Fewer constraints	A2 M/M/k vs. M/M/1 Online knapsack (via constraints-to-costs)	B2
3. Simpler dynamics	A3 SIS epidemics Queues with redundancy	B3

Overview

✓ Part 1

Survey 1:
Sample-Path Coupling

In-Depth Study 1:
Online Resource Allocation

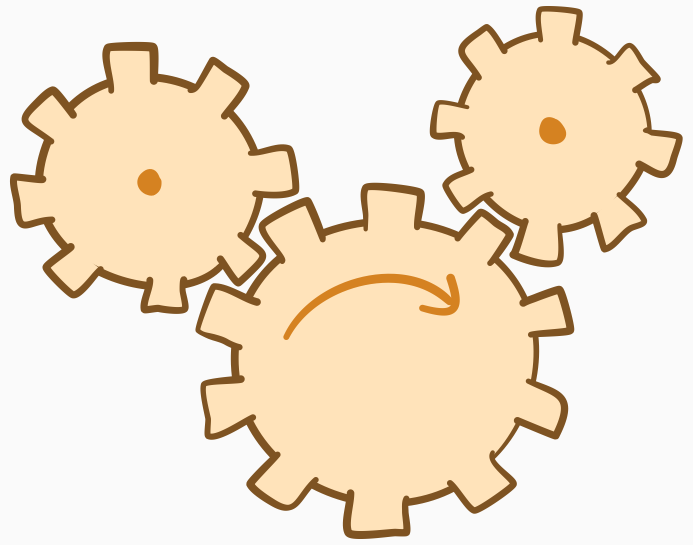
Part 2

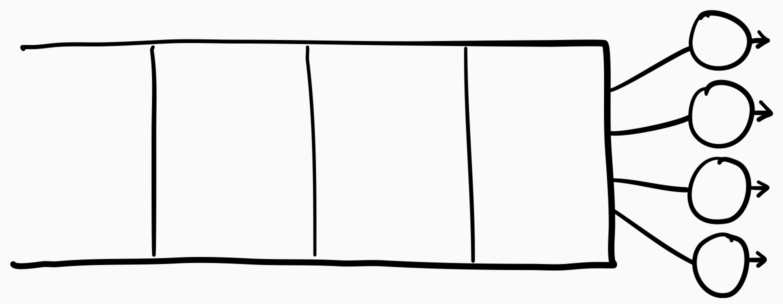
Survey 2:
Steady-State Coupling

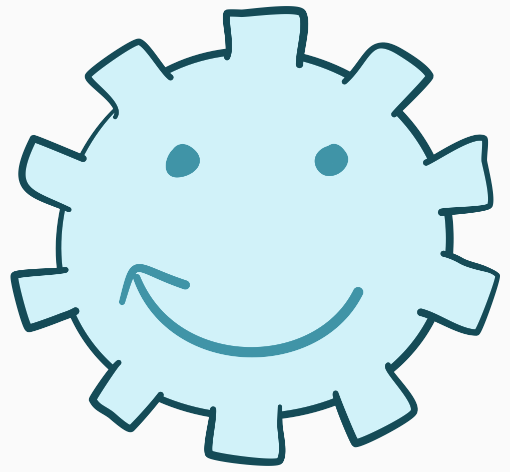
In-Depth Study 2:
Gittins in the M/G/k

Survey 2:
Steady-State Coupling

M/M/k in steady-state

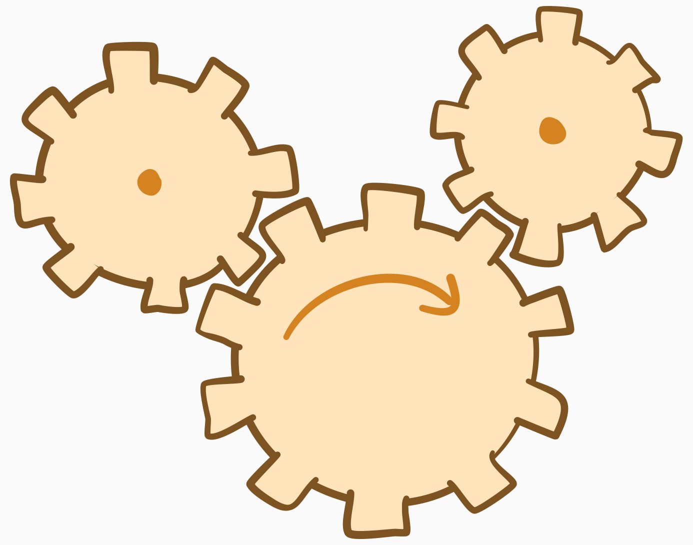


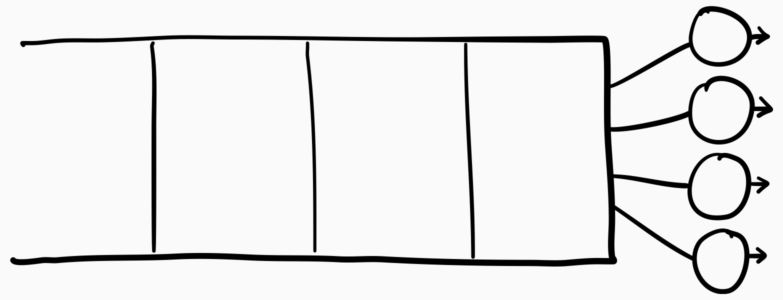


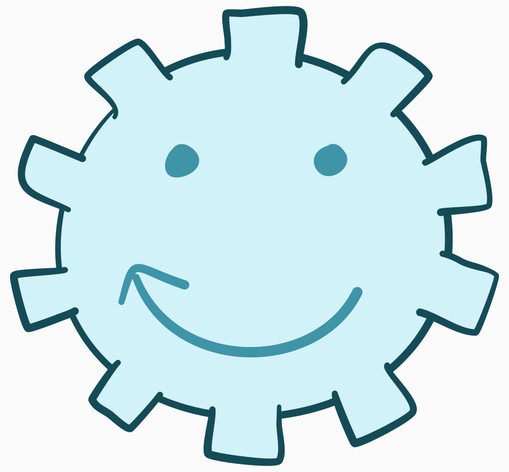


- $X(\infty)$ = **M/M/k in steady-state**
- CDF has jumps at points in $\mathbb{N}$

M/M/k in steady-state



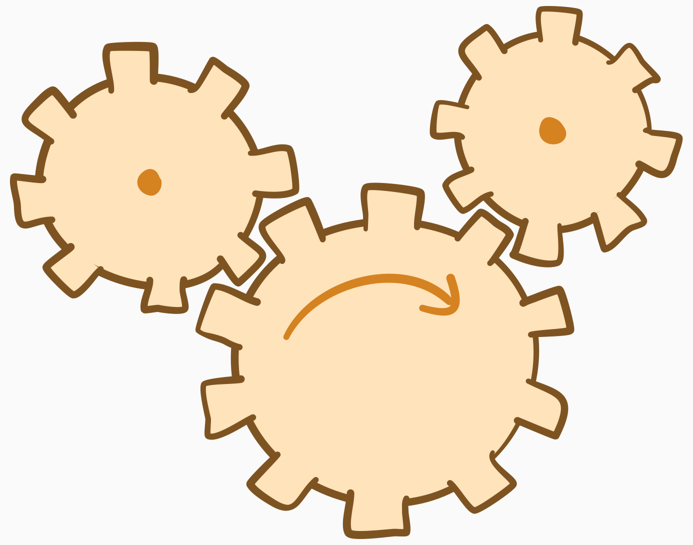


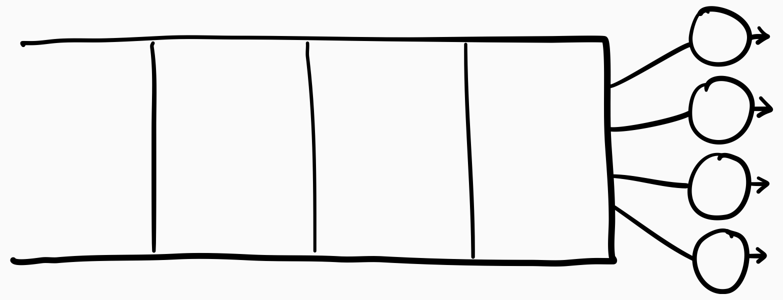


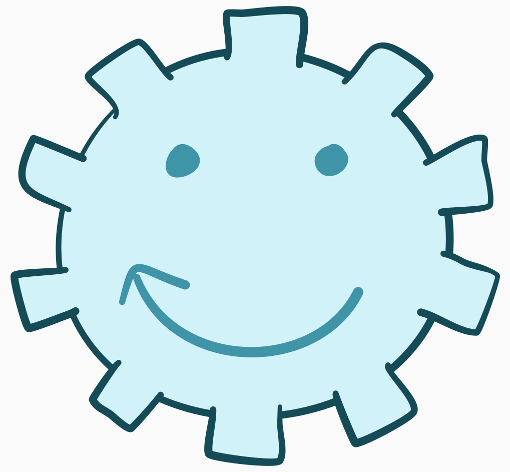
- $X(\infty)$ = **M/M/k in steady-state**
- CDF has jumps at points in $\mathbb{N}$

- Y= absolutely continuous distribution
- CDF is smooth everywhere in $\mathbb{R}$

M/M/k in steady-state

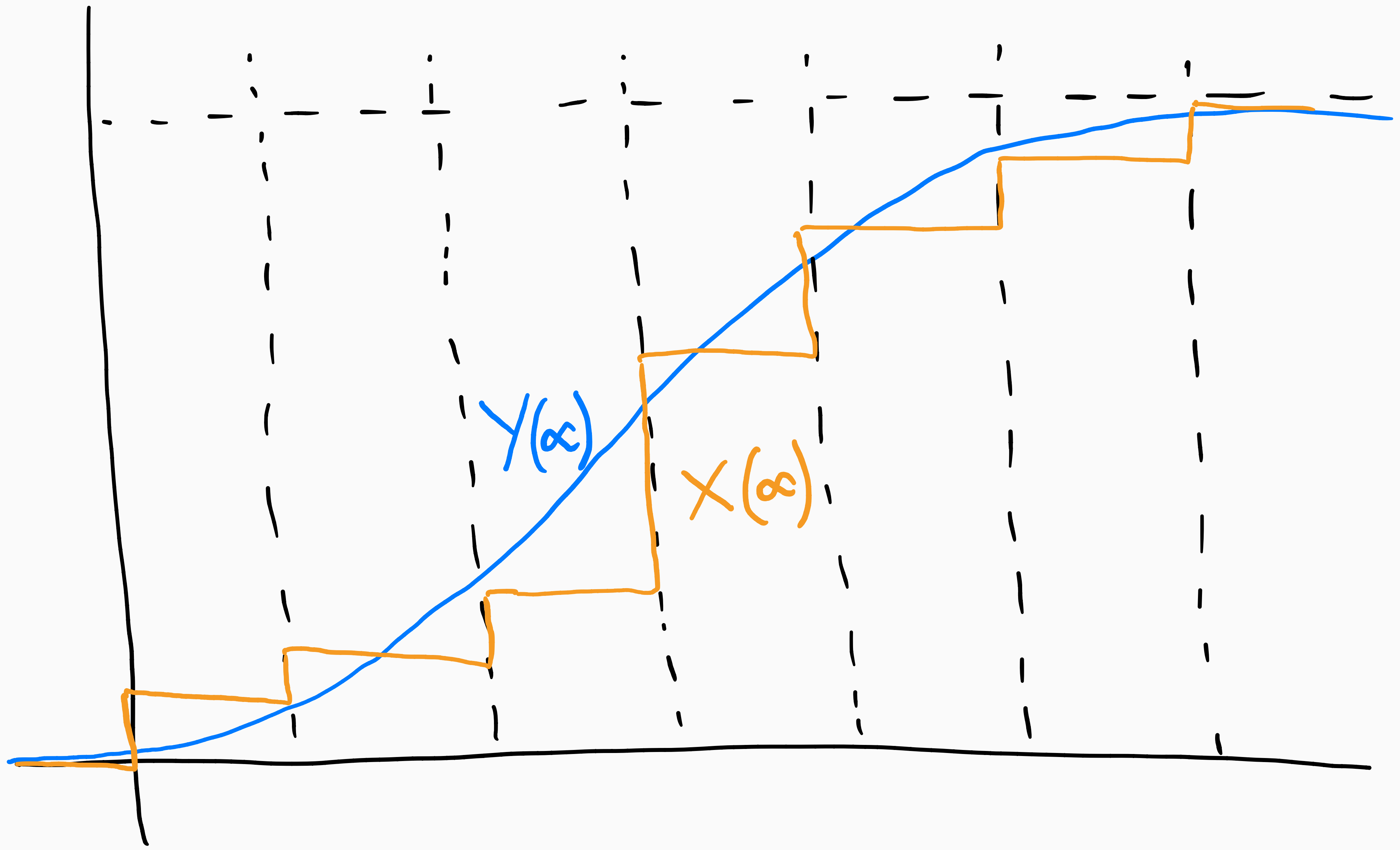




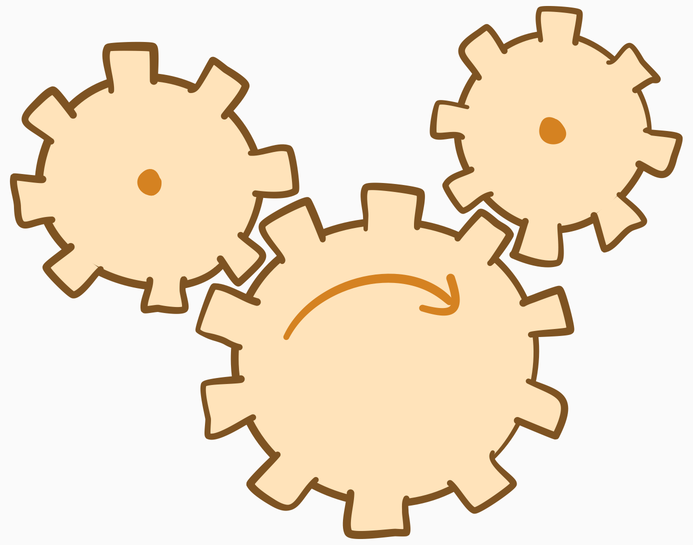


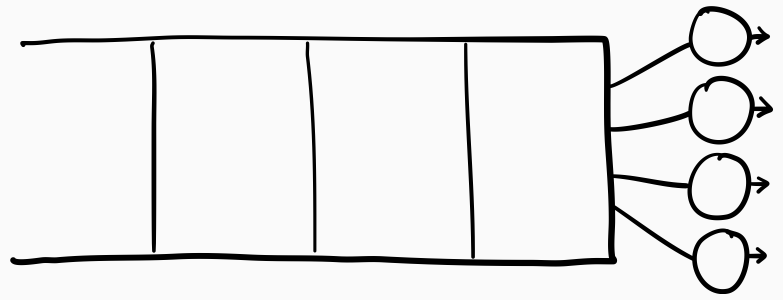
- $X(\infty)$ = **M/M/k in steady-state**
- CDF has jumps at points in $\mathbb{N}$

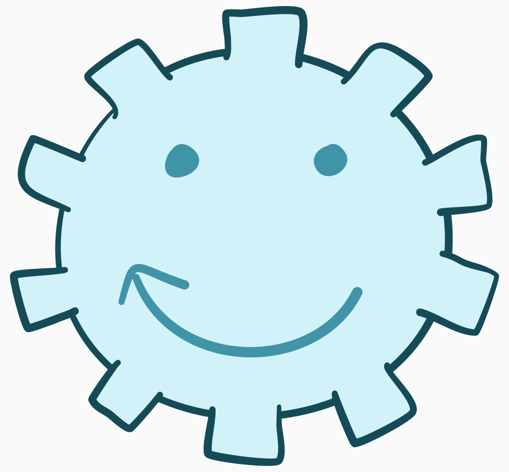
- Y= absolutely continuous distribution
- CDF is smooth everywhere in $\mathbb{R}$



M/M/k in steady-state







- $X(\infty)$ = **M/M/k in steady-state**
- CDF has jumps at points in $\mathbb{N}$

- Y= absolutely continuous distribution
- CDF is smooth everywhere in $\mathbb{R}$

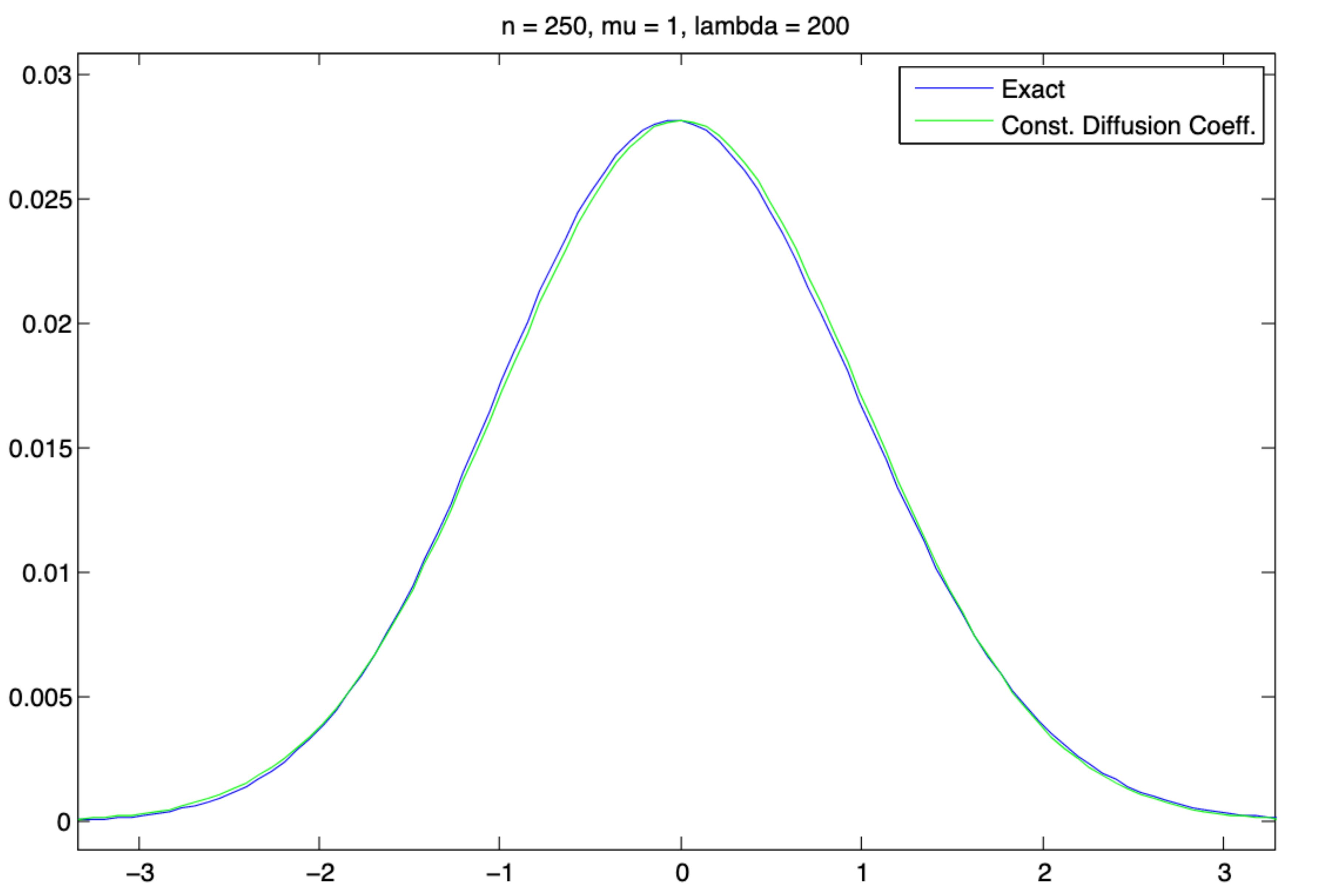


Figure: $P(\tilde{X}(\infty) = x)$, $\mathbb{P}(x - 0.5 \leq Y(\infty) \leq x + 0.5)$

Braverman, Dai & Feng: Stein's method for steady-state diffusion approximations (2015)

Stein’s method

Stein's method

Ingredient 1: Distances between distributions have **variational definitions**

$$d_{\mathcal{H}}(X,Y) = \sup_{h\in\mathcal{H}} |\mathbb{E}[h(X)] - \mathbb{E}[h(Y)]|$$

Stein's method

Ingredient 1: Distances between distributions have **variational definitions**

$$d_{\mathcal{H}}(X,Y) = \sup_{h \in \mathcal{H}} |\mathbb{E}[h(X)] - \mathbb{E}[h(Y)]|$$

- $\mathcal{H} = \{\text{bounded functions}\} \iff d_{\mathcal{H}} =$ total variation distance
- $\mathcal{H} = \{\text{Lipschitz functions}\} \iff d_{\mathcal{H}} =$ Wasserstein-1 distance

Stein's method

Ingredient 1: Distances between distributions have **variational definitions**

$$d_{\mathcal{H}}(X,Y) = \sup_{h\in\mathcal{H}} |\mathbb{E}[h(X)] - \mathbb{E}[h(Y)]|$$

- $\mathcal{H} = \{\text{bounded functions}\} \iff d_{\mathcal{H}} = \text{ total variation distance}$
- $\mathcal{H} = \{\text{Lipschitz functions}\} \iff d_{\mathcal{H}} = \text{ Wasserstein-1 distance}$

Ingredient 2: Every distribution X has a **characterizing operator**

$$\mathbb{E}[\mathcal{A}g(X)] = 0 \text{ for any `good' function } g$$

Stein's method

Ingredient 1: Distances between distributions have **variational definitions**

$$d_{\mathcal{H}}(X,Y) = \sup_{h \in \mathcal{H}} |\mathbb{E}[h(X)] - \mathbb{E}[h(Y)]|$$

- $\mathcal{H} = \{\text{bounded functions}\} \iff d_{\mathcal{H}} =$ total variation distance
- $\mathcal{H} = \{\text{Lipschitz functions}\} \iff d_{\mathcal{H}} =$ Wasserstein-1 distance

Ingredient 2: Every distribution X has a **characterizing operator**

$$\mathbb{E}[\mathcal{A}g(X)] = 0 \text{ for any `good' function } g$$

- $X \sim \mathcal{N}(0,1) \iff \mathcal{A}g(x) = g'(x) - xg(x)$
- What if X is the steady-state distribution of a Markov chain?

Stein's method

Ingredient 2: Every distribution X has a **characterizing operator**

$$\mathbb{E}[\mathcal{A}g(X)] = 0 \text{ for any `good' function } g$$

What about for $X(\infty) \equiv$ steady-state distribution of a **birth-death chain**?

Stein's method

Ingredient 2: Every distribution X has a **characterizing operator**

$$\mathbb{E}[\mathcal{A}g(X)] = 0 \text{ for any `good' function } g$$

What about for $X(\infty) \equiv$ steady-state distribution of a **birth-death chain**?

- Let $\lambda(k), \mu(k)$ be the birth/death rates of any state $k \in \mathbb{N}$, and

$$\mathcal{A}g(x) = \lambda(x)g(x+1) + \mu(x)g((x-1)^+) - g(x)$$

then

$$\mathbb{E}[\mathcal{A}g(X(\infty))] = 0$$

Stein's method

Ingredient 2: Every distribution X has a **characterizing operator**

$$\mathbb{E}[\mathcal{A}g(X)] = 0 \text{ for any 'good' function } g$$

What about for $X(\infty) \equiv$ steady-state distribution of a **birth-death chain**?

- Let $\lambda(k), \mu(k)$ be the birth/death rates of any state $k \in \mathbb{N}$, and

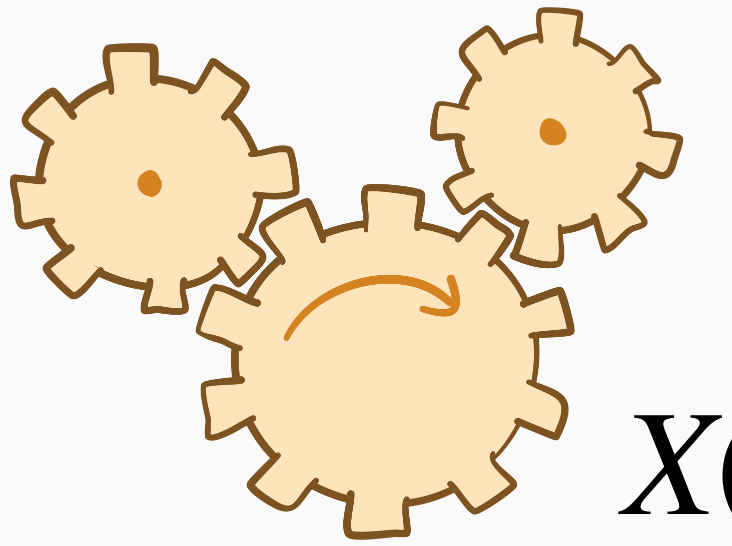
$$\mathcal{A}g(x) = \lambda(x)g(x+1) + \mu(x)g((x-1)^+) - g(x)$$

then

$$\mathbb{E}[\mathcal{A}g(X(\infty))] = 0$$

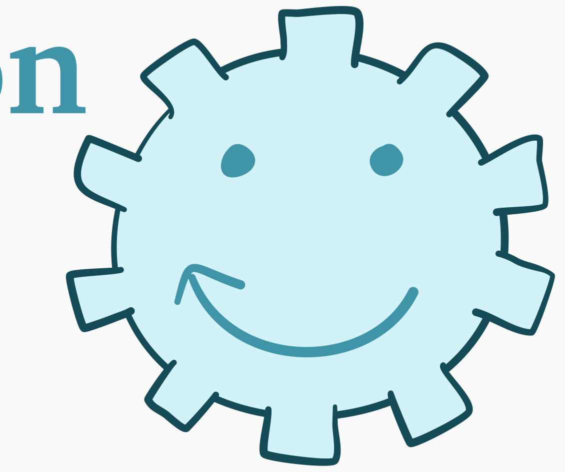
$$X \sim \text{steady-state of MC on } \mathbb{N} \text{ with generator } G \iff \mathcal{A}g(k) = G(g(0), g(1), \ldots)^T$$

Stein's method



$X(\infty)$: **M/M/k in steady-state**

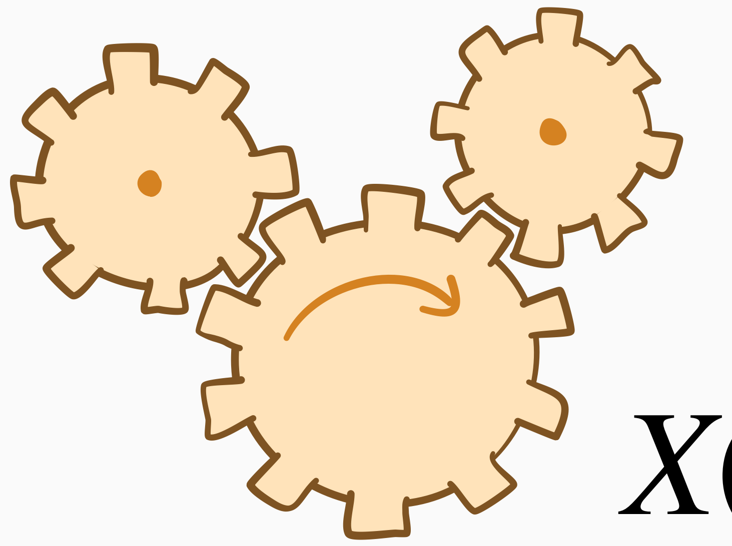
Y : **candidate distribution**



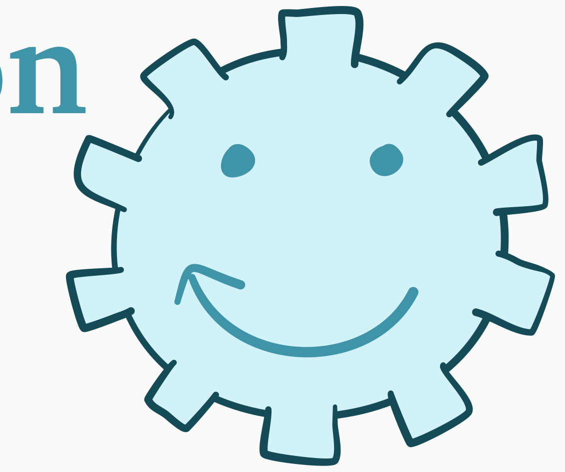
Recipe for bounding distances between $X(\infty)$, Y :

- For any $h \in \mathcal{H}$, find f such that $\mathcal{A}_Y f(z) = h(z) - \mathbb{E}[h(Y)]$

Stein's method



Y : **candidate distribution**



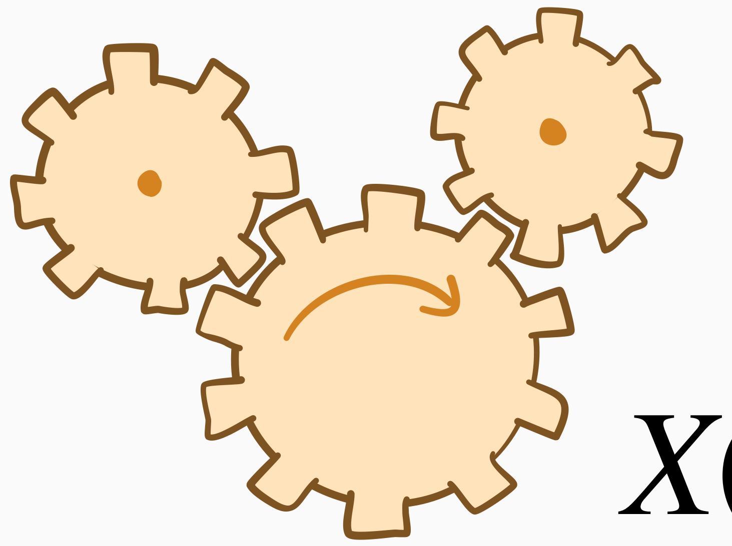
$X(\infty)$: **M/M/k in steady-state**

Poisson equation

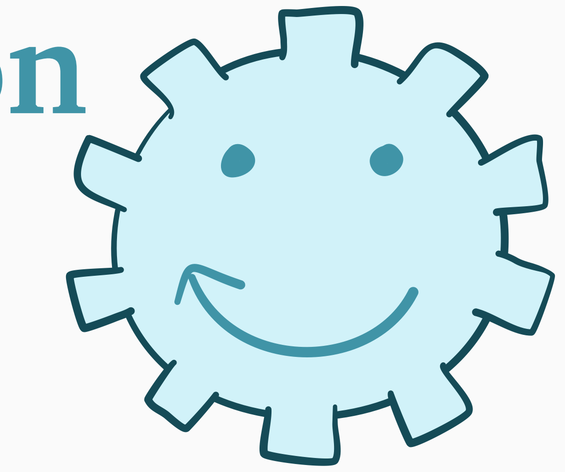
Recipe for bounding distances between $X(\infty)$, Y :

- For any $h \in \mathcal{H}$, find f such that $\mathcal{A}_Y f(z) = h(z) - \mathbb{E}[h(Y)]$

Stein's method



Y : **candidate distribution**



$X(\infty)$: **M/M/k in steady-state**

Poisson equation

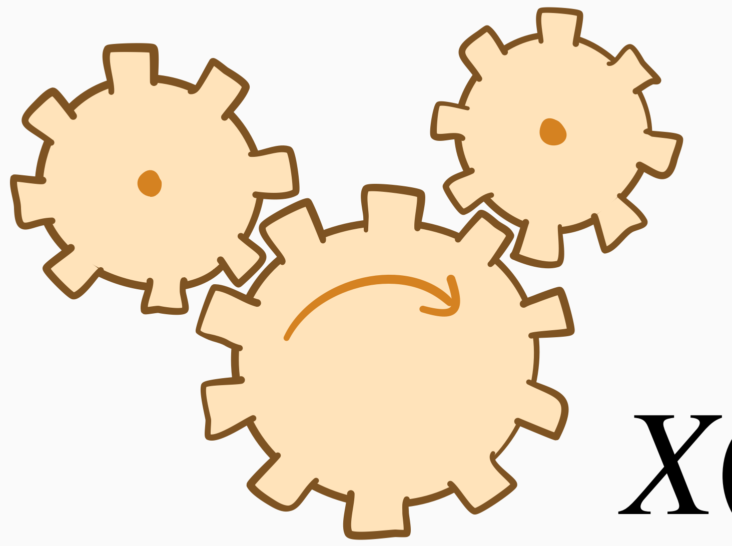
Recipe for bounding distances between $X(\infty)$, Y :

- For any $h \in \mathcal{H}$, find f such that $\mathcal{A}_Y f(z) = h(z) - \mathbb{E}[h(Y)]$
- Then taking expectations, we get

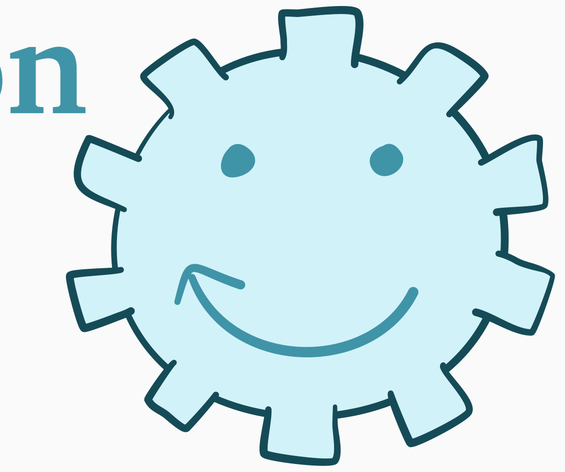
=0 !

$$\mathbb{E}[h(X(\infty)) - h(Y)] = \mathbb{E}[\mathcal{A}_Y f(X(\infty)) - \mathcal{A}_X f(X(\infty))]$$

Stein's method



Y : **candidate distribution**



$X(\infty)$: **M/M/k in steady-state**

Recipe for bounding distances between $X(\infty)$, Y :

Poisson equation

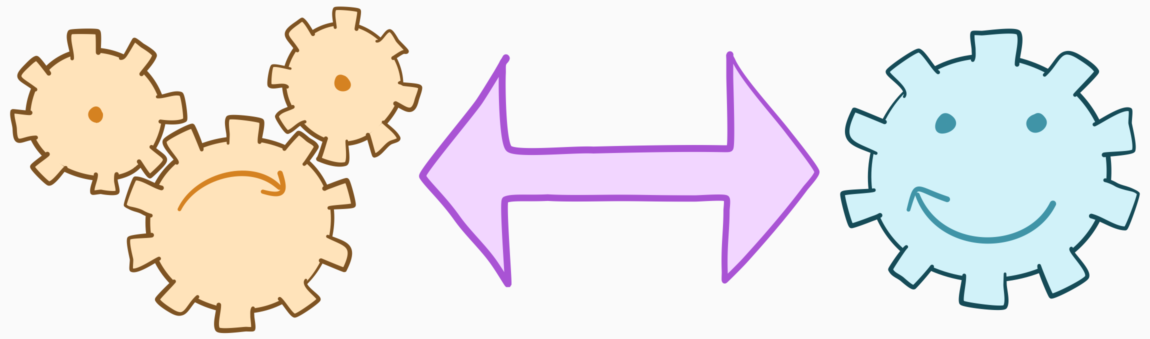
- For any $h \in \mathcal{H}$, find f such that $\mathcal{A}_Y f(z) = h(z) - \mathbb{E}[h(Y)]$
- Then taking expectations, we get

=0 !

$$\mathbb{E}[h(X(\infty)) - h(Y)] = \mathbb{E}[\mathcal{A}_Y f(X(\infty)) - \mathcal{A}_X f(X(\infty))]$$

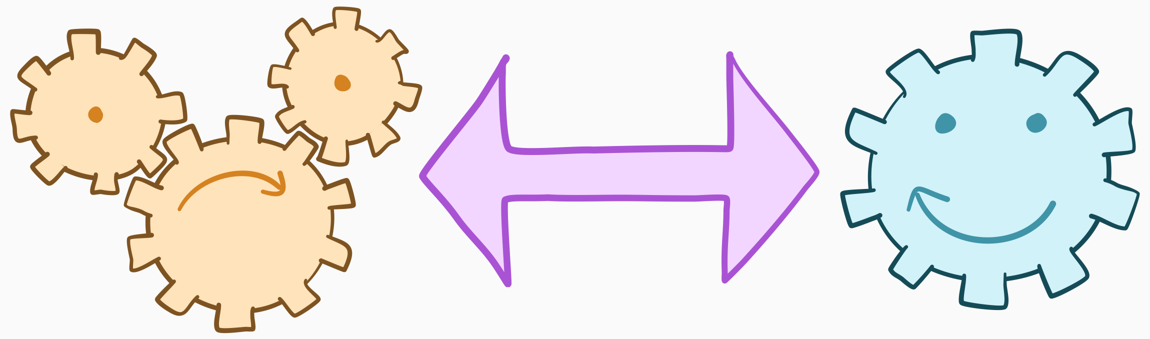
Generator coupling: if $\mathcal{A}_X$ and $\mathcal{A}_Y$ are 'close', then the RHS is small

Classifying coupling techniques



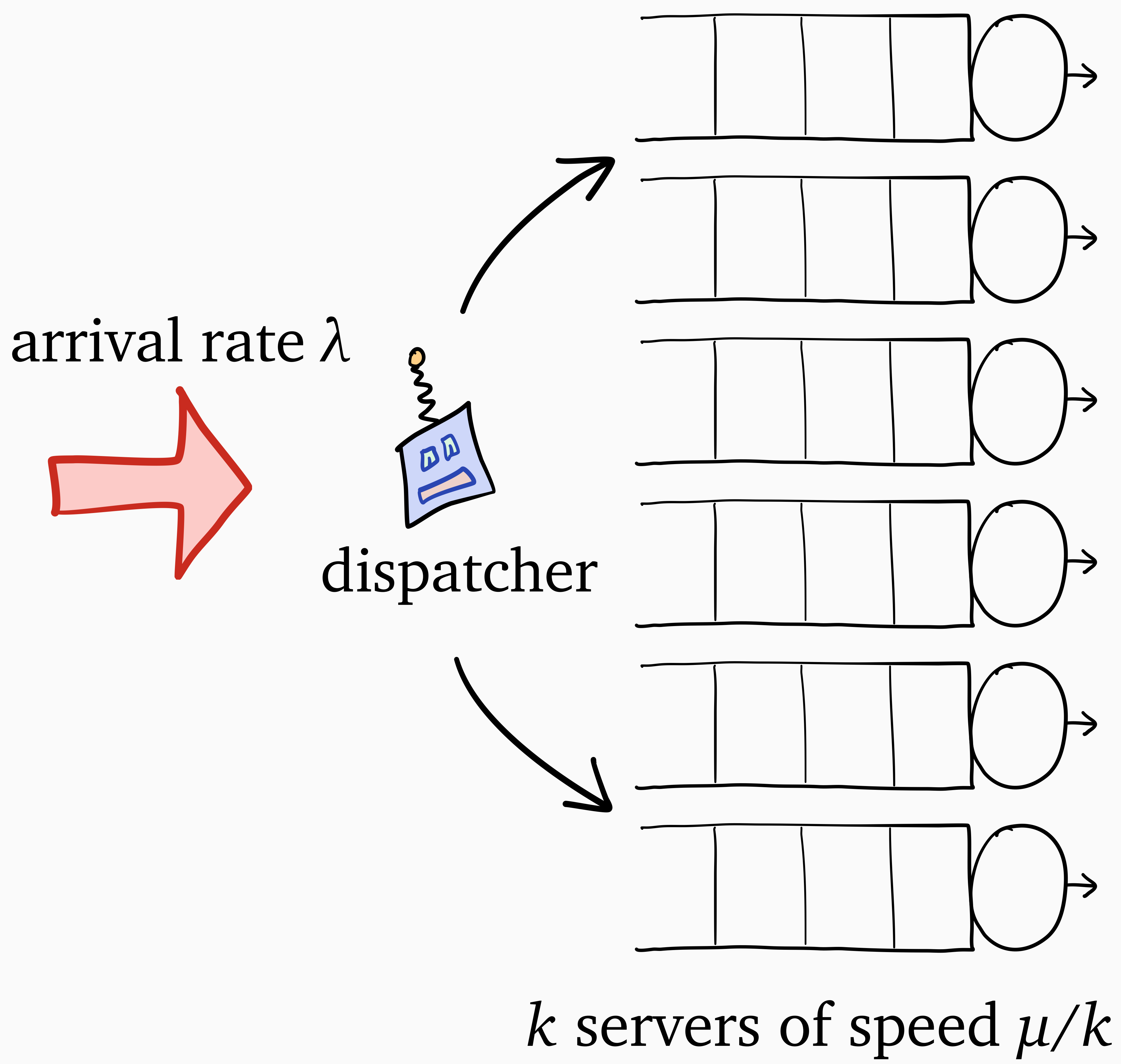
	A. Every sample path	B. Steady-state distribution
1. More information	A1 BIG online knapsack (via compensated coupling)	B1
2. Fewer constraints	A2 M/M/k vs. M/M/1 Online knapsack (via constraints-to-costs)	B2
3. Simpler dynamics	A3 SIS epidemics Queues with redundancy	B3

Classifying coupling techniques

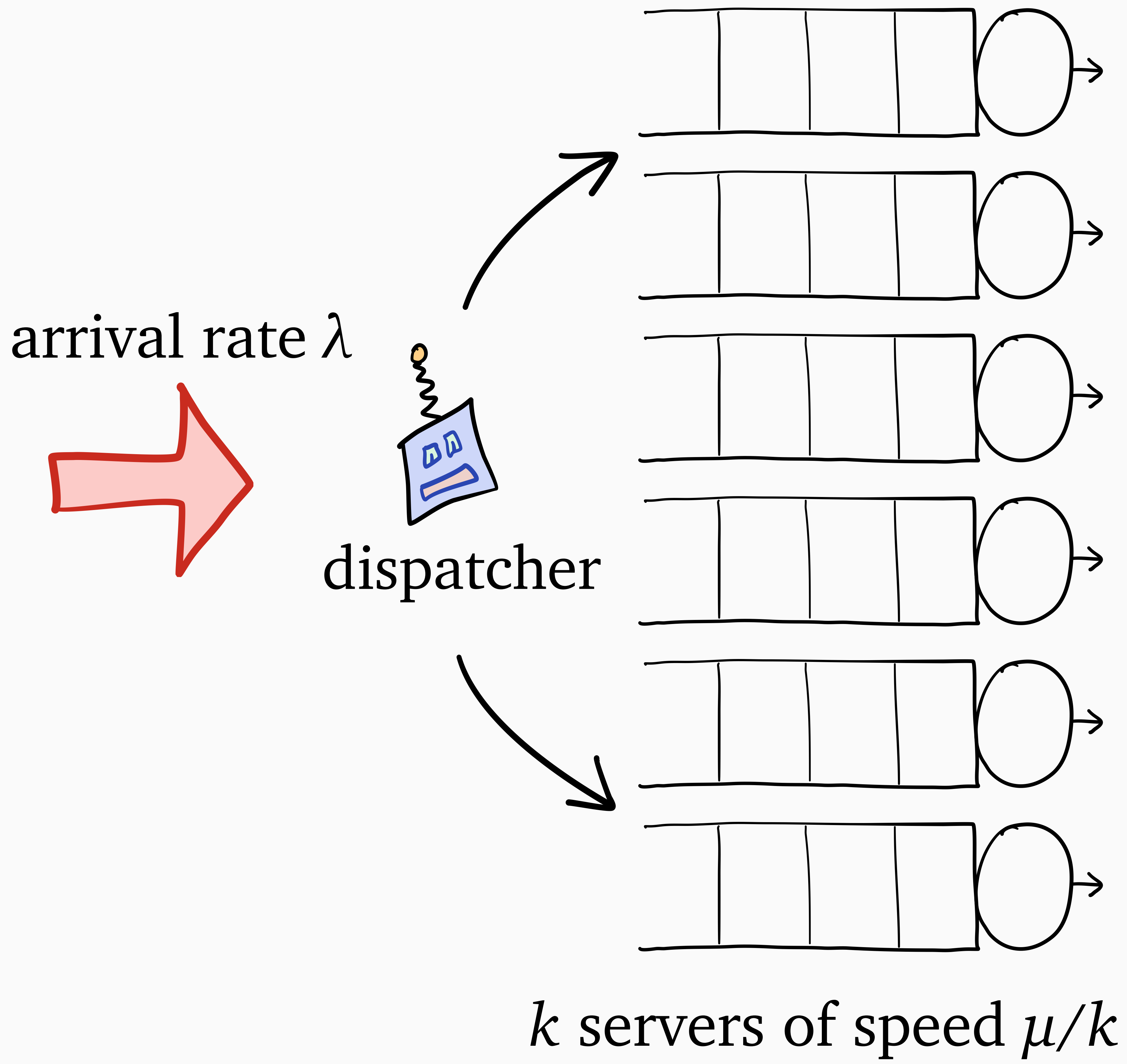


	A. Every sample path	B. Steady-state distribution
1. More information	A1 BIG online knapsack (via compensated coupling)	B1
2. Fewer constraints	A2 M/M/k vs. M/M/1 Online knapsack (via constraints-to-costs)	B2
3. Simpler dynamics	A3 SIS epidemics Queues with redundancy	B3 Stein's method

Load-balancing systems

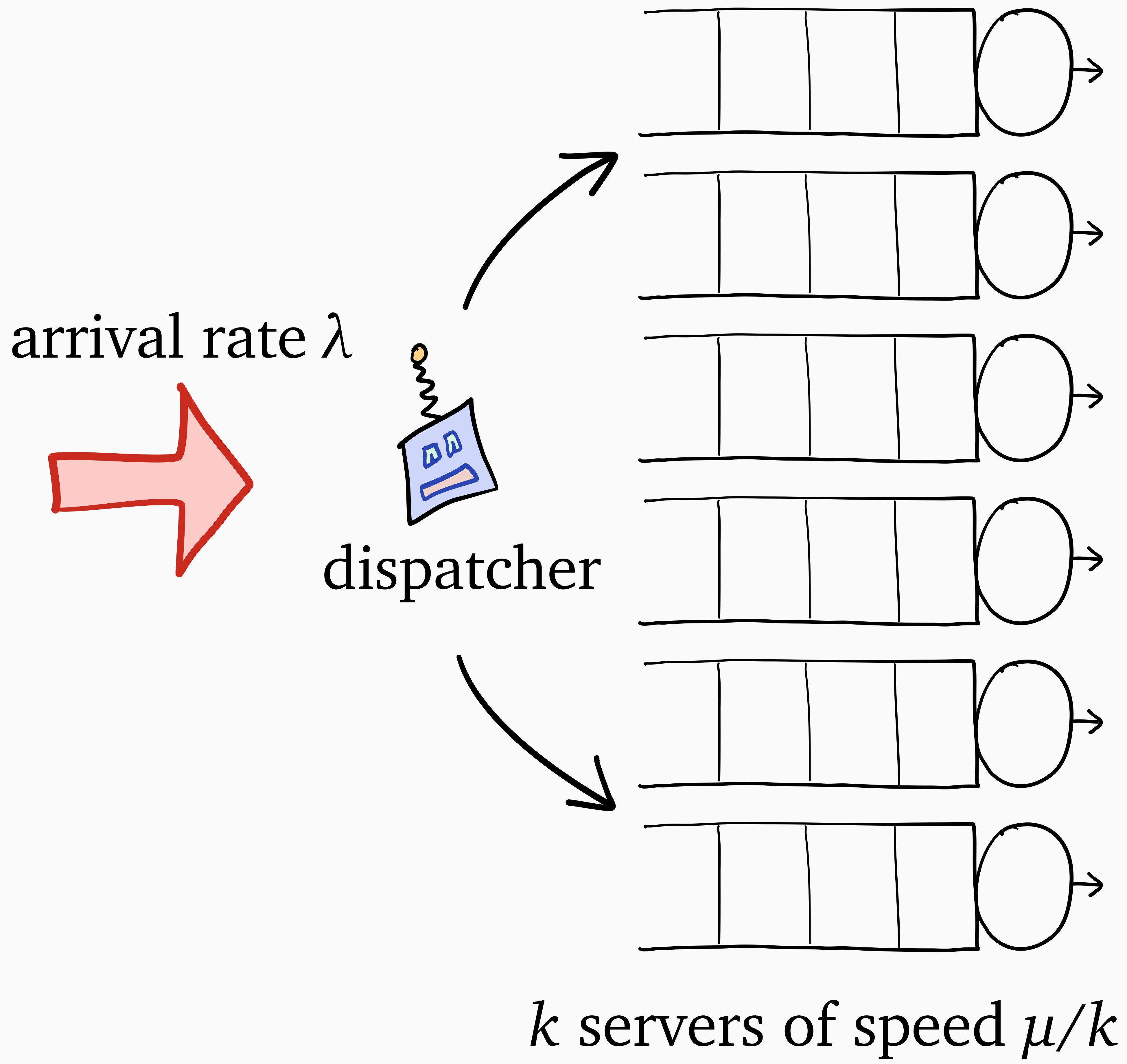


Load-balancing systems



Load: $\rho = \lambda/\mu$

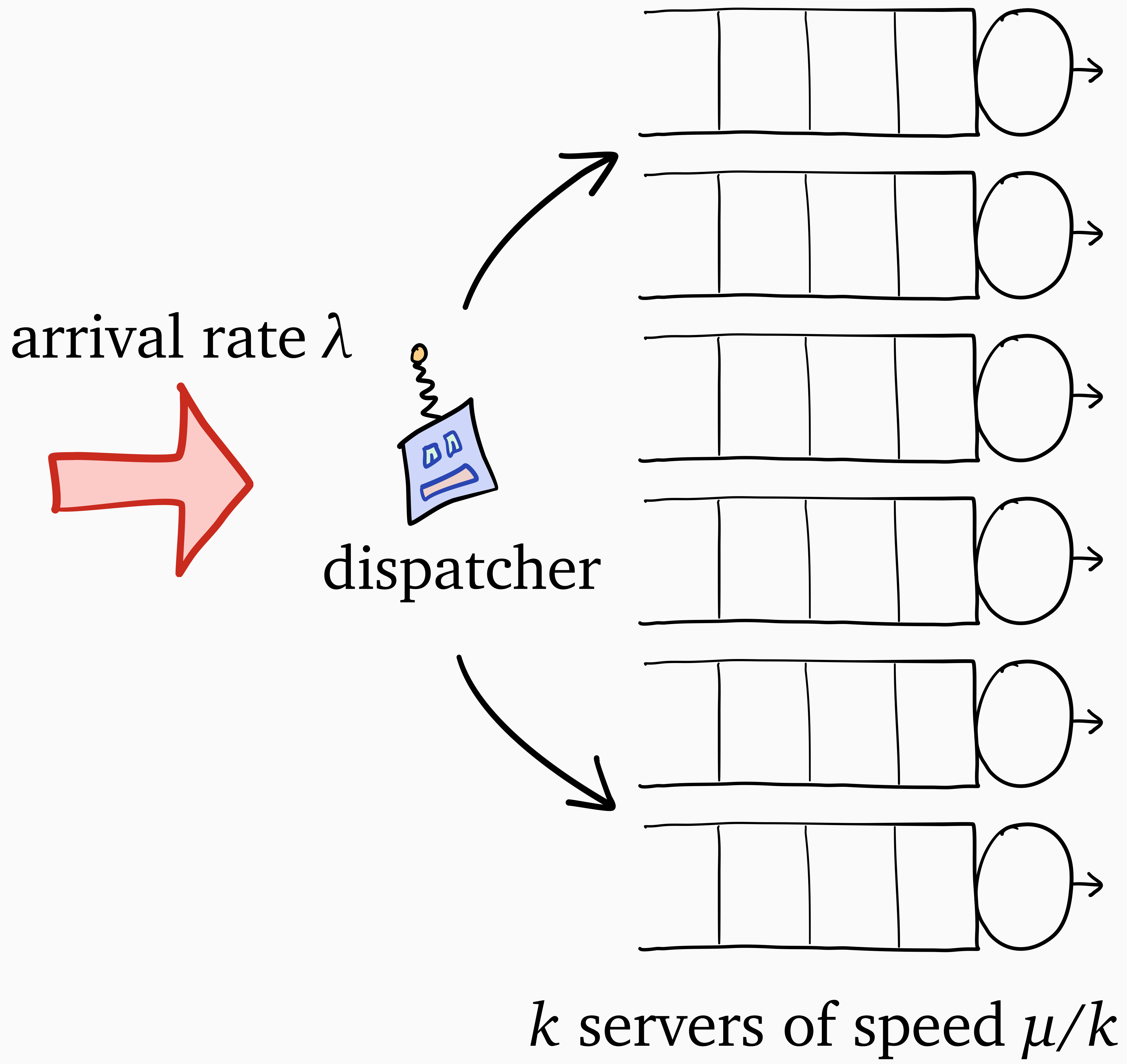
Load-balancing systems



Load: $\rho = \lambda/\mu$

Join the Shortest Queue (JSQ):
always dispatch job to server with fewest jobs

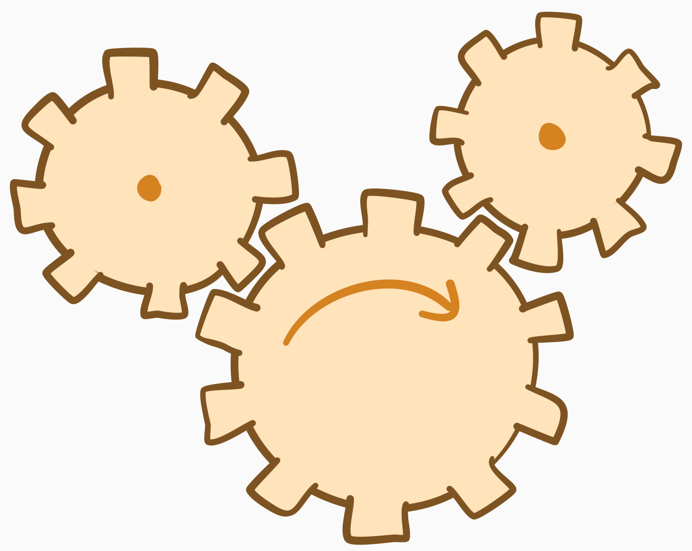
Load-balancing systems



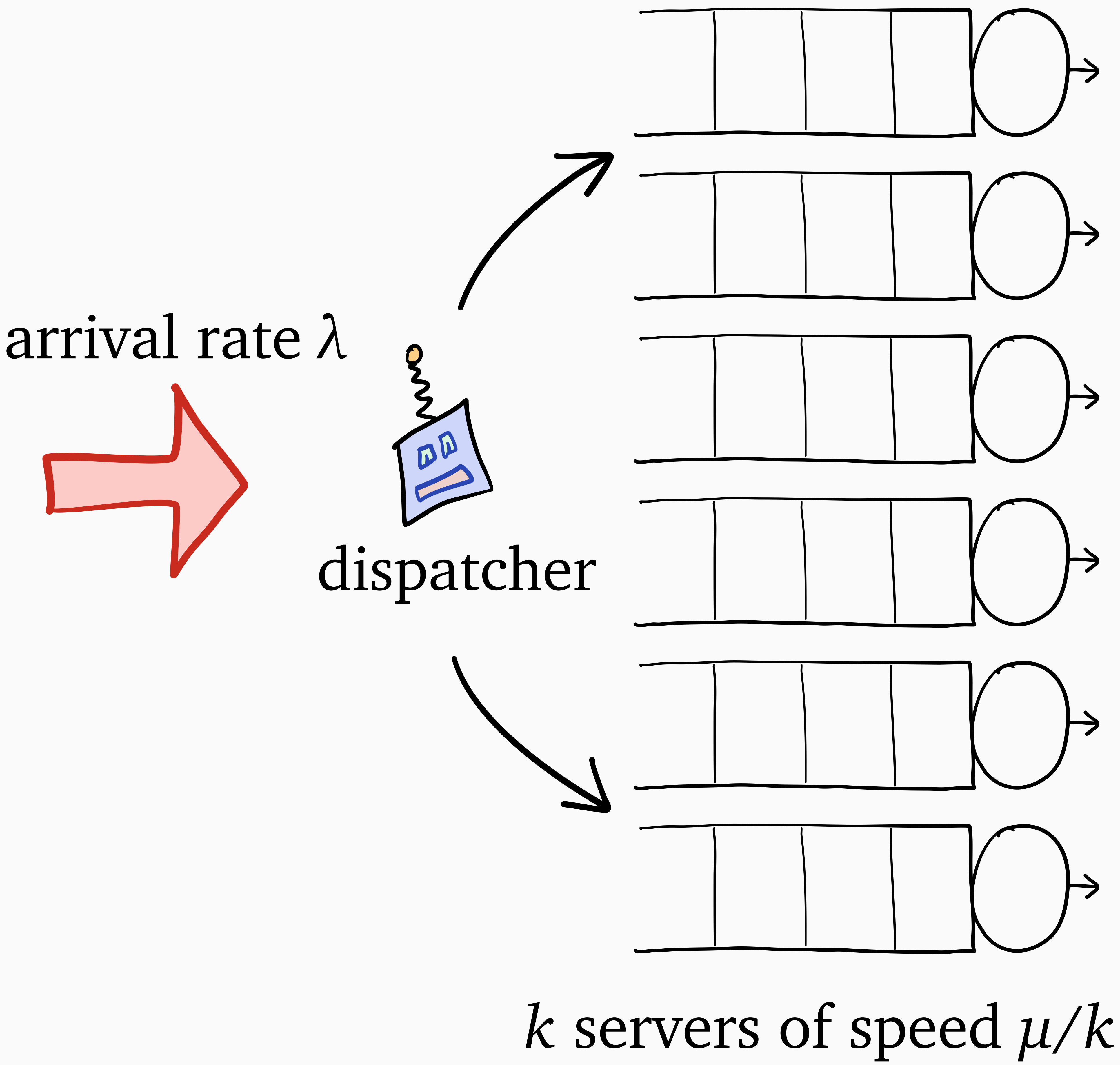
Load: $\rho = \lambda/\mu$

Join the Shortest Queue (JSQ):
always dispatch job to server with fewest jobs

With just 2 servers, already a **famously hard** Markov chain



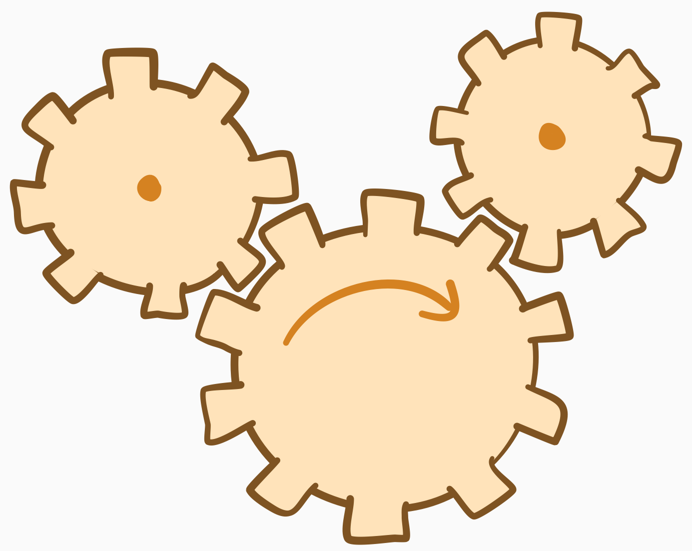
Load-balancing systems



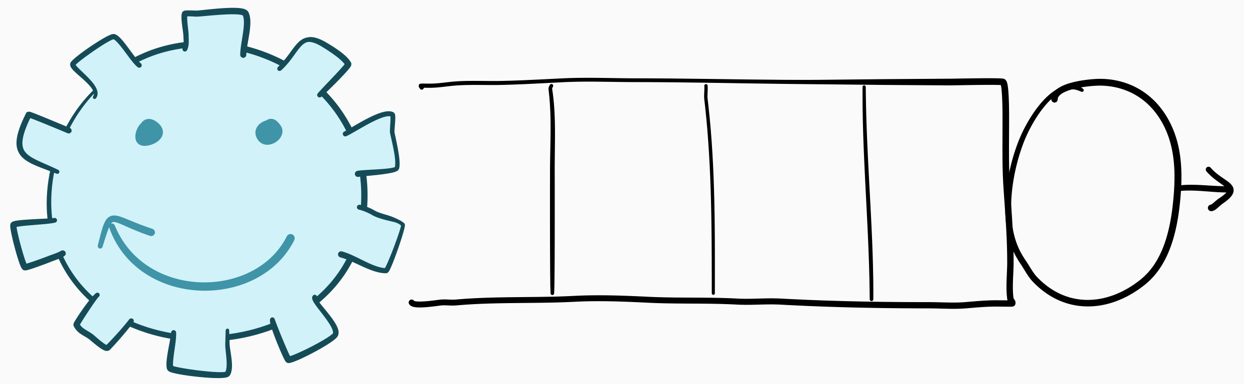
Load: $\rho = \lambda/\mu$

Join the Shortest Queue (JSQ):
always dispatch job to server with fewest jobs

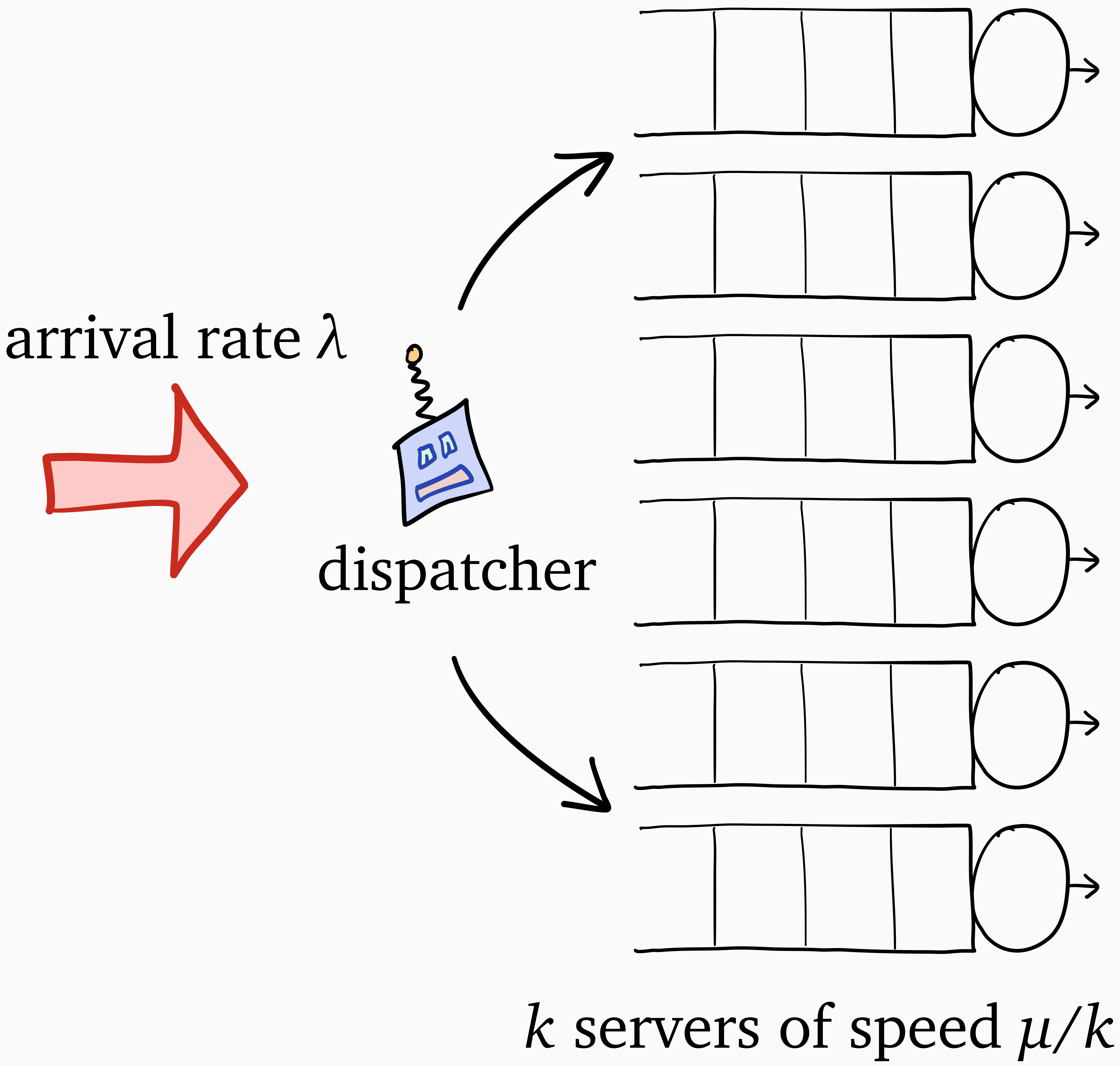
With just 2 servers, already a **famously hard** Markov chain



M/M/1 is simple lower bound



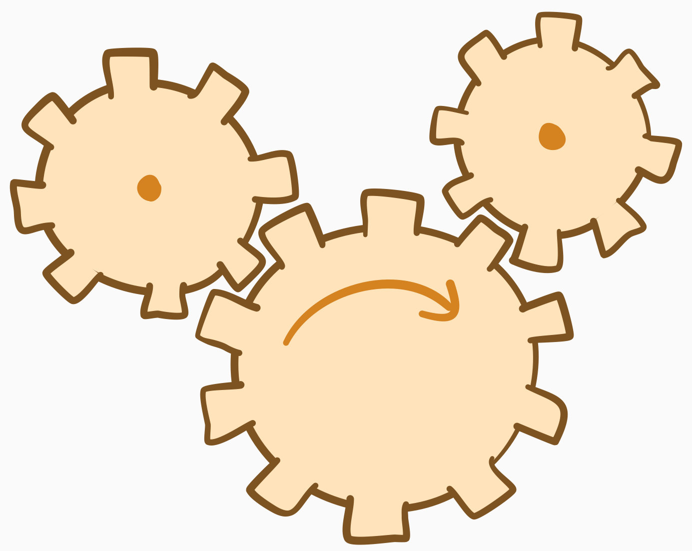
Load-balancing systems



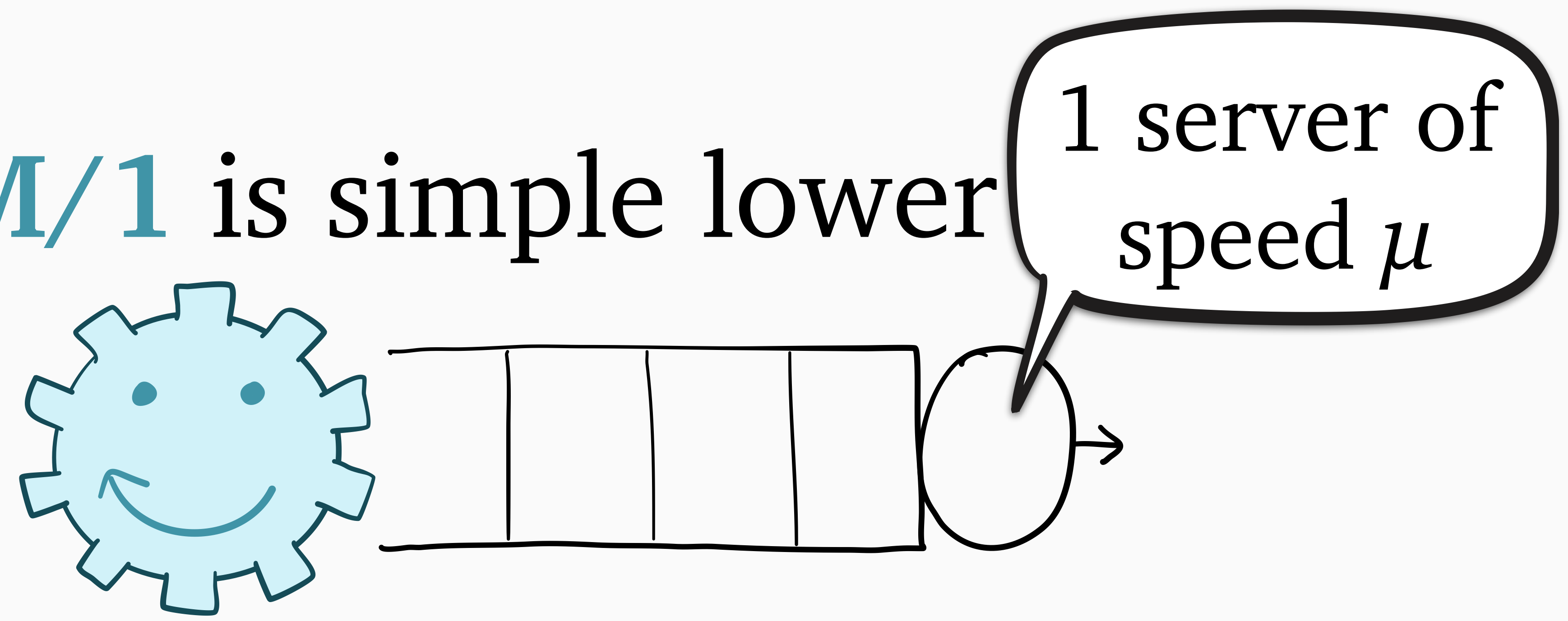
Load: $\rho = \lambda/\mu$

Join the Shortest Queue (JSQ):
always dispatch job to server with fewest jobs

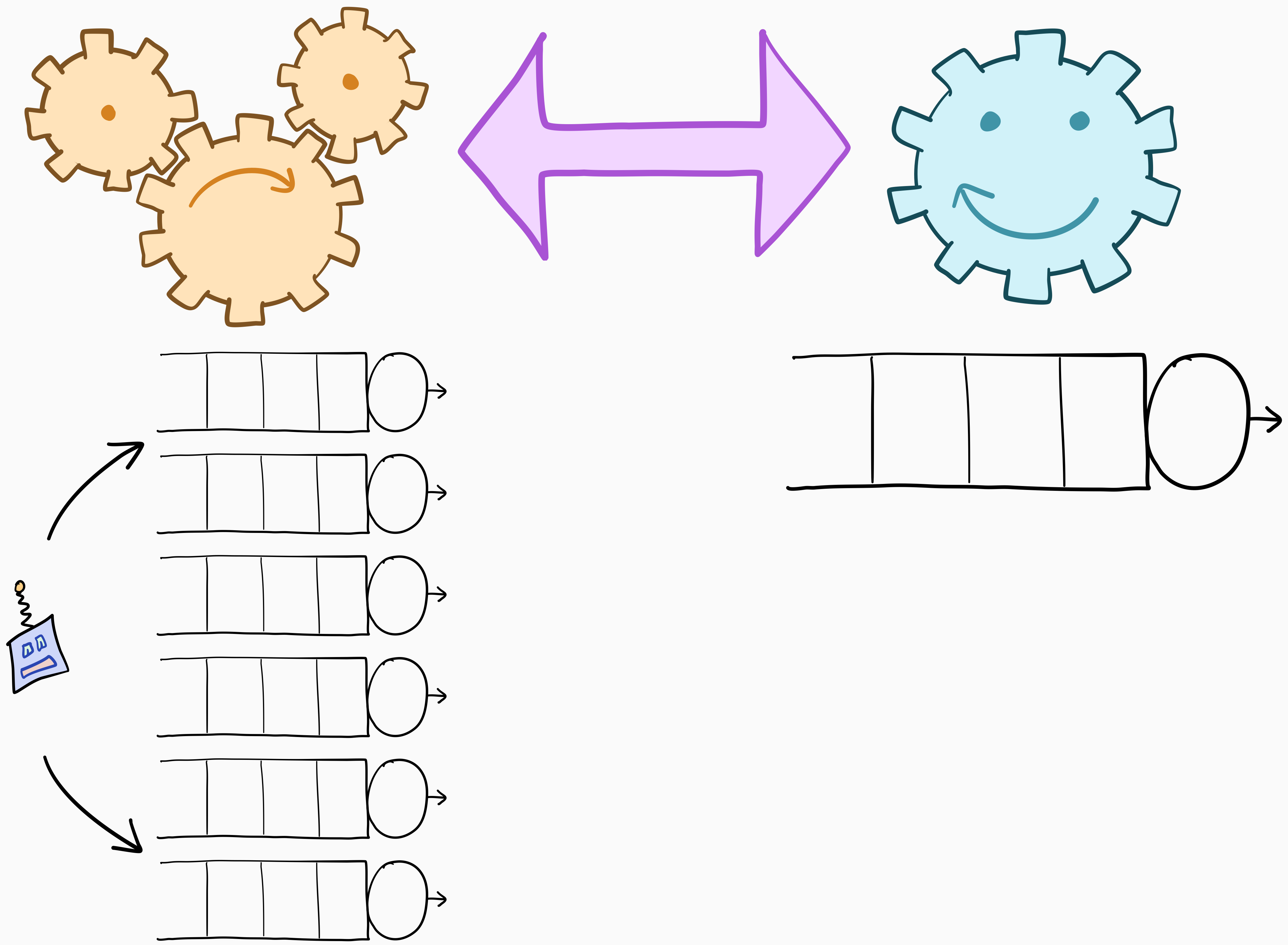
With just 2 servers, already a **famously hard** Markov chain



M/M/1 is simple lower

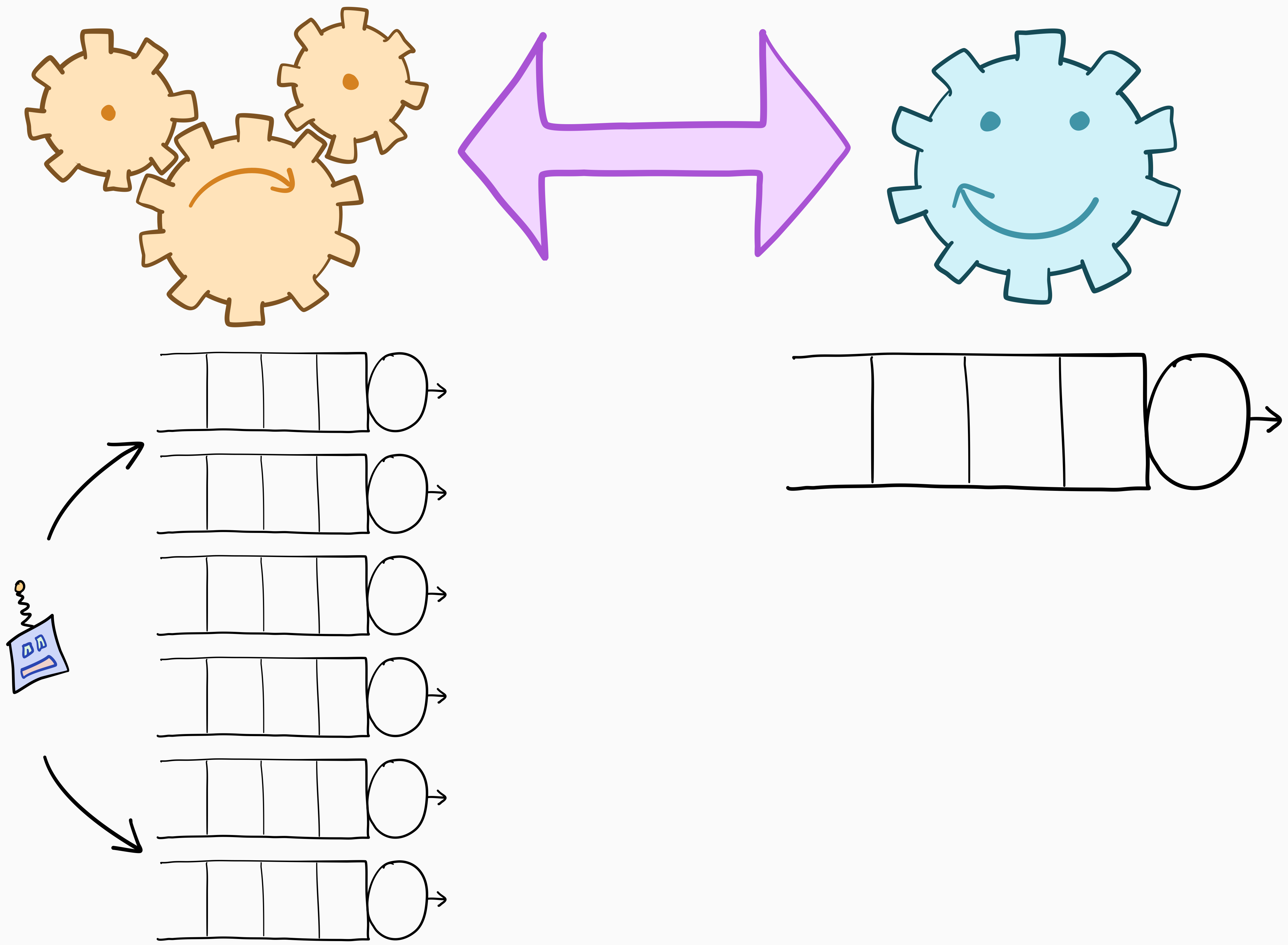


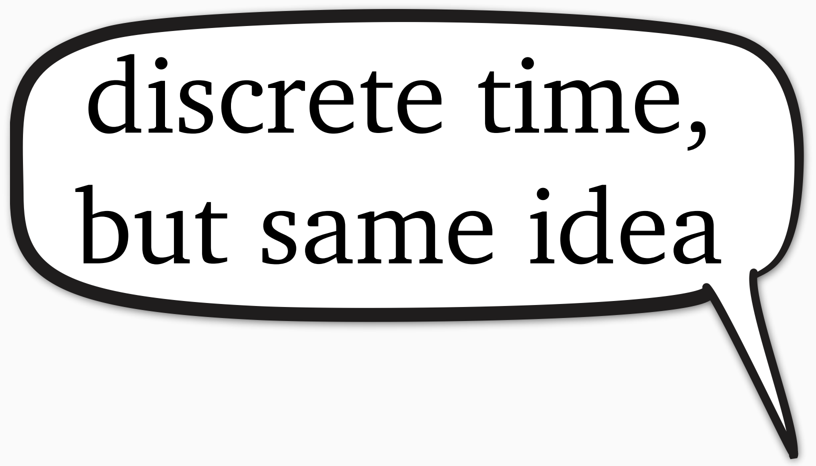
JSQ vs. M/M/1



Eryilmaz and Srikant: Asymptotically tight steady-state queue length bounds implied by drift conditions (2012)

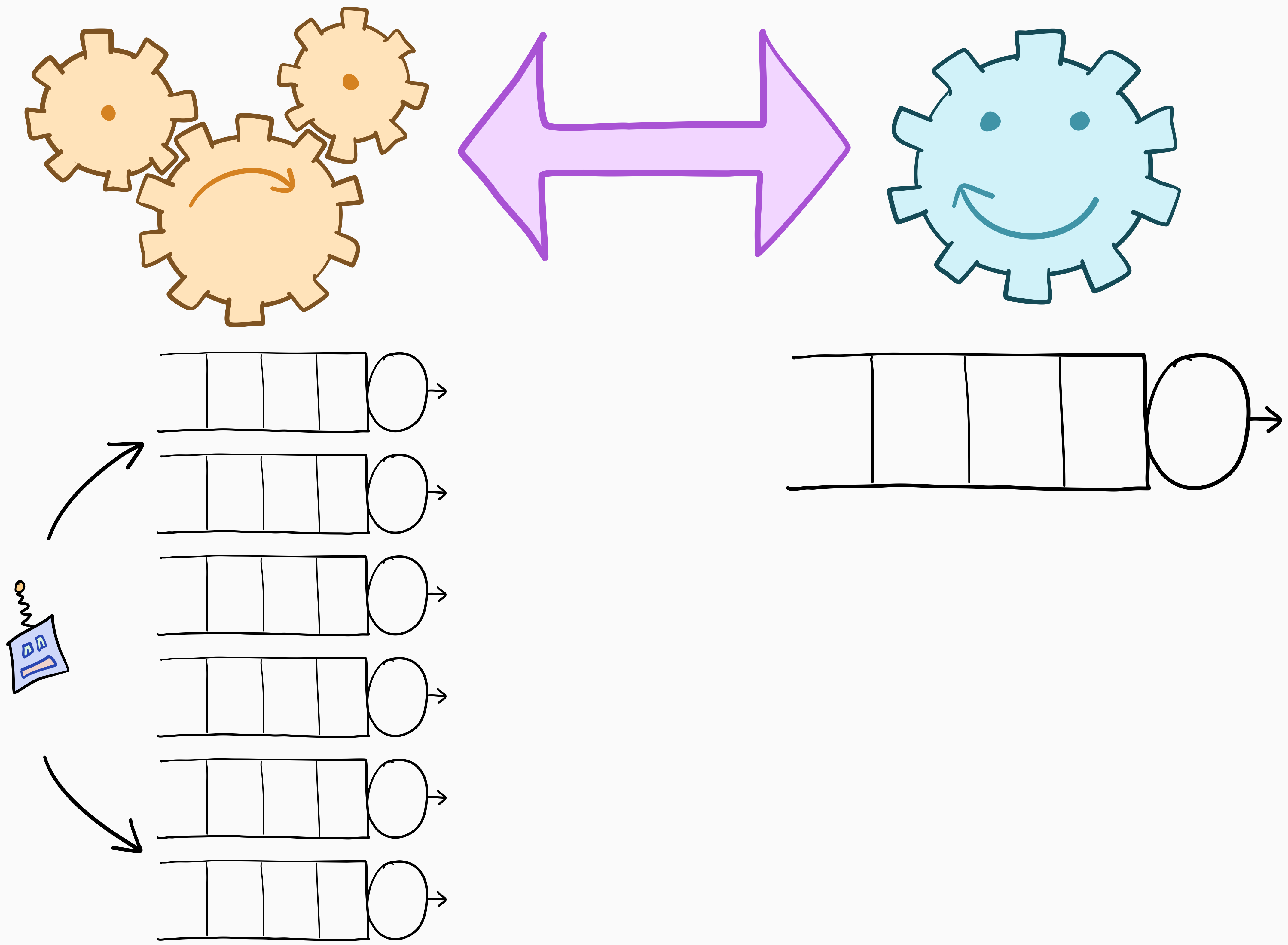
JSQ vs. M/M/1





Eryilmaz and Srikant: Asymptotically tight steady-state queue length bounds implied by drift conditions (2012)

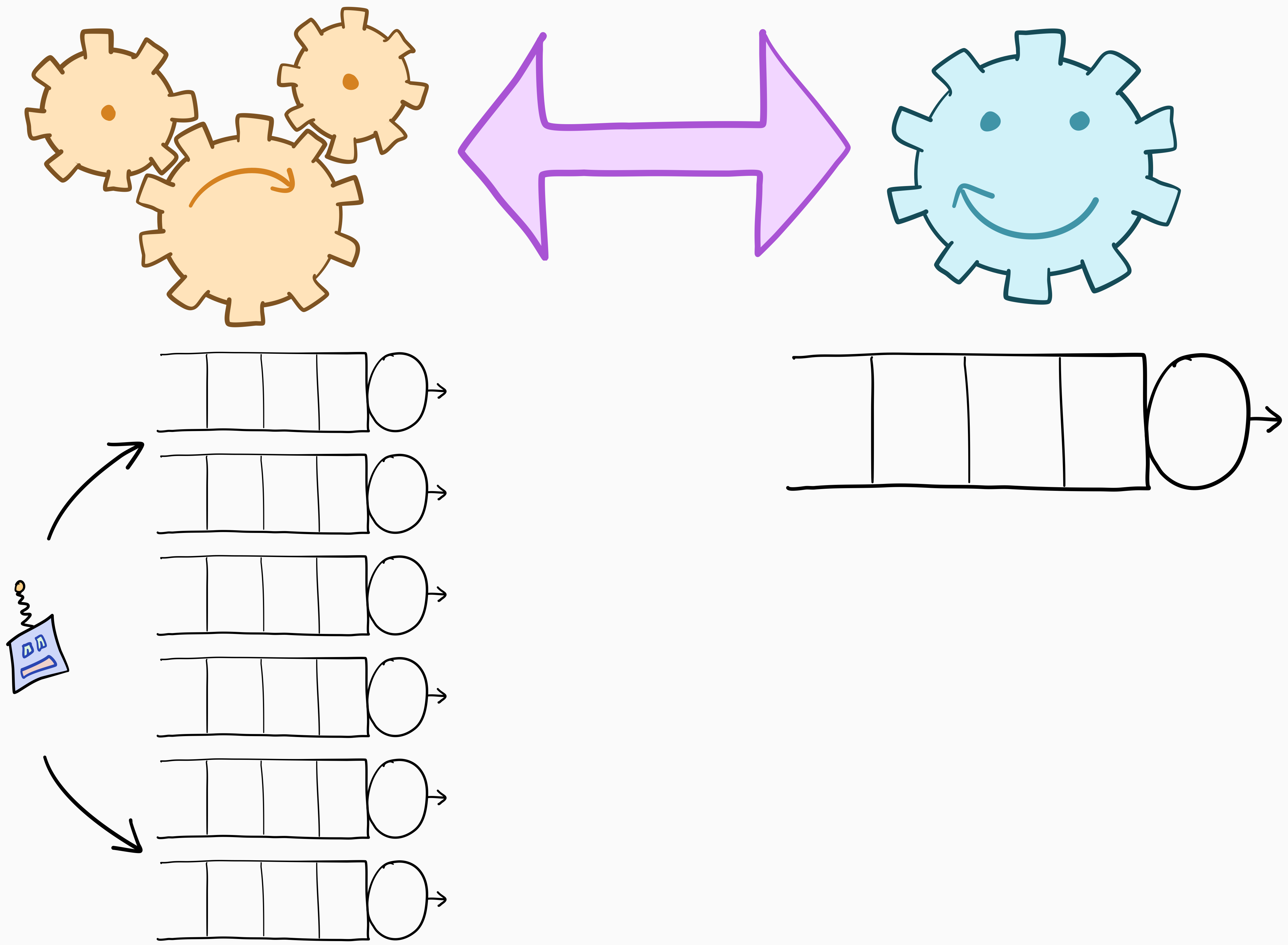
JSQ vs. M/M/1



Clear that $\mathbf{E}[N_k] \geq \mathbf{E}[N_1]$

Eryilmaz and Srikant: Asymptotically tight steady-state queue length bounds implied by drift conditions (2012)

JSQ vs. M/M/1

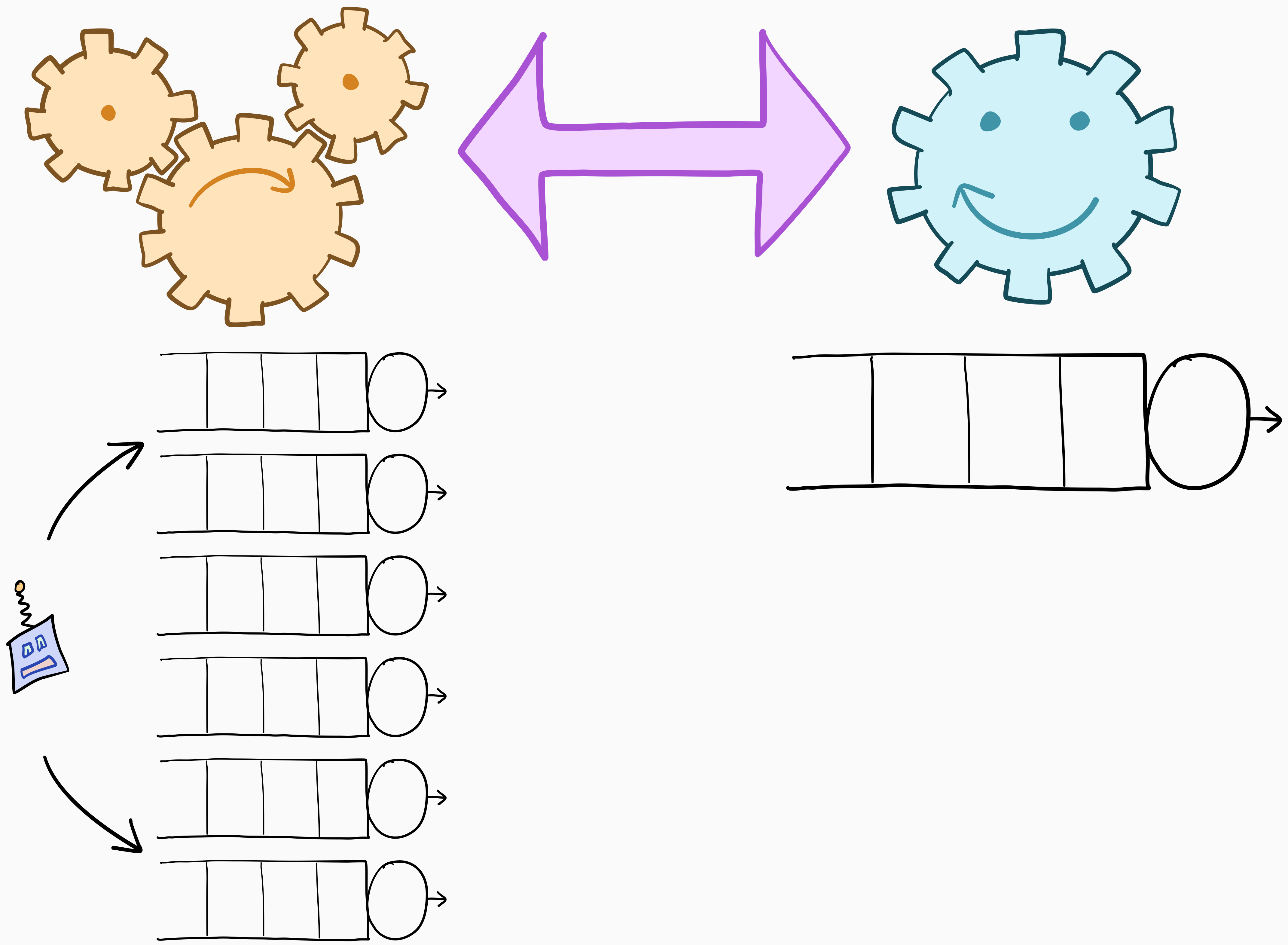


Clear that $\mathbf{E}[N_k] \geq \mathbf{E}[N_1]$

Question: *upper bound* on $\mathbf{E}[N_k]$?

Eryilmaz and Srikant: Asymptotically tight steady-state queue length bounds implied by drift conditions (2012)

JSQ vs. M/M/1



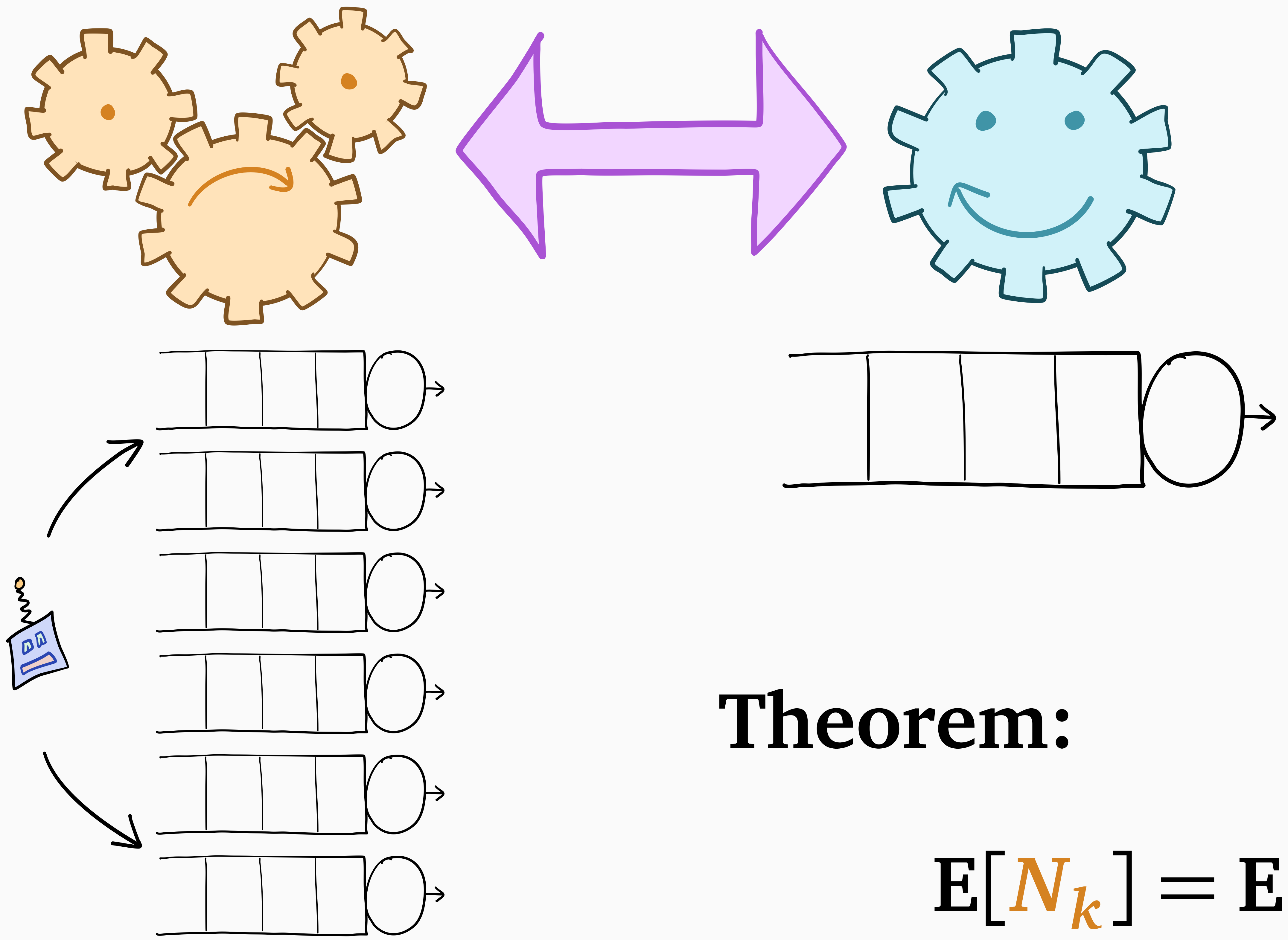
Clear that $\mathbf{E}[N_k] \geq \mathbf{E}[N_1]$

Question: *upper bound* on $\mathbf{E}[N_k]$?

Intuition: should be close to $\mathbf{E}[N_1]$, because JSQ does a good job of keeping servers busy

Eryilmaz and Srikant: Asymptotically tight steady-state queue length bounds implied by drift conditions (2012)

JSQ vs. M/M/1



Clear that $\mathbf{E}[N_k] \geq \mathbf{E}[N_1]$

Question: *upper bound* on $\mathbf{E}[N_k]$?

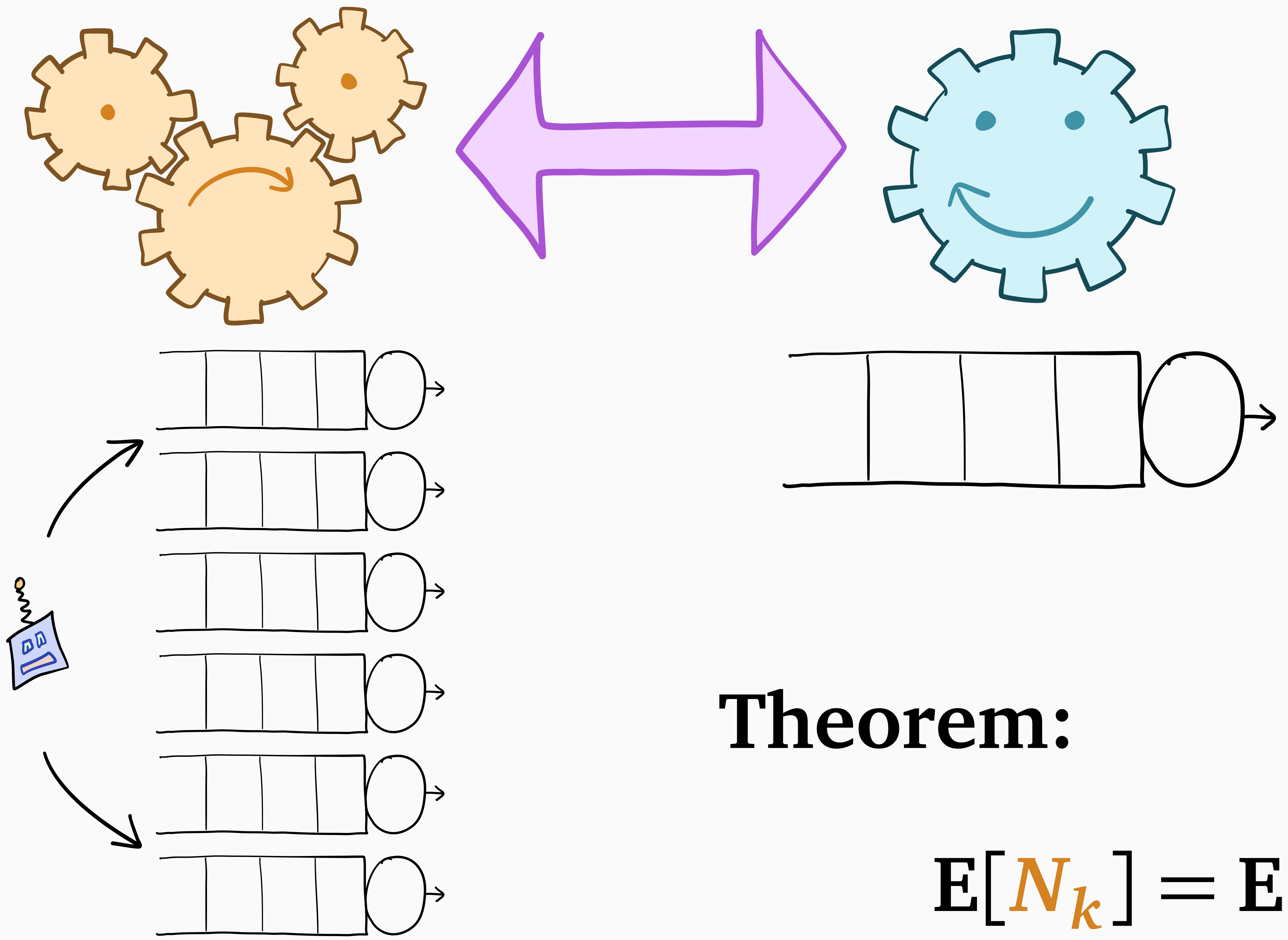
Intuition: should be close to $\mathbf{E}[N_1]$, because JSQ does a good job of keeping servers busy

Theorem:

$$\mathbf{E}[N_k] = \mathbf{E}[N_1] + \frac{\mathbf{E}[(1 - B_k)N_k]}{1 - \rho}$$

Eryilmaz and Srikant: Asymptotically tight steady-state queue length bounds implied by drift conditions (2012)

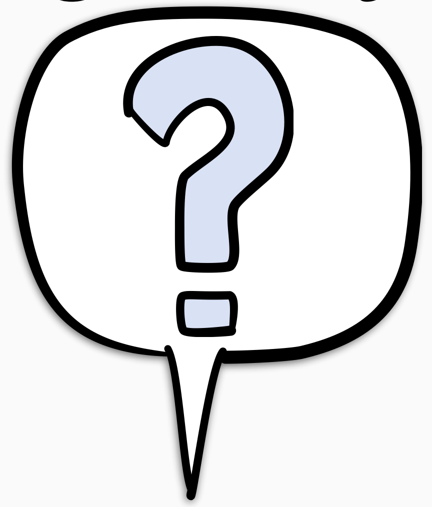
JSQ vs. M/M/1



Clear that $\mathbf{E}[N_k] \geq \mathbf{E}[N_1]$

Question: *upper bound* on $\mathbf{E}[N_k]$?

Intuition: should be close to $\mathbf{E}[N_1]$, because JSQ does a good job of keeping servers busy

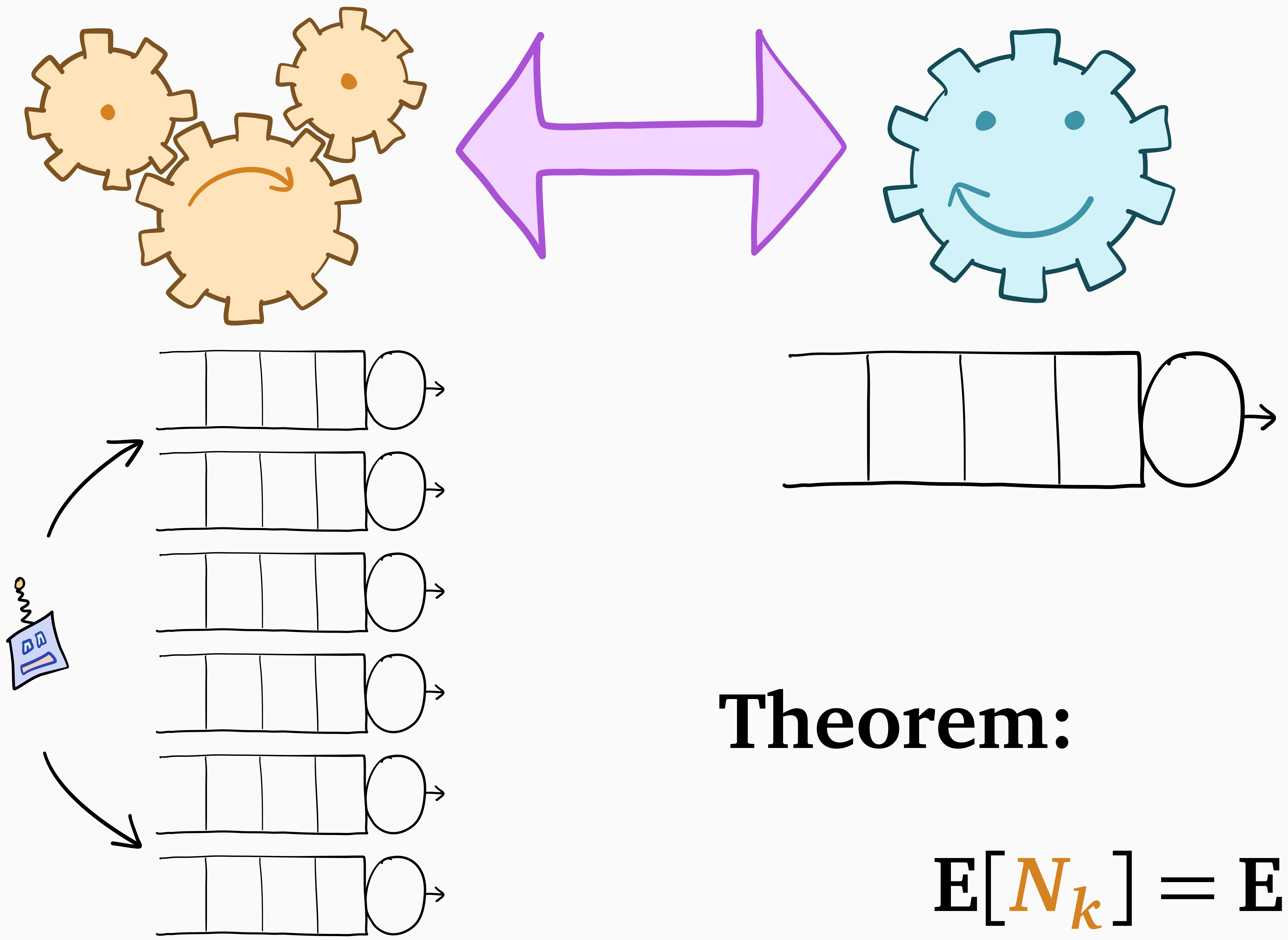


Theorem:

$$\mathbf{E}[N_k] = \mathbf{E}[N_1] + \frac{\mathbf{E}[(1 - B_k)N_k]}{1 - \rho}$$

Eryilmaz and Srikant: Asymptotically tight steady-state queue length bounds implied by drift conditions (2012)

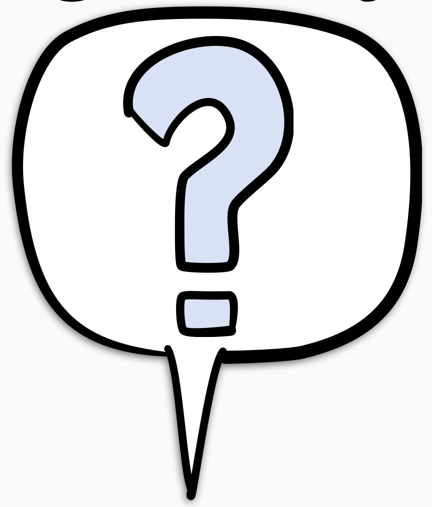
JSQ vs. M/M/1



Clear that $\mathbf{E}[N_k] \geq \mathbf{E}[N_1]$

Question: *upper bound* on $\mathbf{E}[N_k]$?

Intuition: should be close to $\mathbf{E}[N_1]$, because JSQ does a good job of keeping servers busy



Theorem:

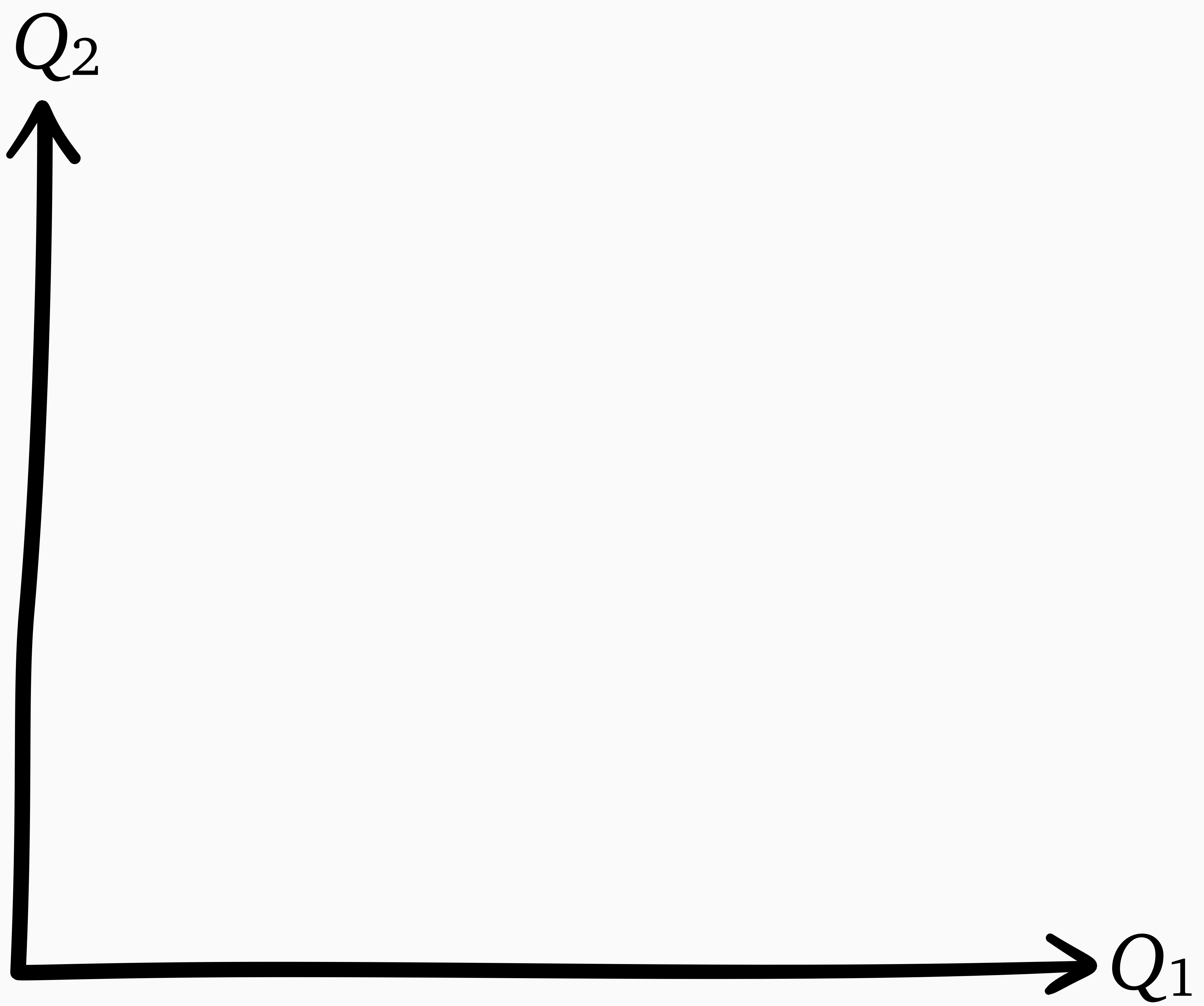
$$\mathbf{E}[N_k] = \mathbf{E}[N_1] + \frac{\mathbf{E}[(1 - B_k)N_k]}{1 - \rho}$$

$$\leq \mathbf{E}[N_1] + \sqrt{\frac{\mathbf{E}[\text{“variance of queue lengths”}]}{1 - \rho}}$$

Eryilmaz and Srikant: Asymptotically tight steady-state queue length bounds implied by drift conditions (2012)

State-space collapse

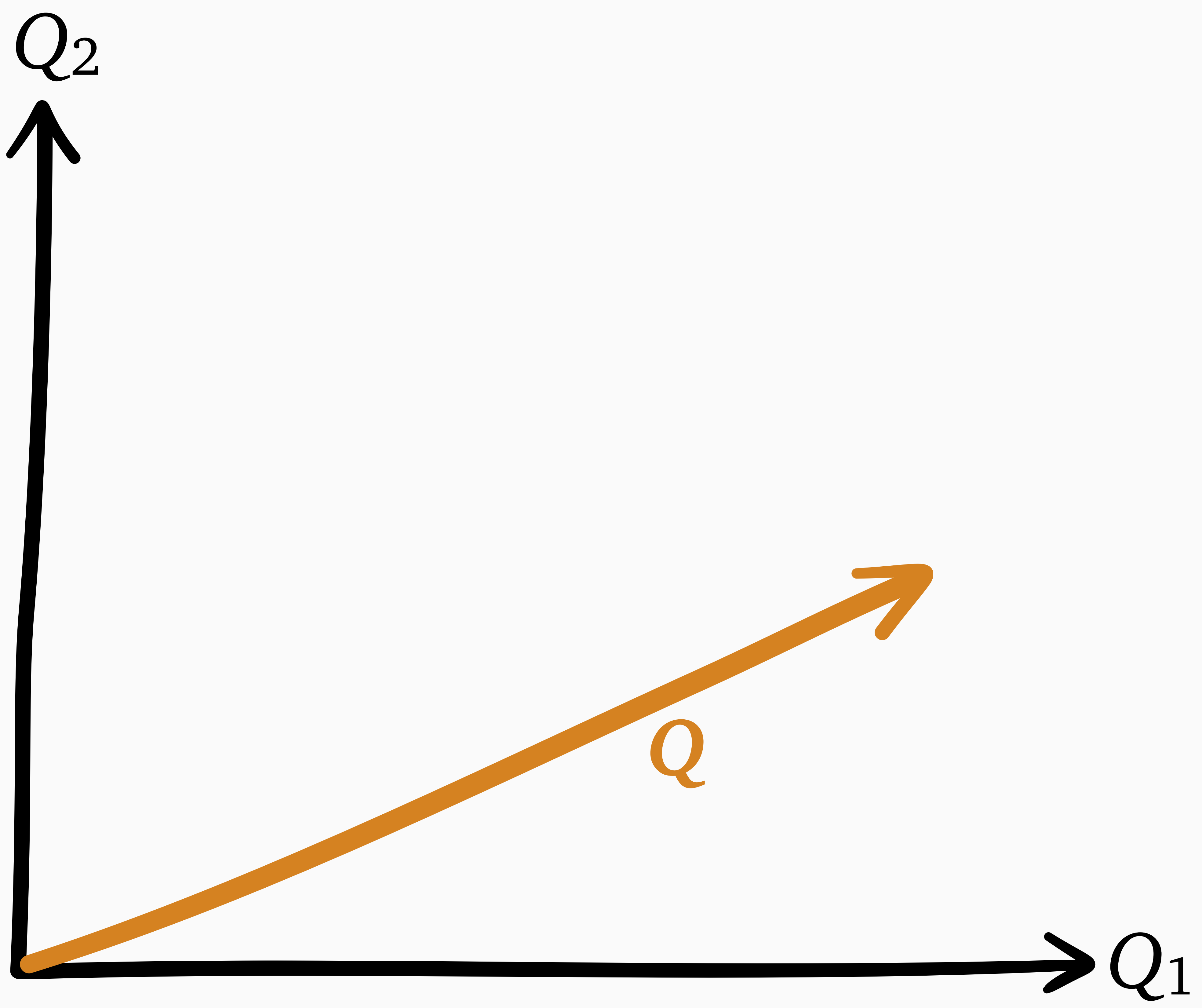
$Q_i = \#$ jobs at server i



State-space collapse

Q_i = # jobs at server i

$\boldsymbol{Q}$ = system state vector

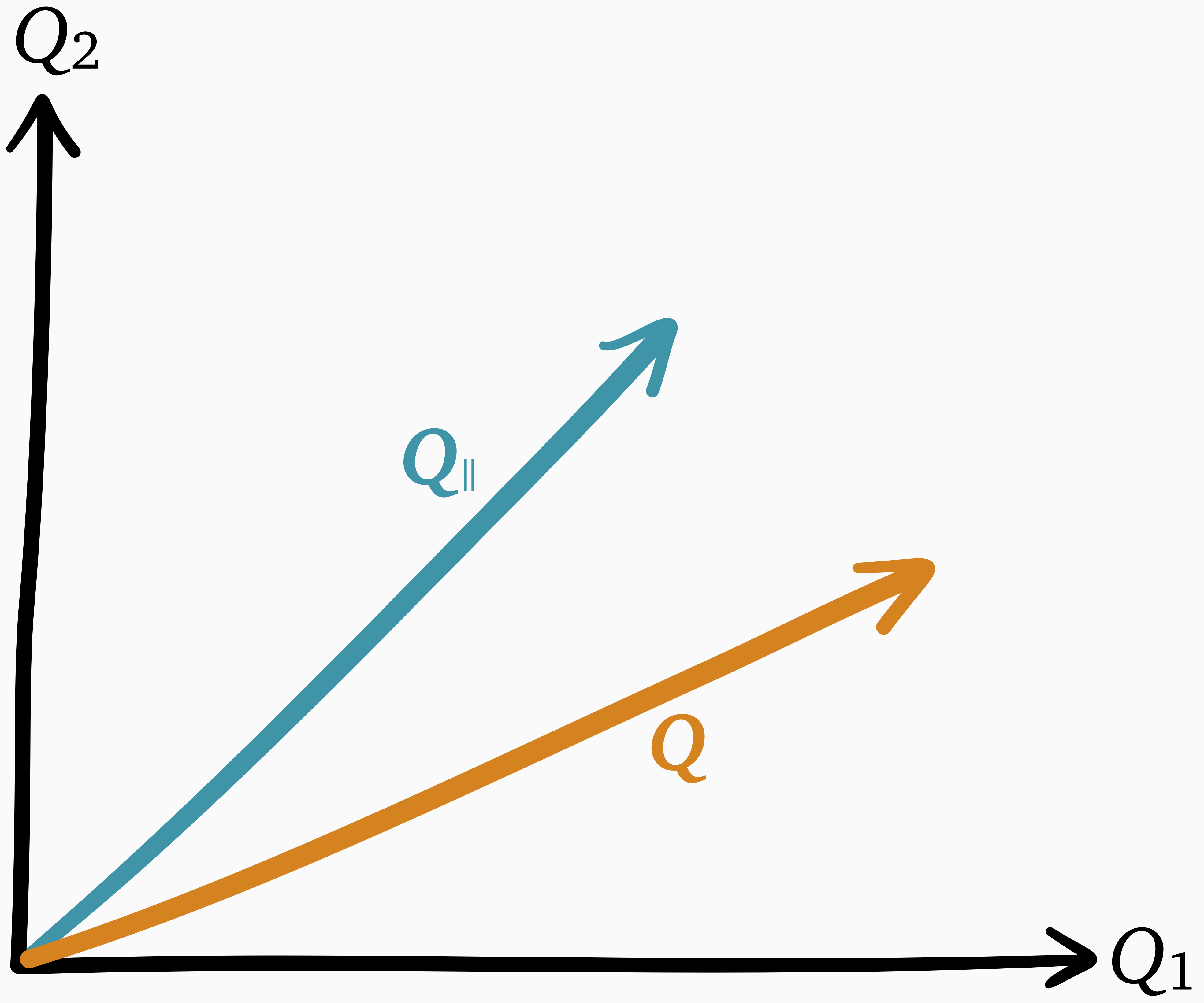


State-space collapse

Q_i = # jobs at server i

$\boldsymbol{Q}$ = system state vector

$\boldsymbol{Q}_{\parallel}$ = avg. # jobs per server = "**M/M/1** part"



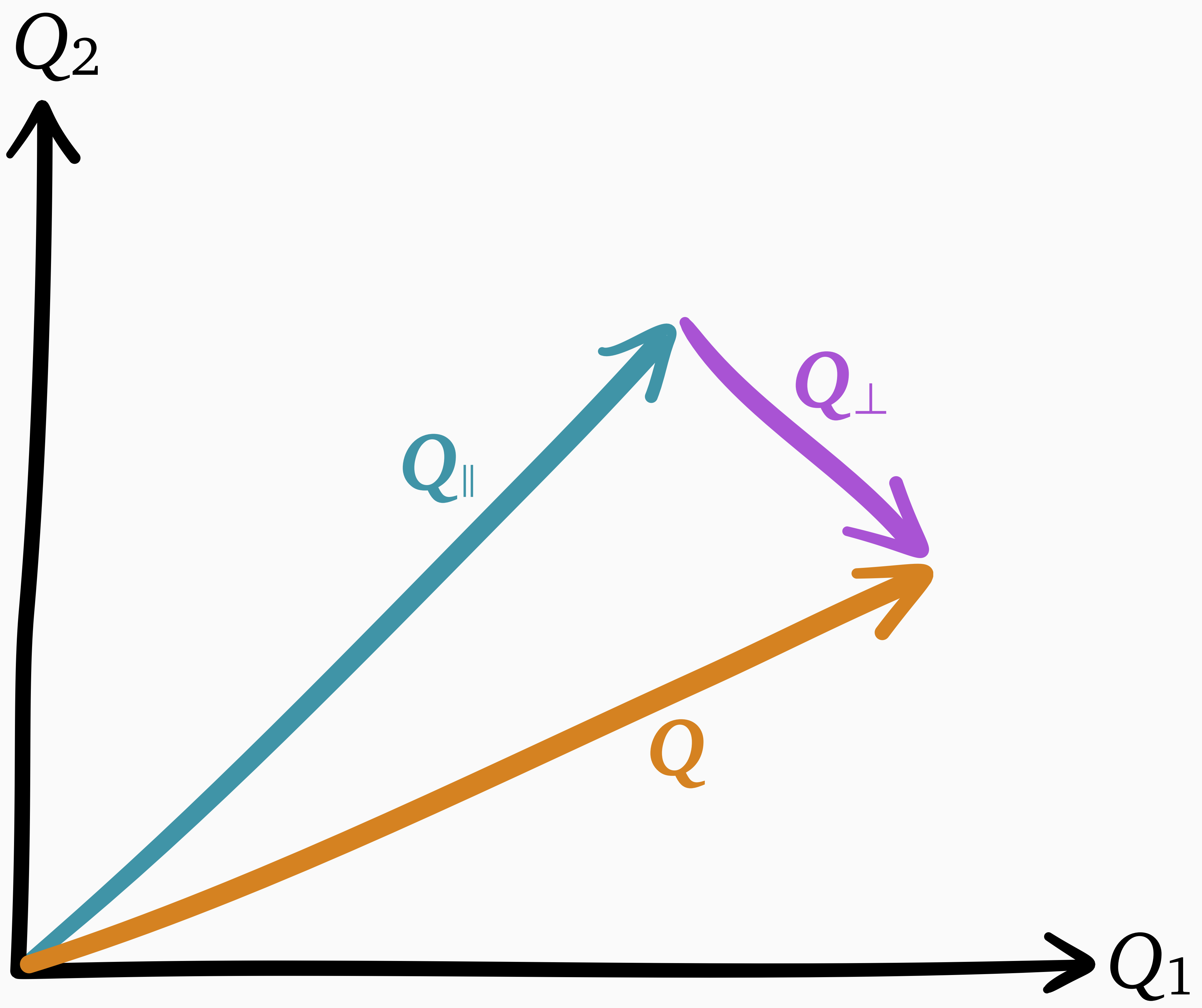
State-space collapse

Q_i = # jobs at server i

$\boldsymbol{Q}$ = system state vector

$\boldsymbol{Q}_{\parallel}$ = avg. # jobs per server = "**M/M/1** part"

$\|\boldsymbol{Q}_{\perp}\|^2$ = "variance of queue lengths"



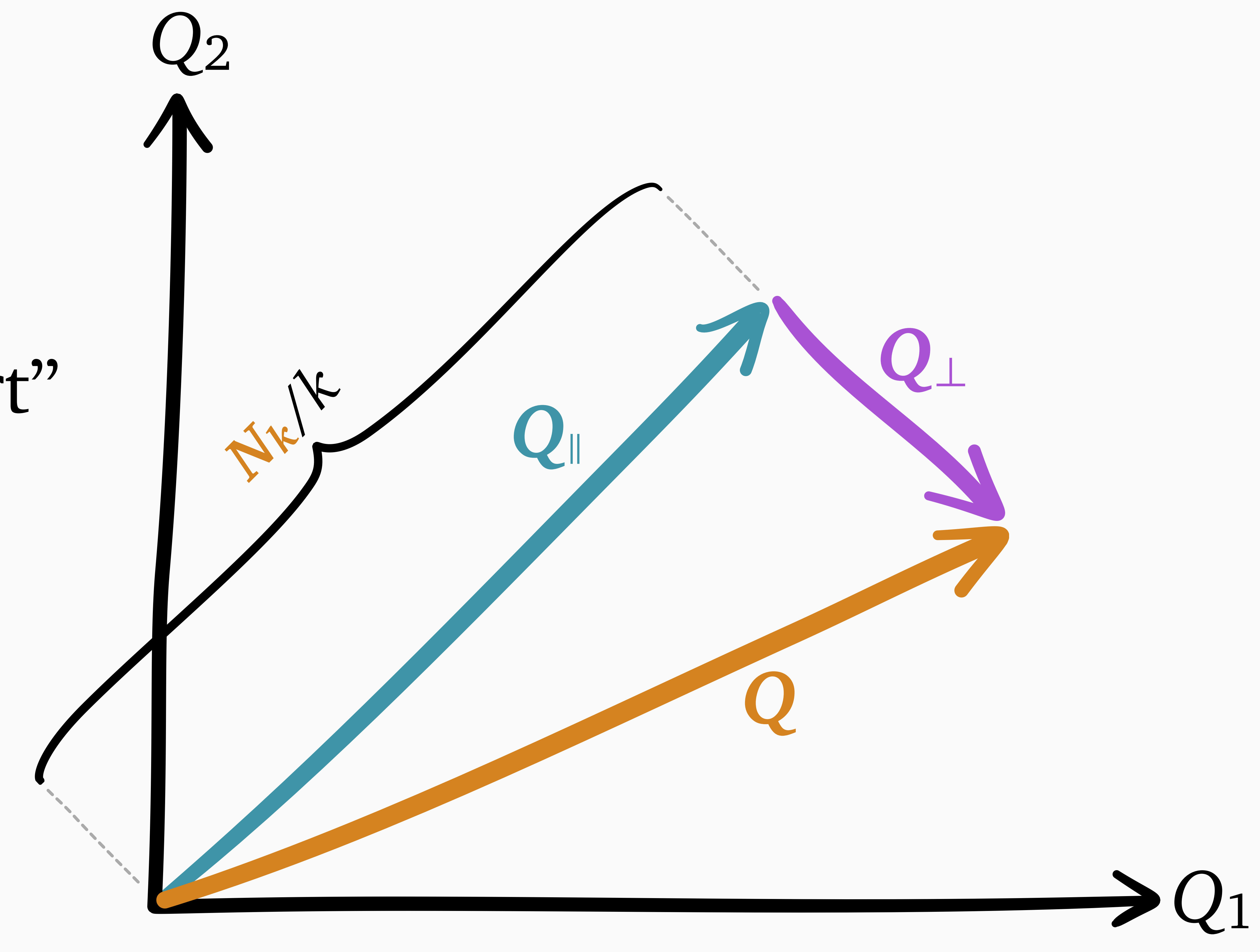
State-space collapse

Q_i = # jobs at server i

$\boldsymbol{Q}$ = system state vector

$\boldsymbol{Q}_{\parallel}$ = avg. # jobs per server = "**M/M/1** part"

$\|\boldsymbol{Q}_{\perp}\|^2$ = "variance of queue lengths"



State-space collapse

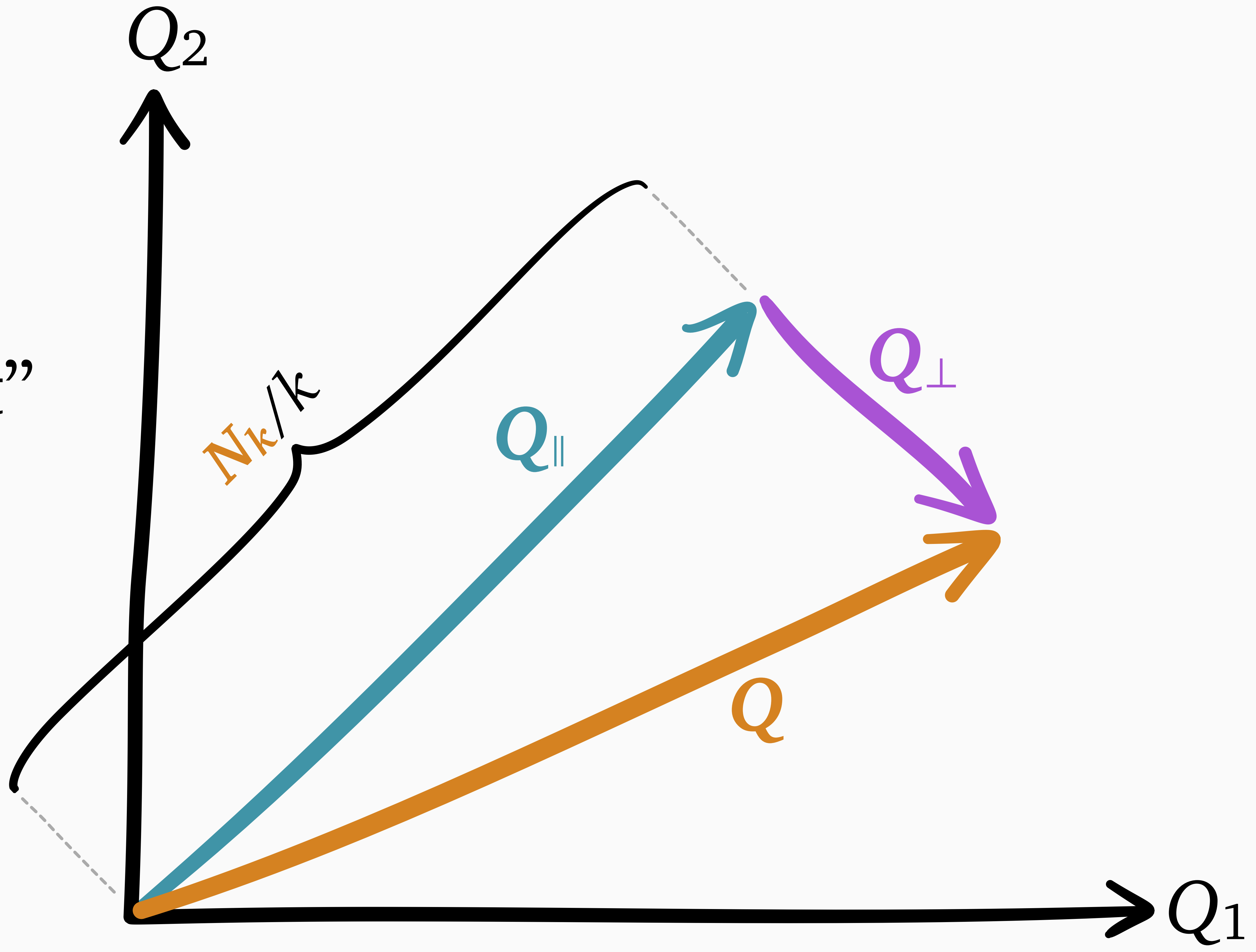
Q_i = # jobs at server i

$\boldsymbol{Q}$ = system state vector

$\boldsymbol{Q}_\parallel$ = avg. # jobs per server = "**M/M/1** part"

$\|\boldsymbol{Q}_\perp\|^2$ = "variance of queue lengths"

Drift method: use a Lyapunov function to show $\mathbf{E}[\|\boldsymbol{Q}_\perp\|^2] = O(1)$ as $\rho \to 1$



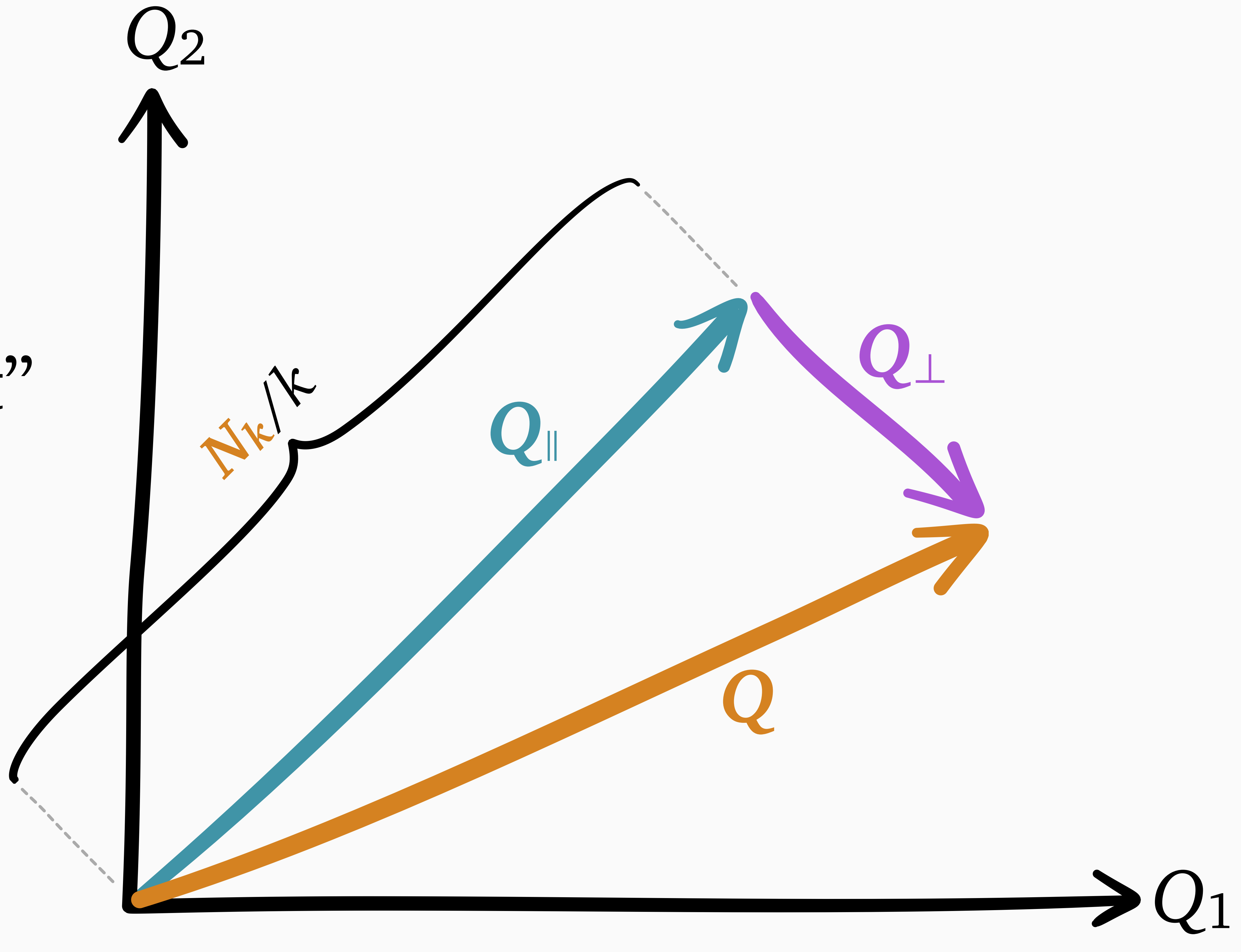
State-space collapse

Q_i = # jobs at server i

$\boldsymbol{Q}$ = system state vector

$\boldsymbol{Q}_\parallel$ = avg. # jobs per server = "**M/M/1** part"

$\|\boldsymbol{Q}_\perp\|^2$ = "variance of queue lengths"



Drift method: use a Lyapunov function to show $\mathbf{E}[\|\boldsymbol{Q}_\perp\|^2] = O(1)$ as $\rho \to 1$

$$\mathbf{E}[N_k] = \mathbf{E}[N_1] + o\left(\frac{1}{1-\rho}\right)$$

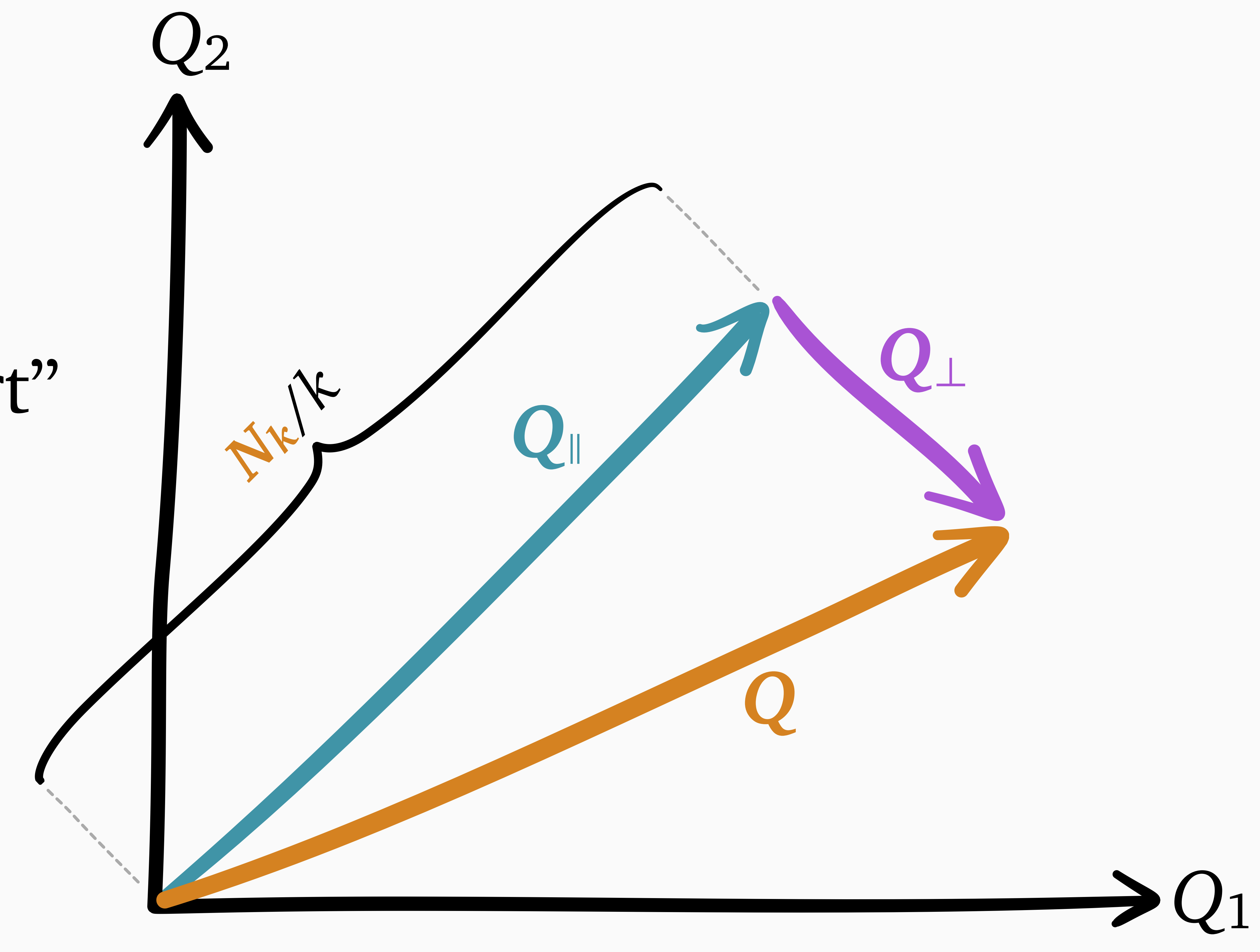
State-space collapse

Q_i = # jobs at server i

$\boldsymbol{Q}$ = system state vector

$\boldsymbol{Q}_\parallel$ = avg. # jobs per server = "**M/M/1** part"

$\|\boldsymbol{Q}_\perp\|^2$ = "variance of queue lengths"



Drift method: use a Lyapunov function to show $\mathbf{E}[\|\boldsymbol{Q}_\perp\|^2] = O(1)$ as $\rho \to 1$

$\frac{\rho}{1-\rho}$

$$\mathbf{E}[N_k] = \mathbf{E}[N_1] + o\left(\frac{1}{1-\rho}\right)$$

State-space collapse

Q_i = # jobs at server i

$\boldsymbol{Q}$ = system state vector

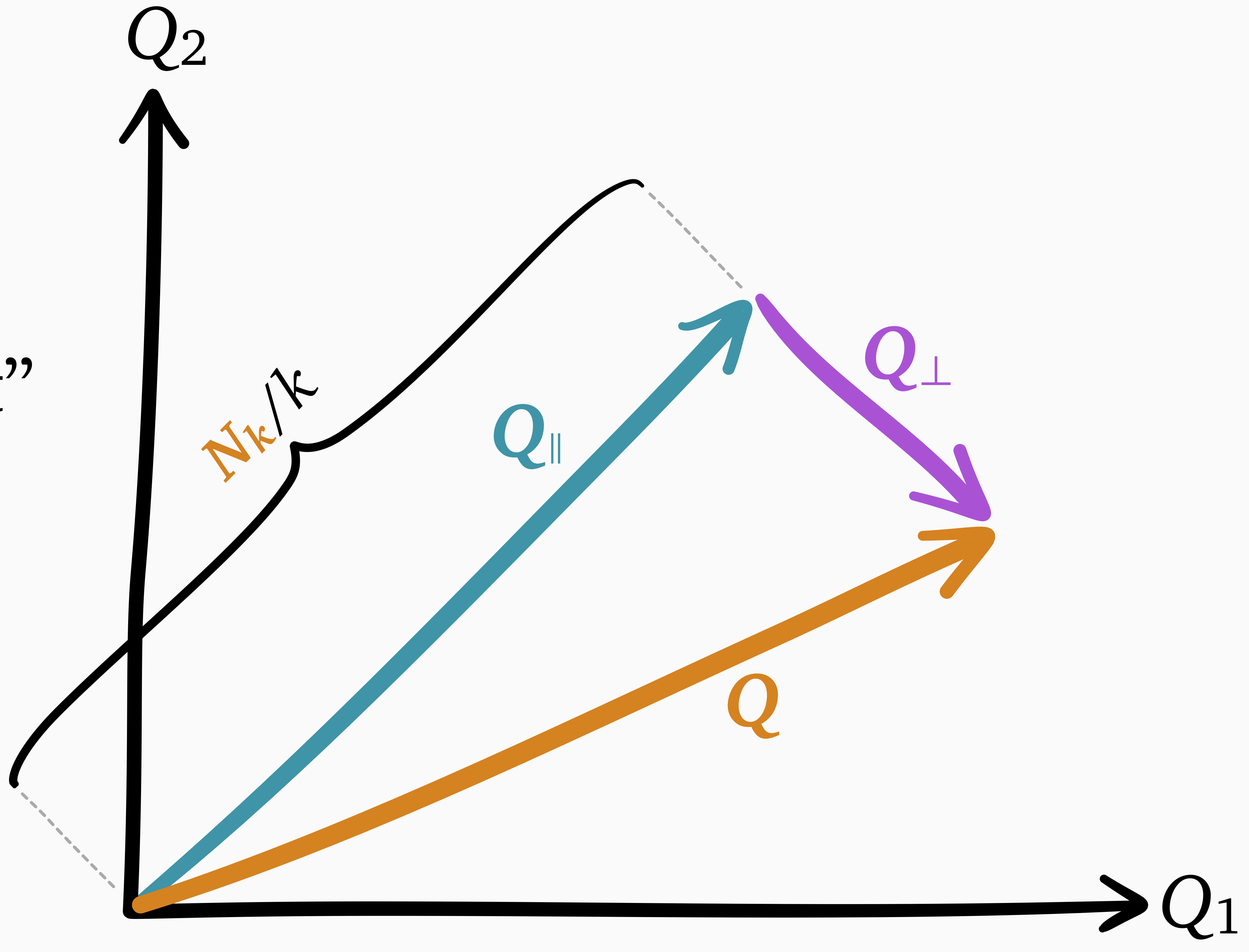
$\boldsymbol{Q}_\parallel$ = avg. # jobs per server = "**M/M/1** part"

$\|\boldsymbol{Q}_\perp\|^2$ = "variance of queue lengths"

Drift method: use a Lyapunov function to show $\mathbf{E}[\|\boldsymbol{Q}_\perp\|^2] = O(1)$ as $\rho \to 1$

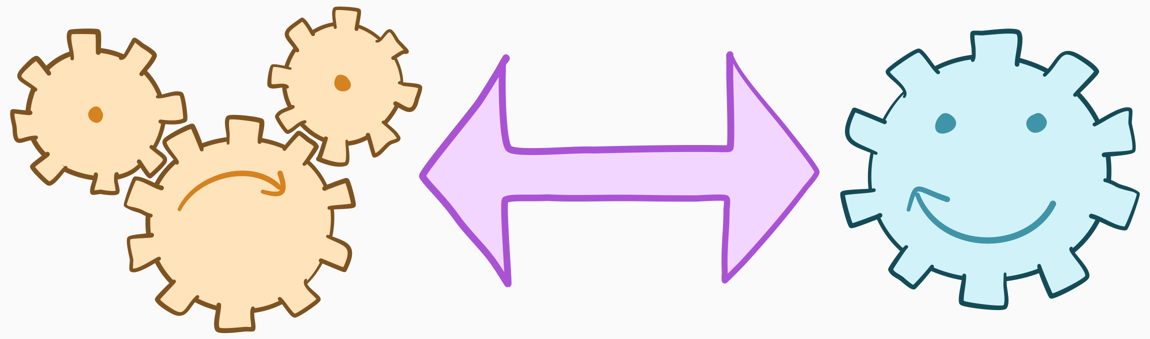
$\frac{\rho}{1-\rho}$

$$\mathbf{E}[N_k] = \mathbf{E}[N_1] + o\left(\frac{1}{1-\rho}\right)$$



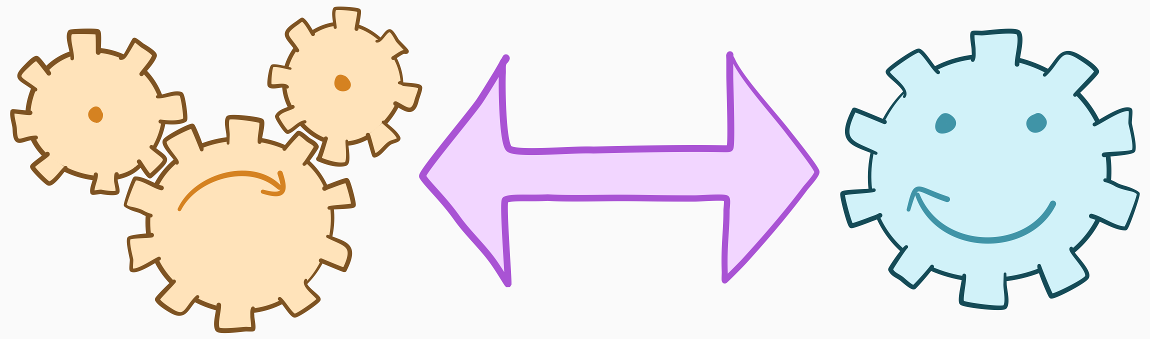
Similar results in switch scheduling and more—see SIGMETRICS 2021 tutorial by Maguluri and Chen

Classifying coupling techniques



	A. Every sample path	B. Steady-state distribution
1. More information	A1 BIG online knapsack (via compensated coupling)	B1
2. Fewer constraints	A2 M/M/k vs. M/M/1 Online knapsack (via constraints-to-costs)	B2
3. Simpler dynamics	A3 SIS epidemics Queues with redundancy	B3 Stein's method

Classifying coupling techniques



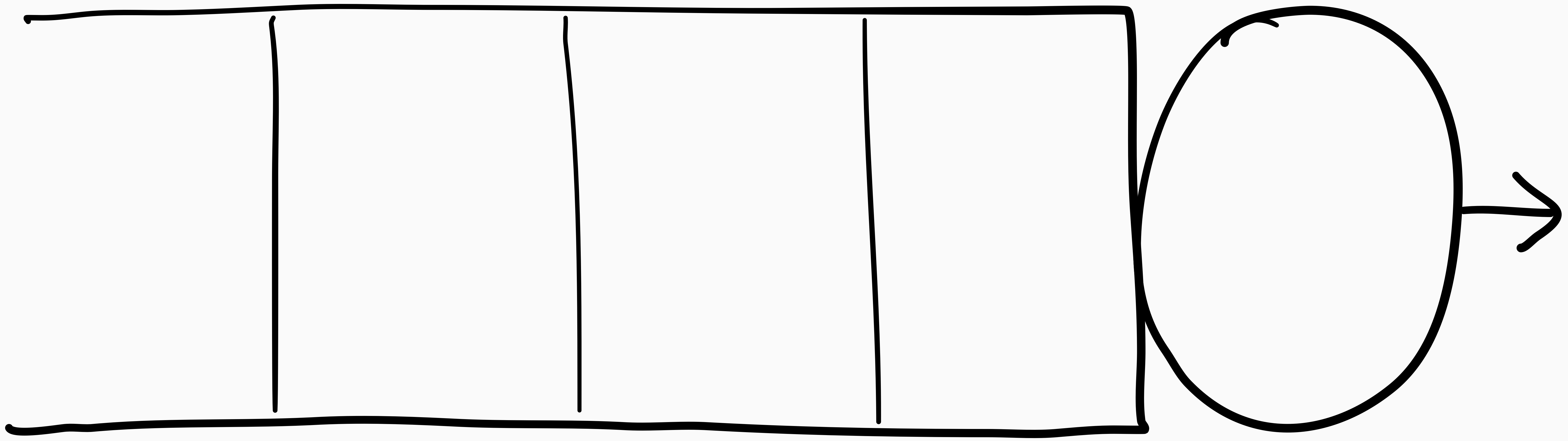
	A. Every sample path	B. Steady-state distribution
1. More information	A1 BIG online knapsack (via compensated coupling)	B1
2. Fewer constraints	A2 M/M/k vs. M/M/1 Online knapsack (via constraints-to-costs)	B2 State-space collapse (load balancing, switch scheduling)
3. Simpler dynamics	A3 SIS epidemics Queues with redundancy	B3 Stein's method

In-Depth Study 2:
Gittins in the M/G/k

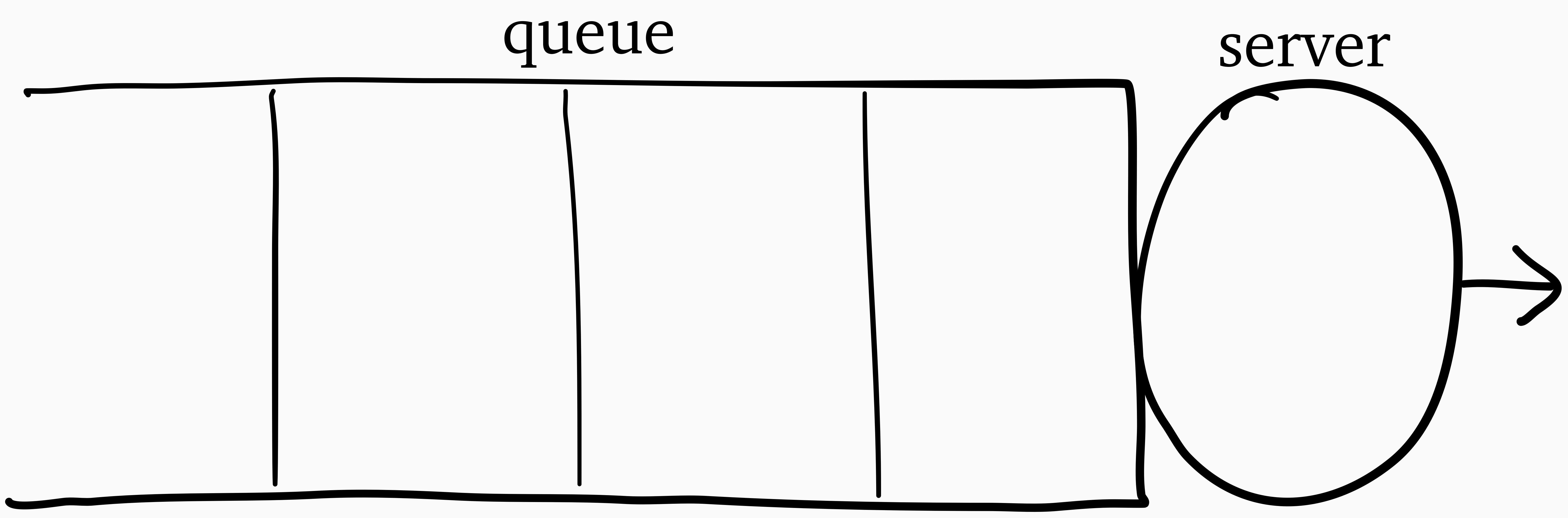
In-Depth Study 2:
Gittins in the M/G/k

Scully, Grosof, and Harchol-Balter: The Gittins Policy is Nearly Optimal in the M/G/k under Extremely General Conditions (2020)

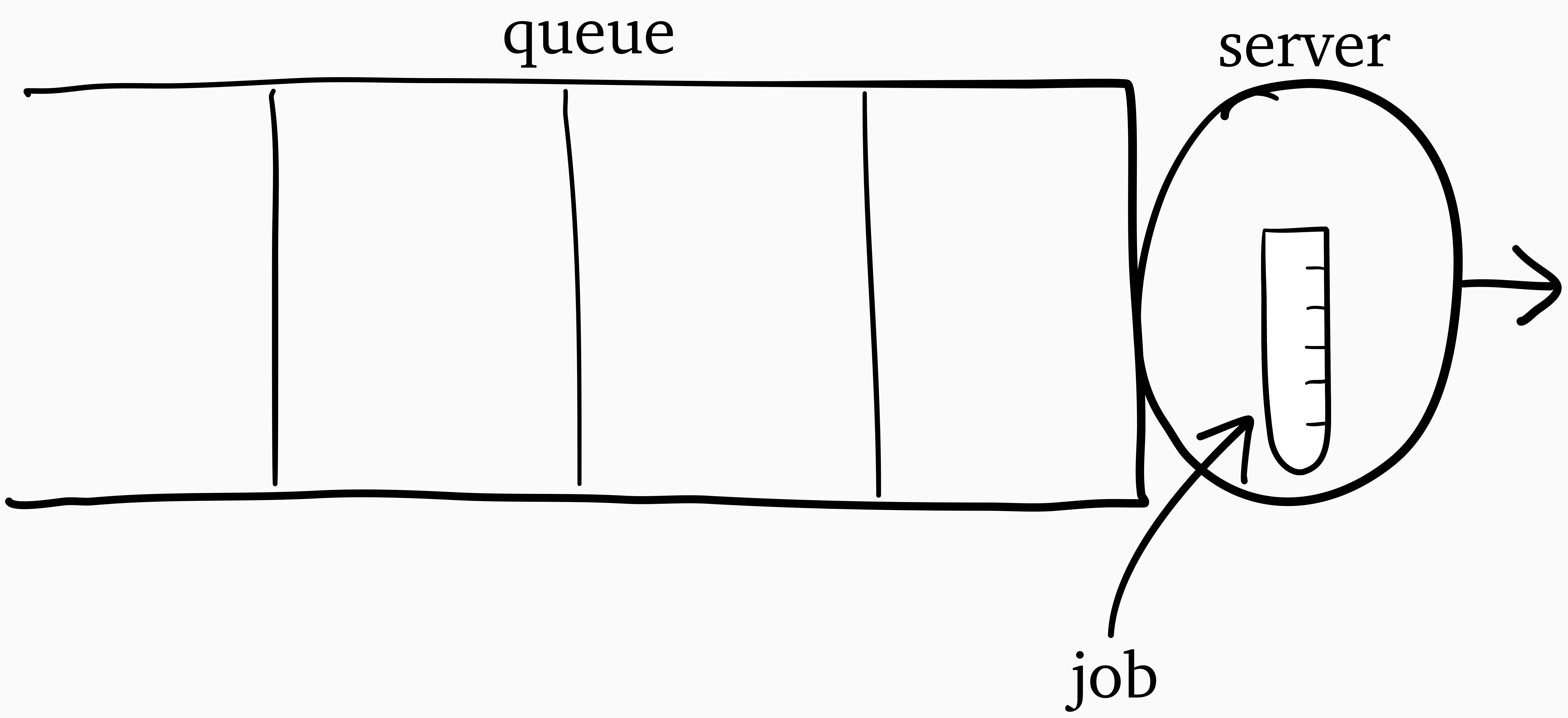
M/G/1 queue



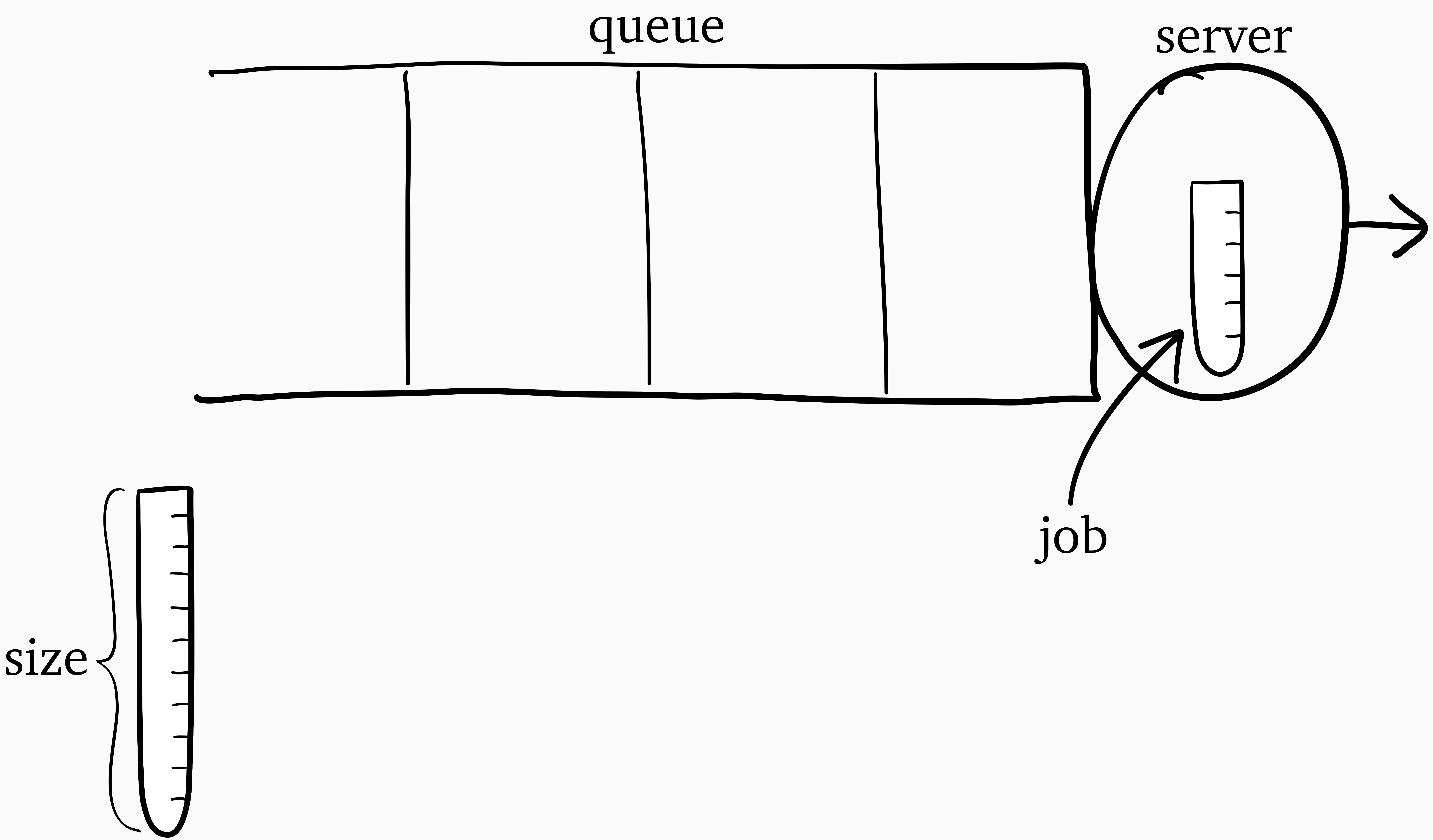
M/G/1 queue



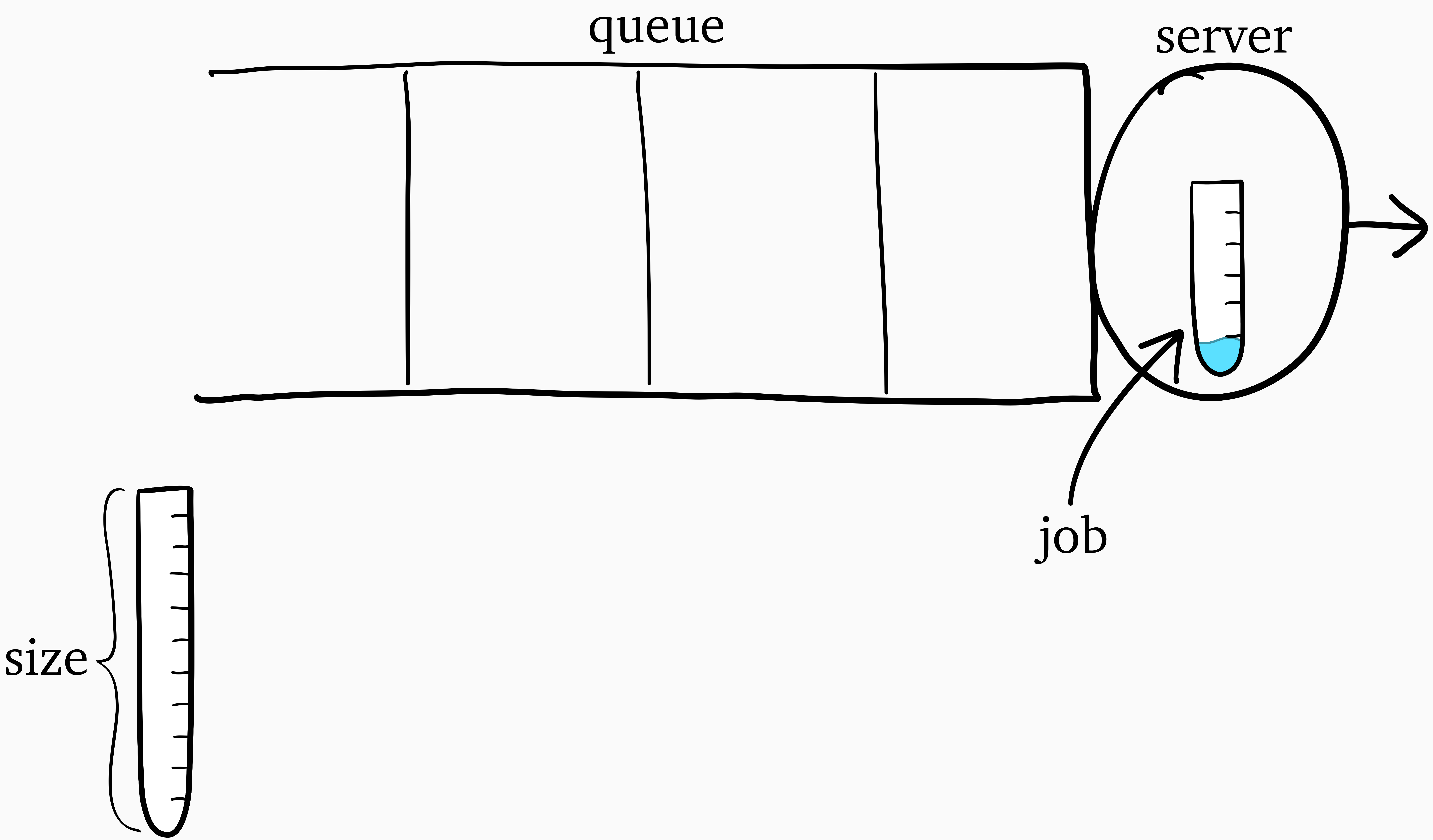
M/G/1 queue



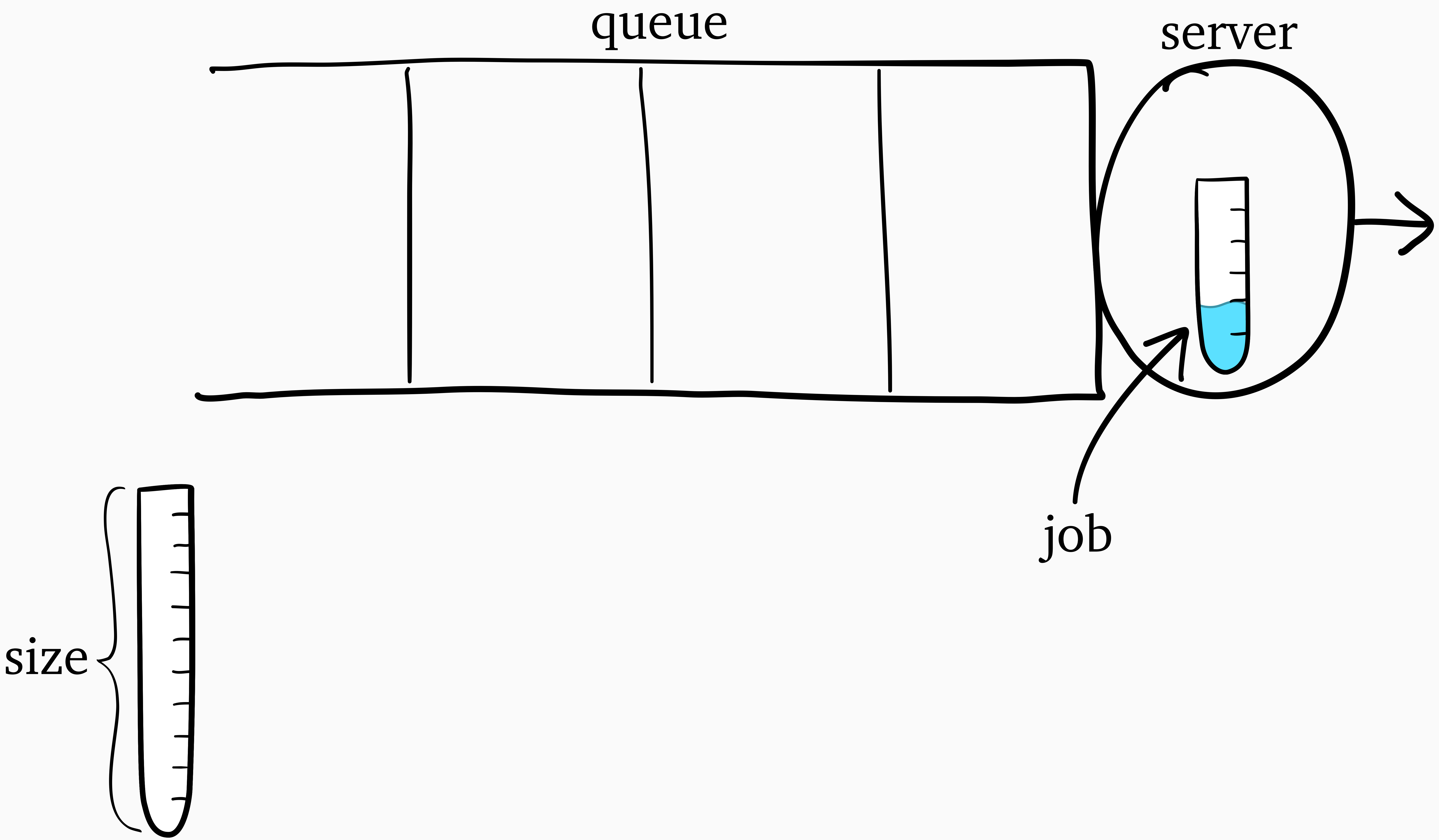
M/G/1 queue



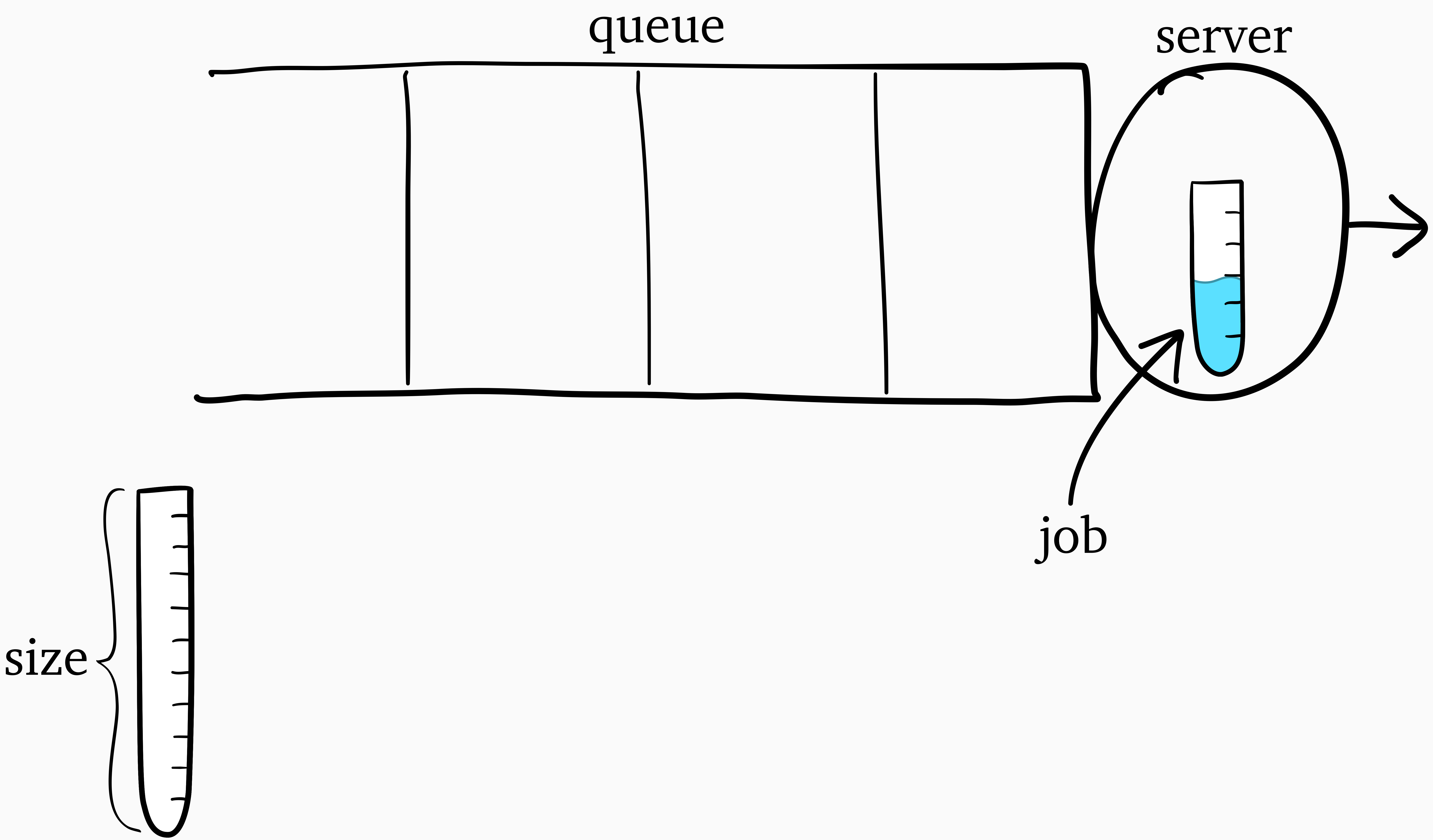
M/G/1 queue



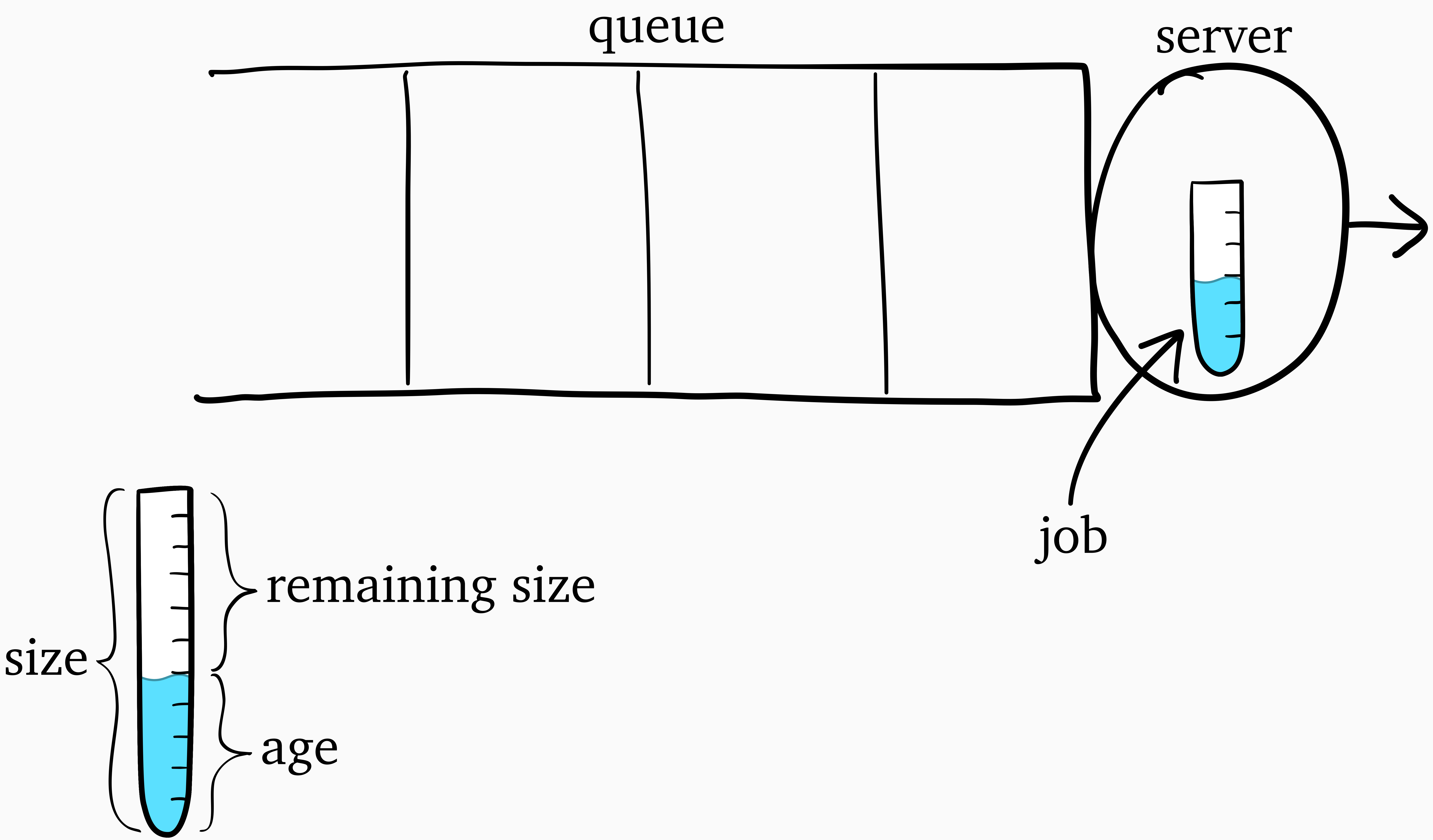
M/G/1 queue



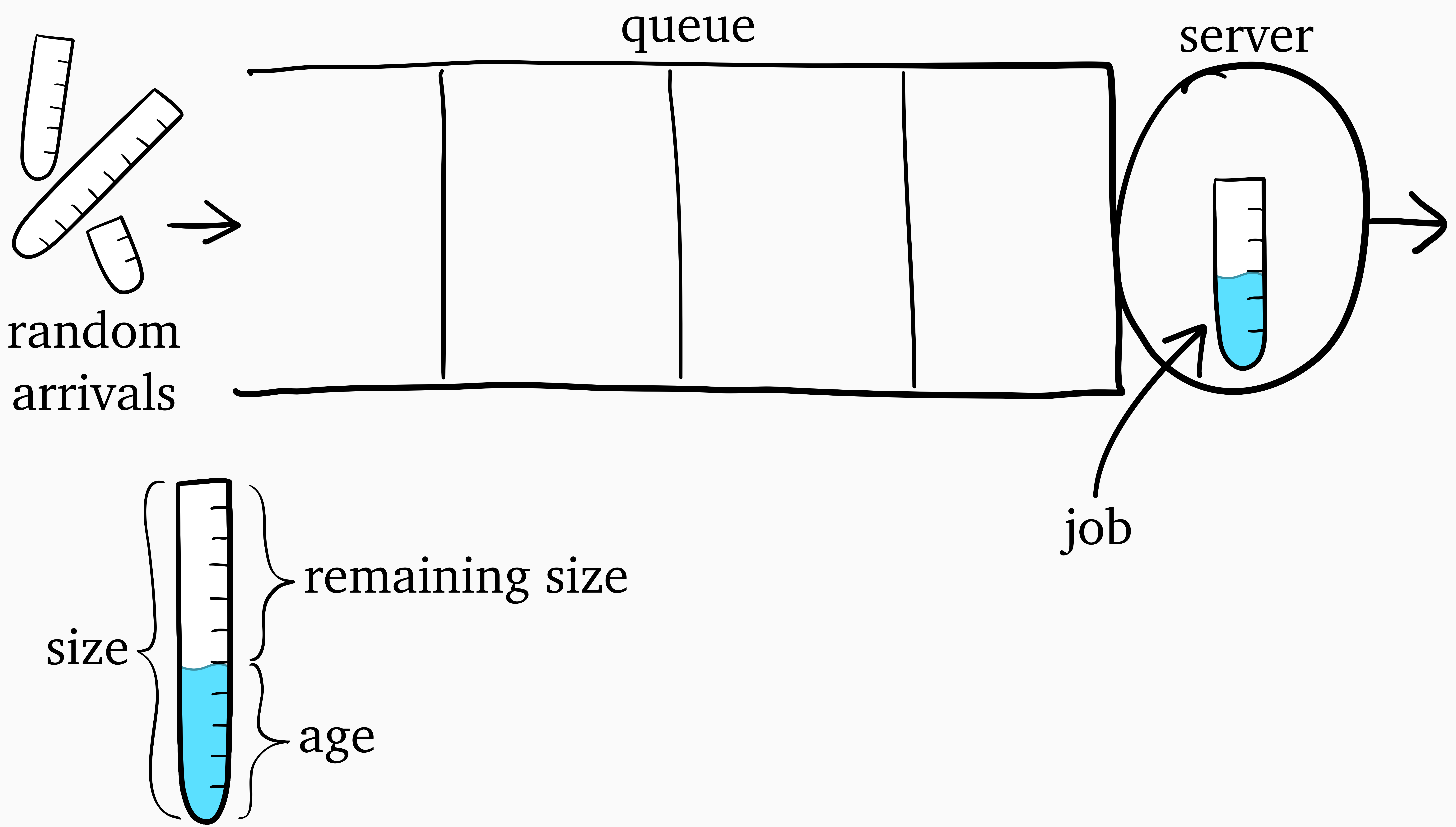
M/G/1 queue



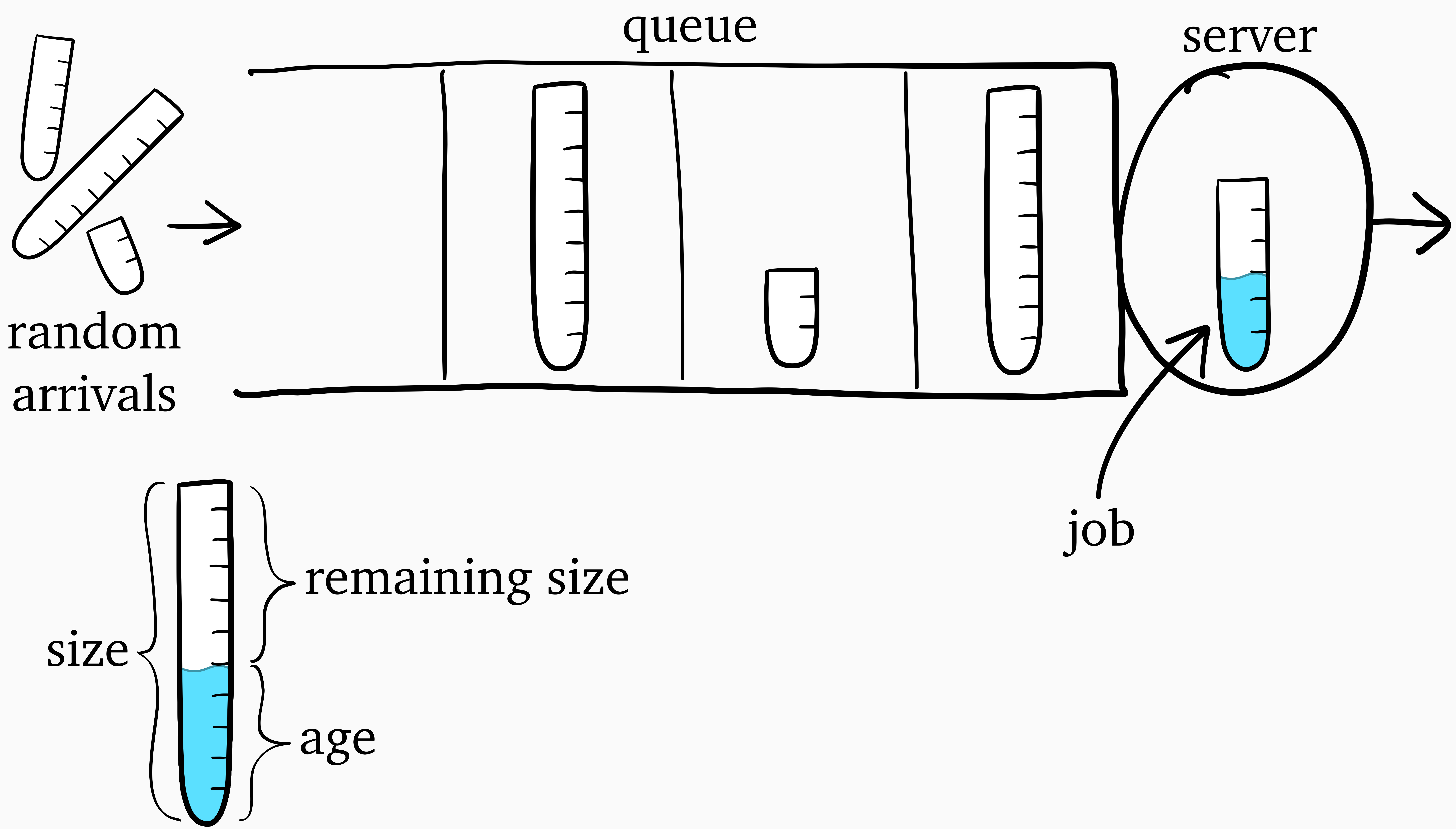
M/G/1 queue



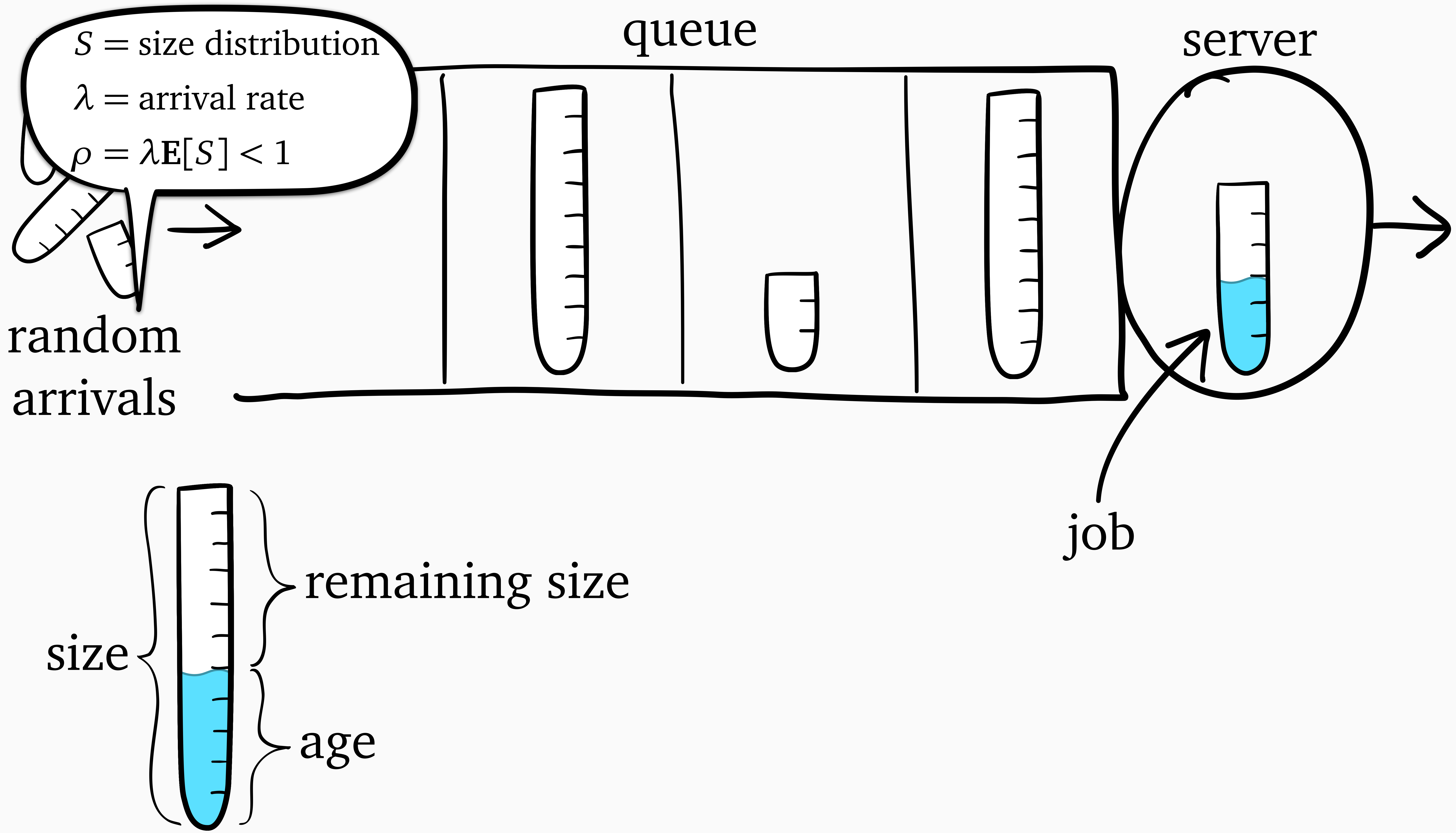
M/G/1 queue



M/G/1 queue

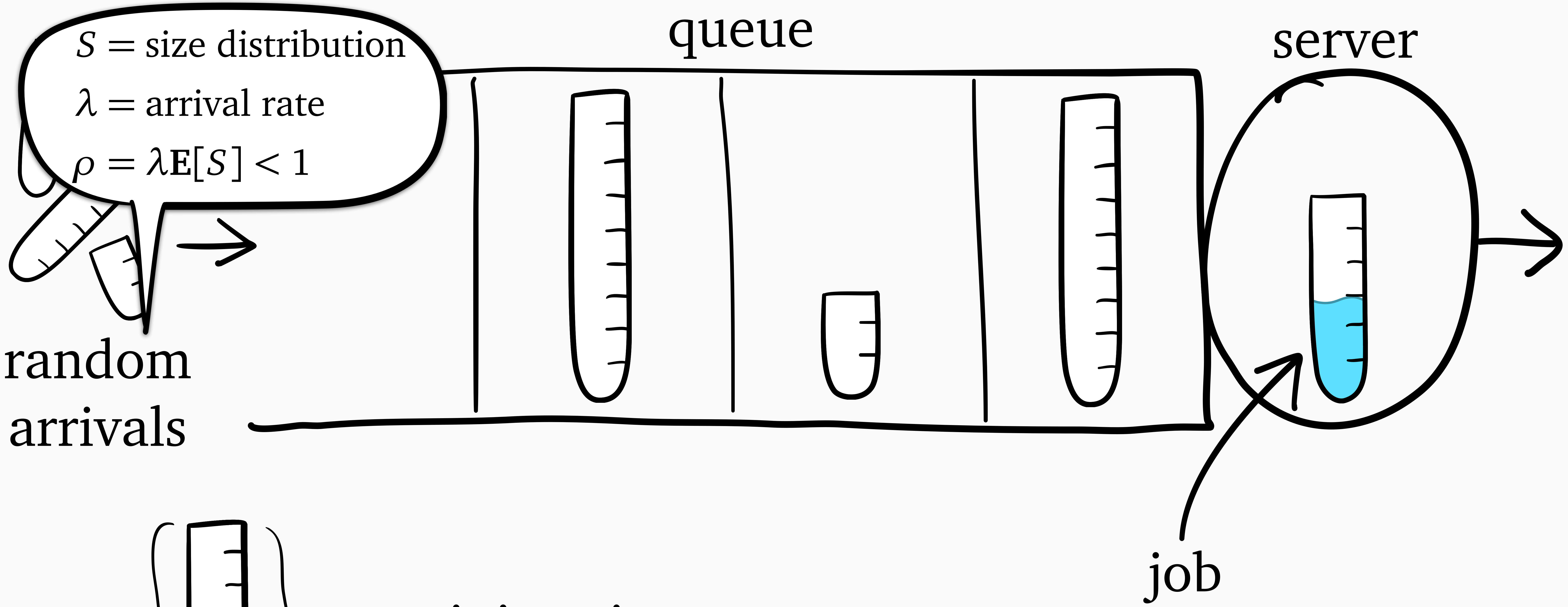


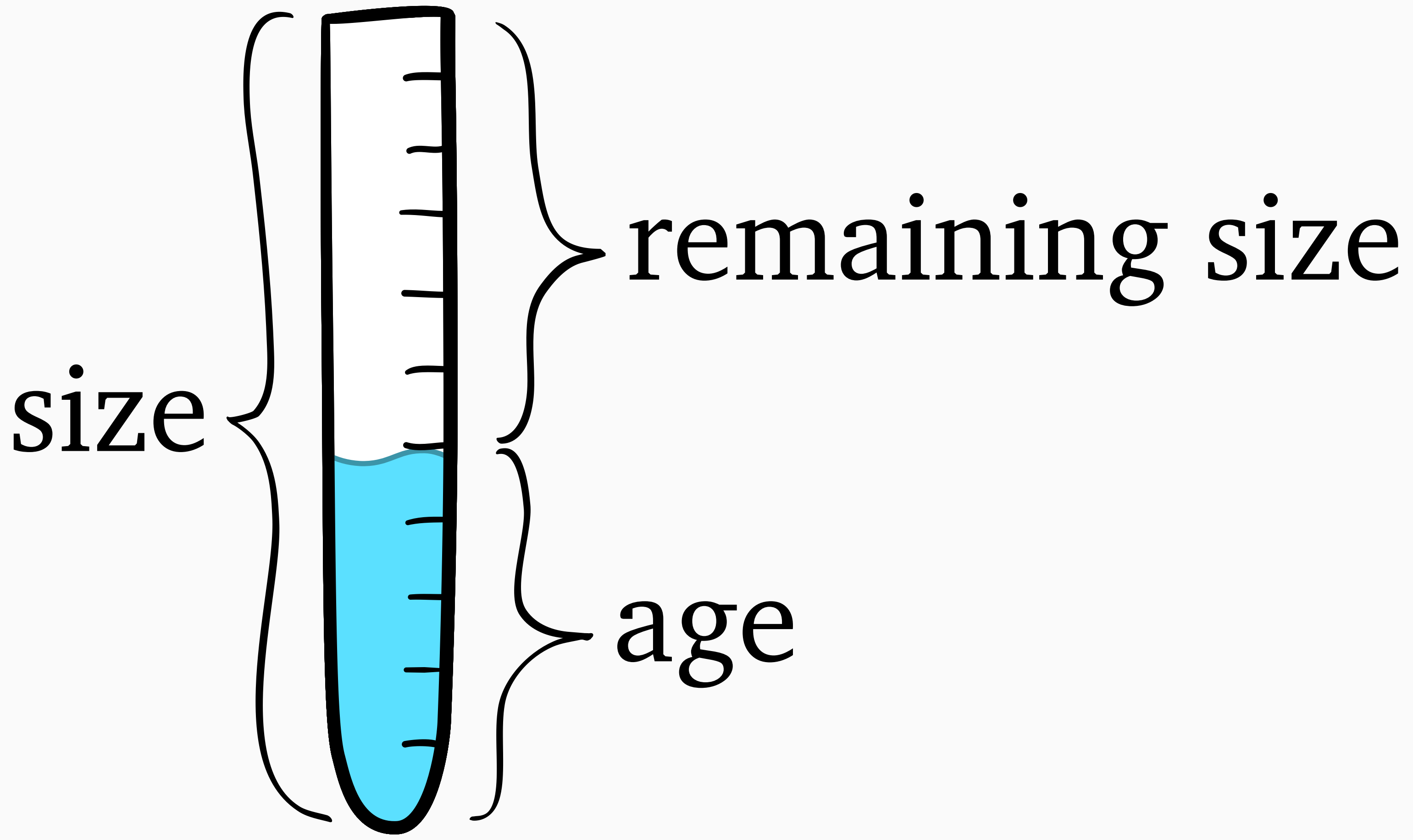
M/G/1 queue

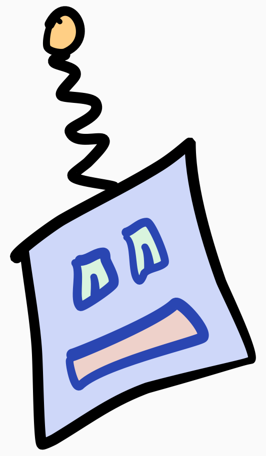


S = size distribution
λ = arrival rate
$\rho = \lambda\mathbf{E}[S] < 1$
random arrivals
queue
server
job
size
remaining size
age

M/G/1 queue

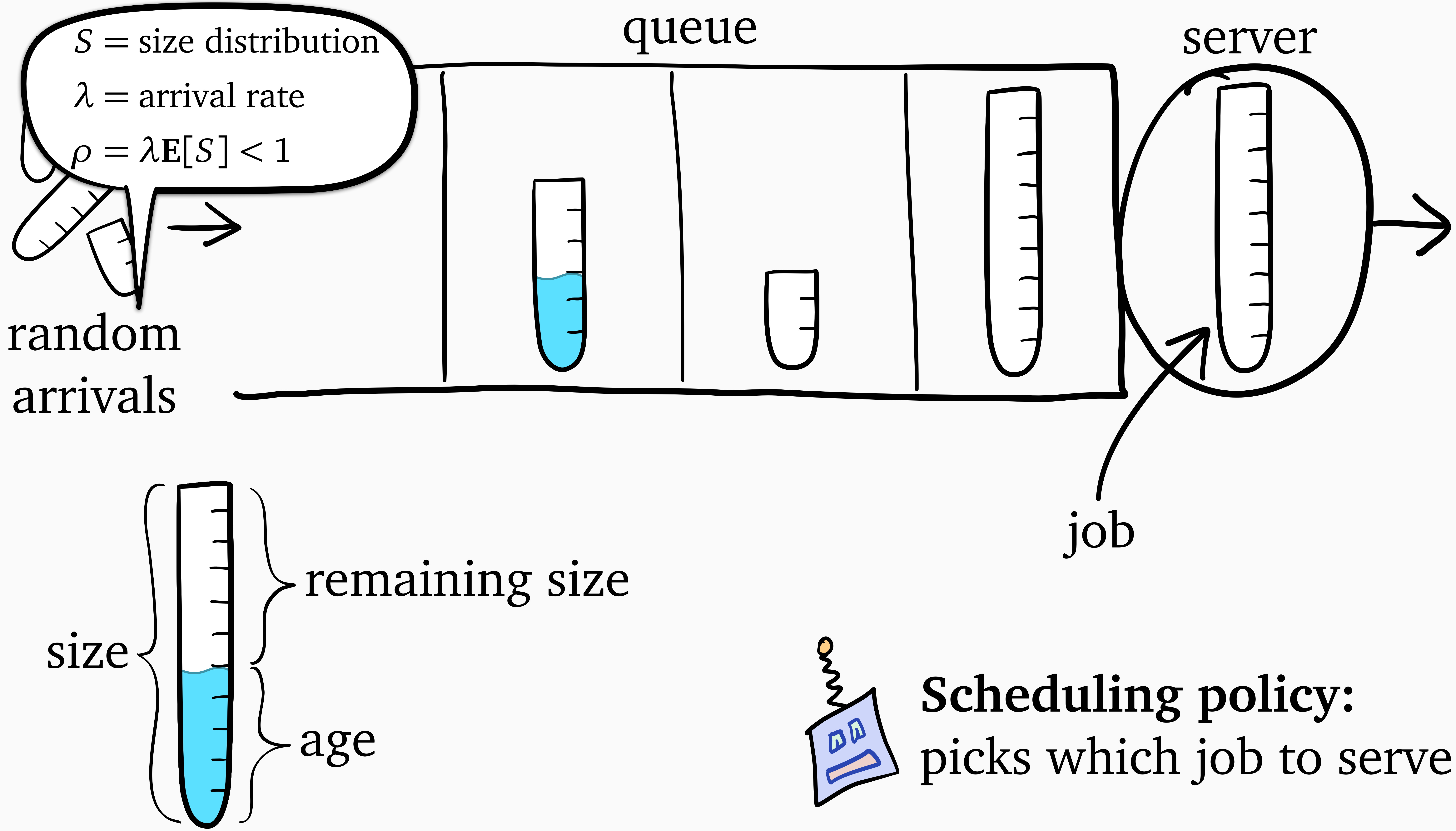






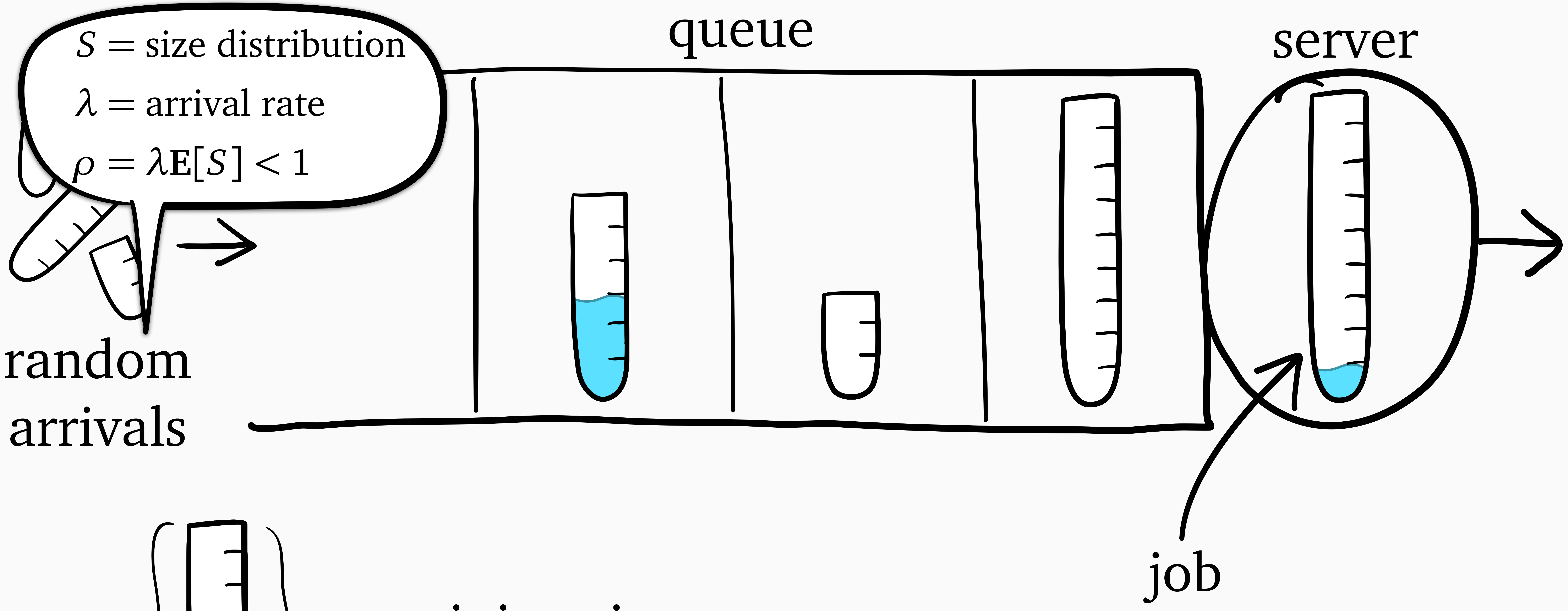
Scheduling policy:
picks which job to serve

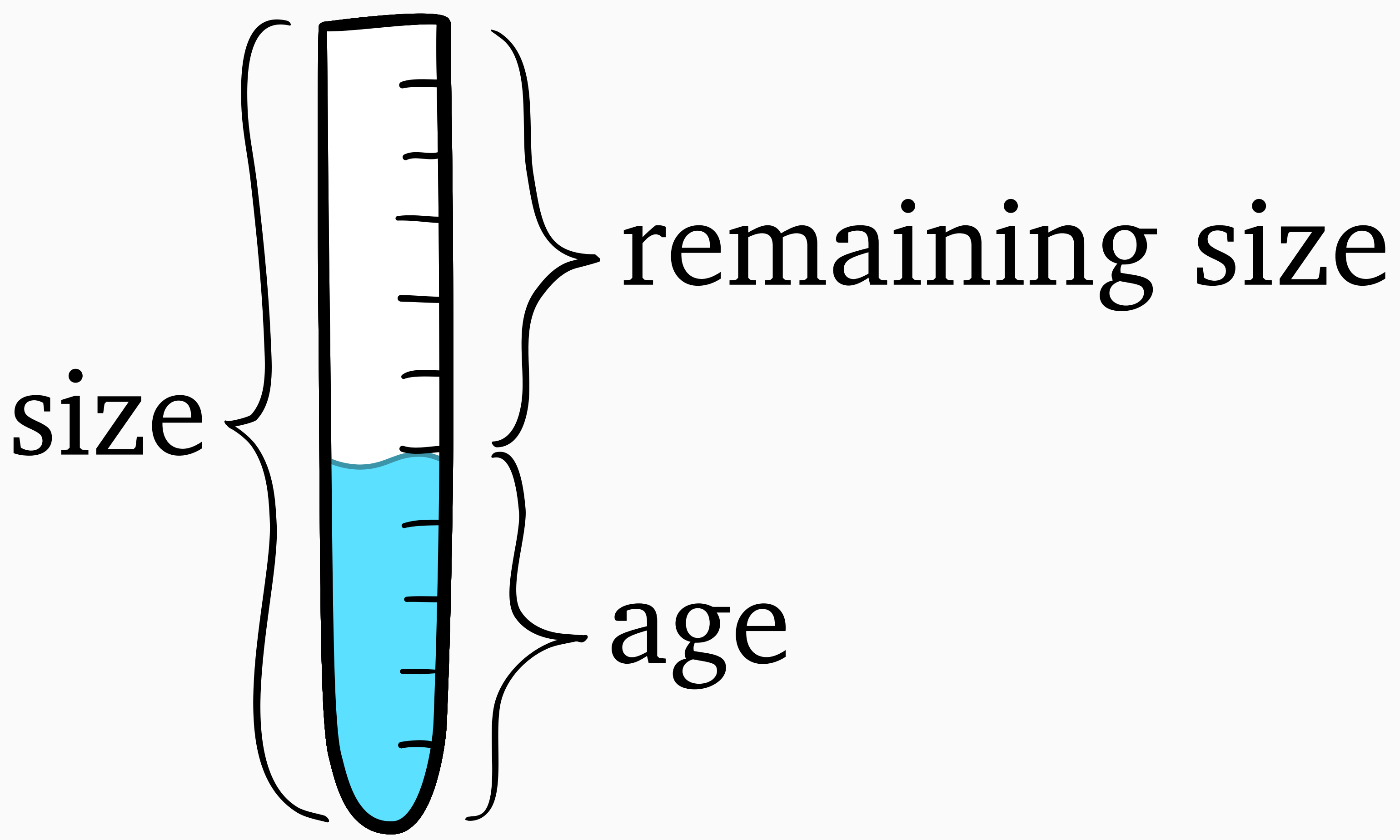
M/G/1 queue

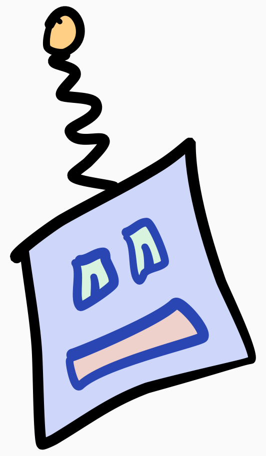


Scheduling policy:
picks which job to serve

M/G/1 queue

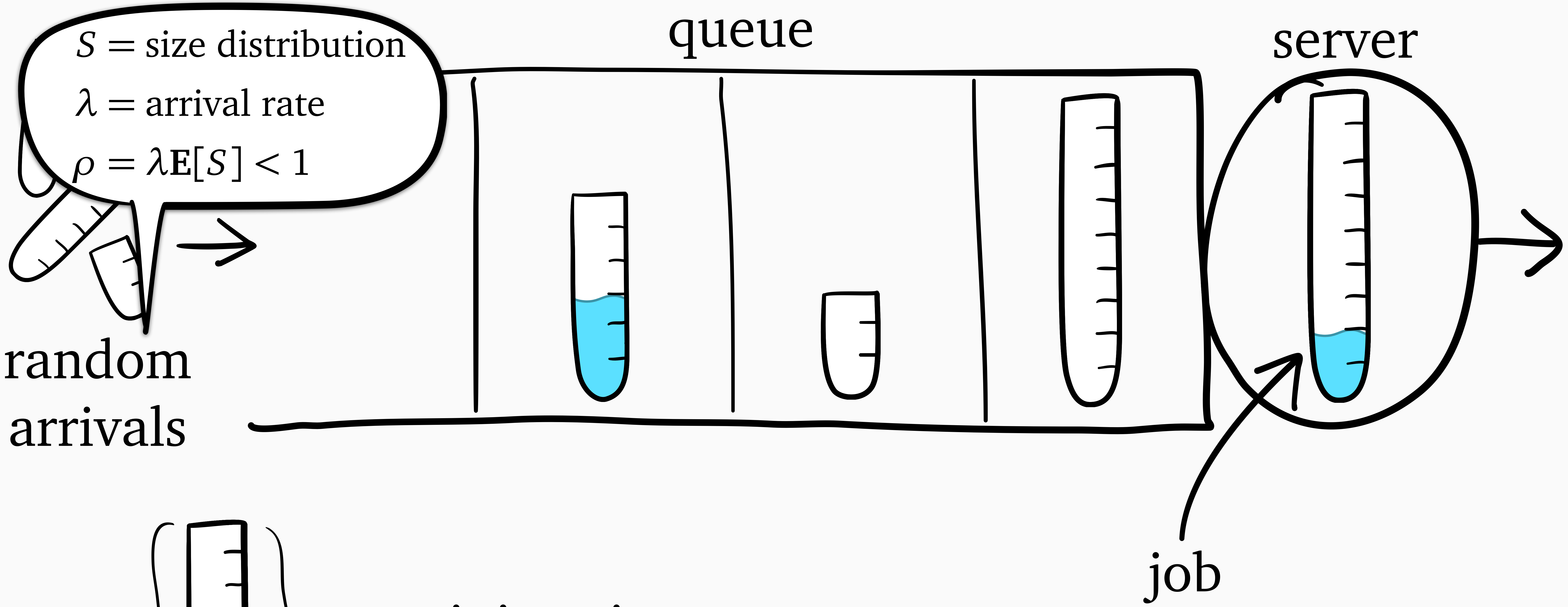


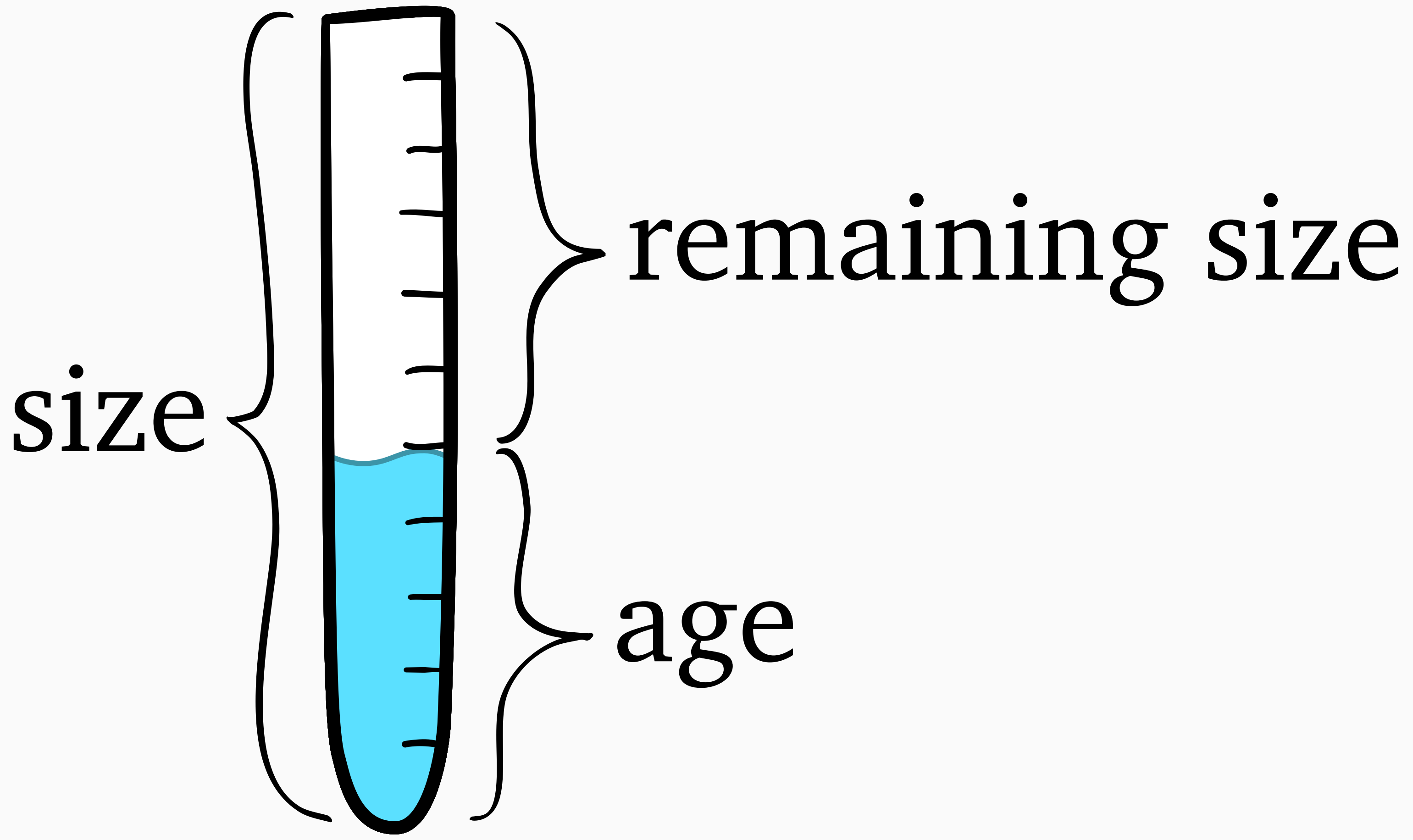


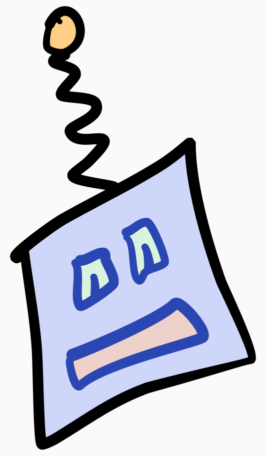


Scheduling policy:
picks which job to serve

M/G/1 queue

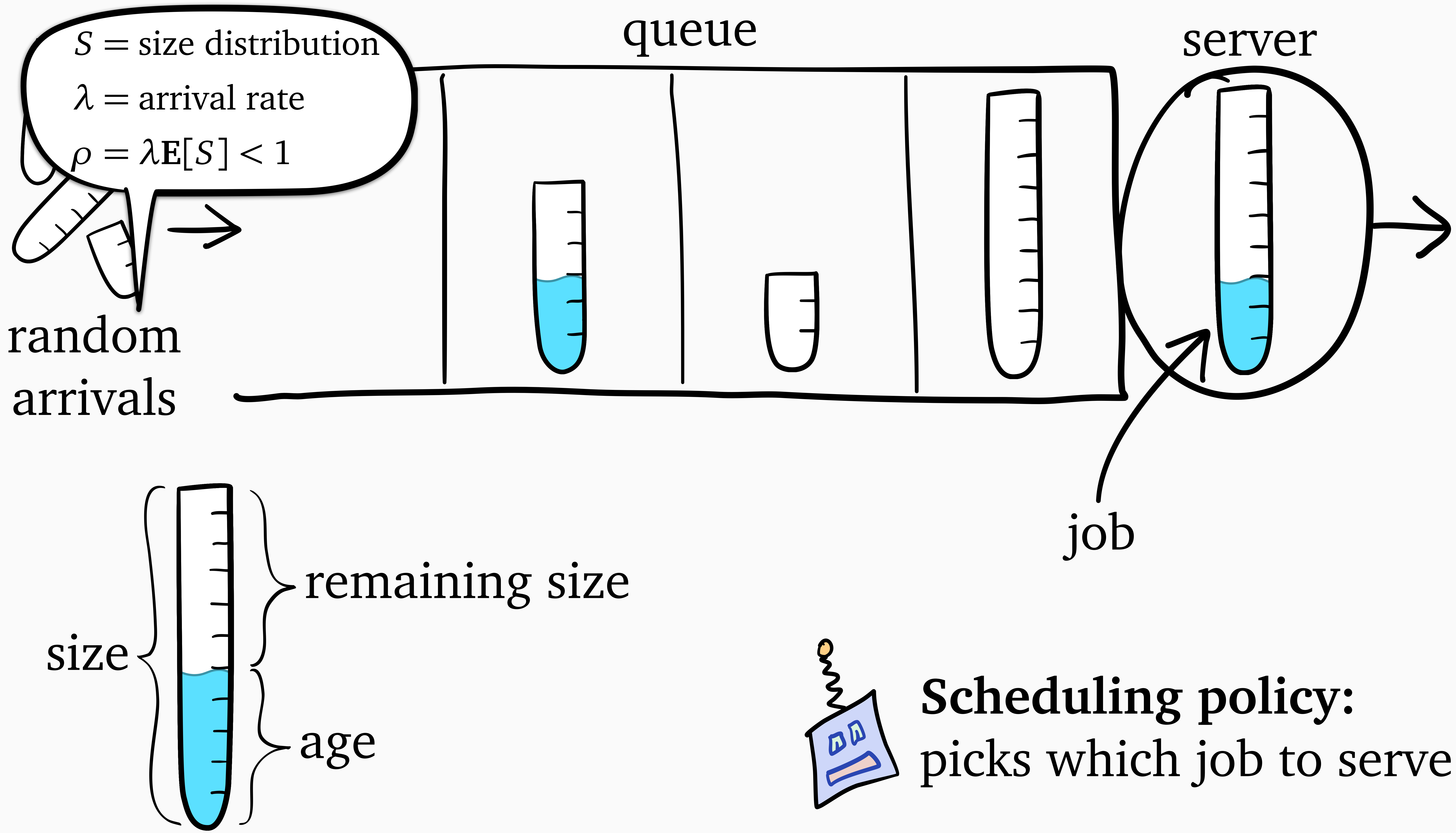






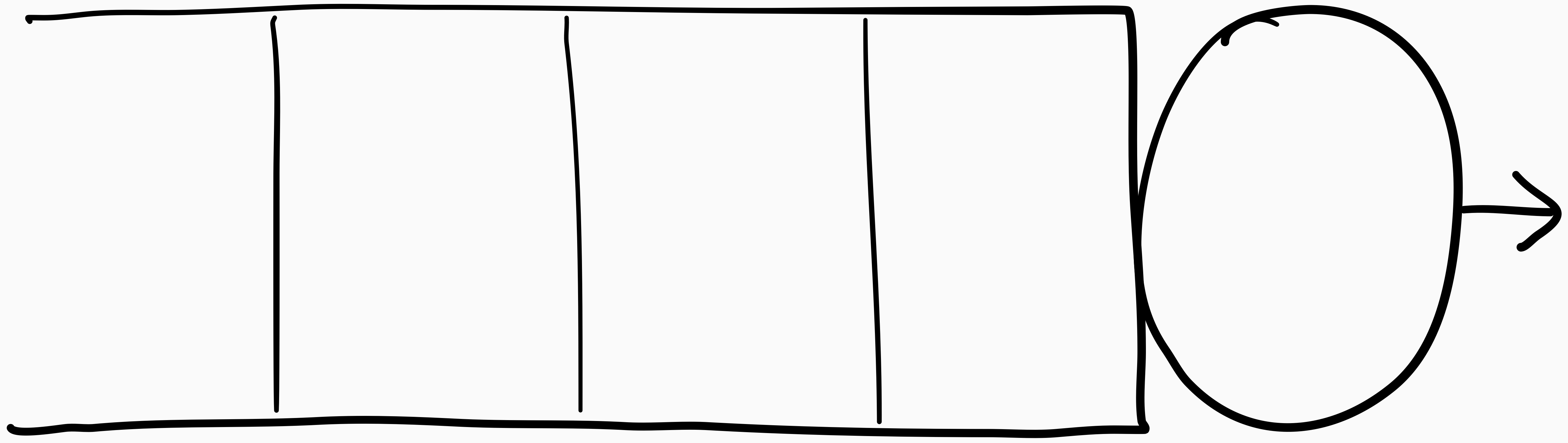
Scheduling policy:
picks which job to serve

M/G/1 queue

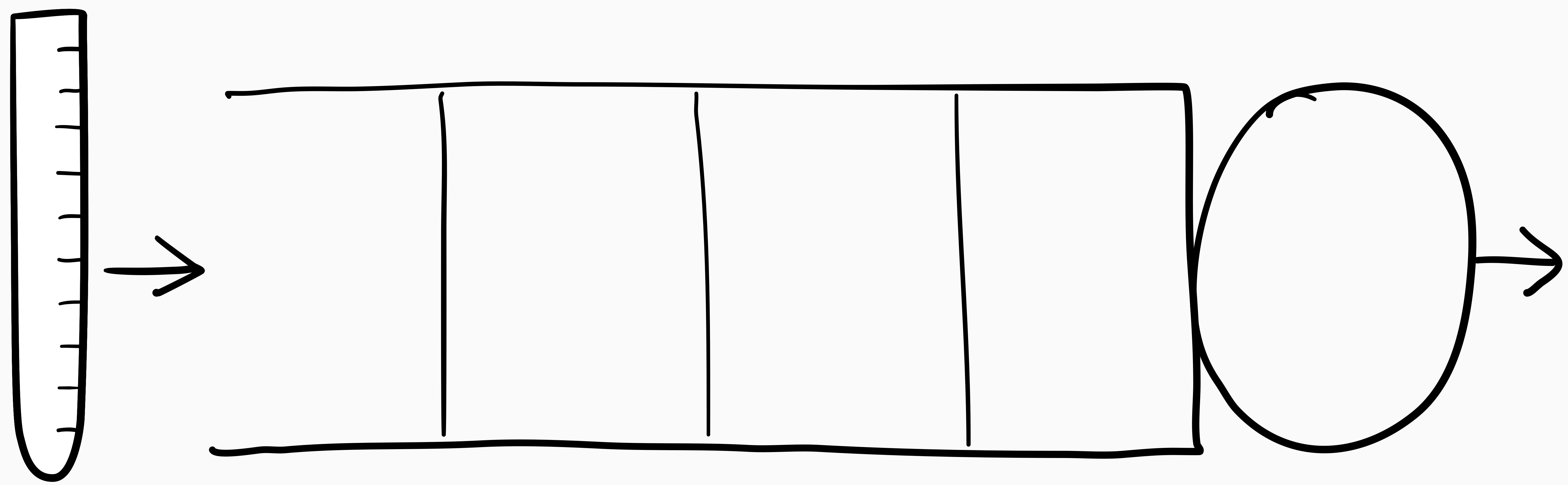


Scheduling policy:
picks which job to serve

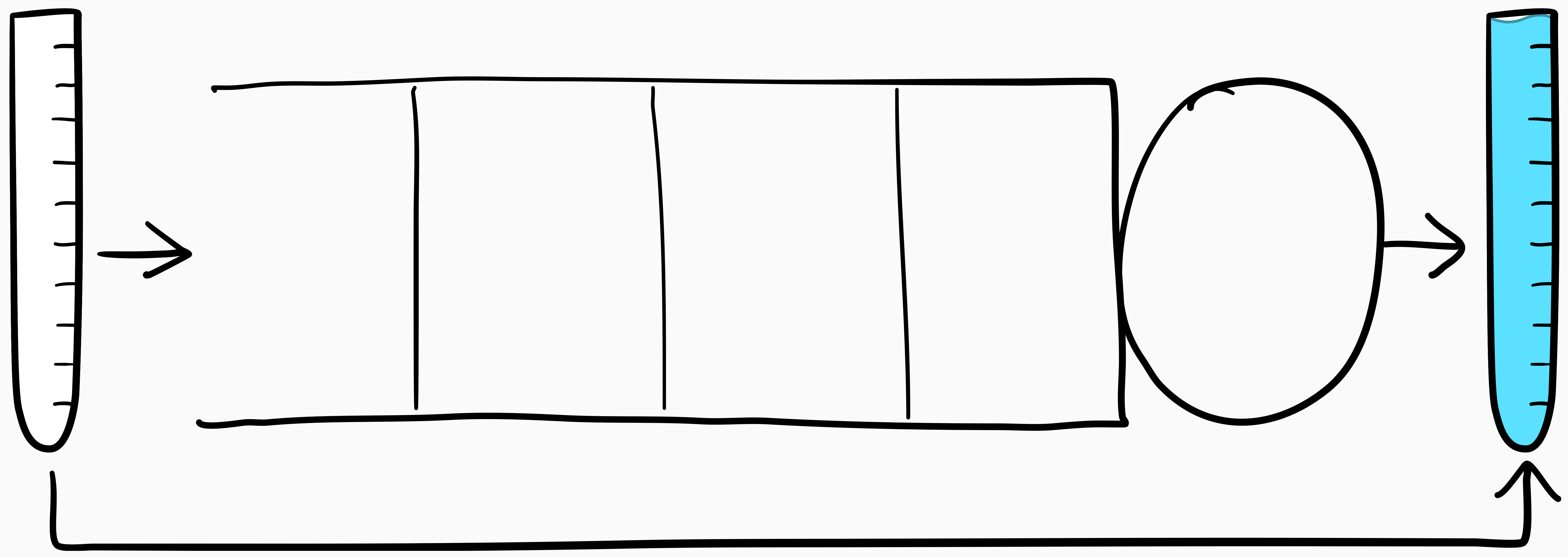
Response time

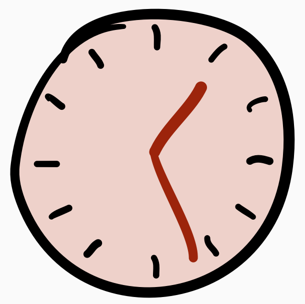


Response time

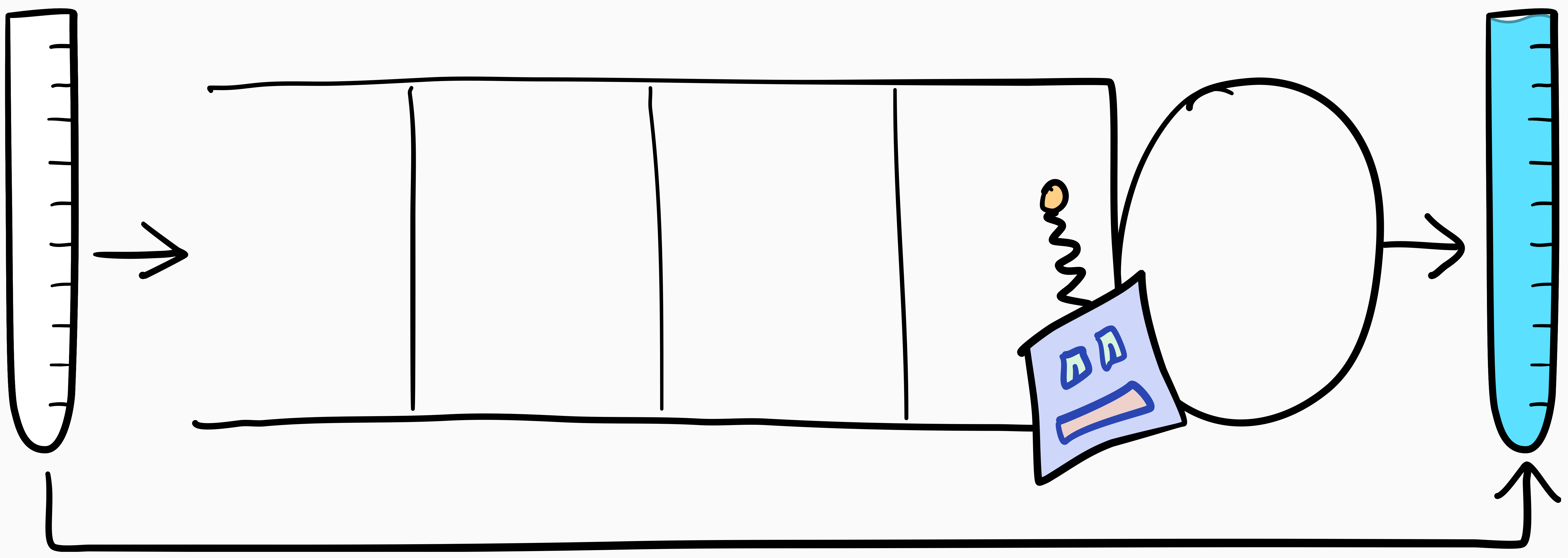


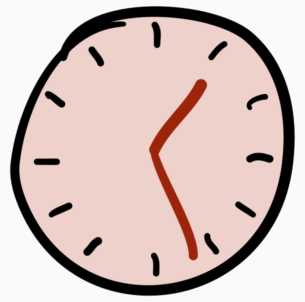
Response time



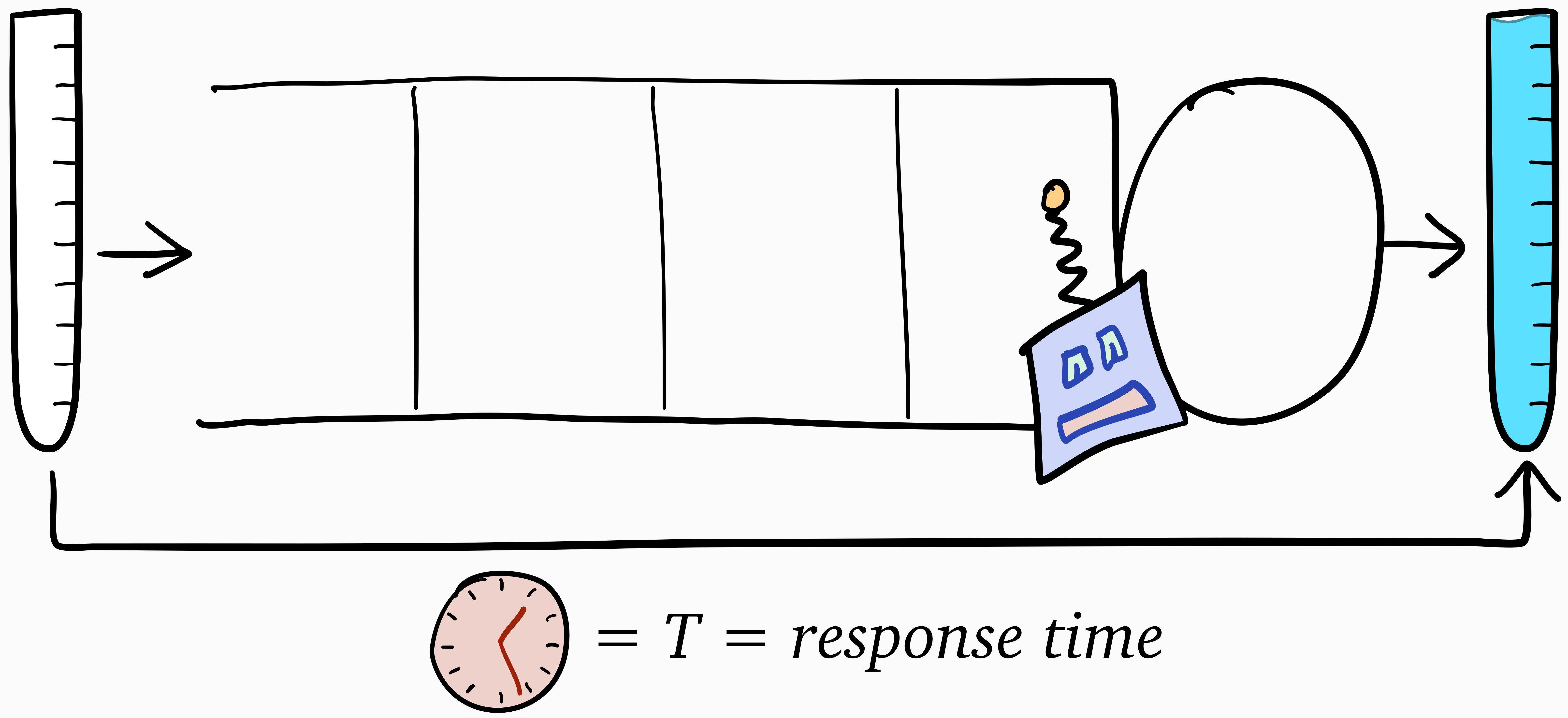
 $= T =$ *response time*

Response time



 = T = *response time*

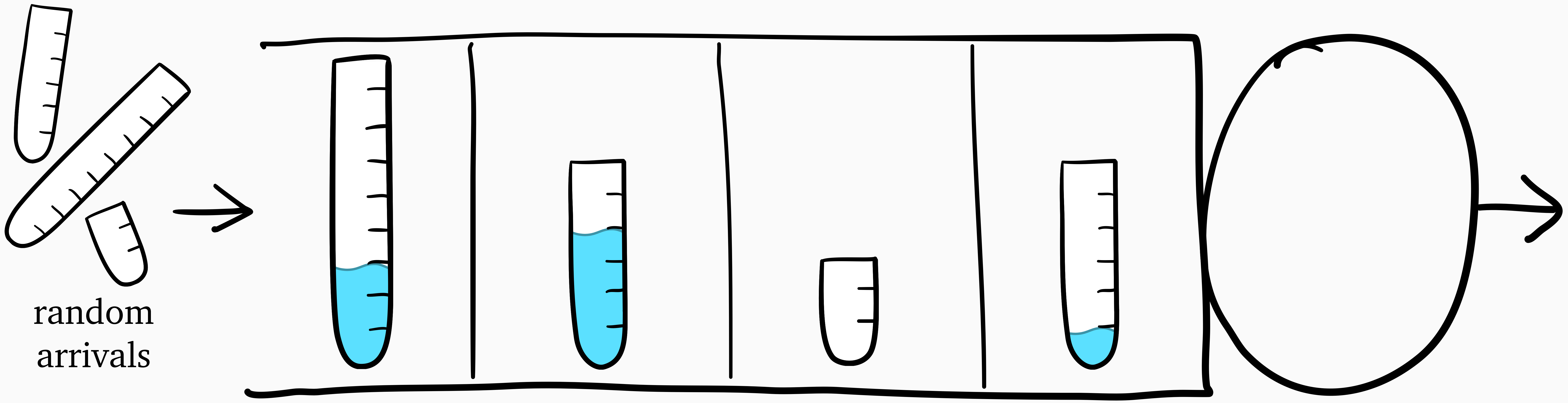
Response time

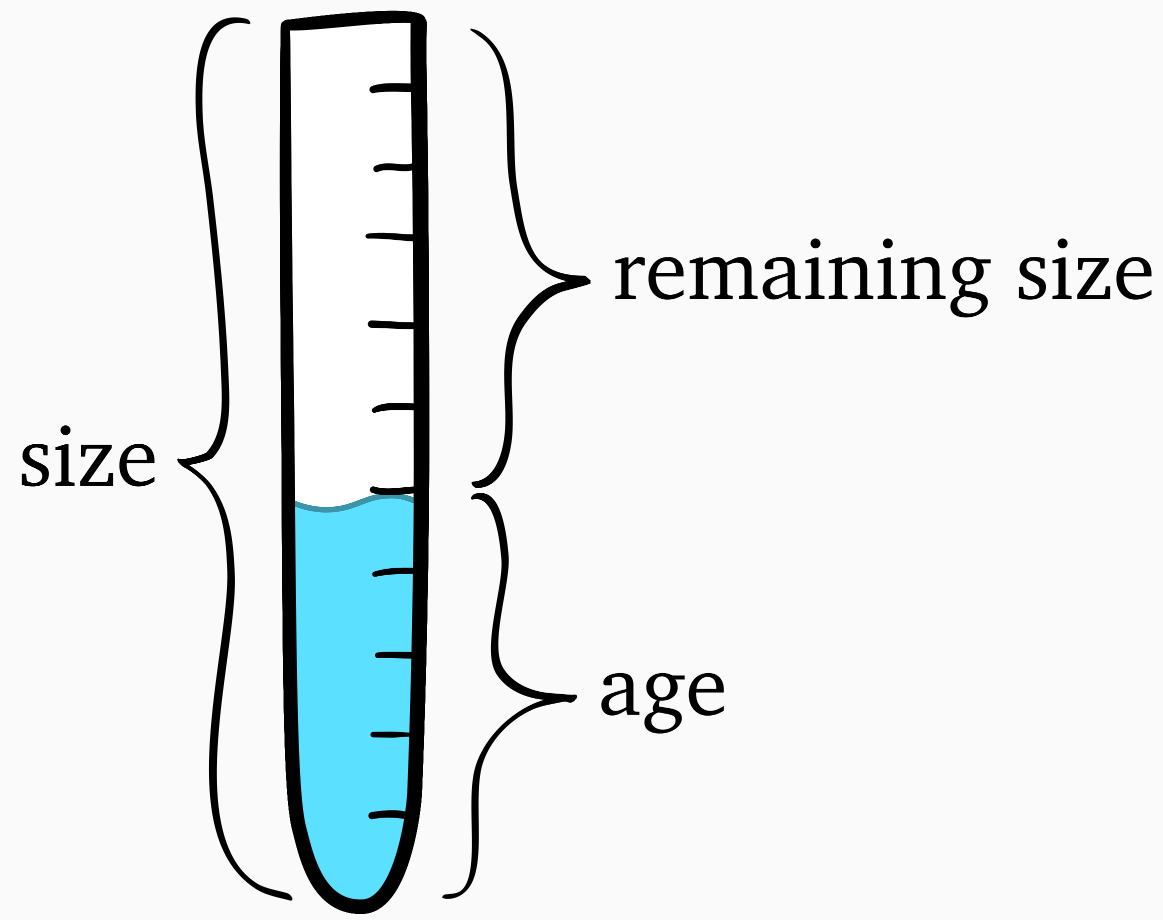


$= T =$ *response time*

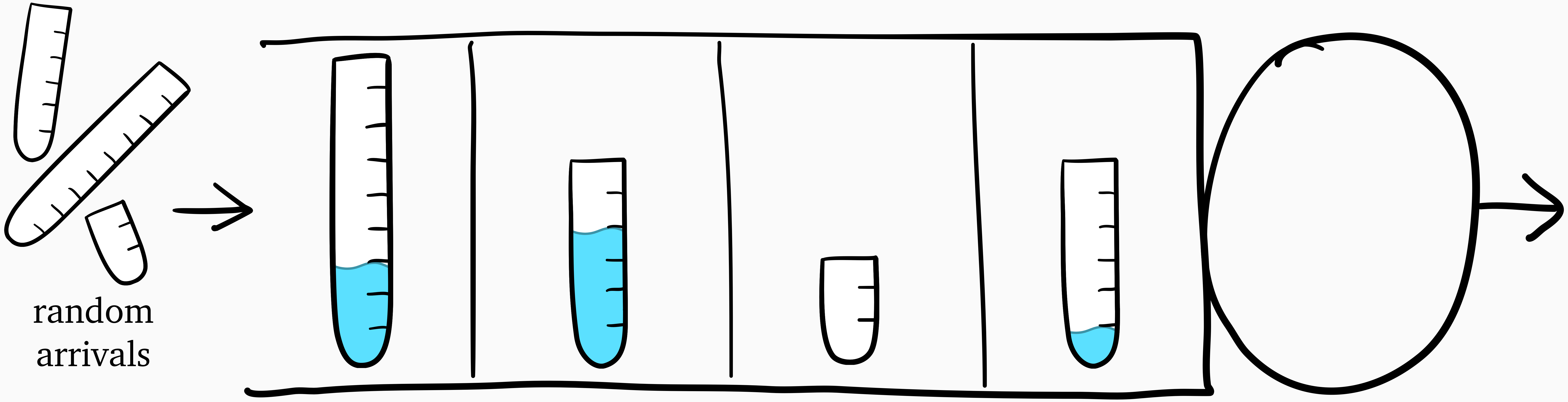
Goal: schedule to minimize *mean response time* $\mathbf{E}[T]$

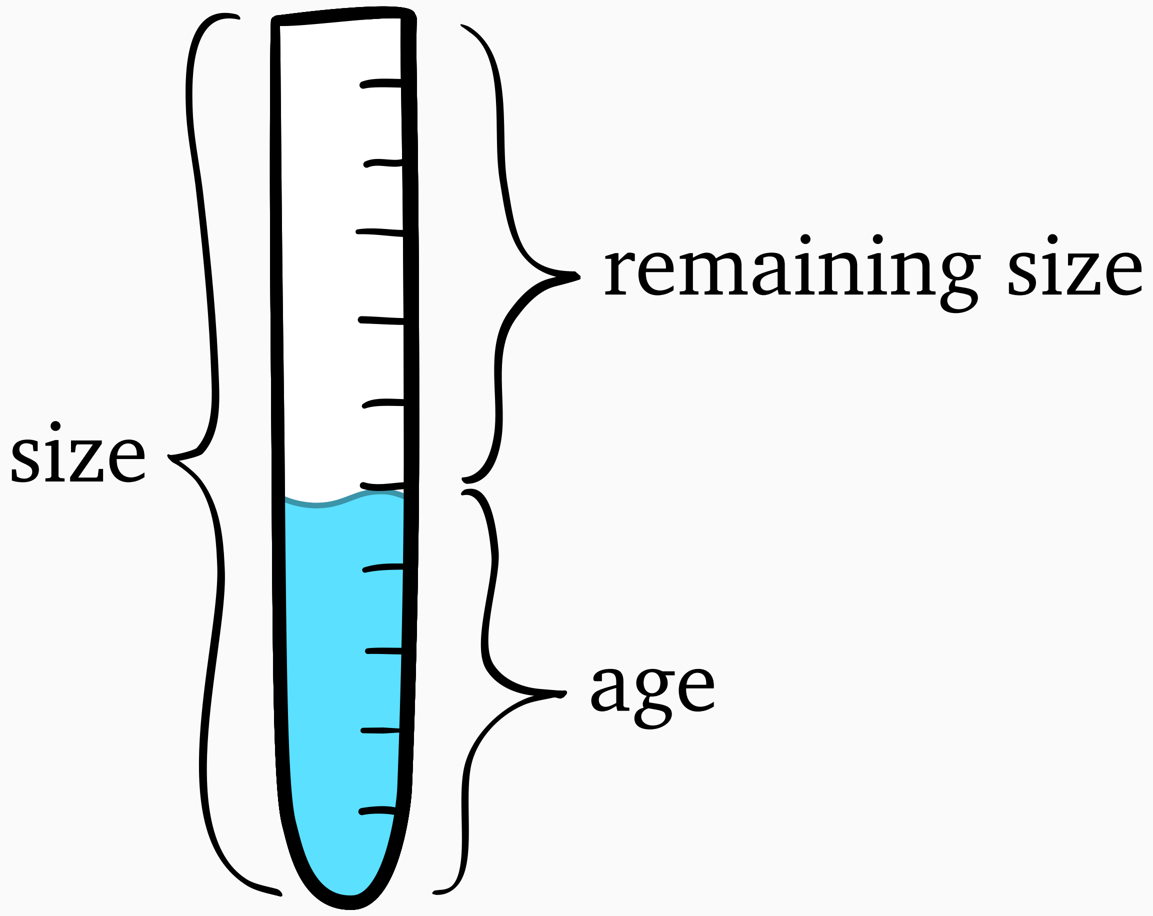
Scheduling with known sizes

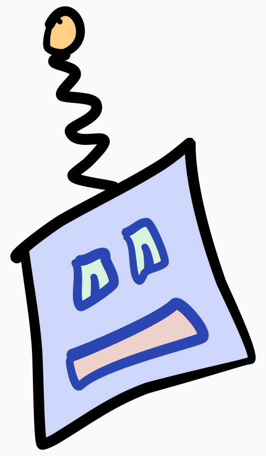




Scheduling with known sizes

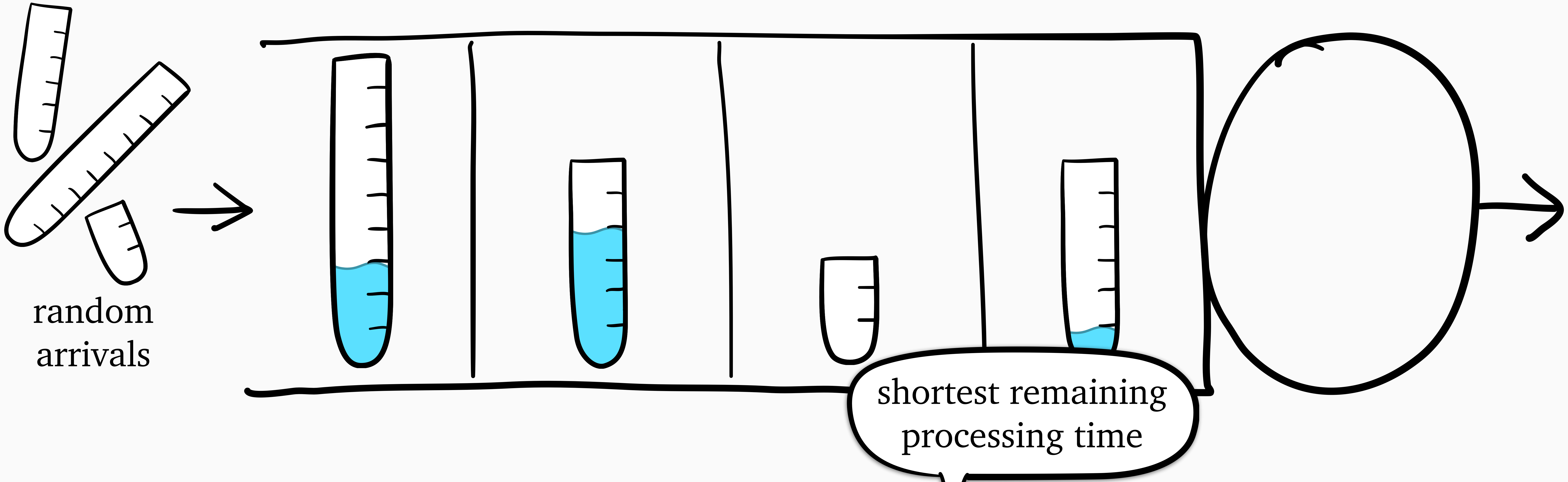


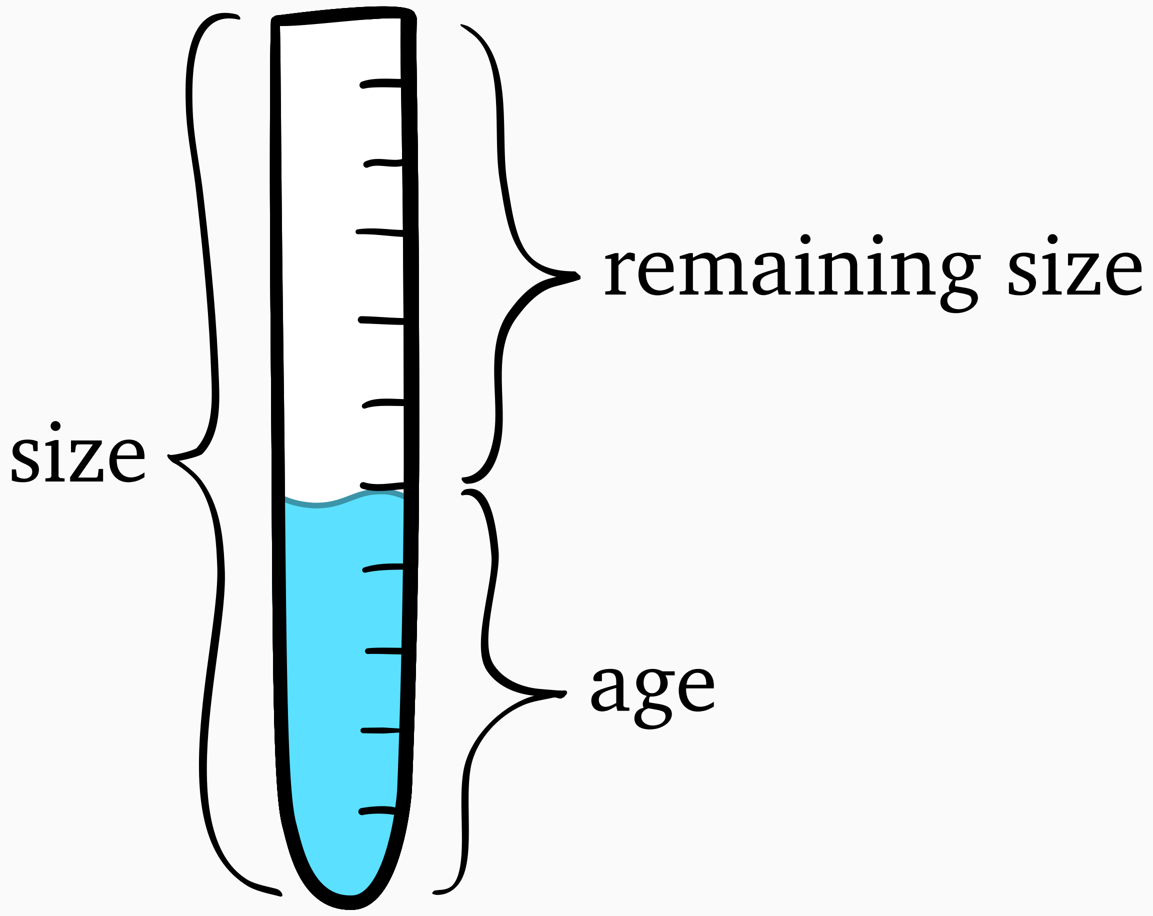


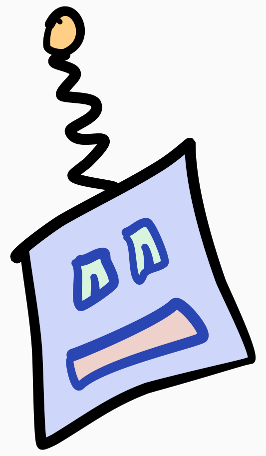


SRPT: always serve job of *least remaining size*

Scheduling with known sizes

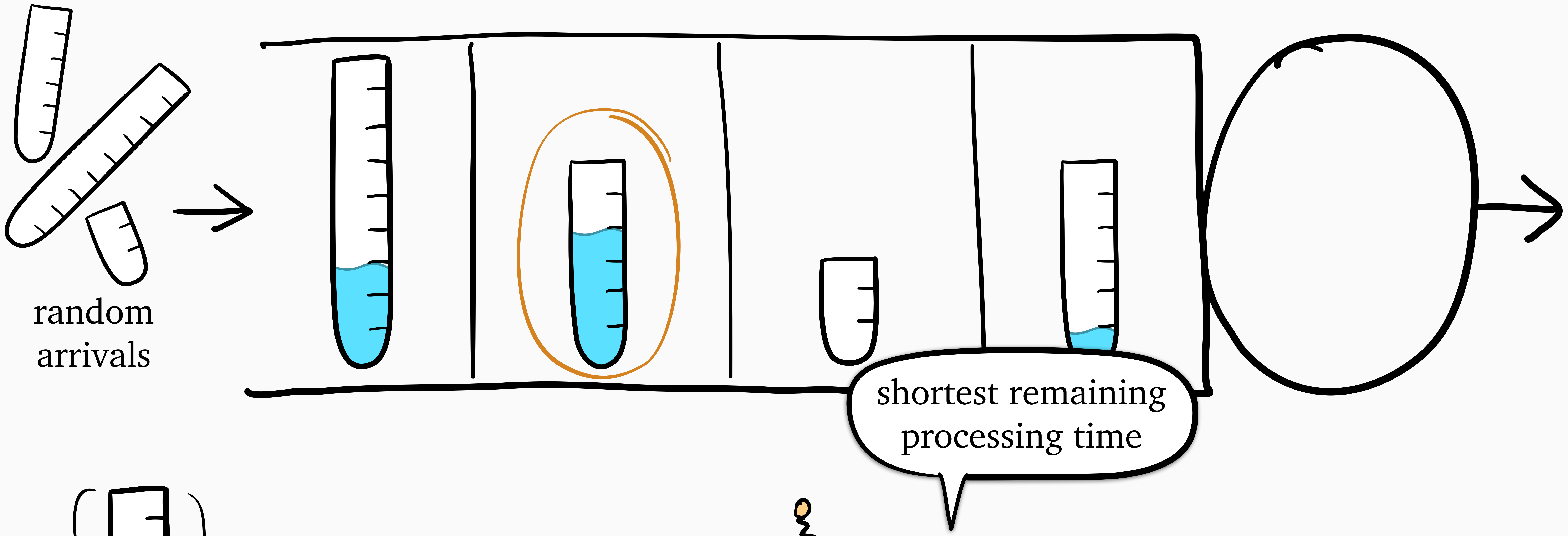


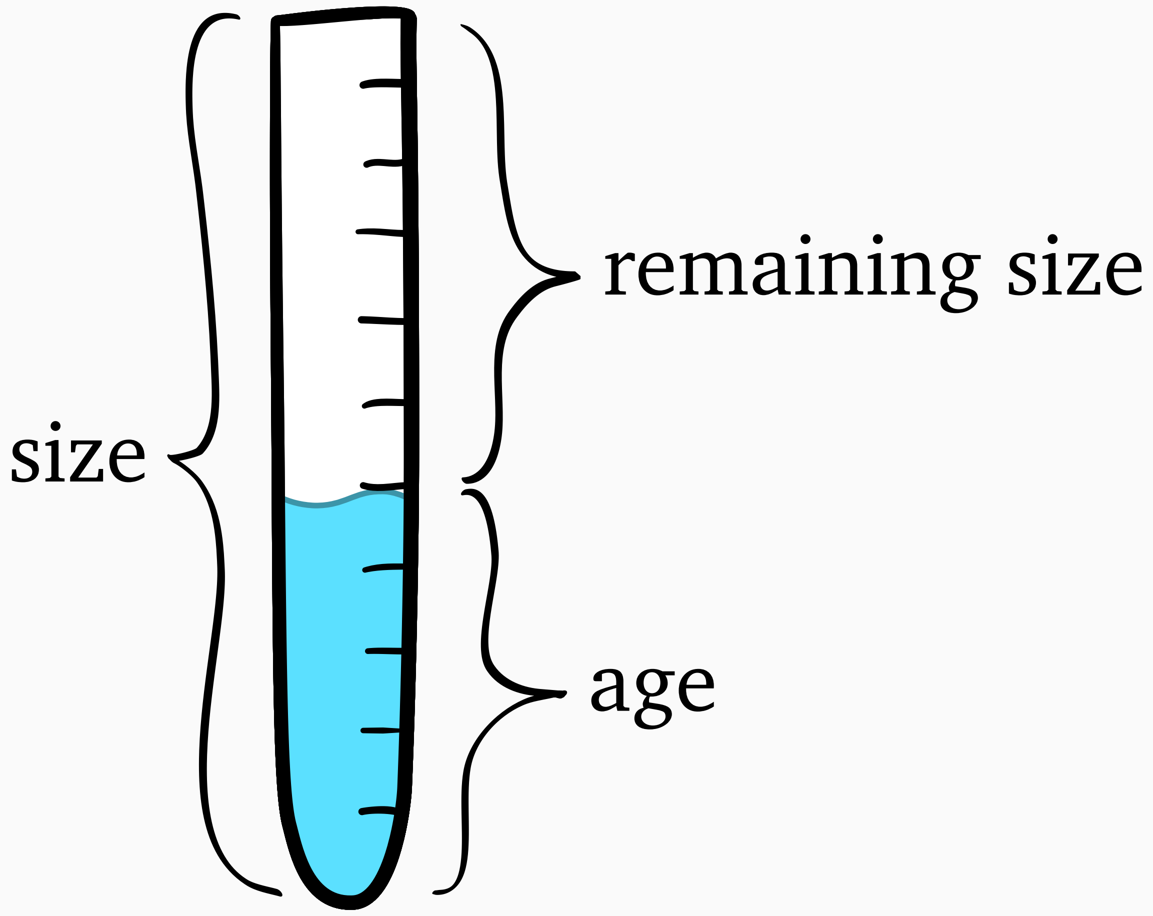




SRPT: always serve job of *least remaining size*

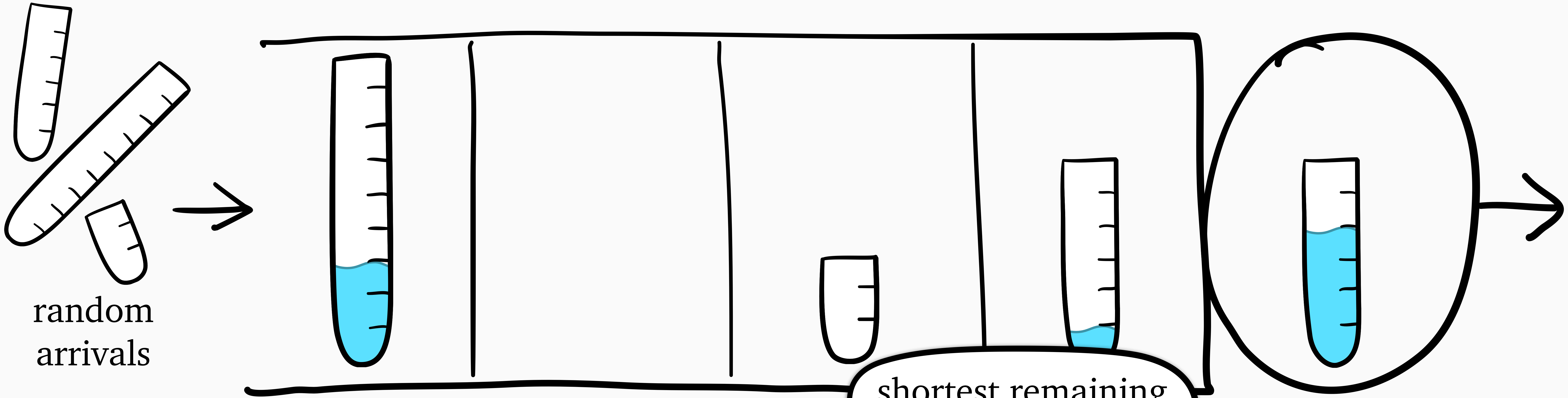
Scheduling with known sizes

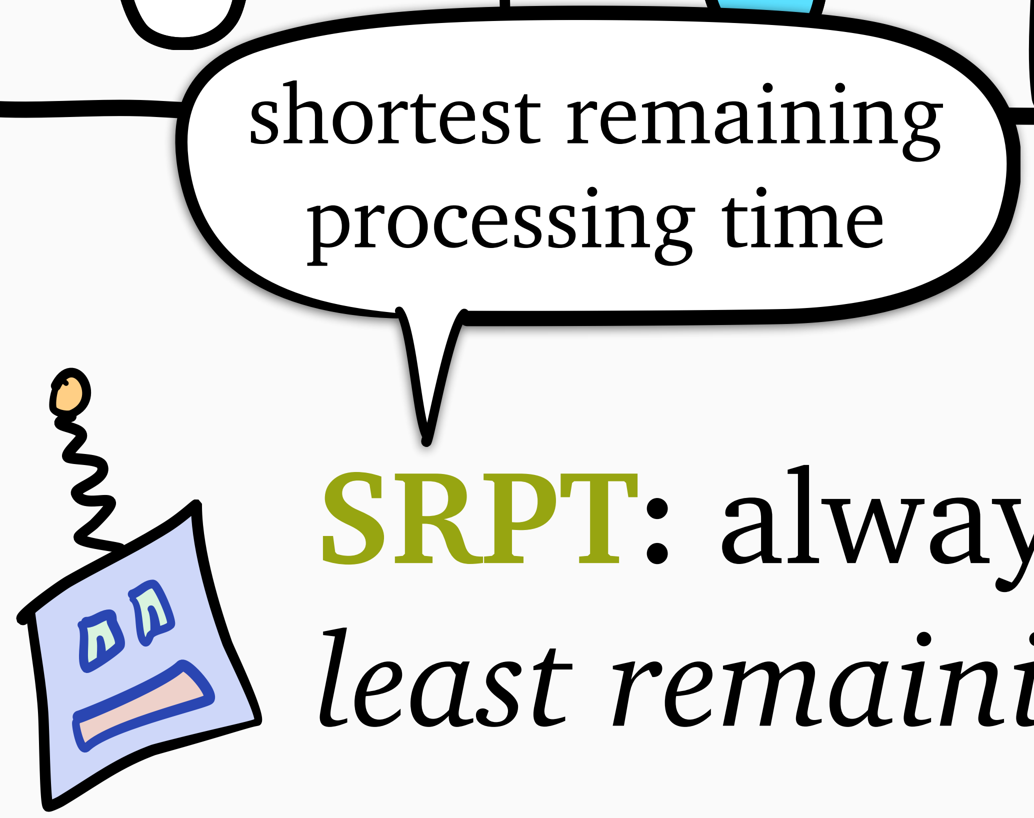




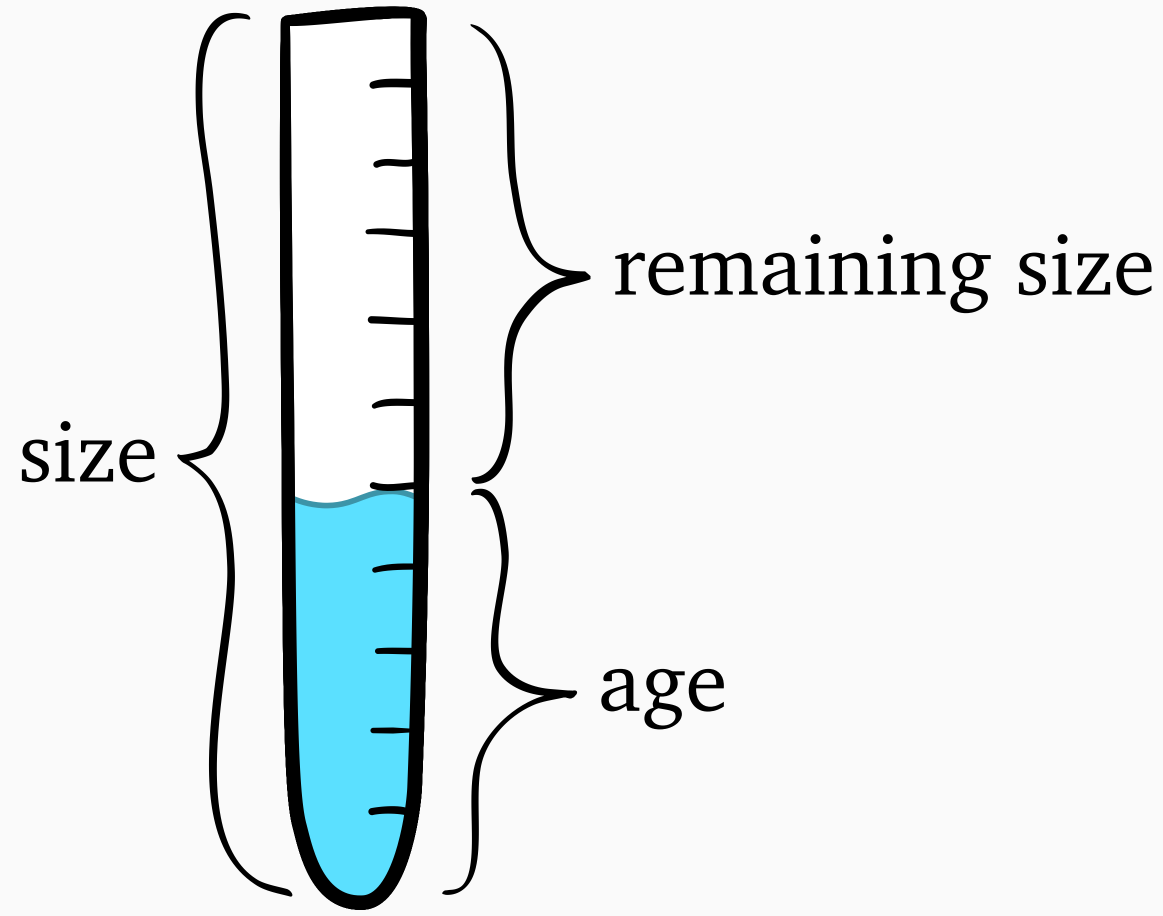
SRPT: always serve job of *least remaining size*

Scheduling with known sizes

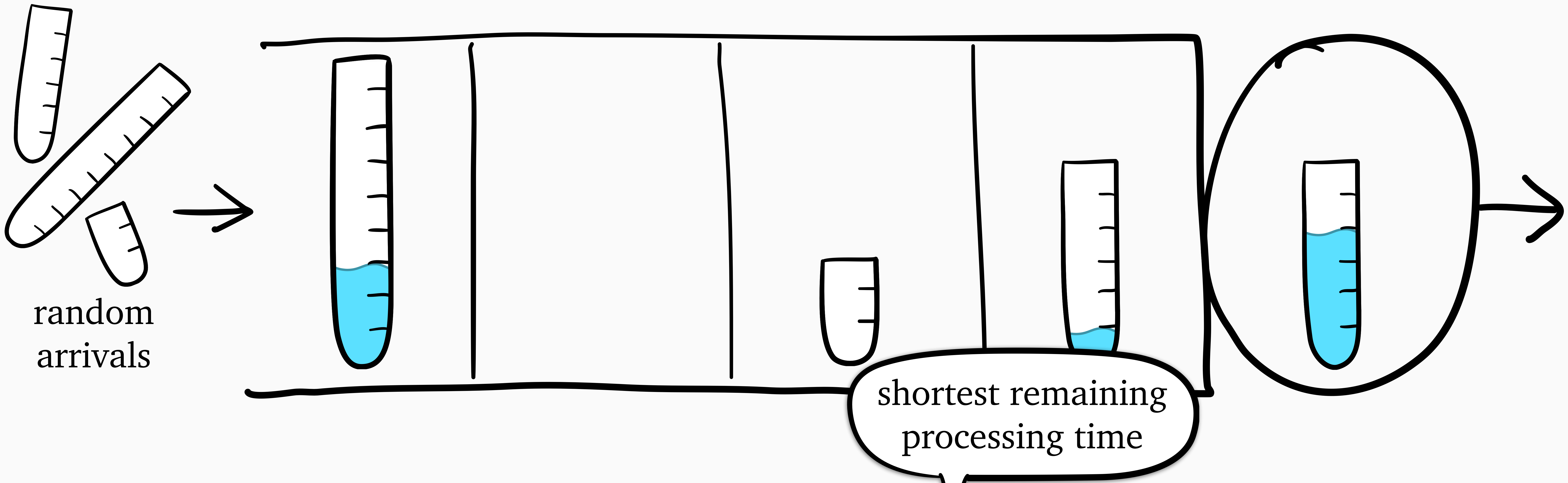


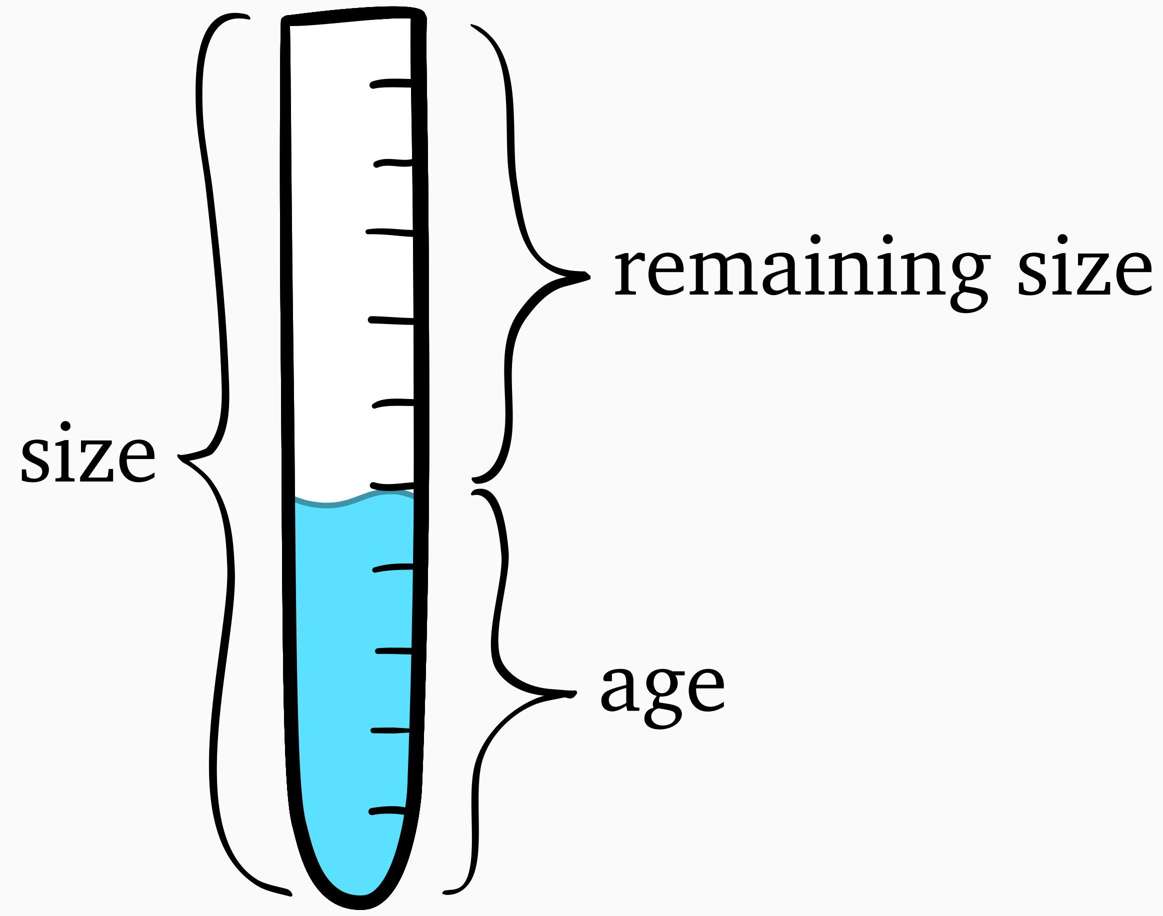


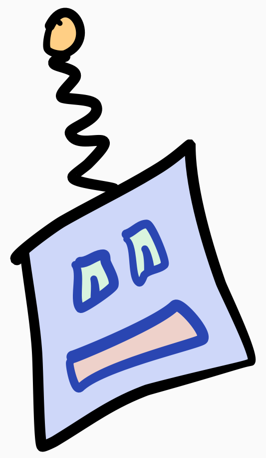
SRPT: always serve job of *least remaining size*



Scheduling with known sizes



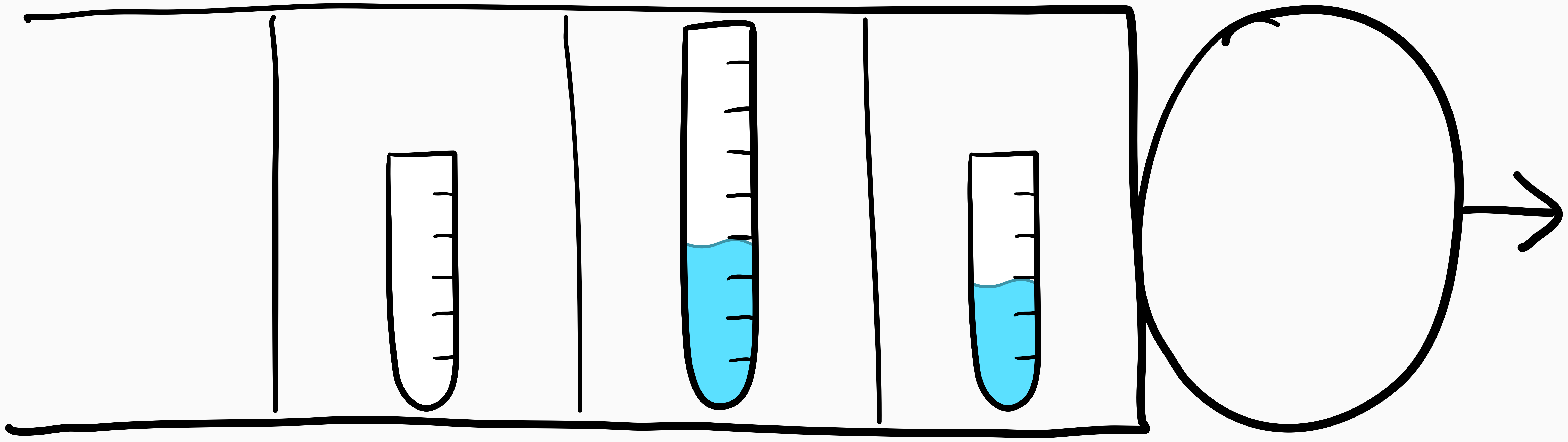




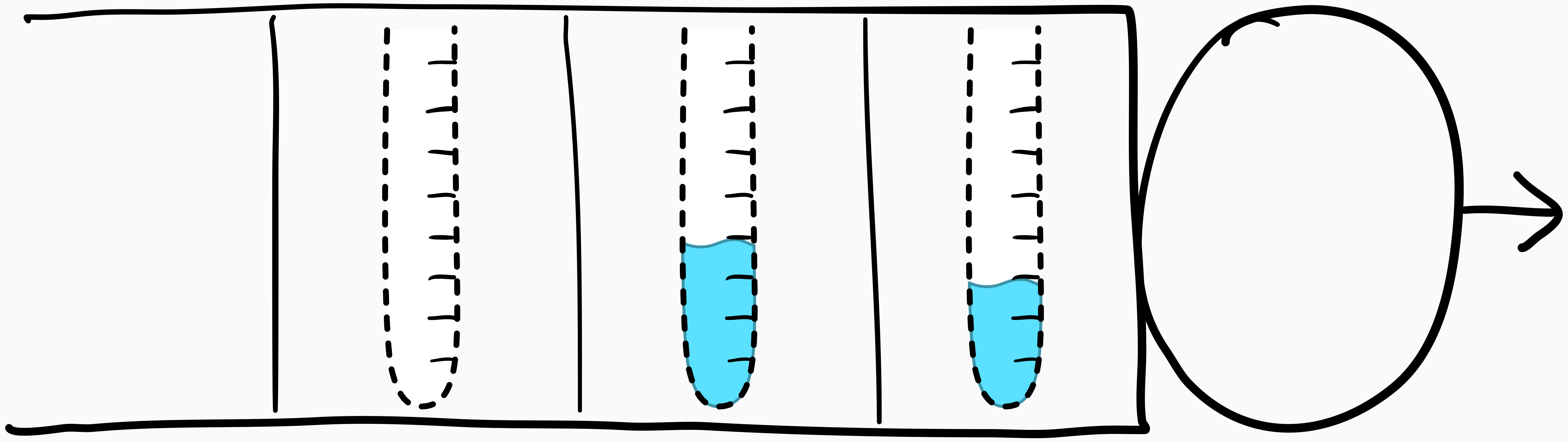
SRPT: always serve job of *least remaining size*

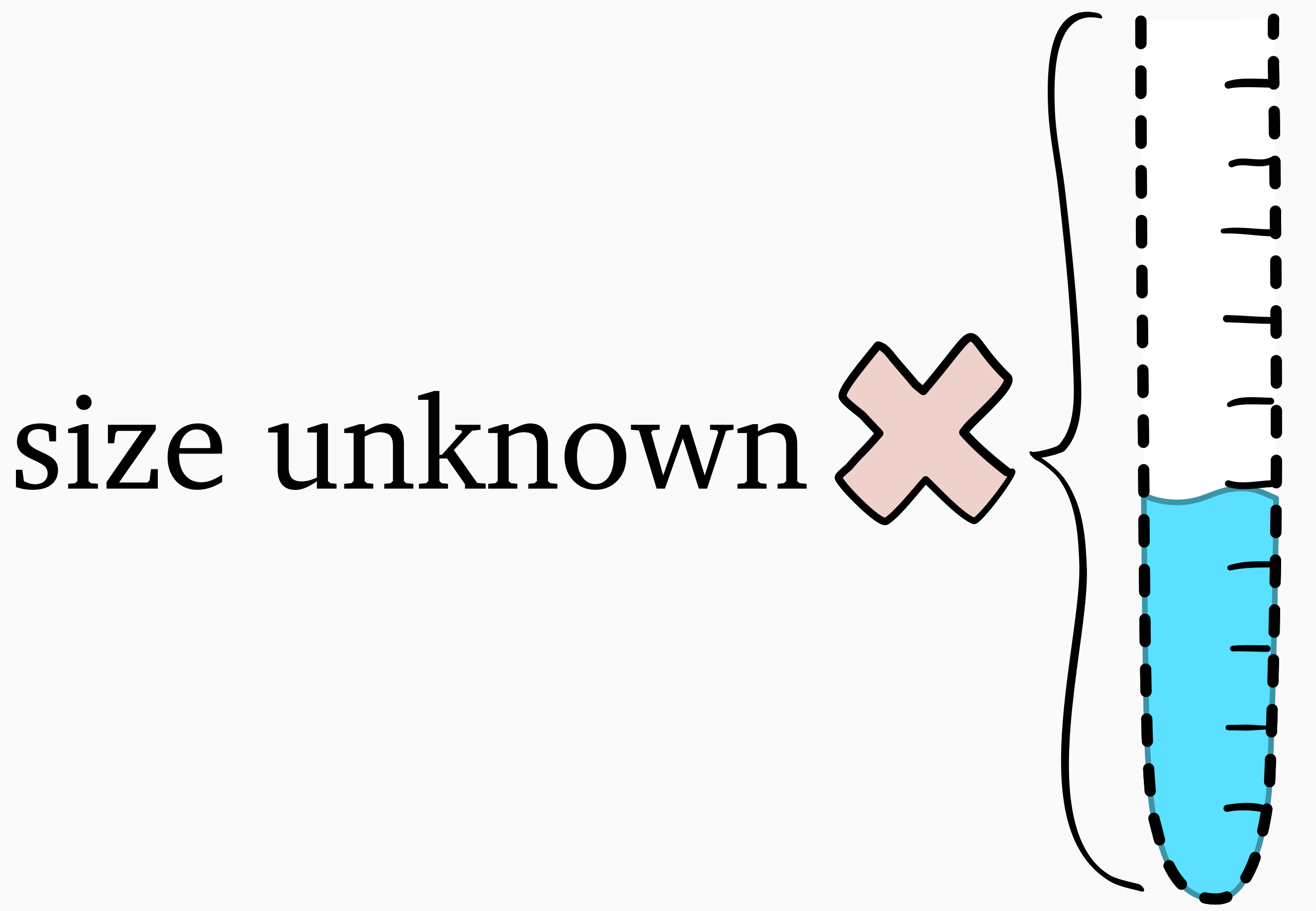
SRPT minimizes $\mathbf{E}[T]$
(Schrage 1968)

Scheduling with *unknown* sizes

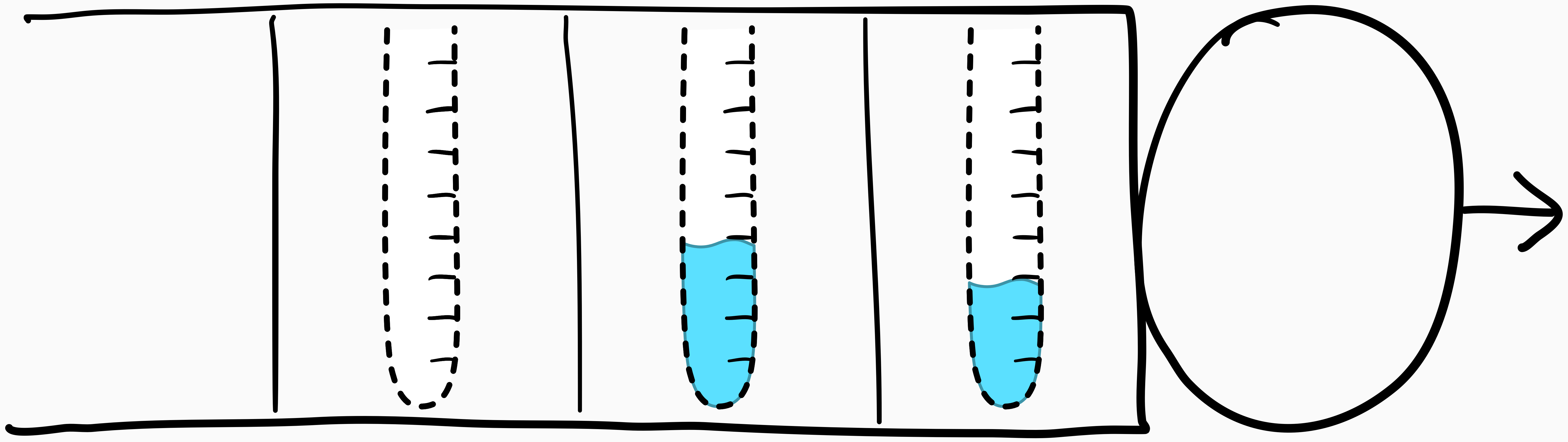


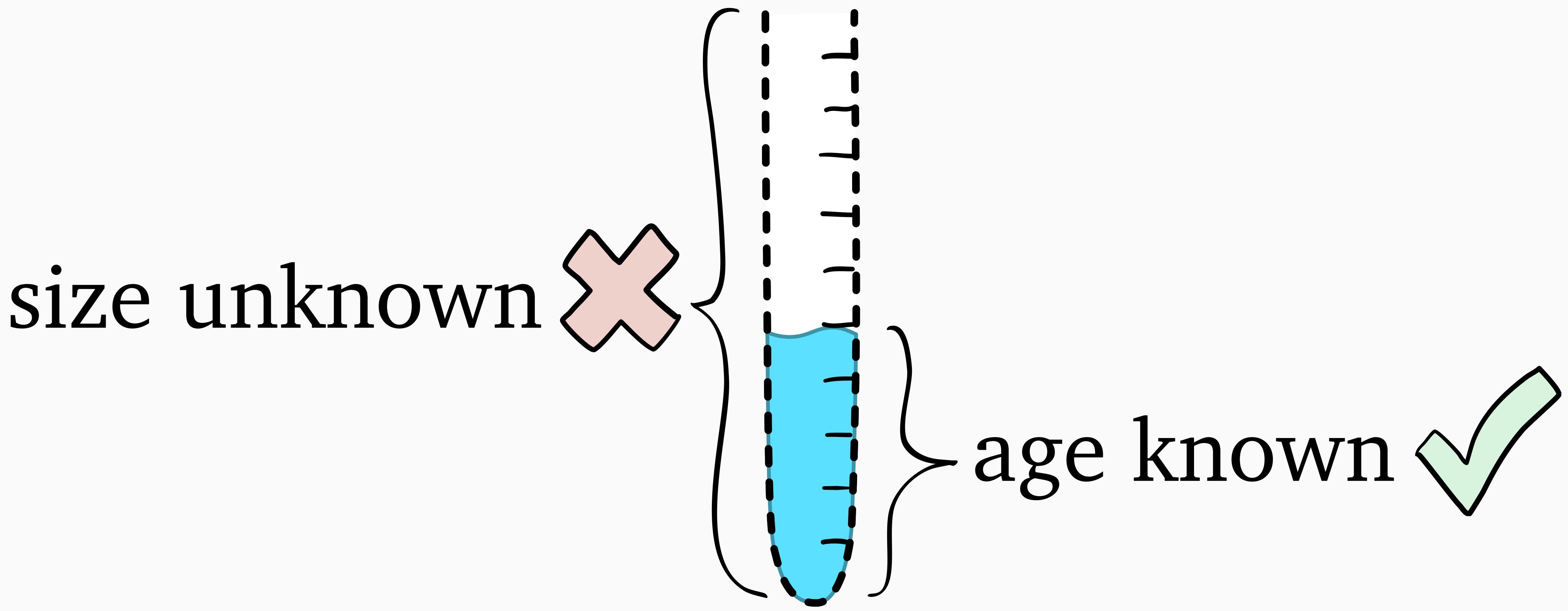
Scheduling with *unknown* sizes



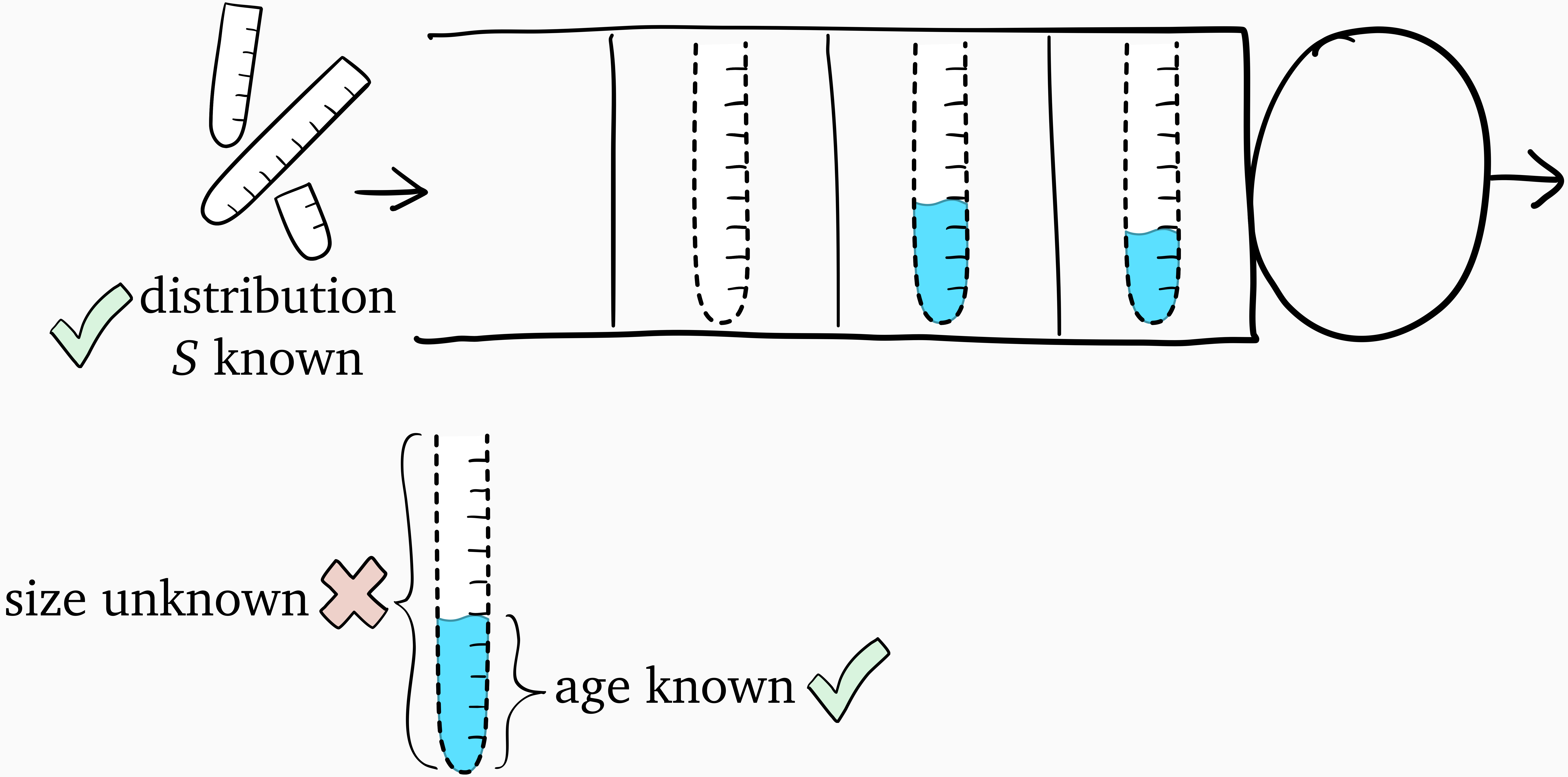


Scheduling with *unknown* sizes

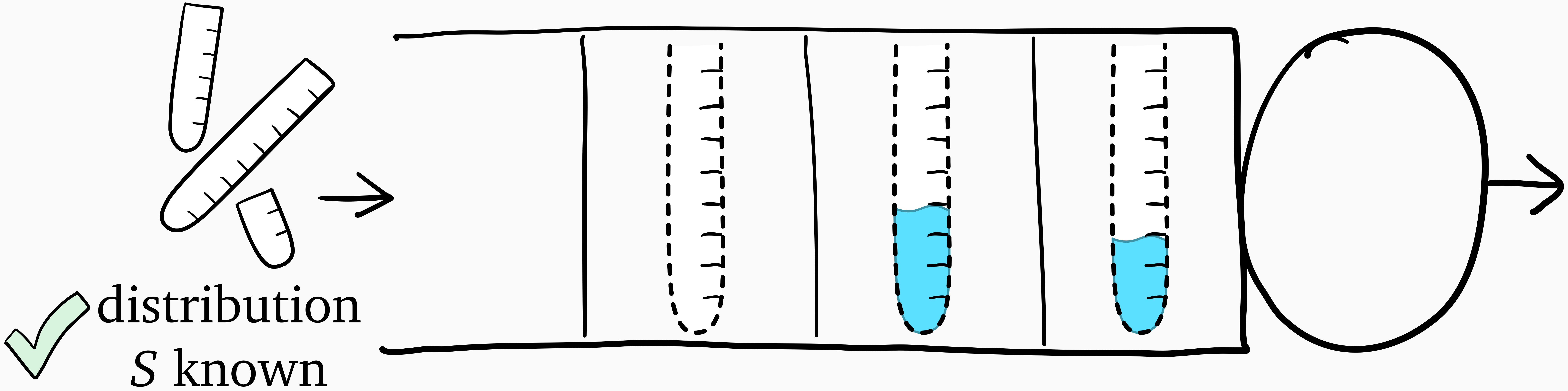


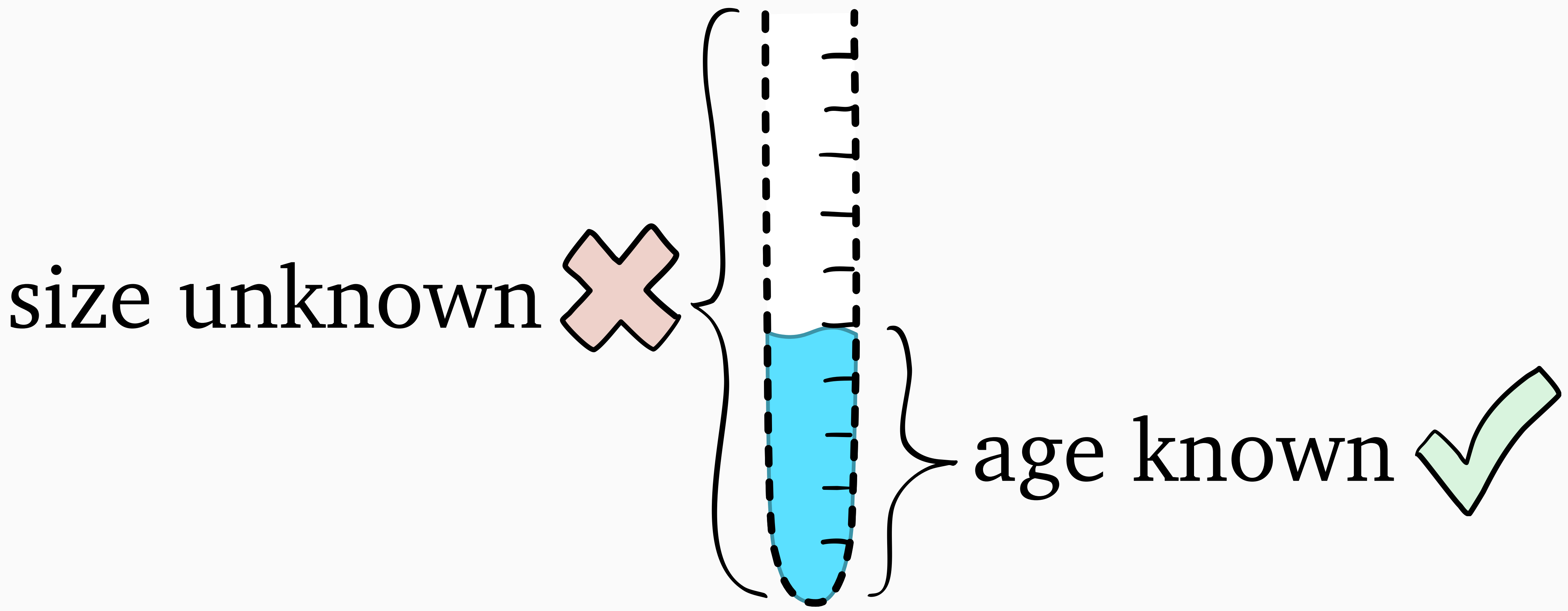


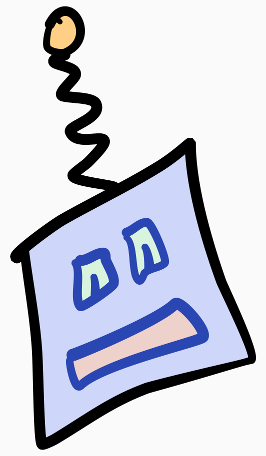
Scheduling with *unknown* sizes



Scheduling with *unknown* sizes

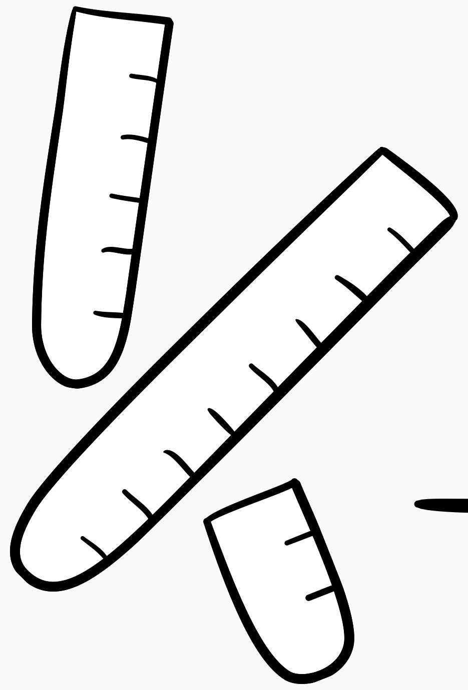


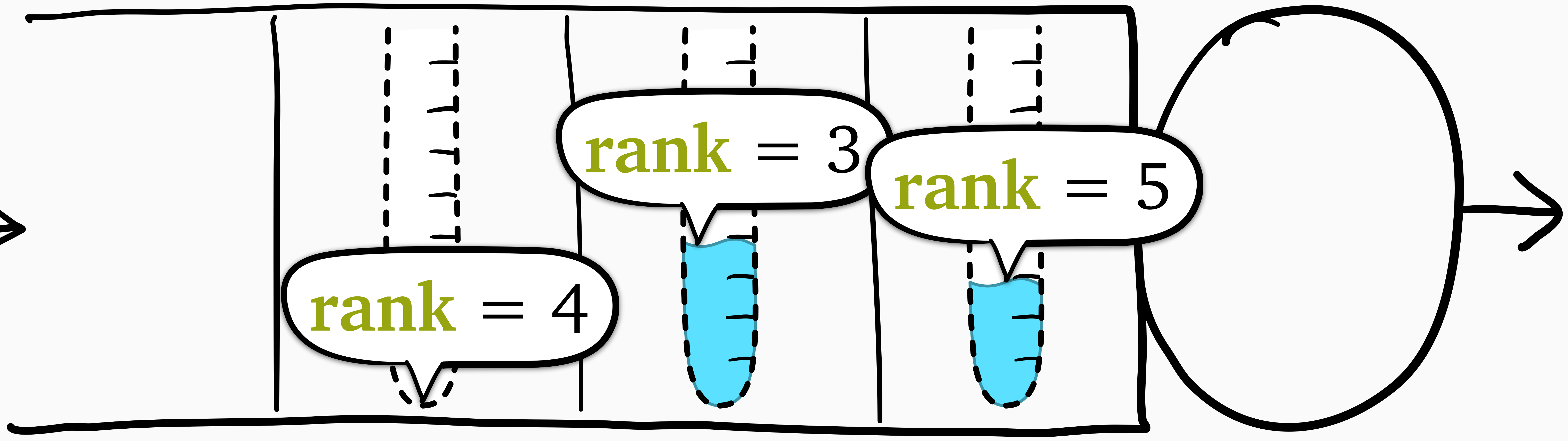




Gittins: assign each job a **rank** based on age and *S*
(lower is better)

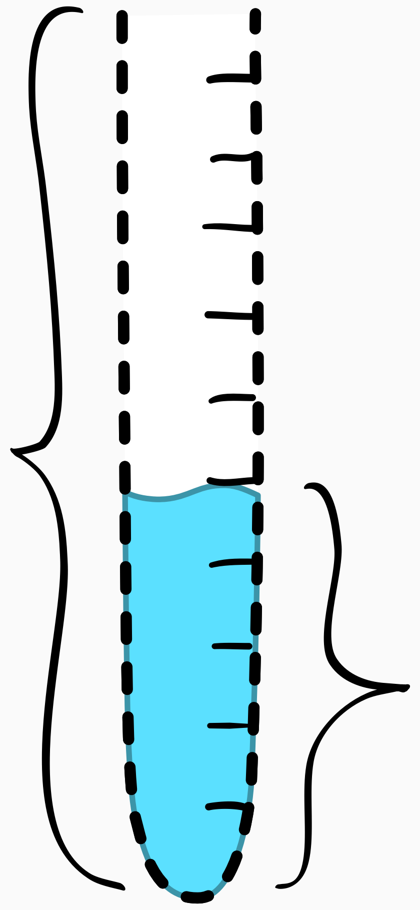
Scheduling with *unknown* sizes



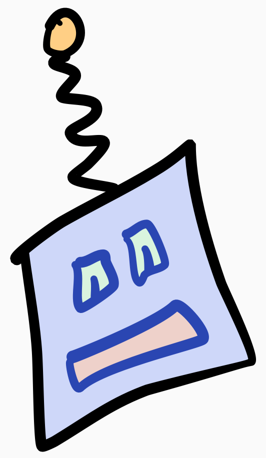


distribution
S known

size unknown

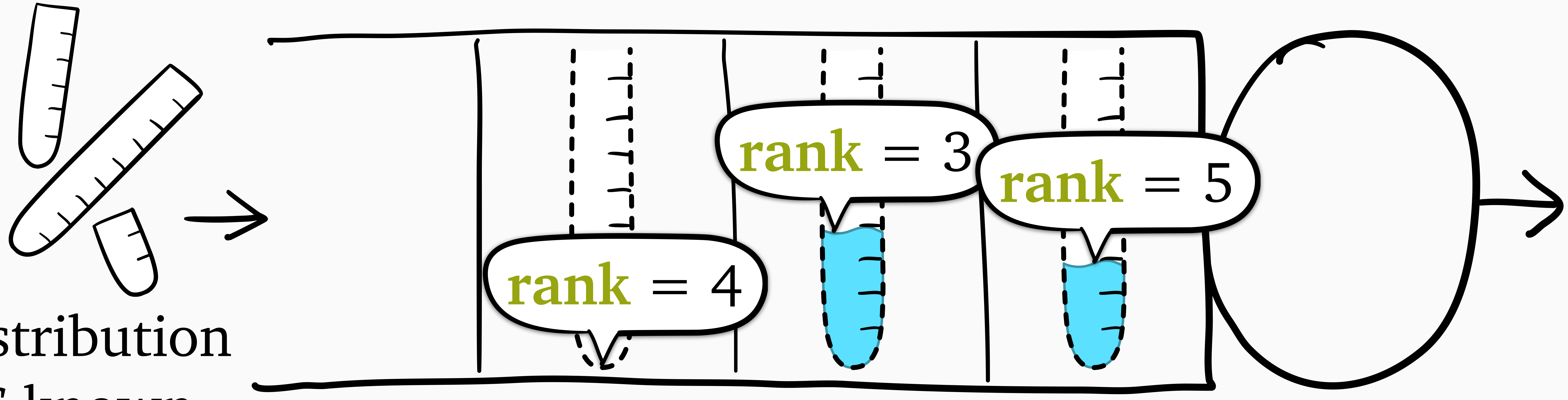


age known

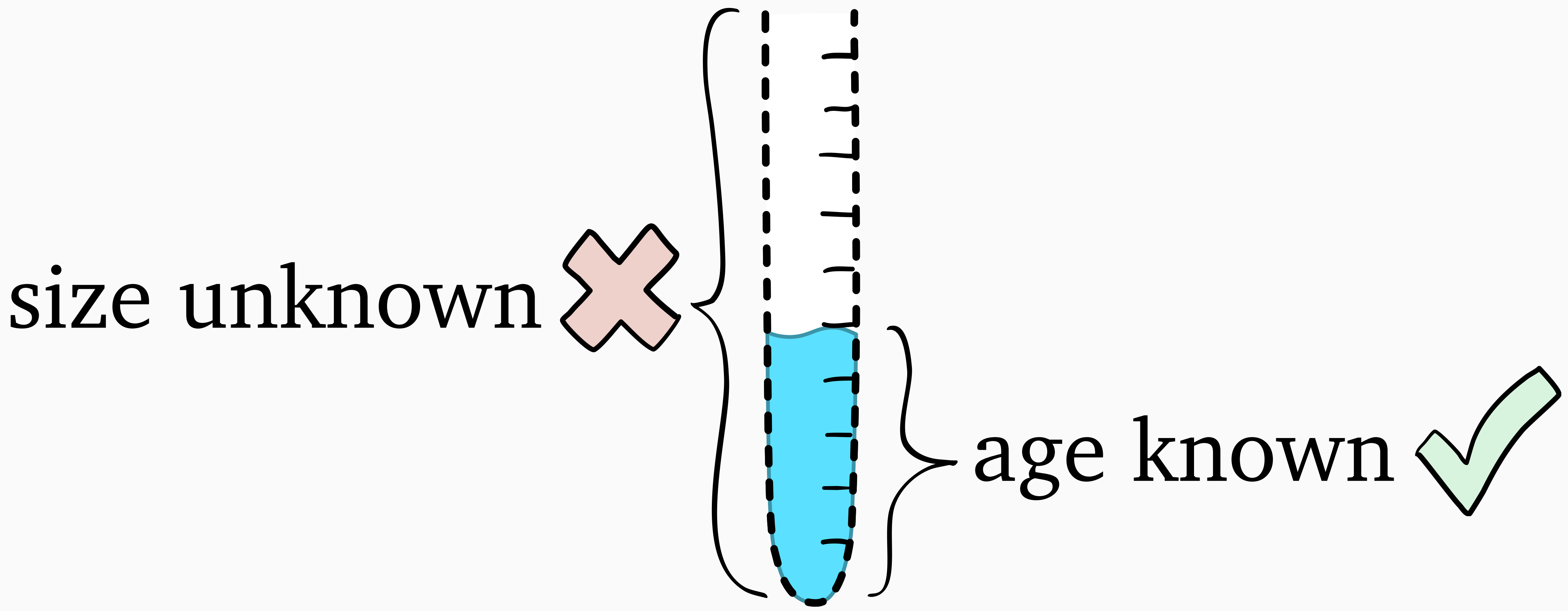


Gittins: assign each job a **rank** based on age and S
(lower is better)

Scheduling with *unknown* sizes



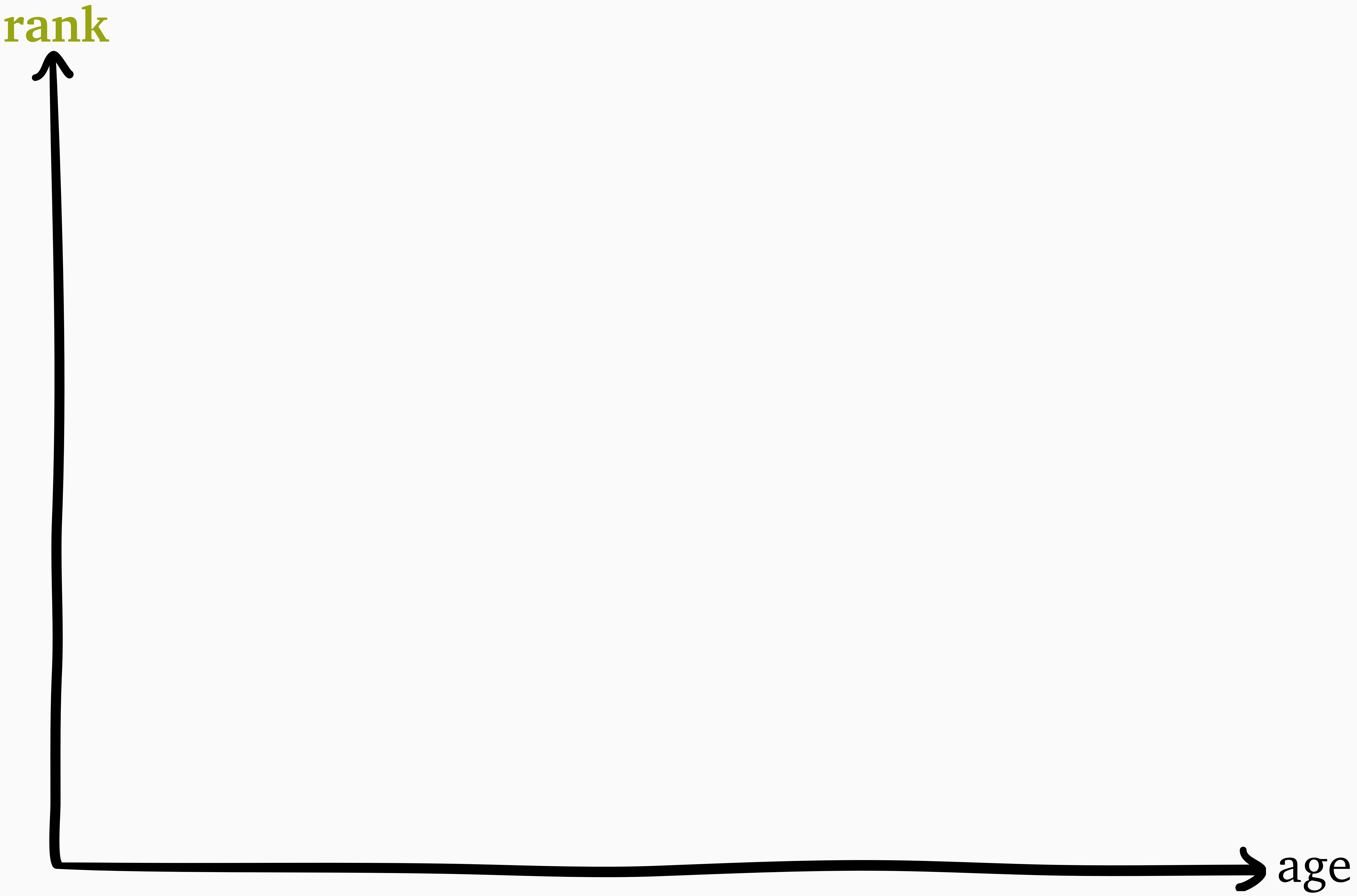
✓ distribution S known



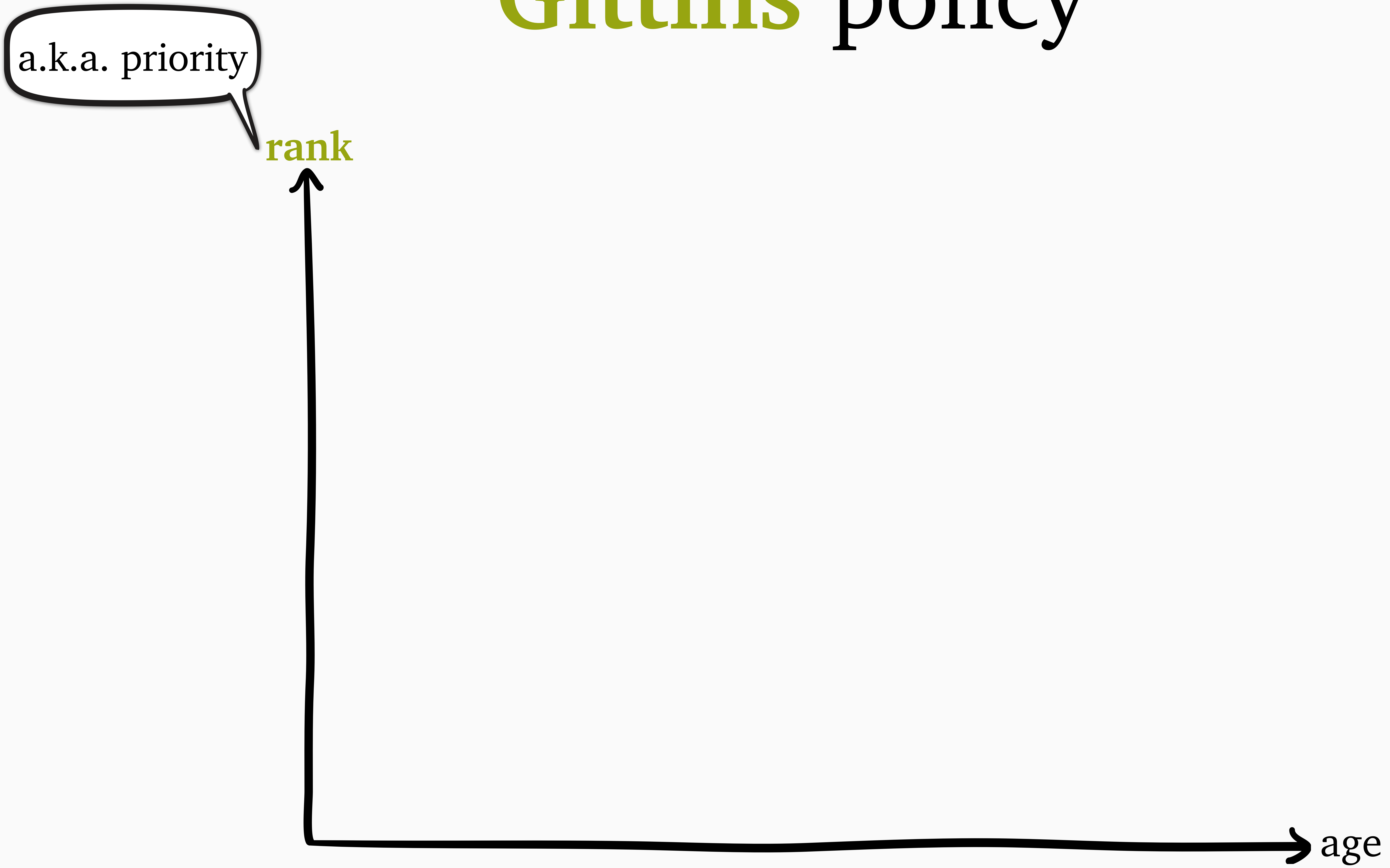
Gittins: assign each job a **rank** based on age and S
(lower is better)

Gittins minimizes $\mathbf{E}[T]$
(Gittins 1989)

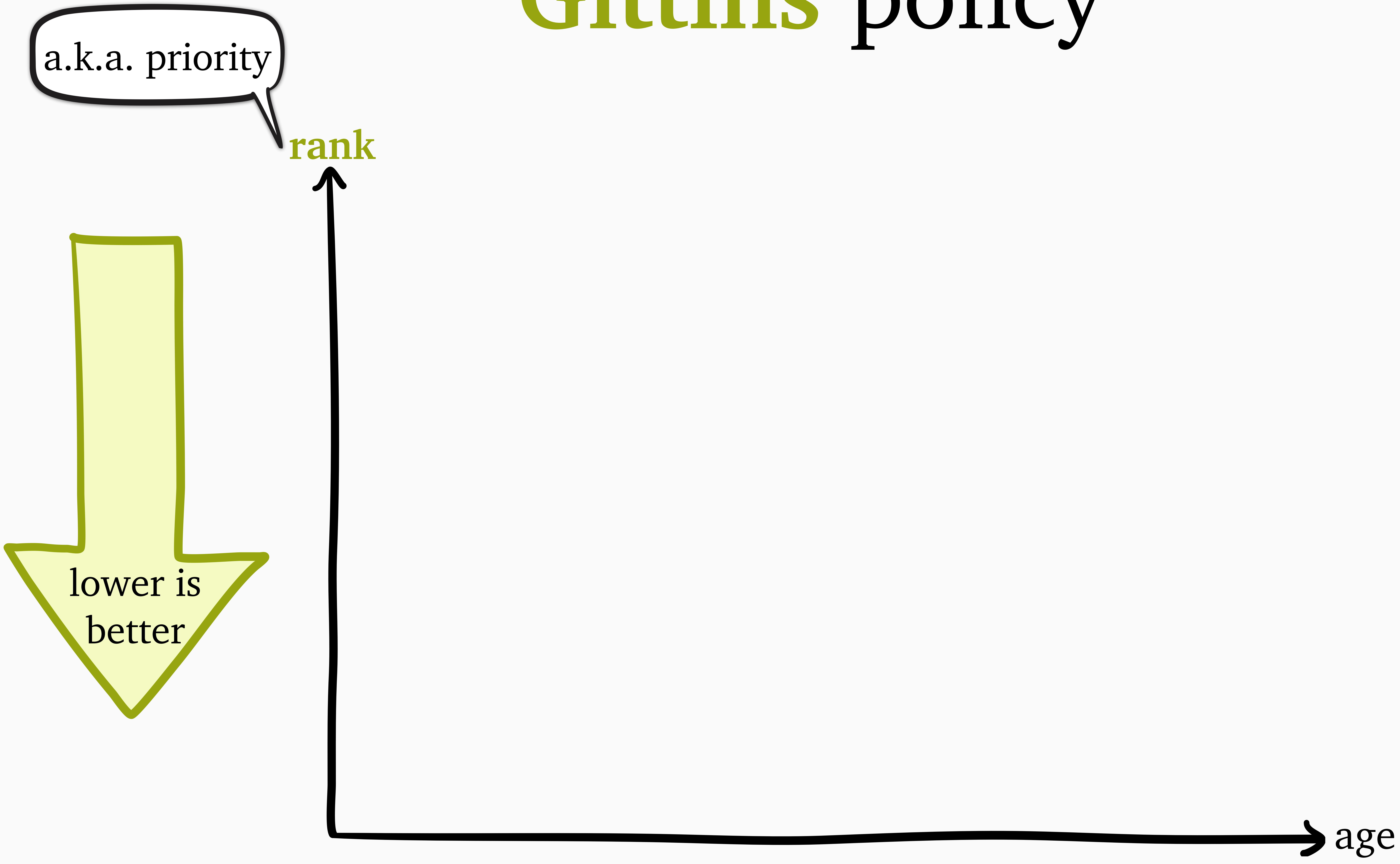
Gittins policy



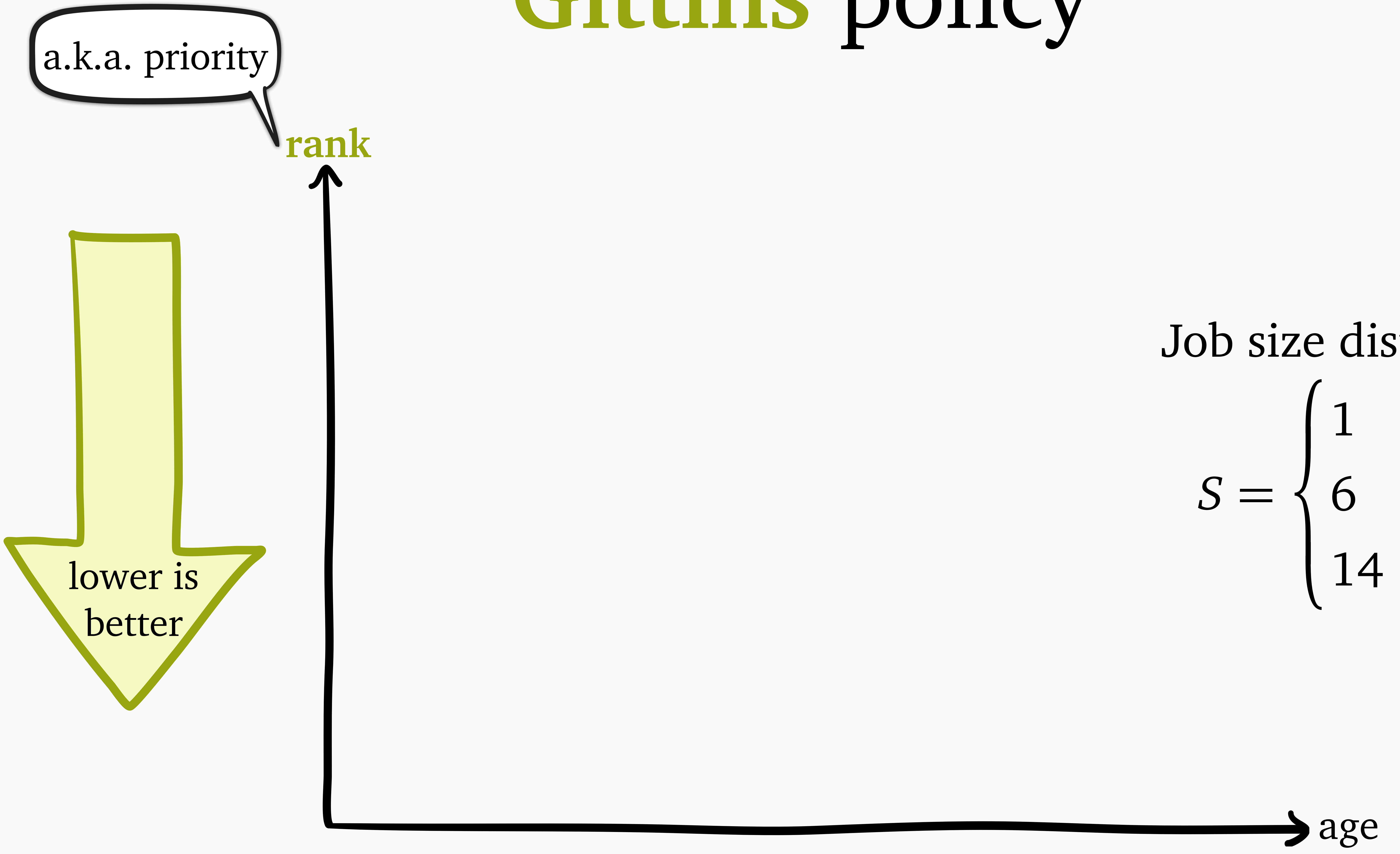
Gittins policy



Gittins policy



Gittins policy

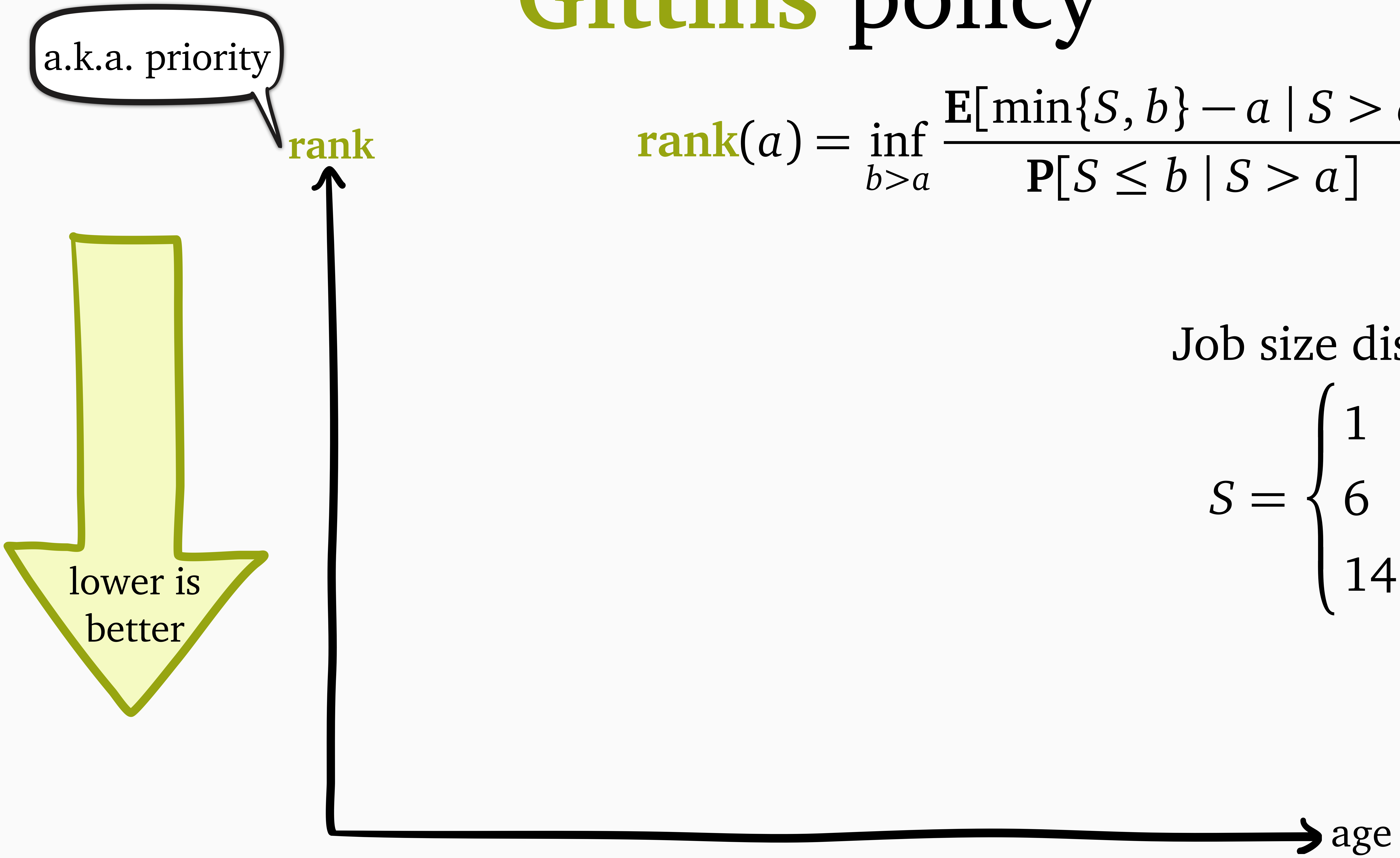


Job size distribution:

$$S = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

Gittins policy

$$\text{rank}(a) = \inf_{b>a} \frac{\mathbf{E}[\min\{S, b\} - a \mid S > a]}{\mathbf{P}[S \le b \mid S > a]}$$

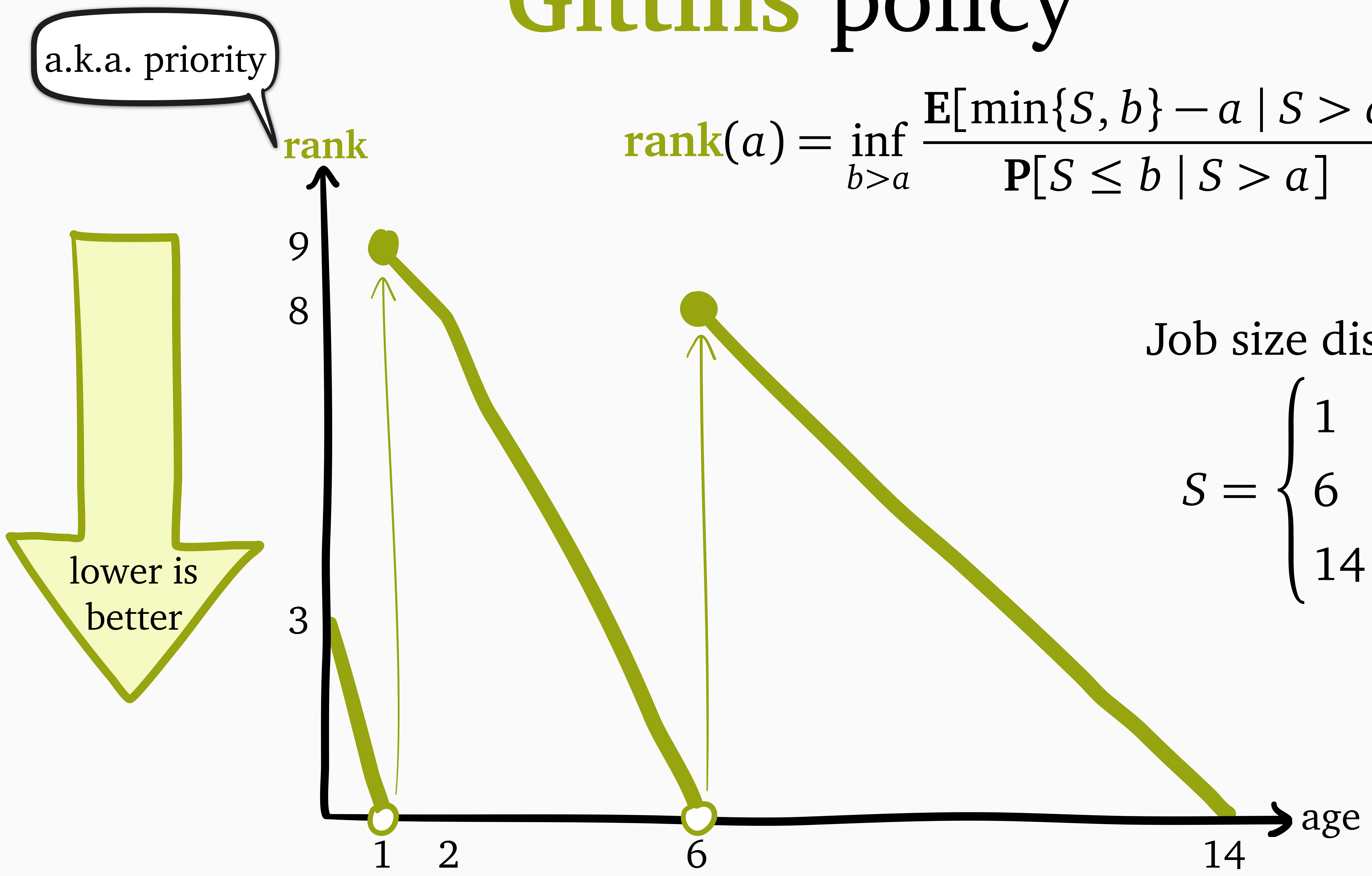


Job size distribution:

$$S = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

Gittins policy

$$\textbf{rank}(a) = \inf_{b>a} \frac{\mathbf{E}[\min\{S,b\} - a \mid S > a]}{\mathbf{P}[S \le b \mid S > a]}$$

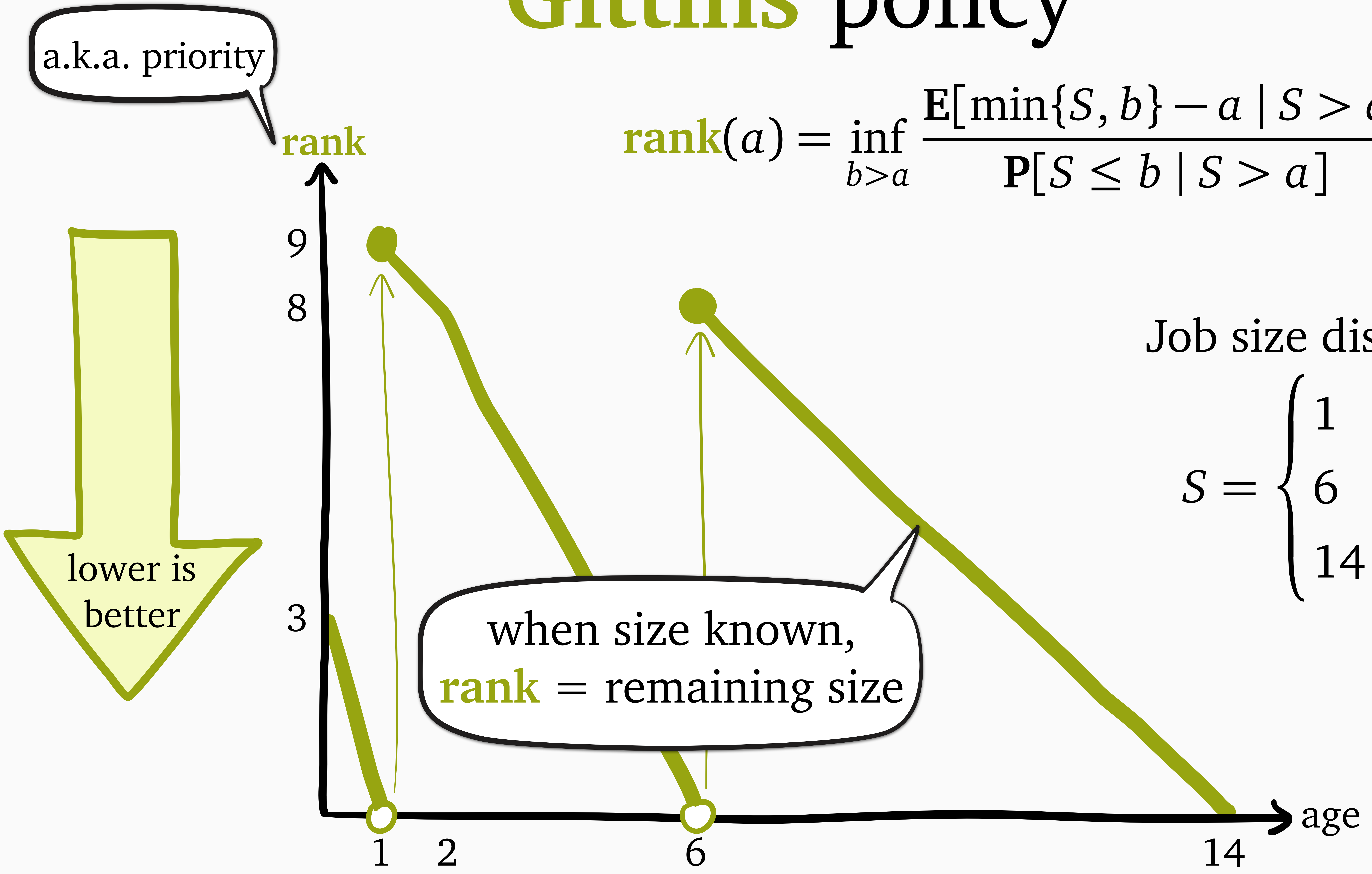


Job size distribution:

$$S = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

Gittins policy

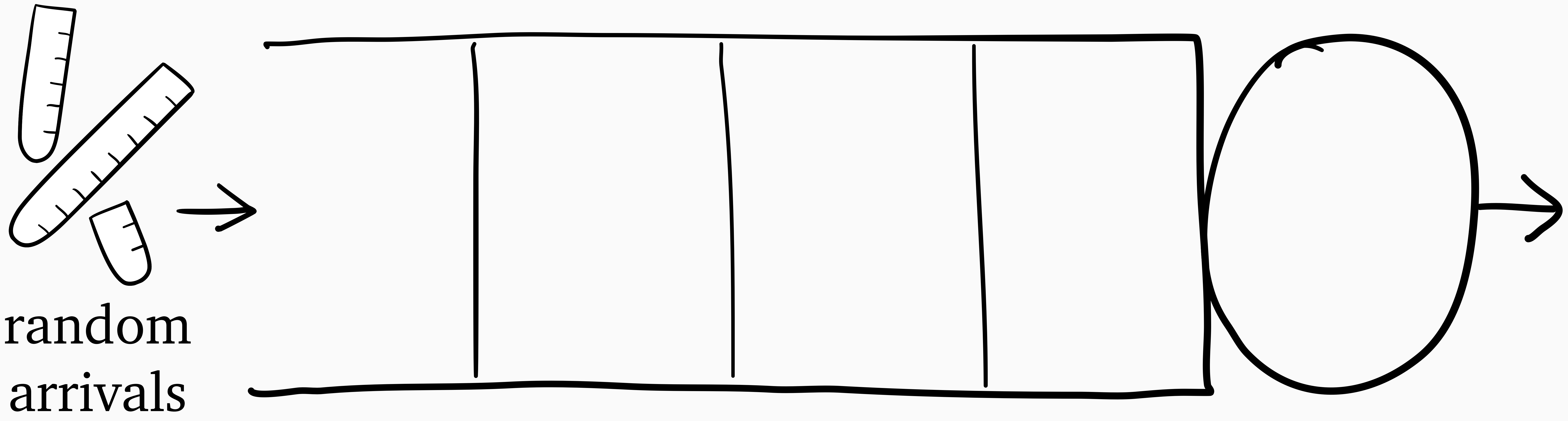
$$\mathbf{rank}(a) = \inf_{b>a} \frac{\mathbf{E}[\min\{S, b\} - a \mid S > a]}{\mathbf{P}[S \leq b \mid S > a]}$$



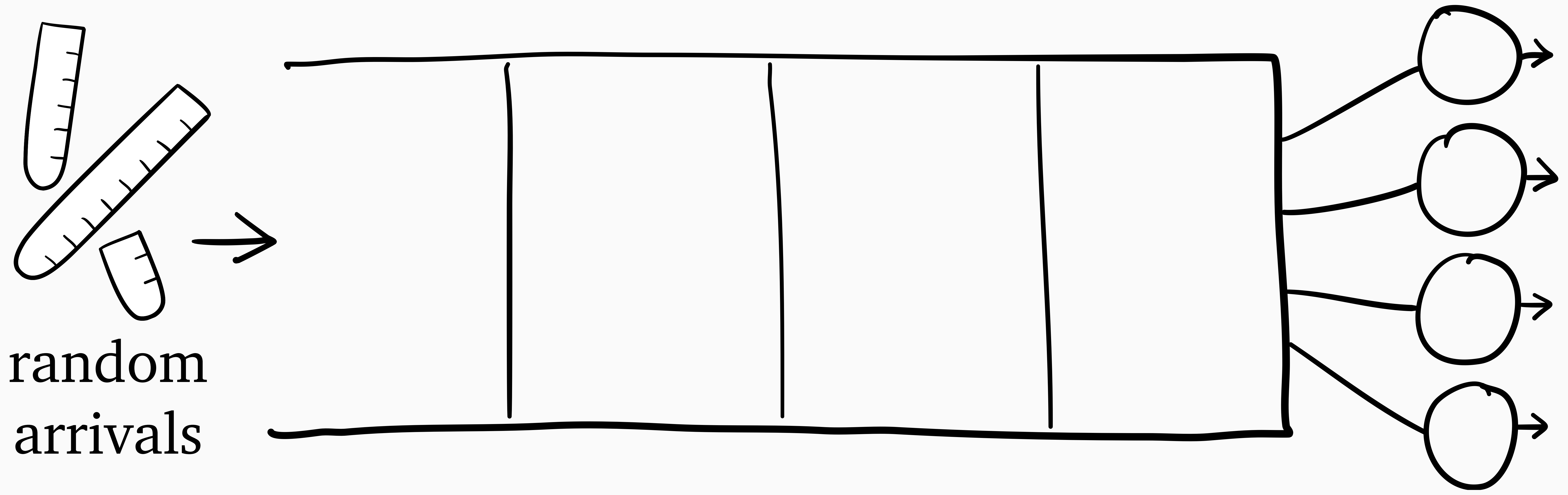
Job size distribution:

$$S = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

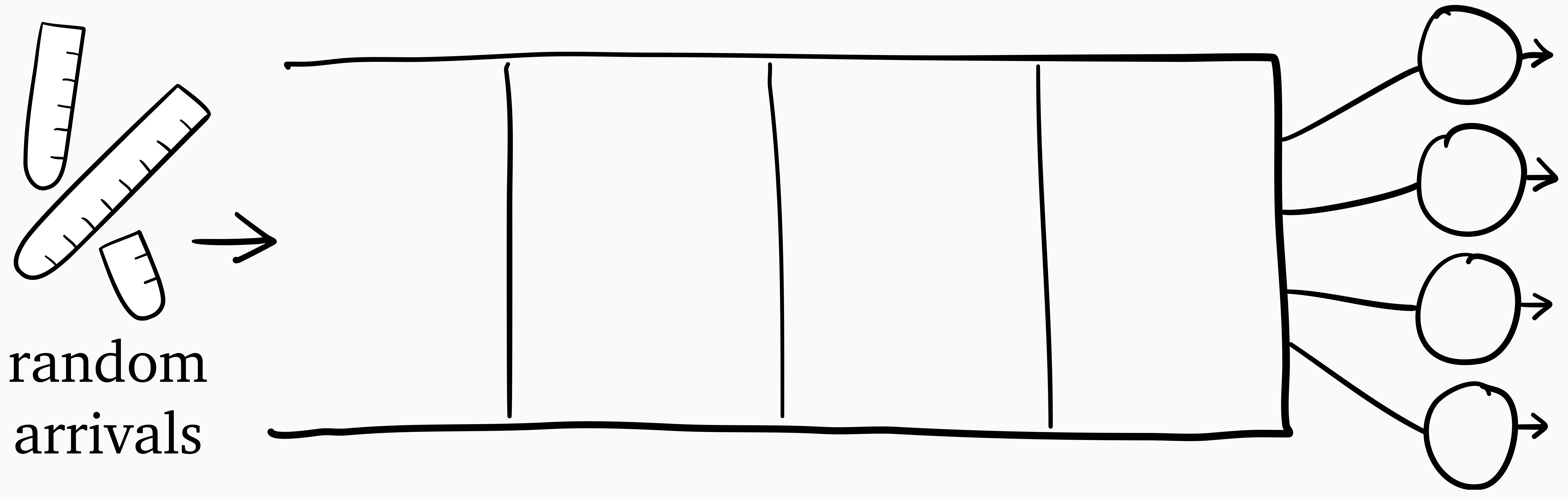
M/G/*k* queue



M/G/*k* queue

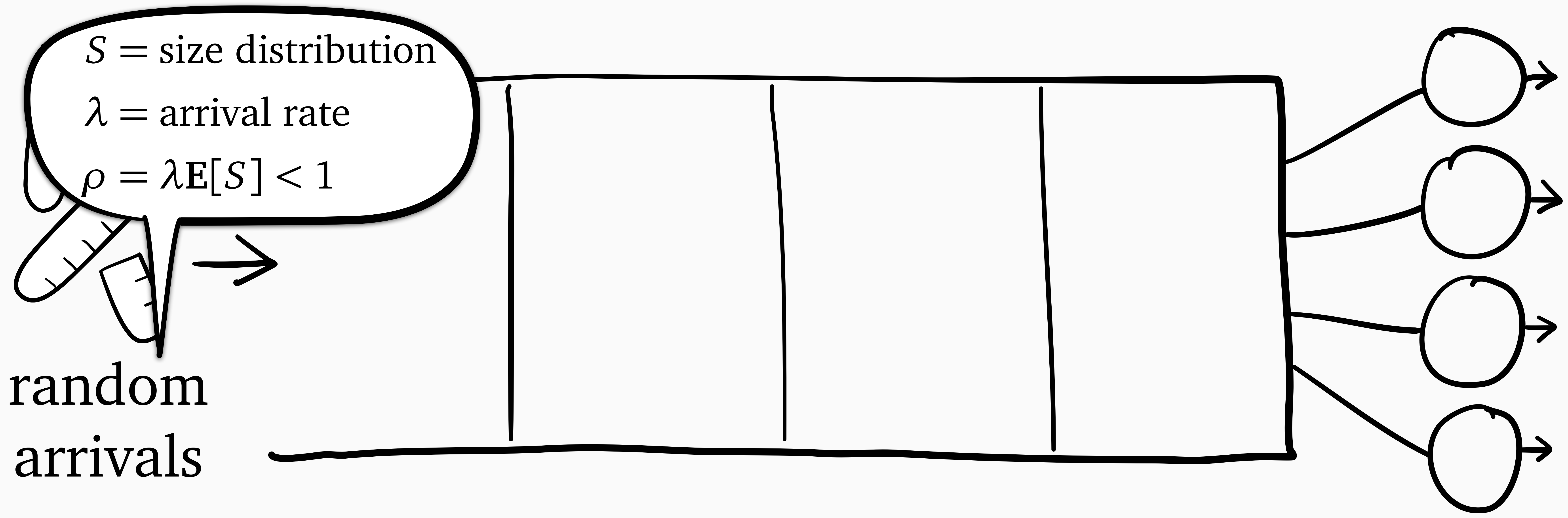


M/G/k queue



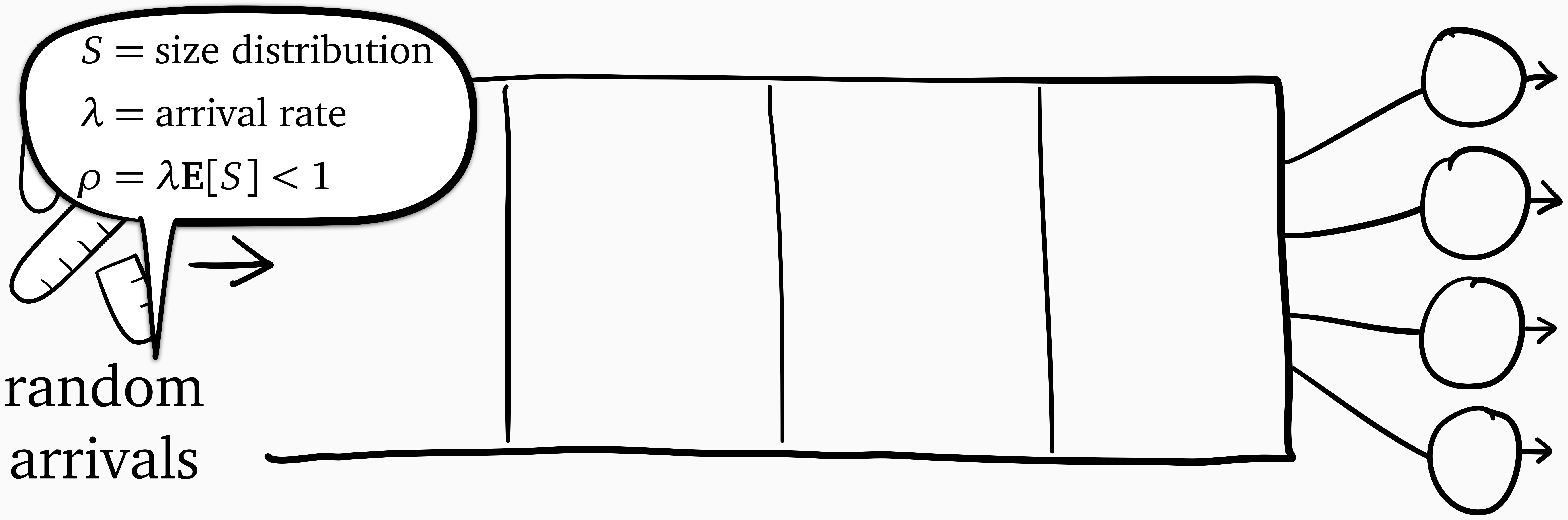
k servers, each speed $1/k$

M/G/*k* queue

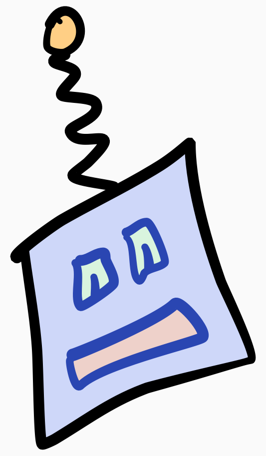


k servers, each speed $1/k$

M/G/k queue

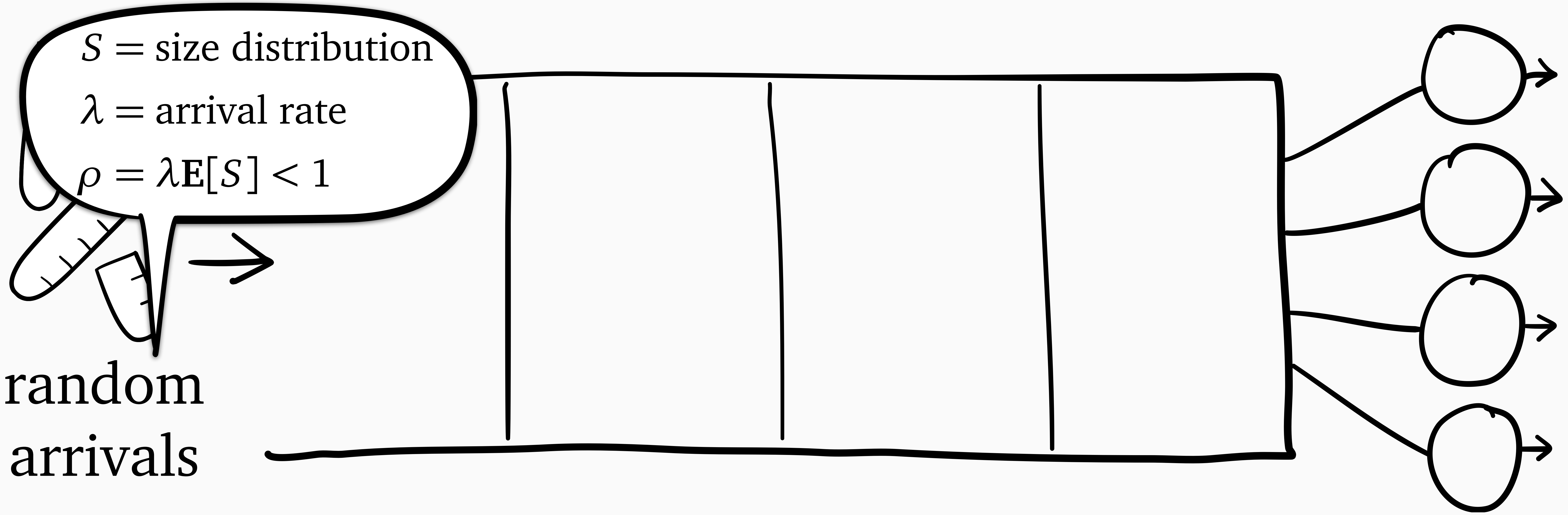


k servers, each speed $1/k$



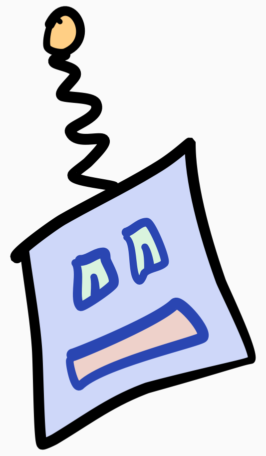
Scheduling policy:
picks which k jobs to serve

M/G/k queue



k servers, each speed $1/k$

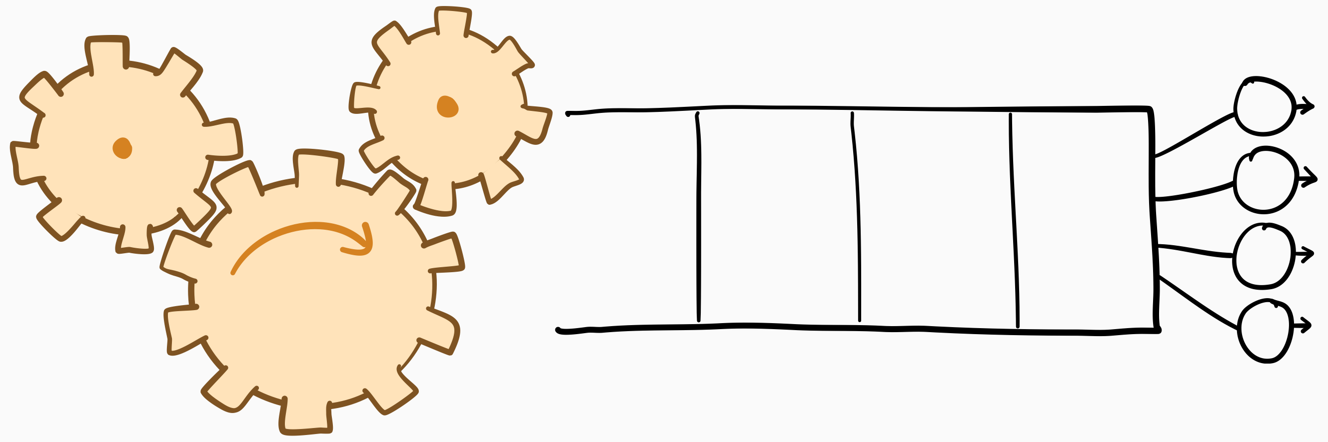
Multiserver Gittins: serves the k jobs with the k lowest **ranks**



Scheduling policy: picks which k jobs to serve

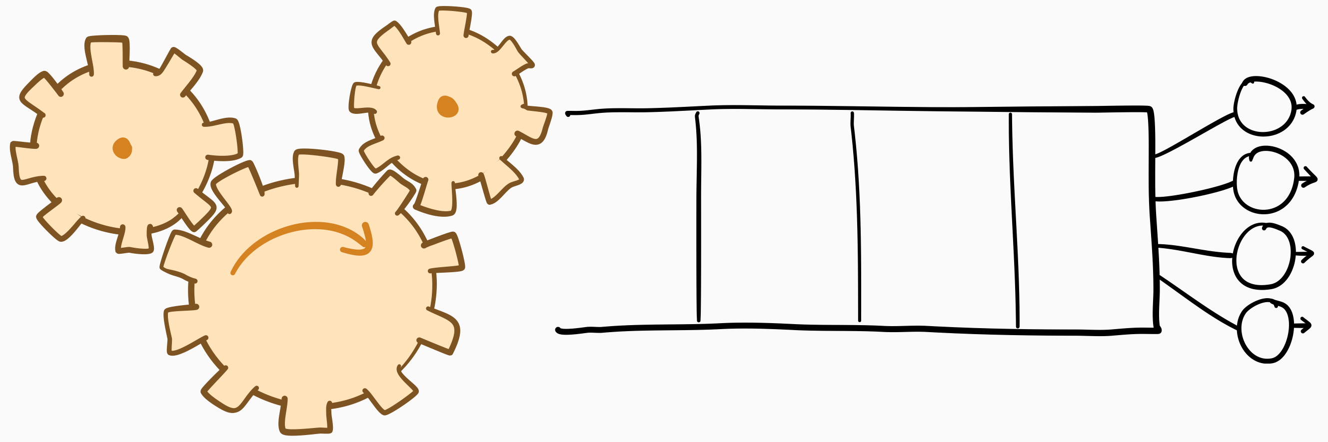
Comparing M/G/k to M/G/1

M/G/k

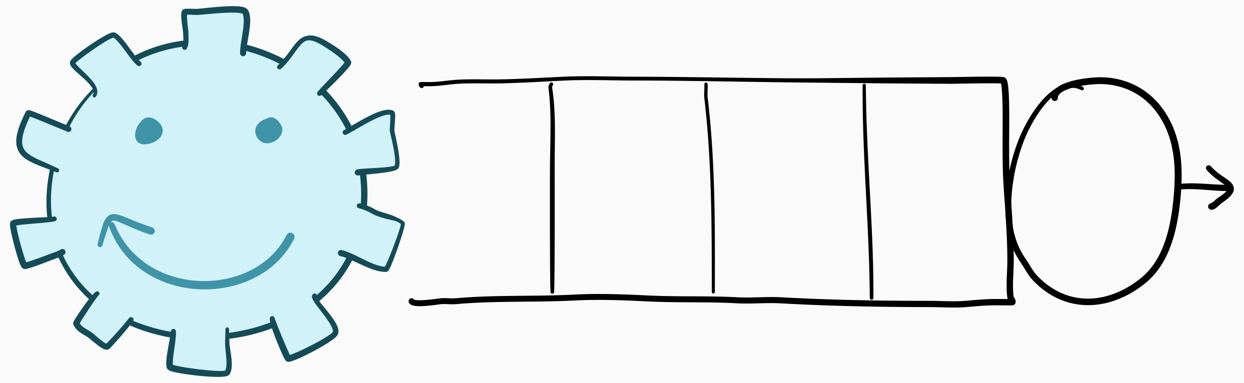


Comparing M/G/*k* to M/G/1

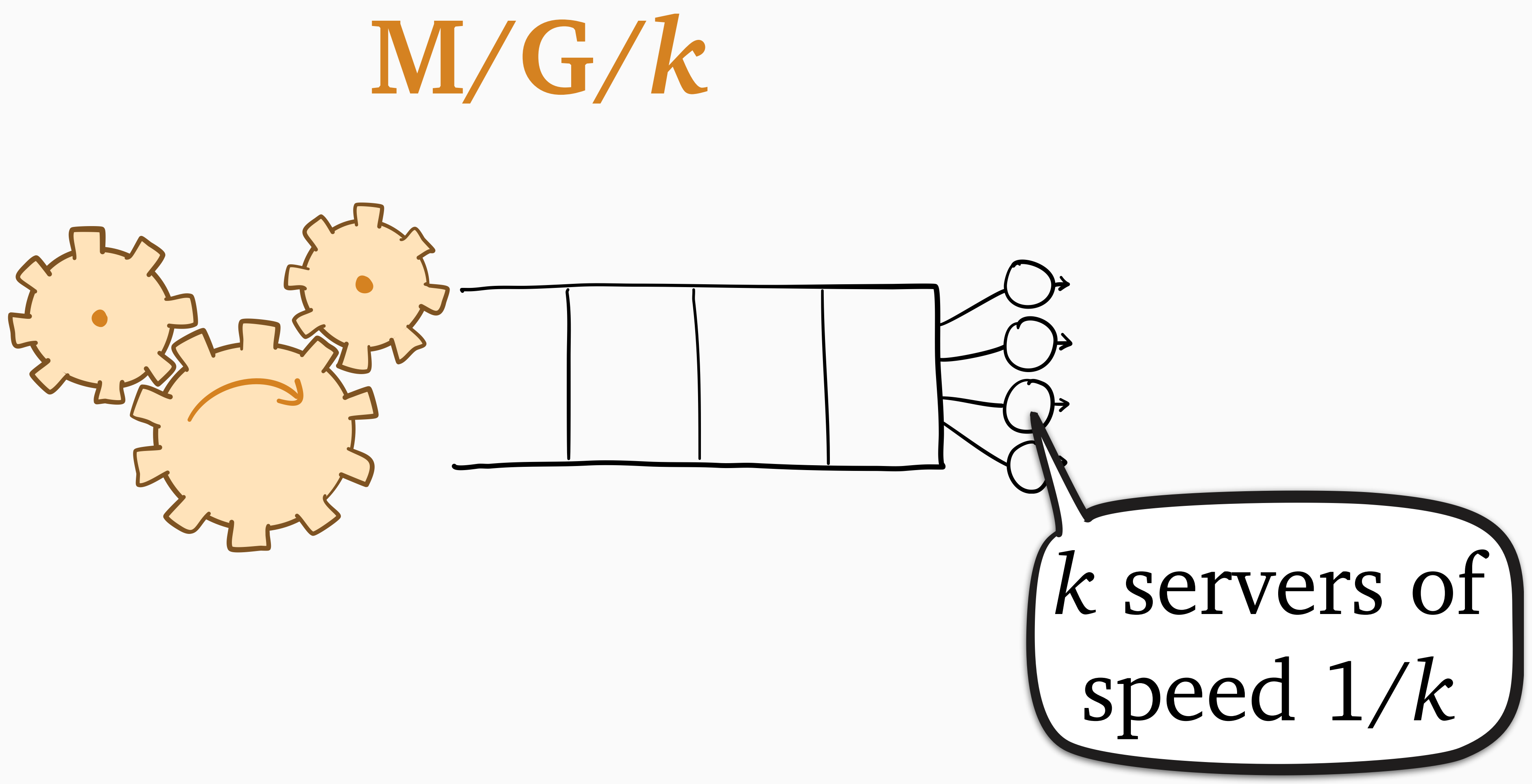
M/G/*k*

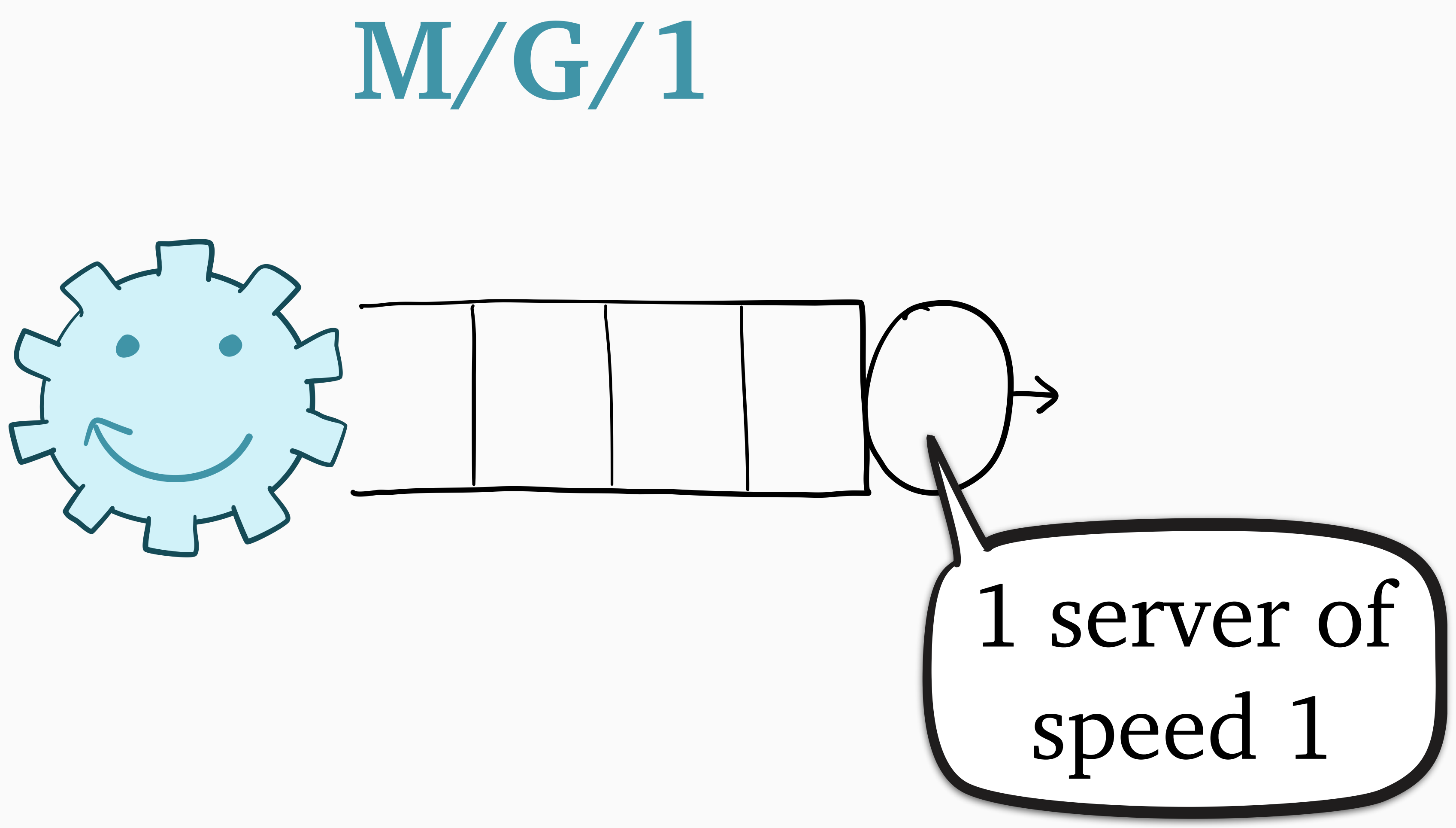


M/G/1

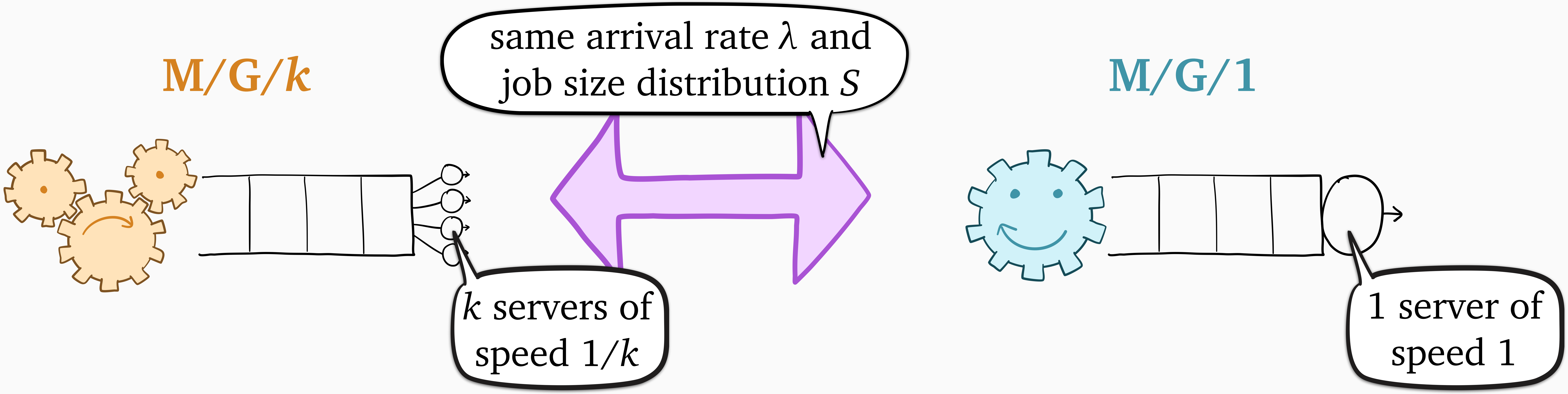


Comparing M/G/k to M/G/1

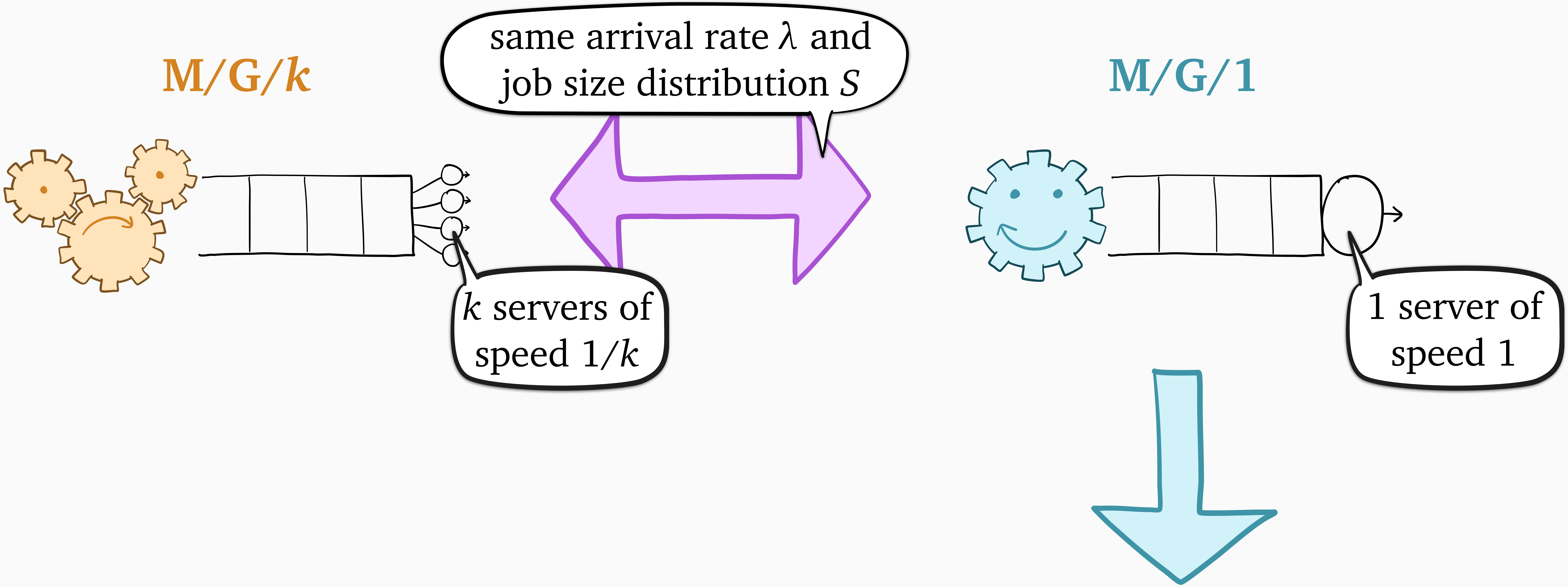




Comparing M/G/k to M/G/1

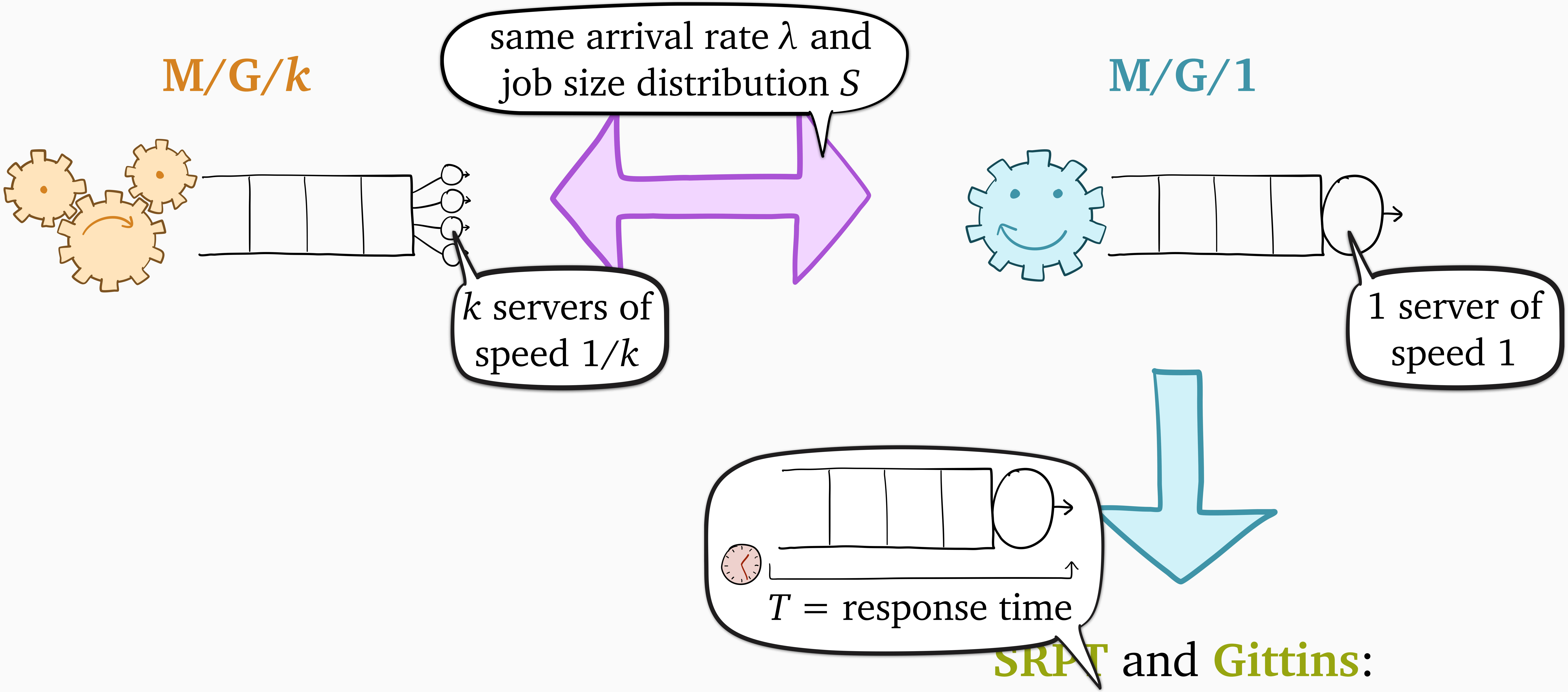


Comparing M/G/k to M/G/1



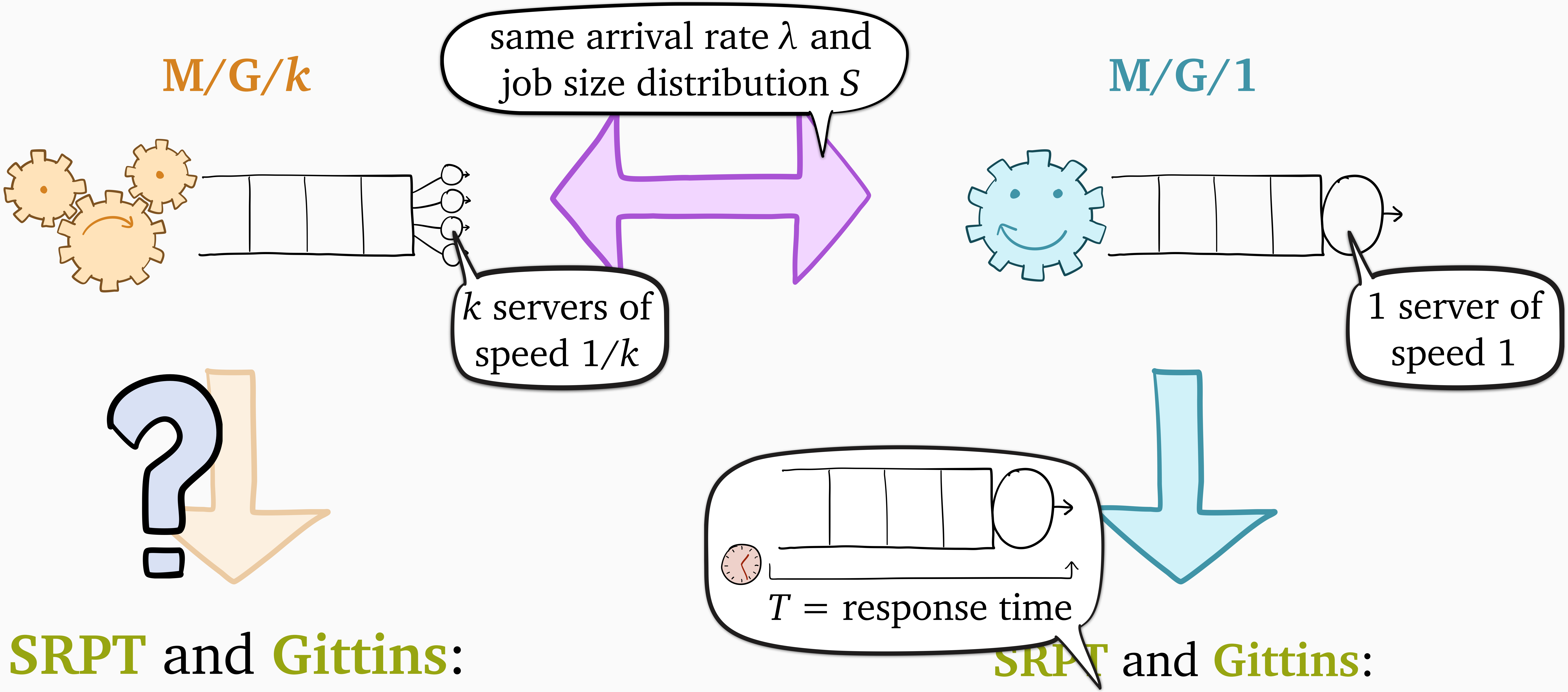
SRPT and Gittins:
$\mathbf{E}[T]$ optimal

Comparing M/G/k to M/G/1



SRPT and Gittins:
$\mathbf{E}[T]$ optimal

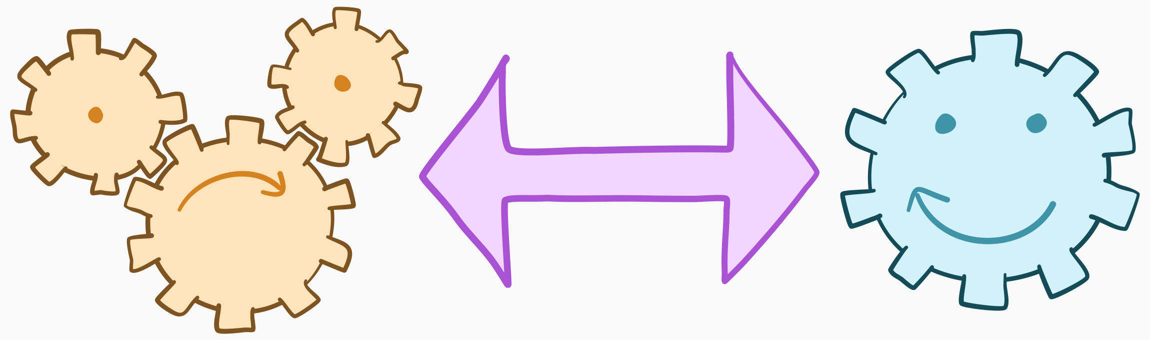
Comparing M/G/k to M/G/1



SRPT and **Gittins**:
E[T] **near-optimal**?

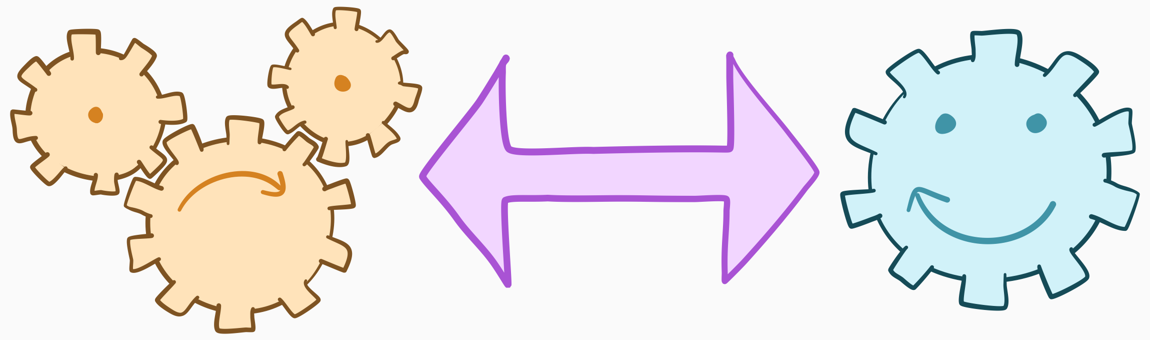
SRPT and **Gittins**:
E[T] **optimal**

Classifying coupling techniques



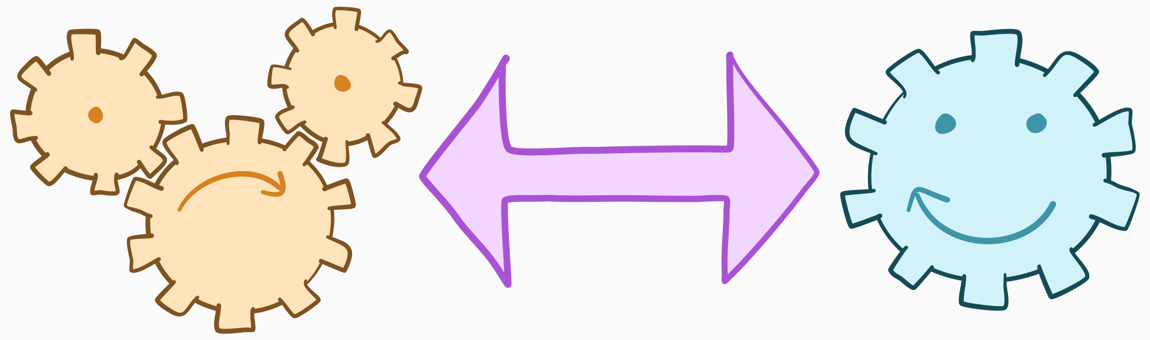
	A. Every sample path	B. Steady-state distribution
1. More information	A1 BIG online knapsack (via compensated coupling)	B1
2. Fewer constraints	A2 M/M/*k* vs. M/M/1 Online knapsack (via constraints-to-costs)	B2 State-space collapse (load balancing, switch scheduling)
3. Simpler dynamics	A3 SIS epidemics Queues with redundancy	B3 Stein's method

Classifying coupling techniques



	A. Every sample path	B. Steady-state distribution
1. More information	A1 BIG online knapsack (via compensated coupling)	B1
2. Fewer constraints	A2 M/M/k vs. M/M/1 SRPT in the M/G/k (Grosof et al. 2018)	B2 State-space collapse (load balancing, switch scheduling)
3. Simpler dynamics	A3 SIS epidemics Queues with redundancy	B3 Stein's method

Classifying coupling techniques



	A. Every sample path	B. Steady-state distribution
1. More information	A1 BIG online knapsack (via compensated coupling)	B1
2. Fewer constraints	A2 M/M/k vs. M/M/1	B2 State-space collapse (load balancing, switch scheduling) SRPT/Gittins in the M/G/k
3. Simpler dynamics	A3 SIS epidemics Queues with redundancy	B3 Stein's method

SRPT in the M/G/k
(Grosof et al. 2018)

Gittins in the M/G/k: result

Theorem: under SRPT and Gittins,

$$\mathbf{E}[T_k] \le \mathbf{E}[T_1] + (k-1) \cdot O\left(\log \frac{1}{1-\rho}\right)$$

Scully, Grosof, and Harchol-Balter: The Gittins Policy is Nearly Optimal in the M/G/k under Extremely General Conditions (2020)

Gittins in the M/G/k: result

Theorem: under SRPT and Gittins,

$$\mathbf{E}[T_k] \leq \mathbf{E}[T_1] + (k-1) \cdot O\left(\log \frac{1}{1-\rho}\right)$$

Theorem: under SRPT and Gittins,

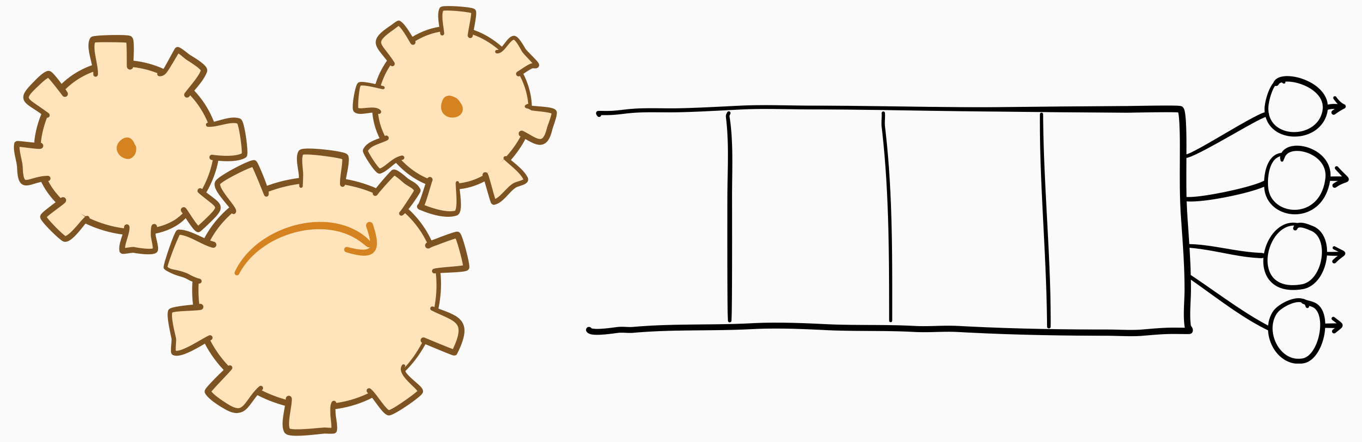
$$\lim_{\rho \to 1} \frac{\mathbf{E}[T_k]}{\mathbf{E}[T_1]} = 1 \quad \text{if } \mathbf{E}[S^2 (\log S)^+] < \infty$$

Scully, Grosof, and Harchol-Balter: The Gittins Policy is Nearly Optimal in the M/G/k under Extremely General Conditions (2020)

Gittins in the M/G/k: result

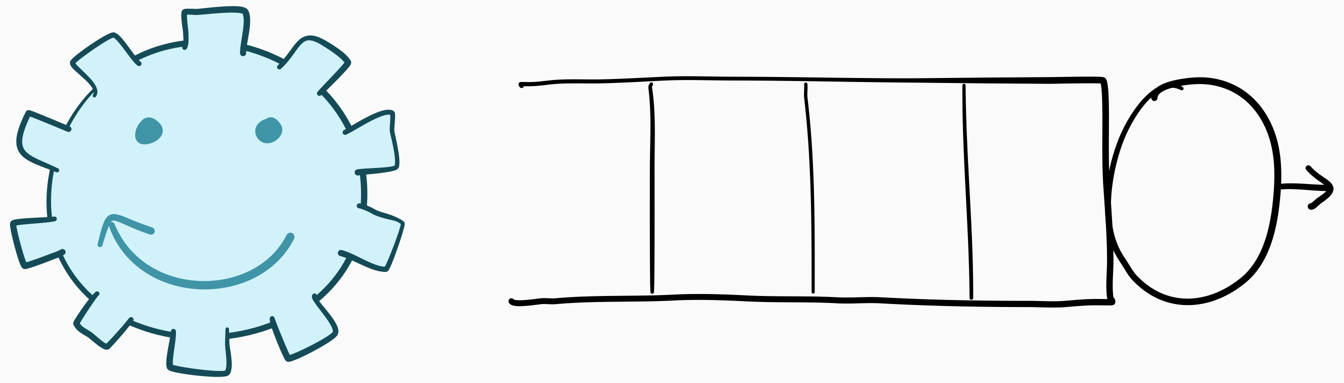
Theorem: under SRPT and Gittins,

$$\mathbf{E}[T_k] \le \mathbf{E}[T_1] + (k-1)\cdot O\left(\log \frac{1}{1-\rho}\right)$$



Theorem: under SRPT and Gittins,

$$\lim_{\rho \to 1} \frac{\mathbf{E}[T_k]}{\mathbf{E}[T_1]} = 1 \quad \text{if } \mathbf{E}[S^2(\log S)^+] < \infty$$

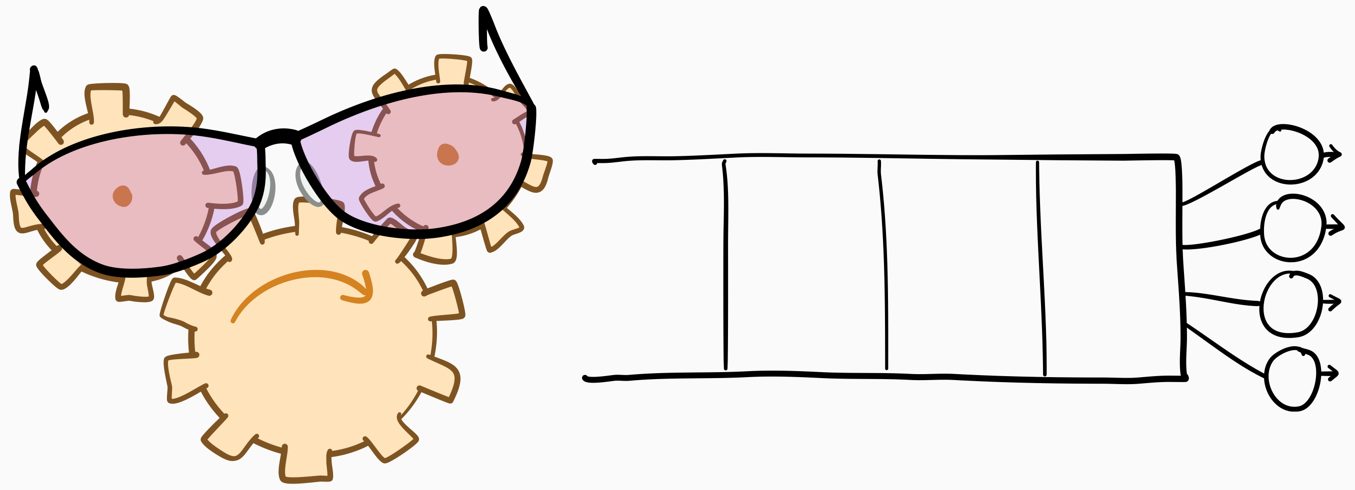


Scully, Grosof, and Harchol-Balter: The Gittins Policy is Nearly Optimal in the M/G/k under Extremely General Conditions (2020)

Gittins in the M/G/k: result

Theorem: under SRPT and Gittins,

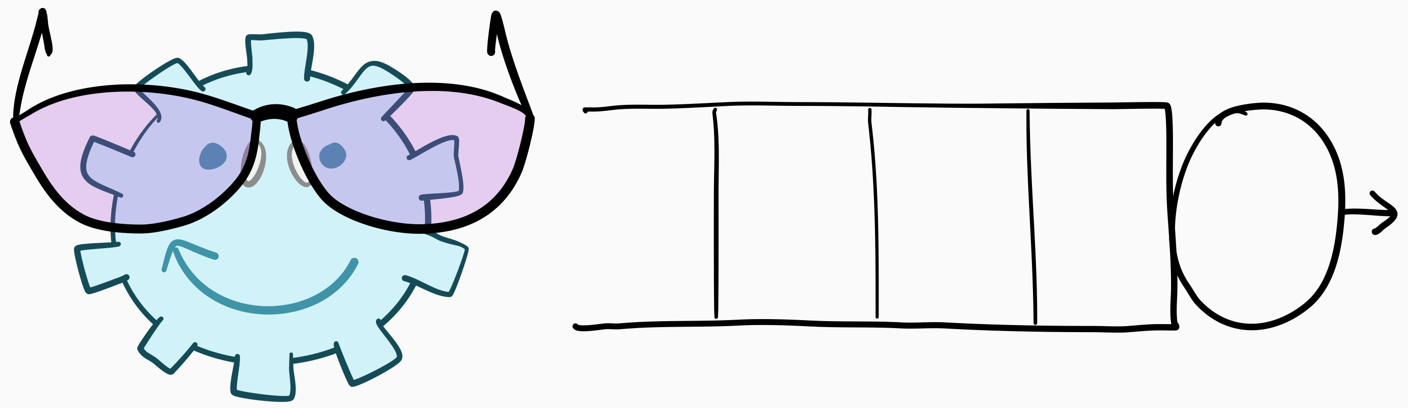
$$\mathbf{E}[T_k] \leq \mathbf{E}[T_1] + (k-1) \cdot O\left(\log \frac{1}{1-\rho}\right)$$



New idea: *r*-work

Theorem: under SRPT and Gittins,

$$\lim_{\rho \to 1} \frac{\mathbf{E}[T_k]}{\mathbf{E}[T_1]} = 1 \quad \text{if } \mathbf{E}[S^2 (\log S)^+] < \infty$$



Scully, Grosof, and Harchol-Balter: The Gittins Policy is Nearly Optimal in the M/G/k under Extremely General Conditions (2020)

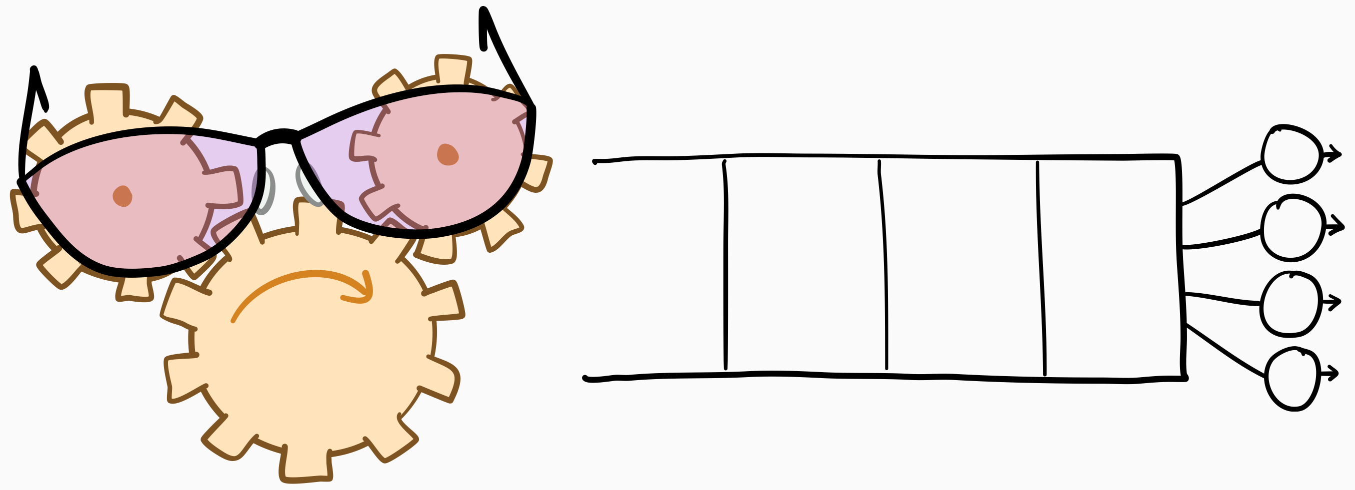
Gittins in the M/G/k: result

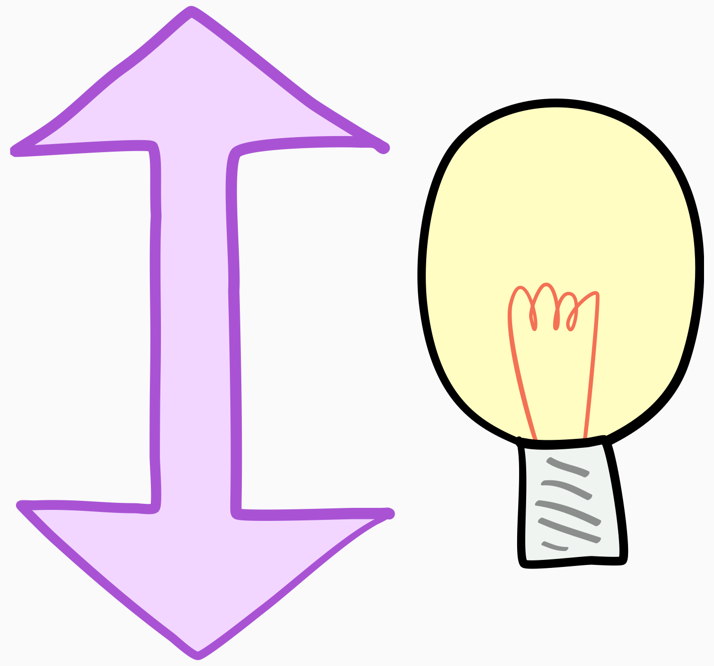
Theorem: under SRPT and Gittins,

$$\mathbf{E}[T_k] \le \mathbf{E}[T_1] + (k-1)\cdot O\left(\log \frac{1}{1-\rho}\right)$$

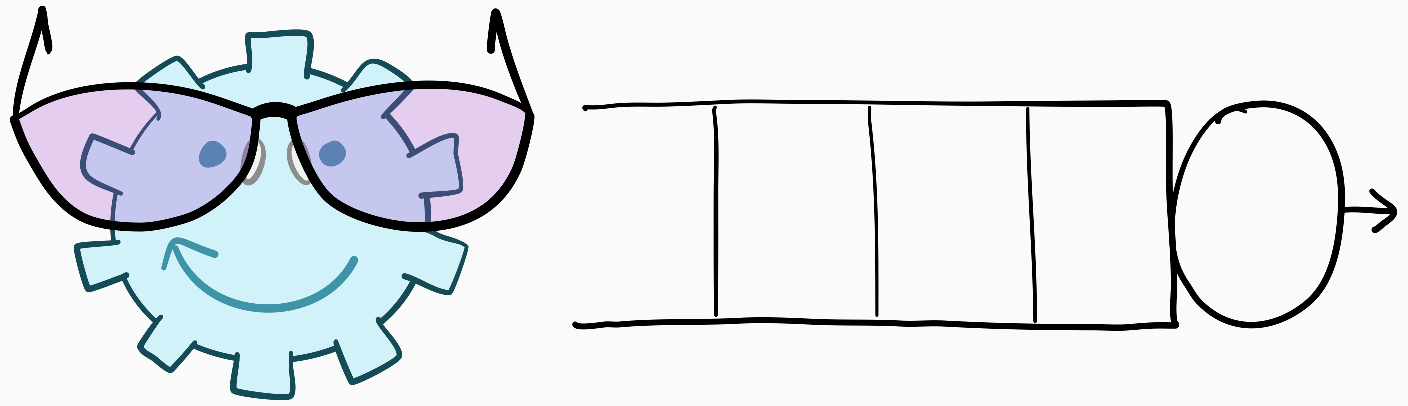
Theorem: under SRPT and Gittins,

$$\lim_{\rho \to 1} \frac{\mathbf{E}[T_k]}{\mathbf{E}[T_1]} = 1 \quad \text{if } \mathbf{E}[S^2(\log S)^+] < \infty$$





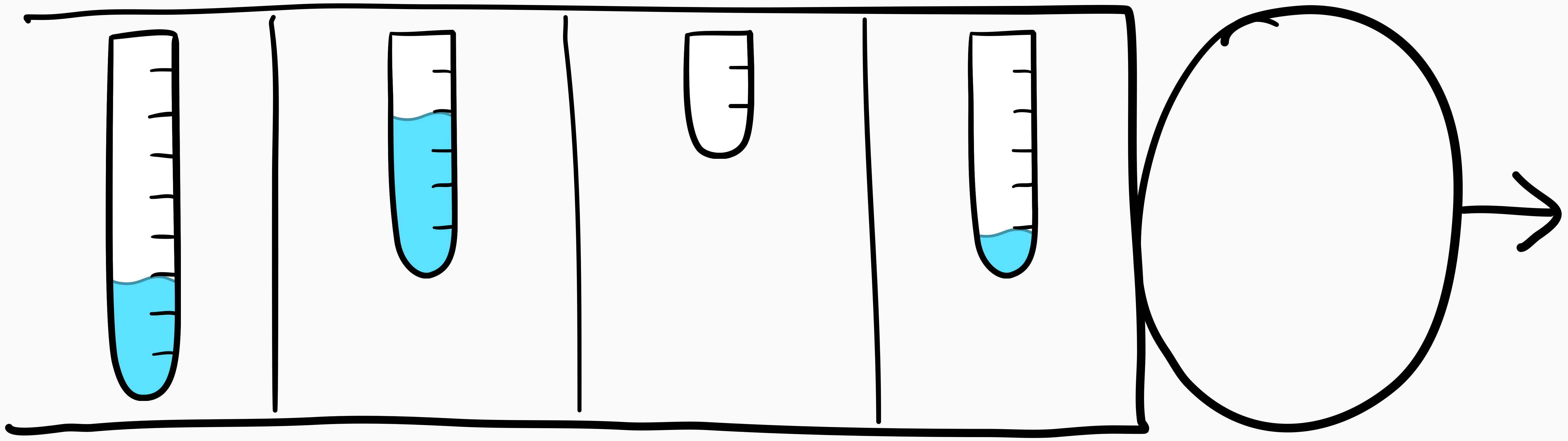
New idea:
***r*-work**



Scully, Grosof, and Harchol-Balter: The Gittins Policy is Nearly Optimal in the M/G/k under Extremely General Conditions (2020)

What is *r*-work? (SRPT)

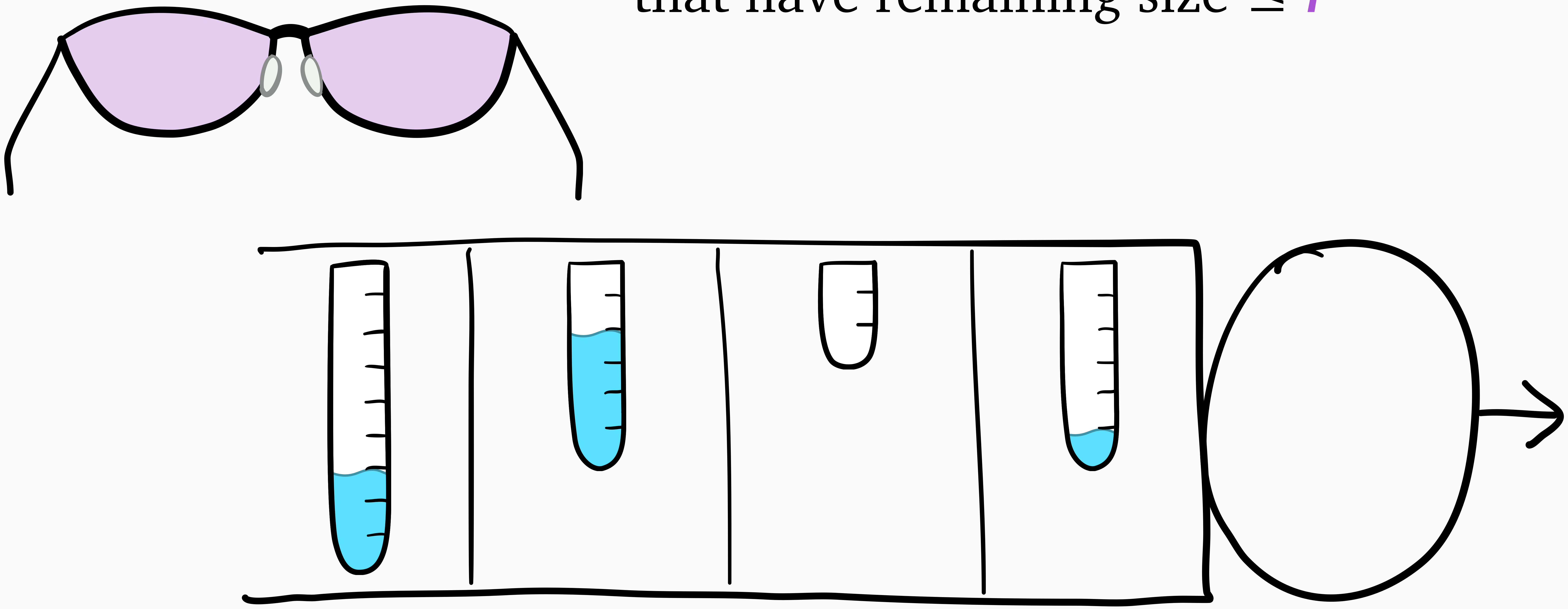
W = work = total remaining size of all jobs



What is *r*-work? (SRPT)

W = work = total remaining size of all jobs

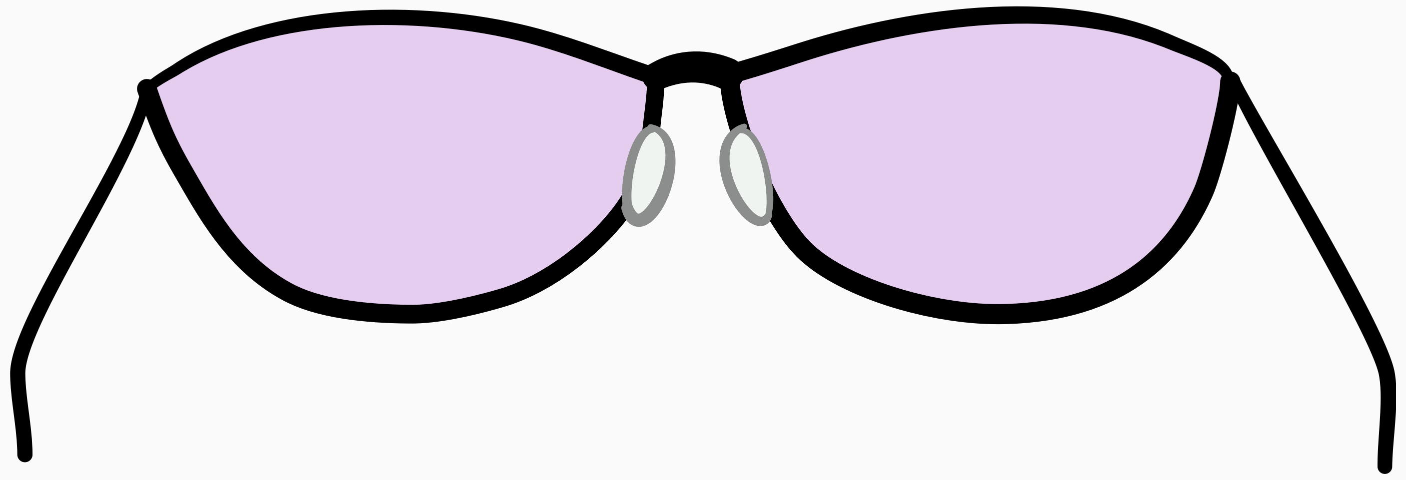
$W(r)$ = r-work = total remaining size of all jobs that have remaining size $\leq r$

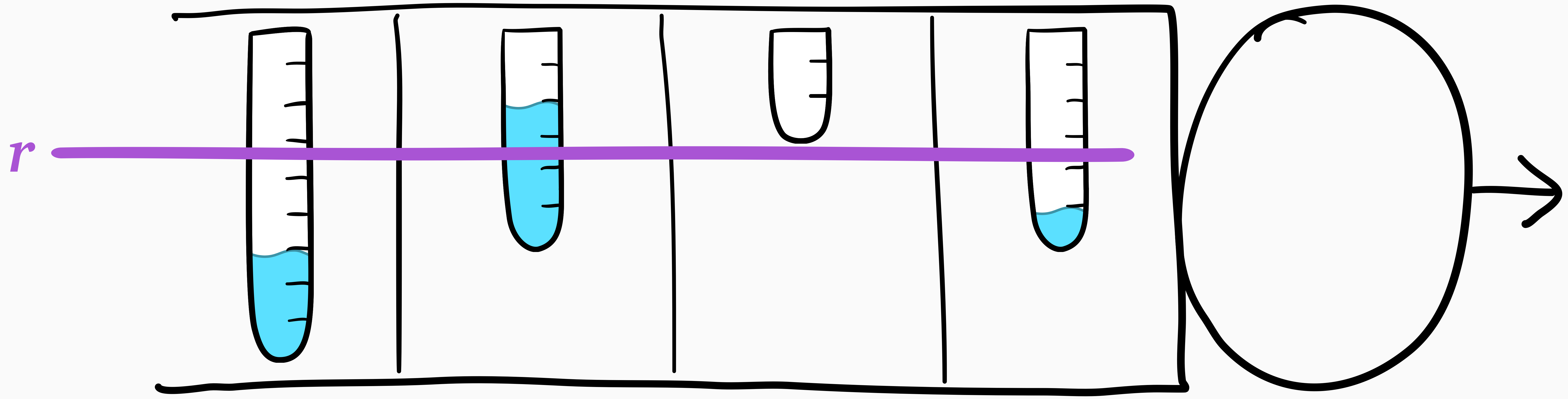


What is *r*-work? (SRPT)

W = work = total remaining size of all jobs

$W(r)$ = r-work = total remaining size of all jobs that have remaining size $\leq r$

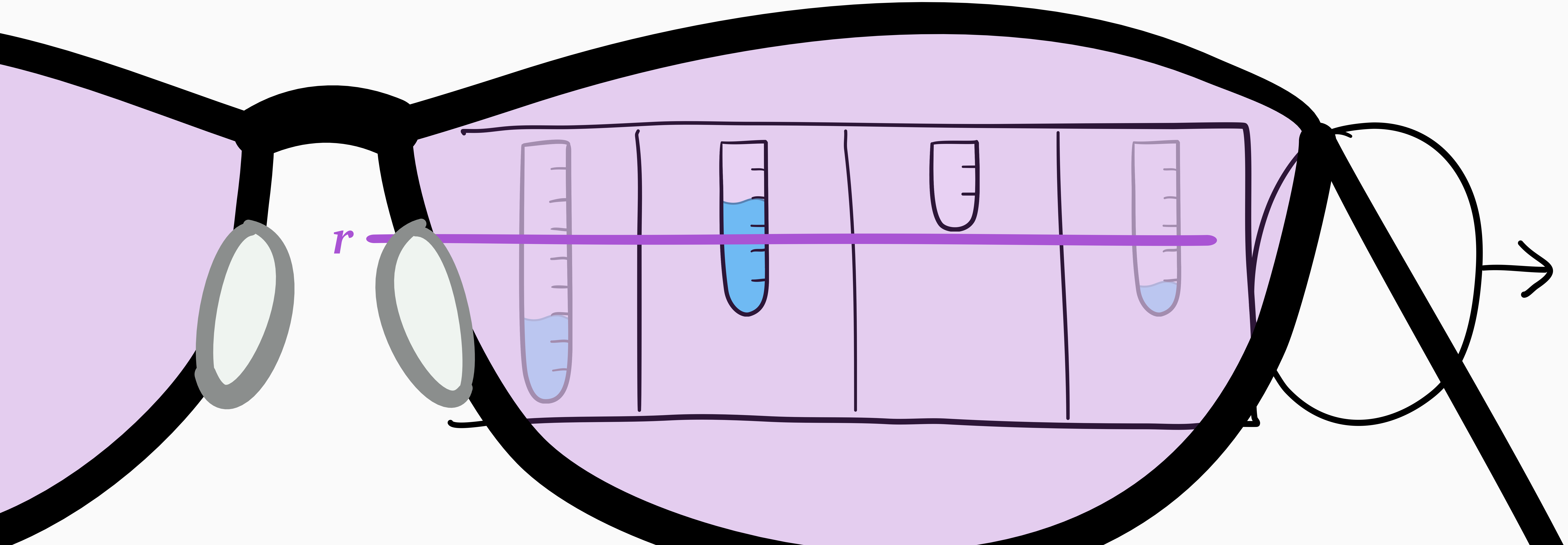




What is r-work? (SRPT)

W = work = total remaining size of all jobs

$W(r)$ = r-work = total remaining size of all jobs that have remaining size $\leq r$



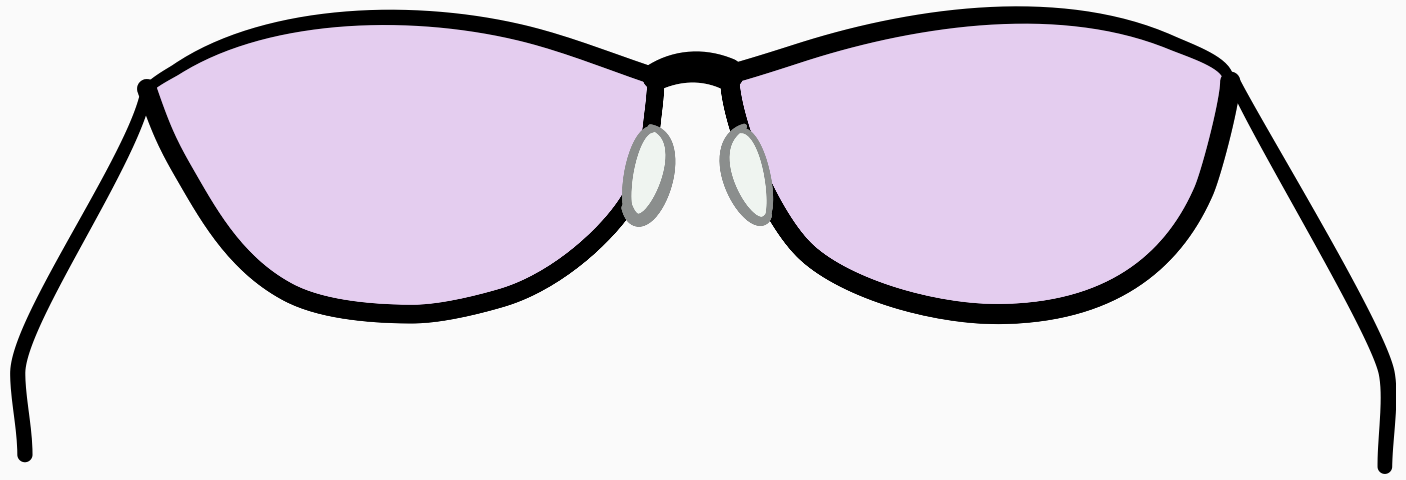
What is ***r*-work**? (**Gittins**)

$W = \text{work} = \text{total remaining size of all jobs}$

What is *r*-work? (Gittins)

W = work = total remaining size of all jobs

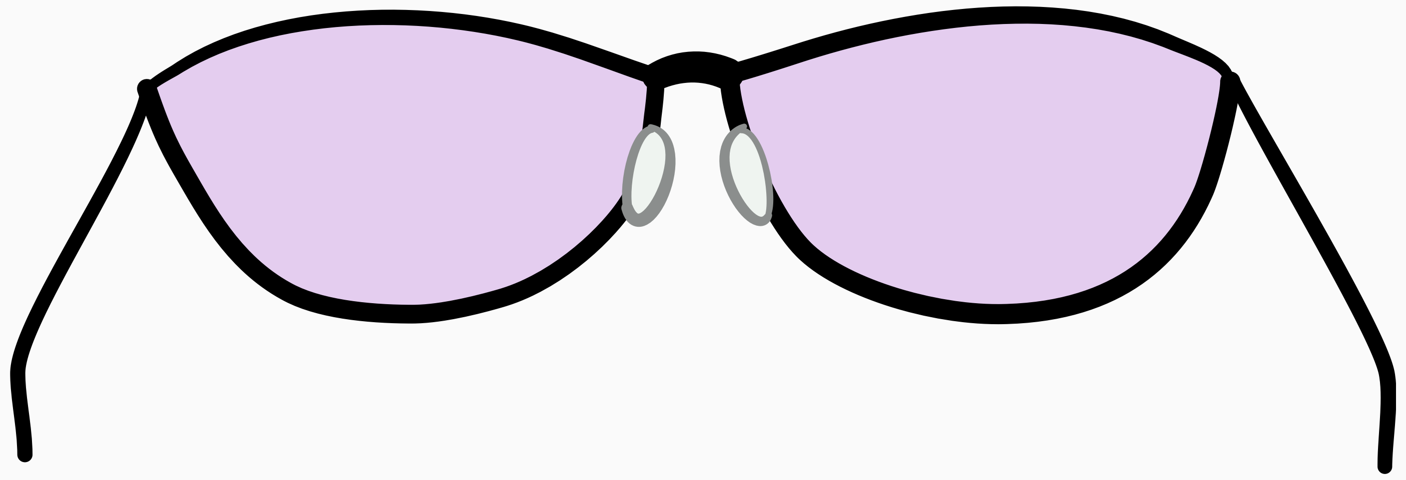
$W(r)$ = *r*-work = work "relevant" to **rank** r



What is *r*-work? (Gittins)

W = work = total remaining size of all jobs

$W(r)$ = r-work = work "relevant" to **rank** r

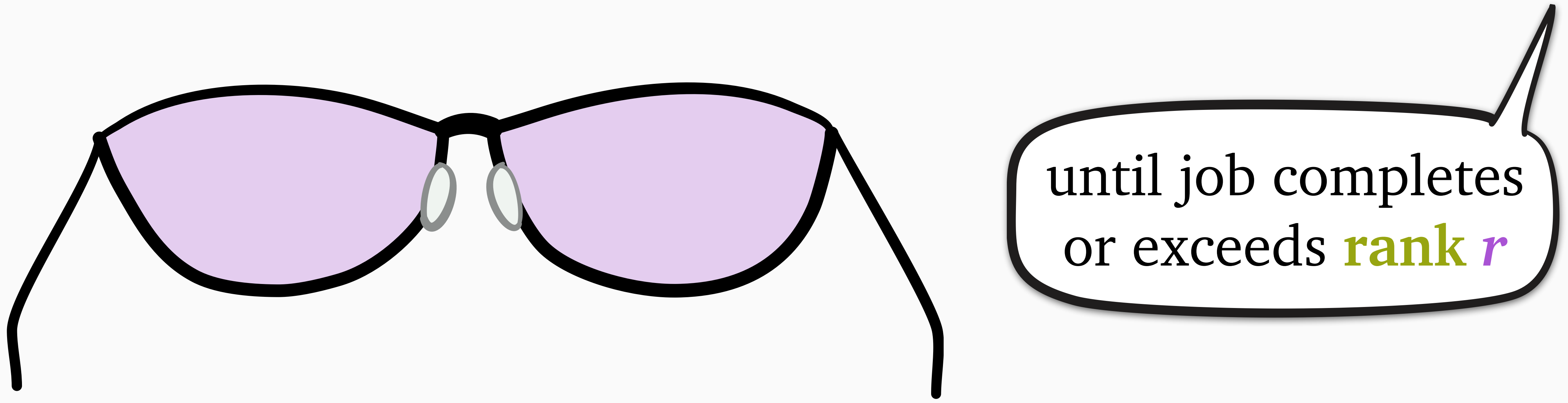


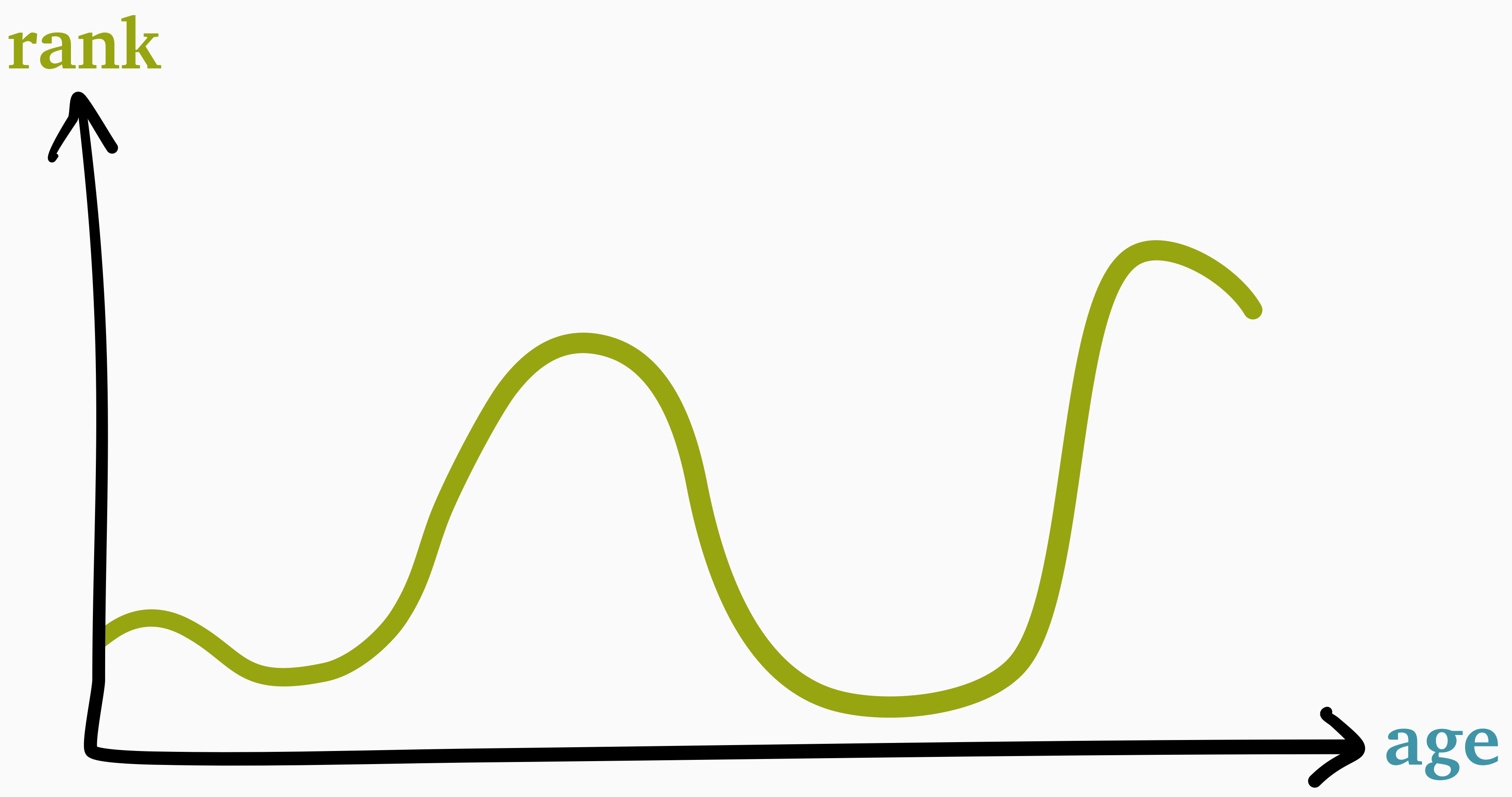
until job completes
or exceeds **rank** r

What is *r*-work? (Gittins)

W = work = total remaining size of all jobs

$W(r)$ = *r*-work = work “relevant” to **rank** r

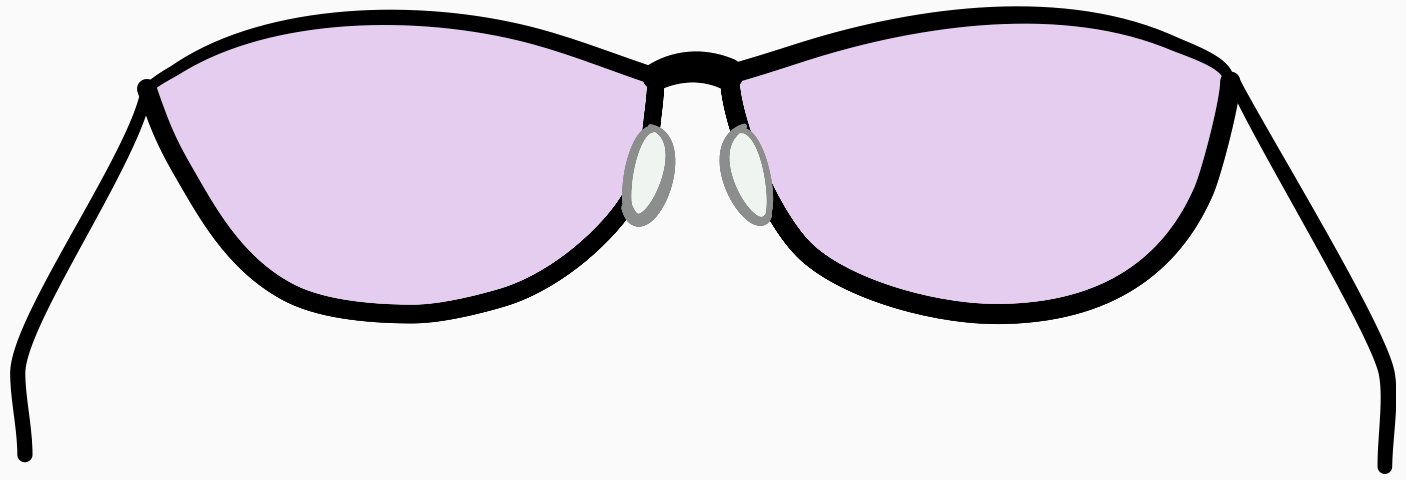




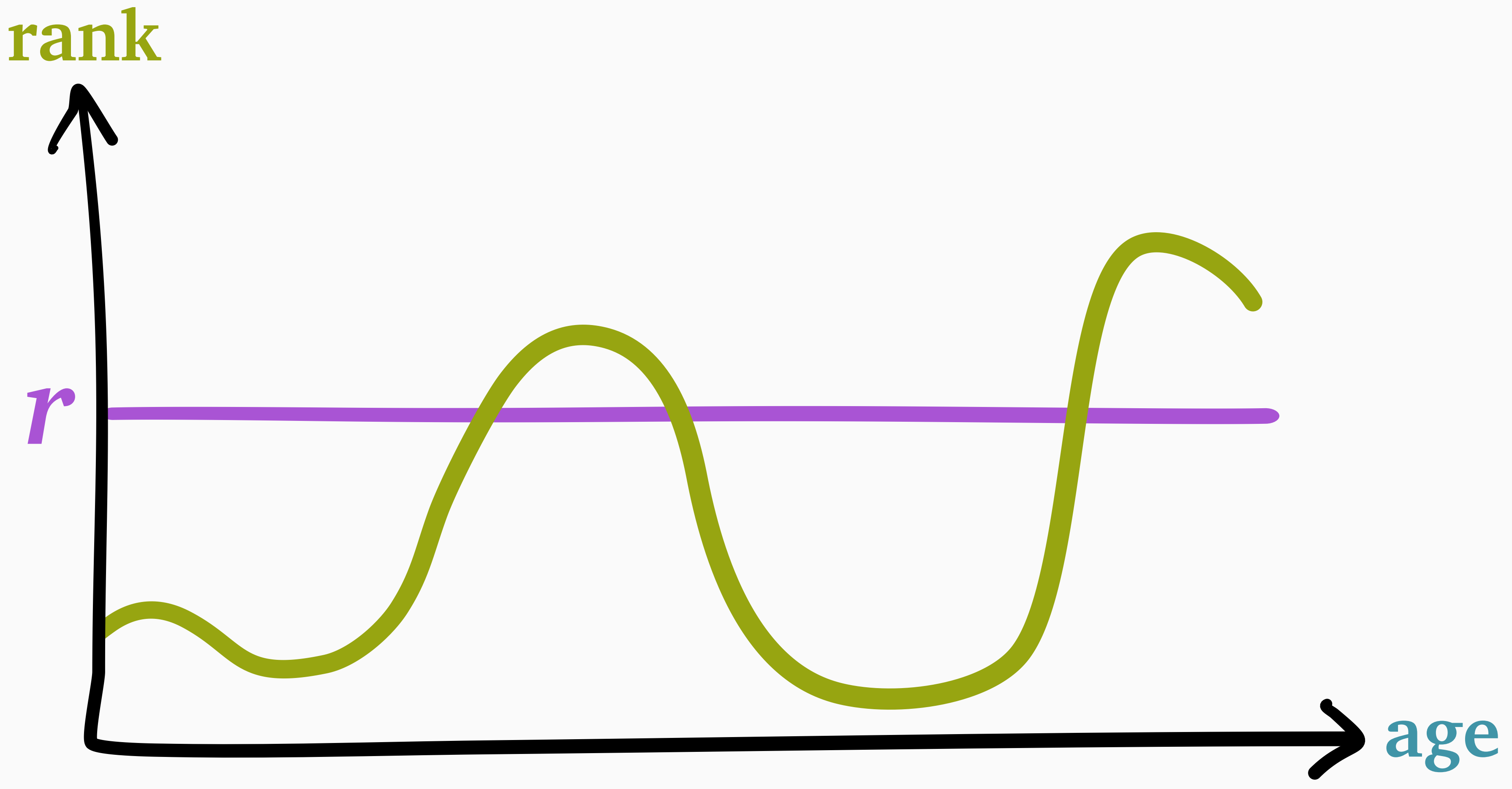
What is *r*-work? (Gittins)

W = work = total remaining size of all jobs

$W(r)$ = r-work = work "relevant" to rank r



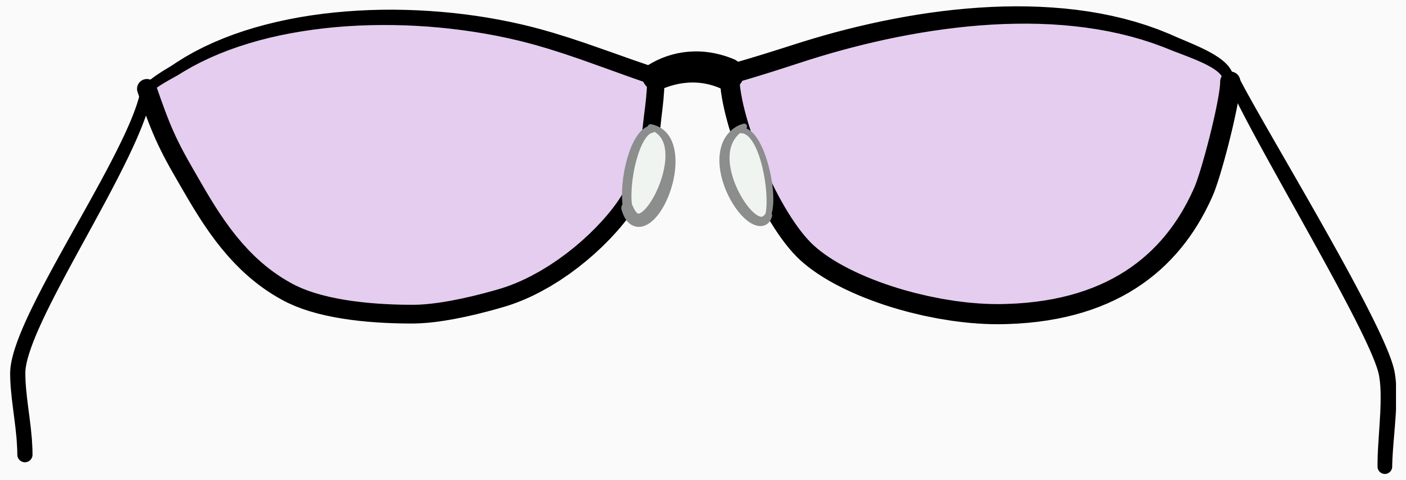
until job completes
or exceeds rank r



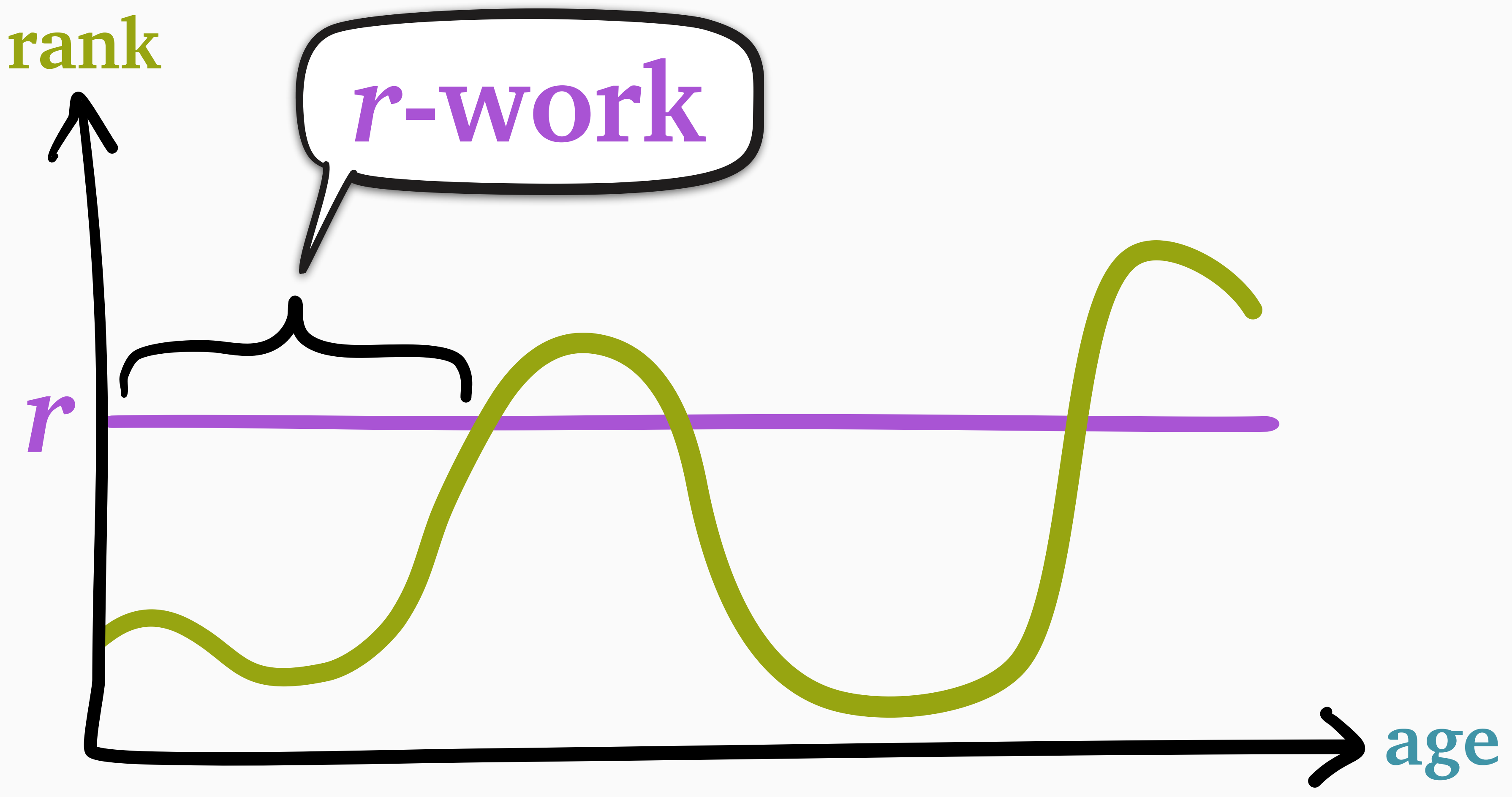
What is *r*-work? (Gittins)

W = work = total remaining size of all jobs

$W(r)$ = r-work = work "relevant" to rank r



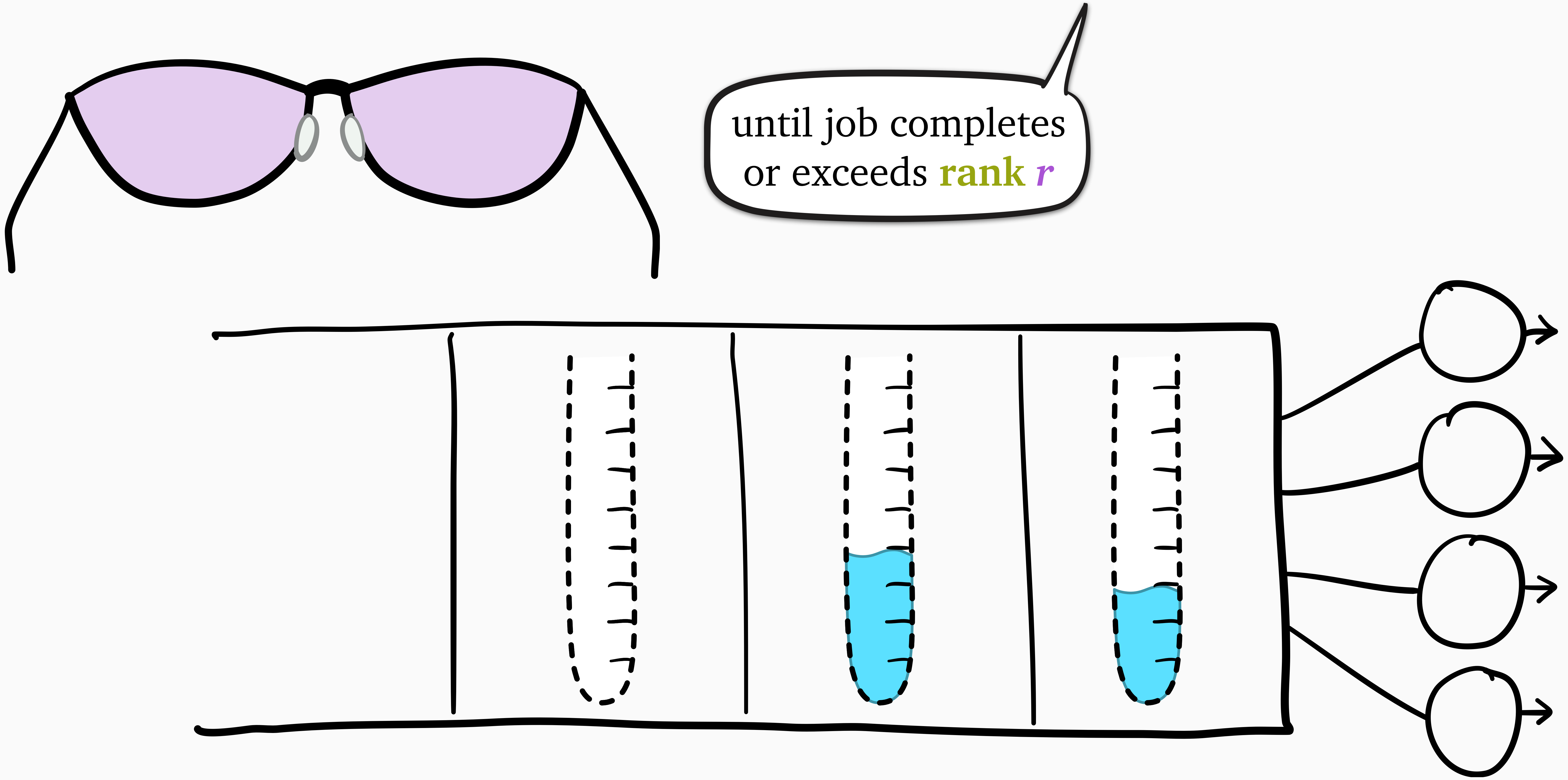
until job completes
or exceeds rank r



What is *r*-work? (Gittins)

W = work = total remaining size of all jobs

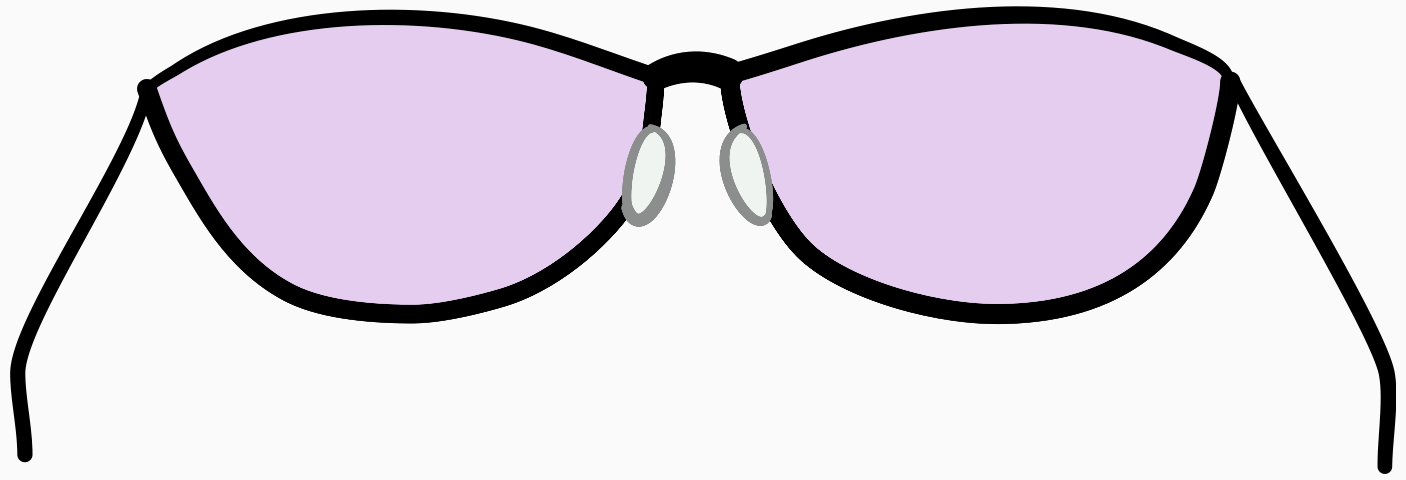
$W(r)$ = r-work = work "relevant" to rank r



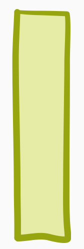
What is *r*-work? (Gittins)

W = work = total remaining size of all jobs

$W(r)$ = r-work = work "relevant" to **rank** r

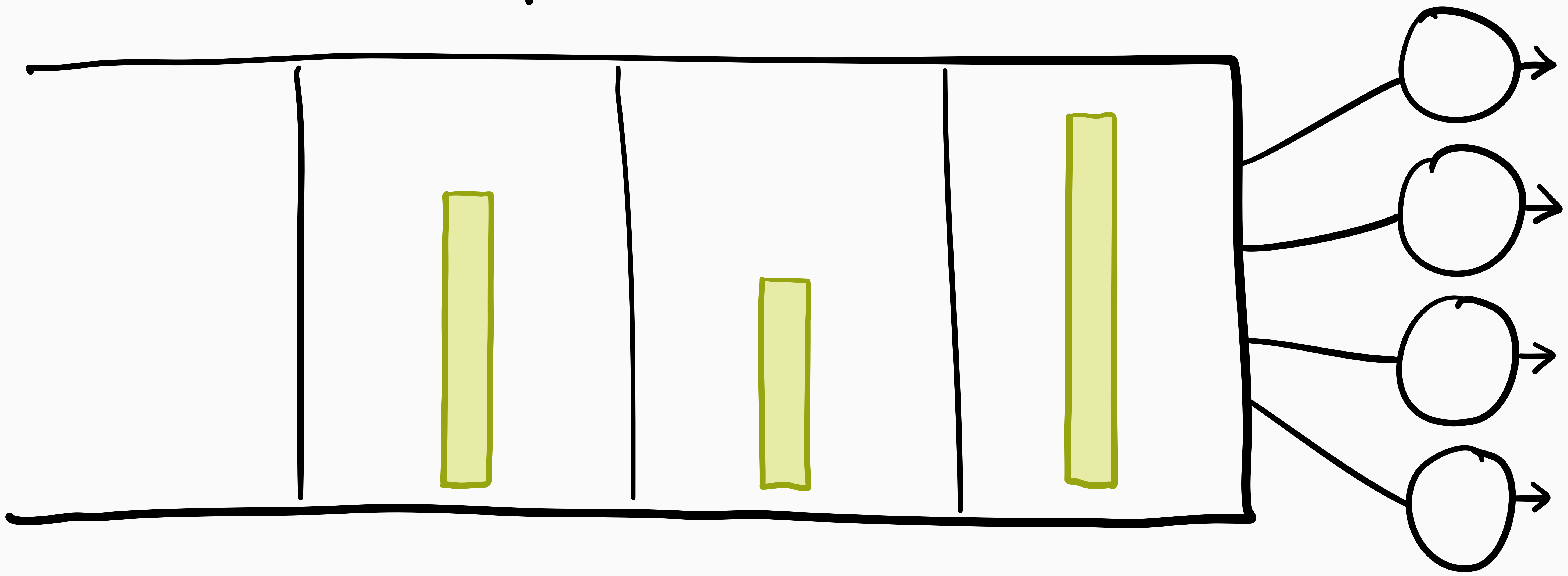


until job completes
or exceeds **rank** r



= **rank**

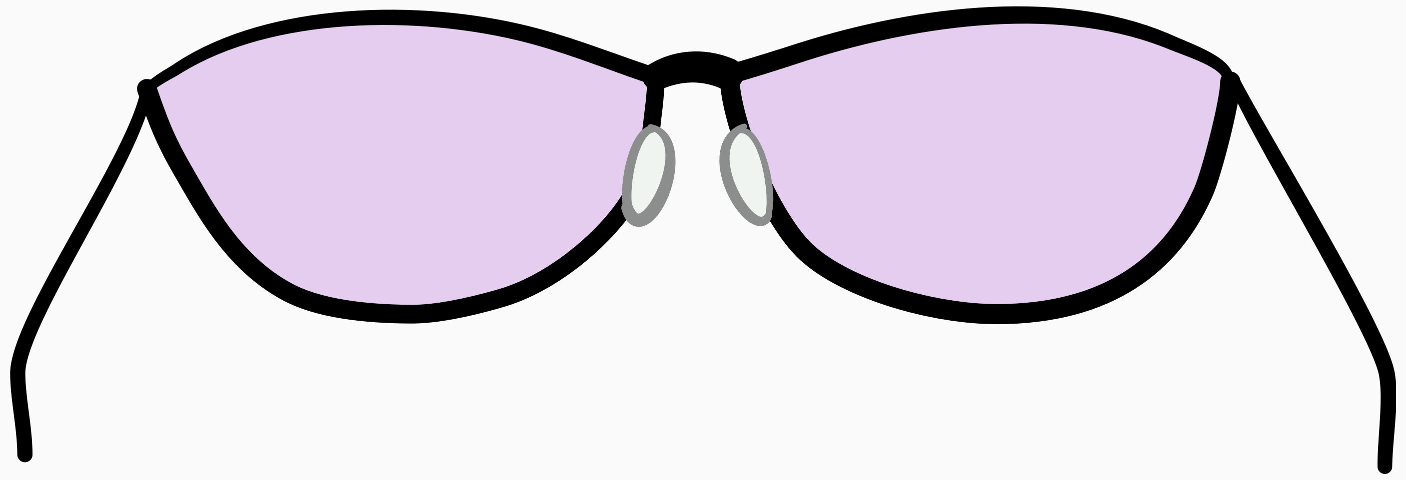
(lower is better)



What is *r*-work? (Gittins)

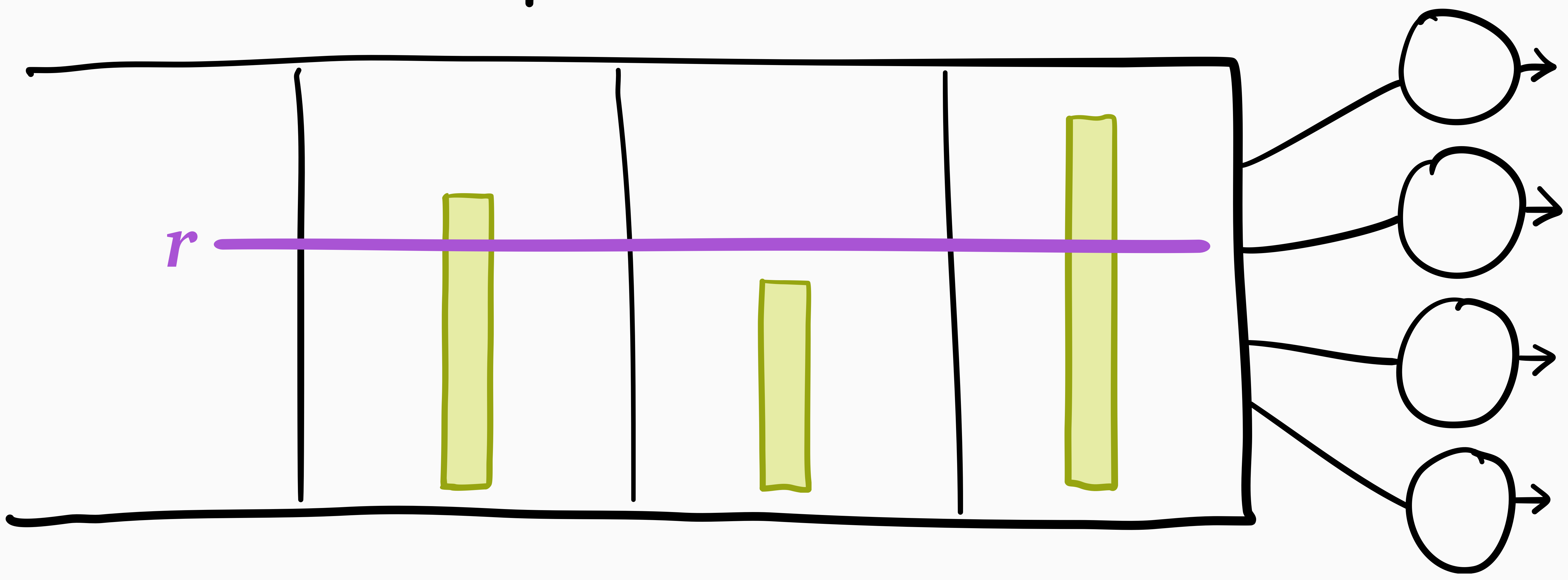
W = work = total remaining size of all jobs

$W(r)$ = r-work = work "relevant" to **rank** r



until job completes
or exceeds **rank** r

= **rank**
(lower is better)

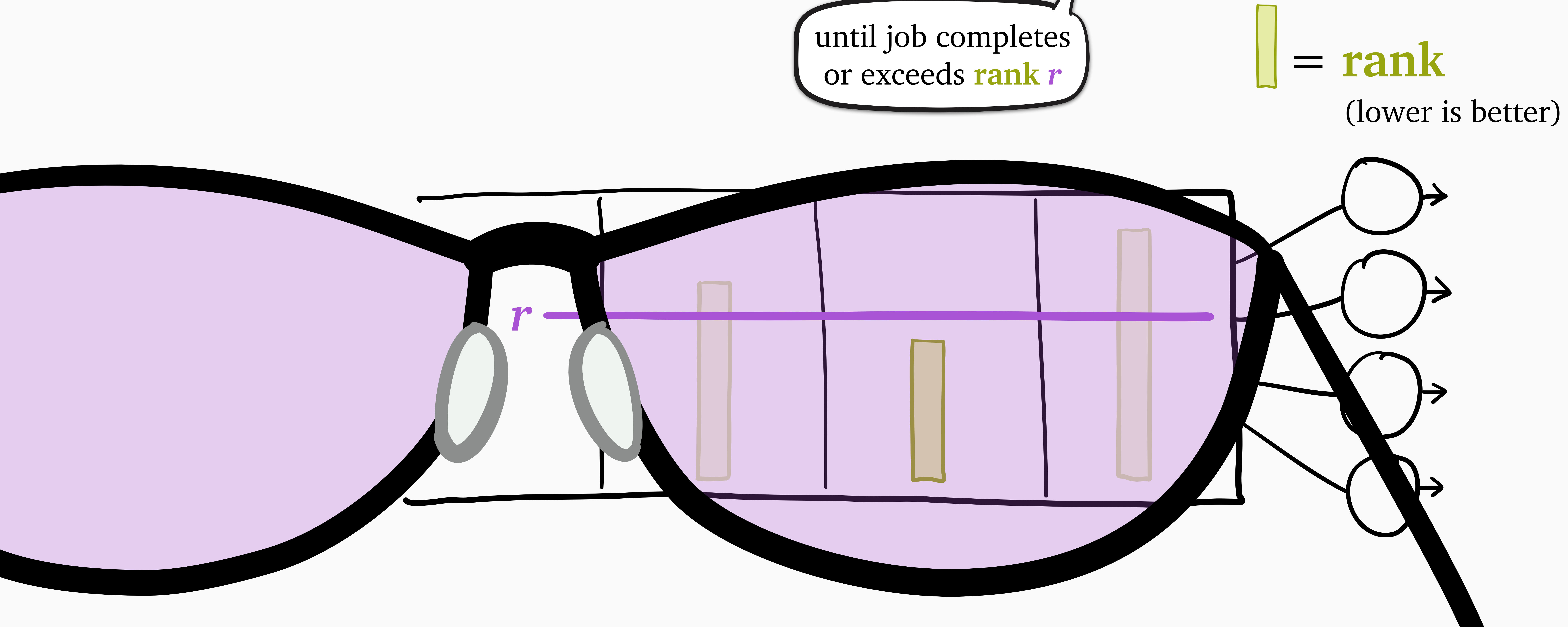


What is *r*-work? (Gittins)

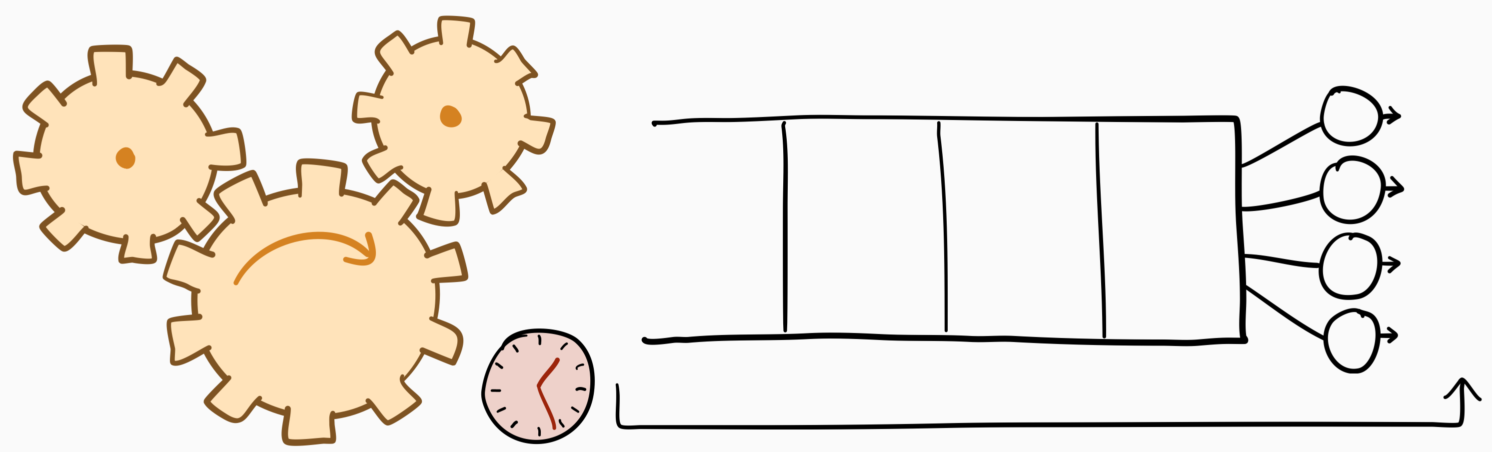
W = work = total remaining size of all jobs

$W(r)$ = r-work = work "relevant" to rank r

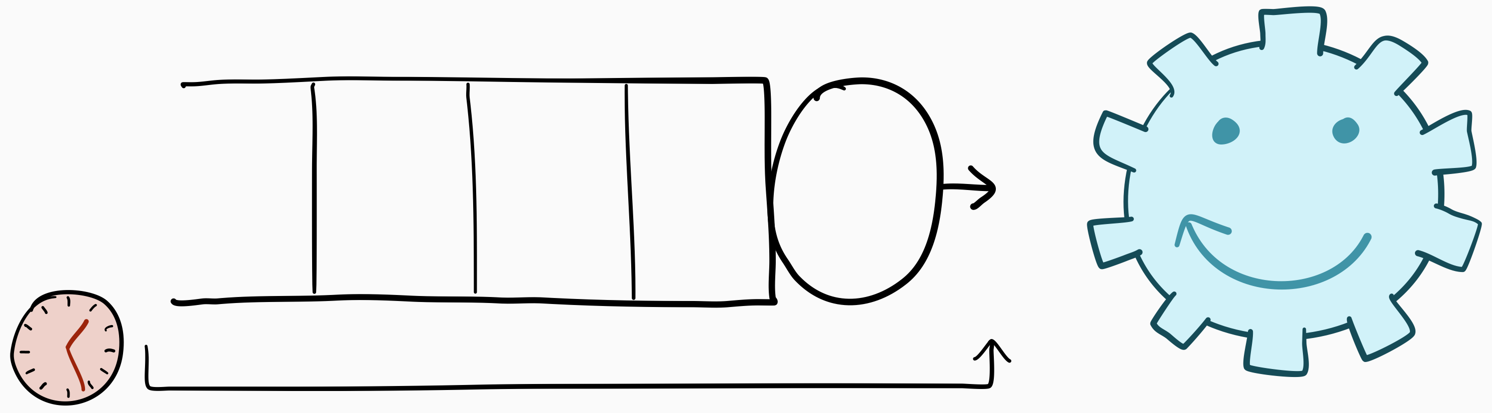
until job completes or exceeds rank r



Response time via *r*-work

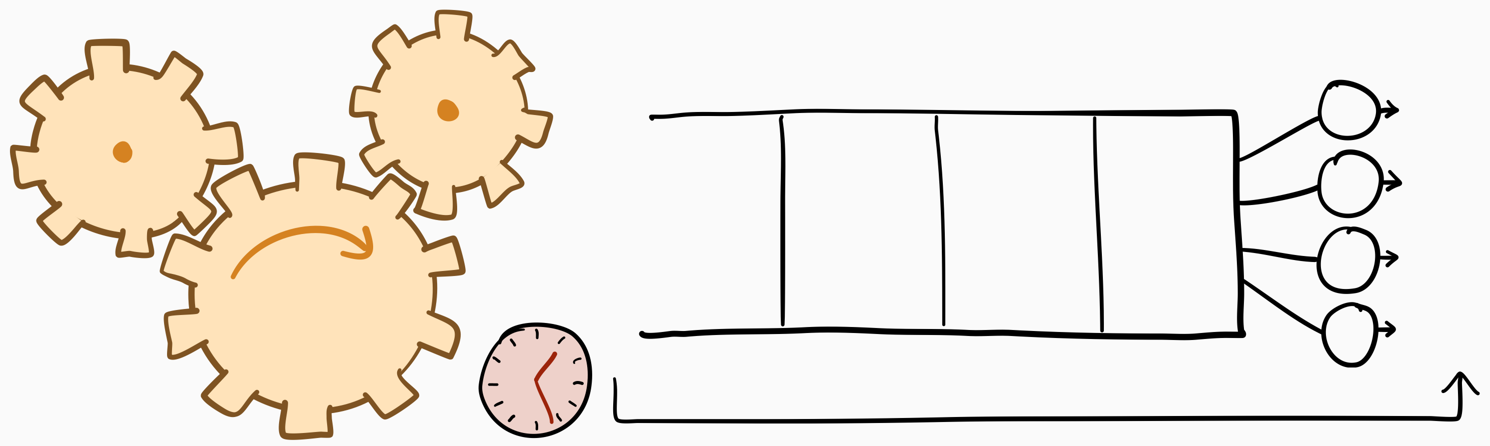


mean response
time in **M/G/*k***

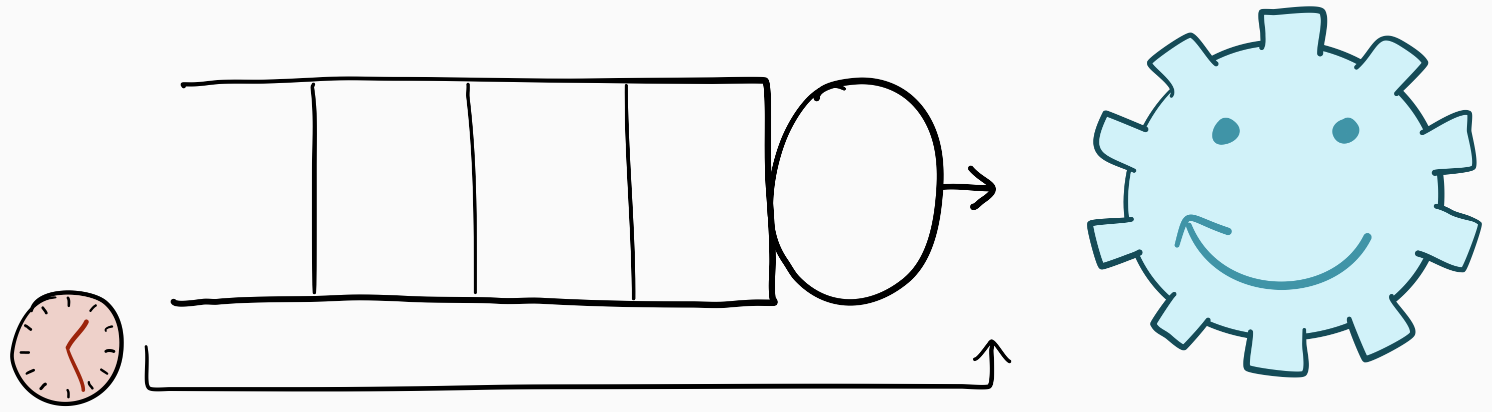


mean response
time in **M/G/1**

Response time via *r*-work

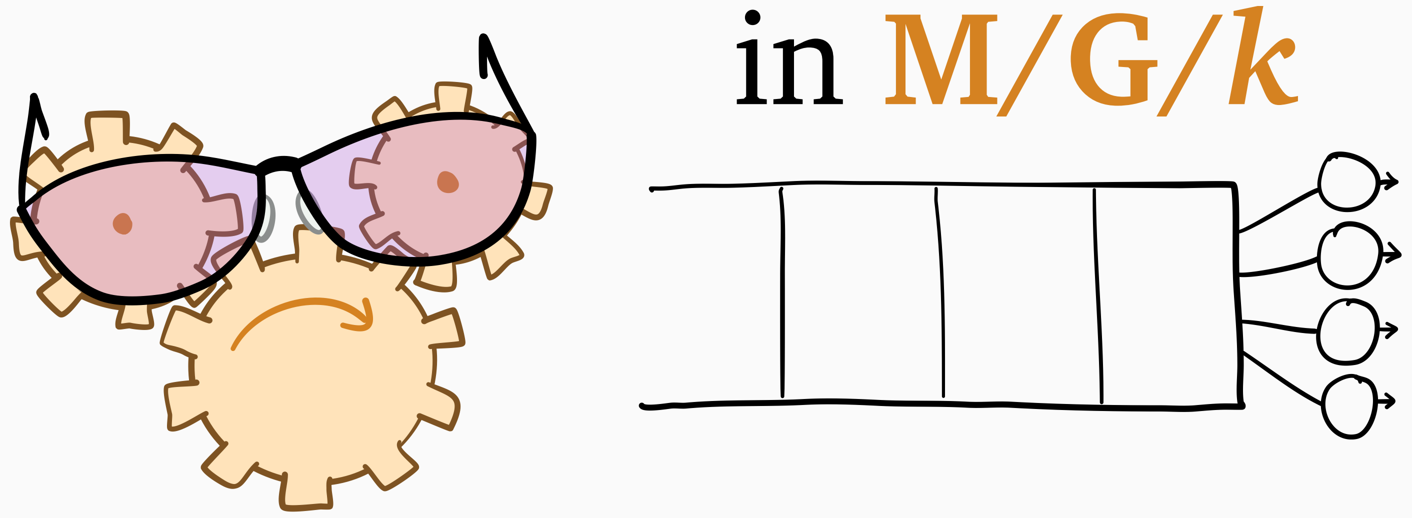


mean response time in **M/G/*k***

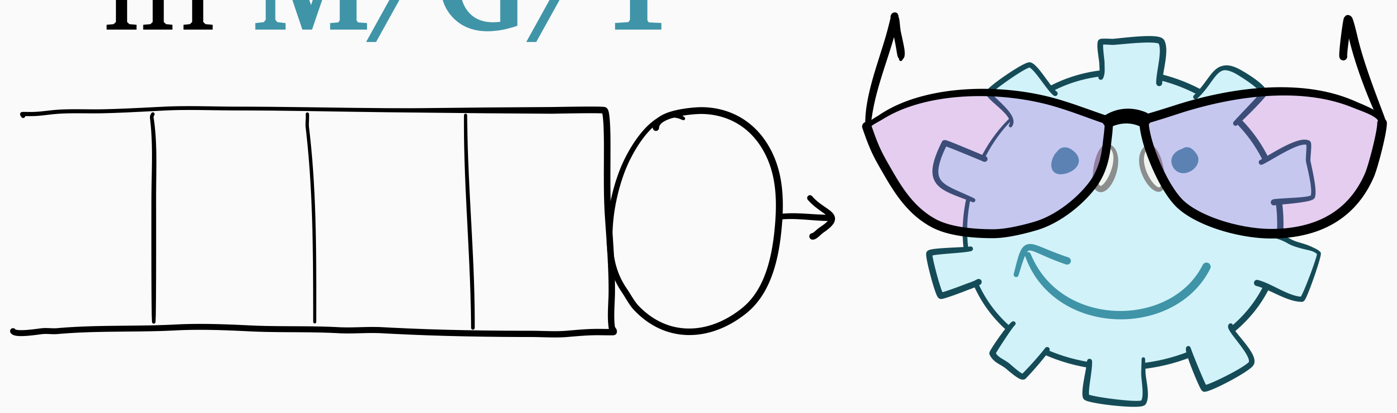


mean response time in **M/G/1**

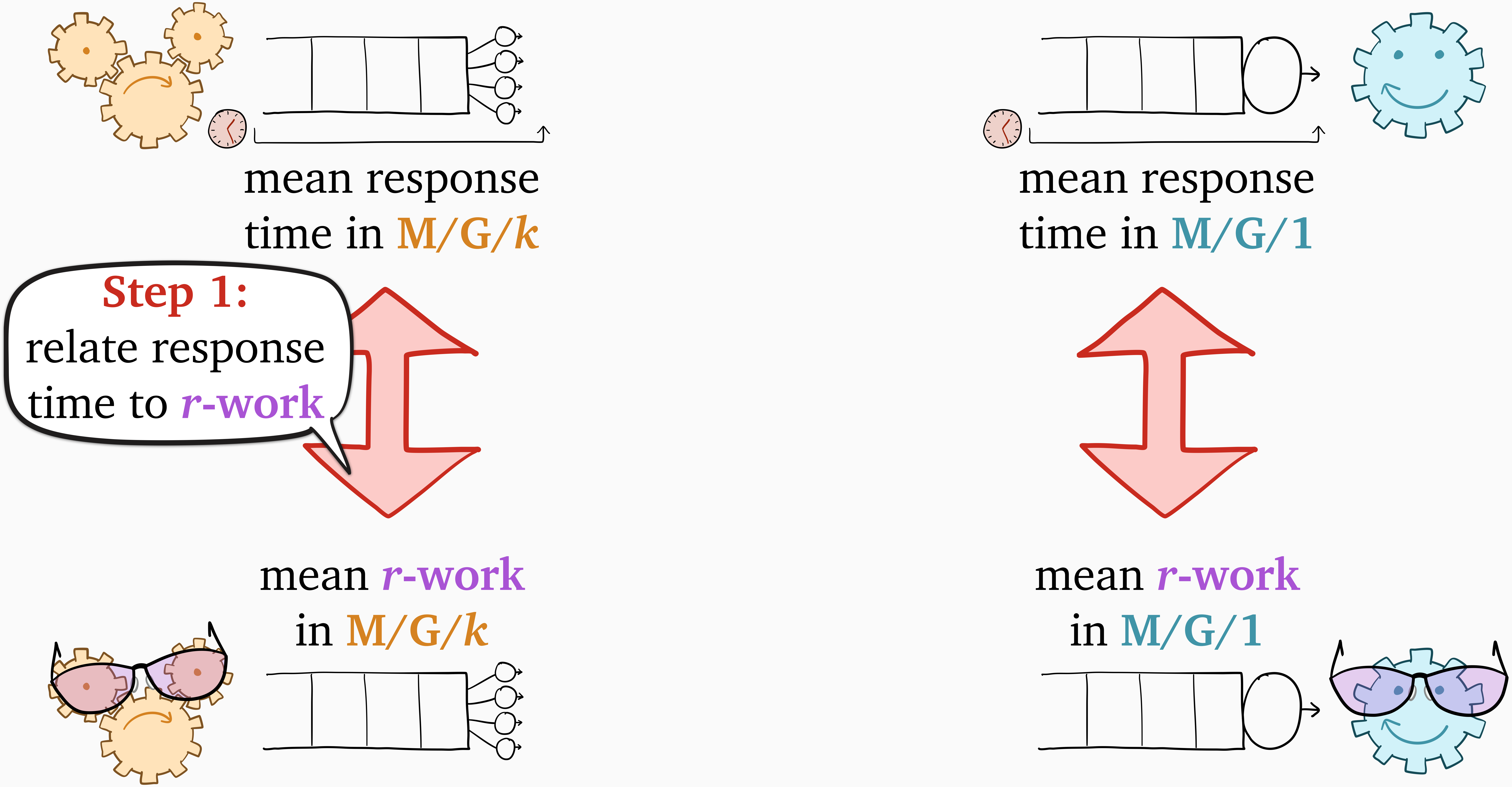
mean ***r*-work** in **M/G/*k***



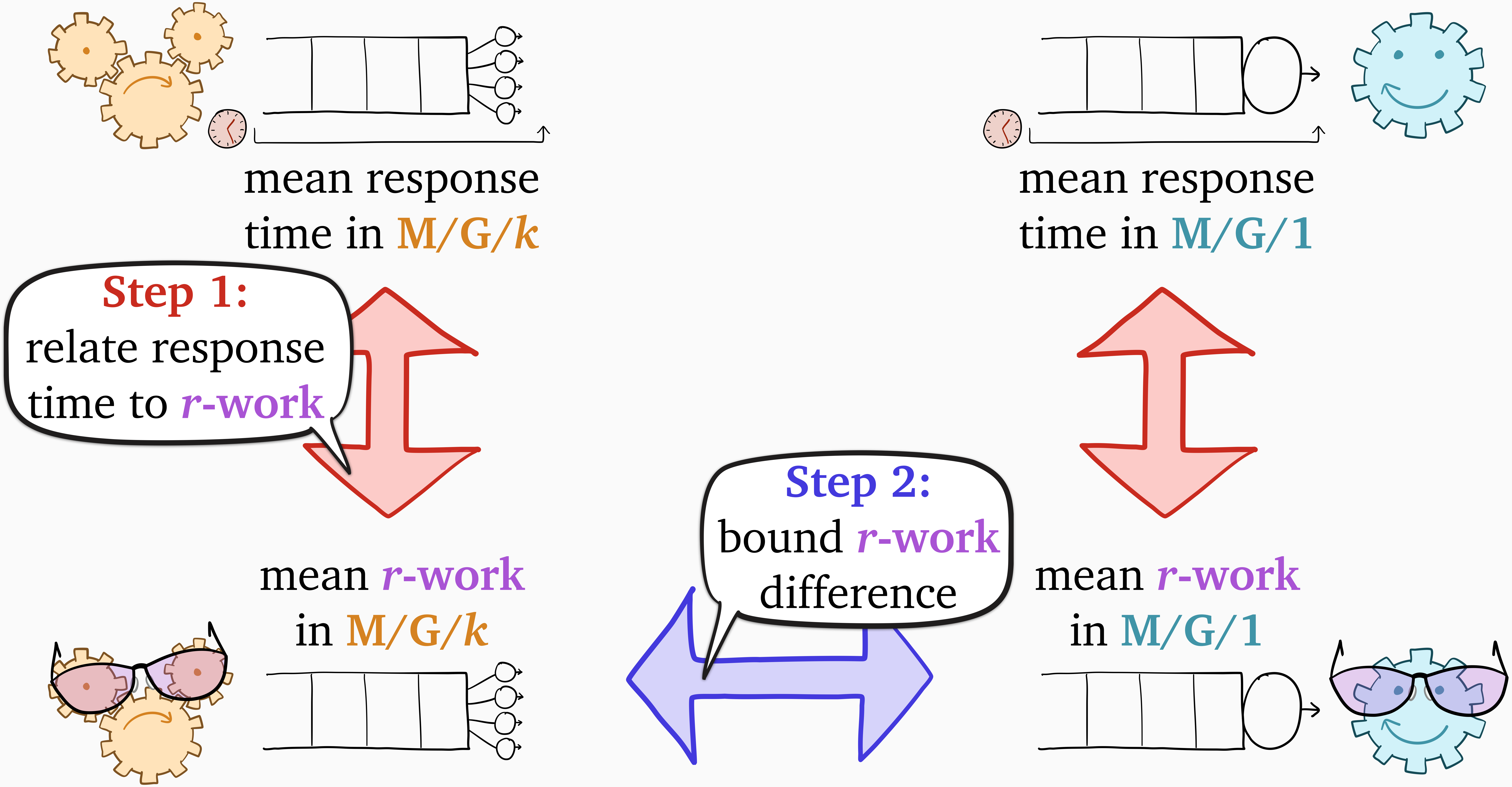
mean ***r*-work** in **M/G/1**



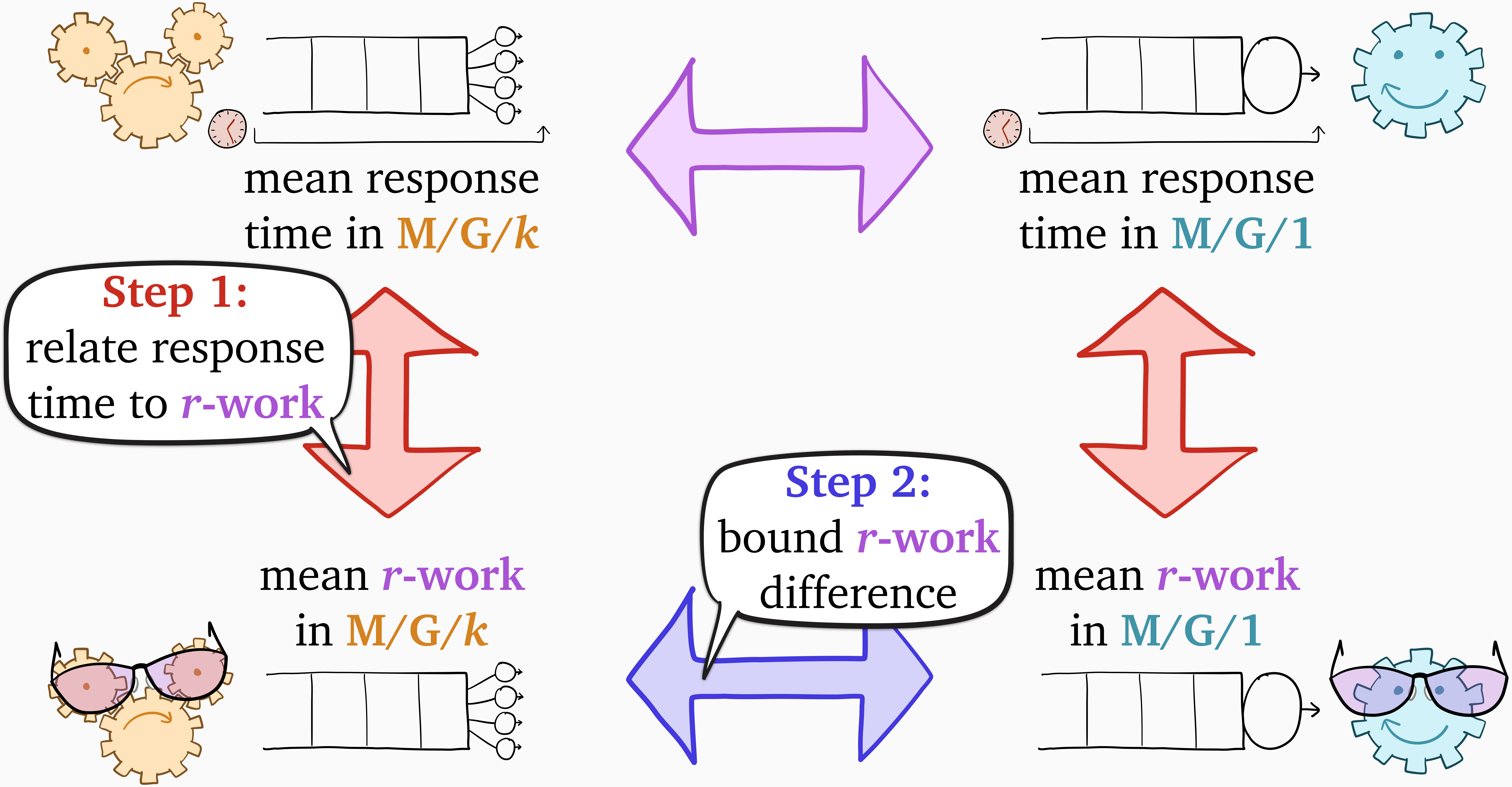
Response time via *r*-work



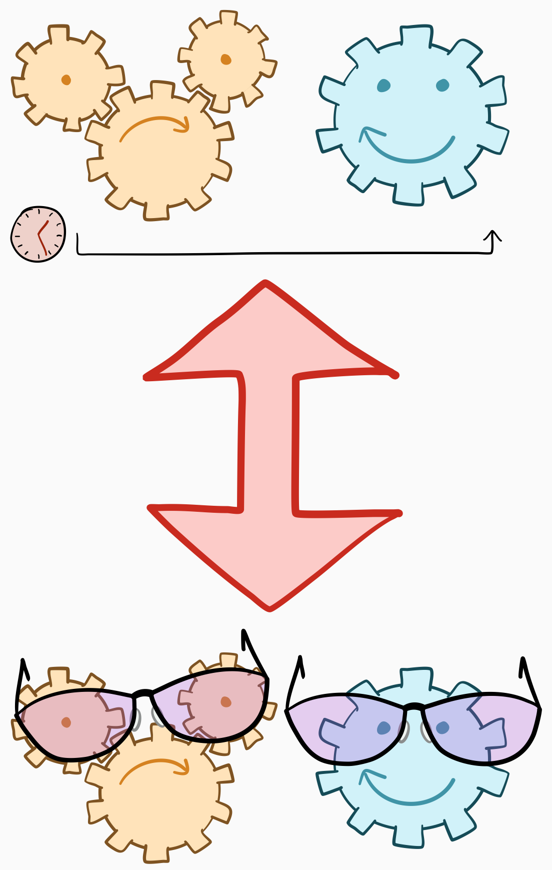
Response time via *r*-work



Response time via *r*-work

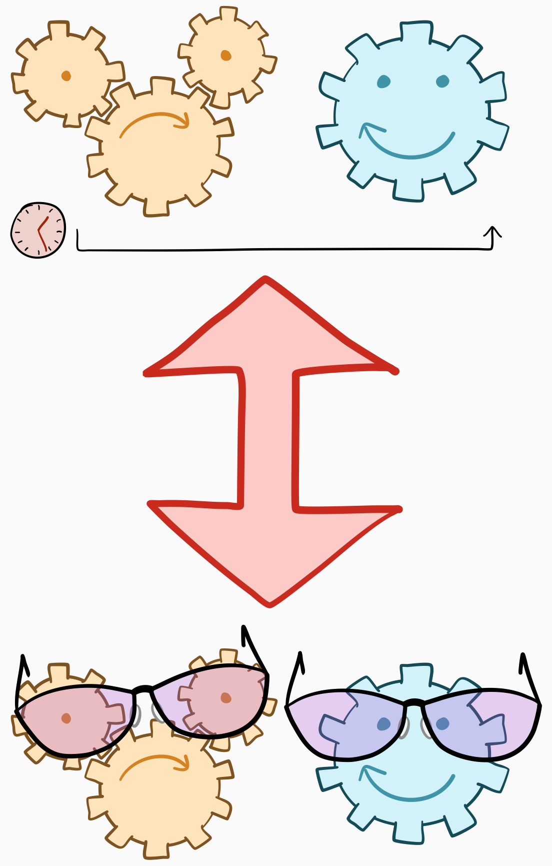


Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)



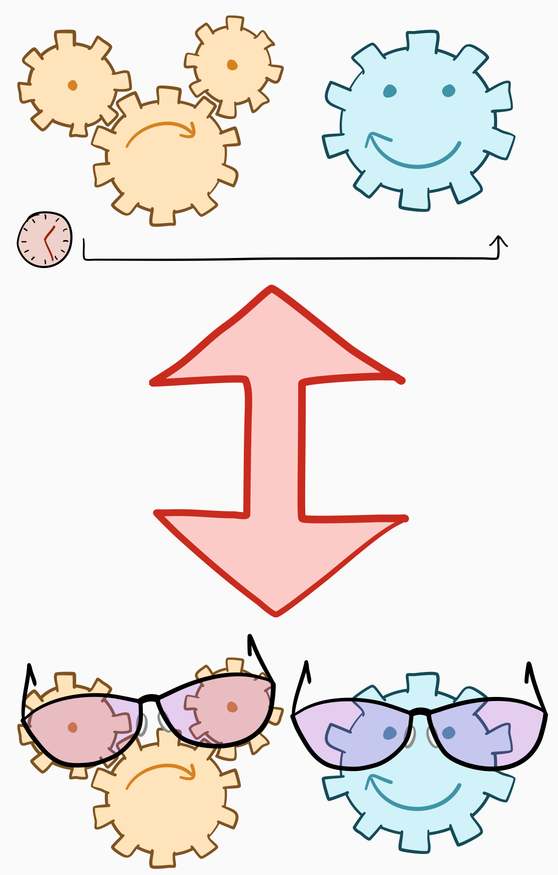
Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)

Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$$



Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)

Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \frac{1}{\lambda}\int_0^\infty \mathbf{E}[W(r)]\,\mathrm{d}(1/r)$$

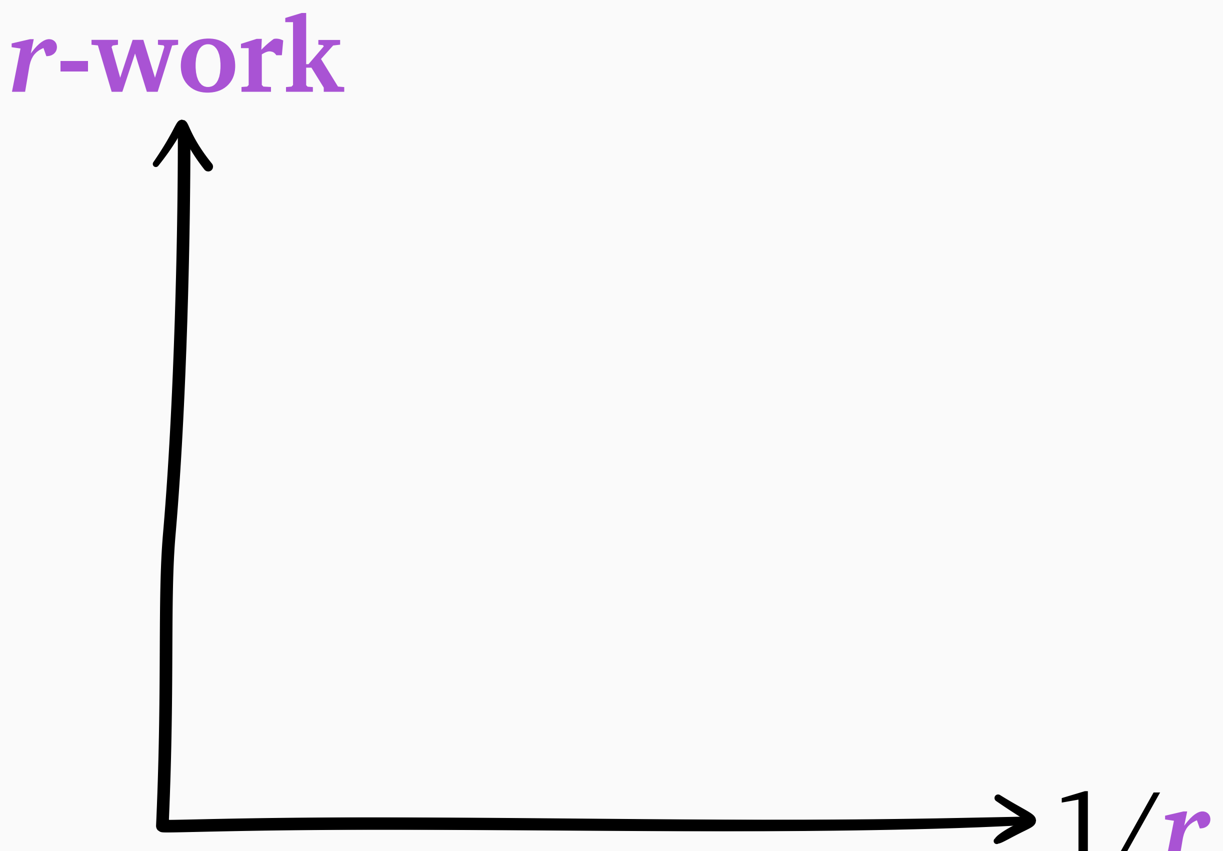


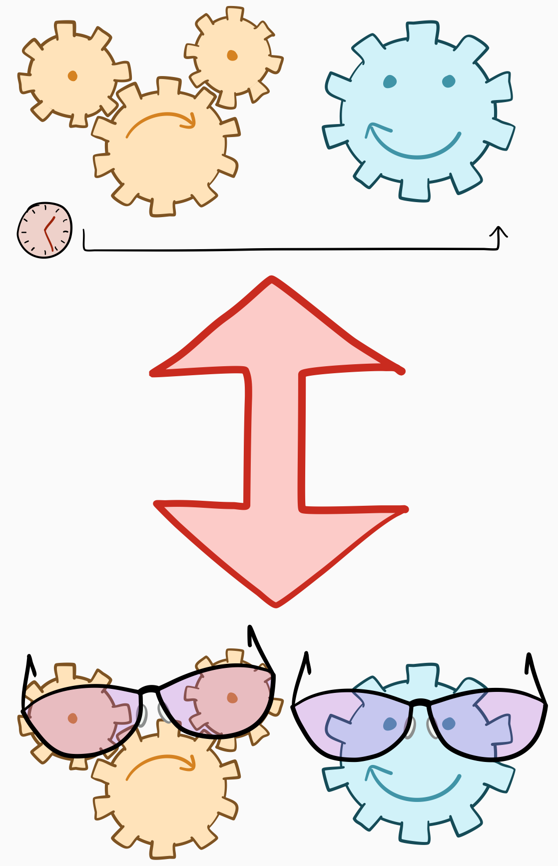
Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)

Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \frac{1}{\lambda}\int_0^\infty \mathbf{E}[W(r)]\,\mathrm{d}(1/r)$$

Proof:

One job's ***r*-work**:





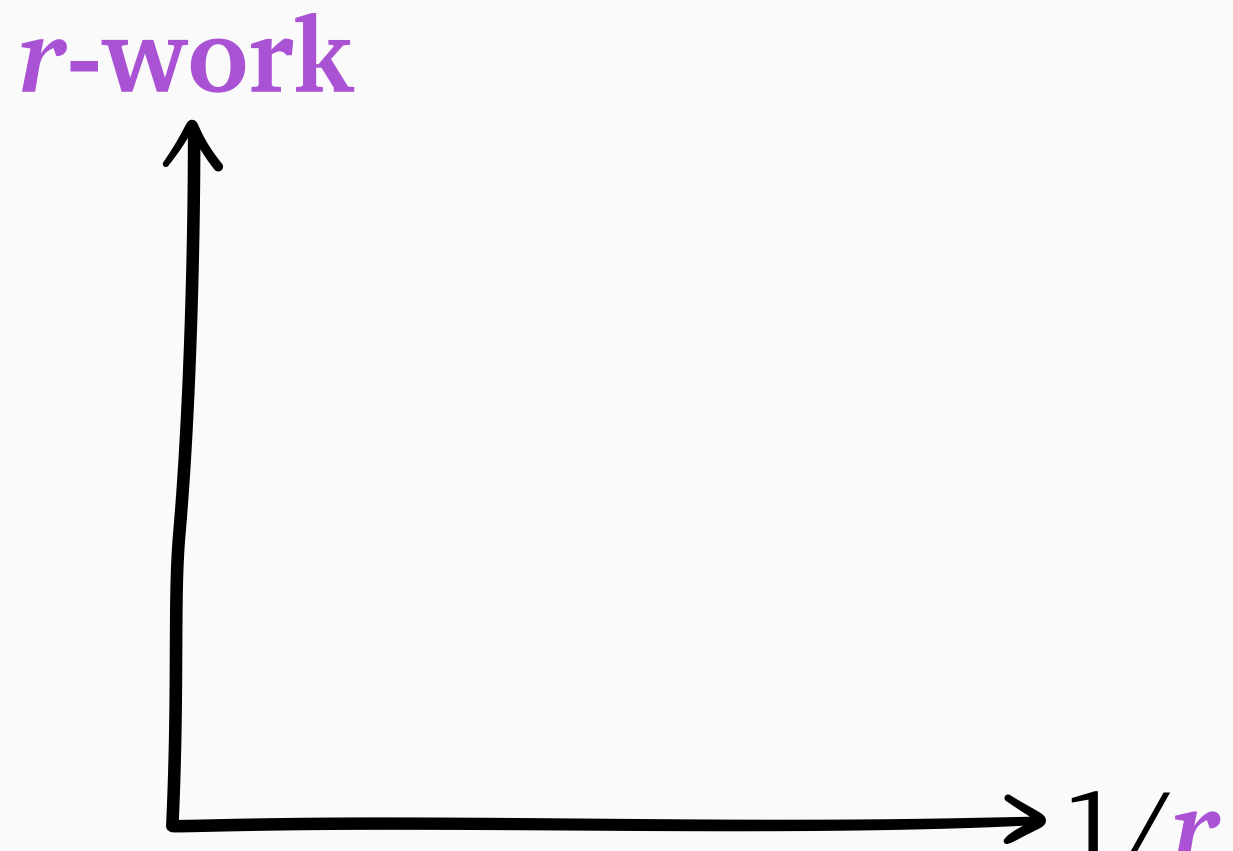
Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)

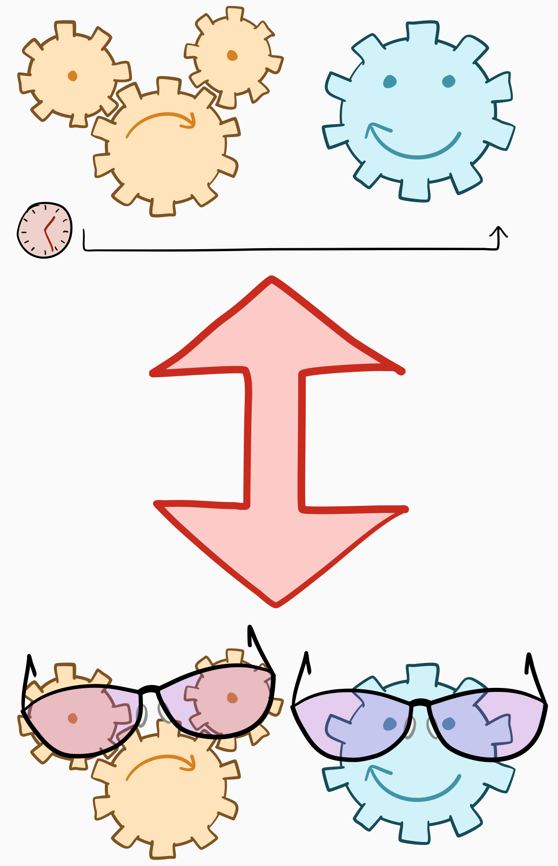
Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \frac{1}{\lambda}\int_0^\infty \mathbf{E}[W(r)]\,\mathrm{d}(1/r)$$

Proof:

remaining size x

One job's ***r*-work**:





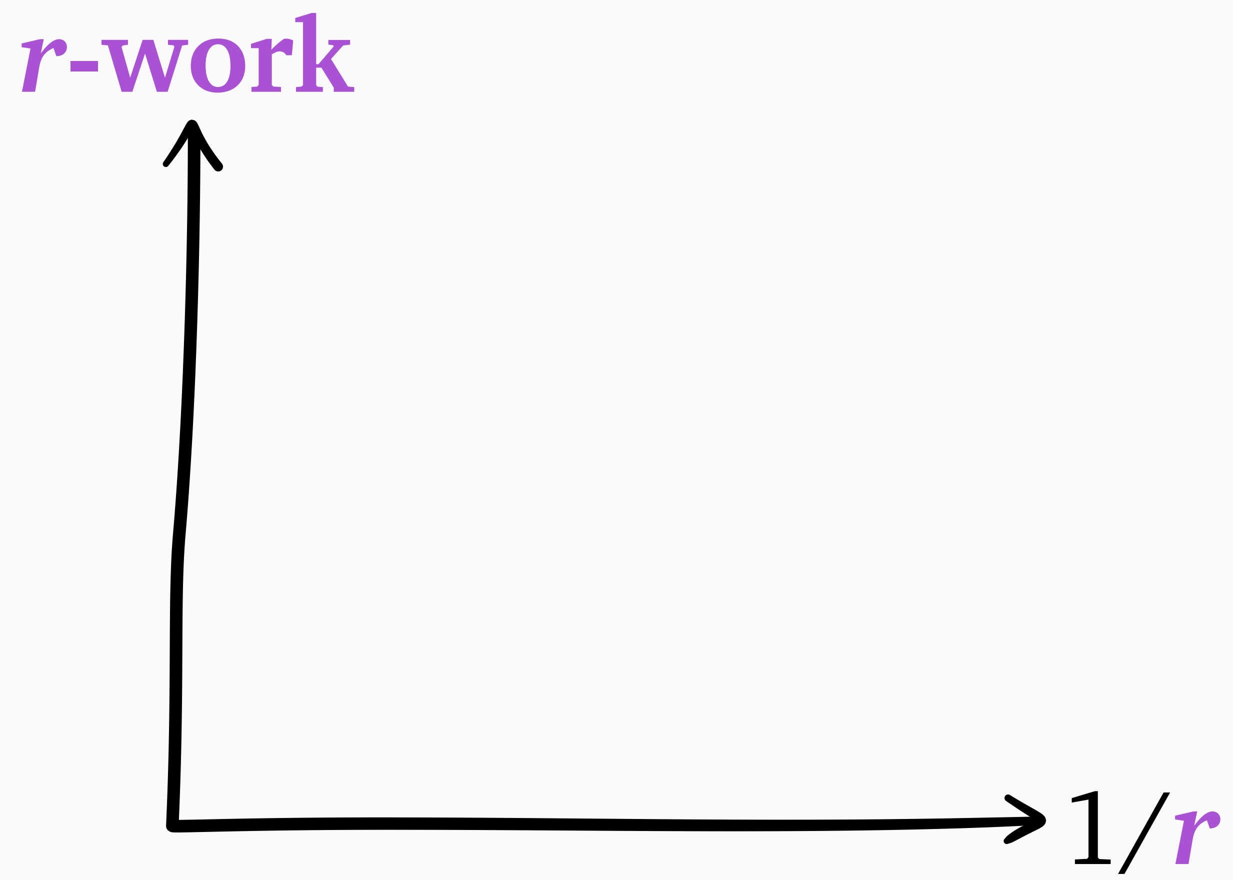
Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)

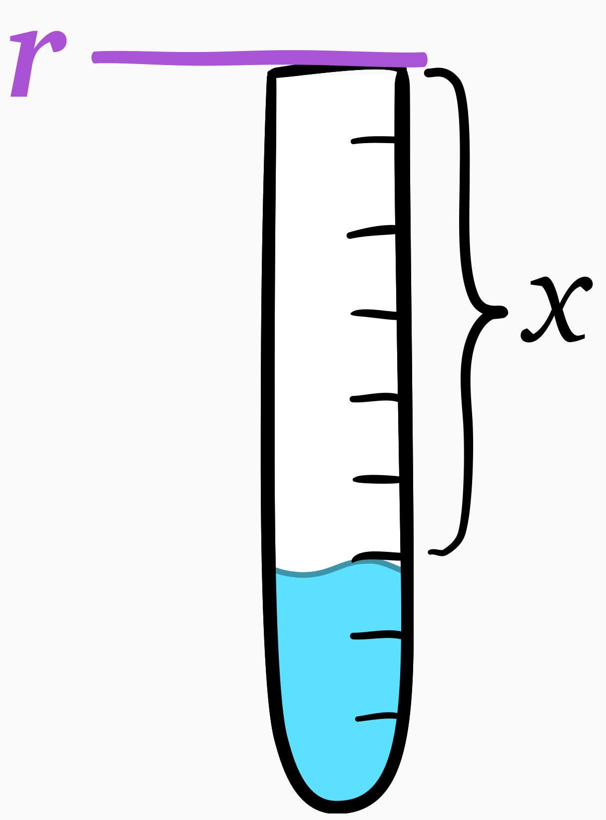
Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \frac{1}{\lambda}\int_0^\infty \mathbf{E}[W(r)]\,\mathrm{d}(1/r)$$

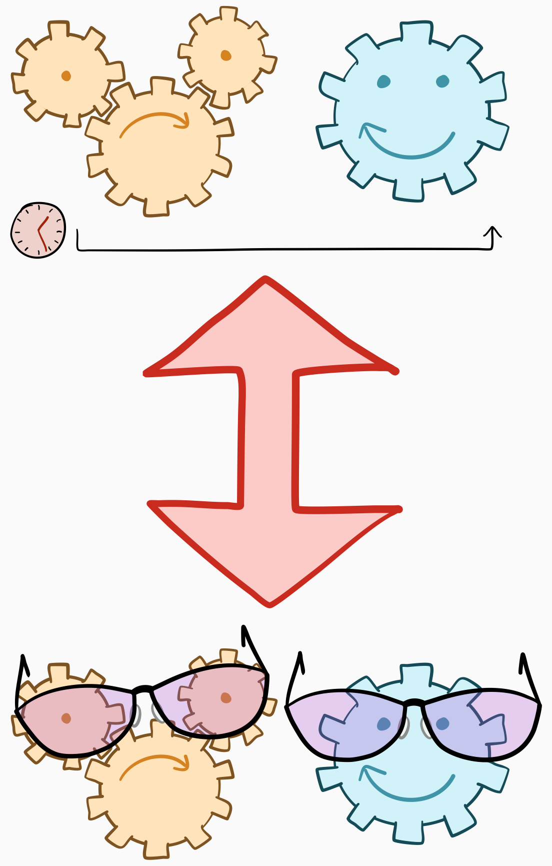
Proof:

remaining size x

One job's ***r*-work**:







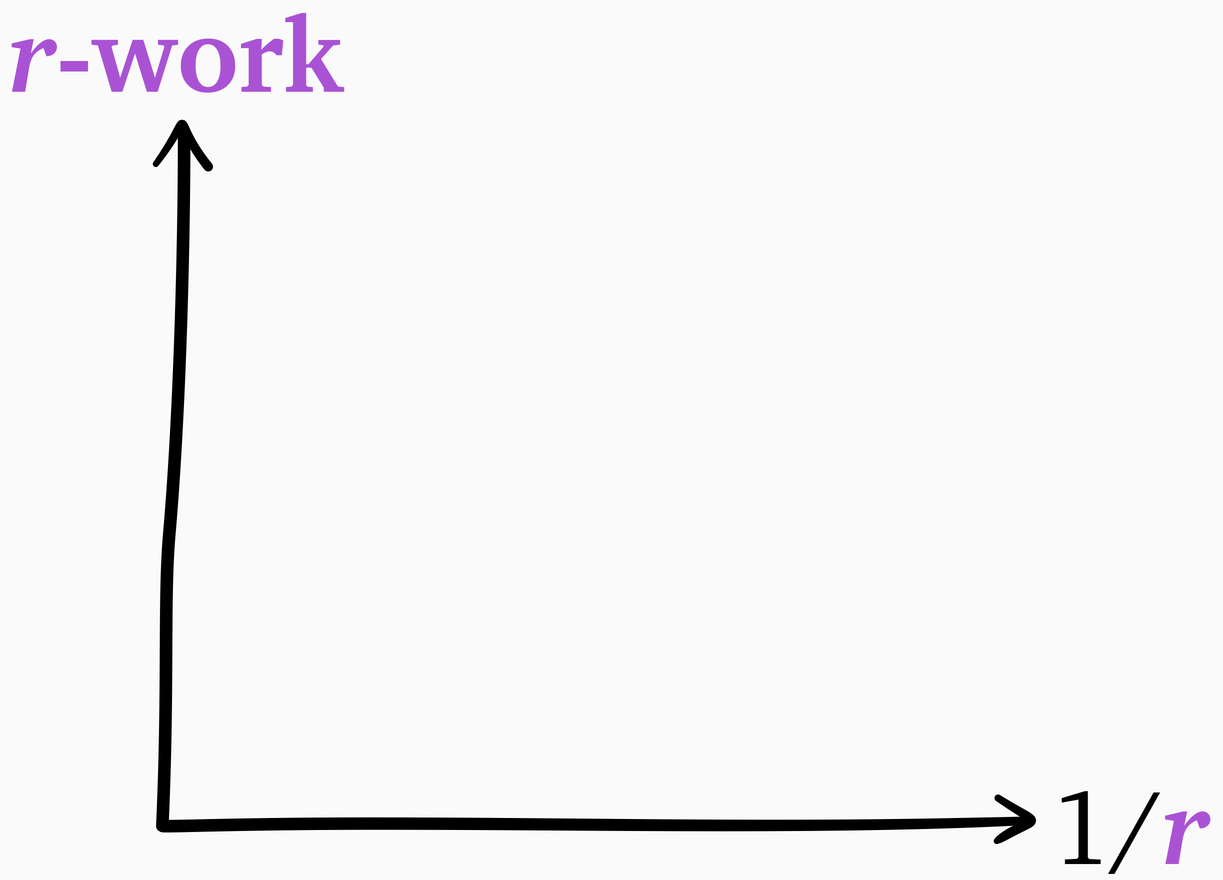
Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)

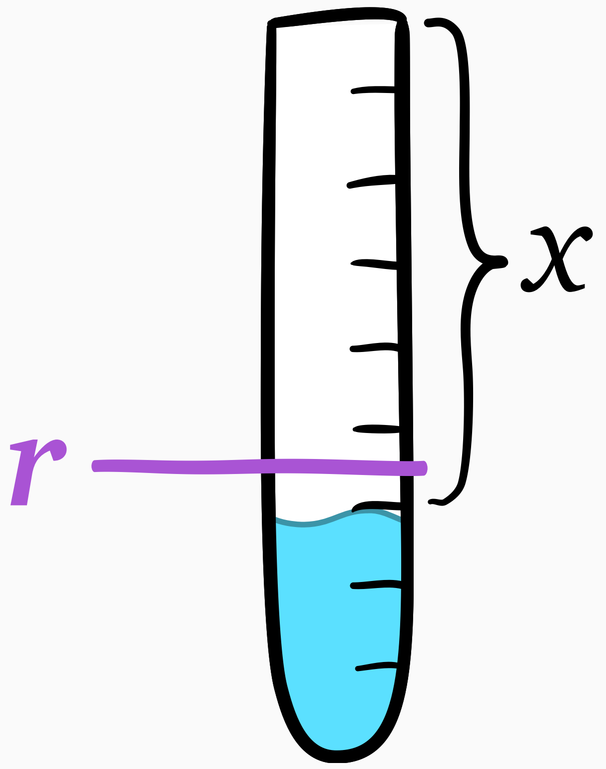
Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \frac{1}{\lambda}\int_0^\infty \mathbf{E}[W(r)]\,\mathrm{d}(1/r)$$

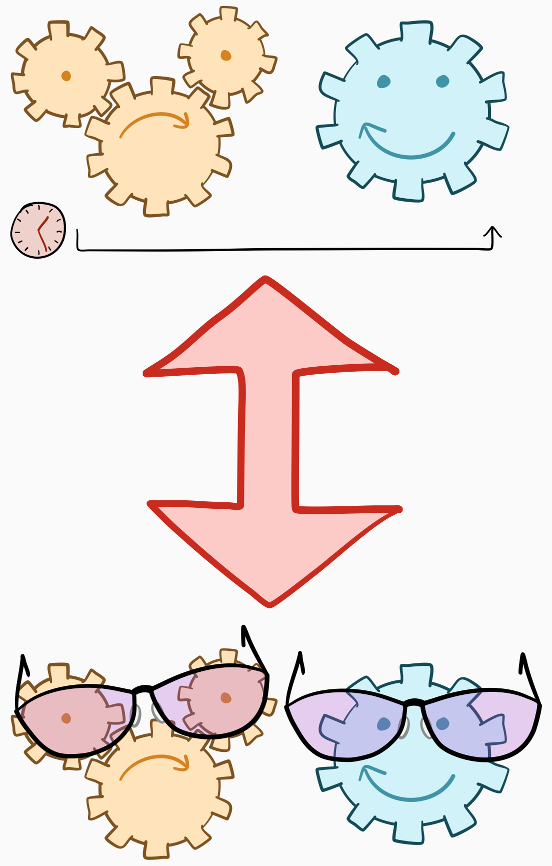
Proof:

remaining size x

One job's ***r*-work**:



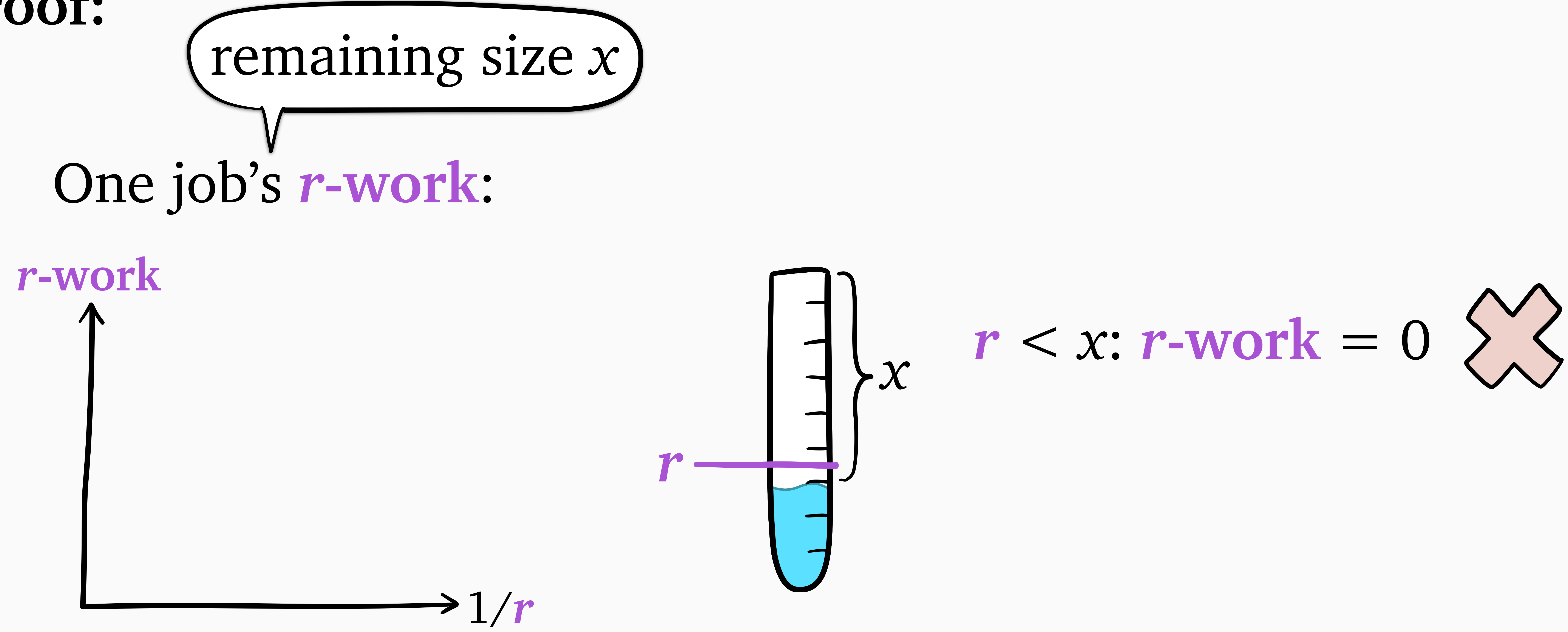


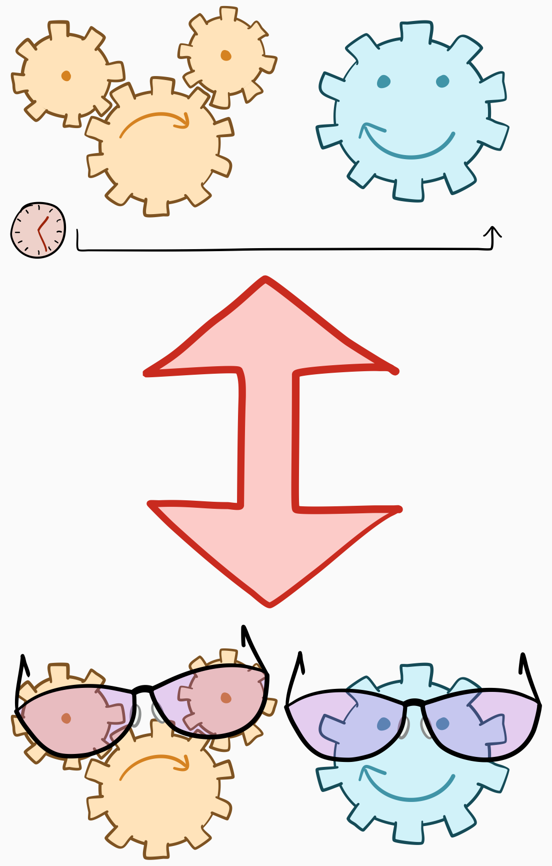


Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)

Theorem: $\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \frac{1}{\lambda}\int_0^\infty \mathbf{E}[W(r)]\,\mathrm{d}(1/r)$

Proof:





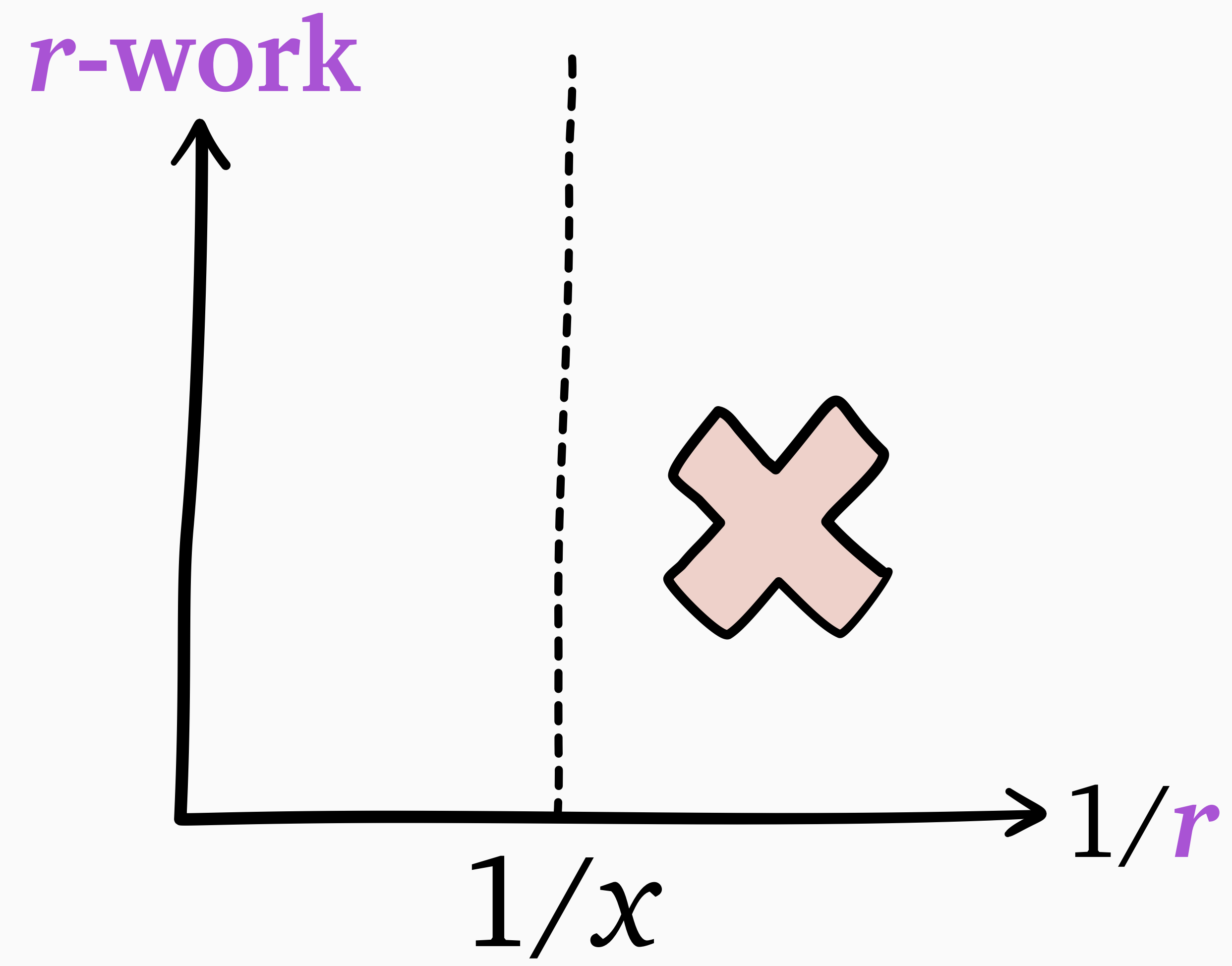
Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)

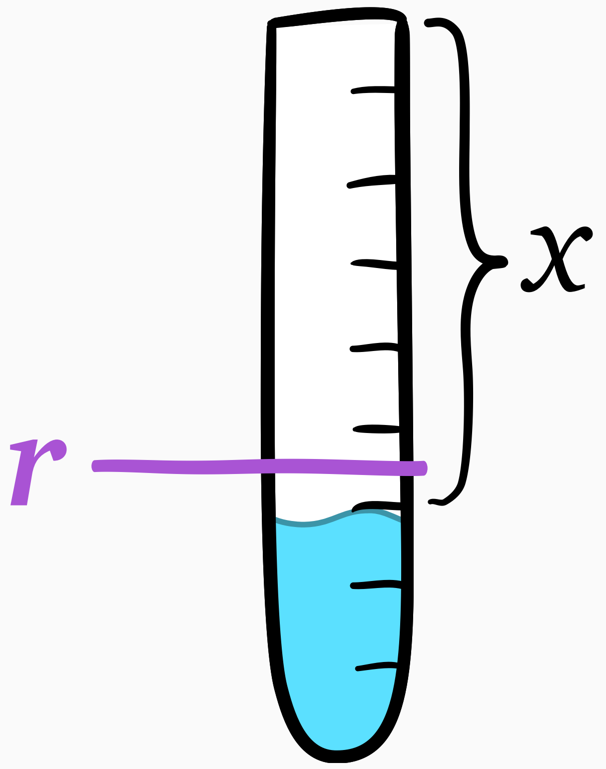
Theorem: $\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \frac{1}{\lambda}\int_0^\infty \mathbf{E}[W(r)]\,\mathrm{d}(1/r)$

Proof:

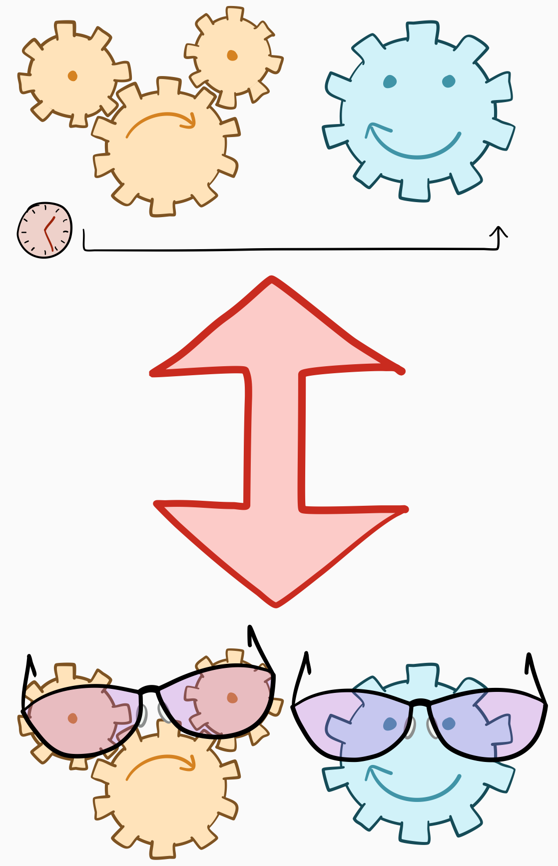
remaining size x

One job's ***r*-work**:





$r < x$: r-work $= 0$



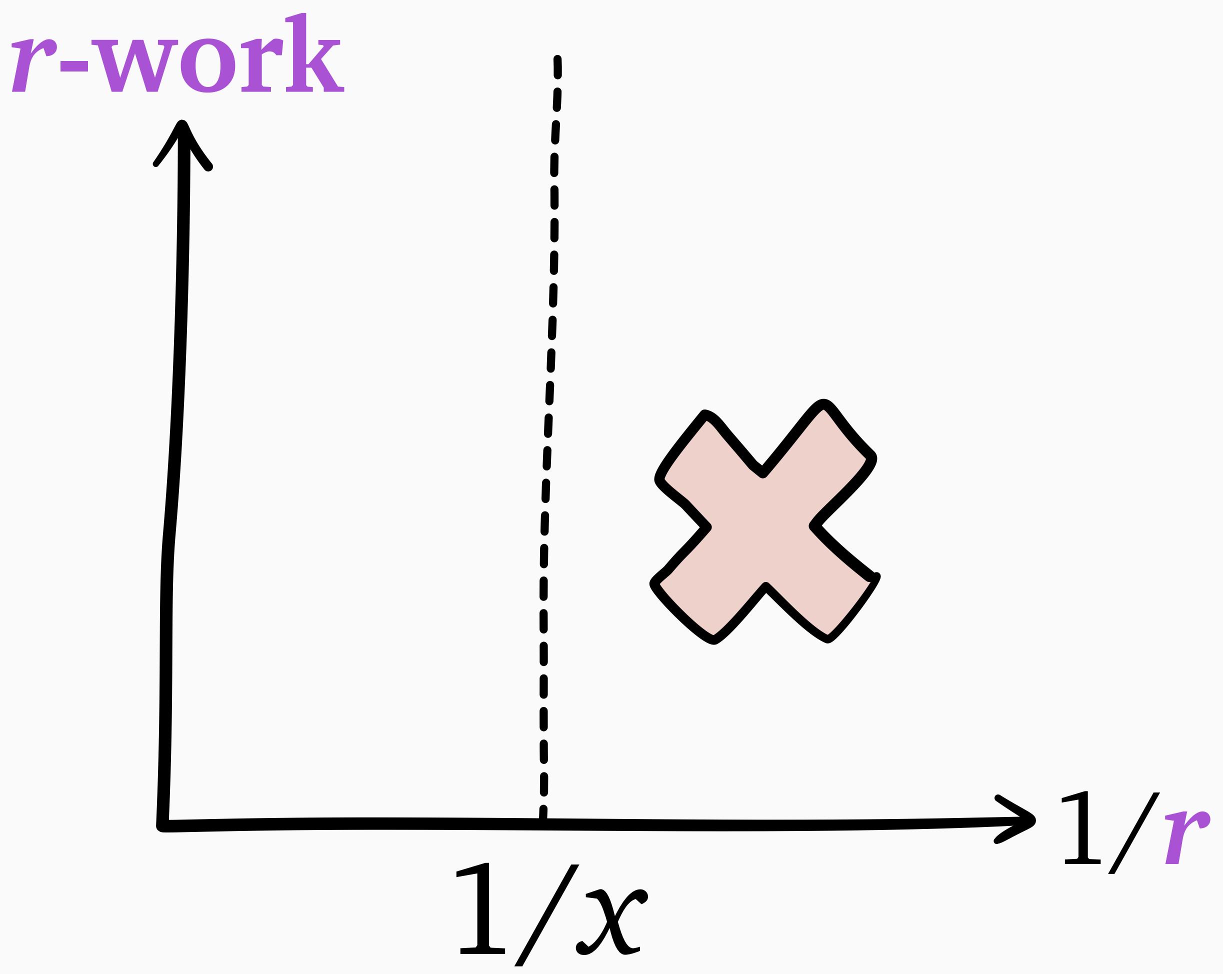
Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)

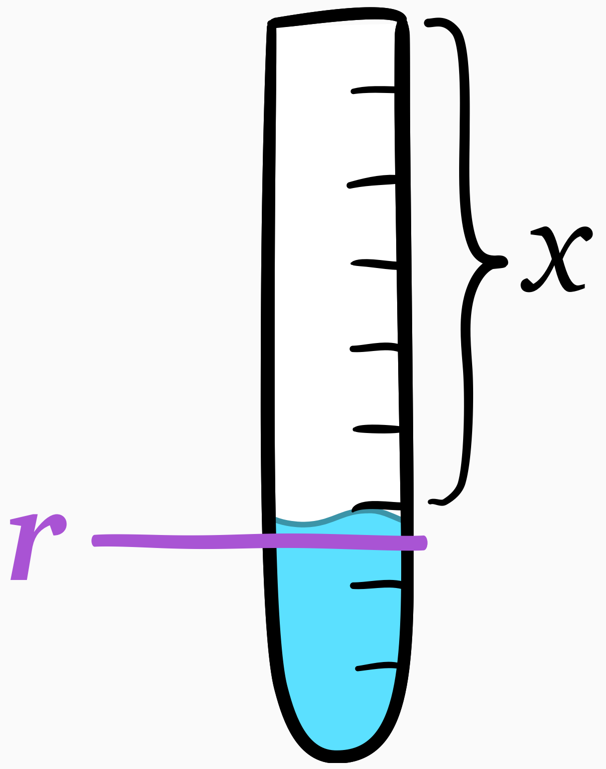
Theorem: $\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \frac{1}{\lambda}\int_0^\infty \mathbf{E}[W(r)]\,\mathrm{d}(1/r)$

Proof:

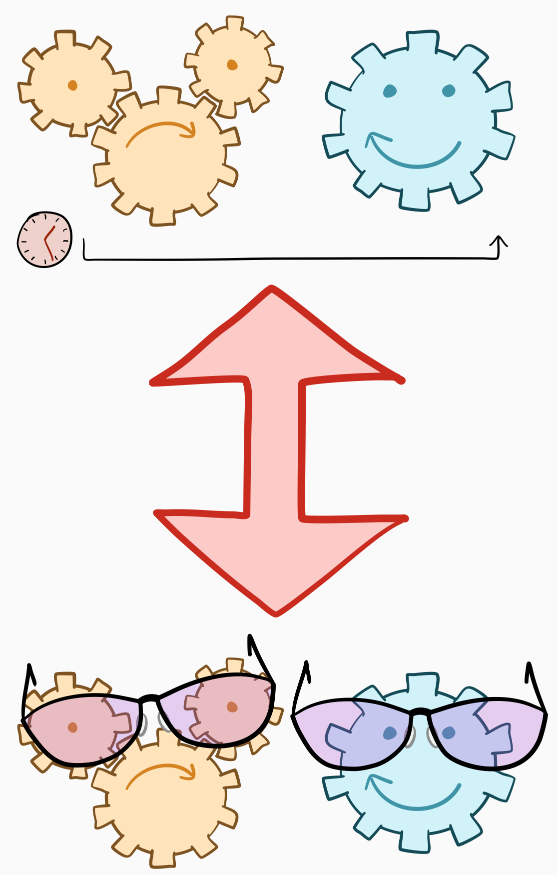
remaining size x

One job's **r-work**:





$r < x$: r-work $= 0$



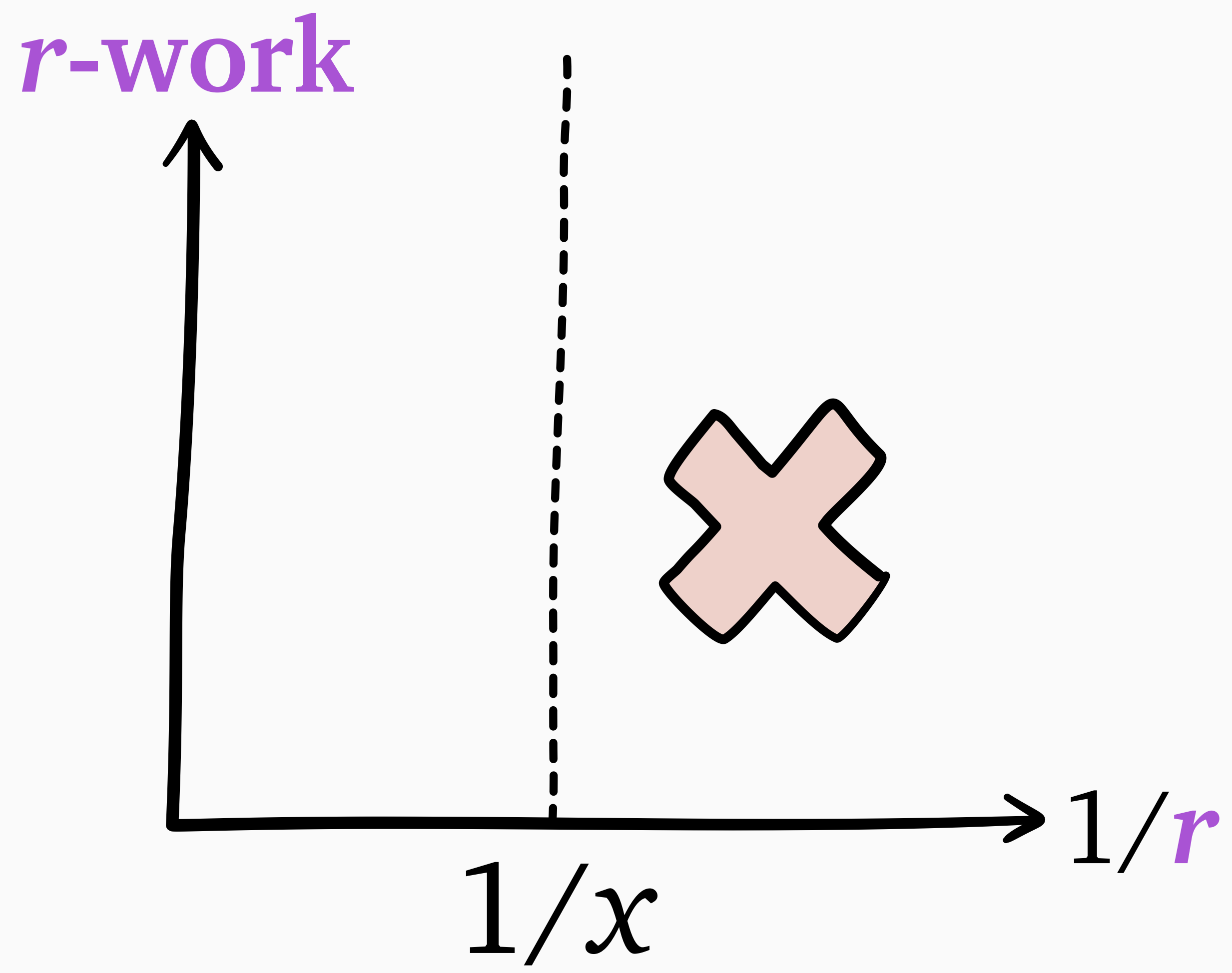
Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)

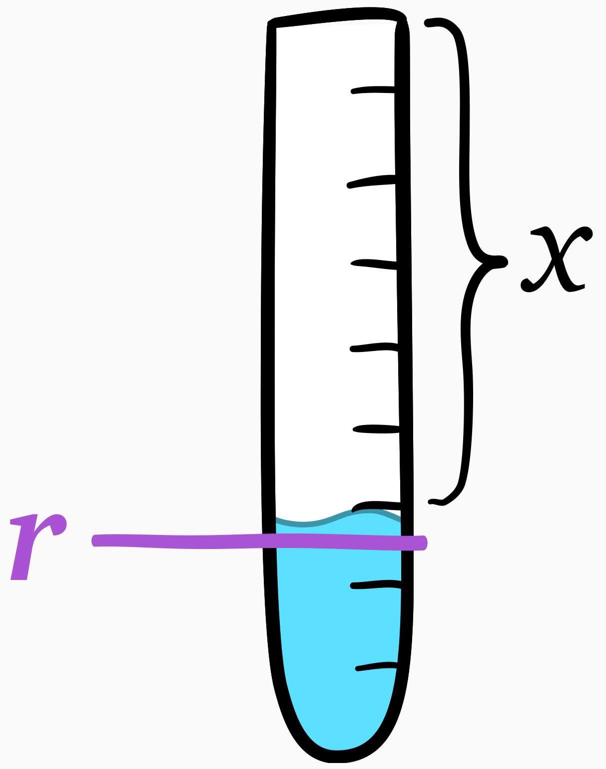
Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \frac{1}{\lambda}\int_0^\infty \mathbf{E}[W(r)]\,\mathrm{d}(1/r)$$

Proof:

remaining size x

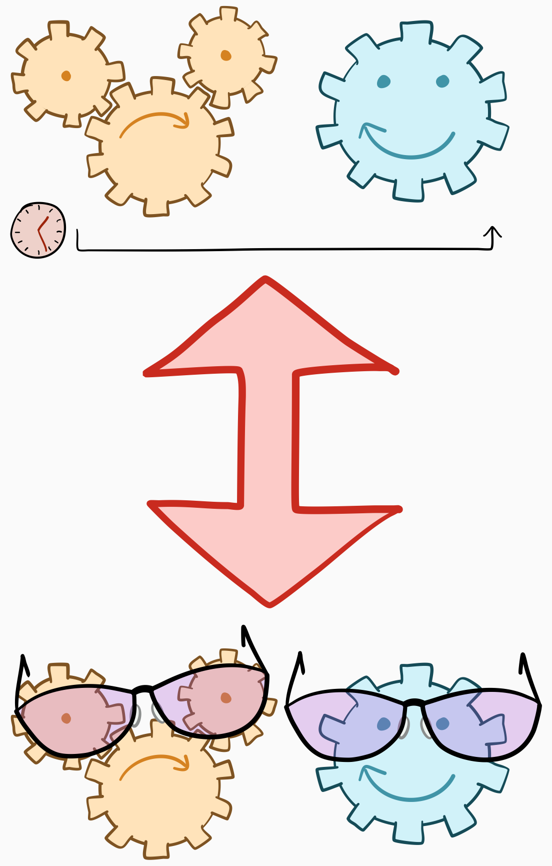
One job's ***r*-work**:





$r < x$: r-work $= 0$

$r \geq x$: r-work $= x$



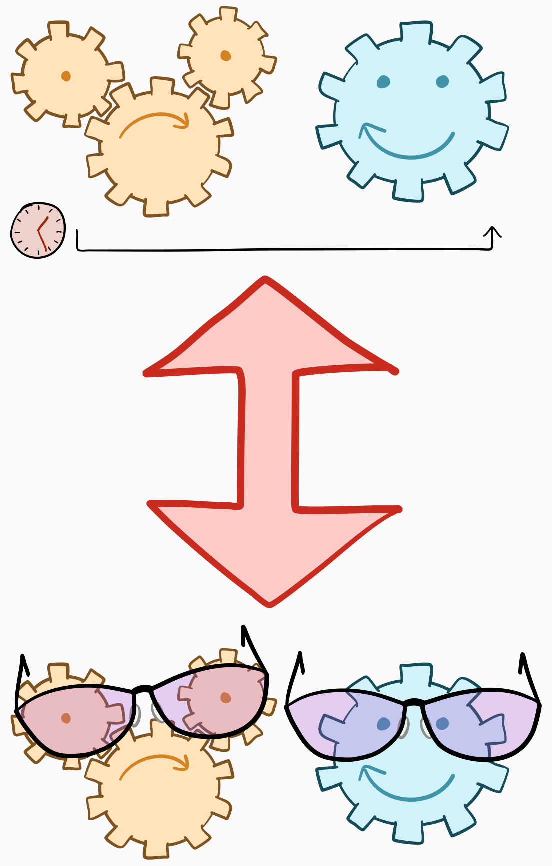
Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)

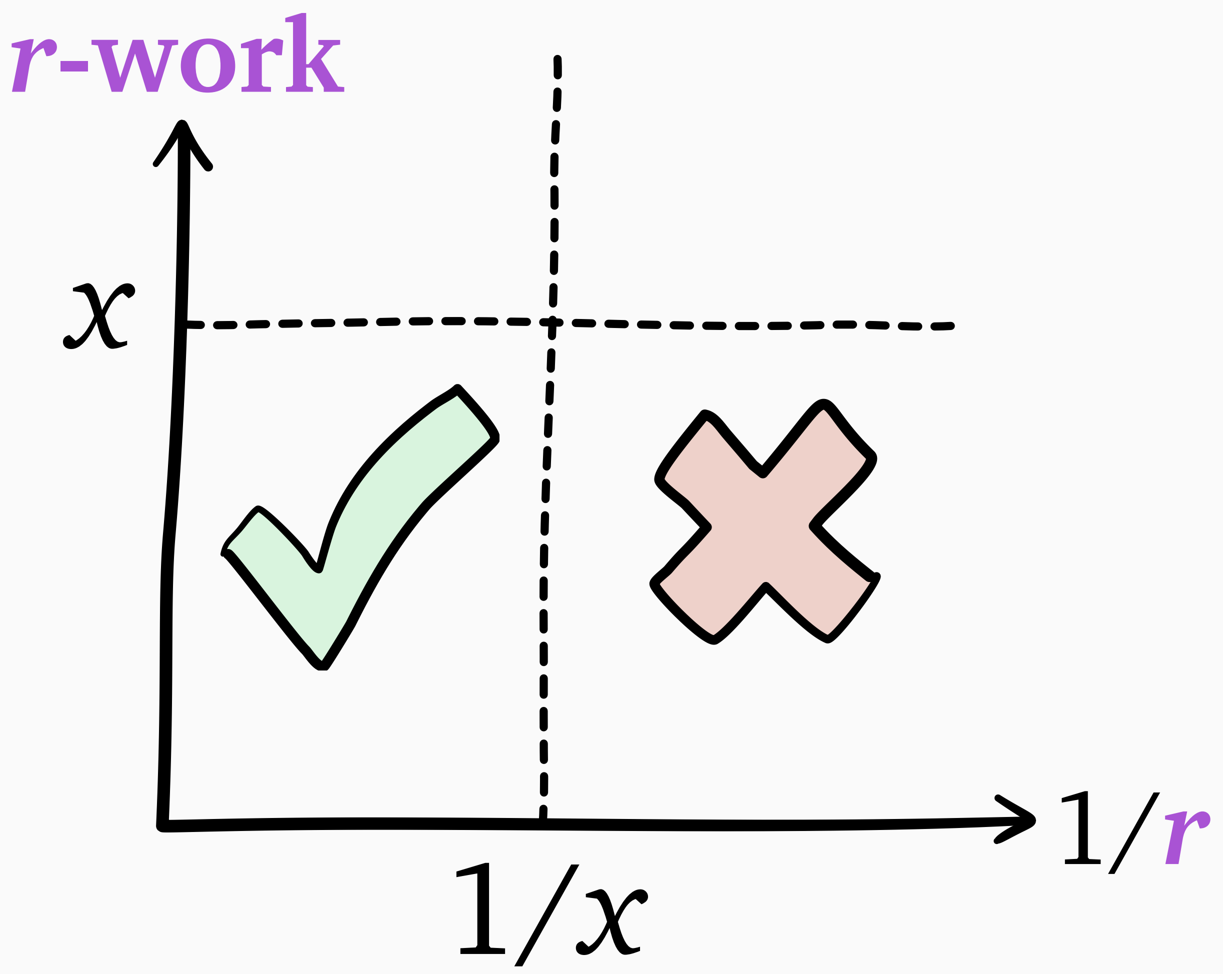
Theorem: $\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \frac{1}{\lambda}\int_0^\infty \mathbf{E}[W(r)]\,\mathrm{d}(1/r)$

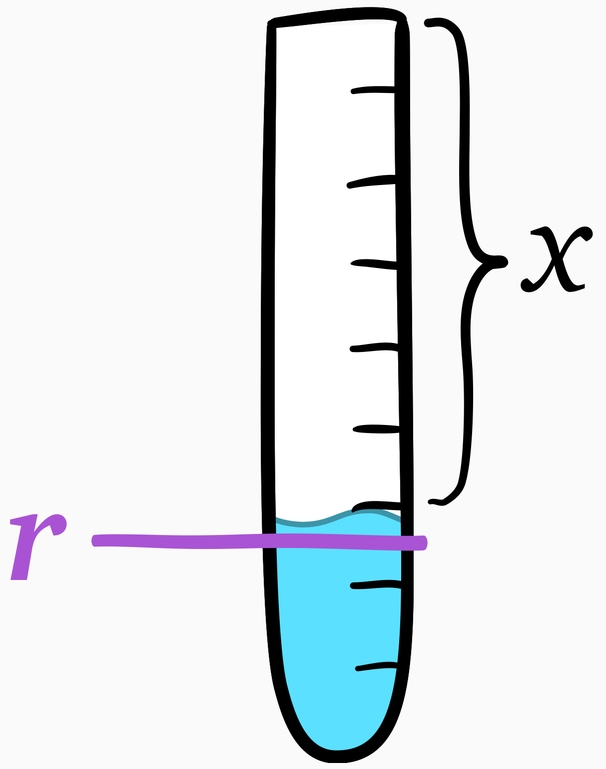
Proof:

remaining size x

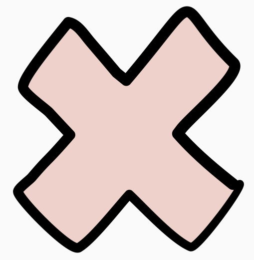
One job's ***r*-work**:



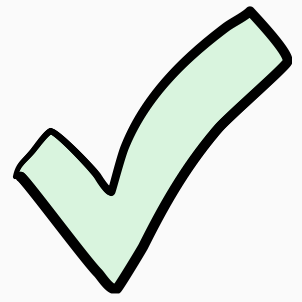




$r < x$: r-work $= 0$



$r \geq x$: r-work $= x$



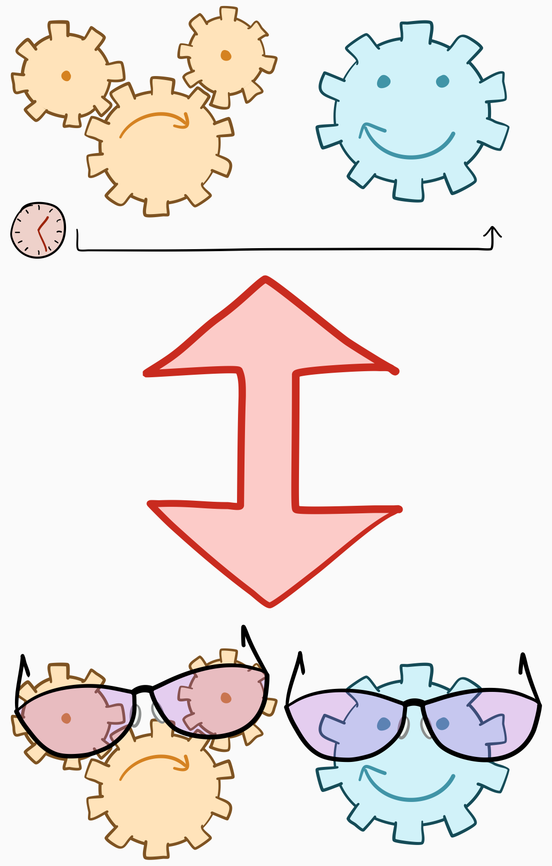
Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)

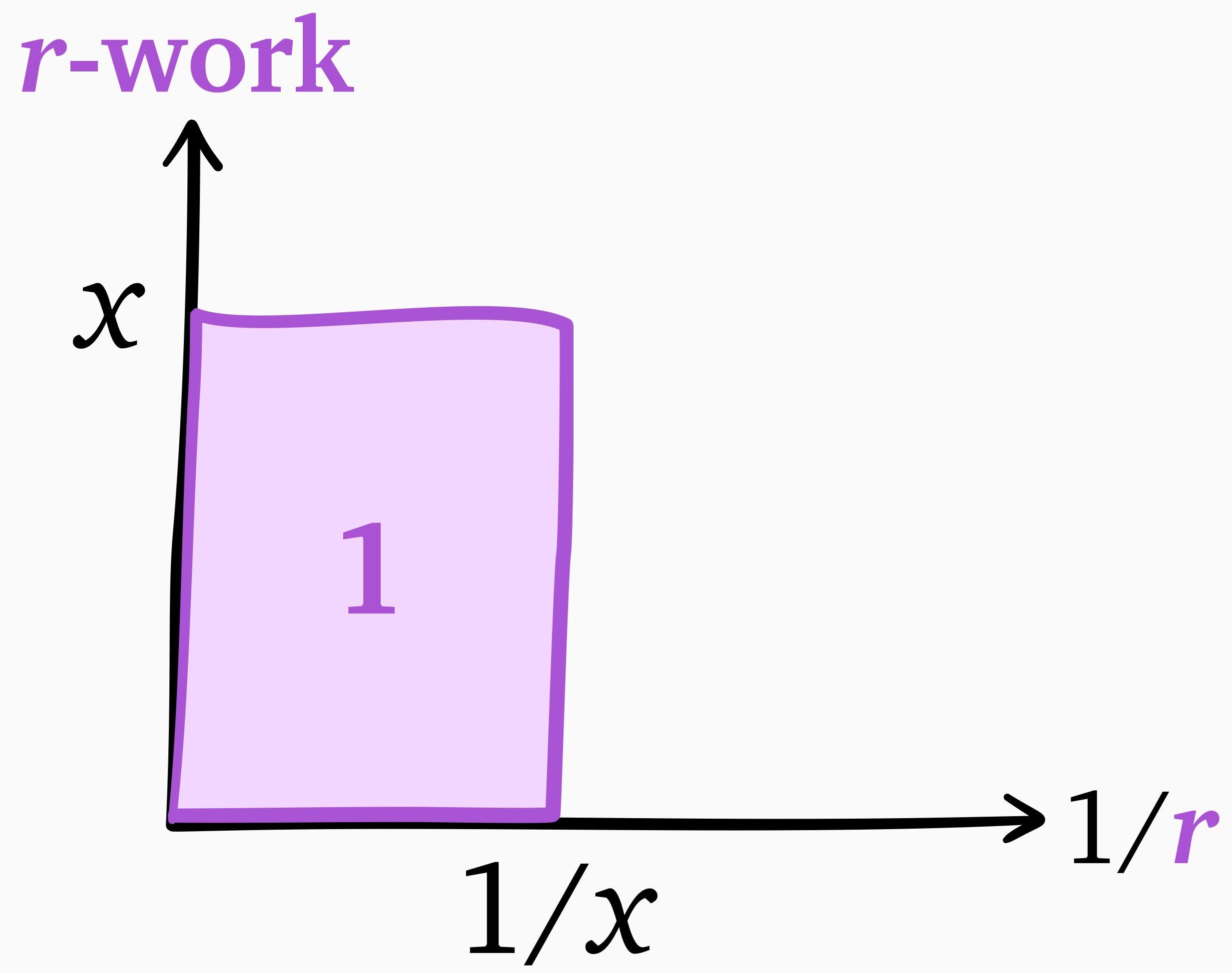
Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \frac{1}{\lambda}\int_0^\infty \mathbf{E}[W(r)]\,\mathrm{d}(1/r)$$

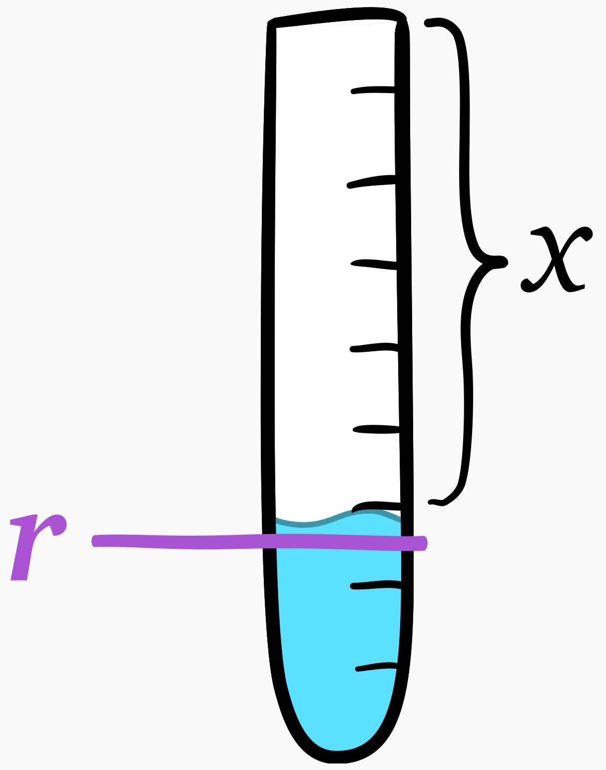
Proof:

remaining size x

One job's ***r*-work**:







$r < x$: r-work $= 0$

$r \geq x$: r-work $= x$

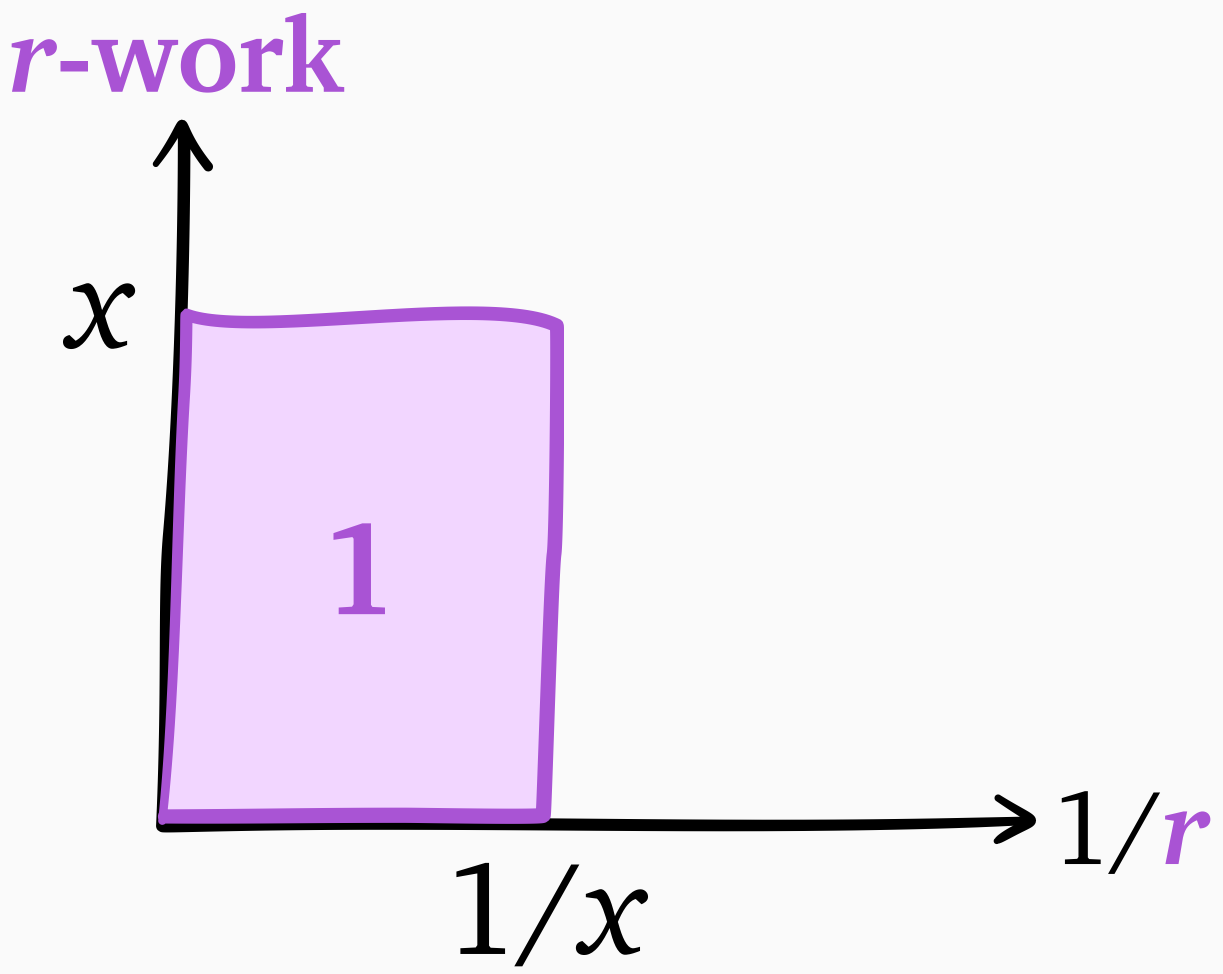
Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)

Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \frac{1}{\lambda}\int_0^\infty \mathbf{E}[W(r)]\,\mathrm{d}(1/r)$$

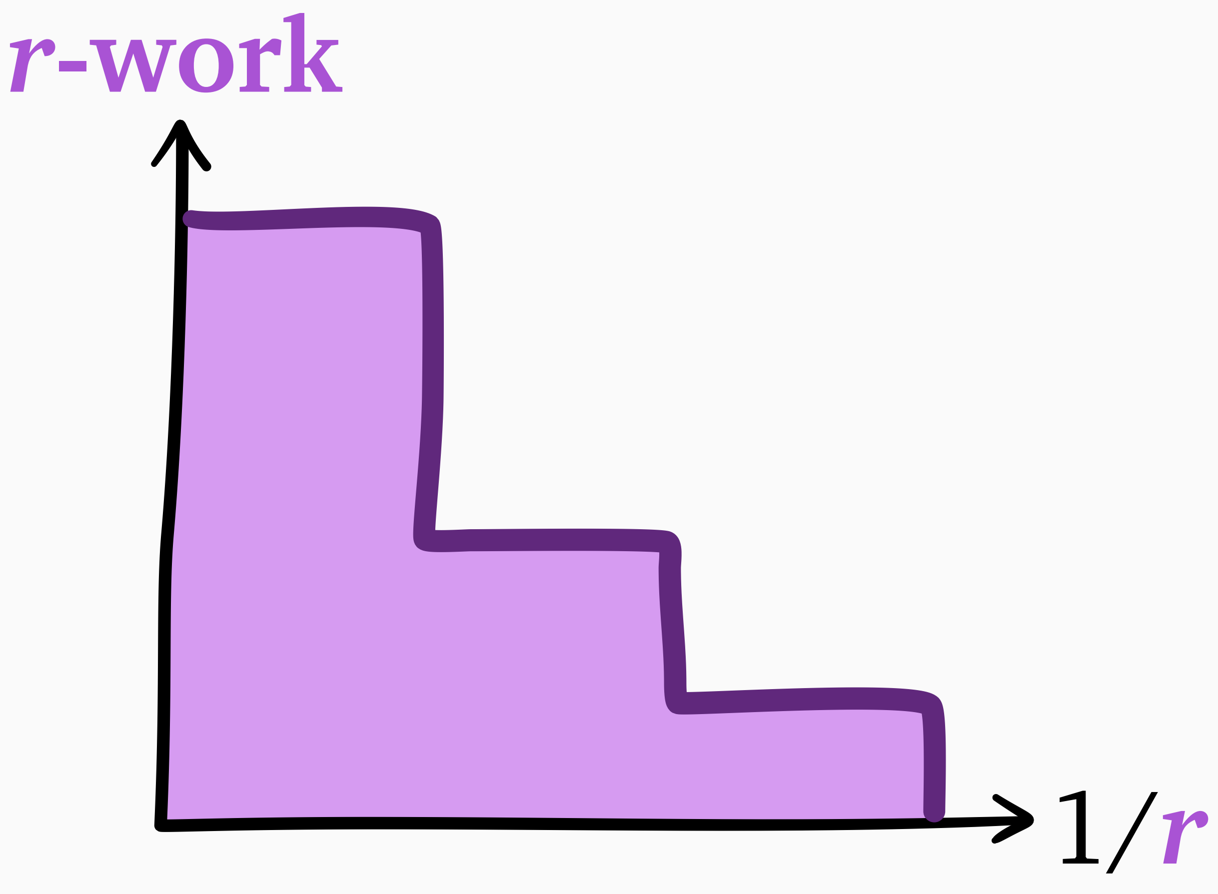
Proof:

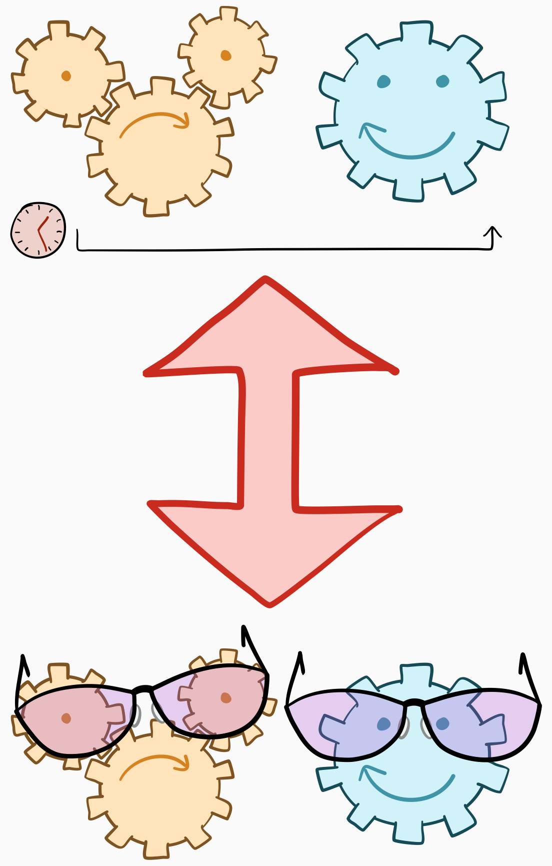
remaining size x

One job's ***r*-work**:



All jobs' ***r*-work**:





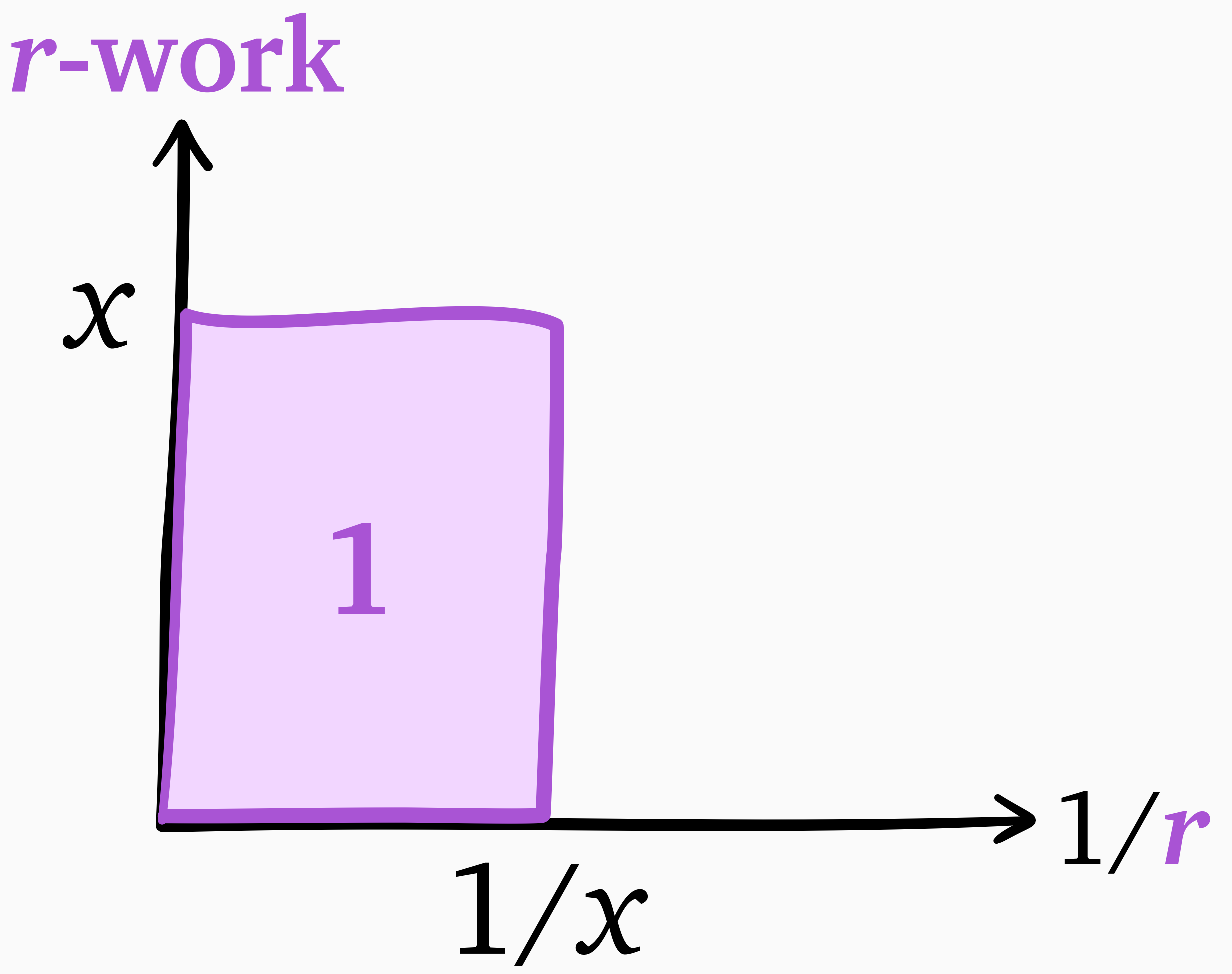
Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)

Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \frac{1}{\lambda}\int_0^\infty \mathbf{E}[W(r)]\,\mathrm{d}(1/r)$$

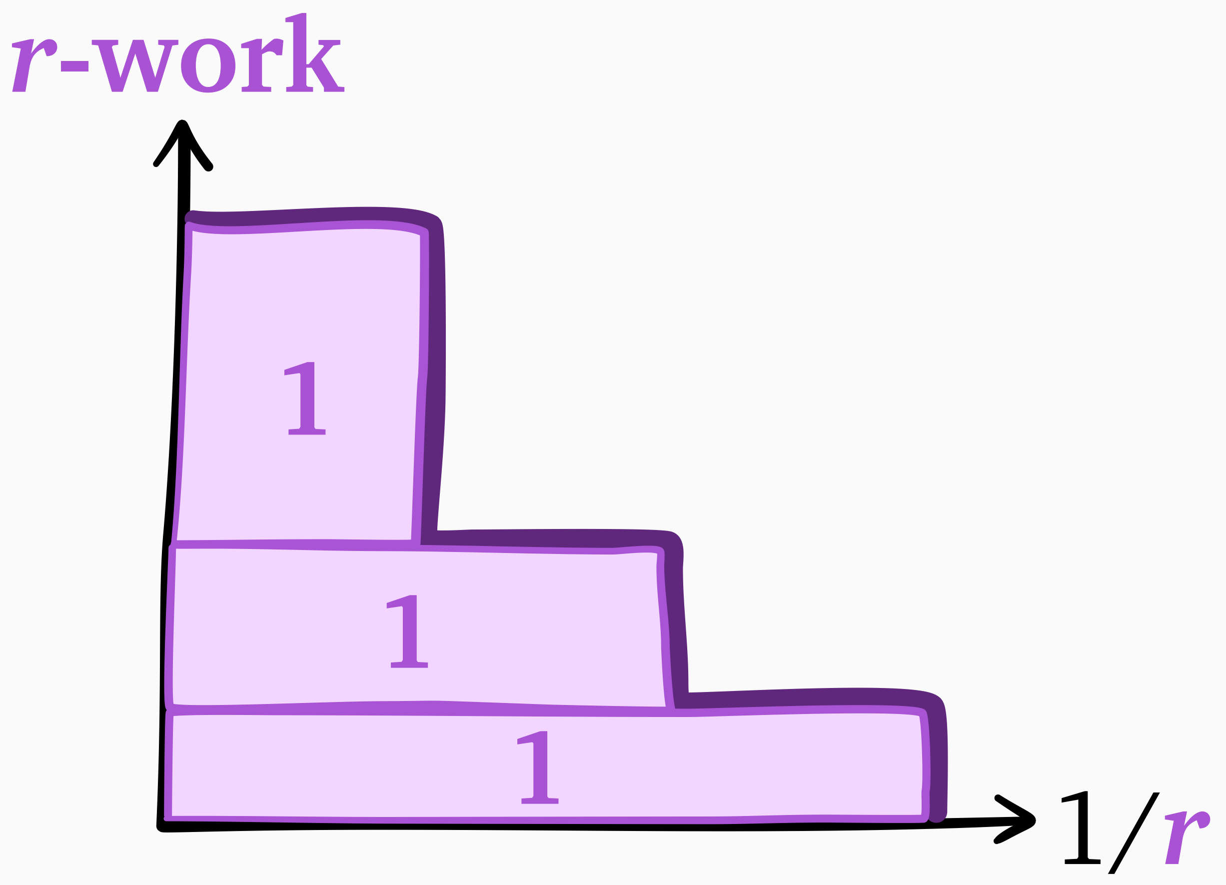
Proof:

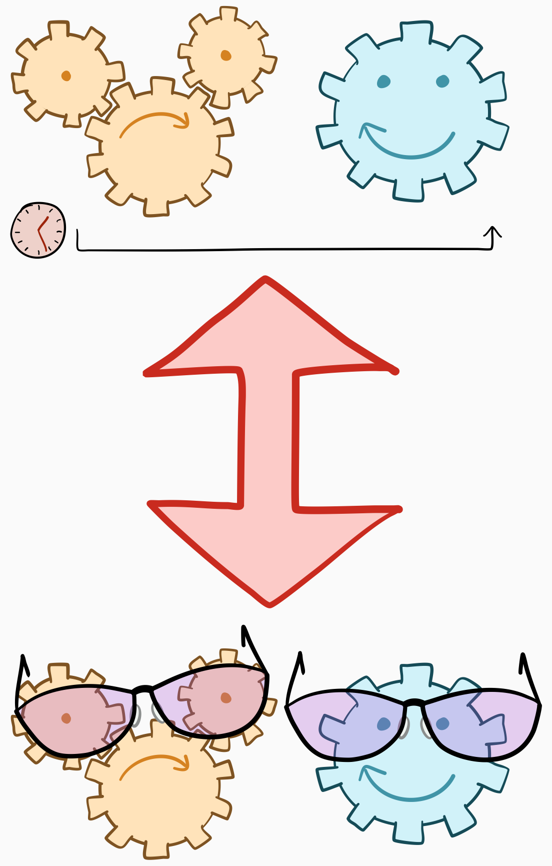
remaining size x

One job's r-work:



All jobs' r-work:





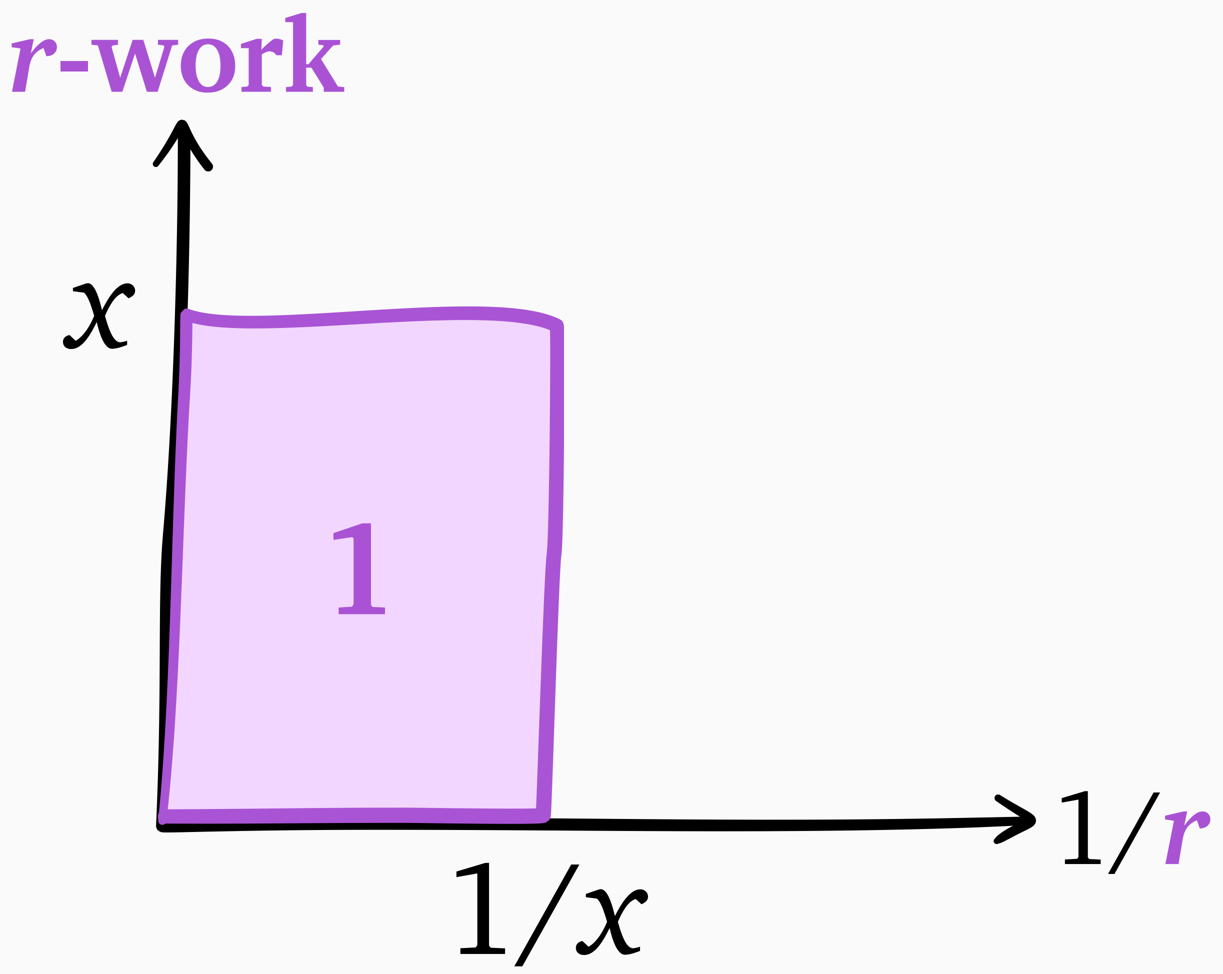
Step 1: $\mathbf{E}[T]$ to $\mathbf{E}[W(r)]$ (SRPT)

Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \frac{1}{\lambda}\int_0^\infty \mathbf{E}[W(r)]\,\mathrm{d}(1/r)$$

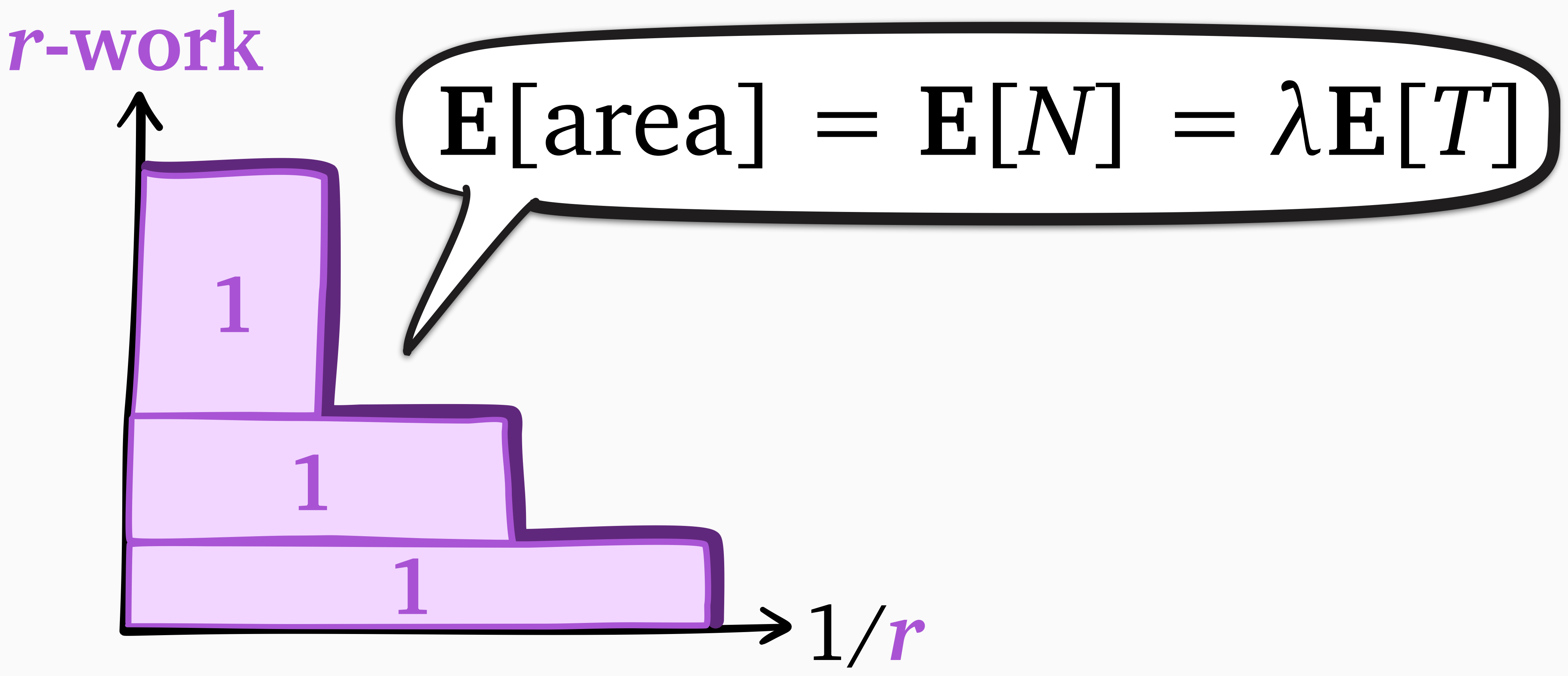
Proof:

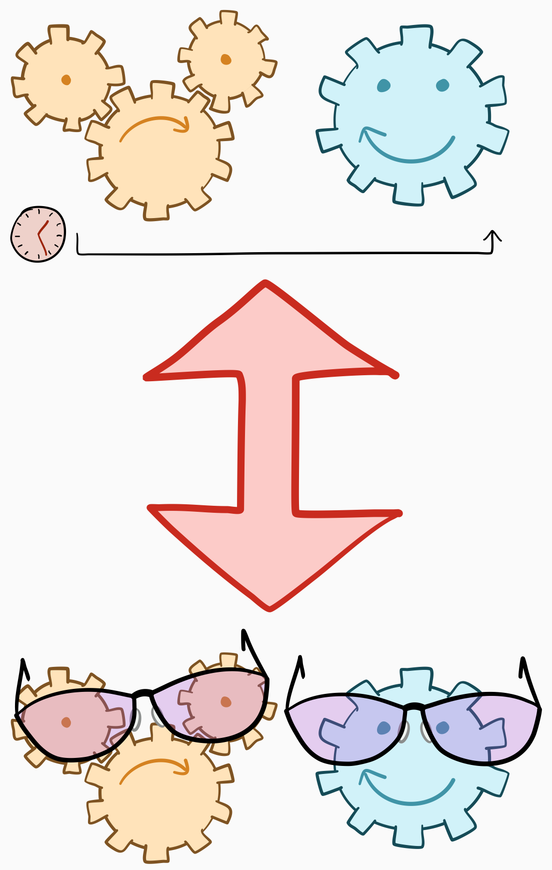
remaining size x

One job's ***r*-work**:

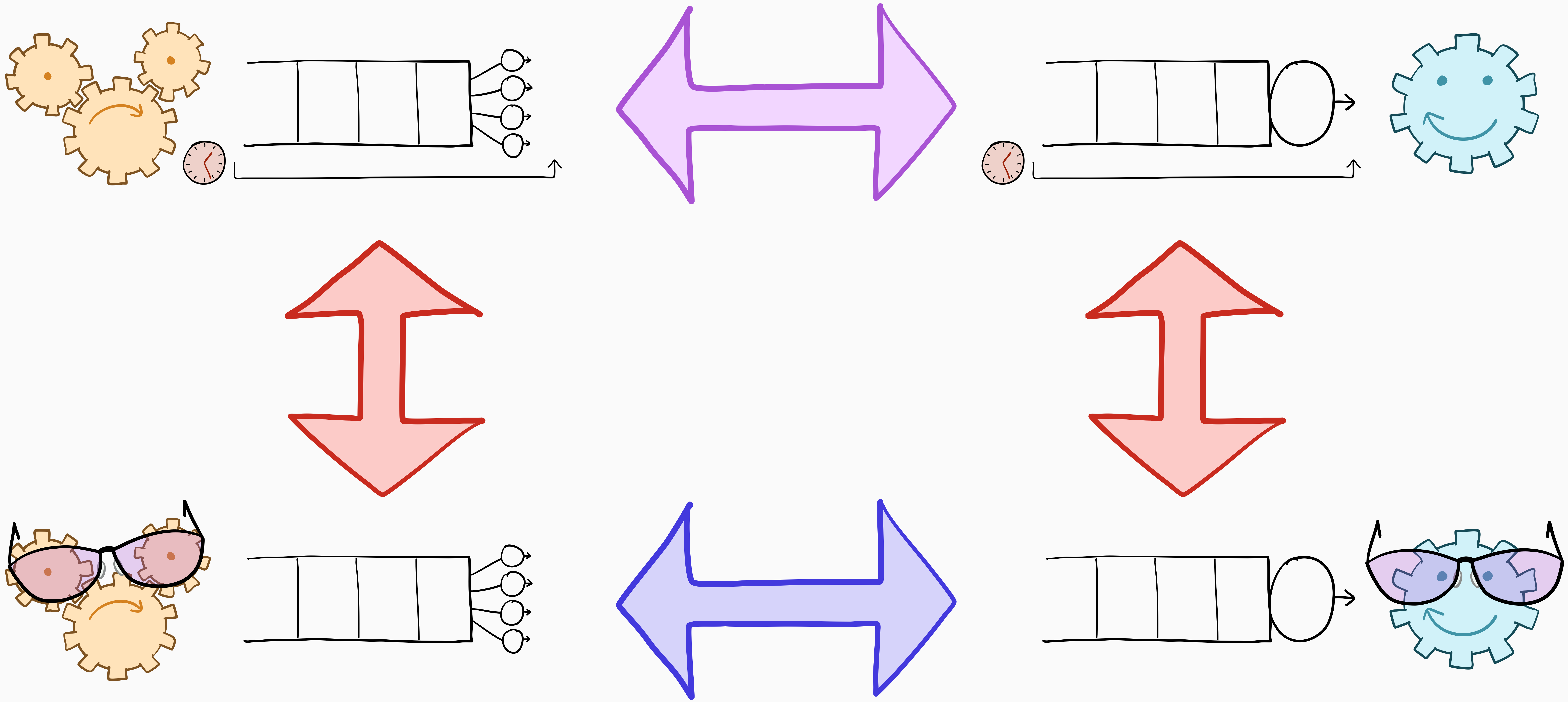


All jobs' ***r*-work**:

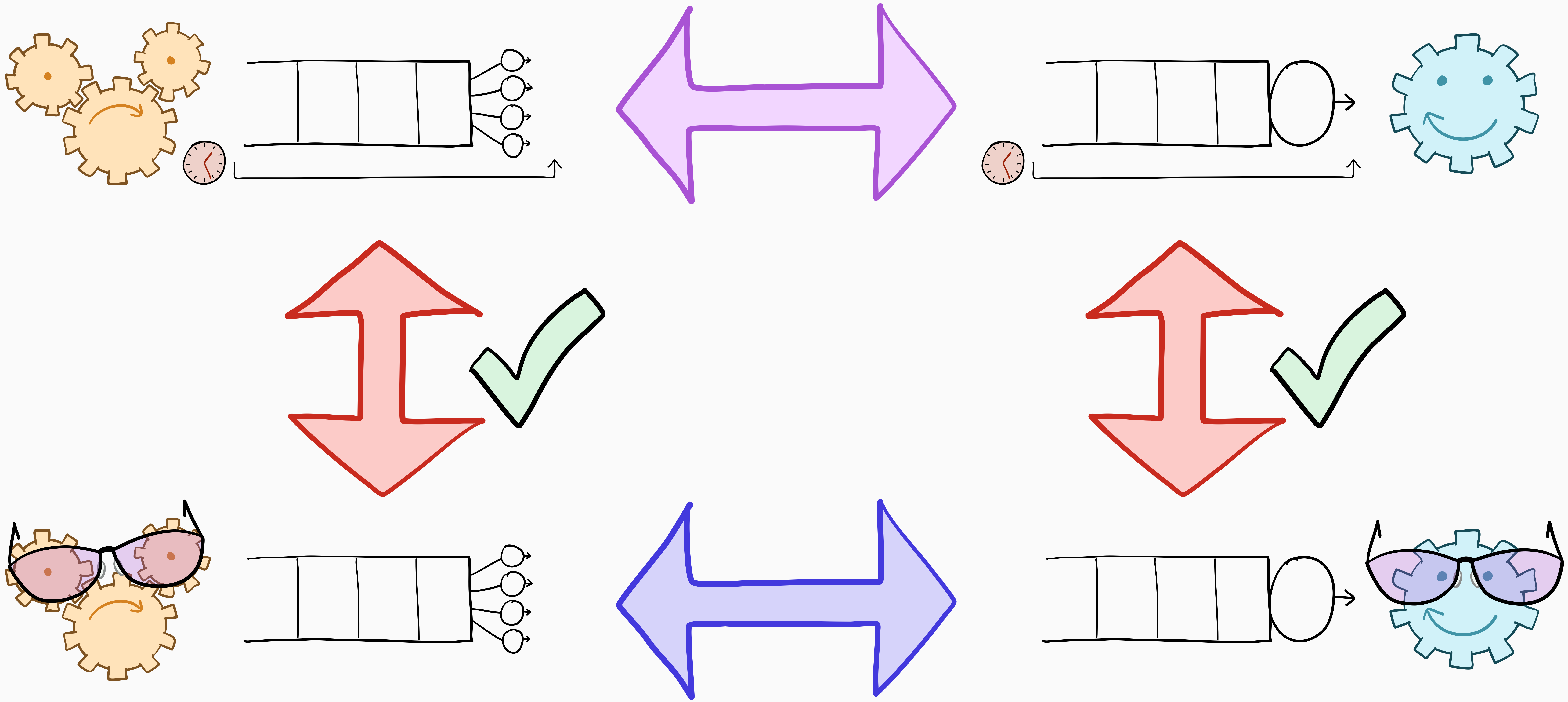




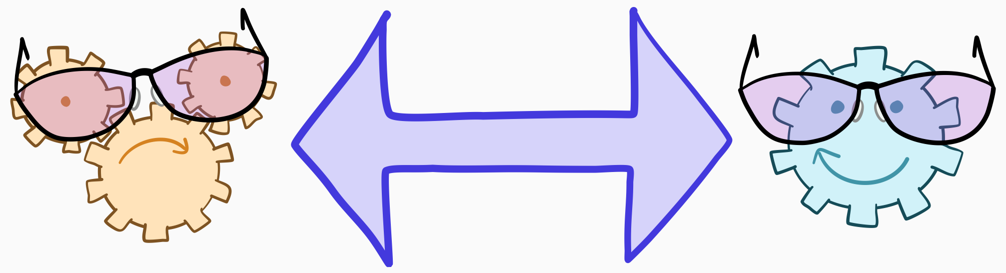
Response time via *r*-work



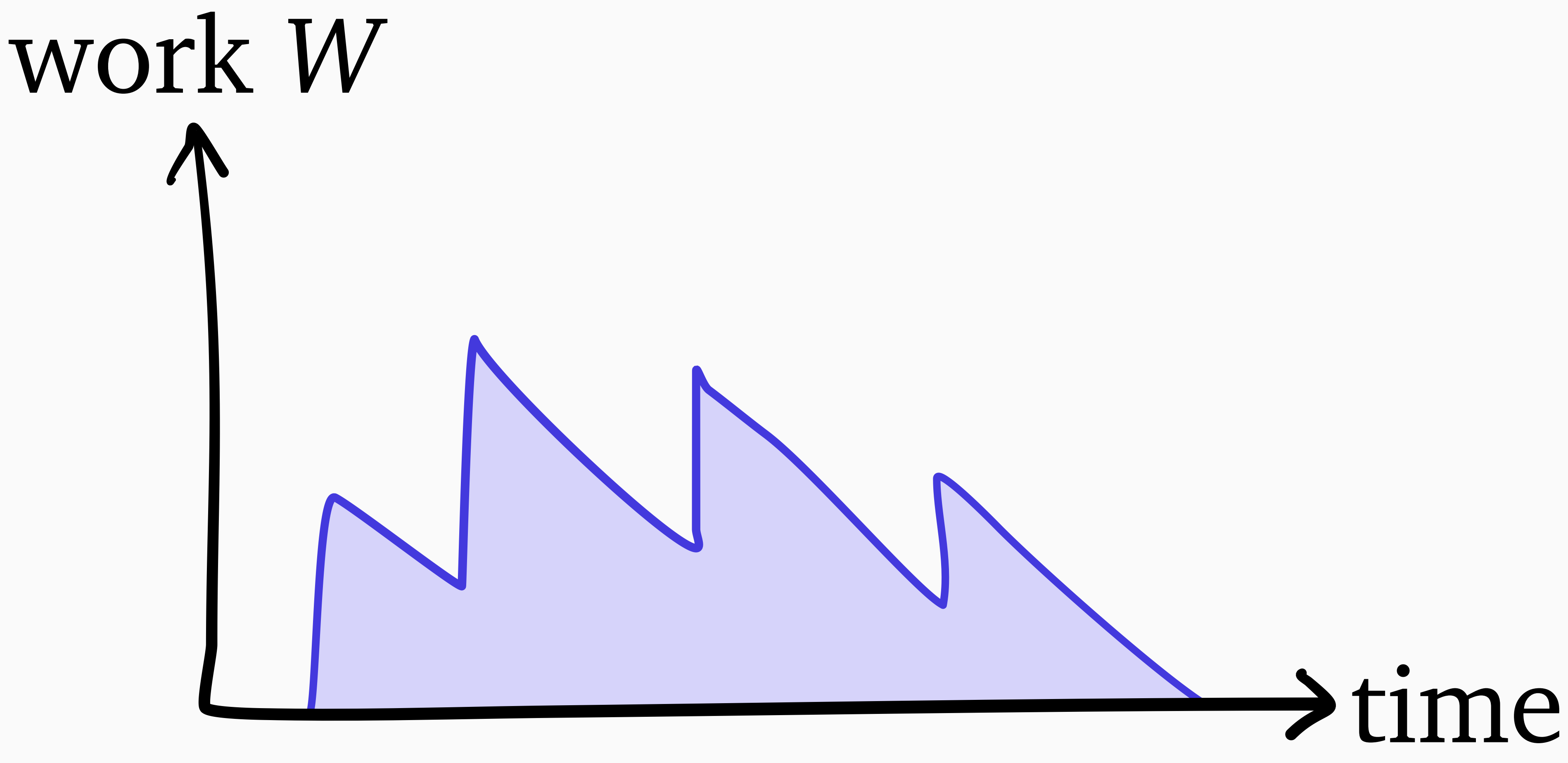
Response time via *r-work*

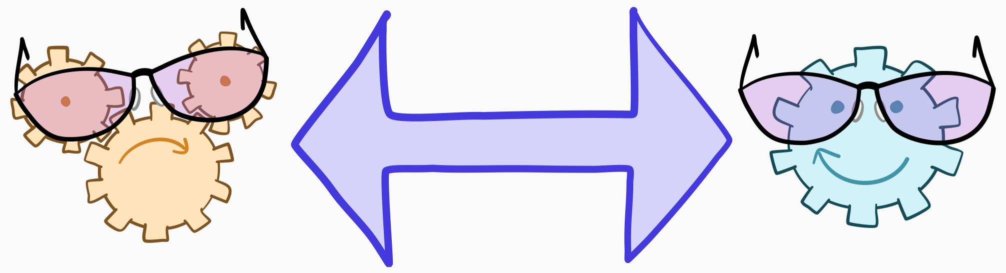


Step 2: $\mathbf{E}[W]$ difference (warmup)

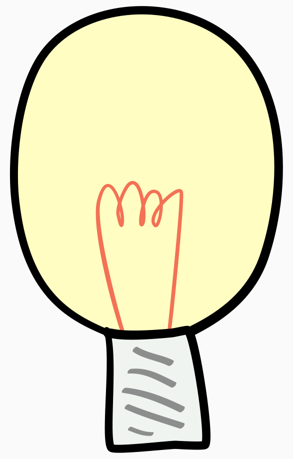


Step 2: $\mathbf{E}[W]$ difference (warmup)

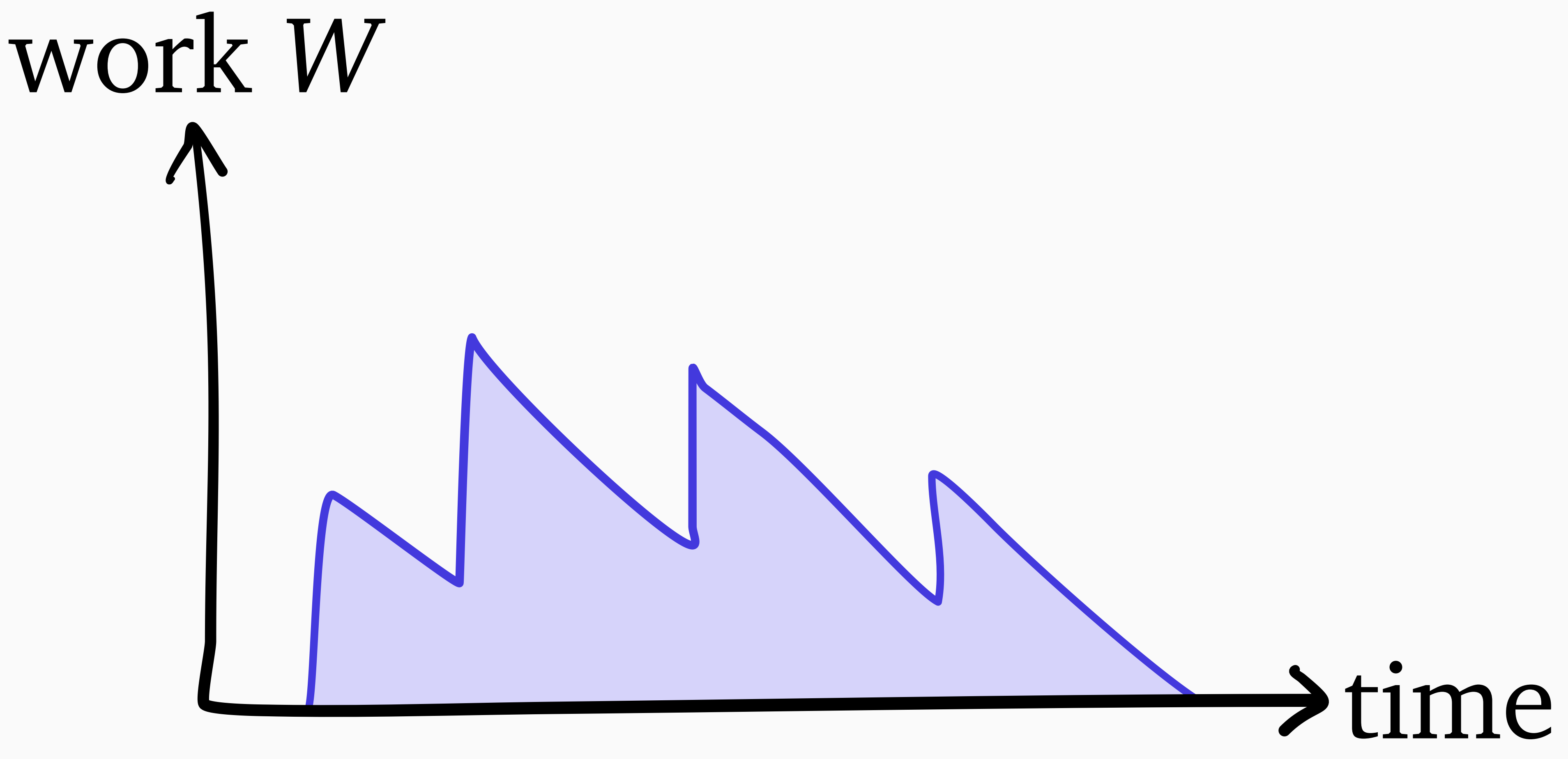




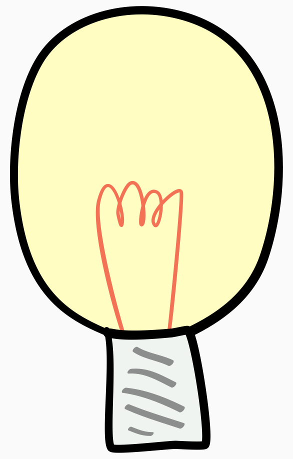
Step 2: $\mathbf{E}[W]$ difference (warmup)



In steady-state system, for any f,
$\mathbf{E}[f(W)]$ constant w.r.t. time

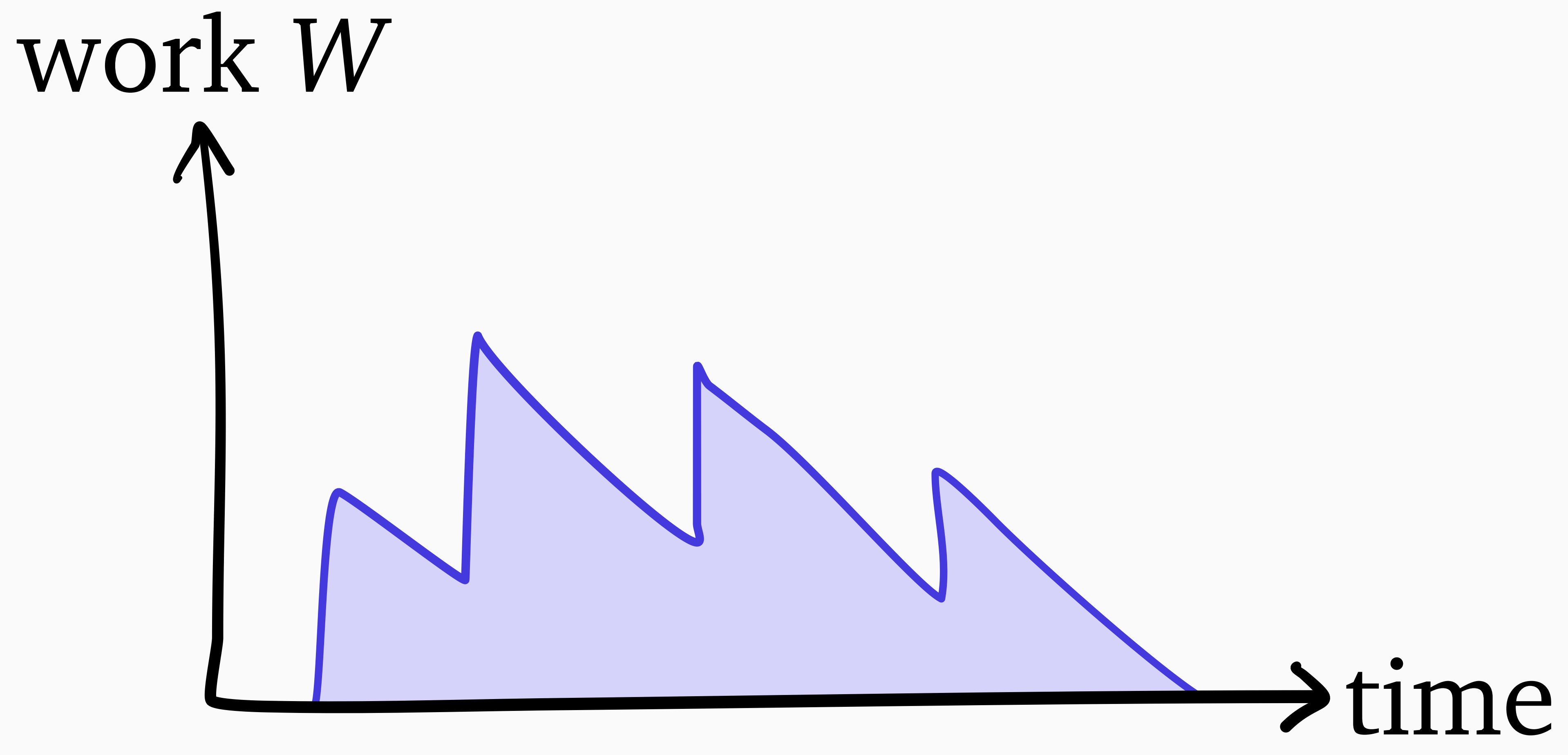


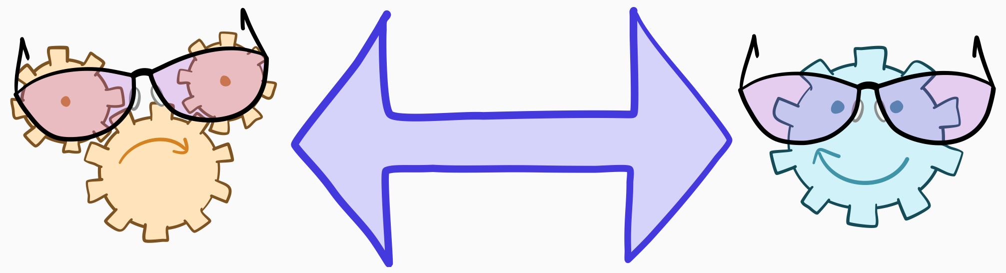
Step 2: $\mathbf{E}[W]$ difference (warmup)



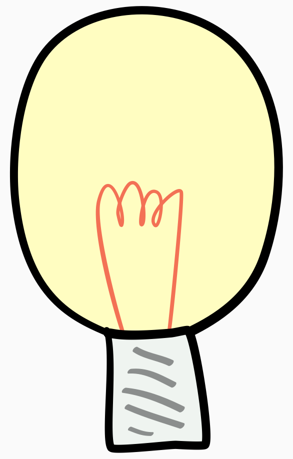
In steady-state system, for any f,
$\mathbf{E}[f(W)]$ constant w.r.t. time

we use
$f(w) = w^2$



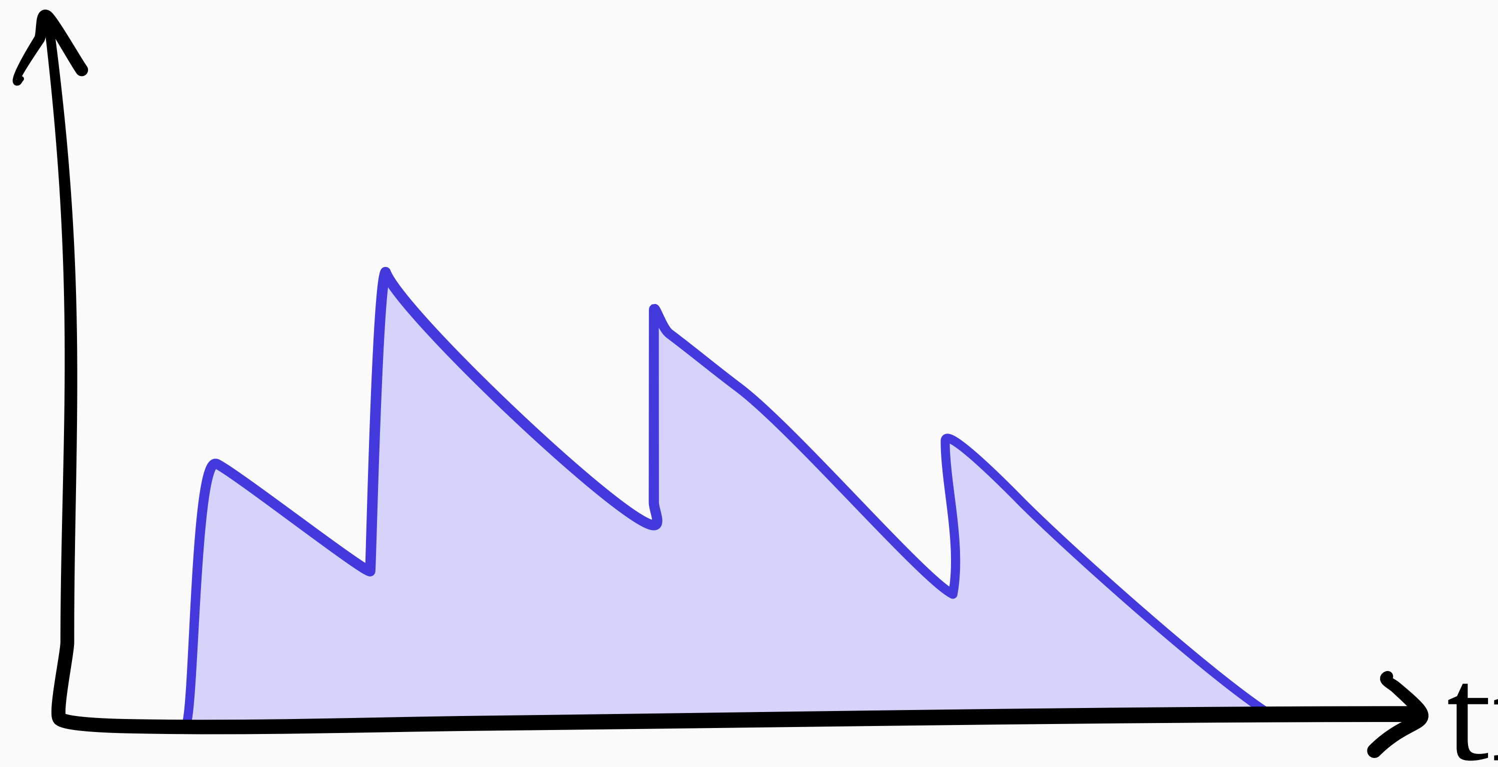


Step 2: $\mathbf{E}[W]$ difference (warmup)



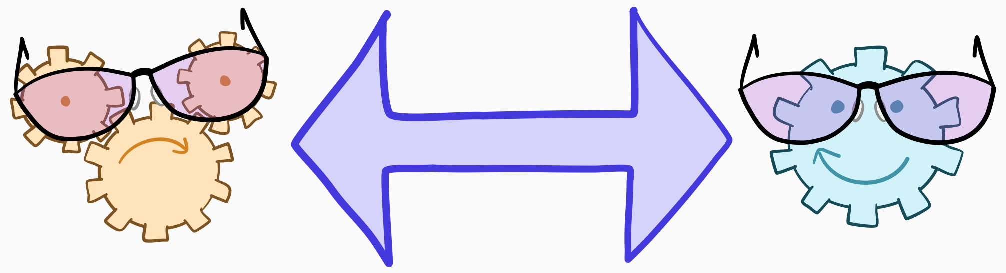
In steady-state system, for any f, $\mathbf{E}[f(W)]$ constant w.r.t. time

we use $f(w) = w^2$



$$\mathbf{E}[W^2 \text{ decrease rate}] = 2\mathbf{E}[BW]$$

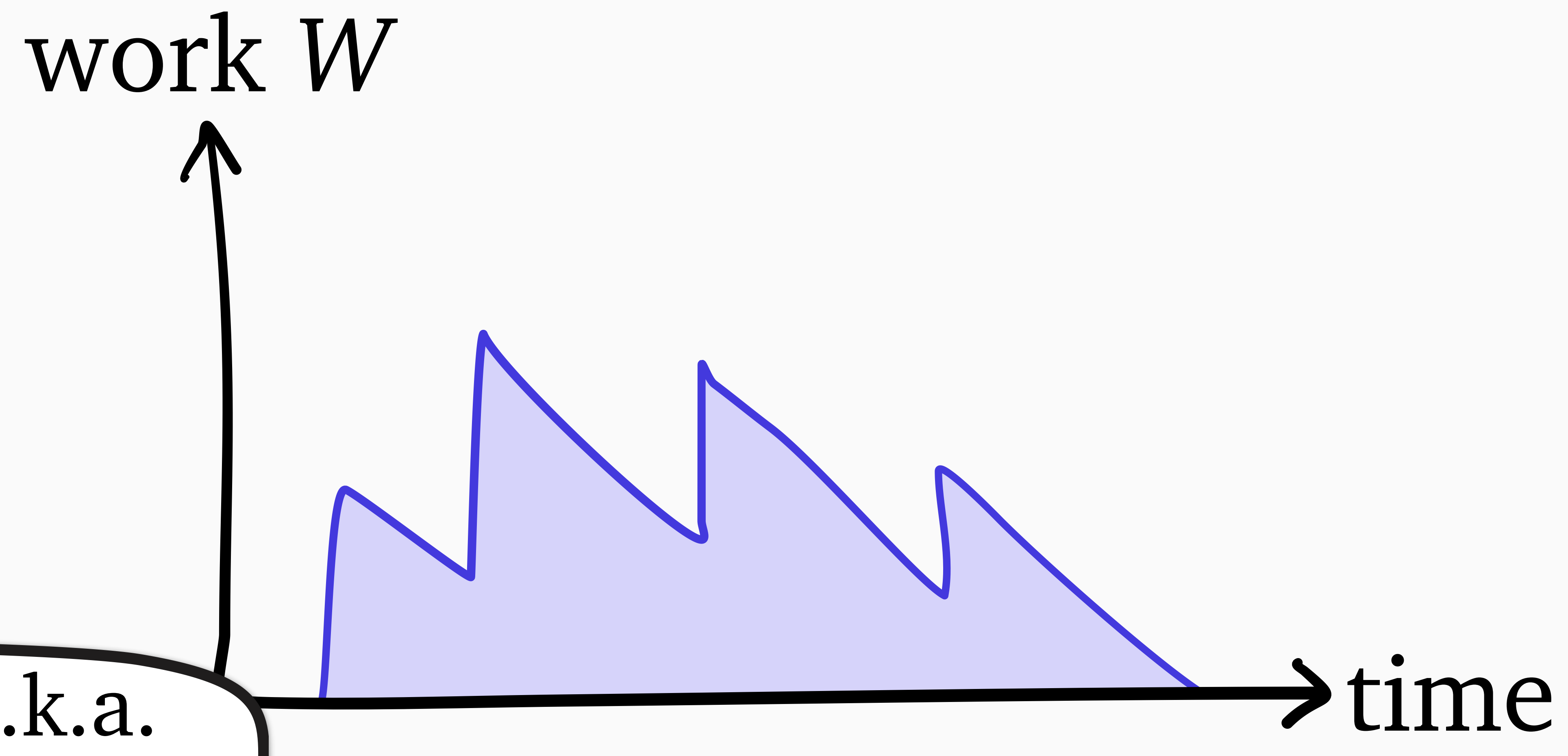
$$\mathbf{E}[W^2 \text{ increase rate}] = \lambda\mathbf{E}[(W+S)^2 - W^2]$$



Step 2: $\mathbf{E}[W]$ difference (warmup)

In steady-state system, for any f,
$\mathbf{E}[f(W)]$ constant w.r.t. time

we use
$f(w) = w^2$

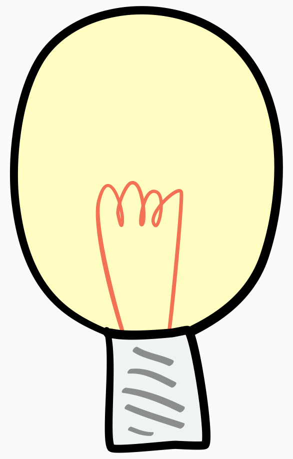


B = service rate, a.k.a.
fraction of servers busy

$$\mathbf{E}[W^2 \text{ decrease rate}] = 2\mathbf{E}[BW]$$

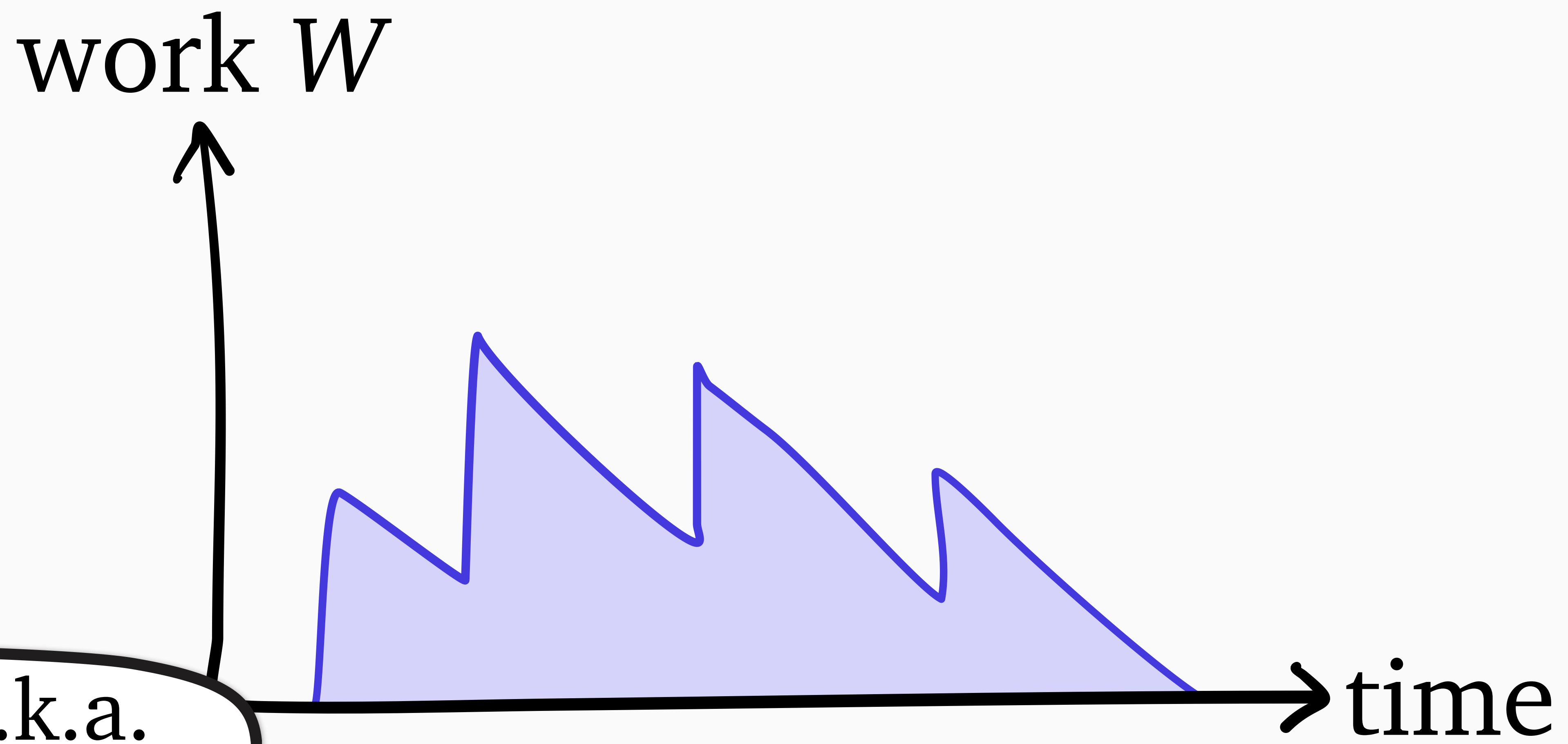
$$\mathbf{E}[W^2 \text{ increase rate}] = \lambda\mathbf{E}[(W+S)^2 - W^2]$$

Step 2: $\mathbf{E}[W]$ difference (warmup)



In steady-state system, for any f,
$\mathbf{E}[f(W)]$ constant w.r.t. time

we use
$f(w) = w^2$



B = service rate, a.k.a.
fraction of servers busy

$$\mathbf{E}[W] = \frac{\frac{\lambda}{2}\mathbf{E}[S^2]}{1-\rho} + \frac{\mathbf{E}[(1-B)W]}{1-\rho}$$

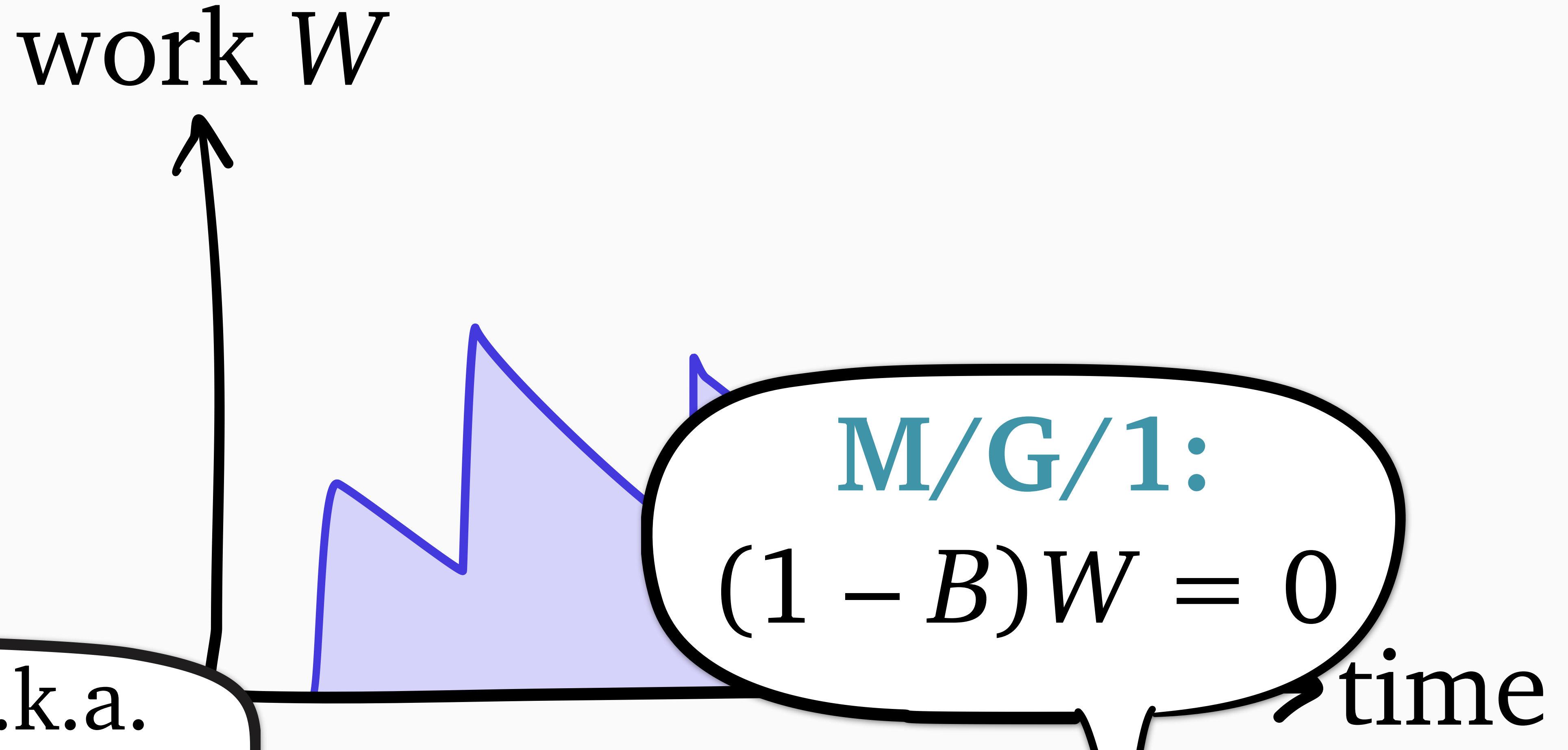
$$\mathbf{E}[W^2 \text{ decrease rate}] = 2\mathbf{E}[BW]$$

$$\mathbf{E}[W^2 \text{ increase rate}] = \lambda\mathbf{E}[(W+S)^2 - W^2]$$

Step 2: $\mathbf{E}[W]$ difference (warmup)

In steady-state system, for any f,
$\mathbf{E}[f(W)]$ constant w.r.t. time

we use
$f(w) = w^2$



M/G/1:
$(1 - B)W = 0$

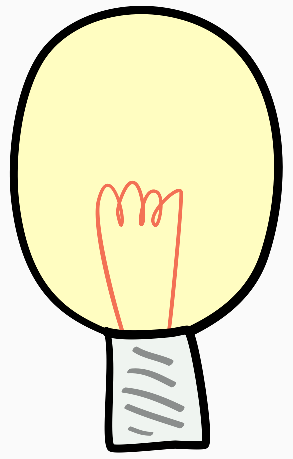
B = service rate, a.k.a.
fraction of servers busy

$$\mathbf{E}[W] = \frac{\frac{\lambda}{2}\mathbf{E}[S^2]}{1-\rho} + \frac{\mathbf{E}[(1-B)W]}{1-\rho}$$

$$\mathbf{E}[W^2 \text{ decrease rate}] = 2\mathbf{E}[BW]$$

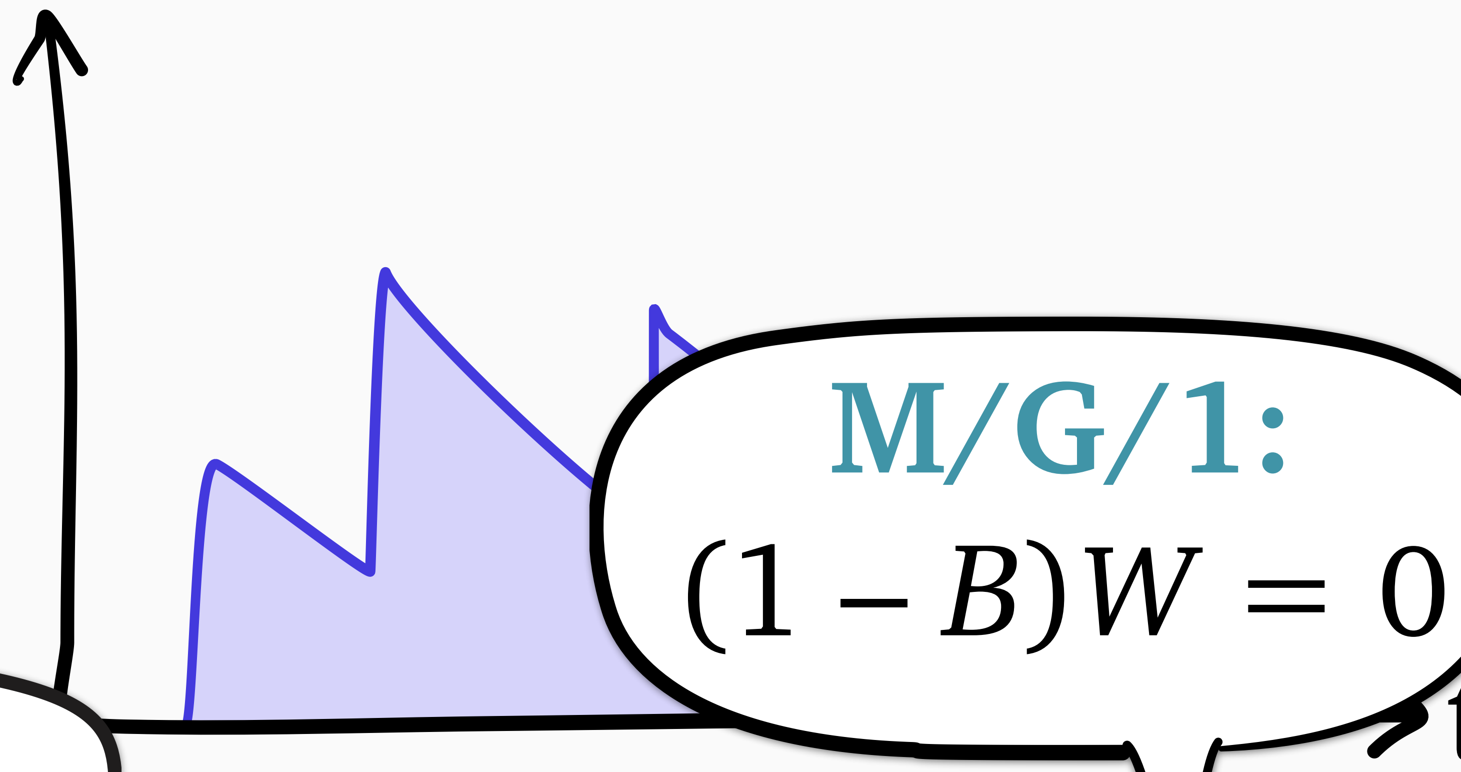
$$\mathbf{E}[W^2 \text{ increase rate}] = \lambda\mathbf{E}[(W+S)^2 - W^2]$$

Step 2: $\mathbf{E}[W]$ difference (warmup)



In steady-state system, for any f, $\mathbf{E}[f(W)]$ constant w.r.t. time

we use $f(w) = w^2$



M/G/1: $(1 - B)W = 0$

B = service rate, a.k.a. fraction of servers busy

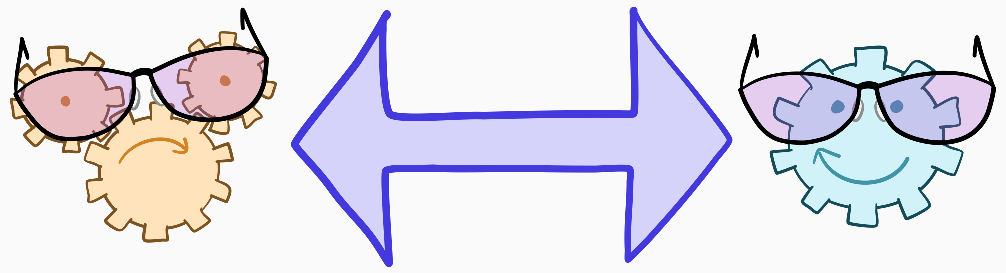
$$\mathbf{E}[W^2 \text{ decrease rate}] = 2\mathbf{E}[BW]$$

$$\mathbf{E}[W^2 \text{ increase rate}] = \lambda\mathbf{E}[(W+S)^2 - W^2]$$

$$\mathbf{E}[W] = \frac{\frac{\lambda}{2}\mathbf{E}[S^2]}{1-\rho} + \frac{\mathbf{E}[(1-B)W]}{1-\rho}$$

Theorem:

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$

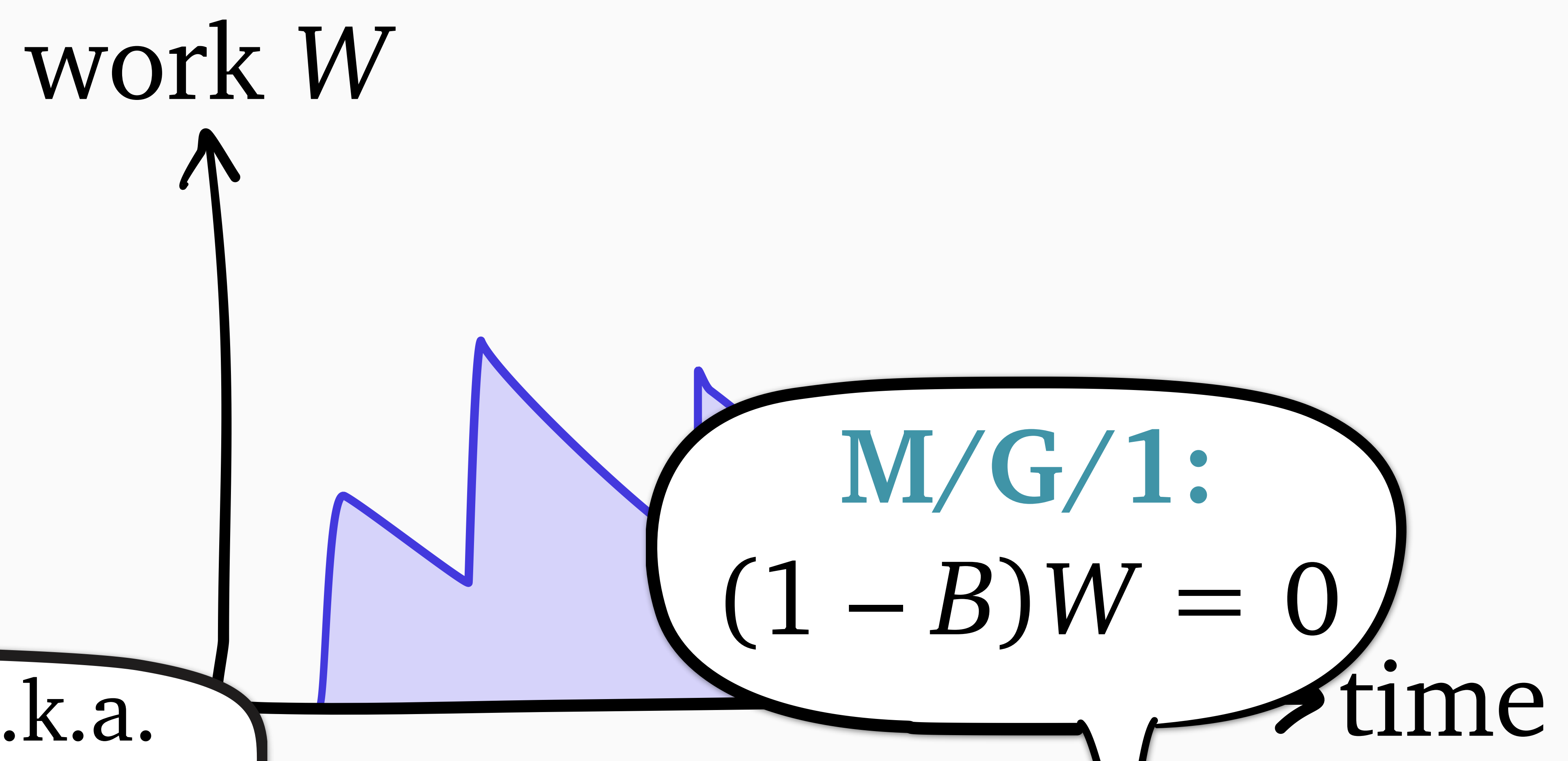


Step 2: $\mathbf{E}[W]$ difference (warmup)

In steady-state system, for any f,
$\mathbf{E}[f(W)]$ constant w.r.t. time

we use $f(w) = w^2$

B = service rate, a.k.a. fraction of servers busy



M/G/1: $(1 - B)W = 0$

$$\mathbf{E}[W] = \frac{\frac{\lambda}{2}\mathbf{E}[S^2]}{1-\rho} + \frac{\mathbf{E}[(1-B)W]}{1-\rho}$$

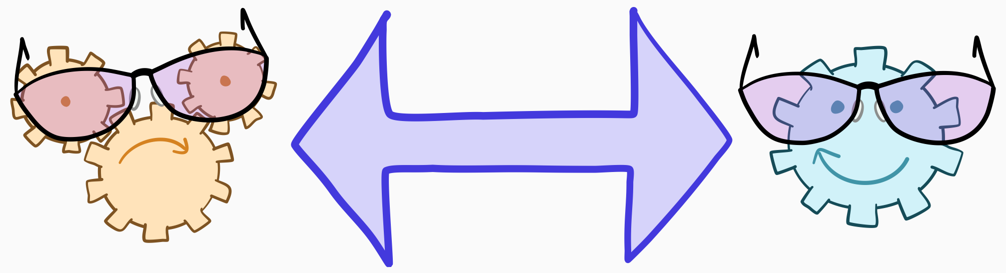
$$\mathbf{E}[W^2 \text{ decrease rate}] = 2\mathbf{E}[BW]$$

$$\mathbf{E}[W^2 \text{ increase rate}] = \lambda\mathbf{E}[(W+S)^2 - W^2]$$

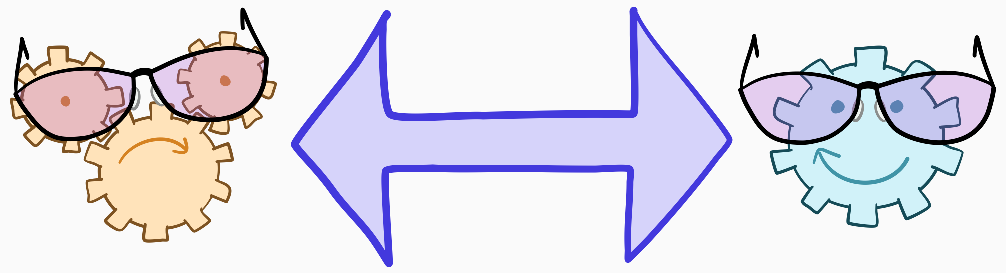
Theorem:

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$

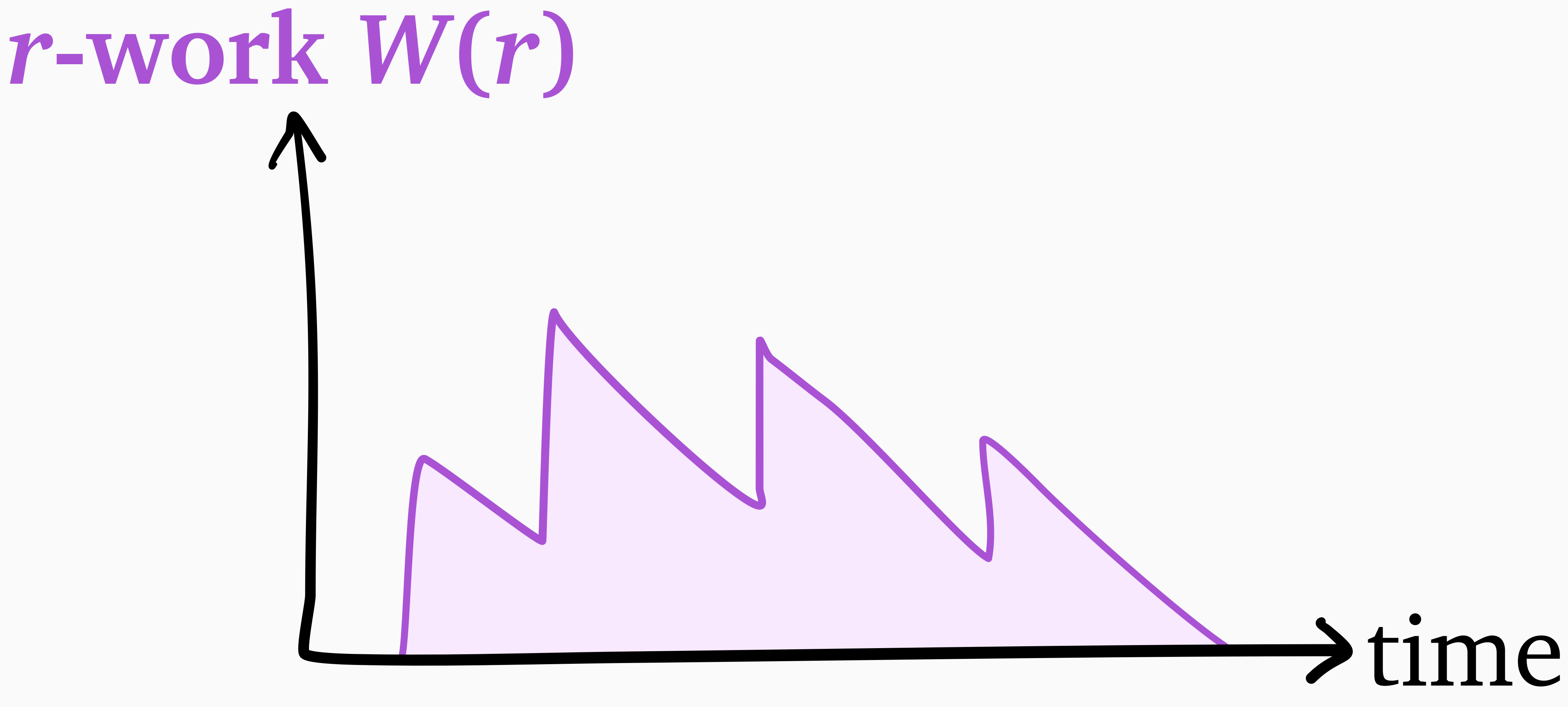
When S is exponential, compares $\mathbf{E}[N_k]$ to $\mathbf{E}[N_1]$

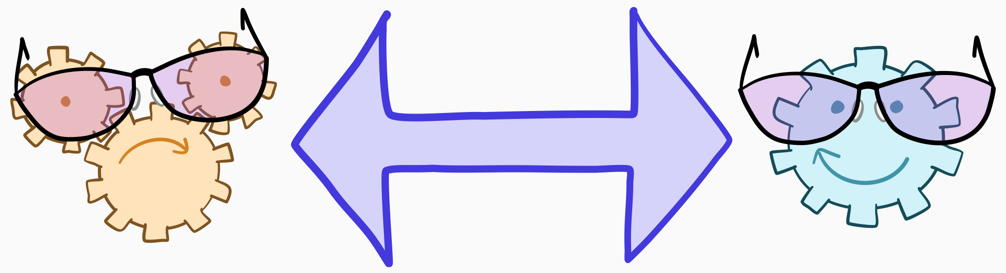


Step 2: $\mathbf{E}[W(r)]$ difference (SRPT)

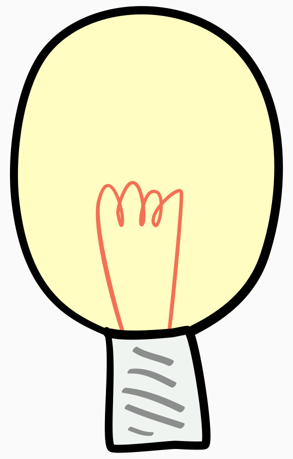


Step 2: E[$W(r)$] difference (SRPT)

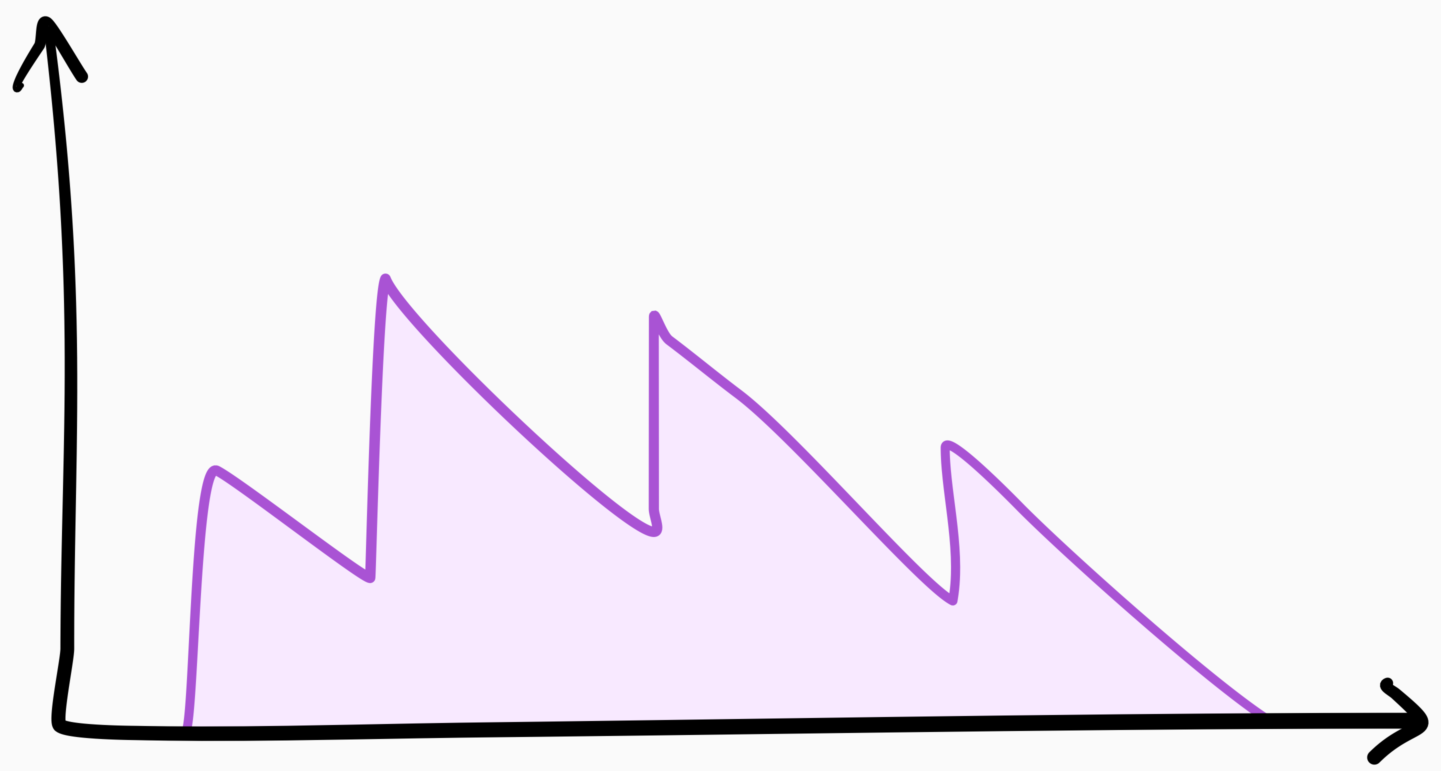


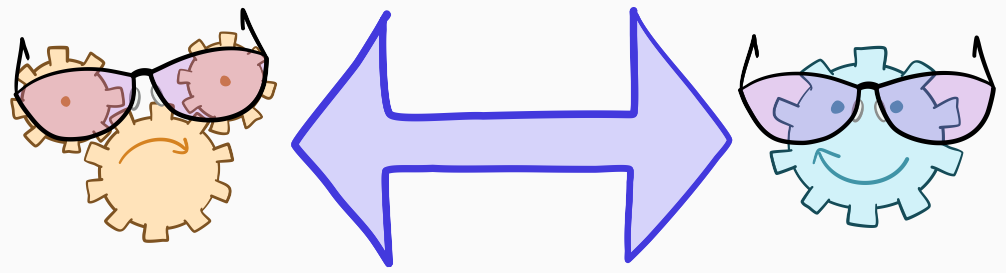


Step 2: $\mathbf{E}[W(r)]$ difference (SRPT)

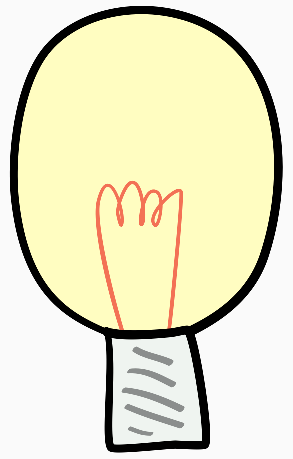


In steady-state system, for any f,
$\mathbf{E}[f(W(r))]$ constant w.r.t. time



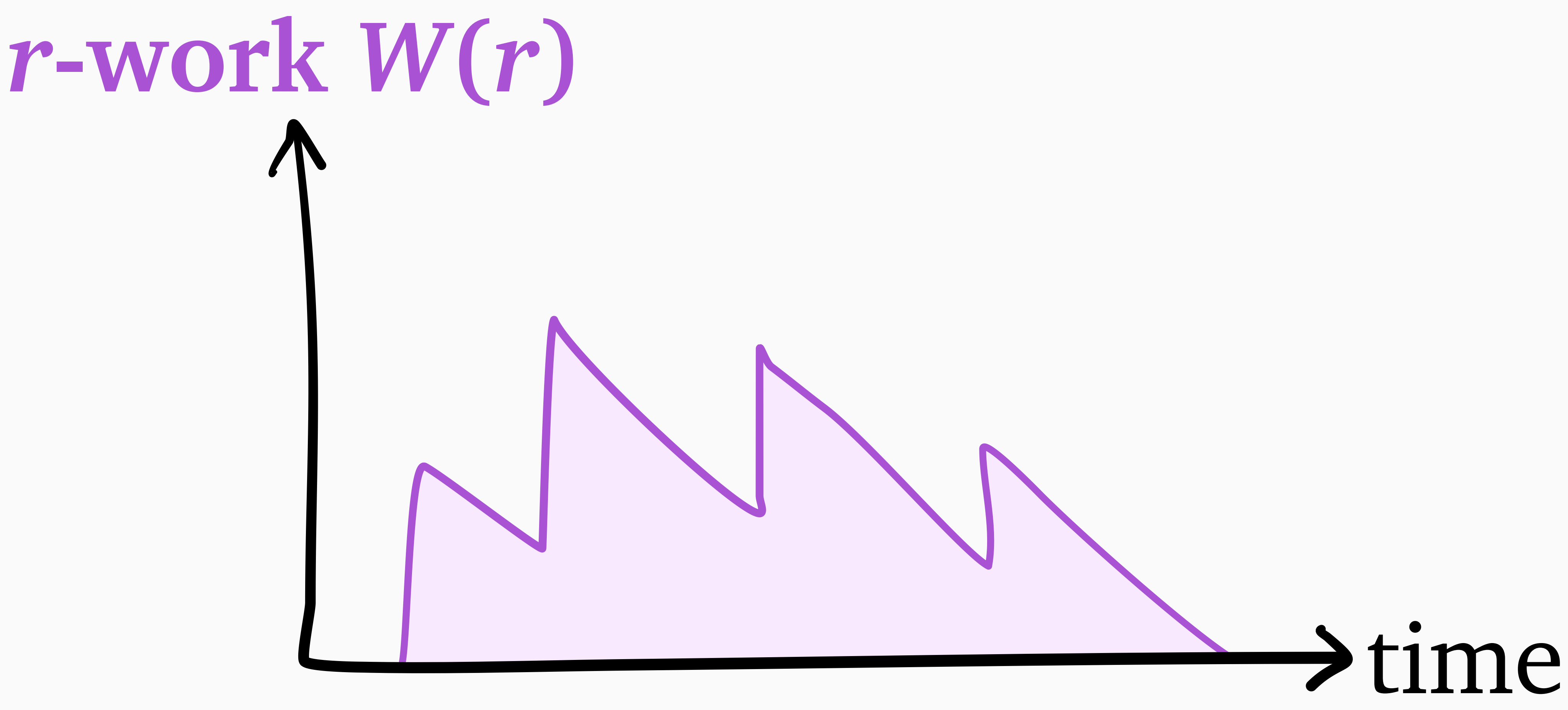


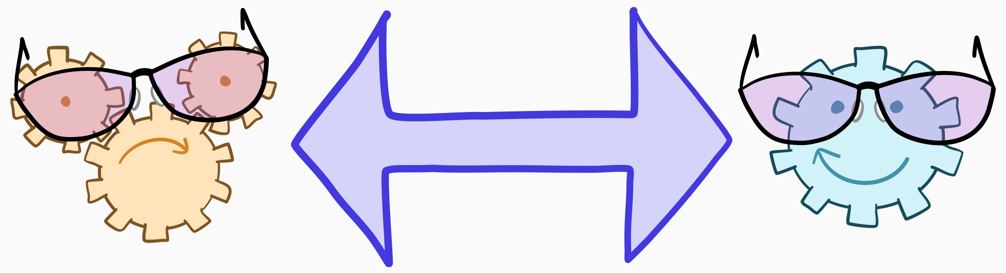
Step 2: $\mathbf{E}[W(r)]$ difference (SRPT)



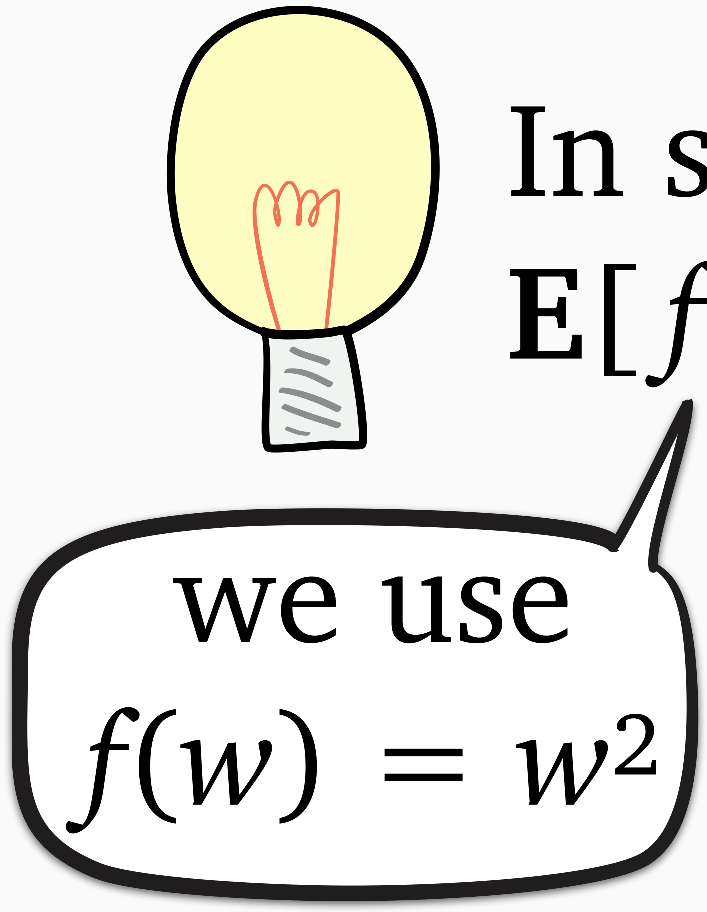
In steady-state system, for any f,
$\mathbf{E}[f(W(r))]$ constant w.r.t. time

we use
$f(w) = w^2$

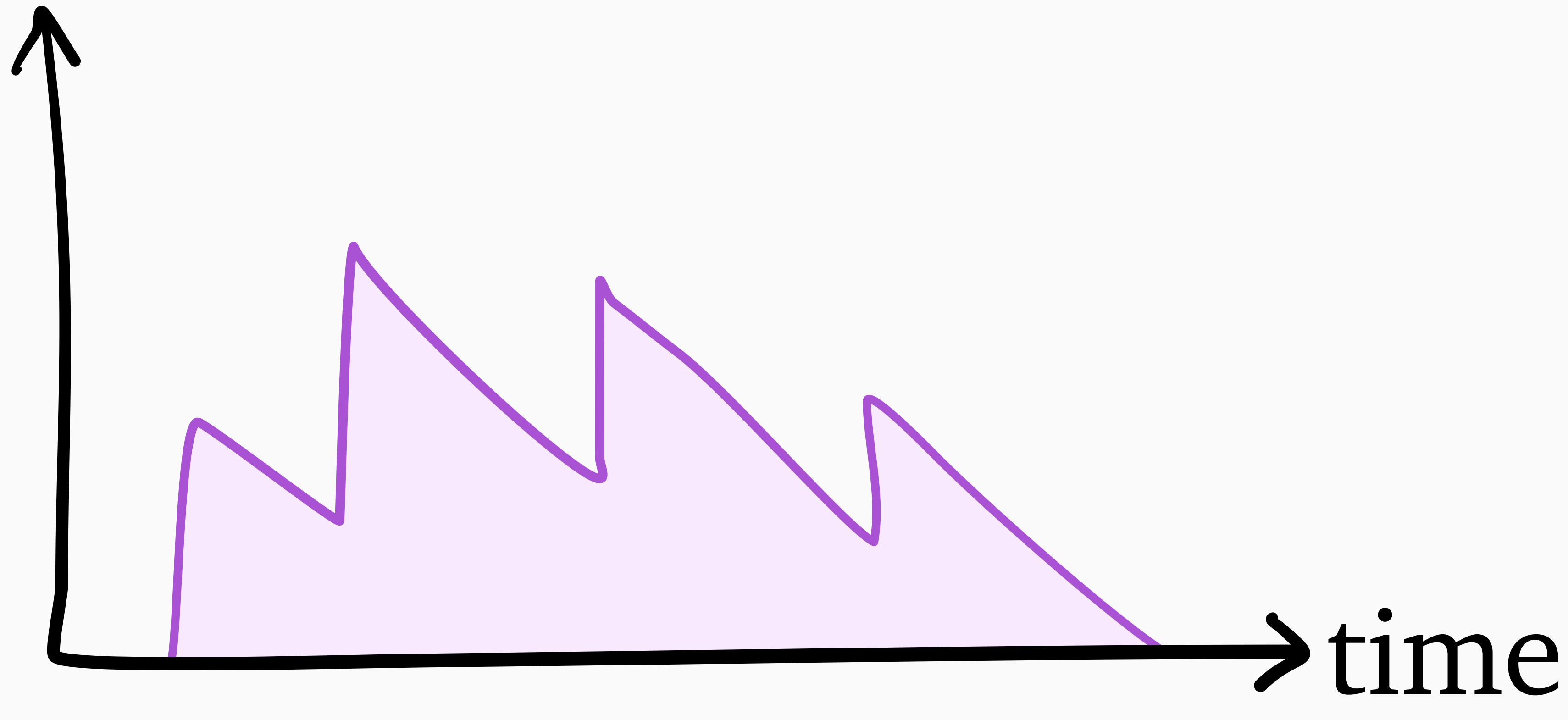




Step 2: $\mathbf{E}[W(r)]$ difference (SRPT)

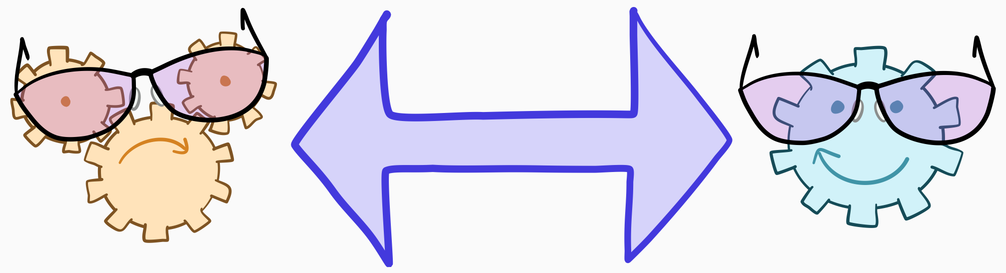


In steady-state system, for any f,
$\mathbf{E}[f(W(r))]$ constant w.r.t. time

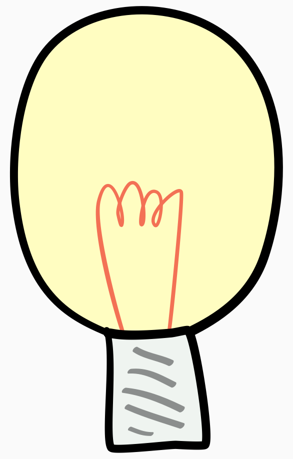


Theorem:

$$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] + \frac{\mathbf{E}[(1 - B_k(r))\, W_k(r)] + \lambda r \mathbf{P}[S > r]\, \mathbf{E}_r[W_k(r)]}{1 - \lambda \mathbf{E}[S\, \mathbb{1}(S \le r)]}$$

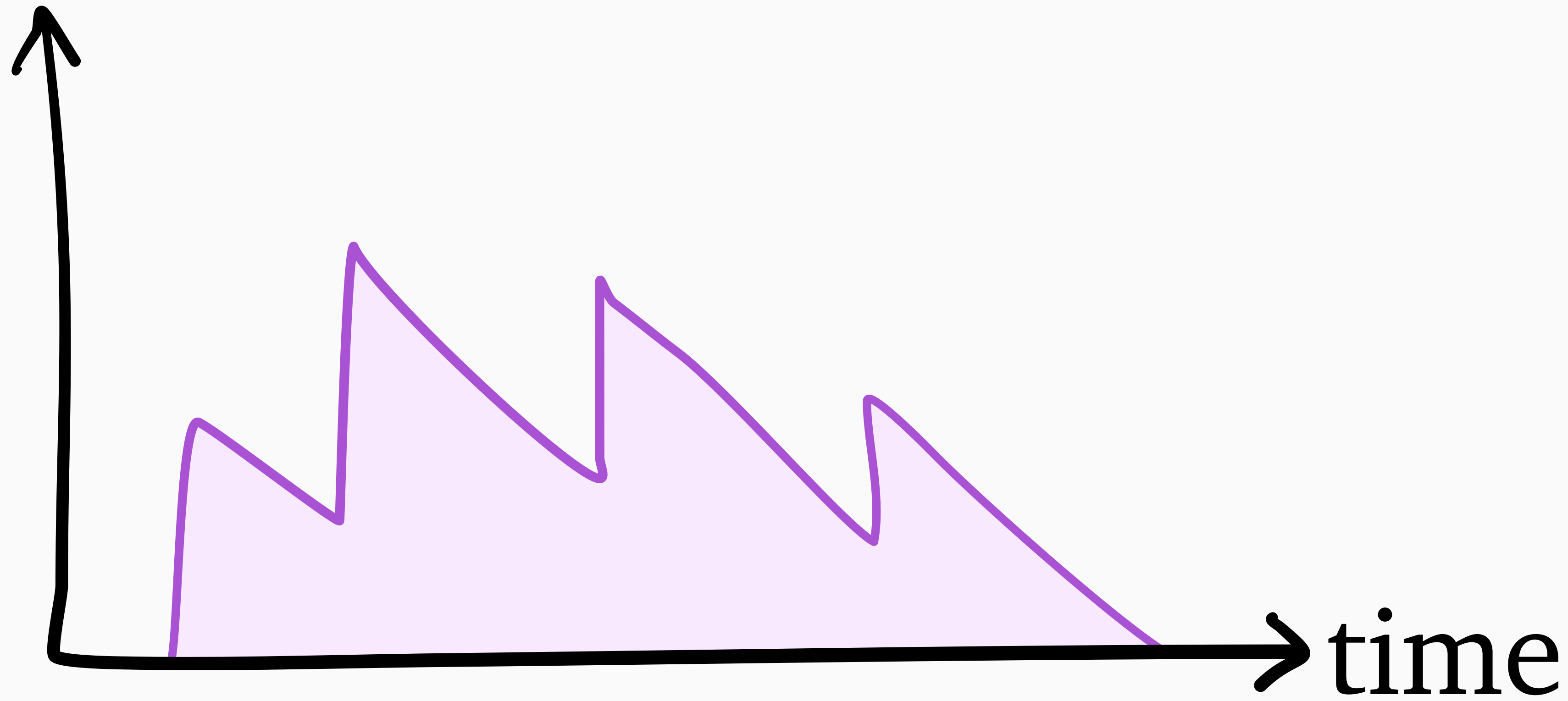


Step 2: $\mathbf{E}[W(r)]$ difference (SRPT)



In steady-state system, for any f,
$\mathbf{E}[f(W(r))]$ constant w.r.t. time

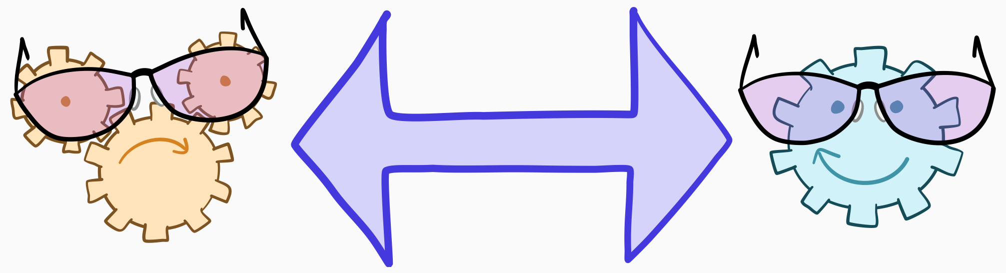
we use $f(w) = w^2$



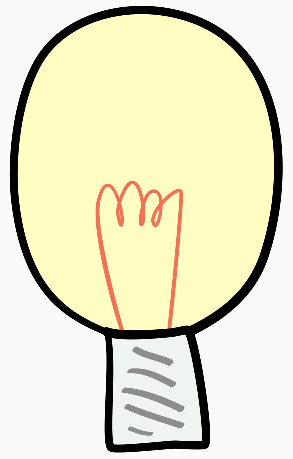
$B(r)$ = service rate on jobs of remaining size $\leq r$

Theorem:

$$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] + \frac{\mathbf{E}[(1 - B_k(r))\,W_k(r)] + \lambda r \mathbf{P}[S > r]\,\mathbf{E}_r[W_k(r)]}{1 - \lambda \mathbf{E}[S\,\mathbb{1}(S \leq r)]}$$

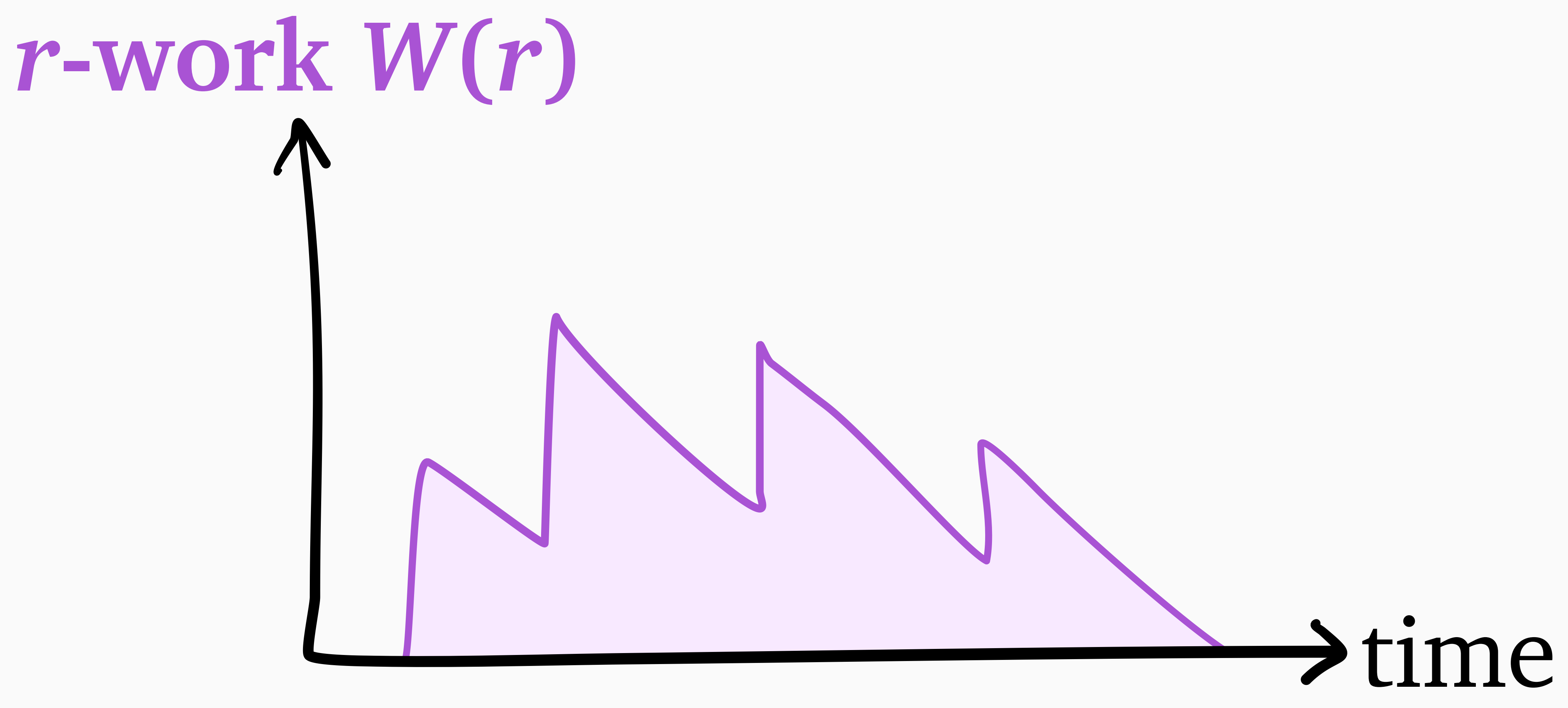


Step 2: $\mathbf{E}[W(r)]$ difference (SRPT)



In steady-state system, for any f, $\mathbf{E}[f(W(r))]$ constant w.r.t. time

we use $f(w) = w^2$

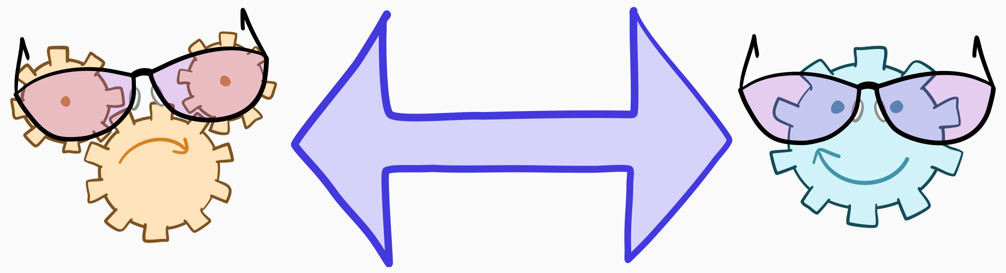


$B(r)$ = service rate on jobs of remaining size $\leq r$

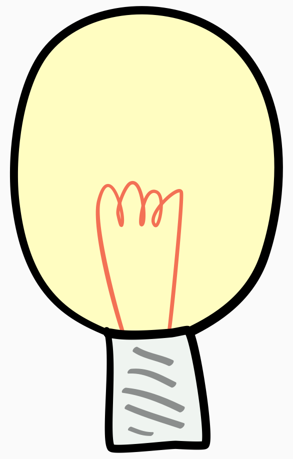
Theorem:

$$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] + \frac{\mathbf{E}[(1 - B_k(r))\, W_k(r)] + \lambda r \mathbf{P}[S > r]\, \mathbf{E}_r[W_k(r)]}{1 - \lambda \mathbf{E}[S\, \mathbb{1}(S \leq r)]}$$

"r-load"

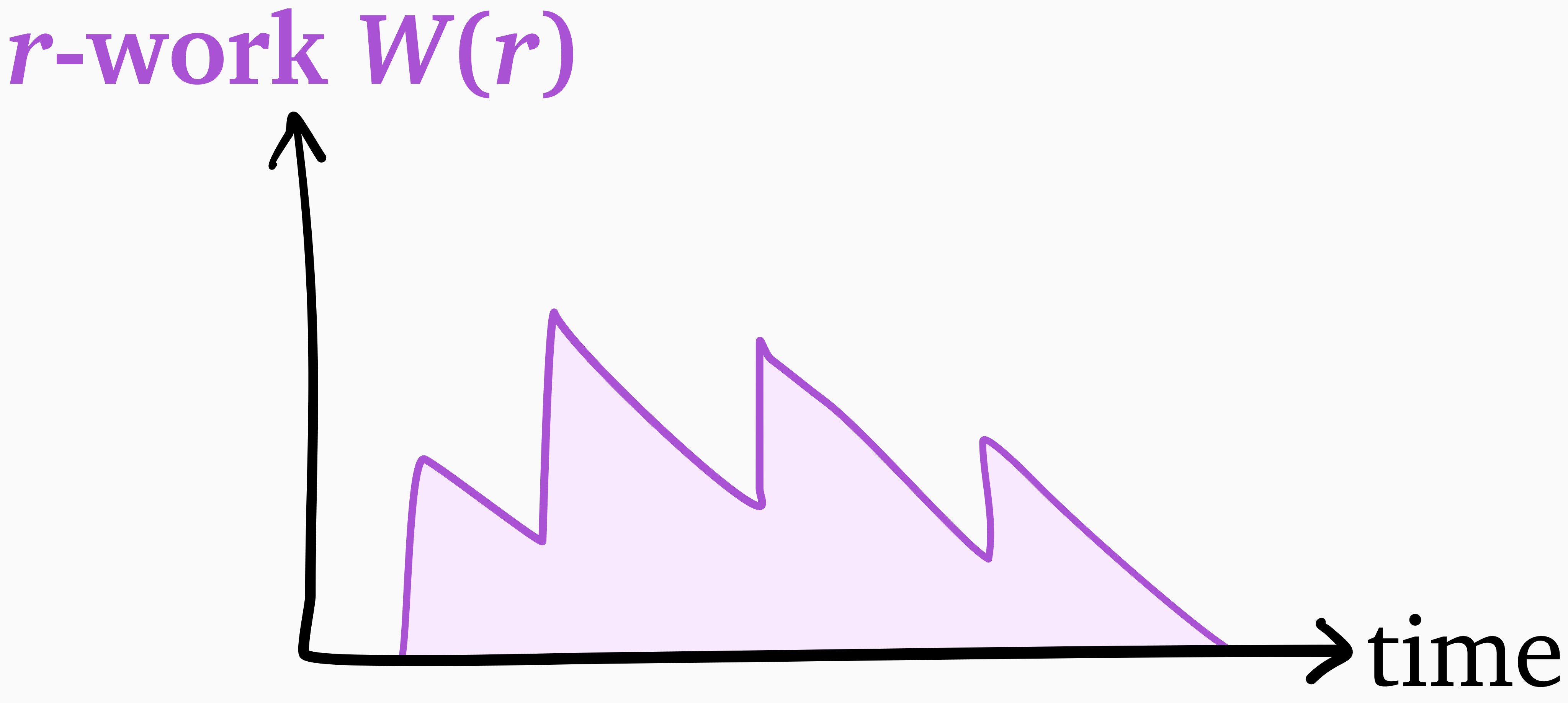


Step 2: $\mathbf{E}[W(r)]$ difference (SRPT)



In steady-state system, for any f,
$\mathbf{E}[f(W(r))]$ constant w.r.t. time

we use
$f(w) = w^2$



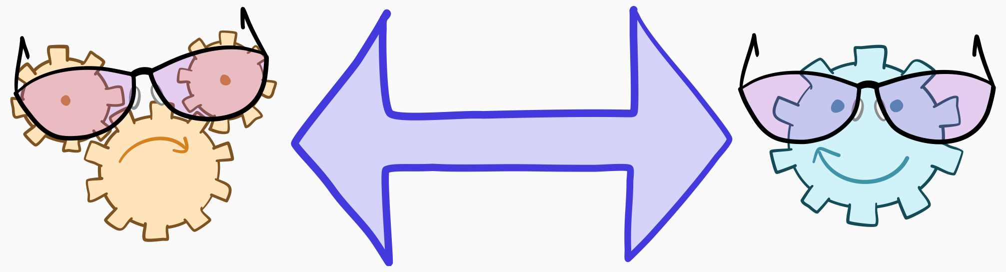
$B(r)$ = service rate on jobs of remaining size $\le r$

$\mathbf{E}_r[\cdot]$ samples whenever a job reaches remaining size r

Theorem:

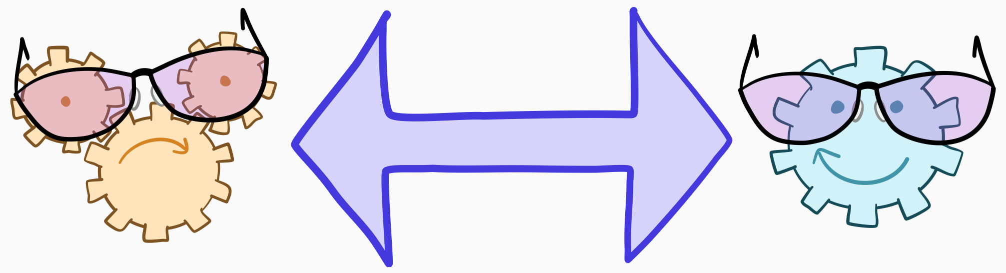
$$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] + \frac{\mathbf{E}[(1 - B_k(r))\, W_k(r)] + \lambda r \mathbf{P}[S > r]\, \mathbf{E}_r[W_k(r)]}{1 - \lambda \mathbf{E}[S\, \mathbb{1}(S \le r)]}$$

"r-load"



Step 2: … so what does it mean?

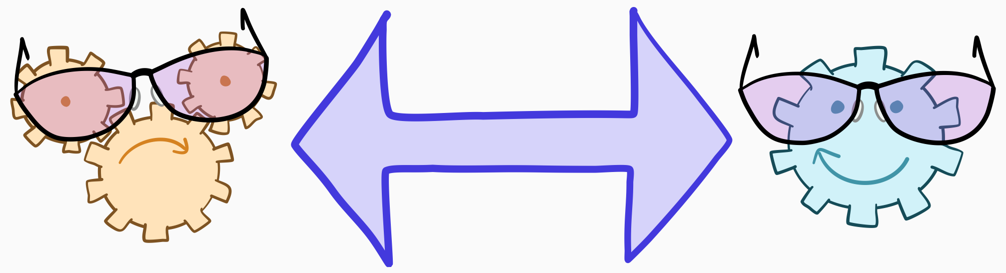
$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1 - B_k)W_k]}{1 - \rho}$$



Step 2: … so what does it mean?

Suppose $S \le s_{\max}$ with probability 1

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1 - B_k)W_k]}{1 - \rho}$$

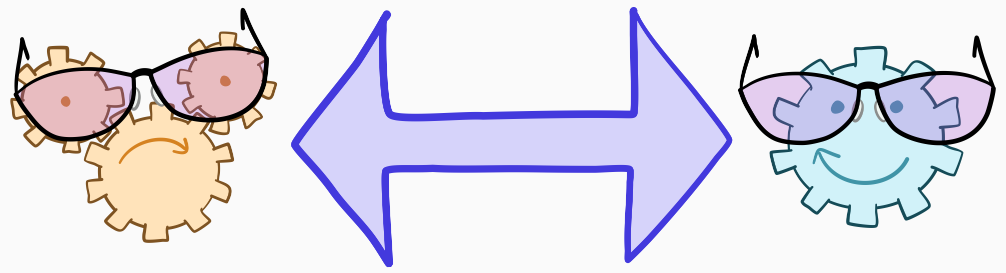


Step 2: … so what does it mean?

Suppose $S \leq s_{\max}$ with probability 1

$\mathbf{E}[B] = \rho$

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1 - B_k)W_k]}{1 - \rho}$$



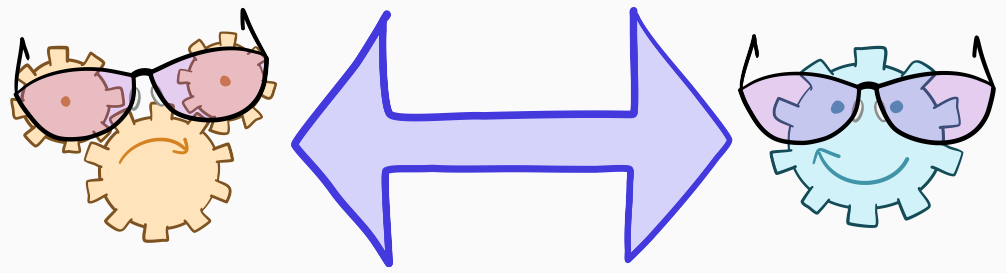
Step 2: … so what does it mean?

Suppose $S \leq s_{\max}$ with probability 1

$\mathbf{E}[B] = \rho$

$\leq (k-1)\, s_{\max}$

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1 - B_k) W_k]}{1 - \rho}$$



Step 2: … so what does it mean?

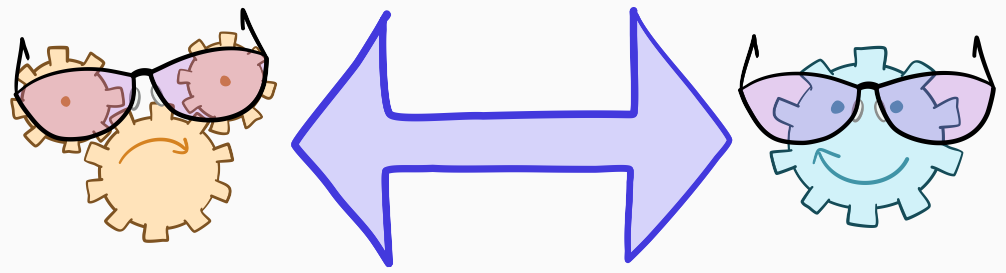
Suppose $S \leq s_{\max}$ with probability 1

$\mathbf{E}[B] = \rho$

$\leq (k-1)\, s_{\max}$

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1 - B_k) W_k]}{1 - \rho}$$

$$\leq \mathbf{E}[W_1] + (k-1)\, s_{\max}$$



Step 2: … so what does it mean?

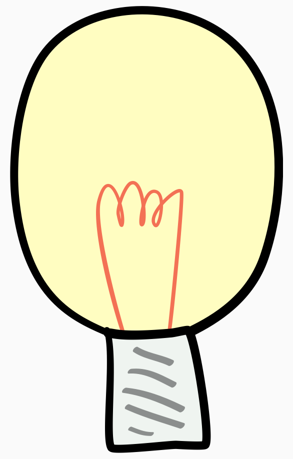
Suppose $S \le s_{\max}$ with probability 1

$\mathbf{E}[B] = \rho$

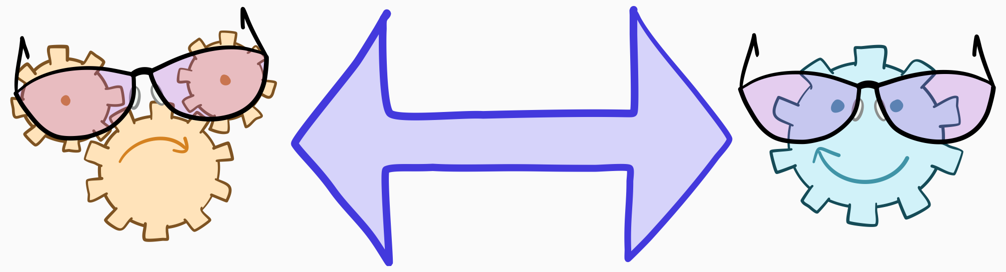
$\le (k-1)\, s_{\max}$

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$

$$\le \mathbf{E}[W_1] + (k-1)\, s_{\max}$$



SRPT: job's ***r*-work** is always $\le$ ***r***

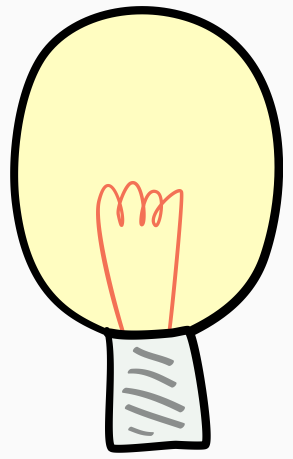


Step 2: … so what does it mean?

Suppose $S \leq s_{\max}$ with probability 1

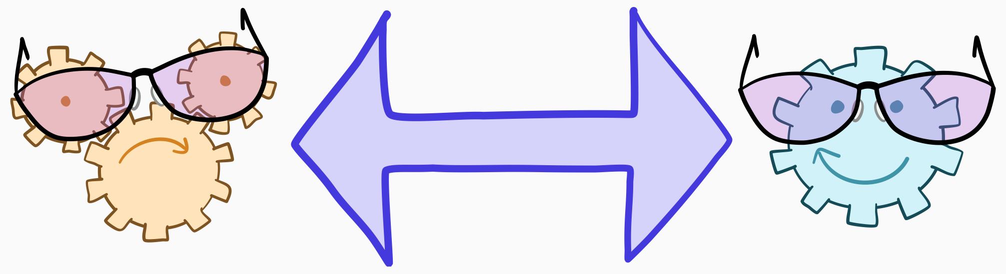
$\mathbf{E}[B] = \rho$ $\quad$ $\leq (k-1)\, s_{\max}$

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$
$$\leq \mathbf{E}[W_1] + (k-1)\, s_{\max}$$



SRPT: job's ***r*-work** is always $\leq r$

$$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] + \text{“}r\text{-work of } \leq k-1 \text{ jobs”}$$
$$\leq \mathbf{E}[W_1(r)] + (k-1)\, r$$



Step 2: … so what does it mean?

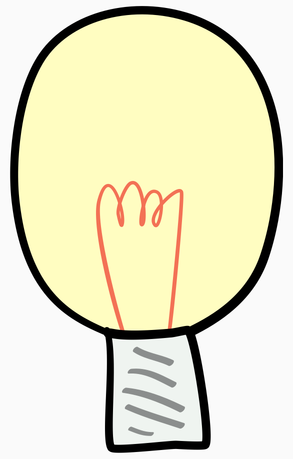
Suppose $S \le s_{\max}$ with probability 1

$\mathbf{E}[B] = \rho$

$\le (k-1)\, s_{\max}$

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1 - B_k) W_k]}{1 - \rho}$$

$$\le \mathbf{E}[W_1] + (k-1)\, s_{\max}$$

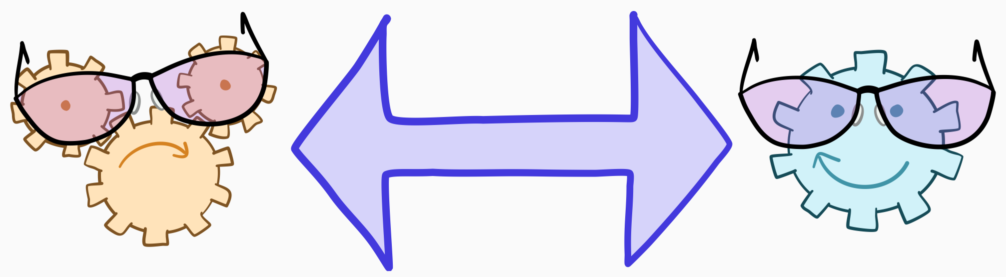


SRPT: job's ***r*-work** is always $\le r$

$$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] + \text{“}r\text{-work of} \le k-1 \text{ jobs”}$$

$$\le \mathbf{E}[W_1(r)] + (k-1)\, r$$

see paper for
better bound



Step 2: … so what does it mean?

Suppose $S \le s_{\max}$ with probability 1

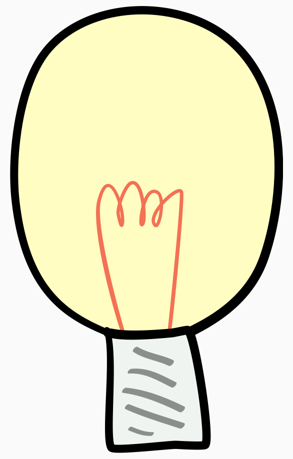
$\mathbf{E}[B] = \rho$

$\le (k-1)\, s_{\max}$

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$

$$\le \mathbf{E}[W$$

still true under Gittins,
but only in expectation

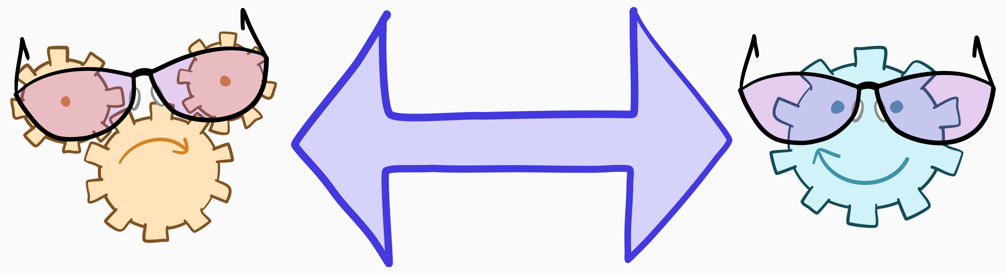


SRPT: job's *r*-work is always $\le r$

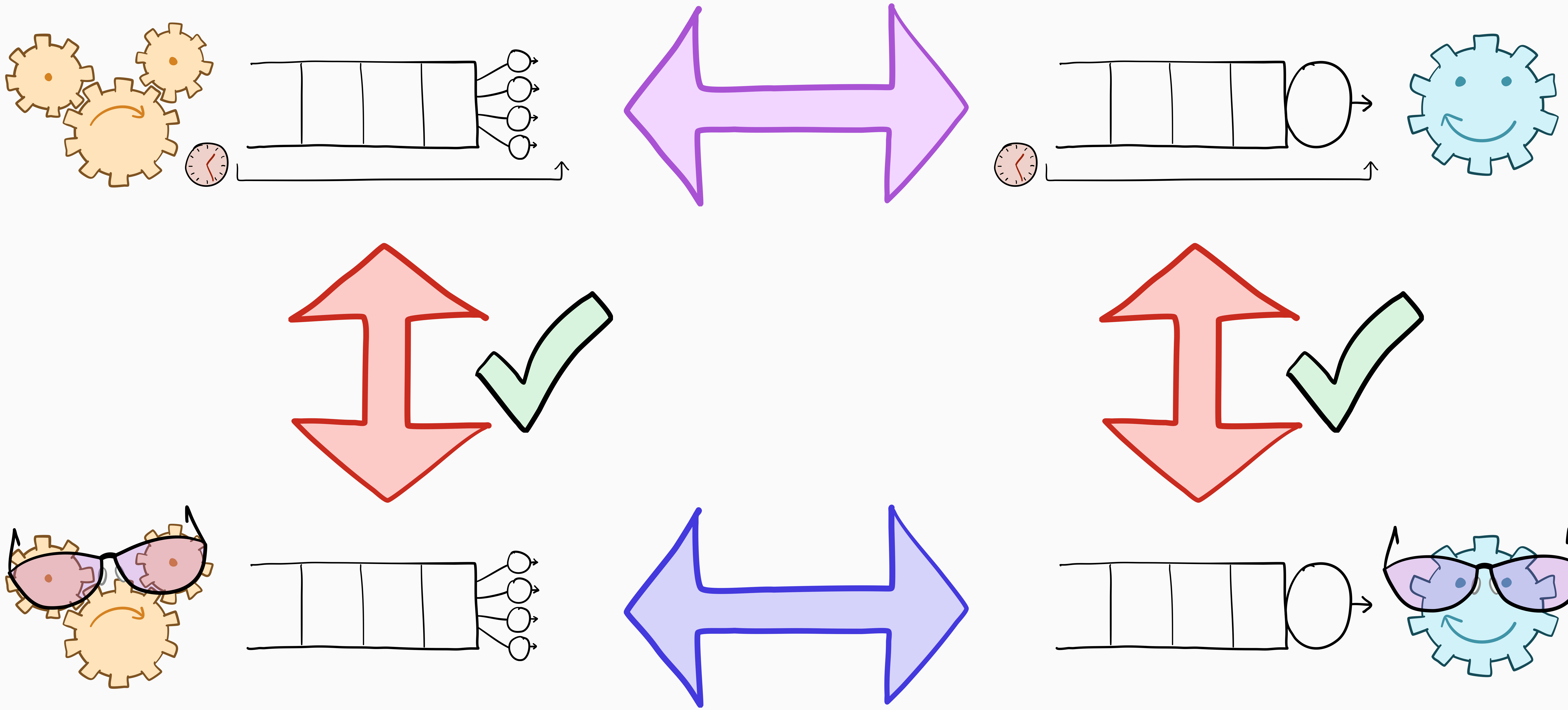
$$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] + \text{“}r\text{-work of} \le k-1 \text{ jobs”}$$

$$\le \mathbf{E}[W_1(r)] + (k-1)\,r$$

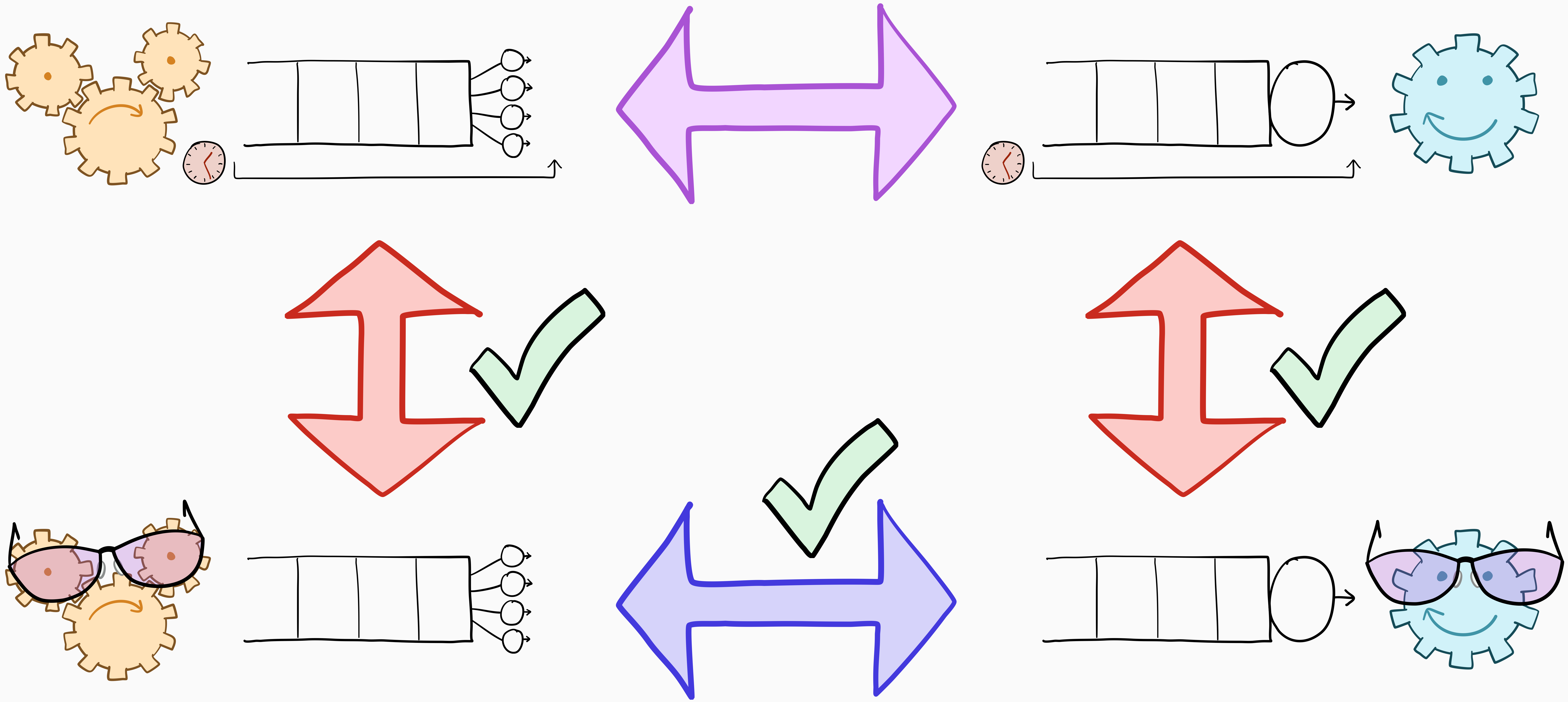
see paper for
better bound



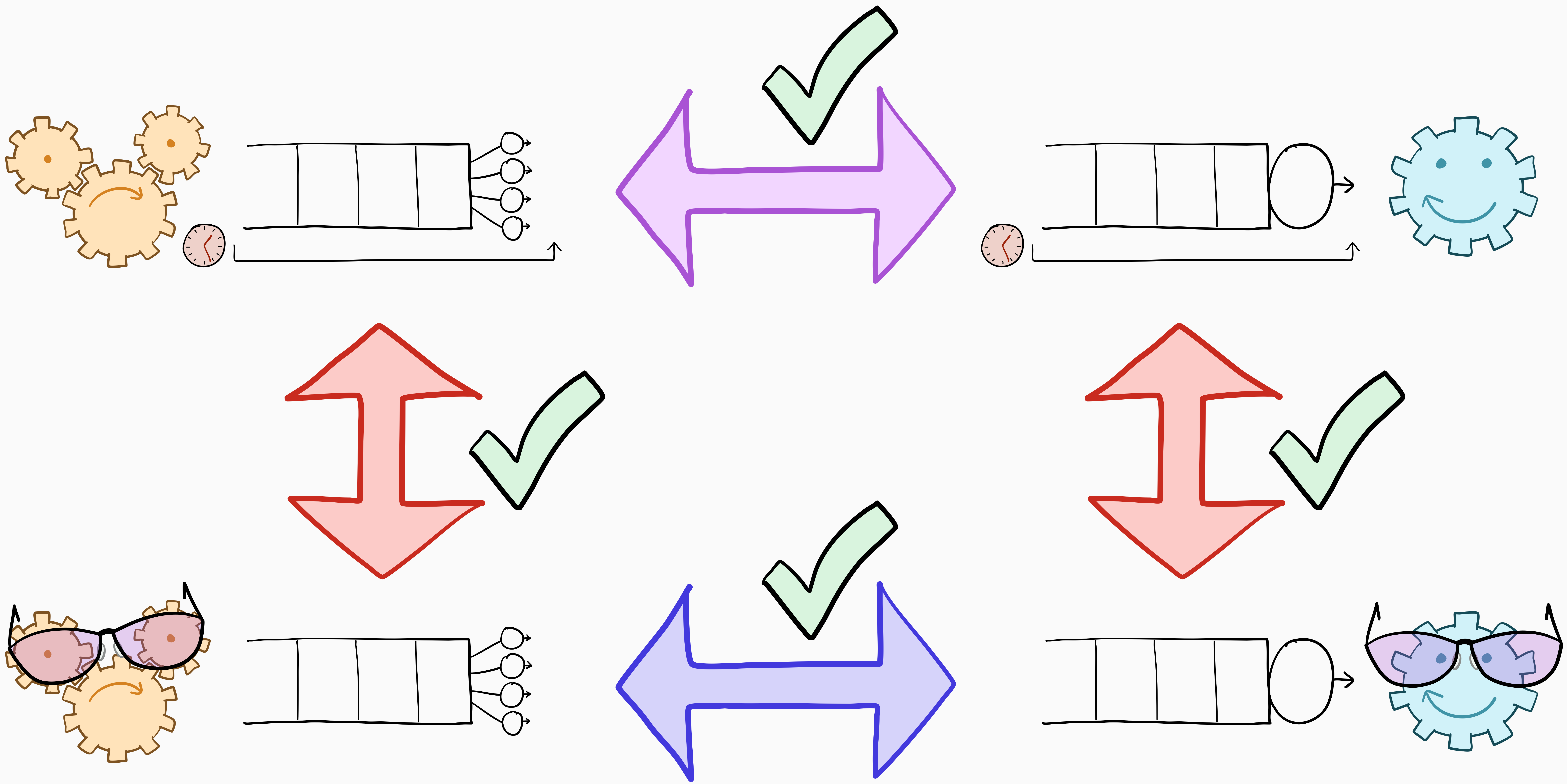
Response time via *r*-work



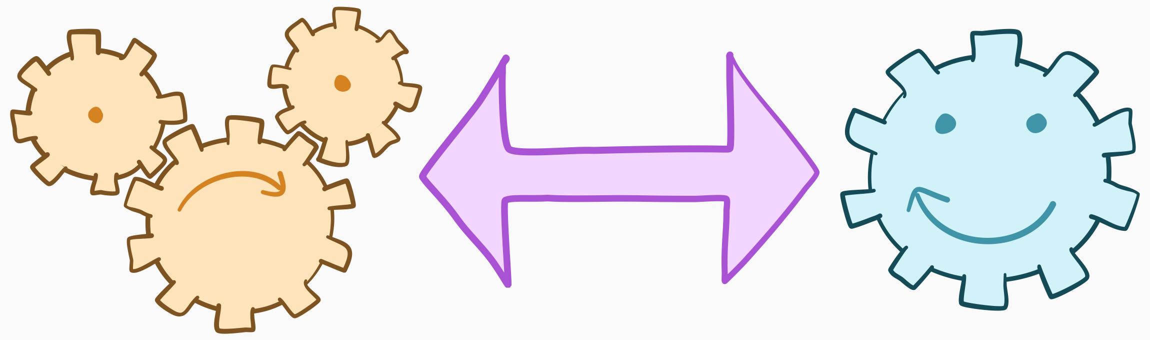
Response time via *r-work*



Response time via *r*-work

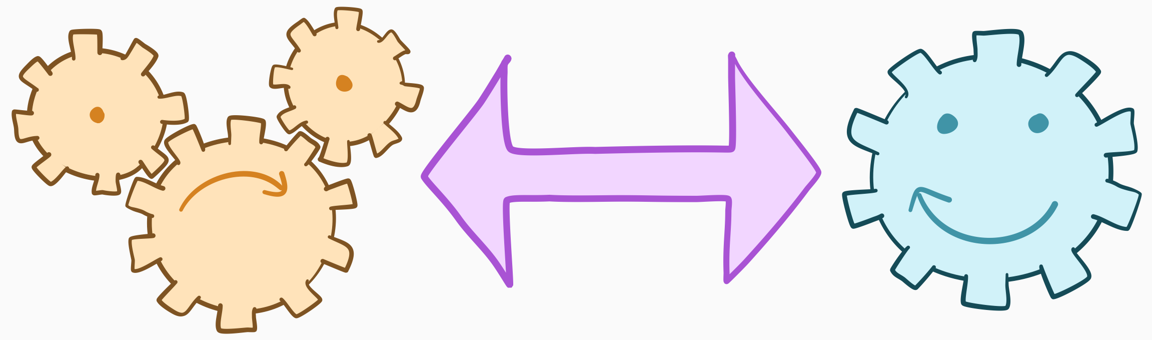


Classifying coupling techniques



	A. Every sample path	B. Steady-state distribution
1. More information	A1 BIG online knapsack (via compensated coupling)	B1
2. Fewer constraints	A2 M/M/k vs. M/M/1 Online knapsack (via constraints-to-costs)	B2 State-space collapse (load balancing, switch scheduling)
3. Simpler dynamics	A3 SIS epidemics Queues with redundancy	B3 Stein's method

Classifying coupling techniques



	A. Every sample path	B. Steady-state distribution
1. More information	A1 BIG online knapsack (via compensated coupling)	B1
2. Fewer constraints	A2 M/M/*k* vs. M/M/1 Online knapsack (via constraints-to-costs)	B2 State-space collapse (load balancing, switch scheduling) SRPT/Gittins in the M/G/*k*
3. Simpler dynamics	A3 SIS epidemics Queues with redundancy	B3 Stein's method

Overview

Part 1

Survey 1:
Sample-Path Coupling

In-Depth Study 1:
Online Resource Allocation

Part 2

Survey 2:
Steady-State Coupling

In-Depth Study 2:
Gittins in the M/G/k

Overview

✓ Part 1

Survey 1:
Sample-Path Coupling

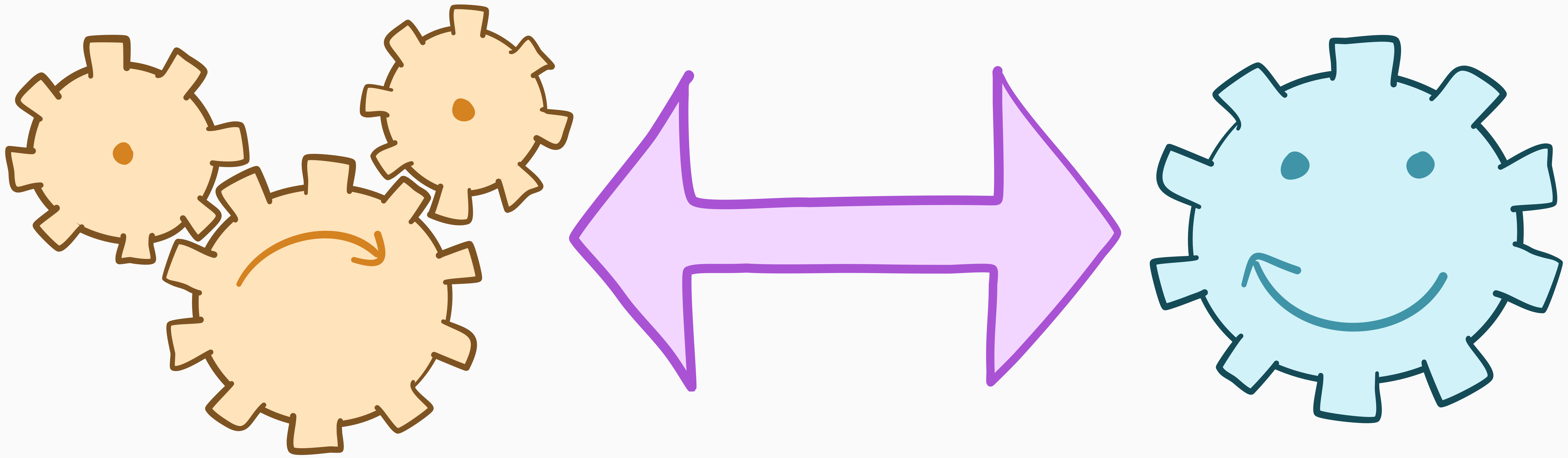
In-Depth Study 1:
Online Resource Allocation

✓ Part 2

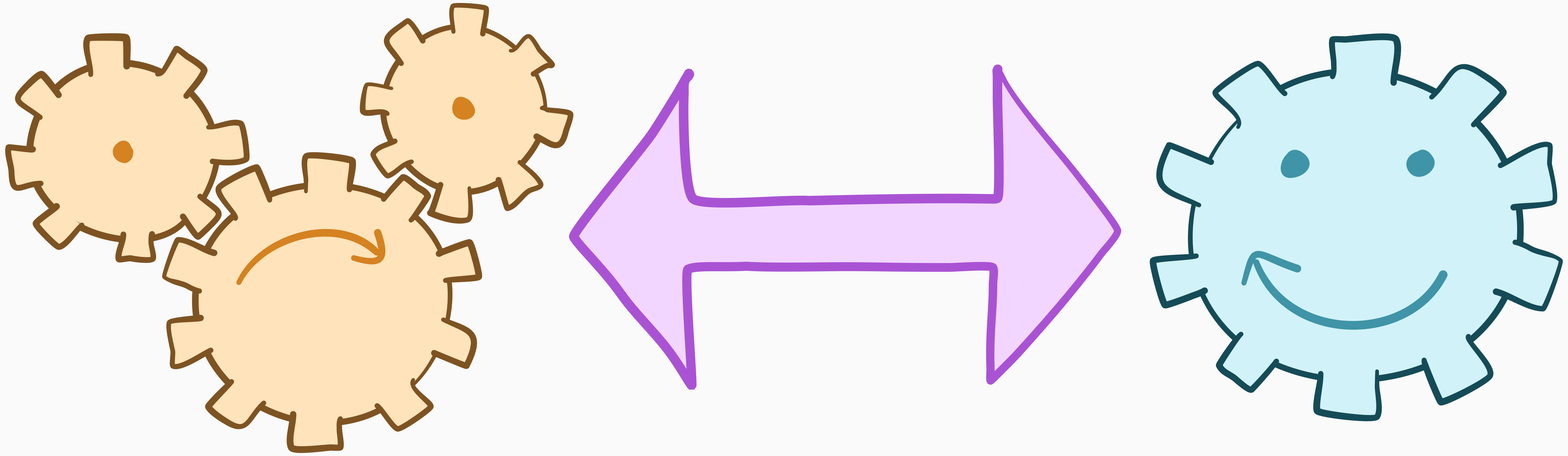
Survey 2:
Steady-State Coupling

In-Depth Study 2:
Gittins in the M/G/k

Conclusion



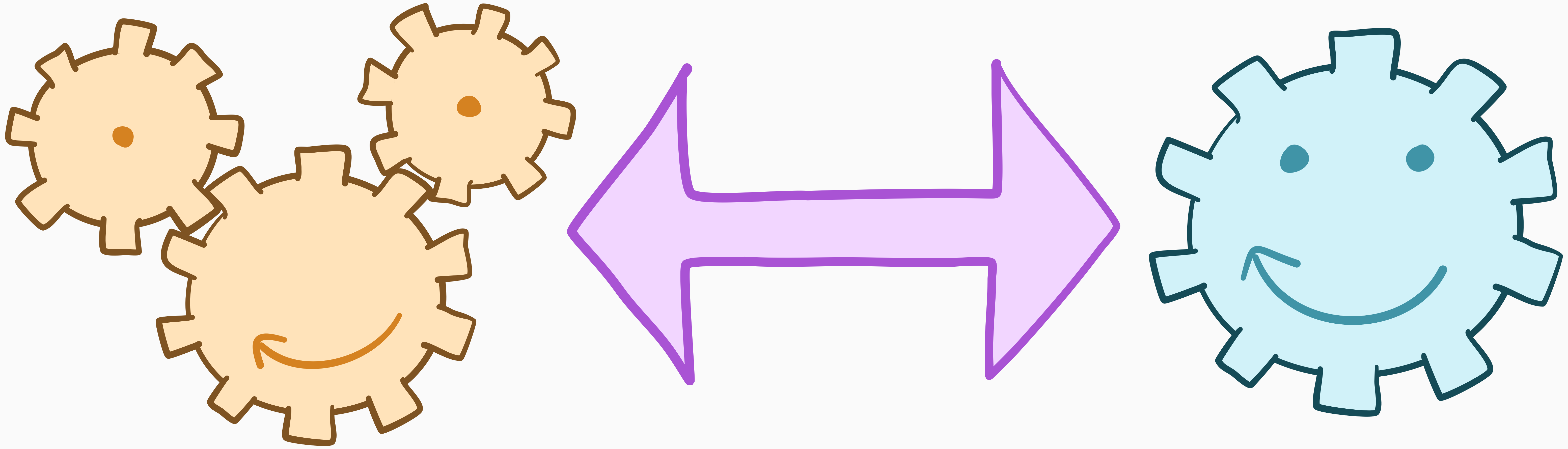
Conclusion



Ziv's email: `zscully@cs.cmu.edu`

Sid's email: `sbanerjee@cornell.edu`

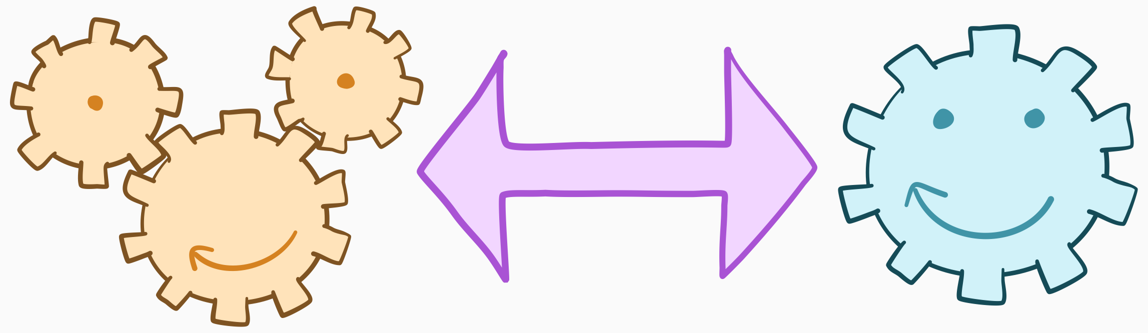
Conclusion



Ziv's email: `zscully@cs.cmu.edu`

Sid's email: `sbanerjee@cornell.edu`

Classifying coupling techniques



	A. Every sample path	B. Steady-state distribution
1. More information	A1 BIG online knapsack (via compensated coupling)	B1
2. Fewer constraints	A2 M/M/*k* vs. M/M/1 Online knapsack (via constraints-to-costs)	B2 State-space collapse (load balancing, switch scheduling) SRPT/Gittins in the M/G/*k*
3. Simpler dynamics	A3 SIS epidemics Queues with redundancy	B3 Stein's method