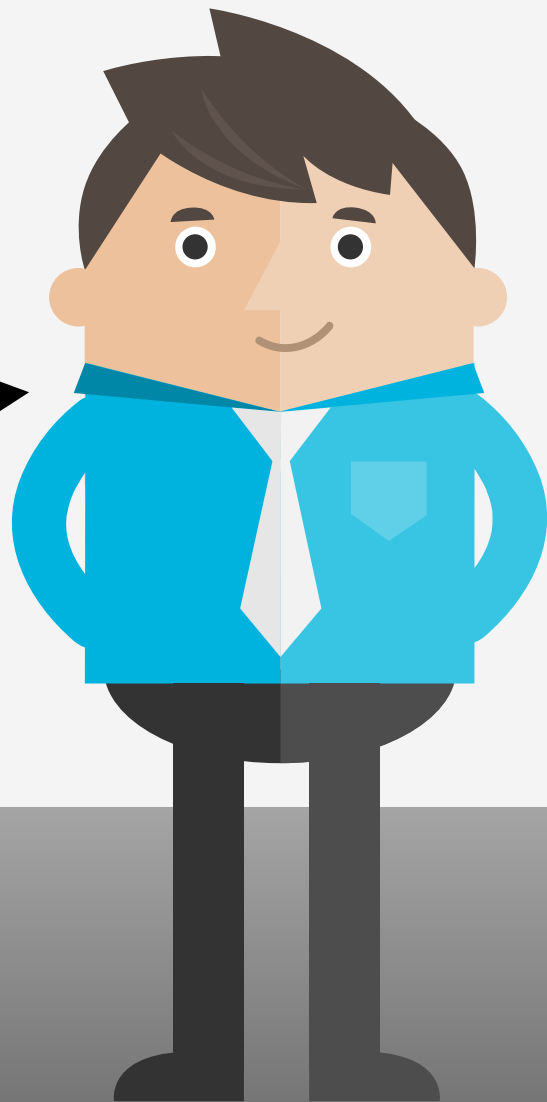
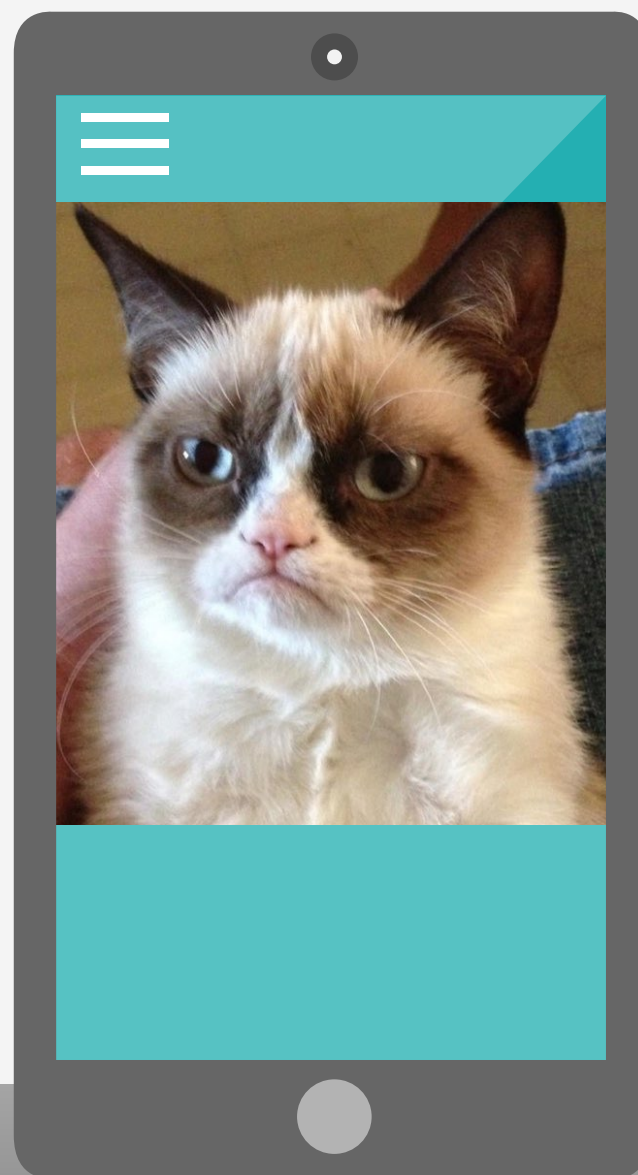
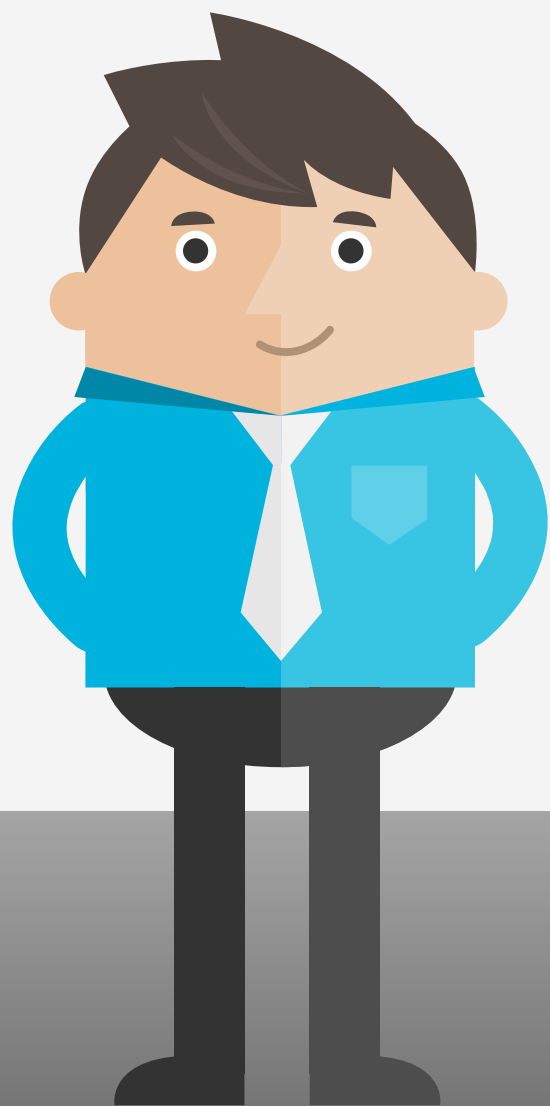
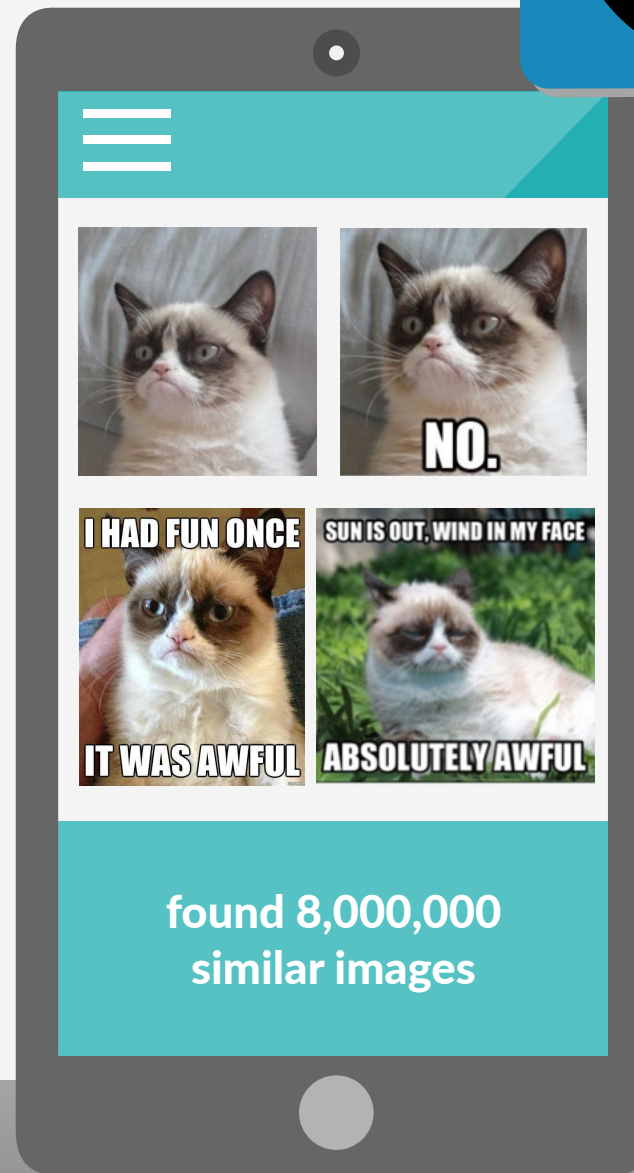
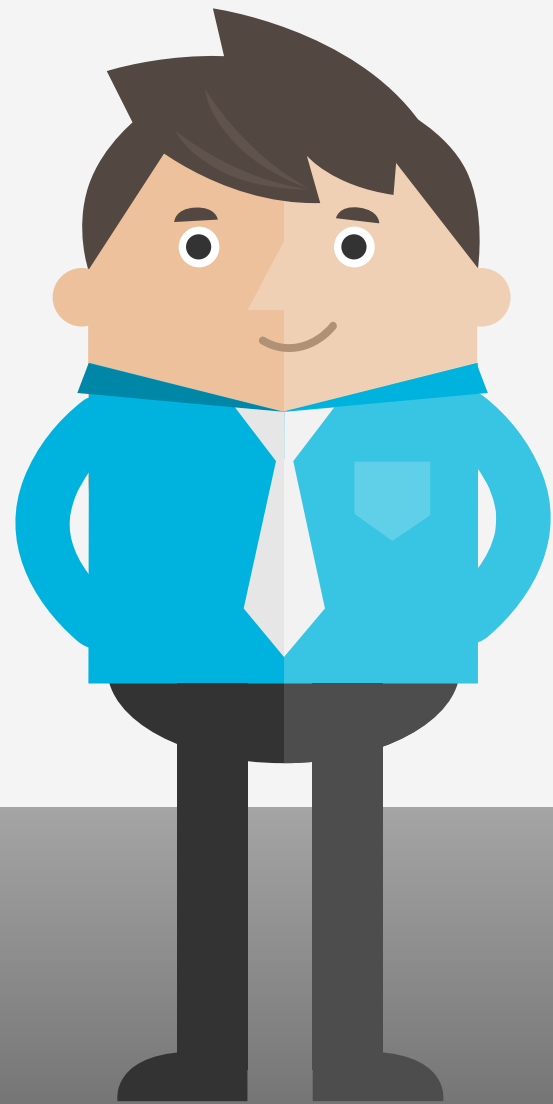


This is Bob.



Bob has an idea
for a startup.



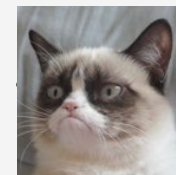
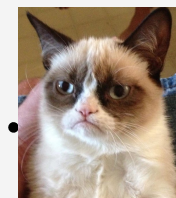
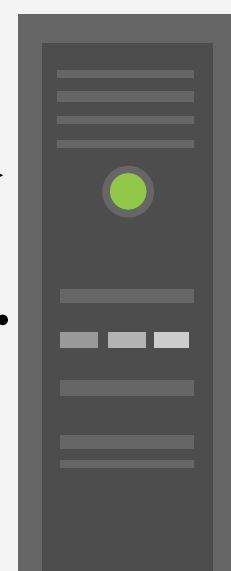
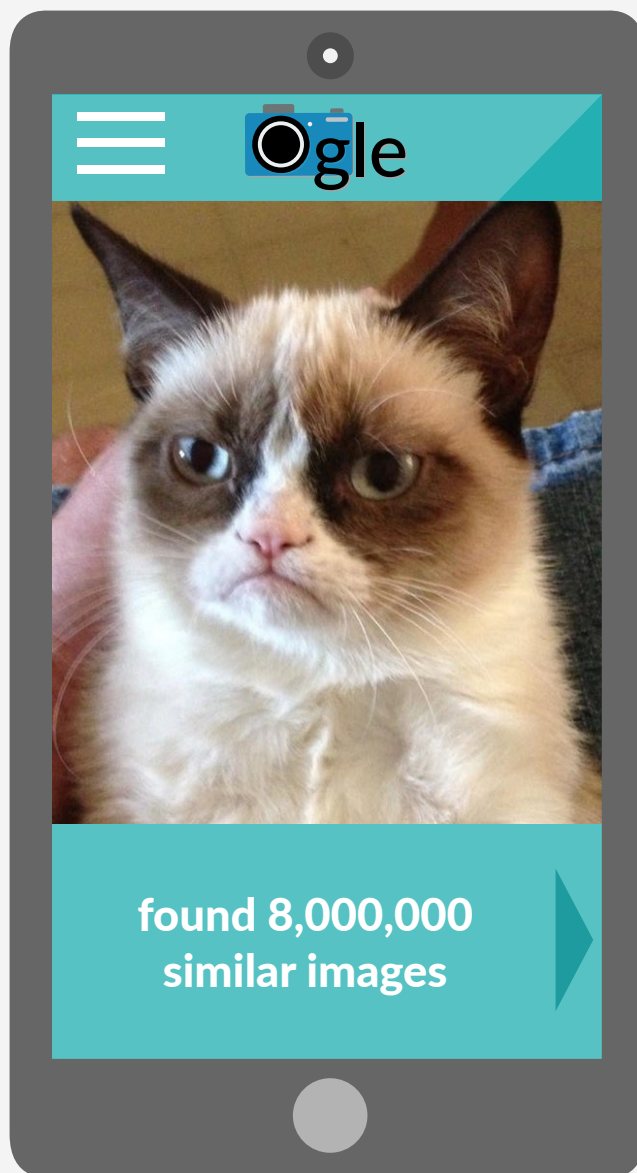


The Prototype gle

Take a picture

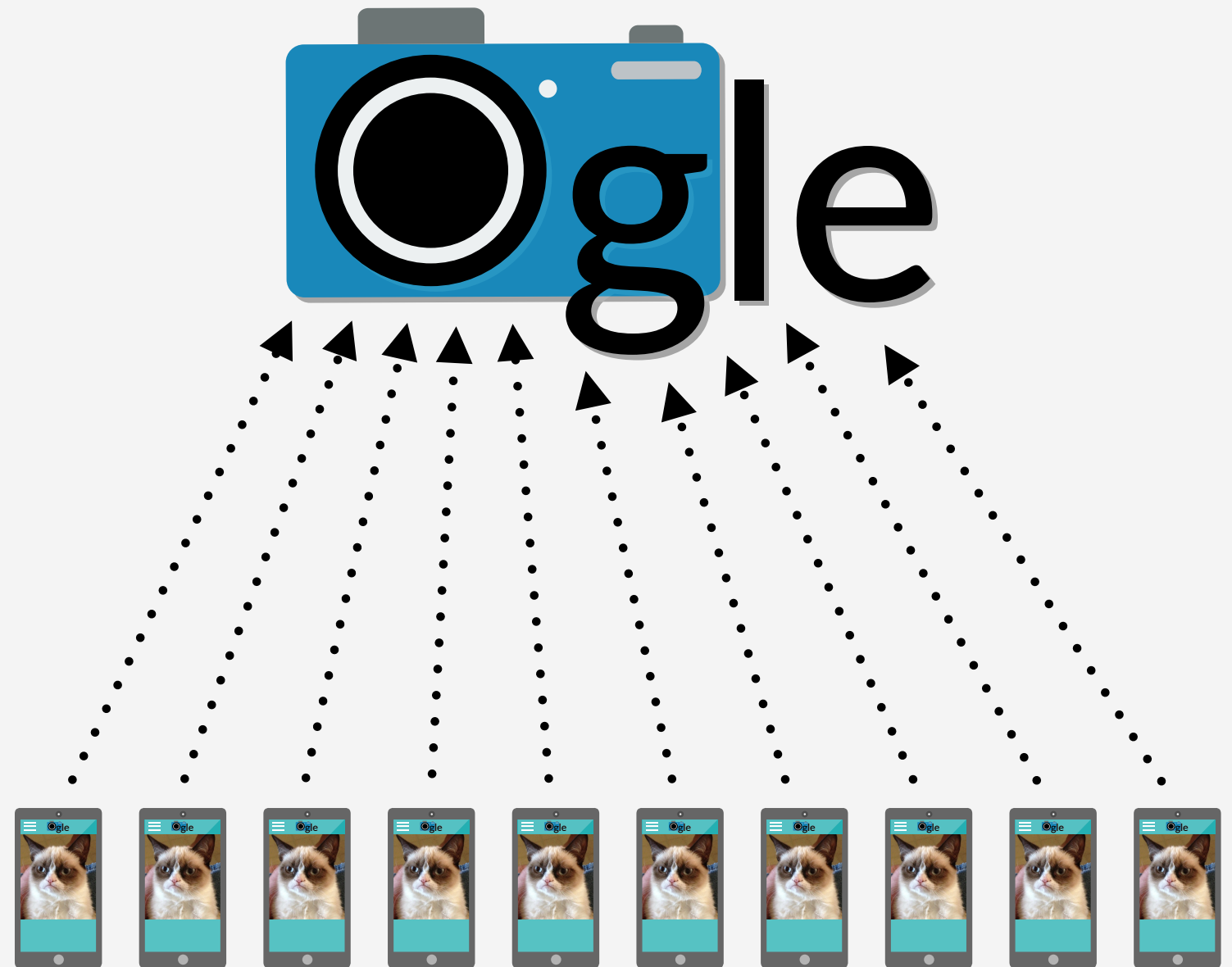
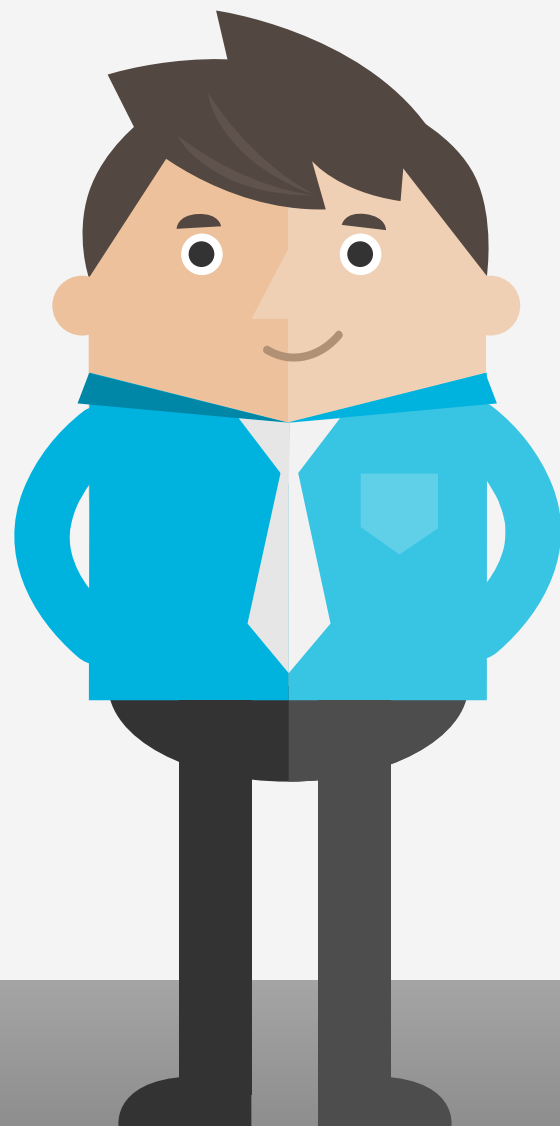
Add it to
the database

Send it to Ogle

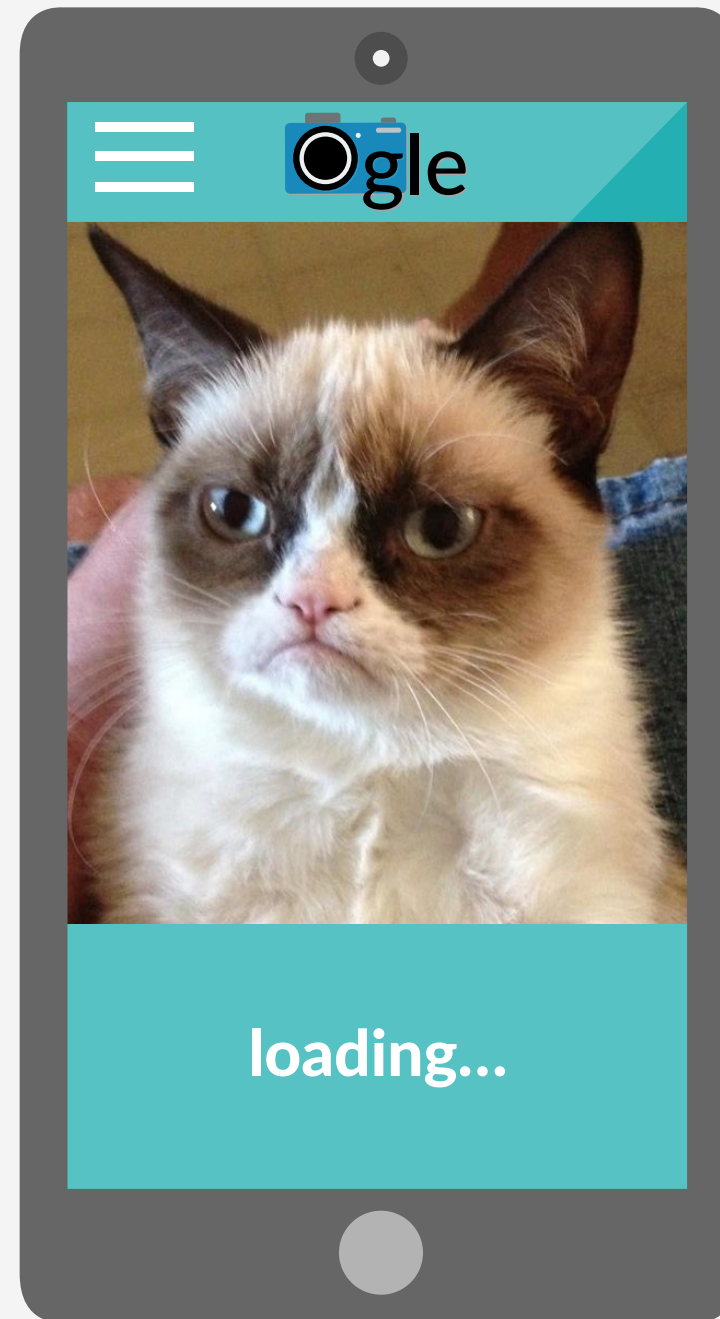
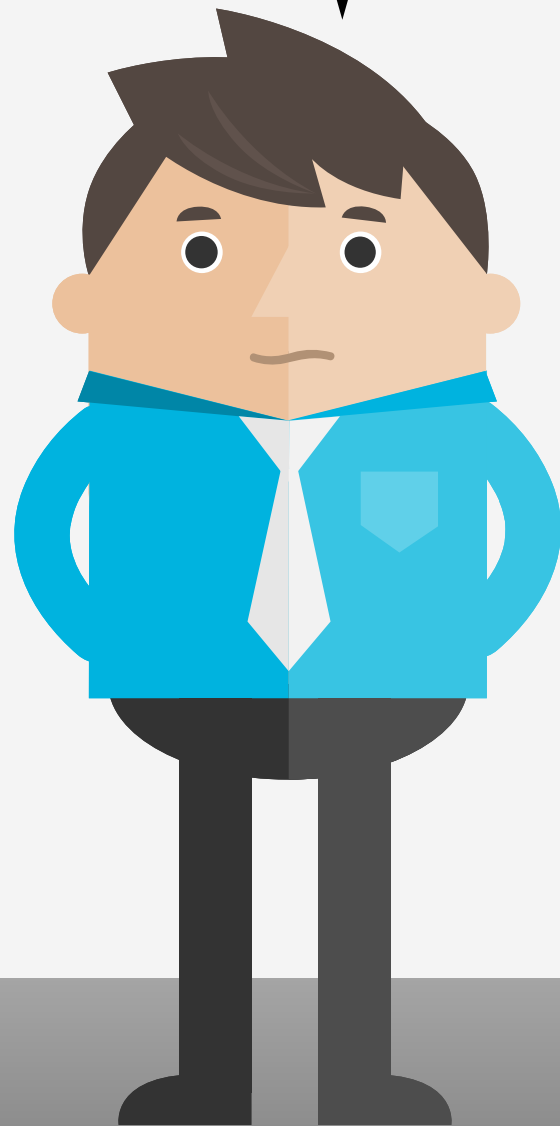


Send results

Find similar pictures



Ogle is too slow!



How can Bob make Ogle faster?

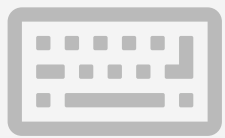
“Optimizations”

Use a faster language

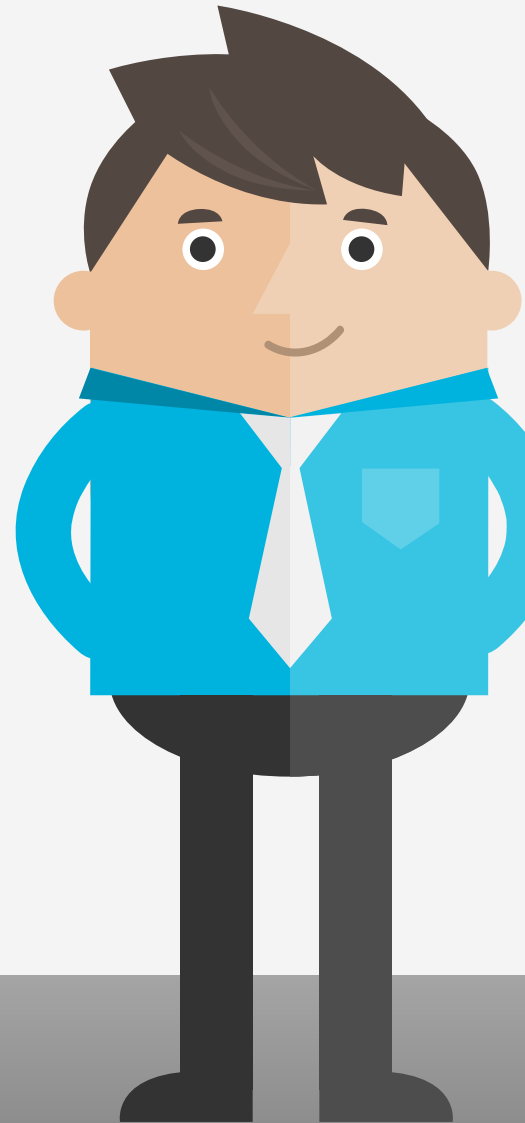
Use a faster algorithm

Parallelize

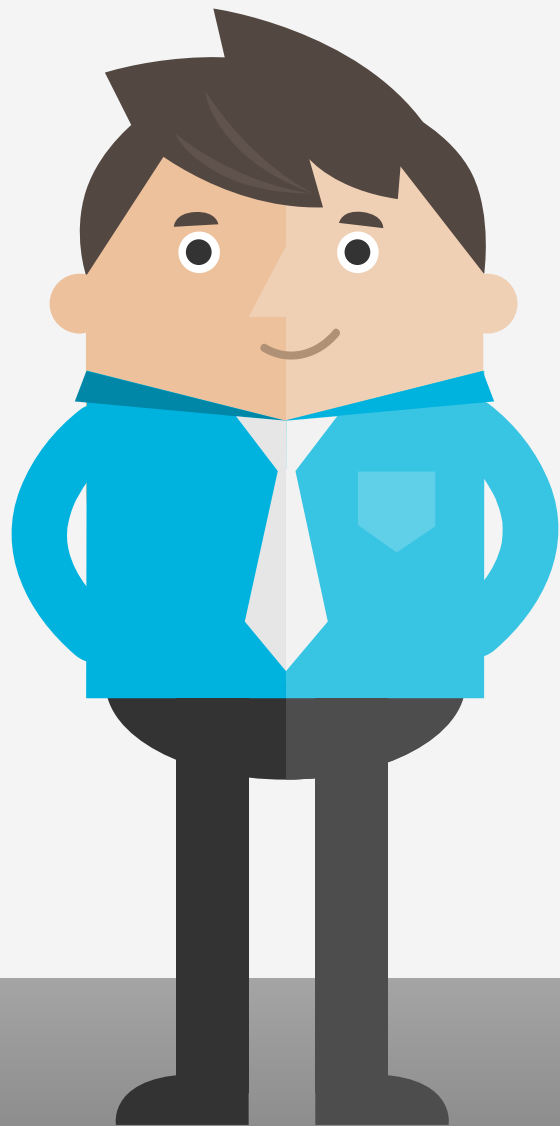
 Why we have to write parallel programs

 How to write parallel programs

 Why parallel programming is hard



Running a program is like running errands



Bob needs a car to run errands



Bob needs a car to run errands

Bob can run errands faster with a faster car



Bob needs a car to run errands

Bob can run errands faster with a faster car



Bob needs a car to run errands

Bob can run errands faster with a faster car

We were promised jetpacks.



Bob needs a car to run errands

Bob can run errands faster with a faster car

We were promised jetpacks.

No such luck. Instead...

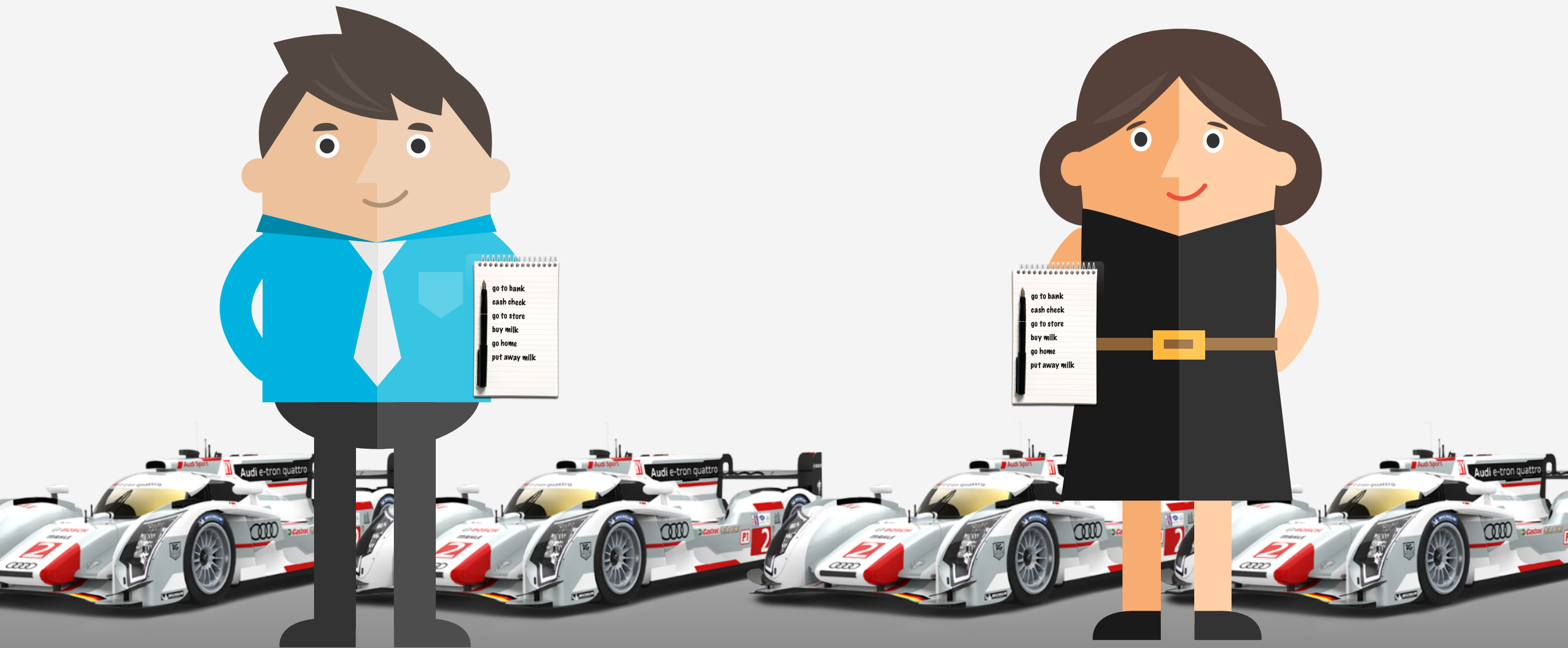
Bob gets a deal on a four-pack of cars.

Can Bob run his errands faster with *four* cars?

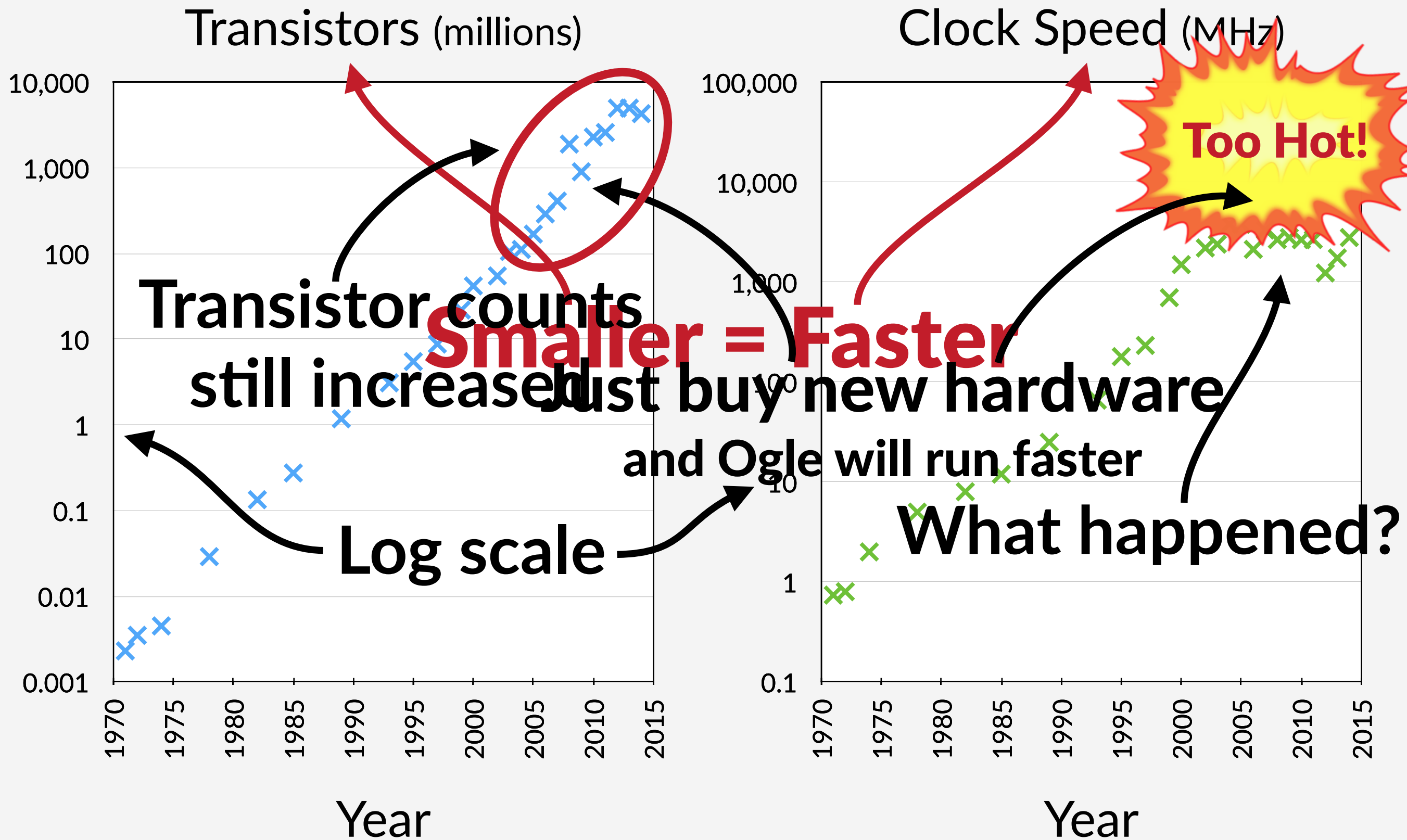


Bob and Alice can split up errands

Now they can use two cars instead of one

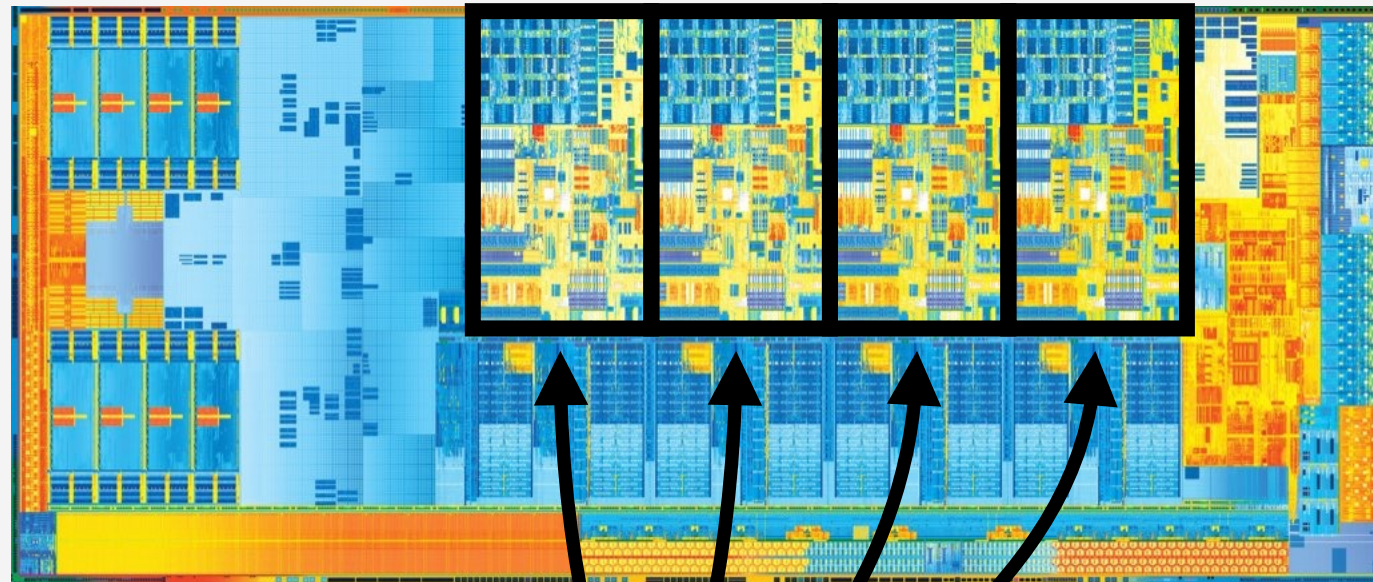


Performance used to be easy



Where do all those transistors go?

Instead of faster processors, we just get more.



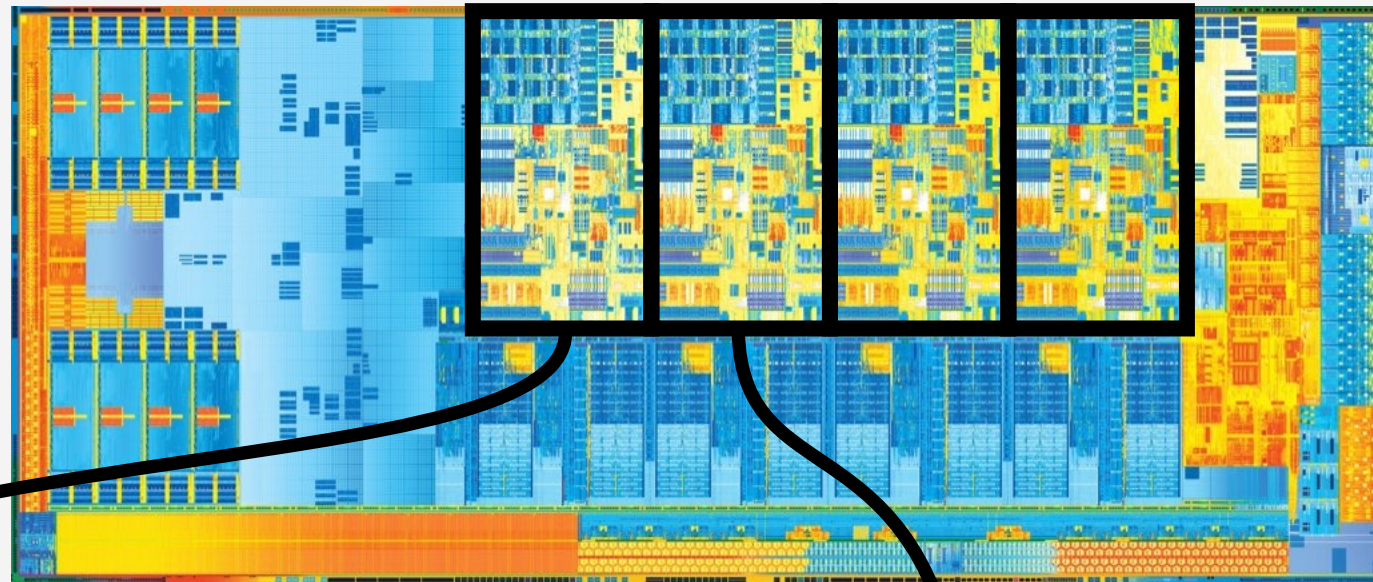
Four cores

Really just four separate processors

Individually, they aren't any faster than a Pentium 4 from 2005

Where do all those transistors go?

Instead of faster processors, we just get more.



```
total = 0
```

```
for i in range(0,10):
```

```
    total += i
```

```
print total
```

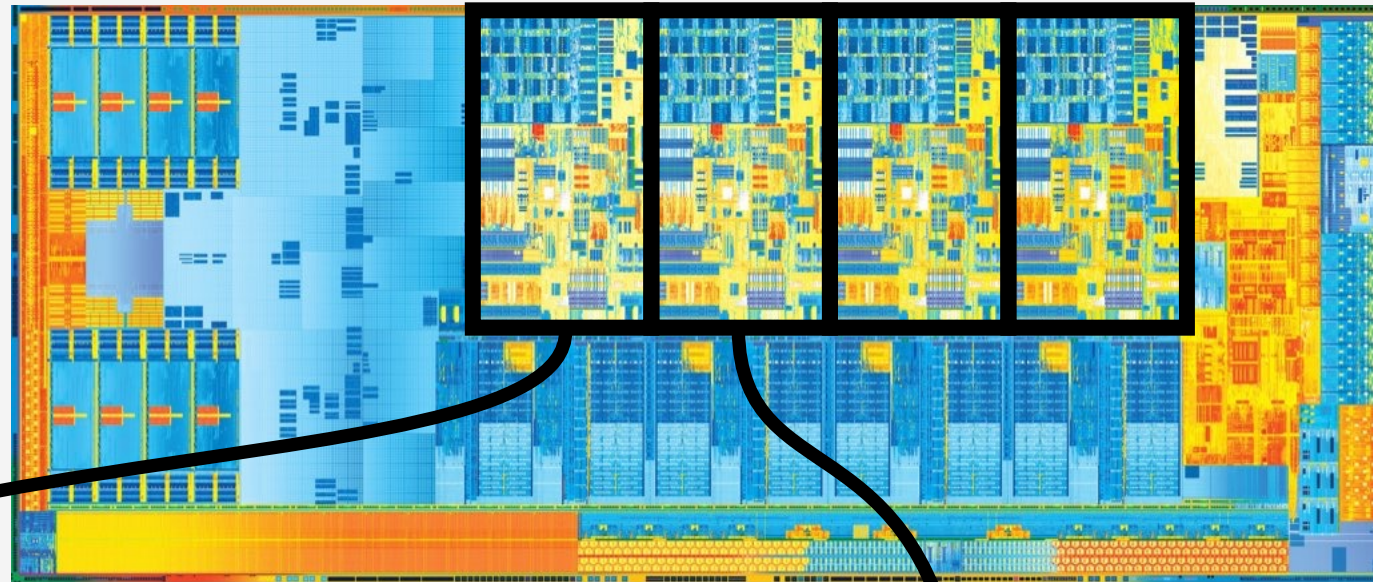
```
product = 0
```

```
for i in range(0,10):
```

```
    product *= i
```

```
print product
```

Where do all those transistors go?



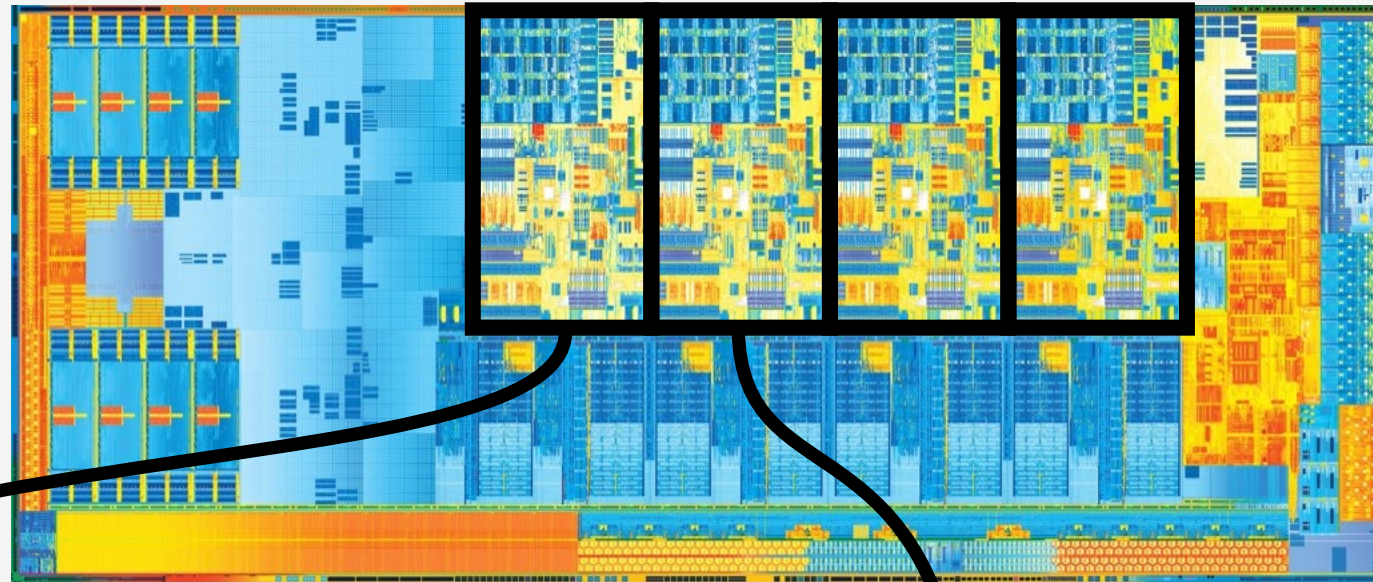
```
total = 0
for i in range(0,10):
    total += i
print total
```

```
product = 0
for i in range(0,10):
    product *= i
print product
```

One core runs a sequence of steps
starting at the beginning of your program.

to use more cores, you have to provide more sequences of steps

Where do all those transistors go?

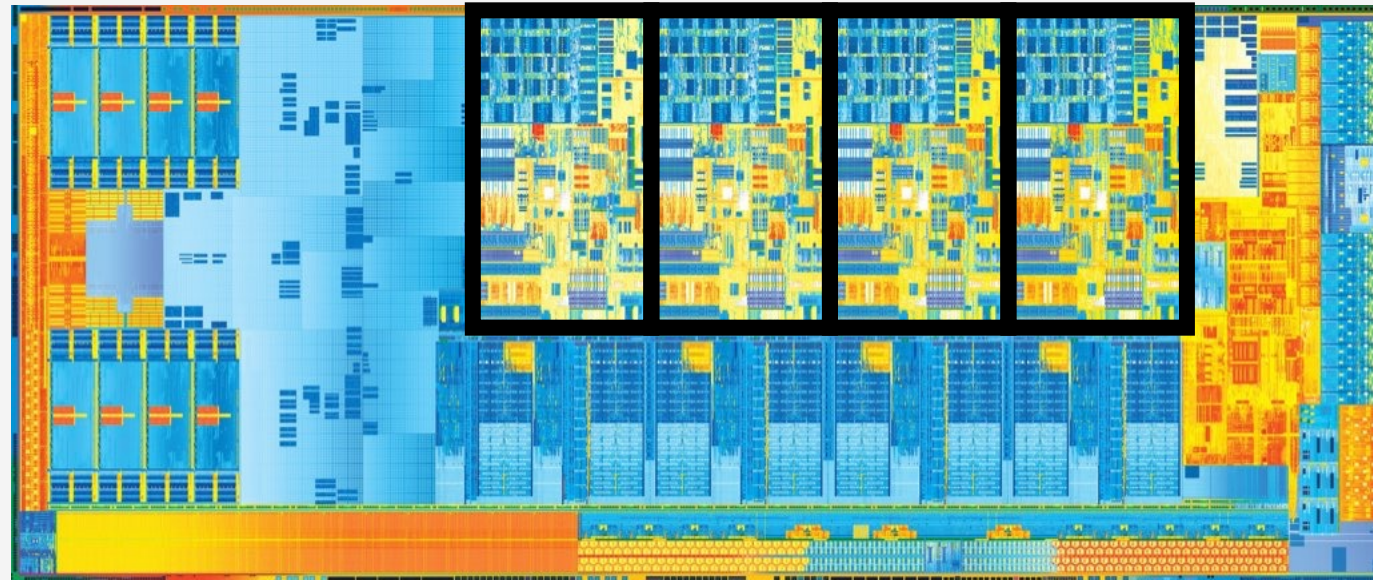


```
total = 0
for i in range(0,10):
    total += i
print total
```

```
product = 0
for i in range(0,10):
    product *= i
print product
```

Each sequence is called a *thread*.

Where do all those transistors go?



Each sequence is called a *thread*.

One thread can only run on one core at a time.

We have to write parallel programs to
fully utilize modern processors.

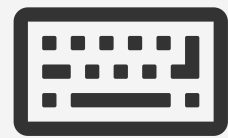
 Why we have to write parallel programs

 How to write parallel programs

 Why parallel programming is hard



Why we have to write parallel programs

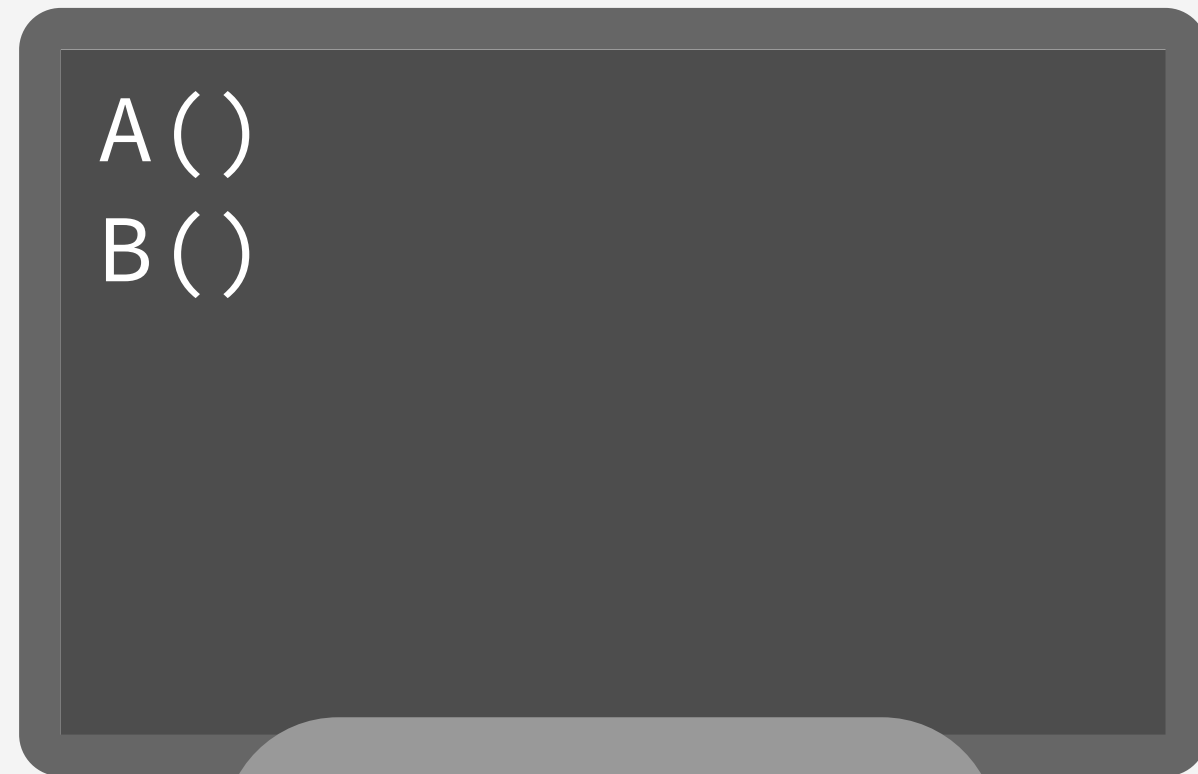
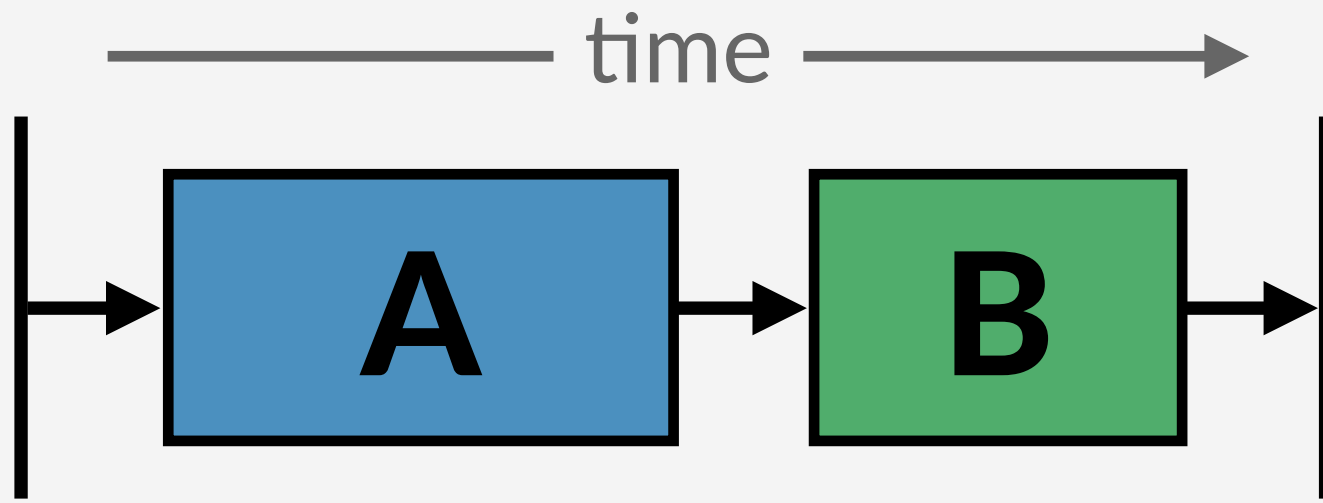


How to write parallel programs

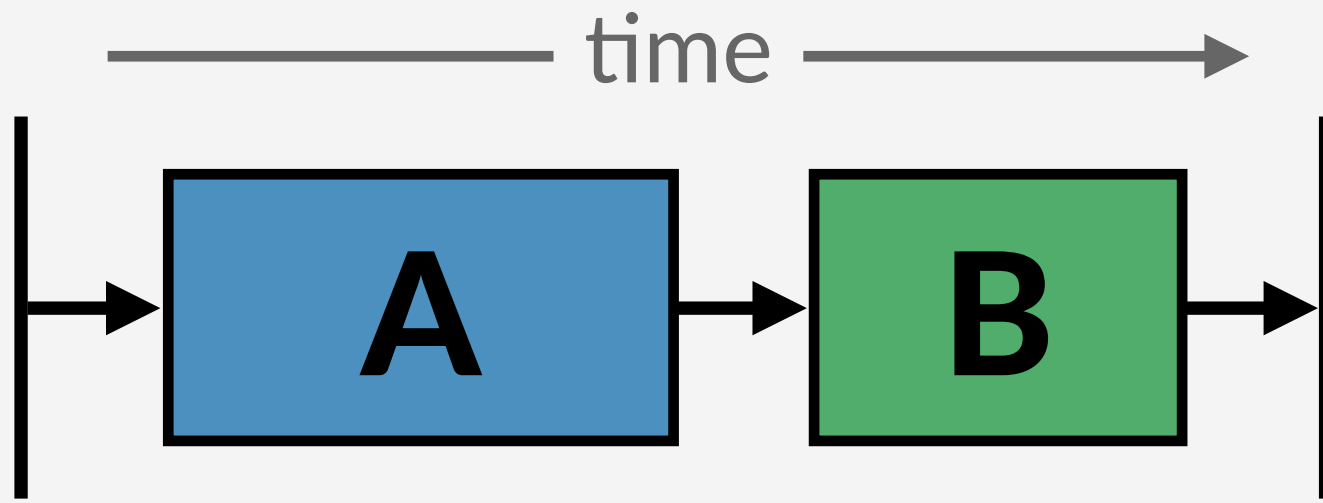


Why parallel programming is hard

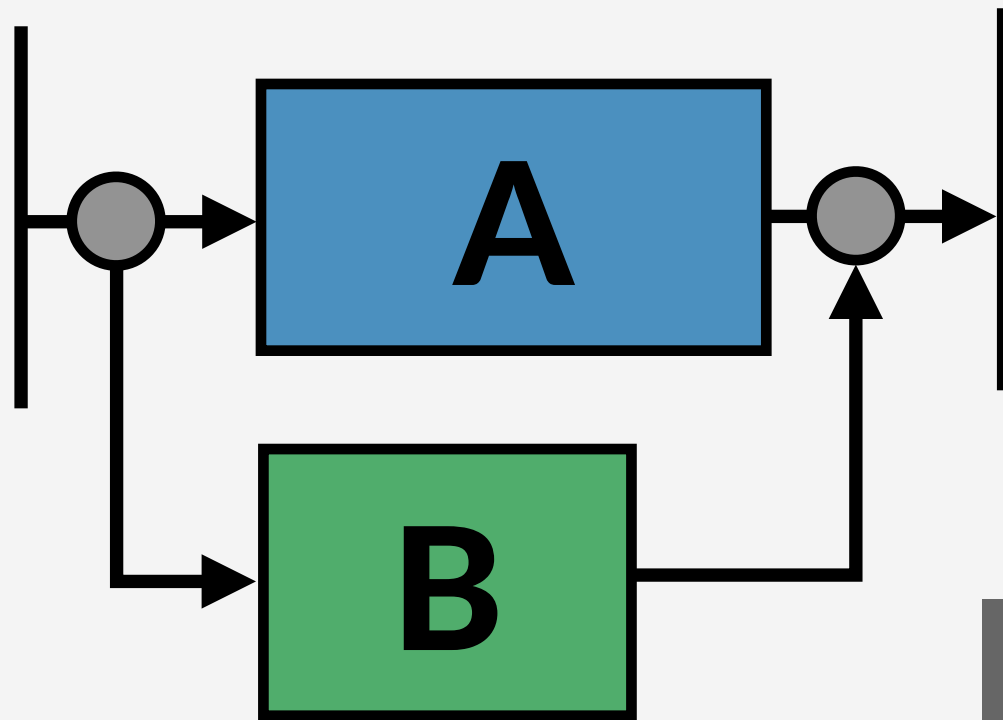
we have to write a parallel program.



we have to write a parallel program.

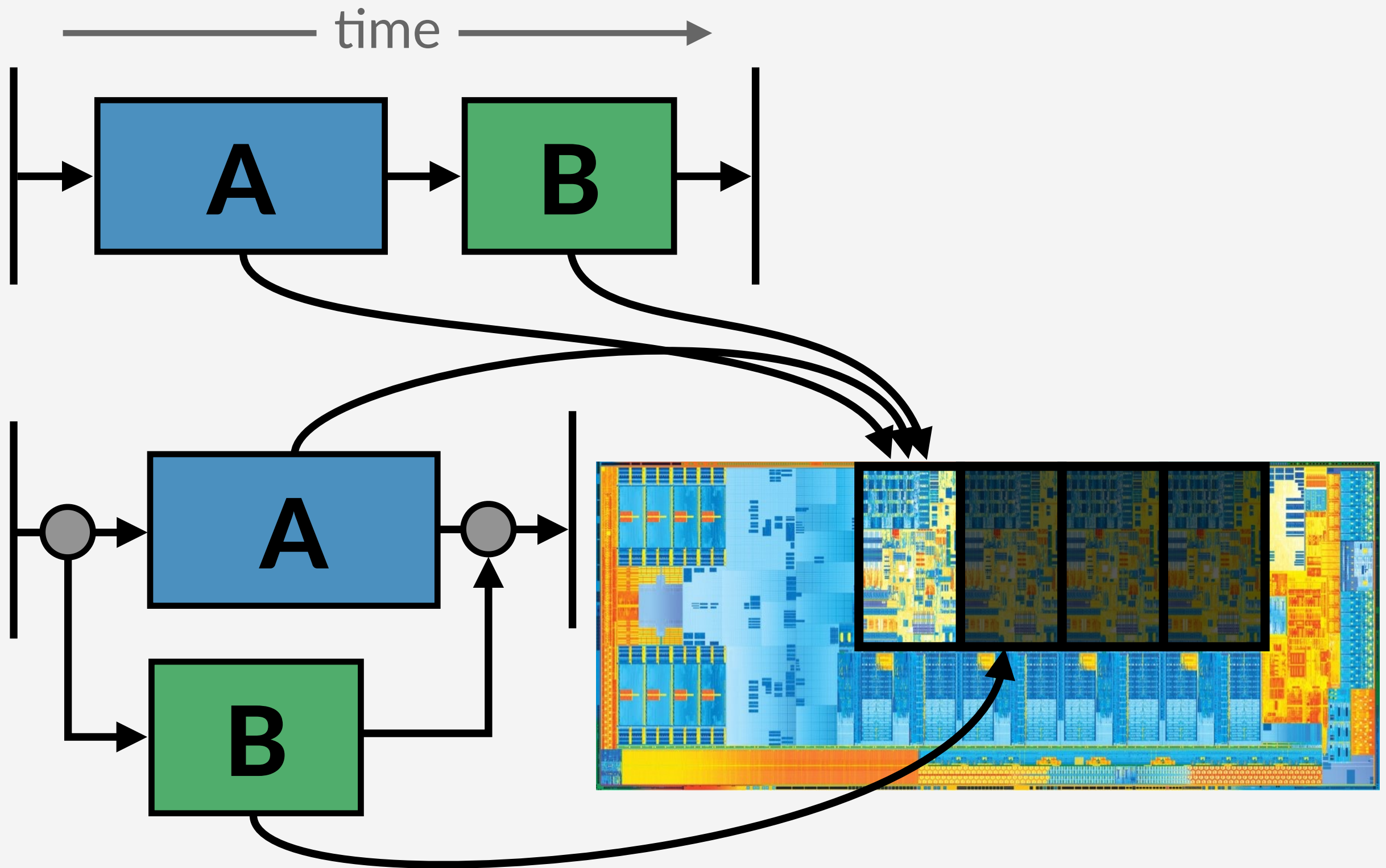


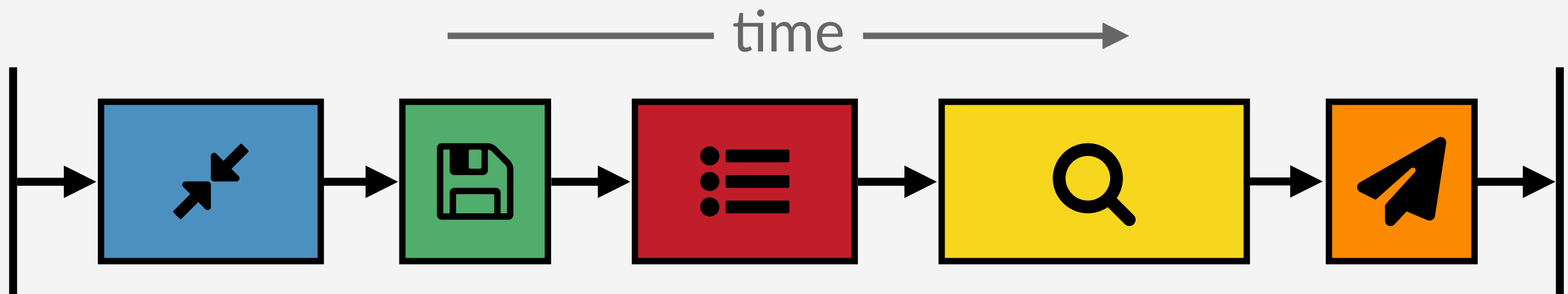
What does the parallel version look like?



```
t = thread(B)
A()
t.join()
```

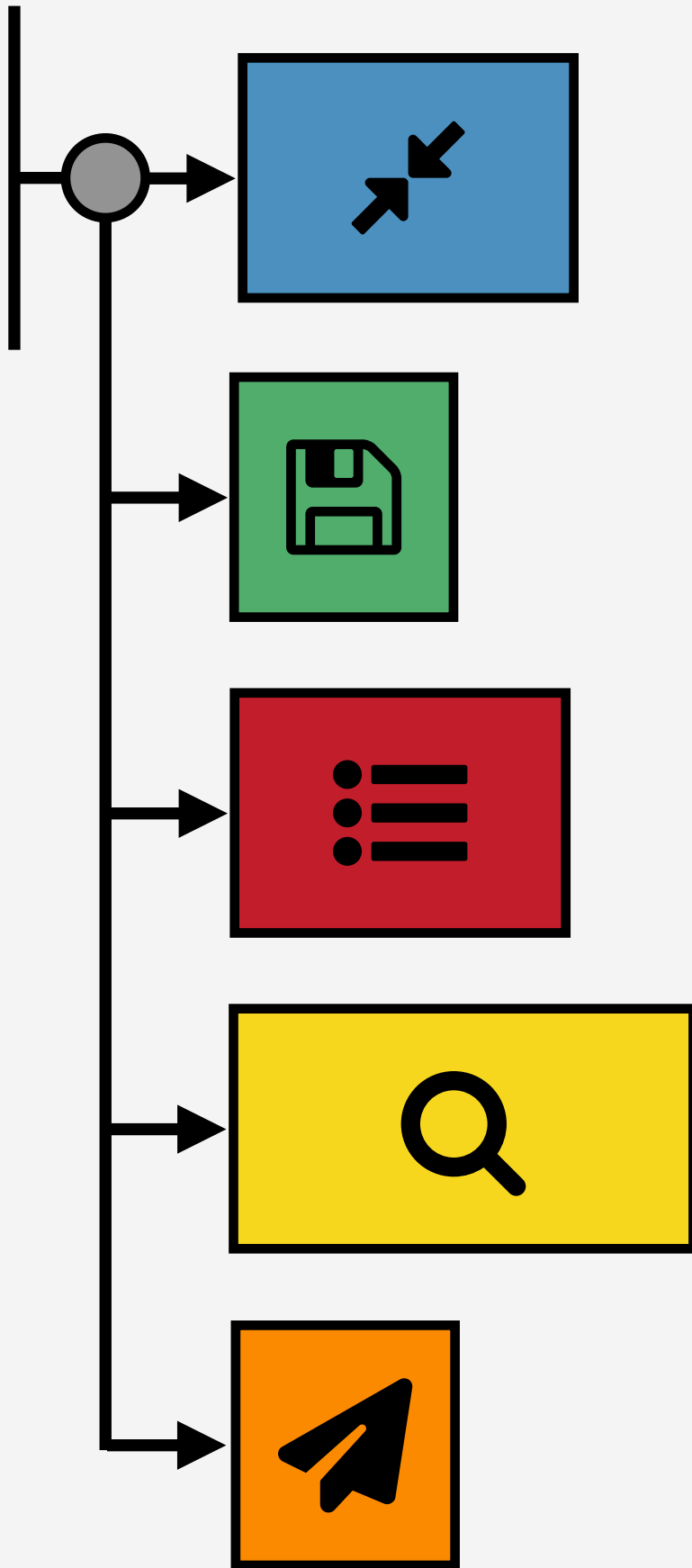
we have to write a parallel program.





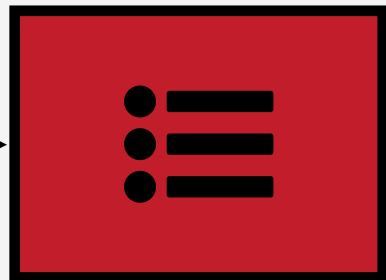
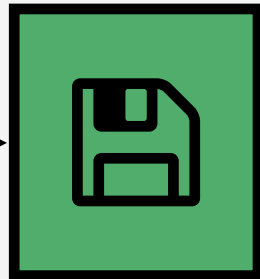
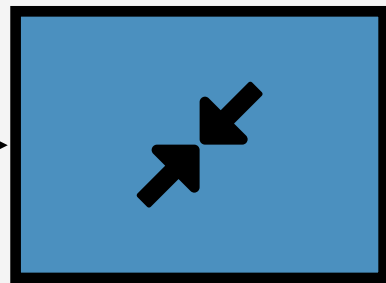
```
c = compress(img)
save(c)
f = features(img)
m = findMatches(f)
return m
```

time →

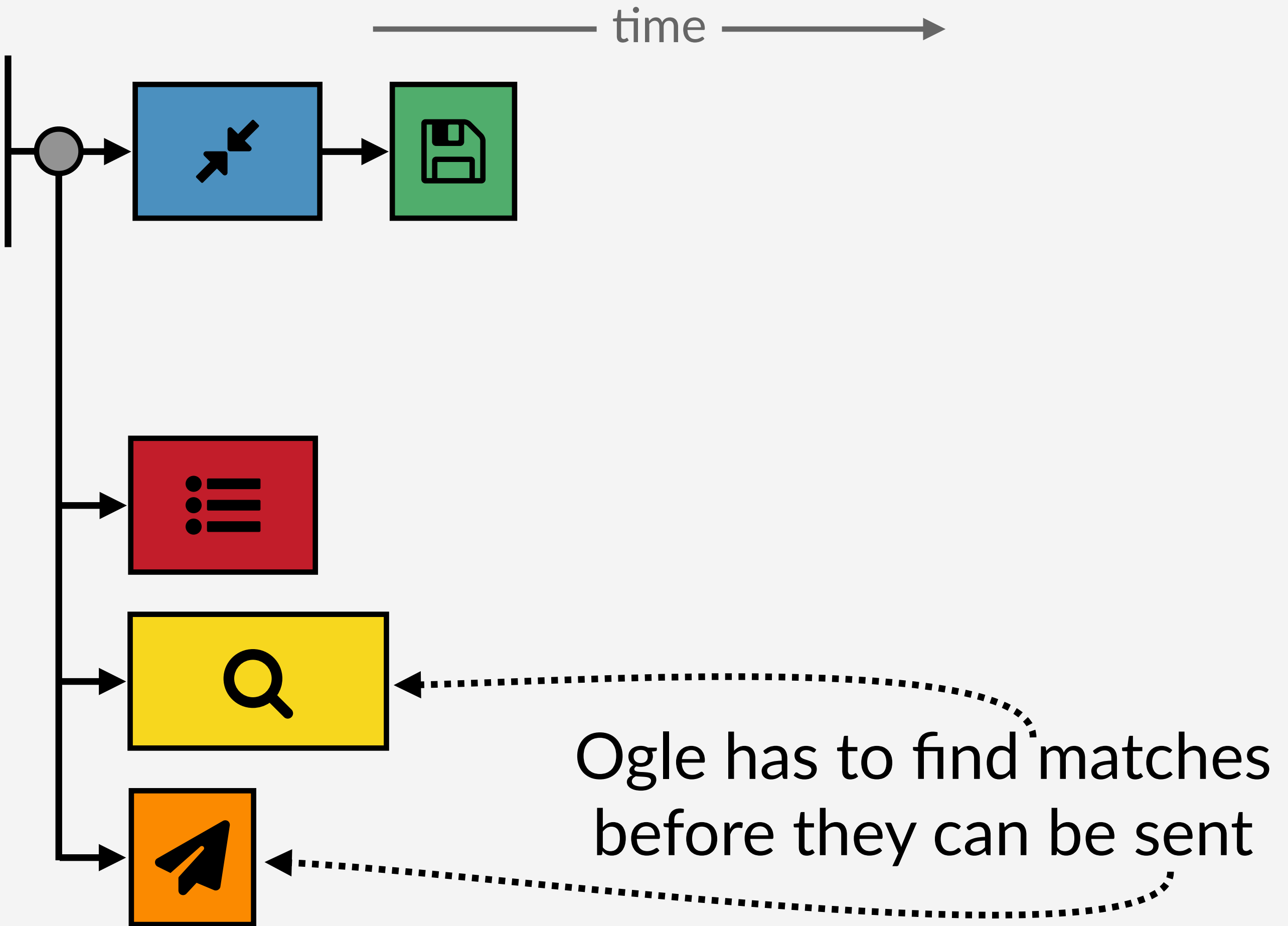


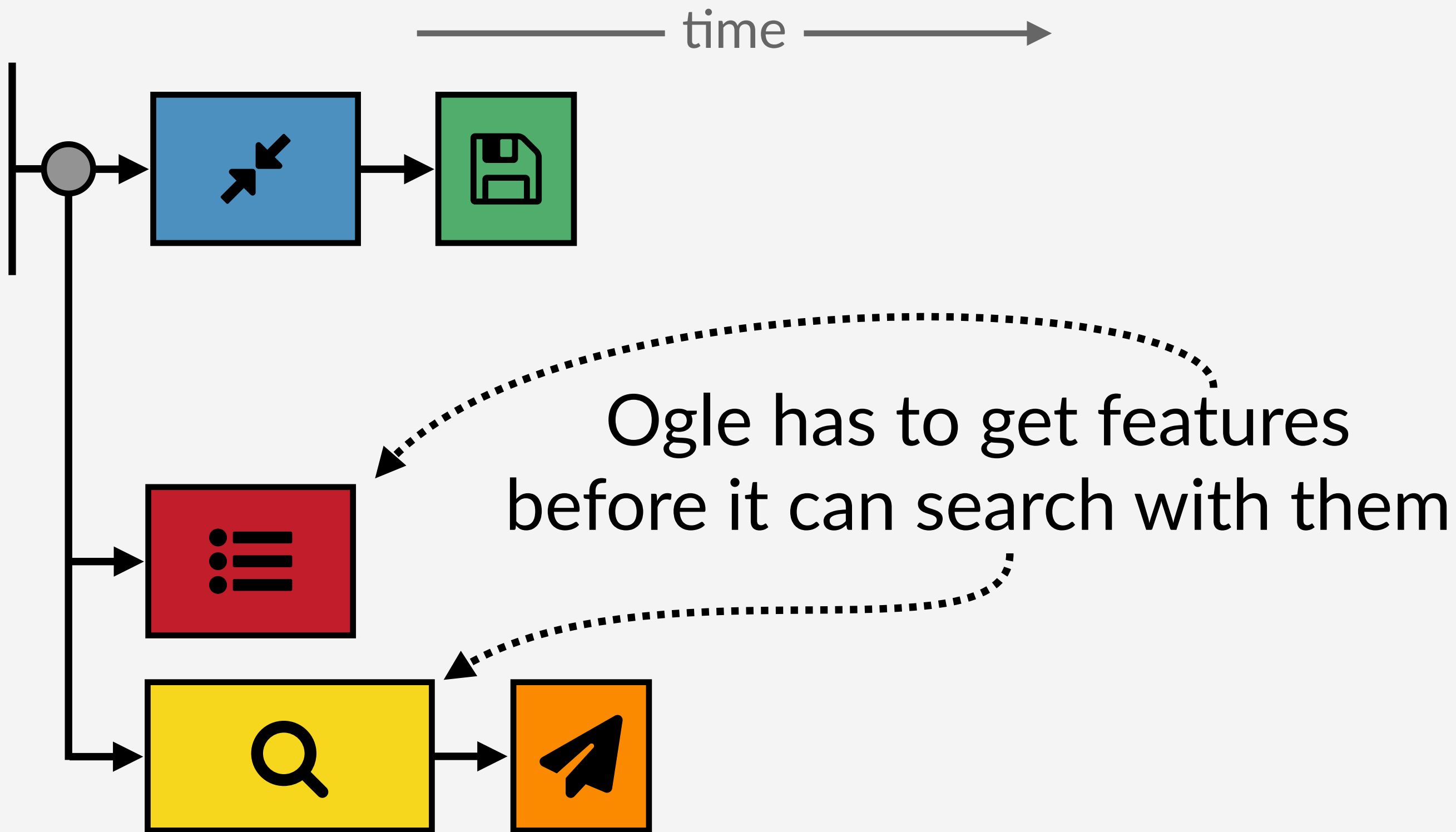
**This version will run *much* faster
but will it work?**

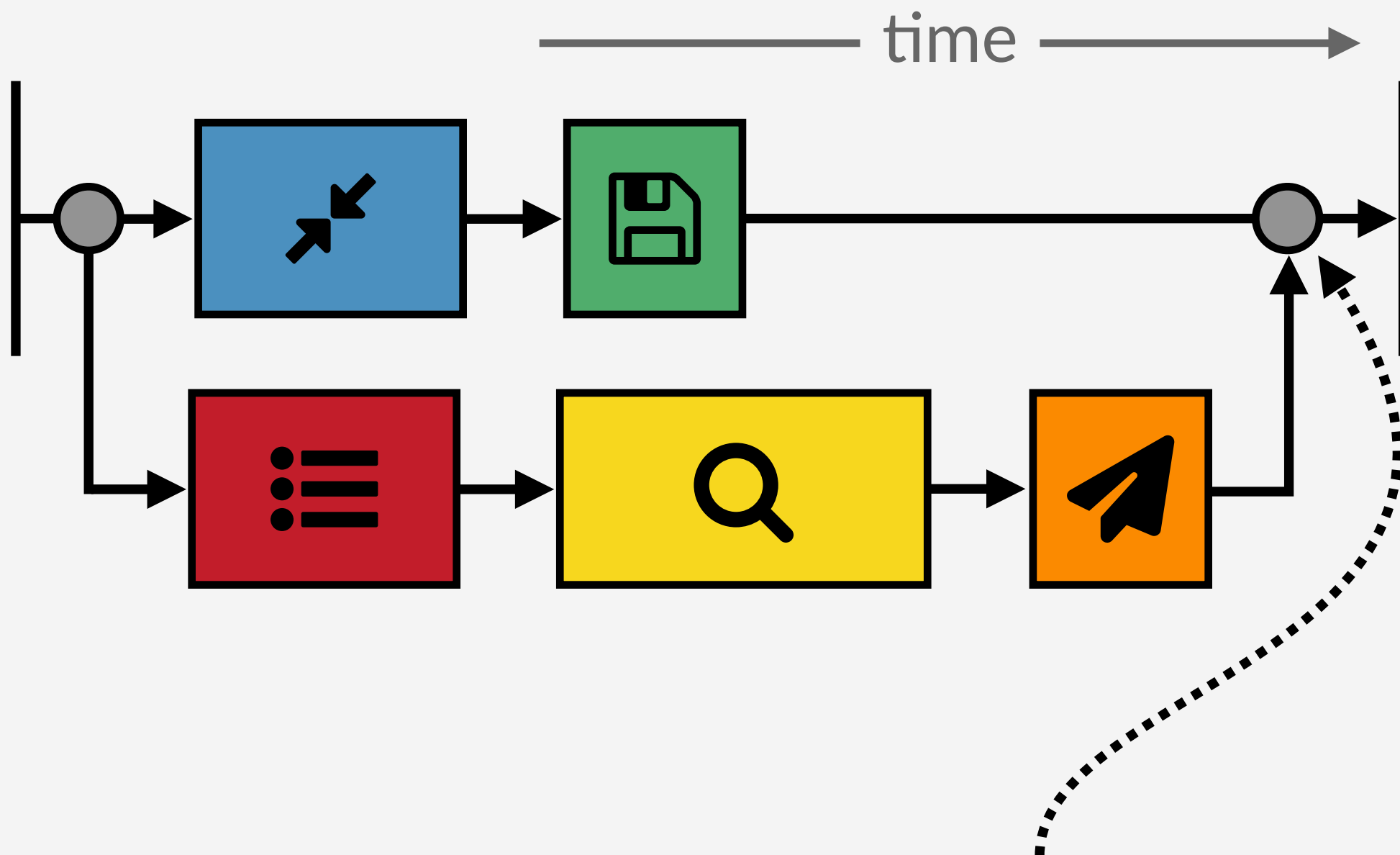
time →



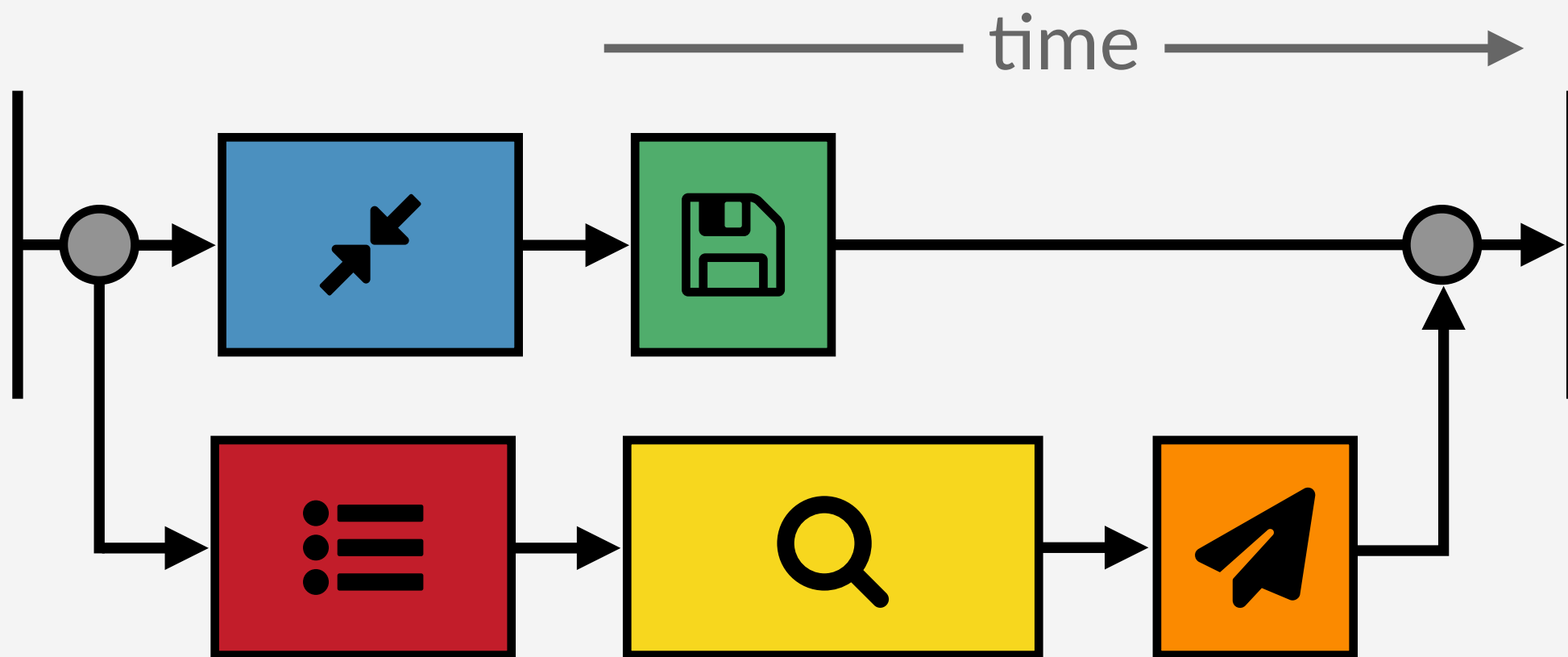
Ogle has to compress the image before it can save it







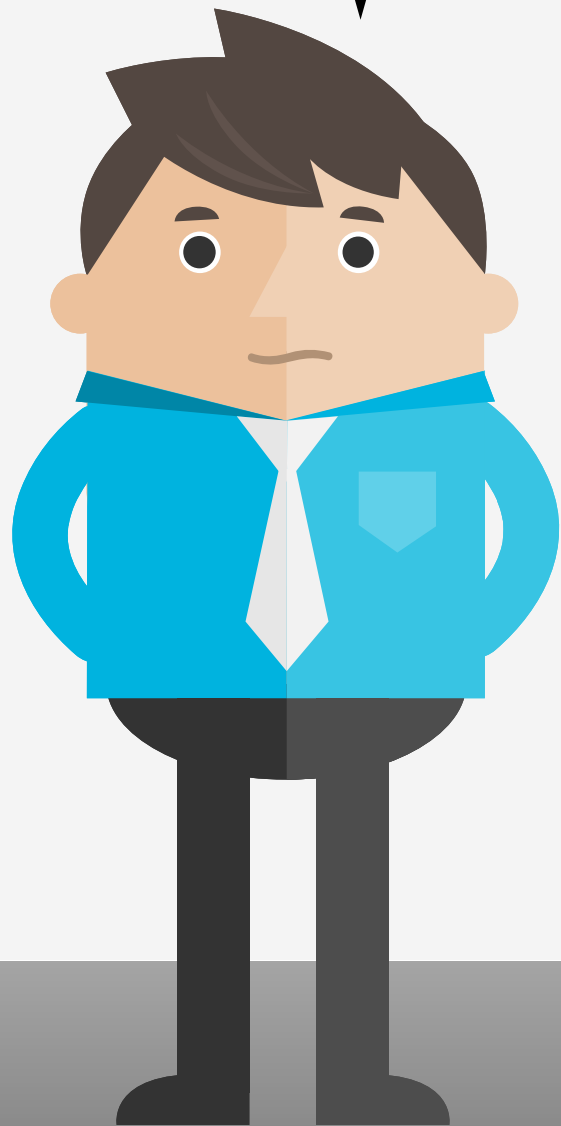
Finally, wait for both threads to finish before exiting.



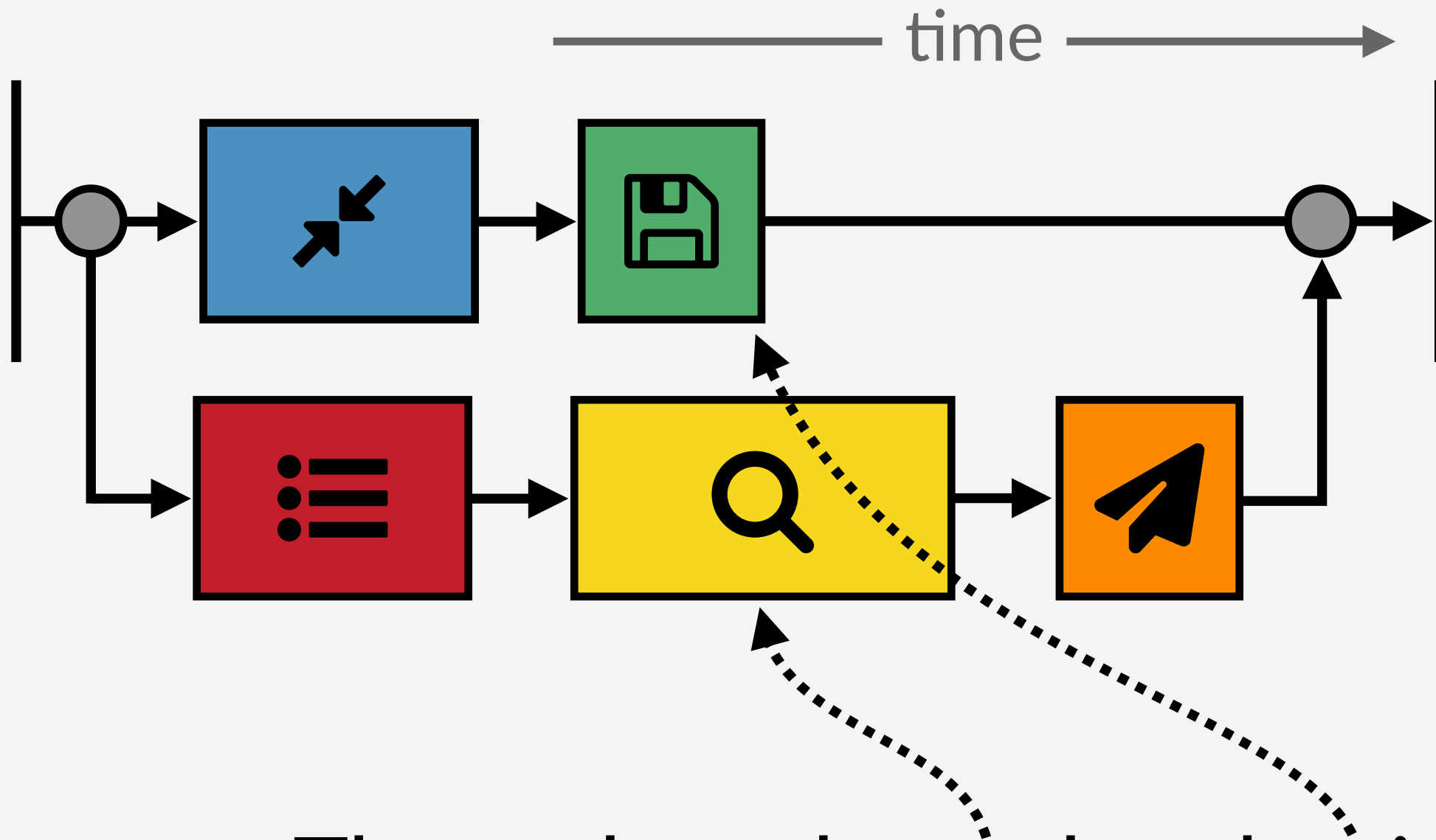
This should work, right?

This should work, right?

Ogle is *crashing*!



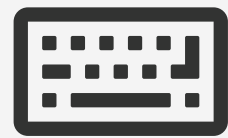
New Kinds of Errors



These threads read and write the image database at the same time.



Why we have to write parallel programs



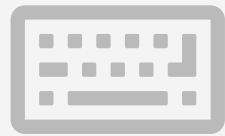
How to write parallel programs



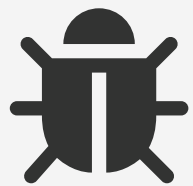
Why parallel programming is hard



Why we have to write parallel programs



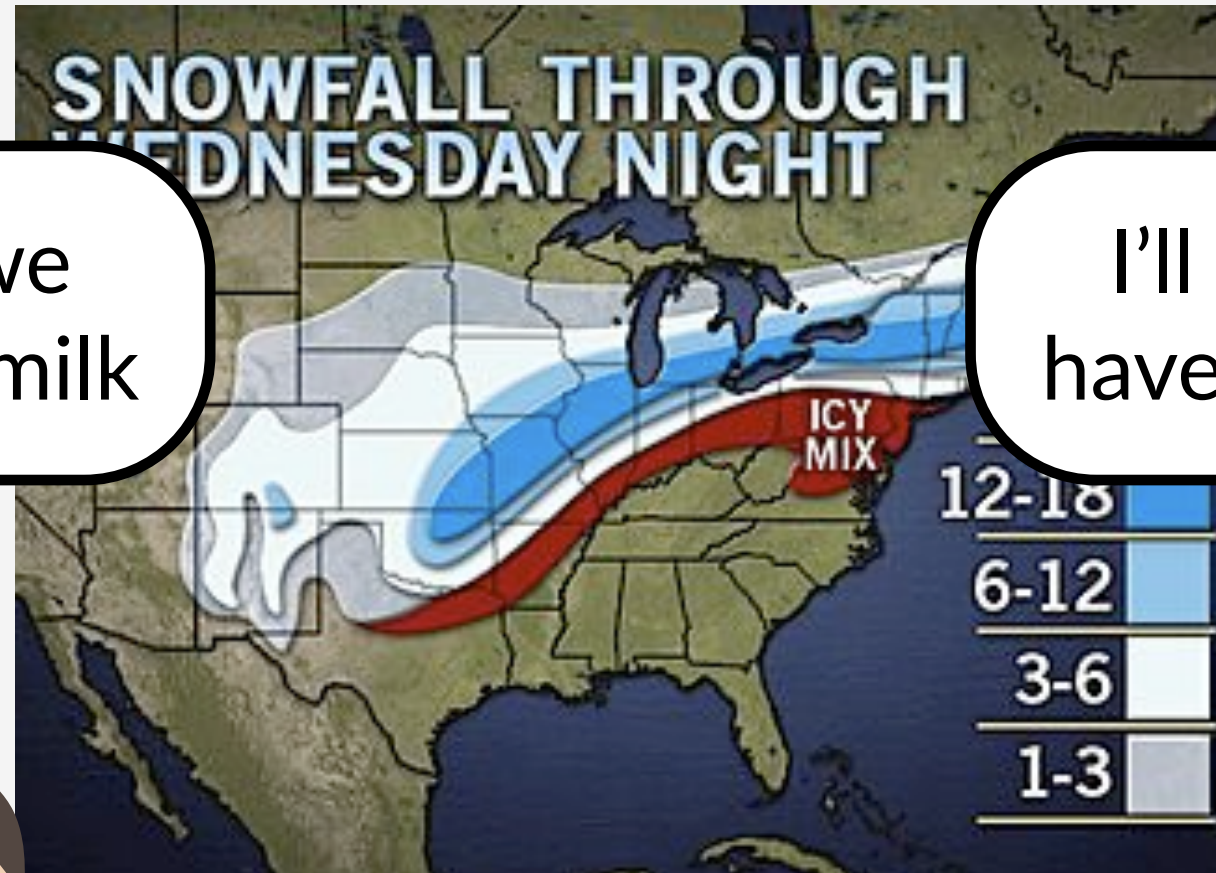
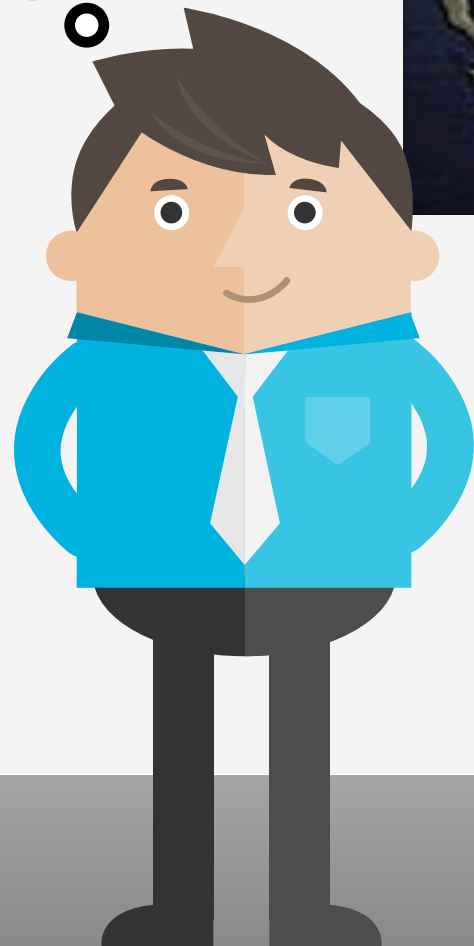
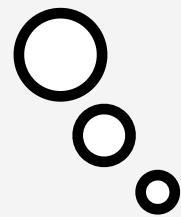
How to write parallel programs



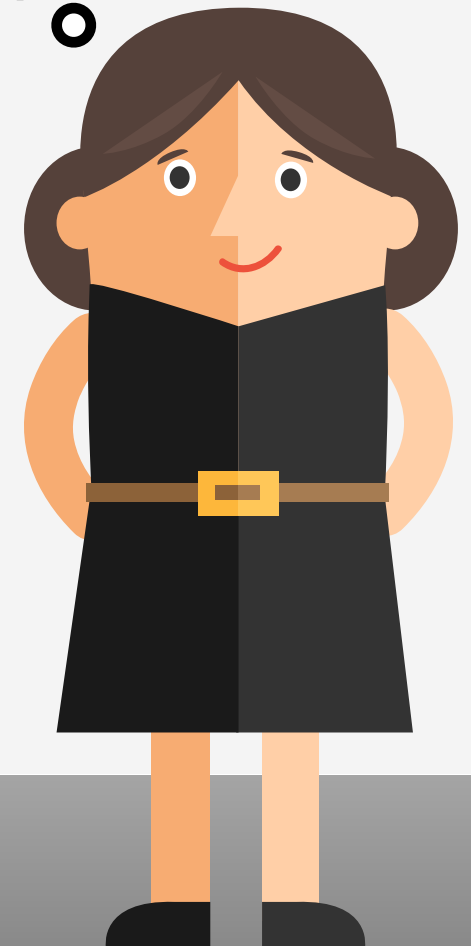
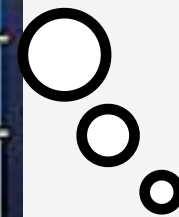
Why parallel programming is hard

Race Conditions

I'll make sure we
have bread and milk



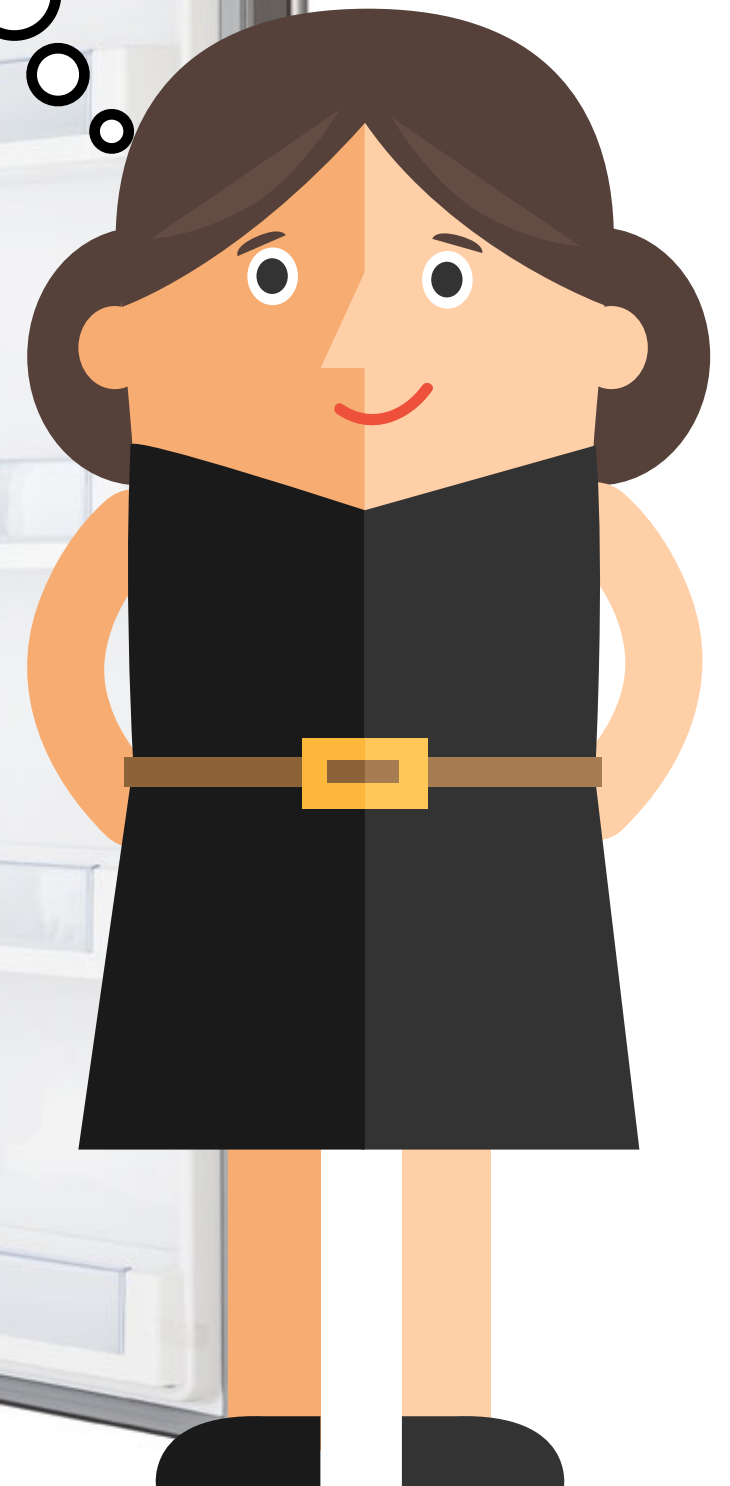
I'll make sure we
have bread and milk

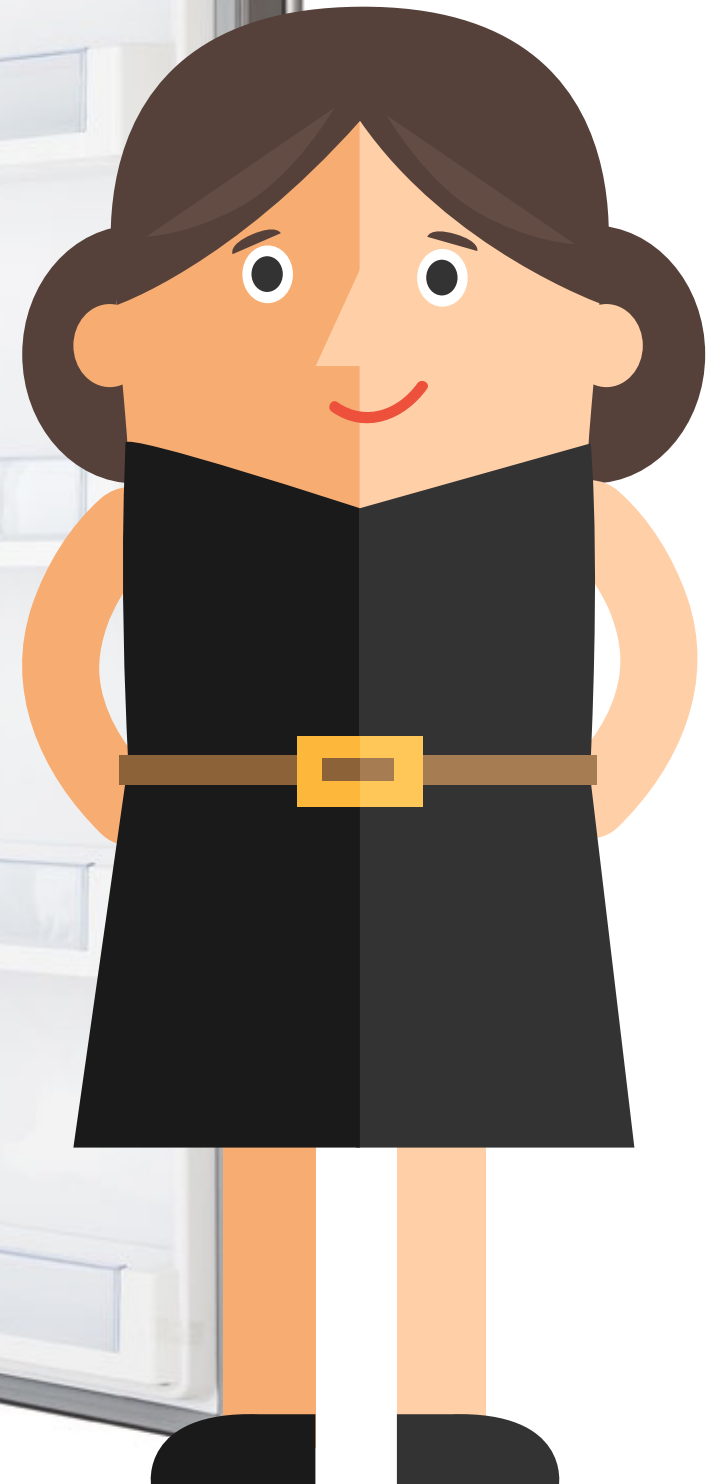


Oh no! Better go to
the store.



Oh no! Better go to
the store.



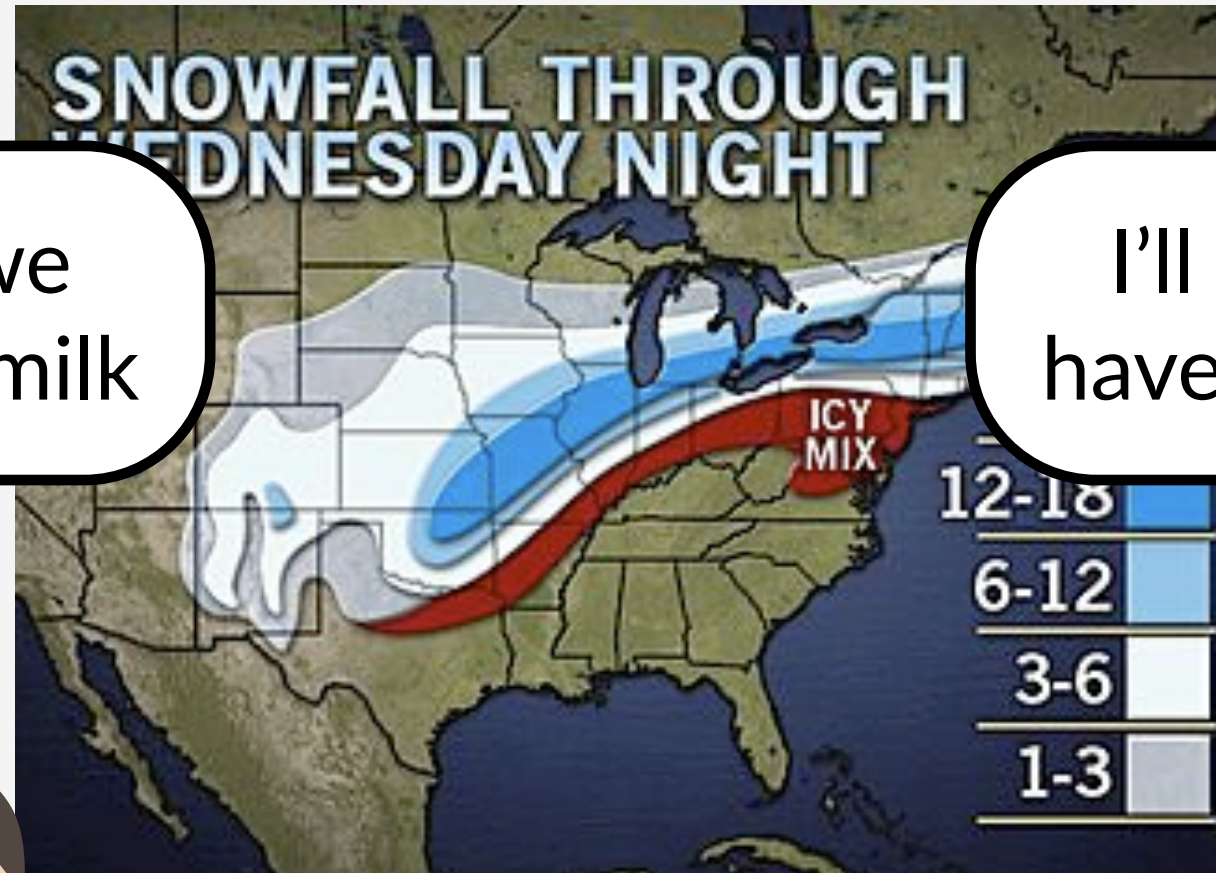
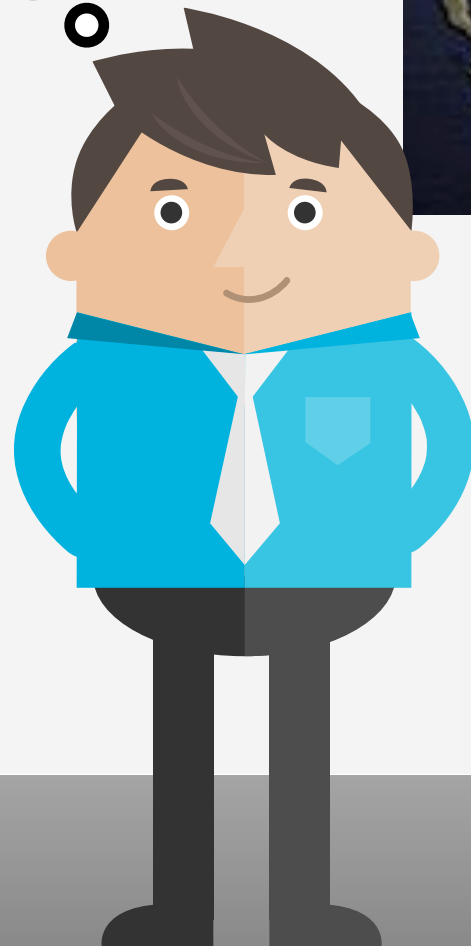
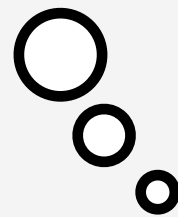


I hate race conditions!

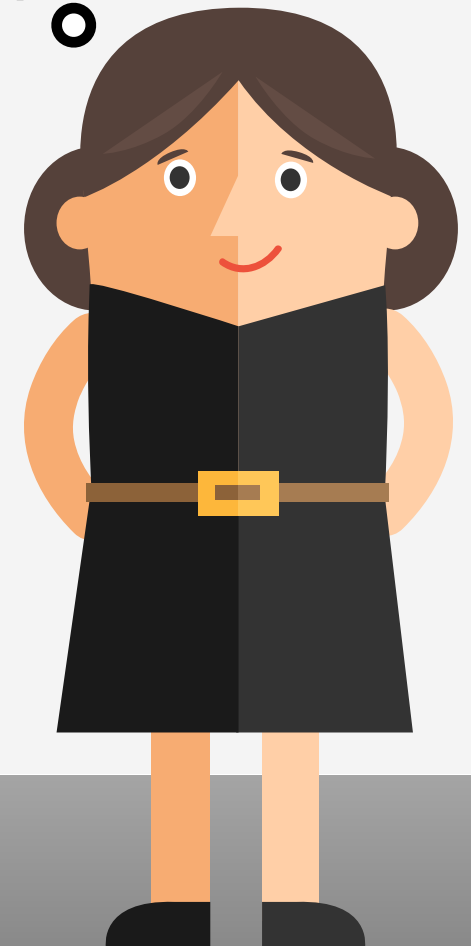
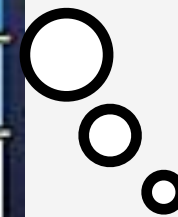


Try again with a lock

I'll make sure we
have bread and milk

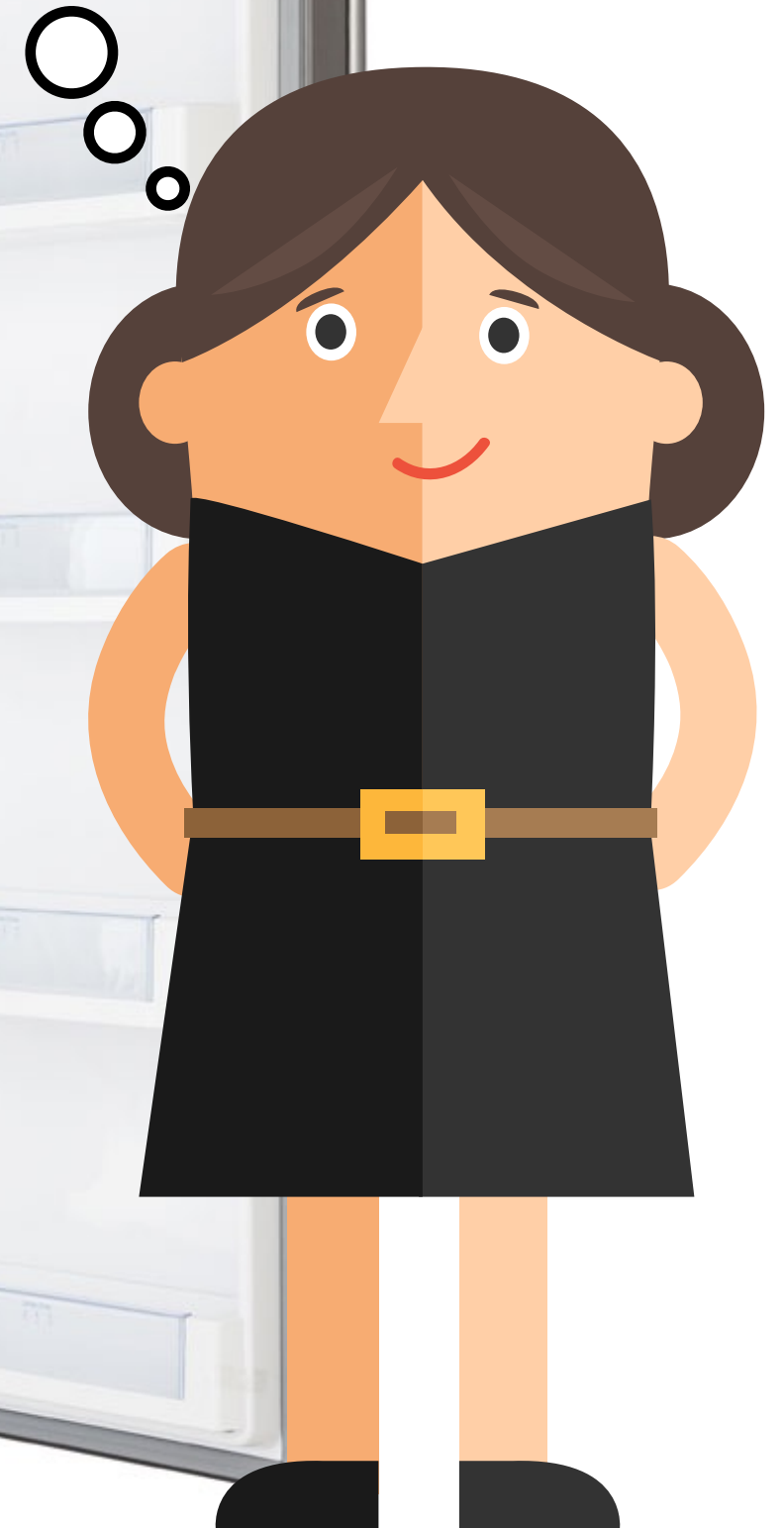


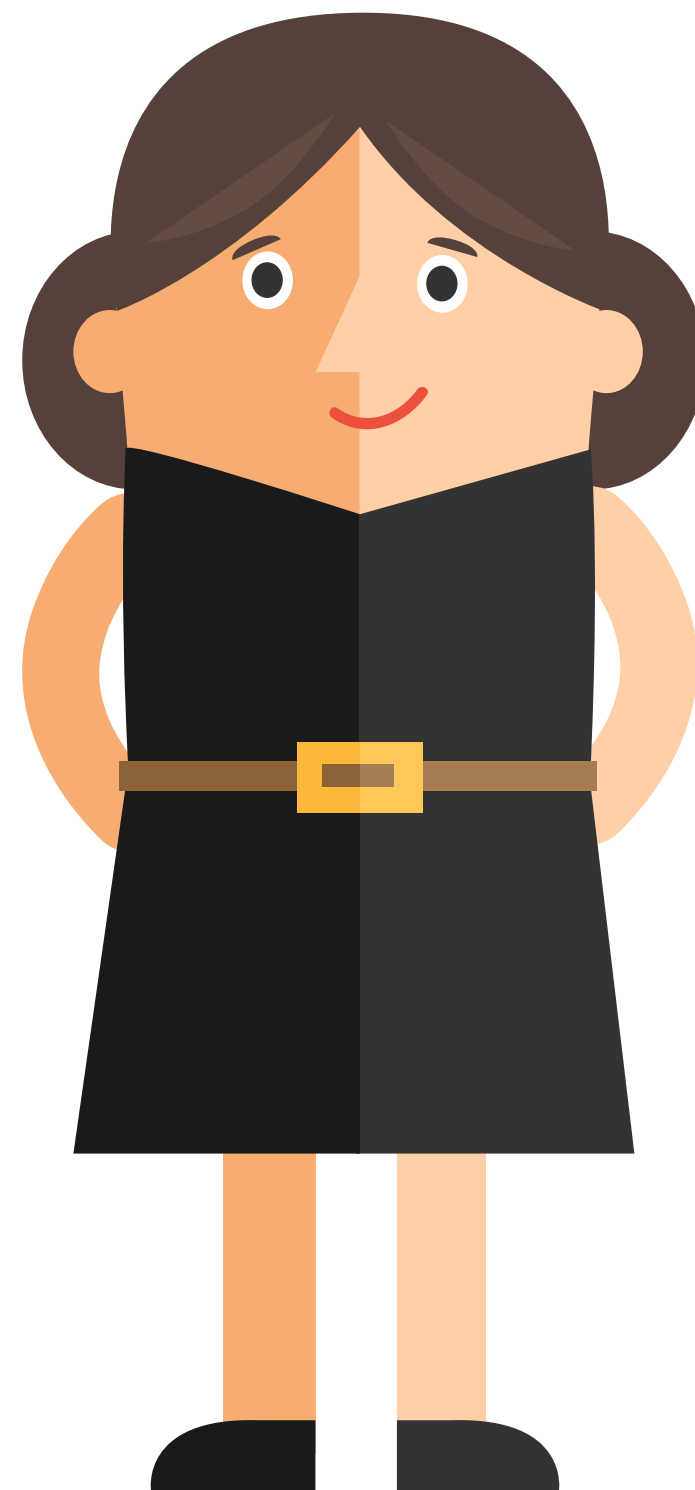
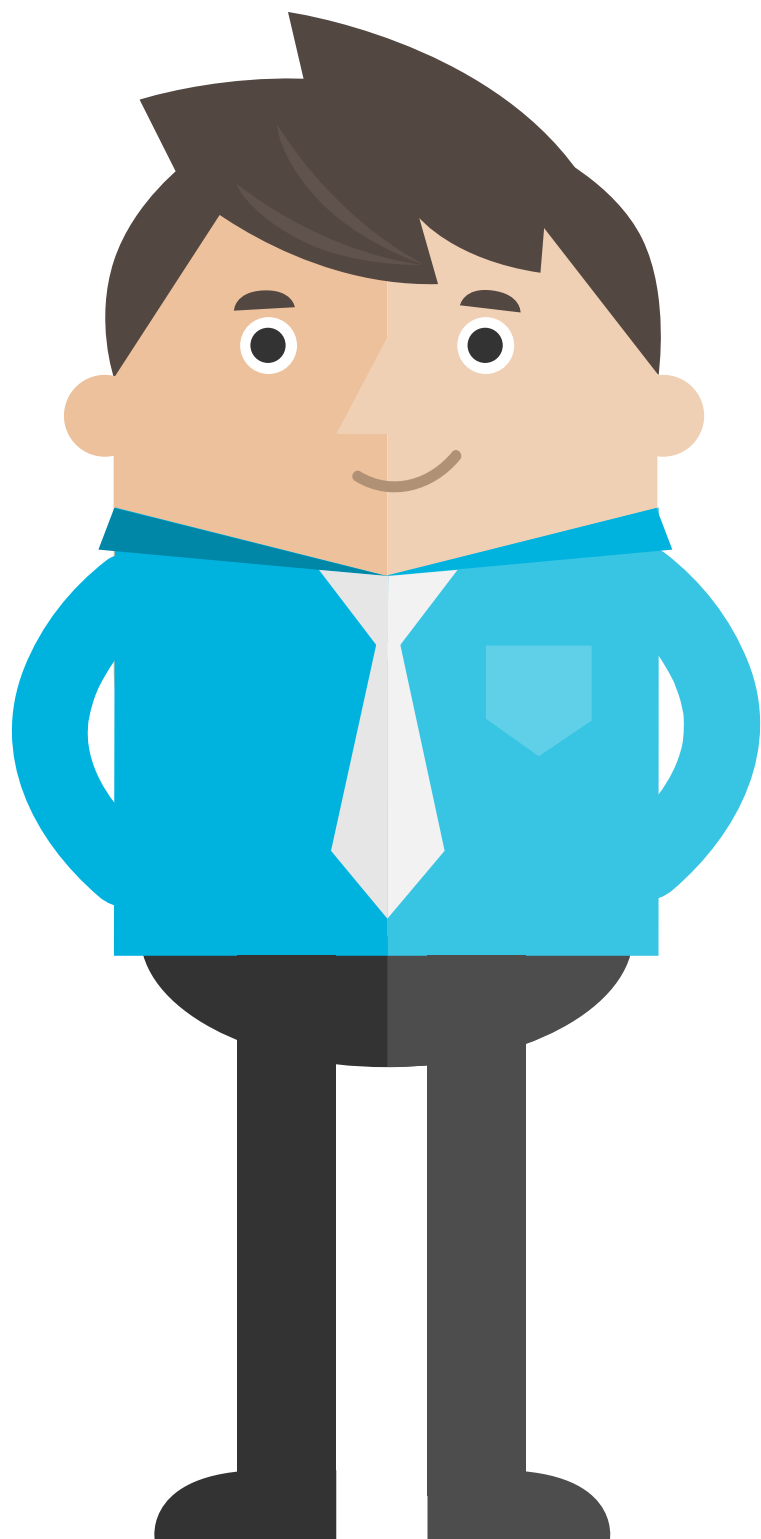
I'll make sure we
have bread and milk





Oh no! Better go to
the store.

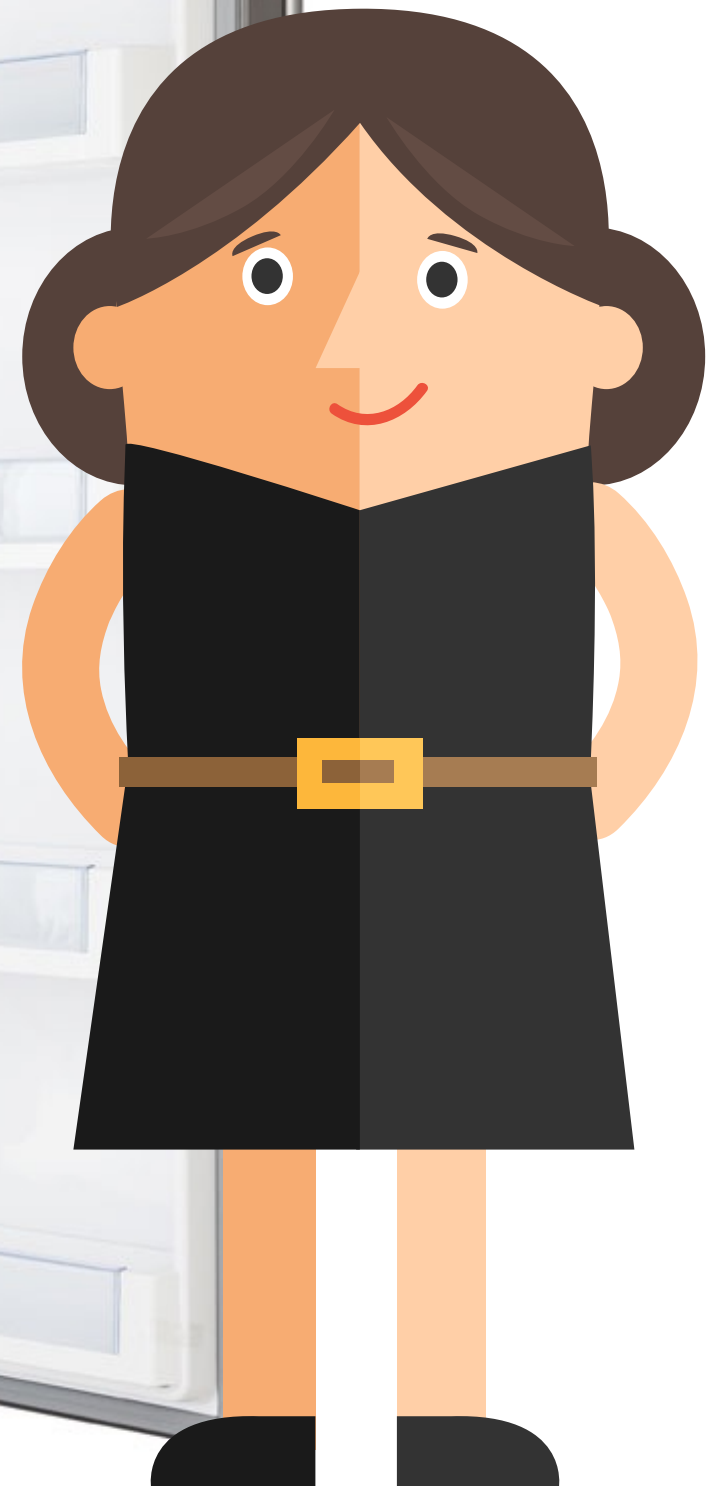






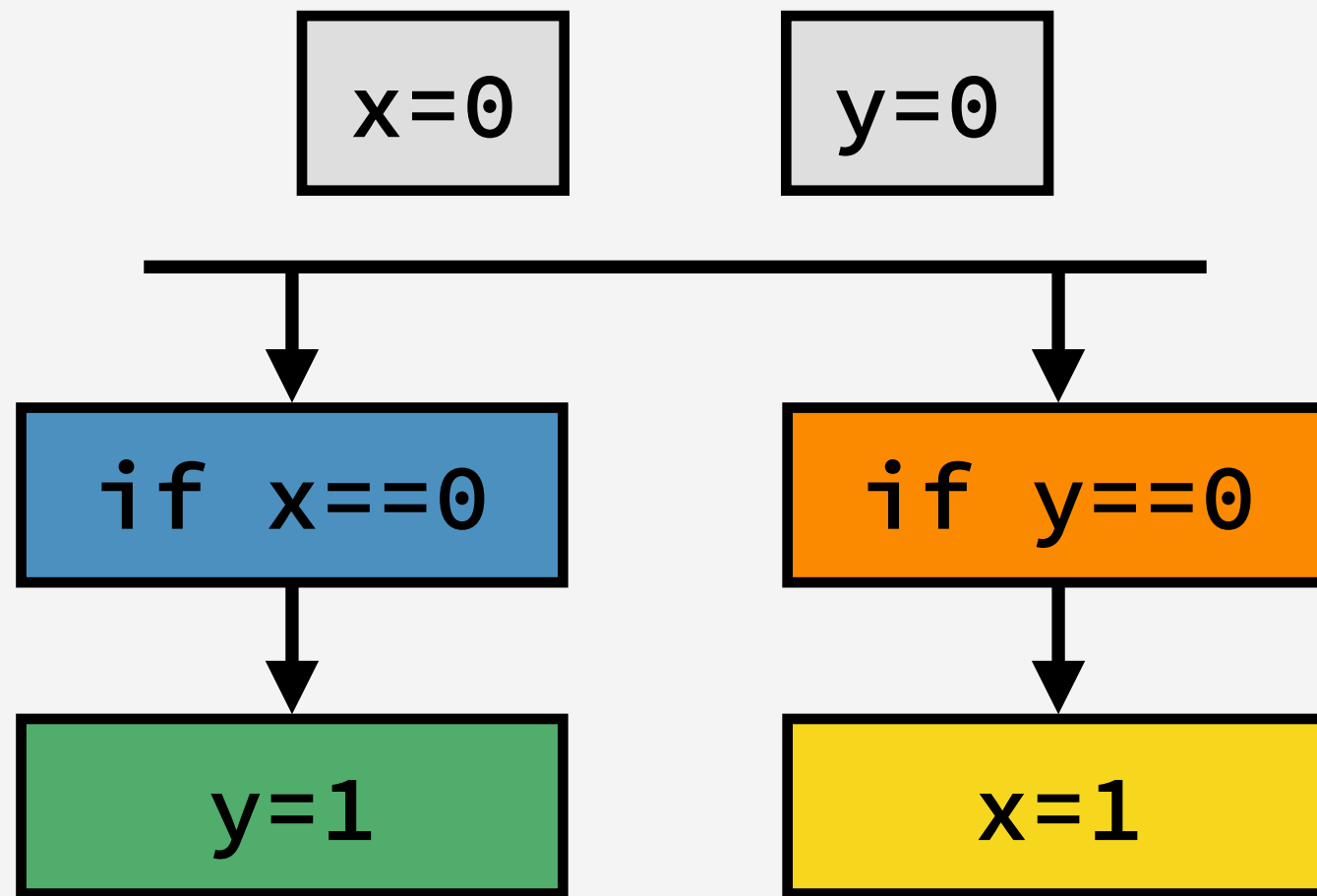


Oh good, we have
bread and milk.



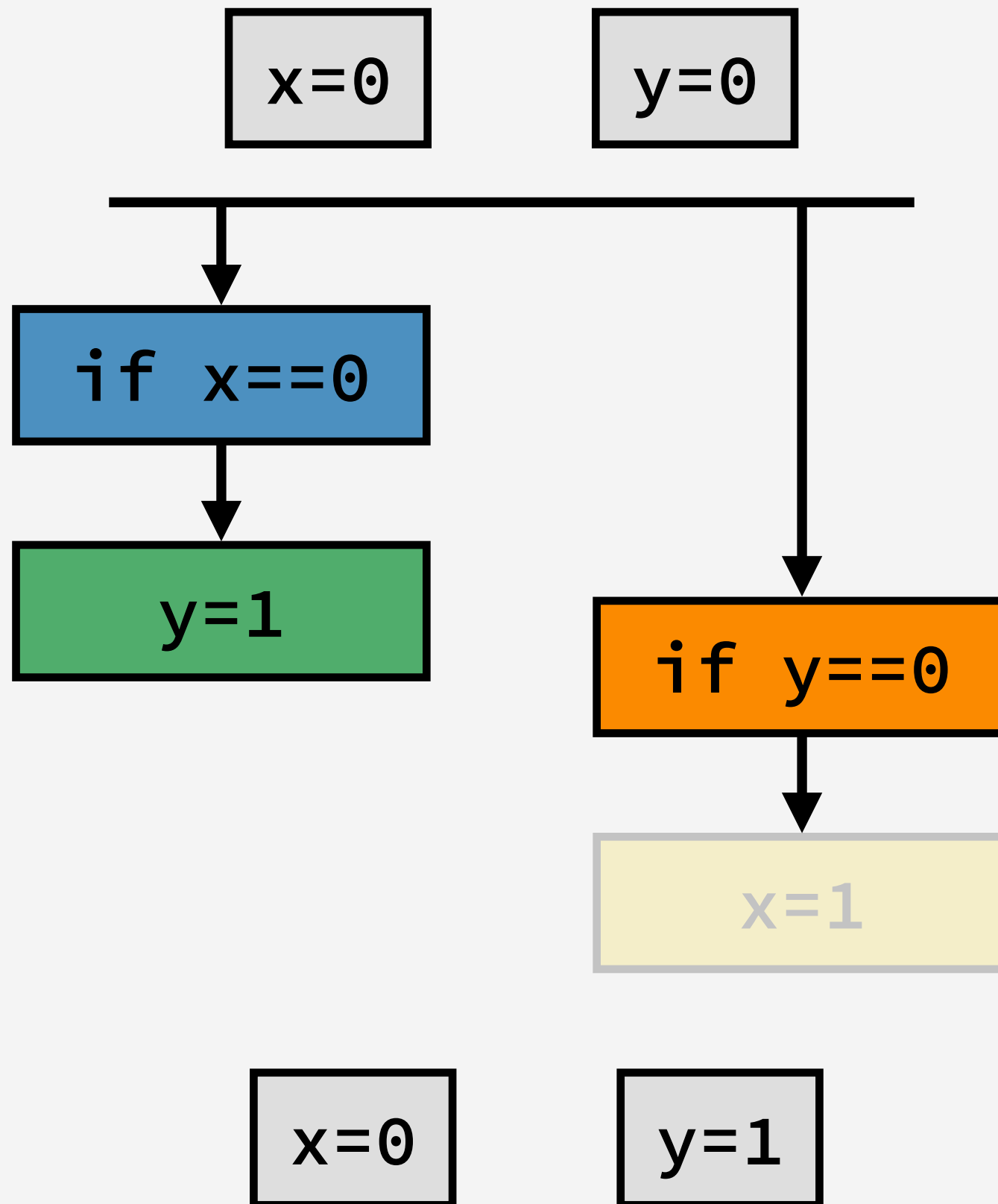
Not so different for programs

Race Conditions

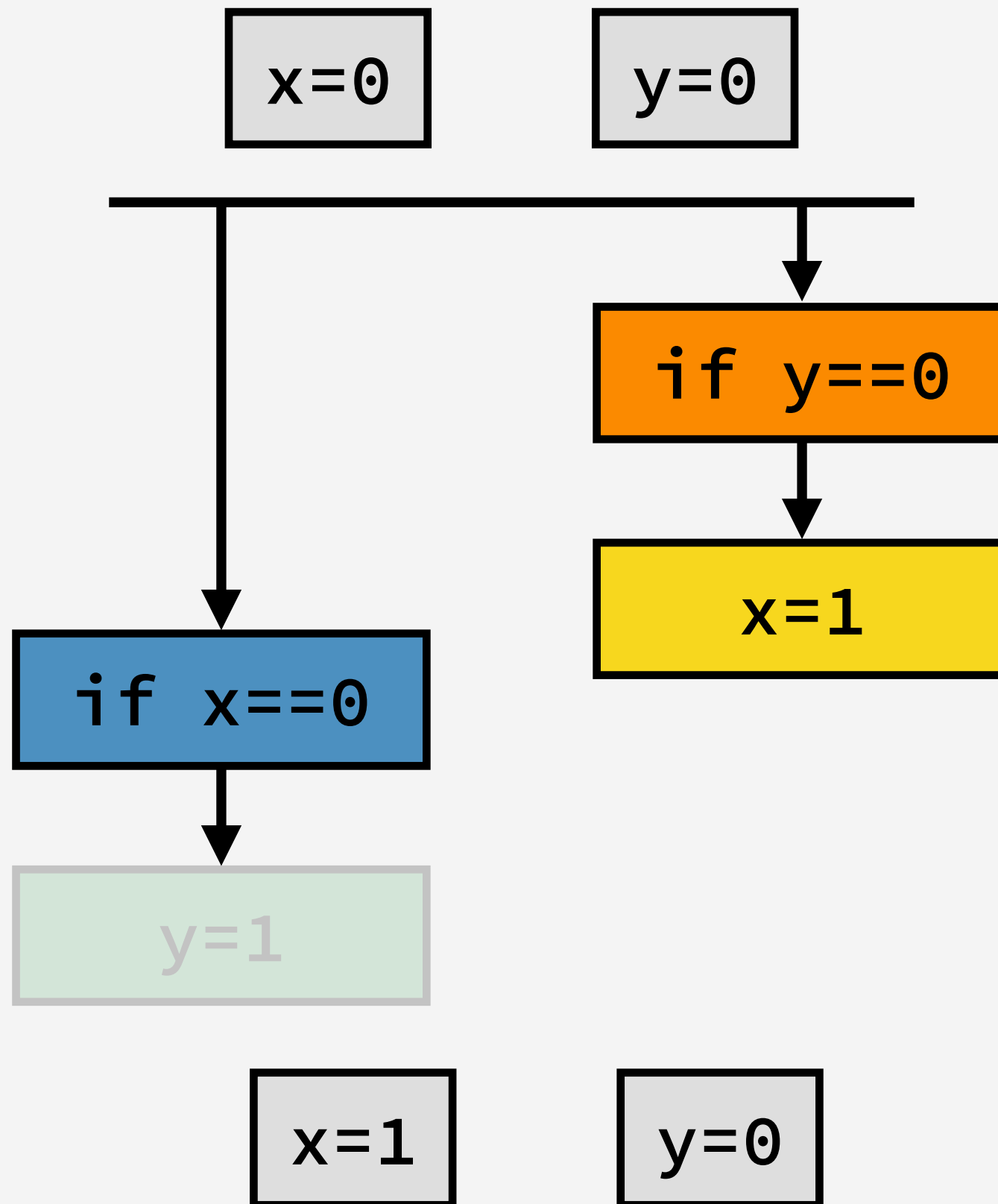


What are the values of x and y?

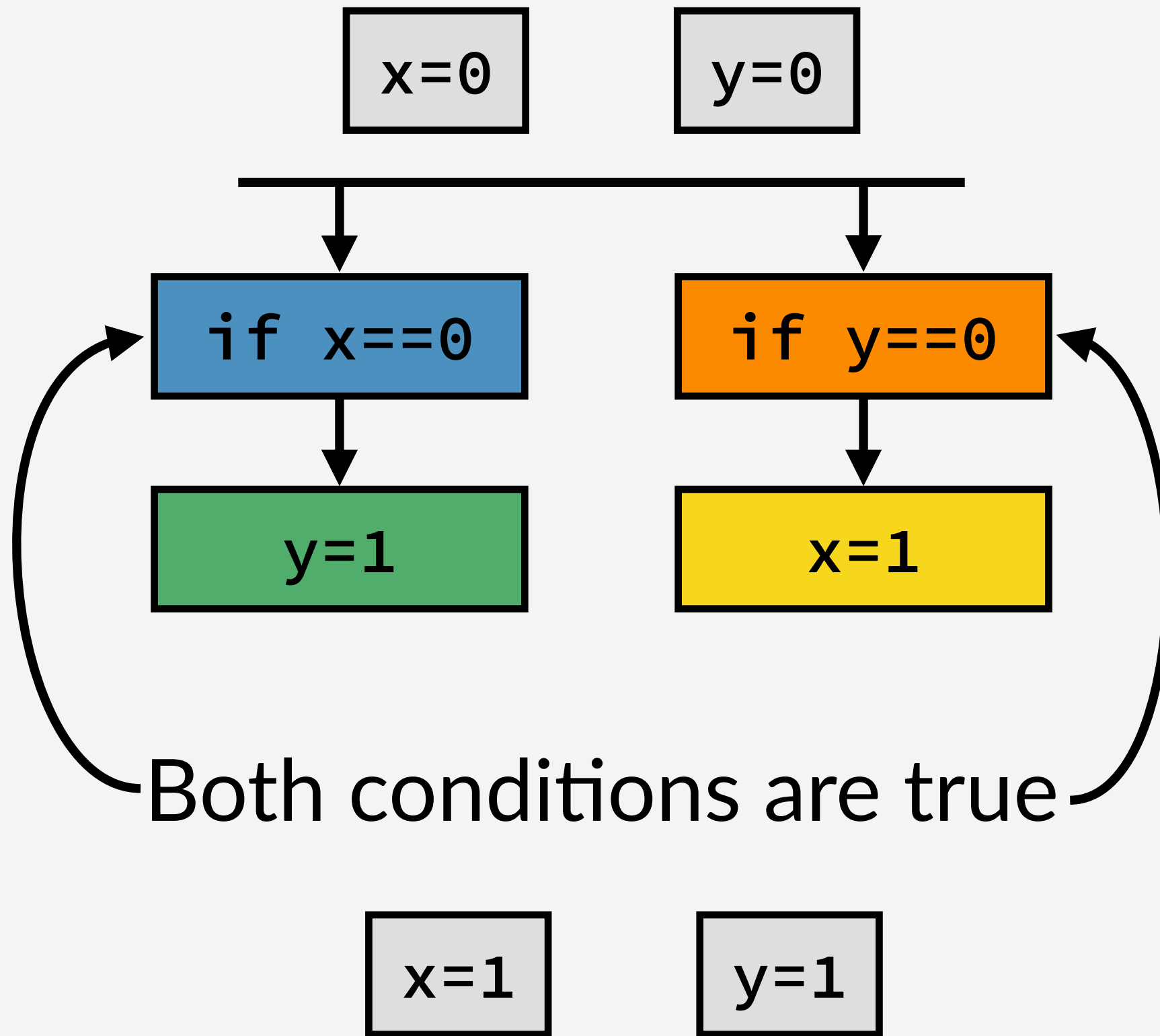
Race Conditions



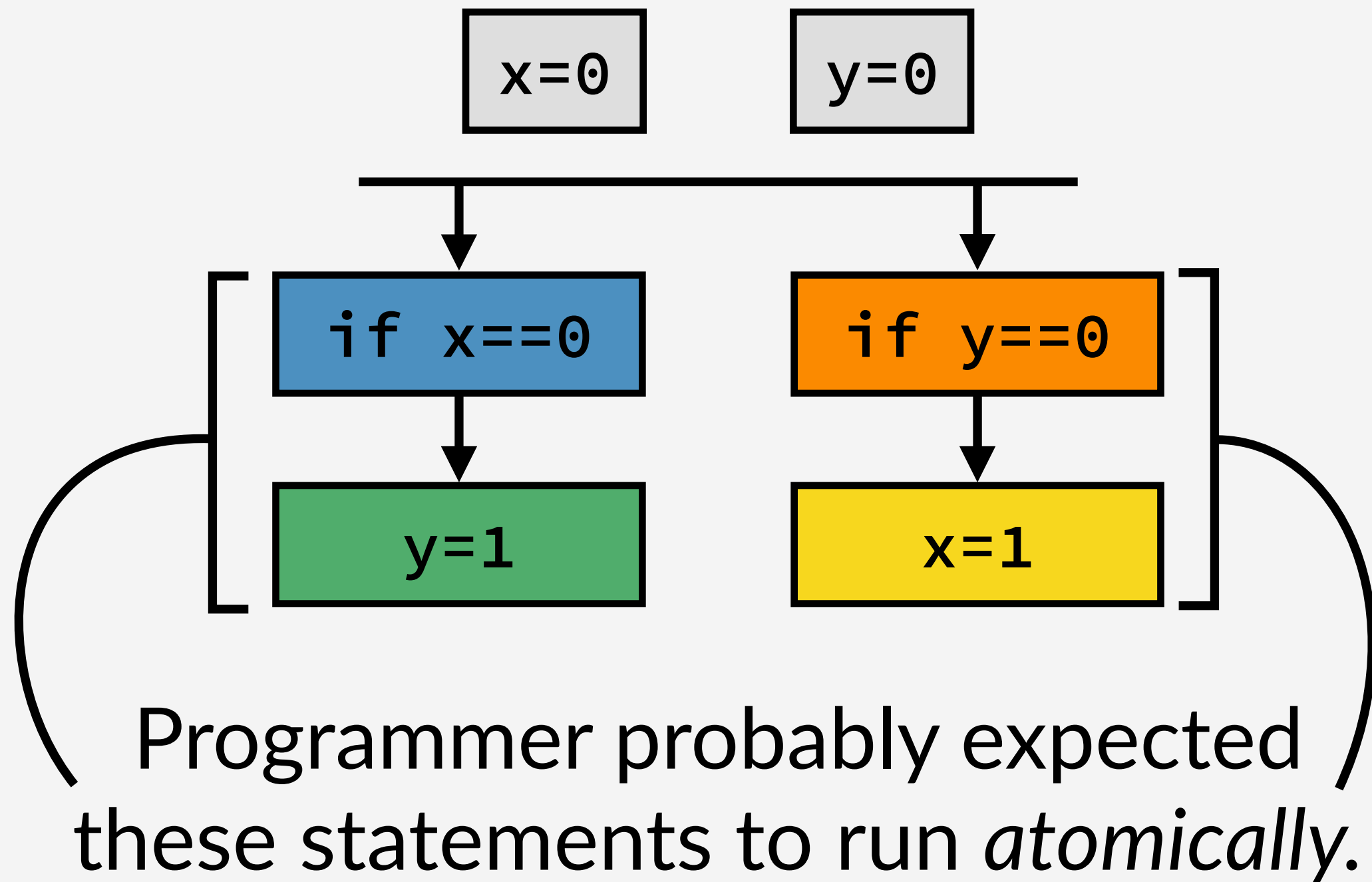
Race Conditions



Race Conditions

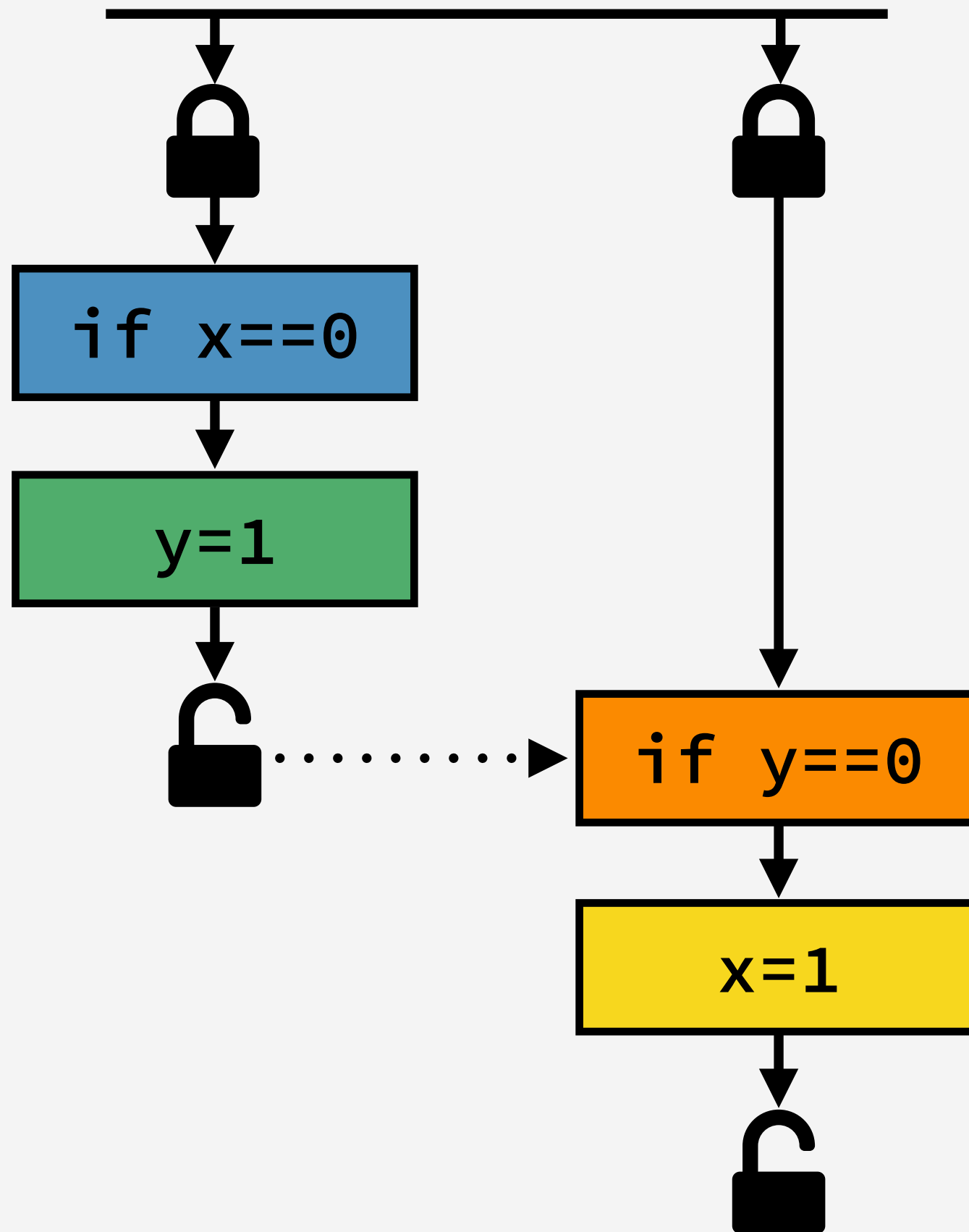


Race Conditions

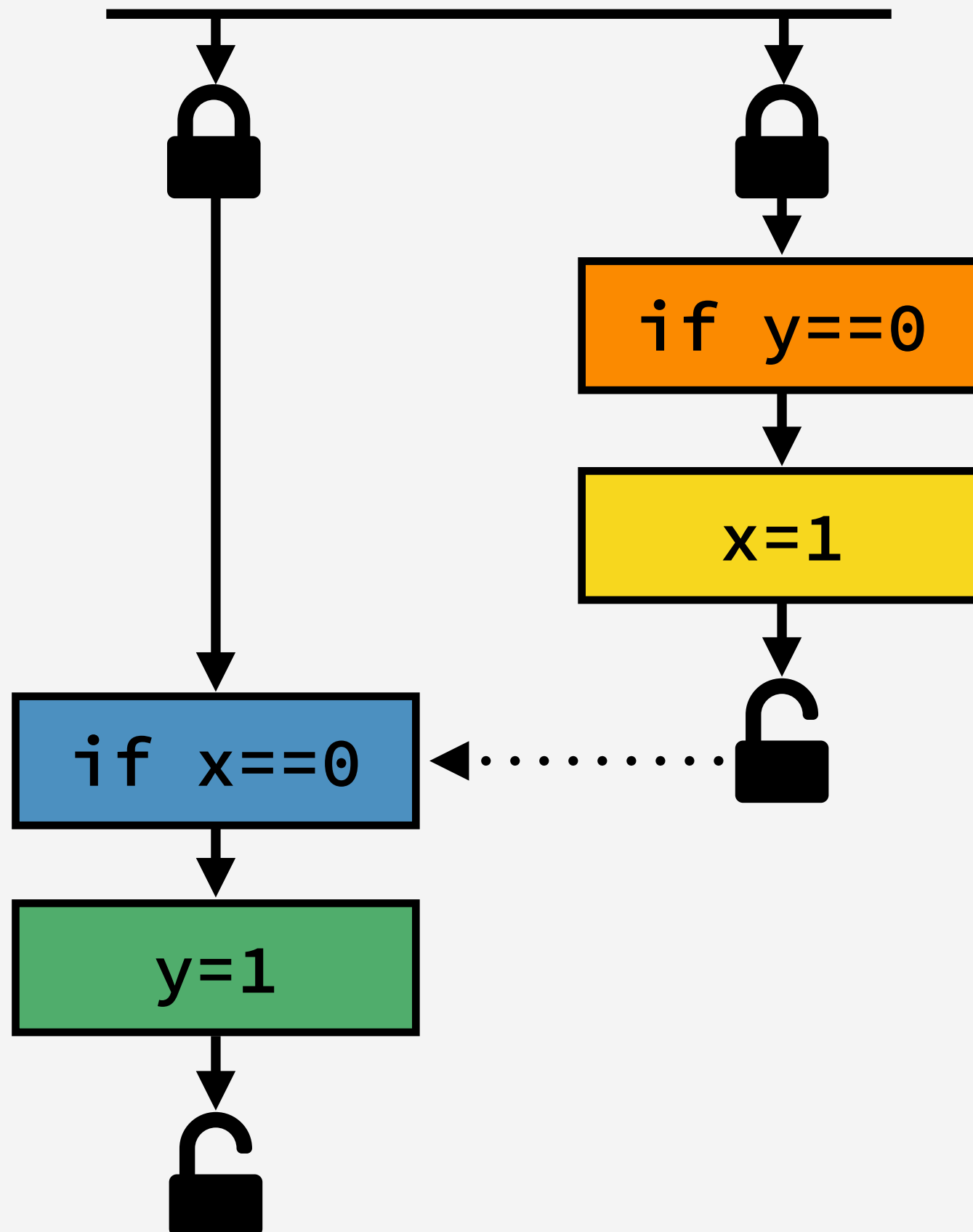


We can fix this program with locks.

Locks

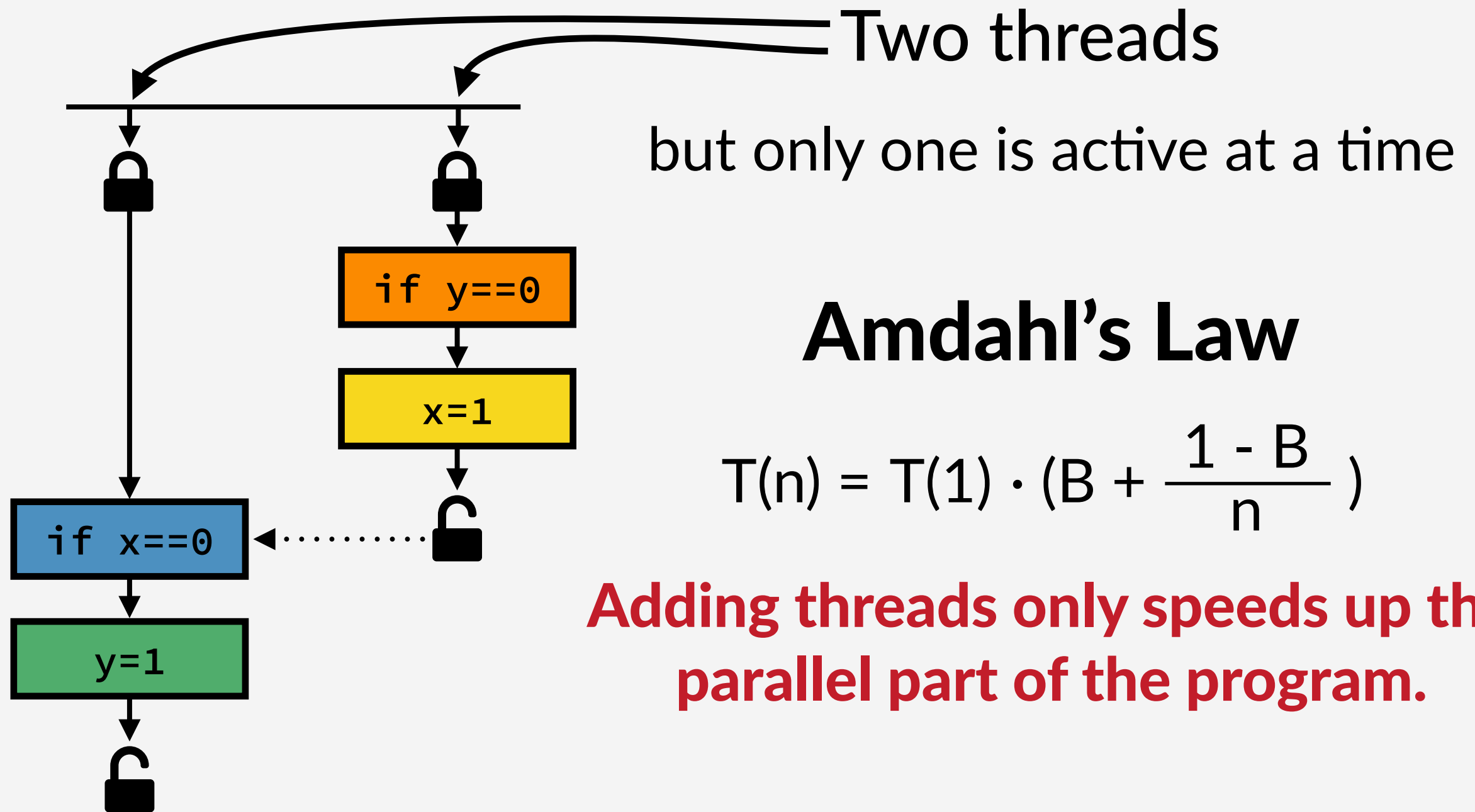


Locks

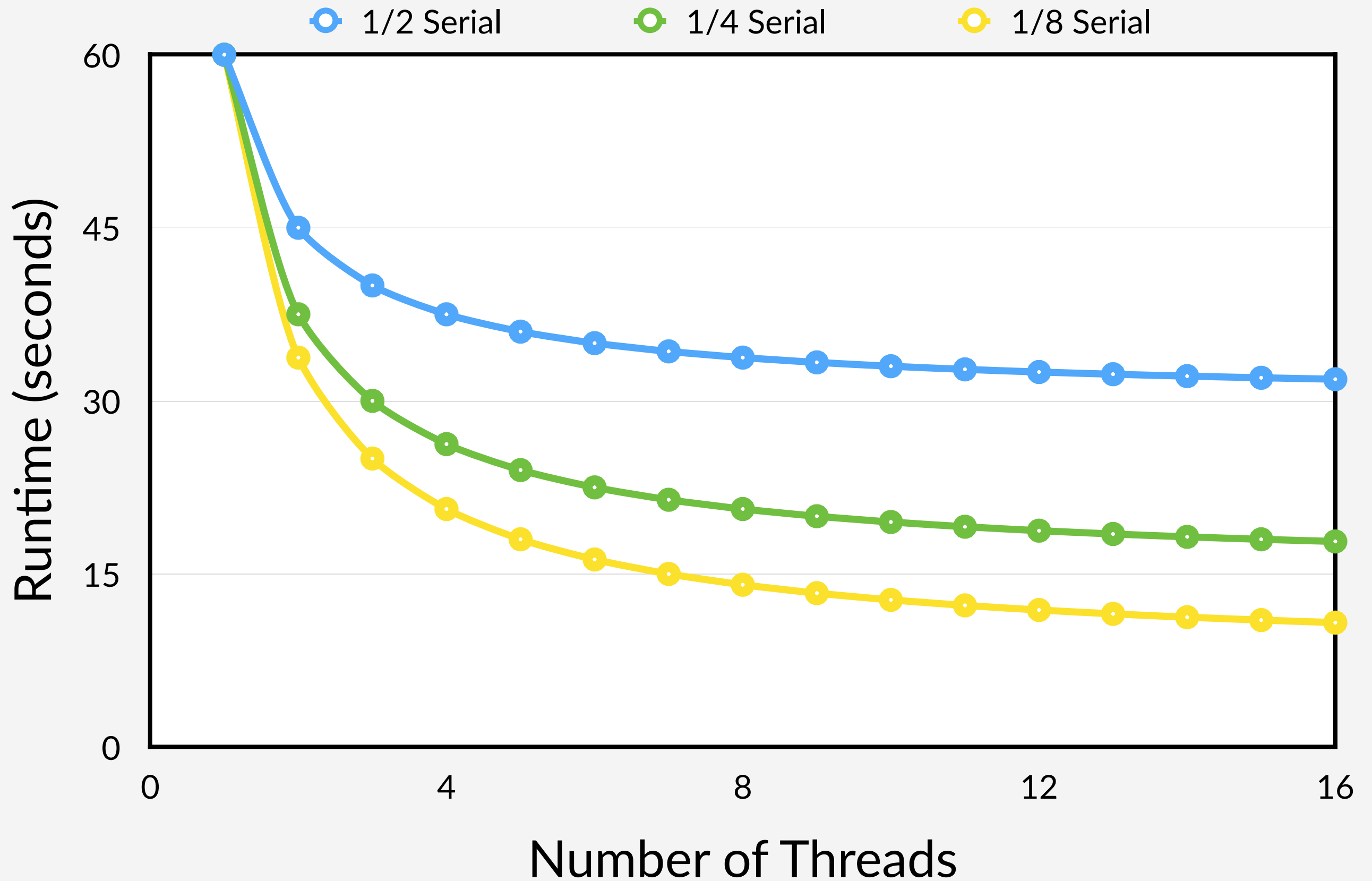


Locks

Atomicity comes at a performance cost

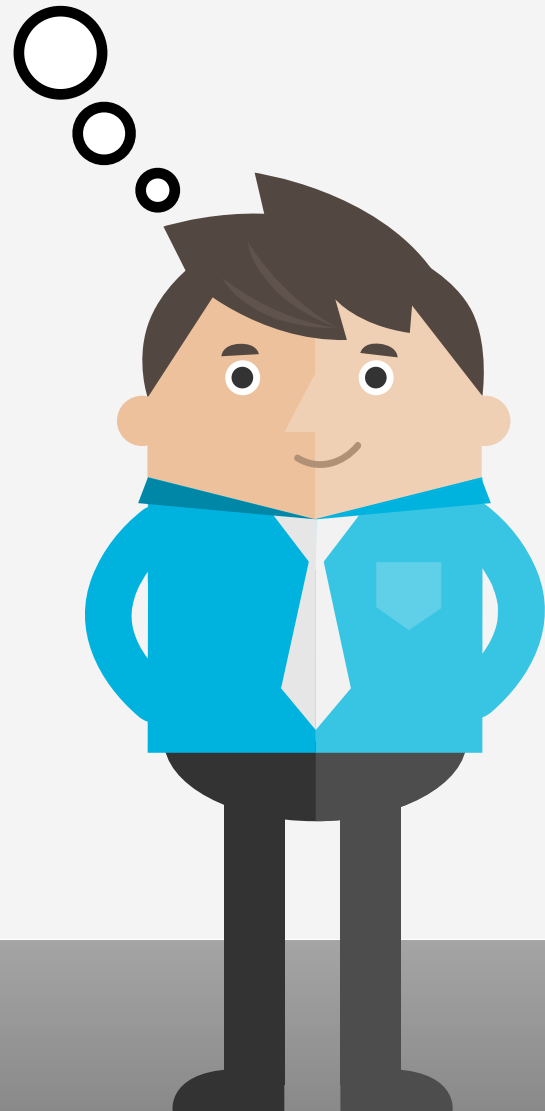


Amdahl's Law



Contention

I wonder if we have
any yogurt

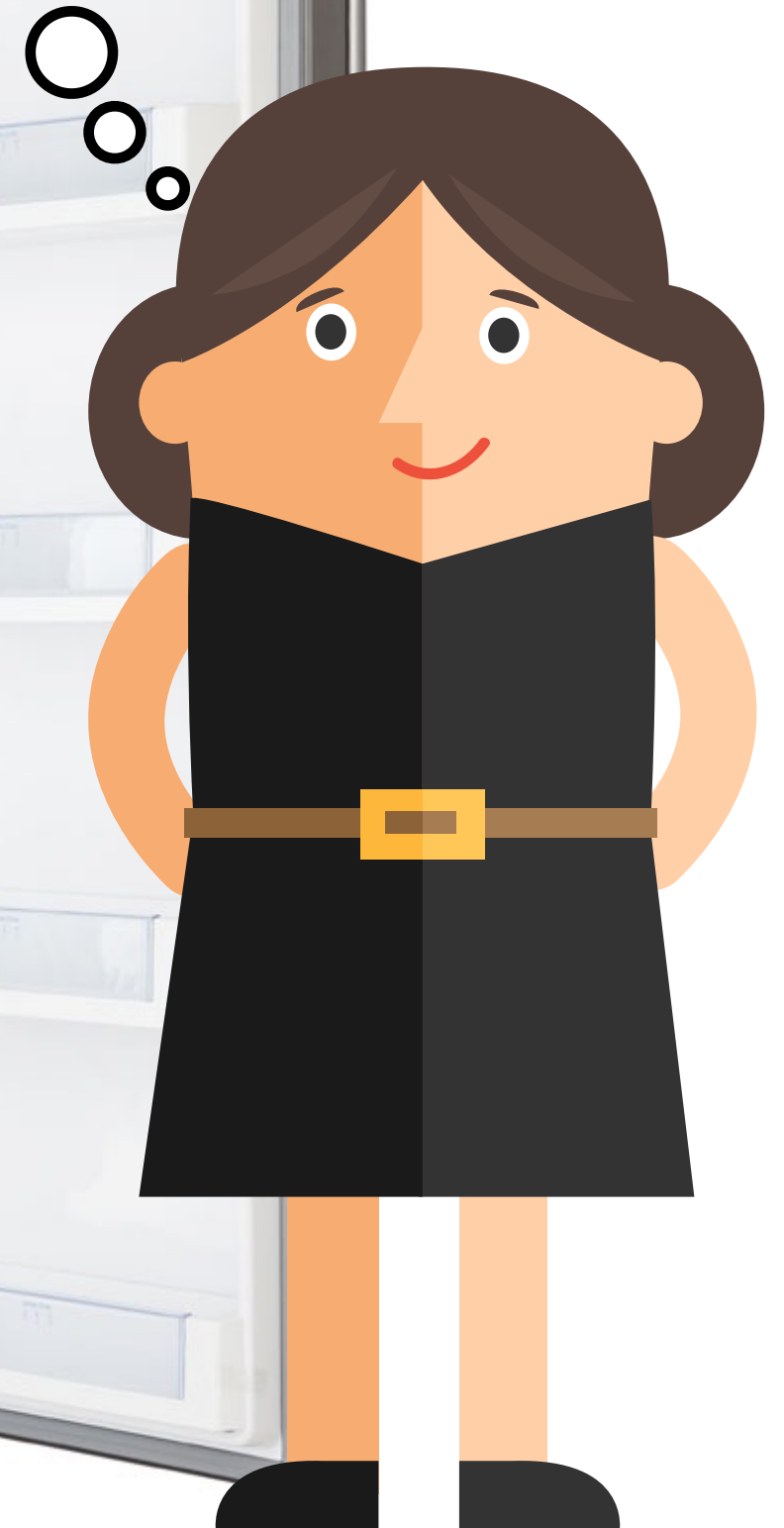


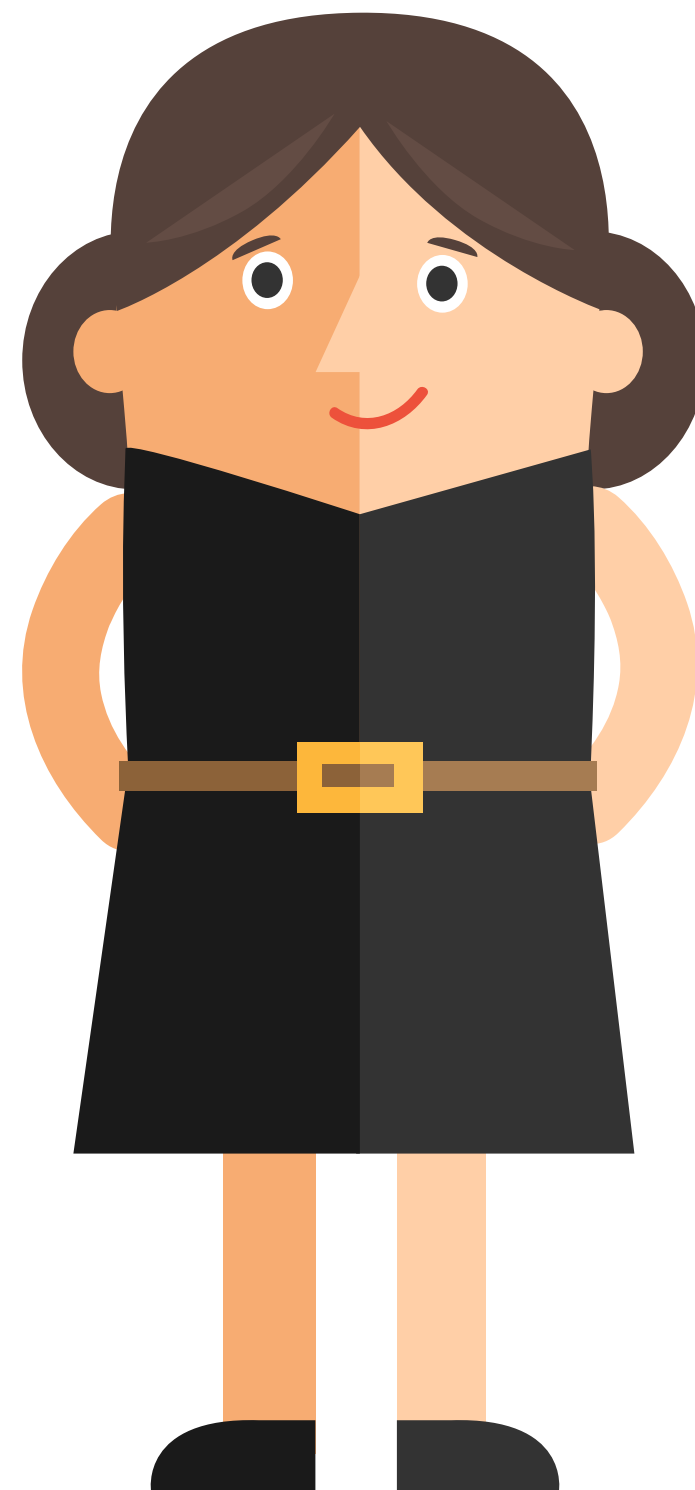
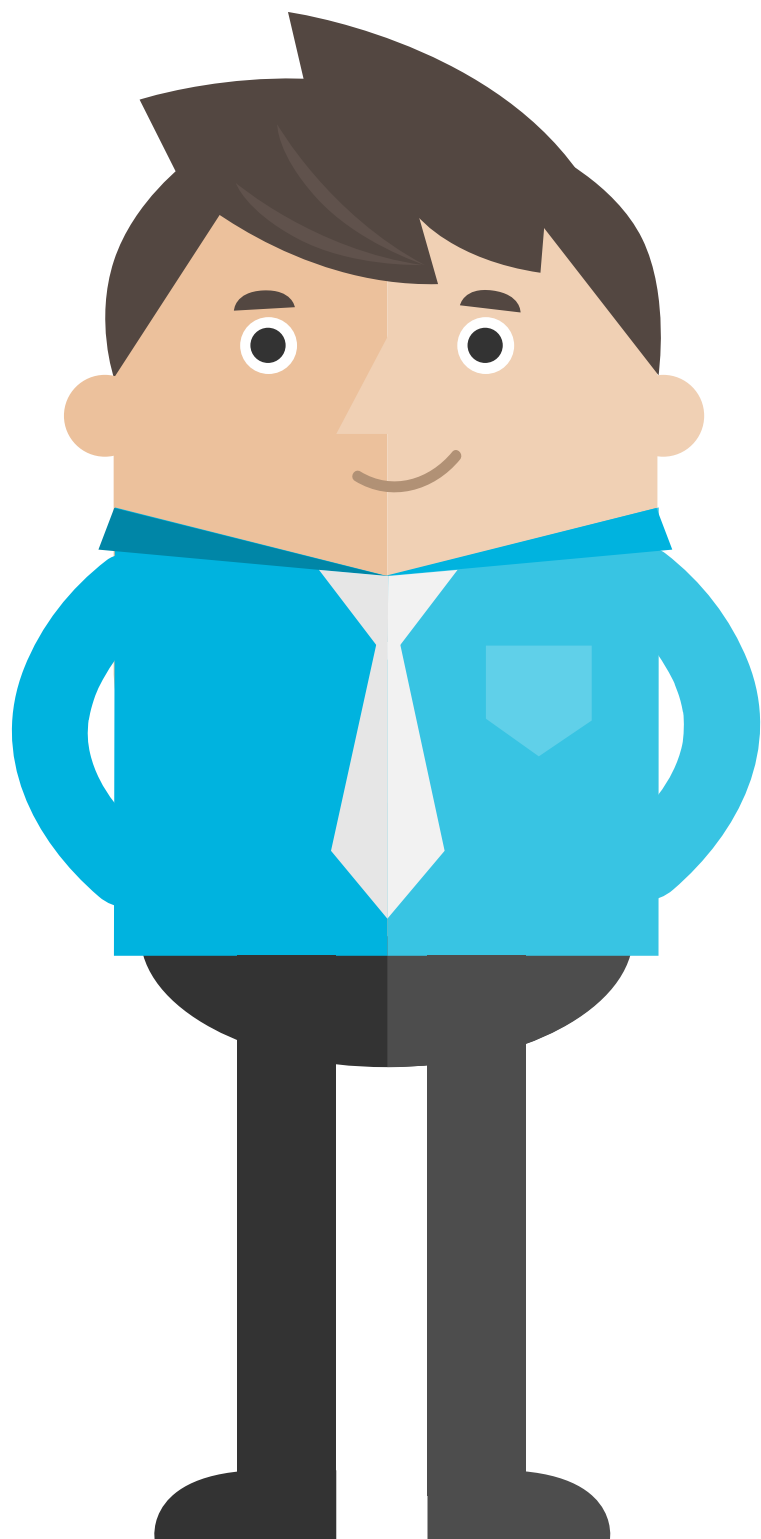
I'll make sure we
have bread and milk





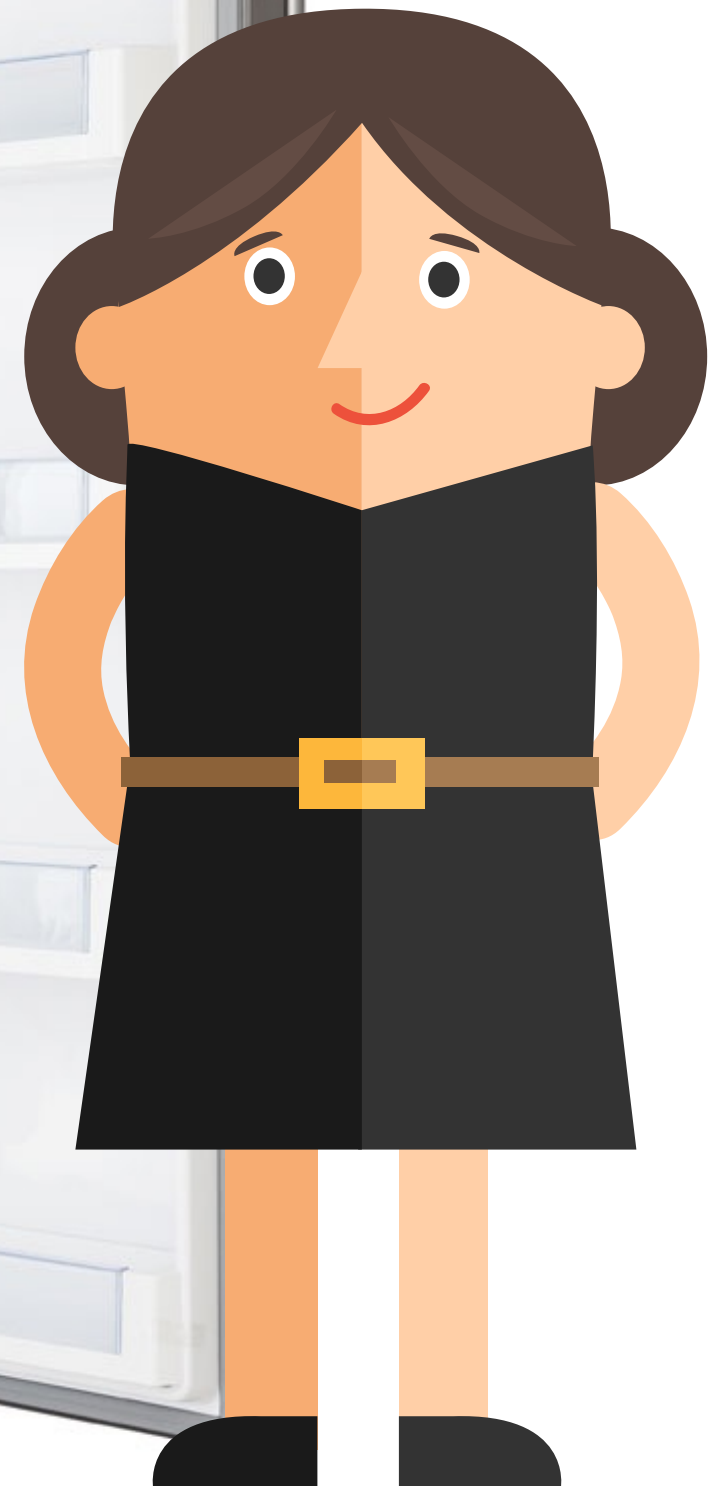
Oh no! Better go to
the store.



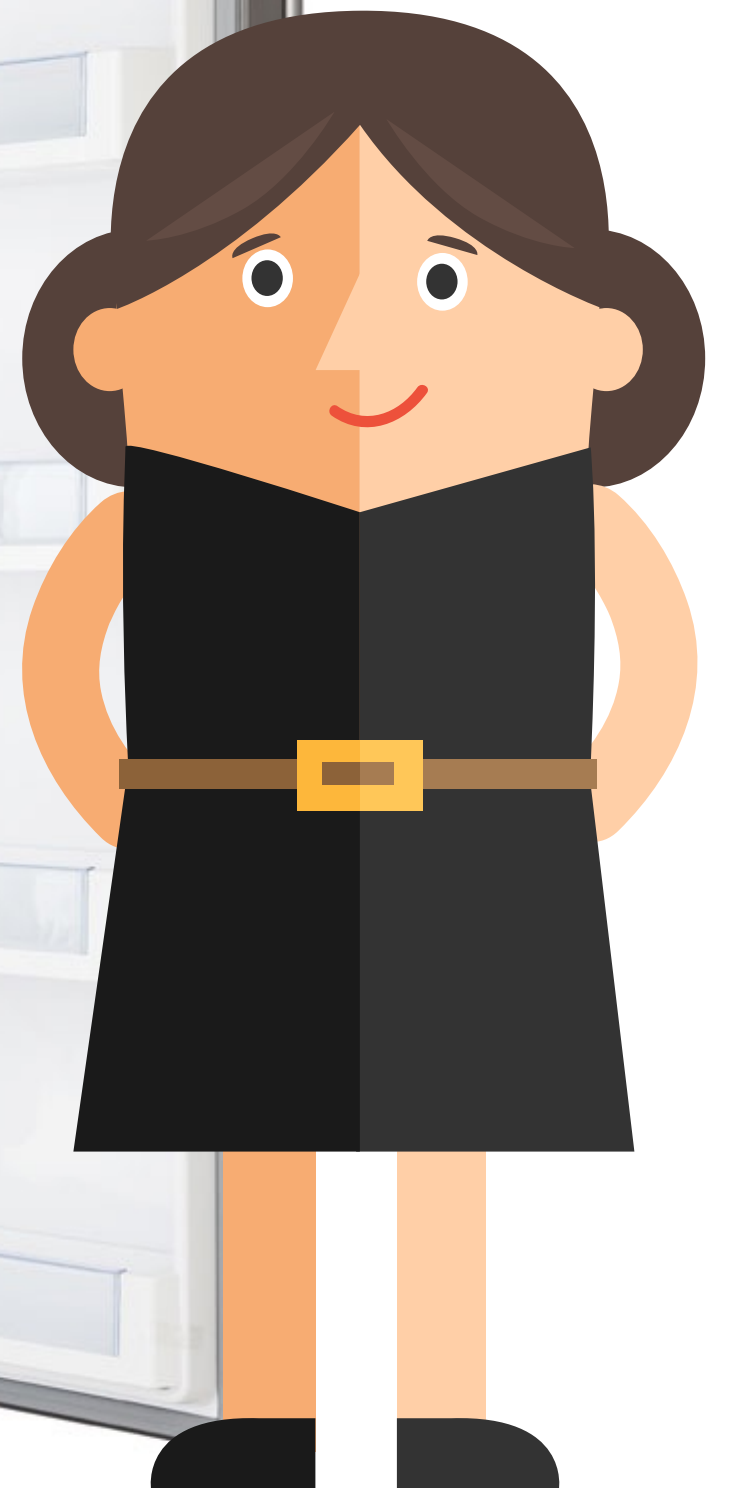
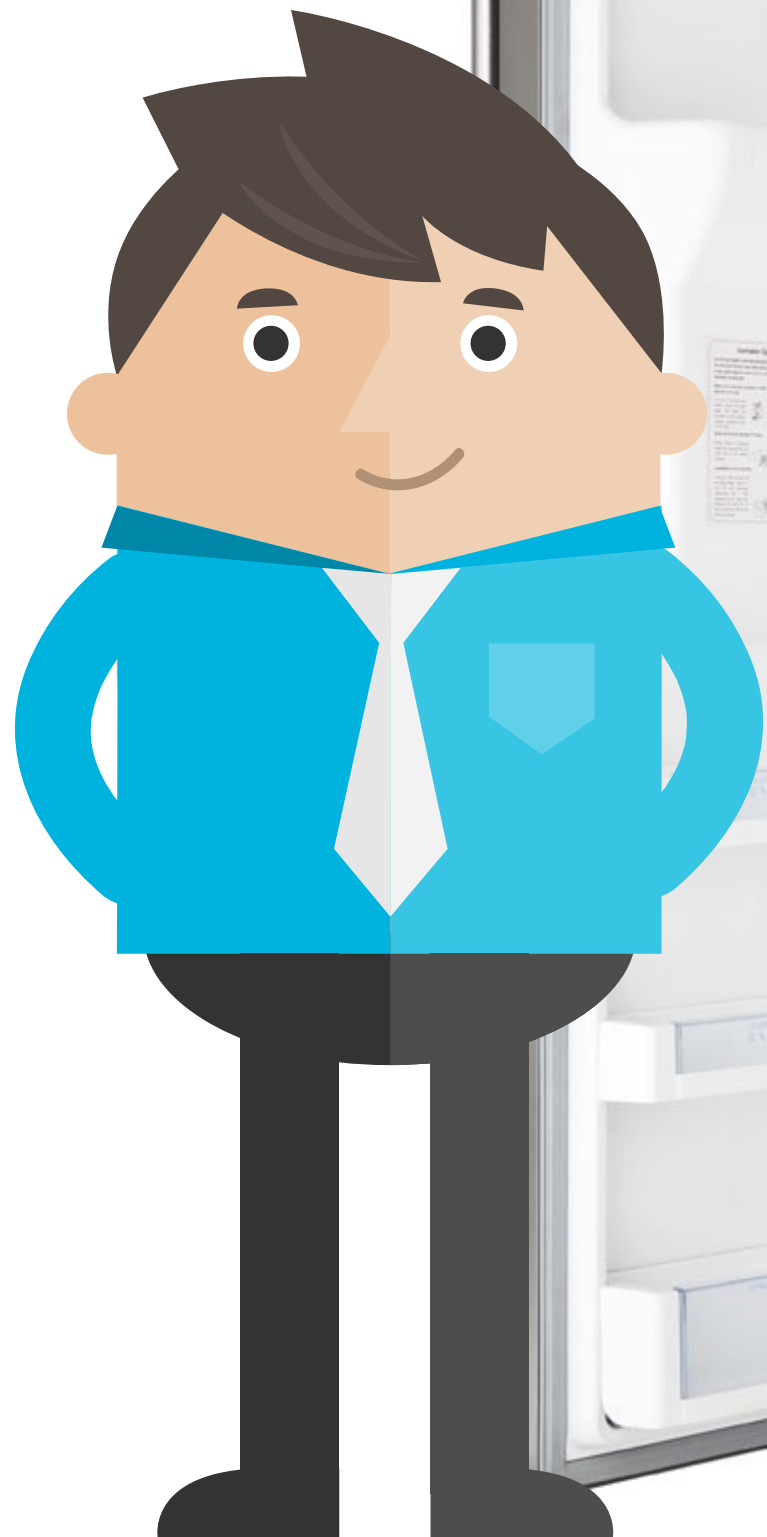




Darn, no yogurt.
Better go to the store.

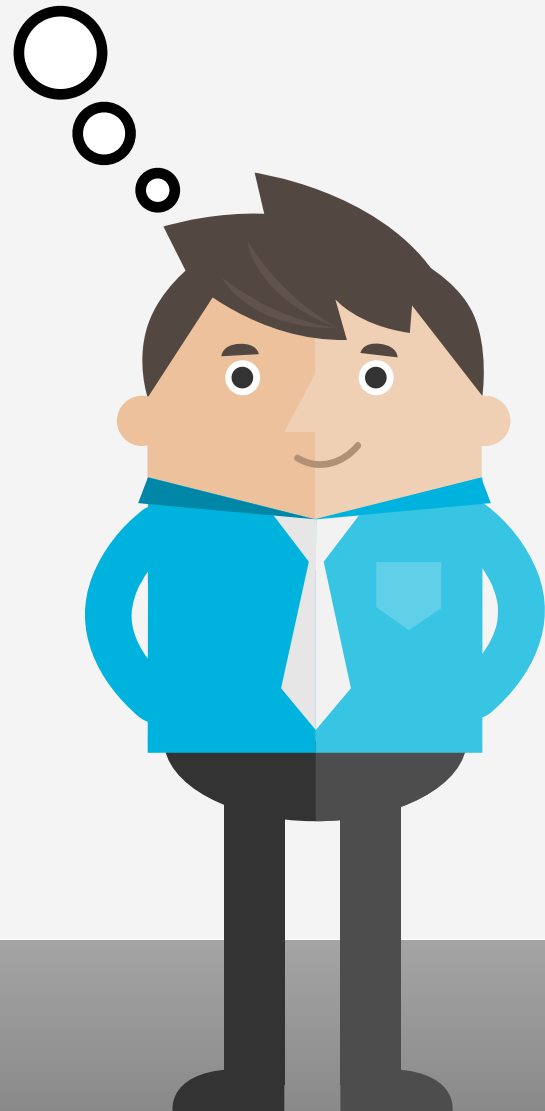






Fine-grained locking

I wonder if we have
any yogurt

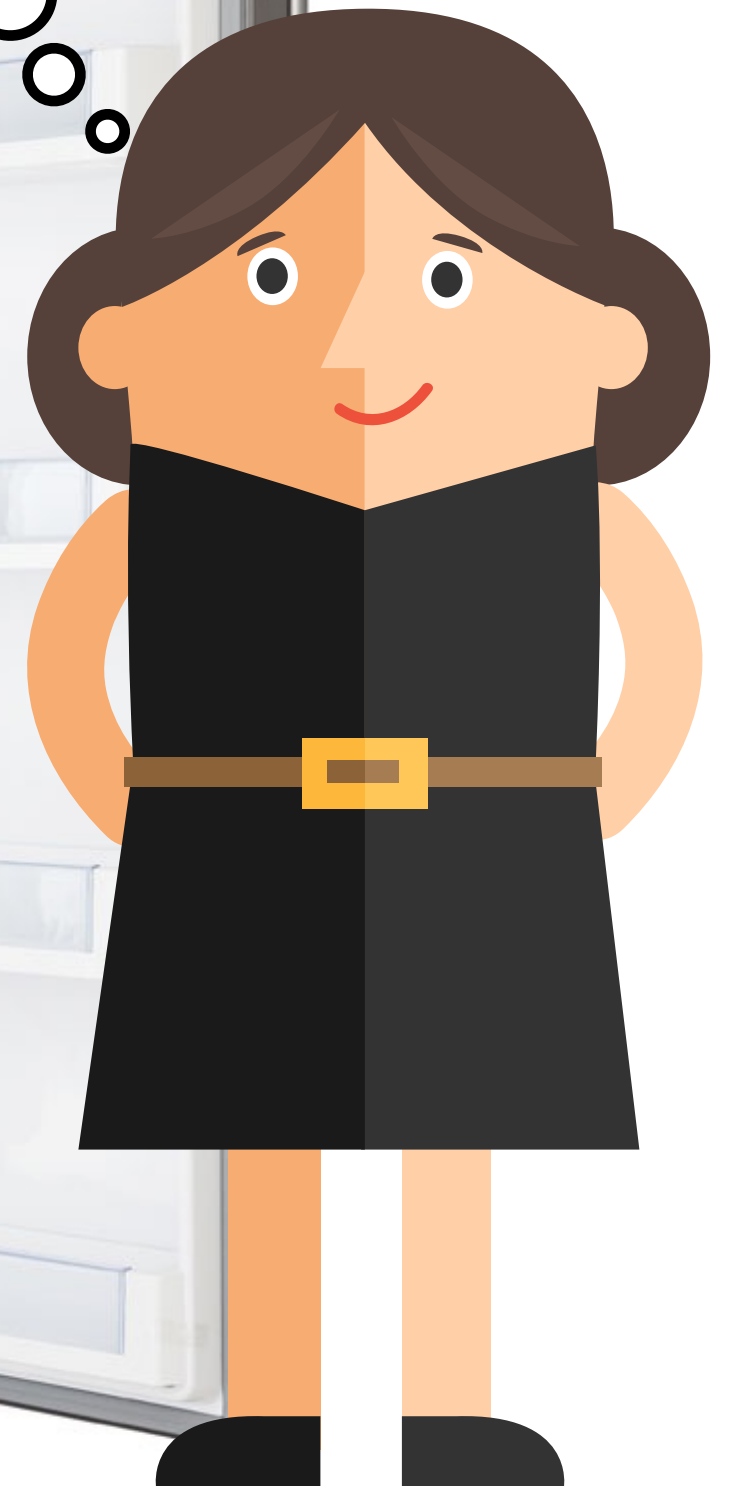


I'll make sure we
have bread and milk



Darn, no yogurt.
Better go to the store.

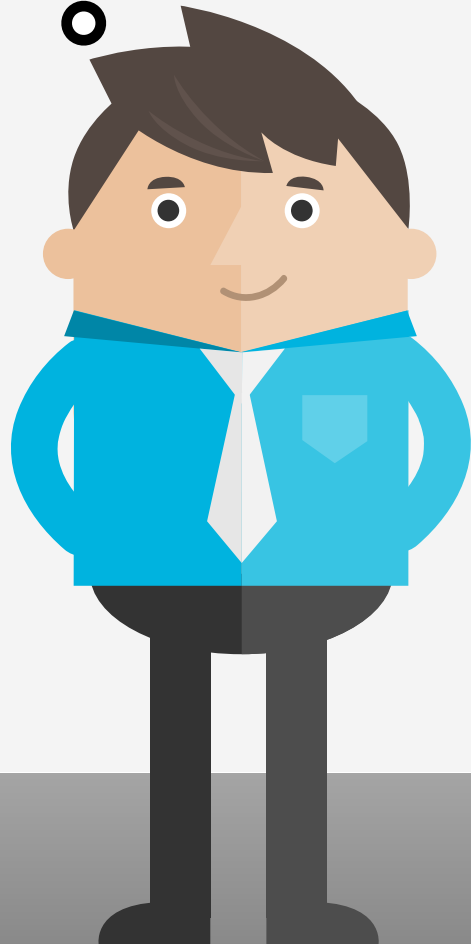
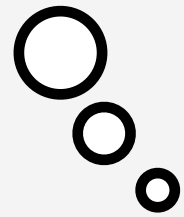
Oh no! Better go to
the store.



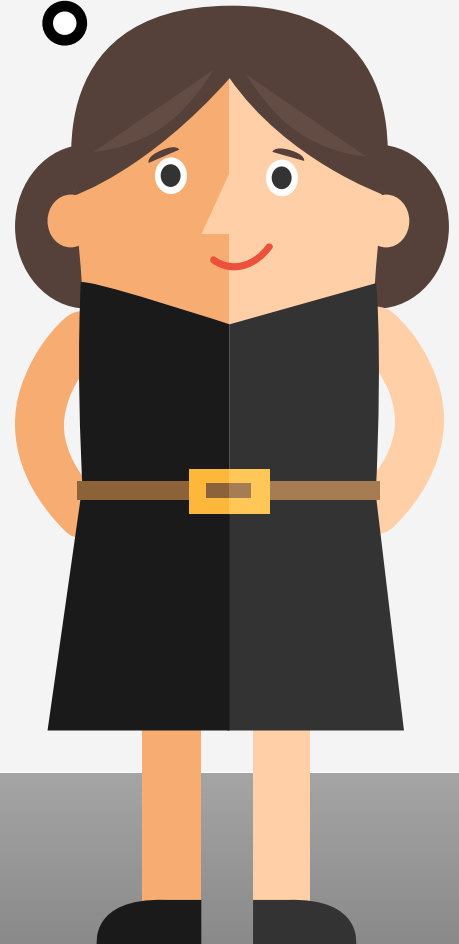
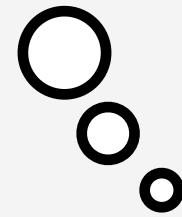


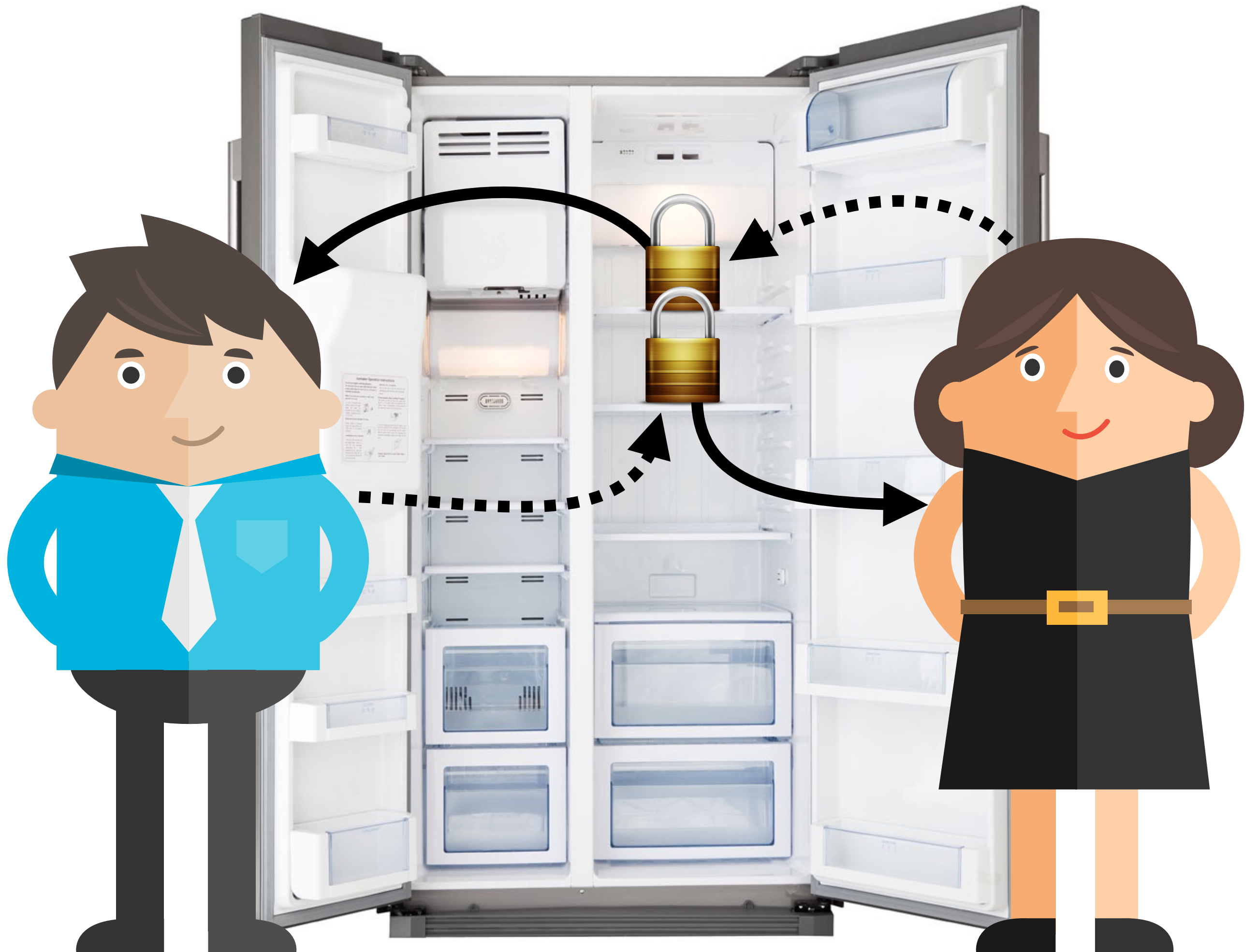
Deadlock

Buy milk and bread.



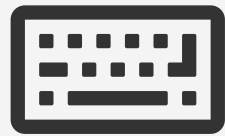
Buy bread and milk.



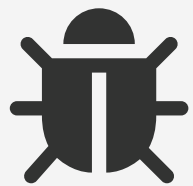




Why we have to write parallel programs



How to write parallel programs



Why parallel programming is hard