

No Busy-wait.

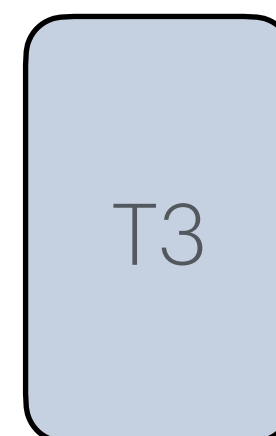
no busy-wait

no busy-wait

Timestep: 0

```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                  // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {  // (2)  
        take_thread_off_wait_queue(); // (3)  
        place_thread_on_ready_queue(); // (4)  
    } else {  
        *lock = FREE;                 // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



shared
resource

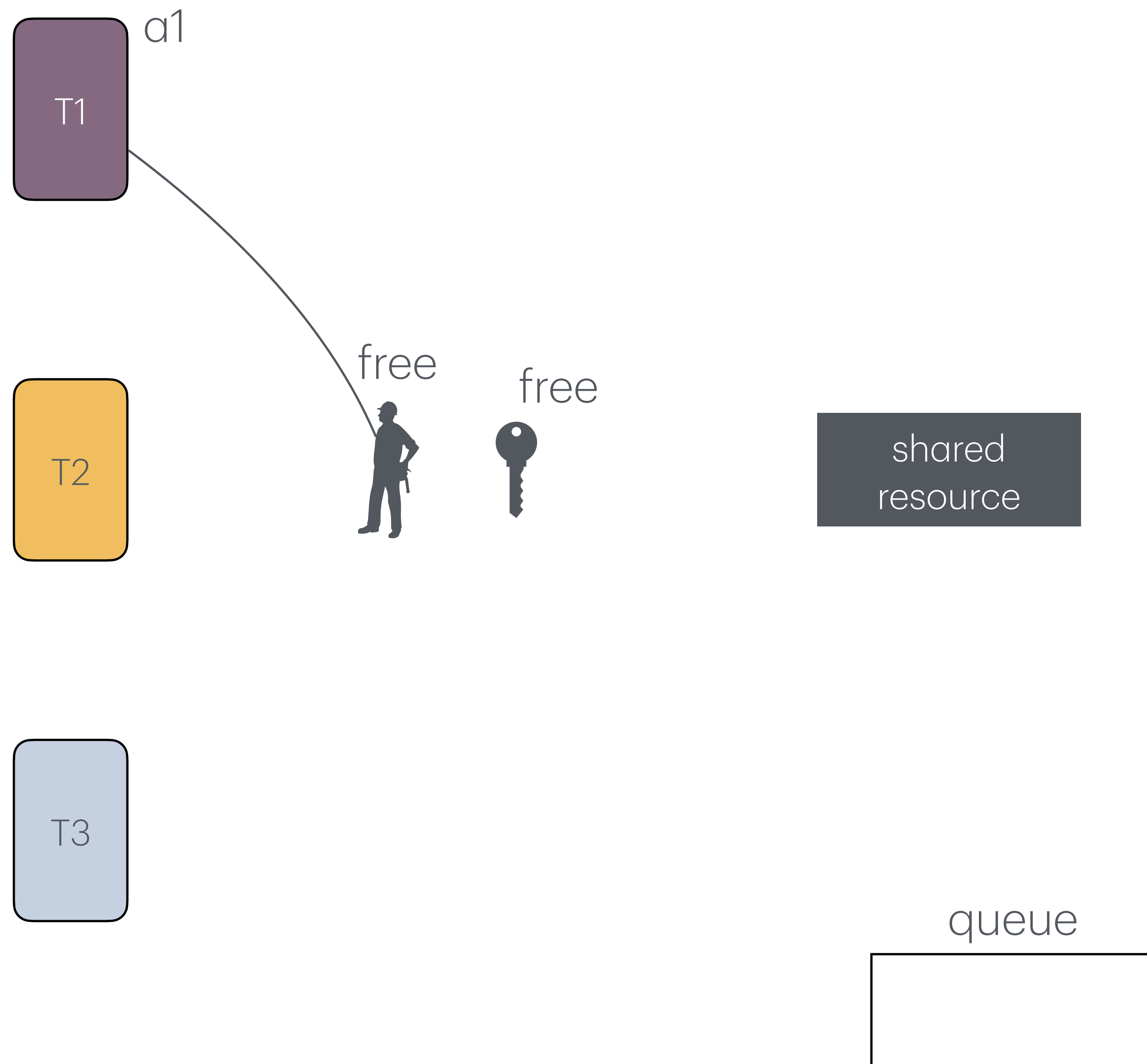
queue

A white rectangle with a black border, representing a queue.

Timestep: 1

```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                  // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

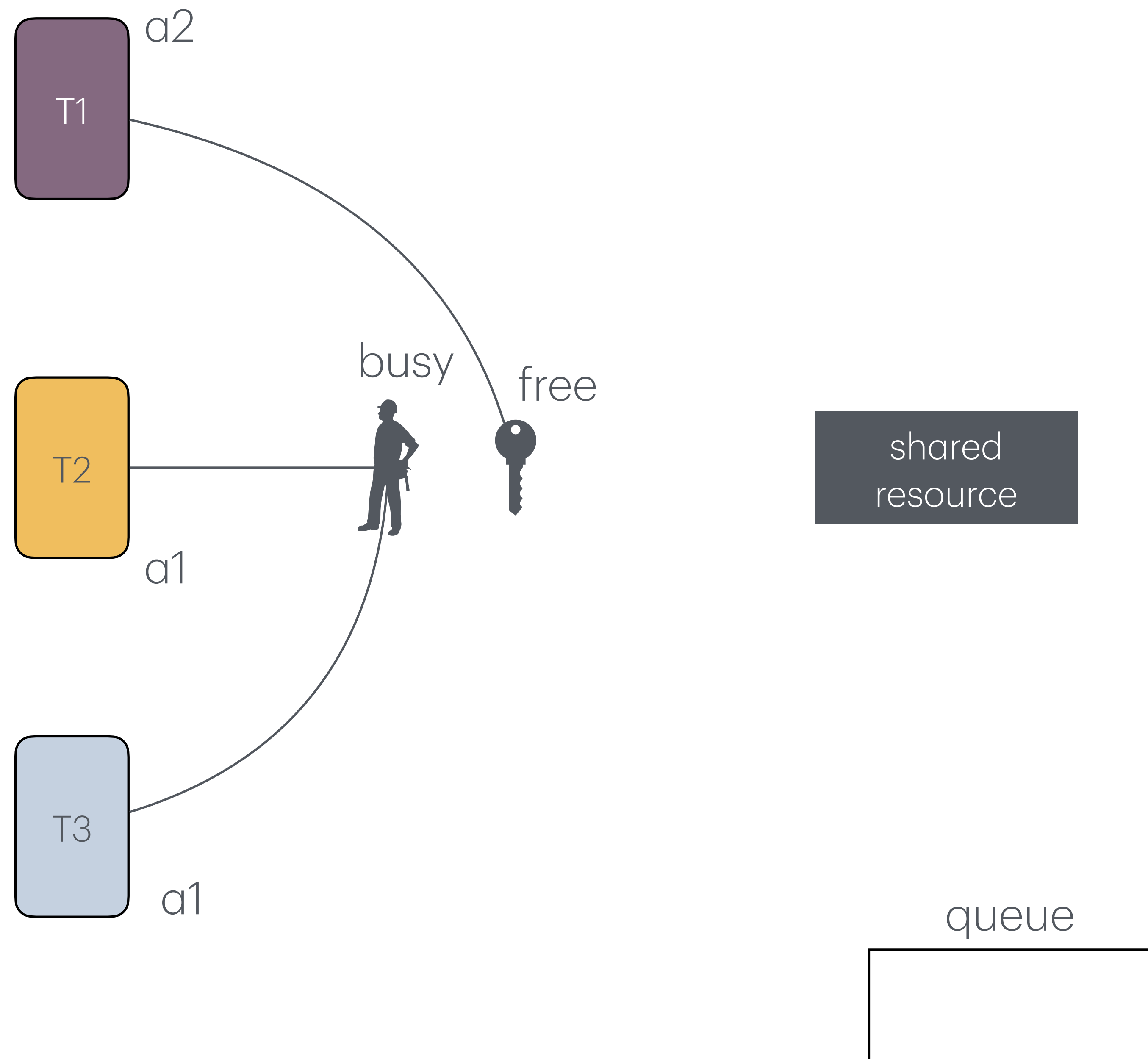
```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {   // (2)  
        take_thread_off_wait_queue();   // (3)  
        place_thread_on_ready_queue();  // (4)  
    } else {  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



Timestep: 2

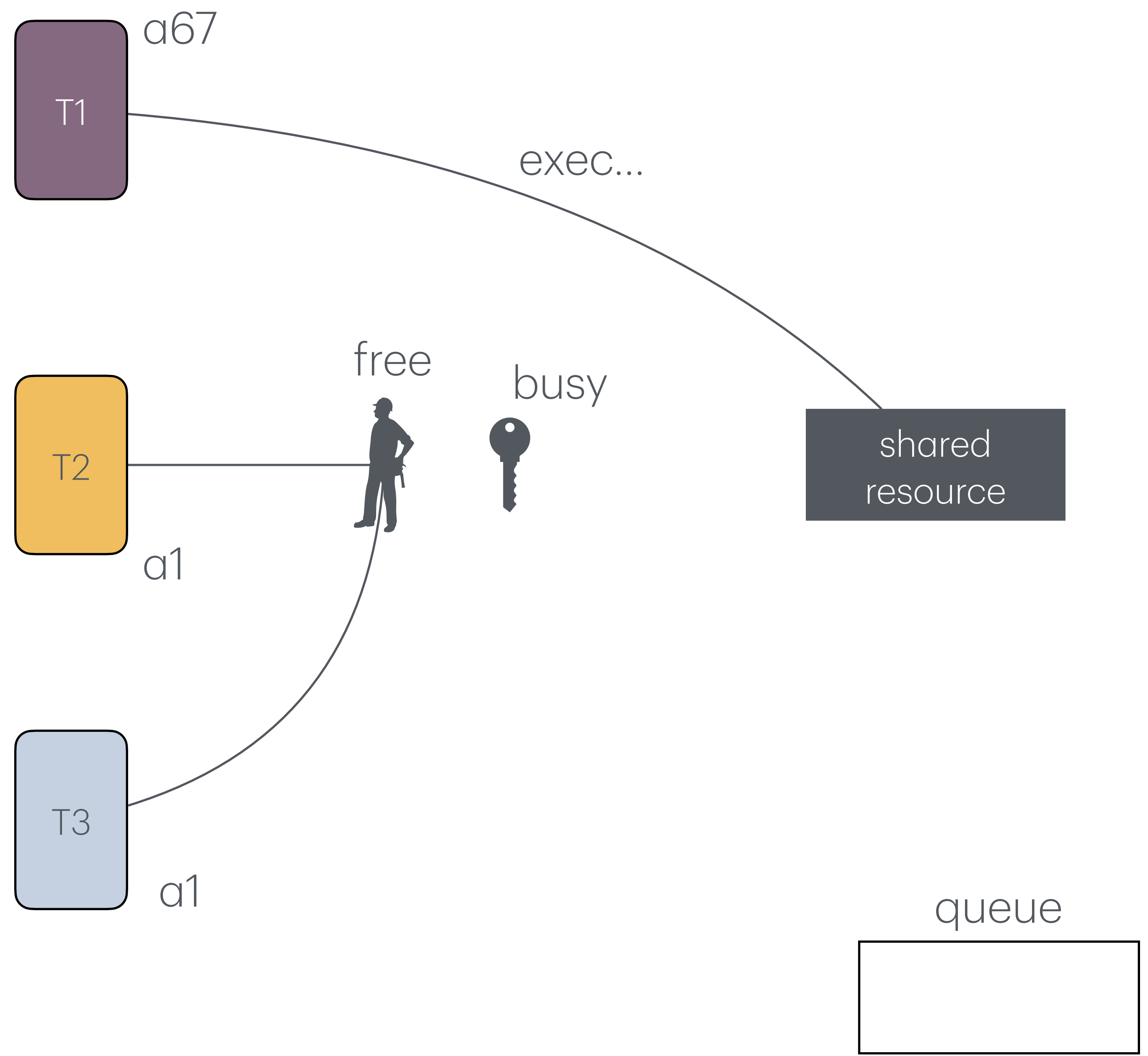
```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                 // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {  // (2)  
        take_thread_off_wait_queue(); // (3)  
        place_thread_on_ready_queue(); // (4)  
    } else {  
        *lock = FREE;                 // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



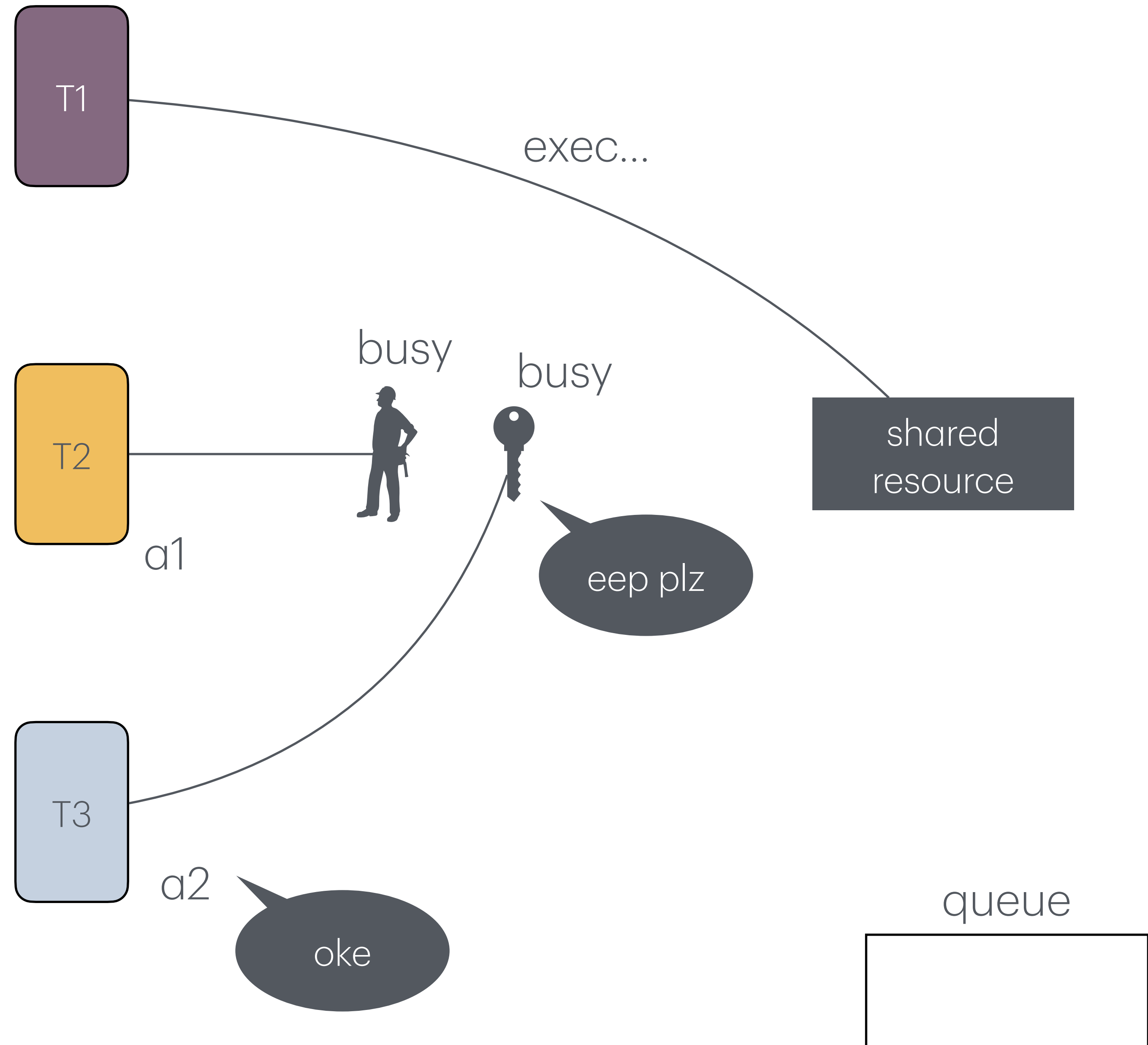
```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                  // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {  // (2)  
        take_thread_off_wait_queue();  // (3)  
        place_thread_on_ready_queue(); // (4)  
    } else {  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                  // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

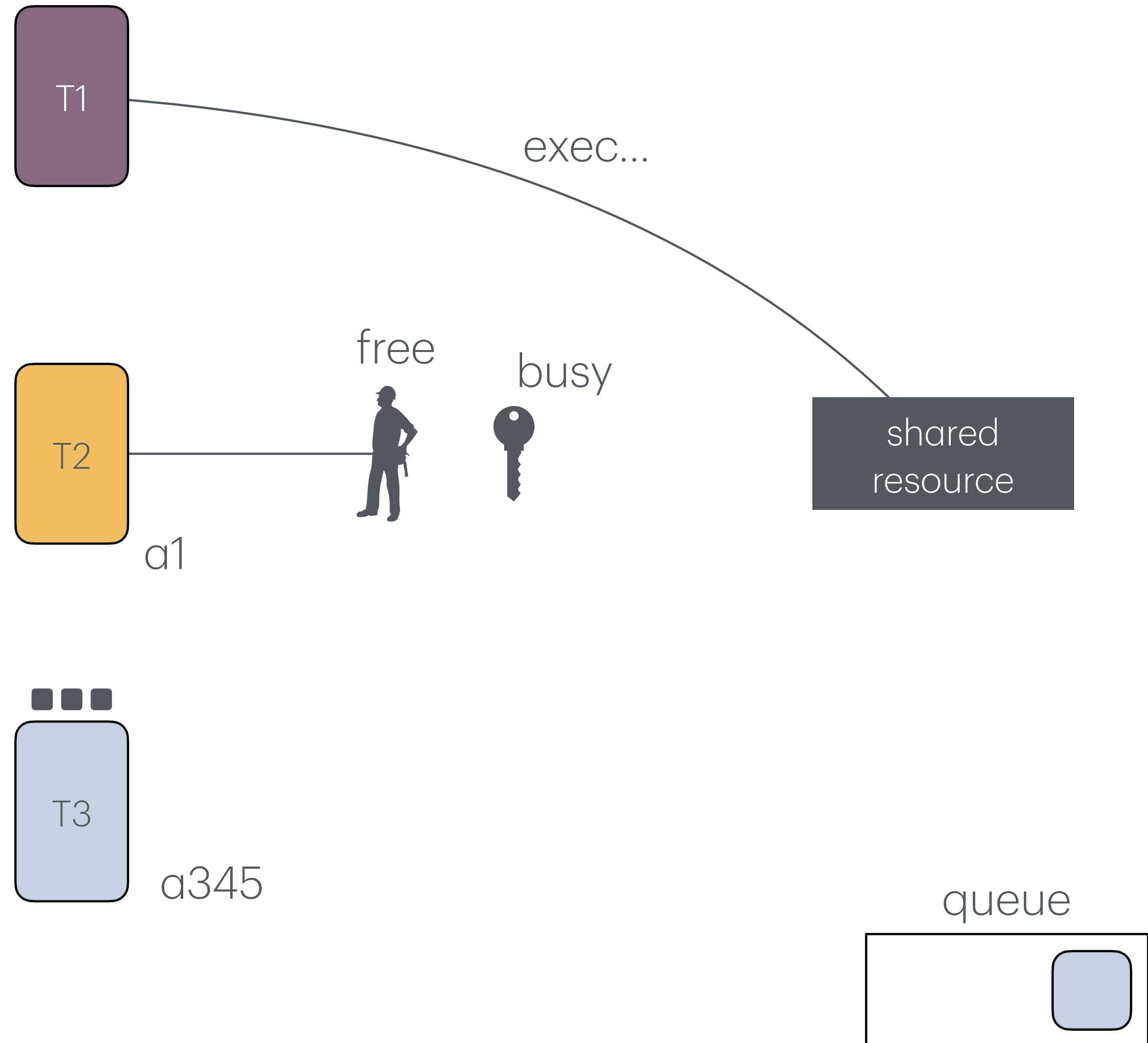
```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {  // (2)  
        take_thread_off_wait_queue();  // (3)  
        place_thread_on_ready_queue(); // (4)  
    } else {  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



Timestep: 5

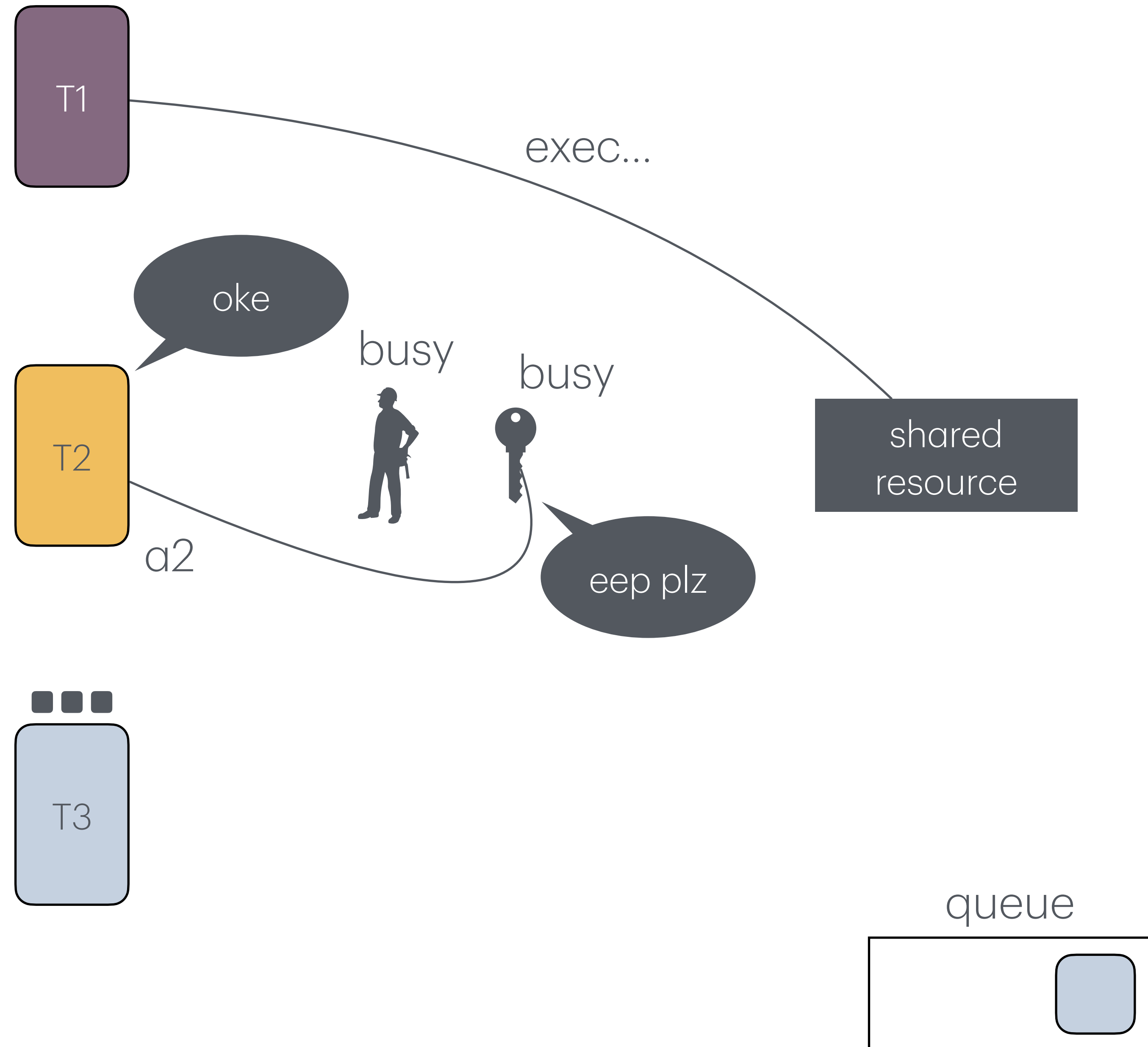
```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                 // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {  // (2)  
        take_thread_off_wait_queue();  // (3)  
        place_thread_on_ready_queue(); // (4)  
    } else {  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                  // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

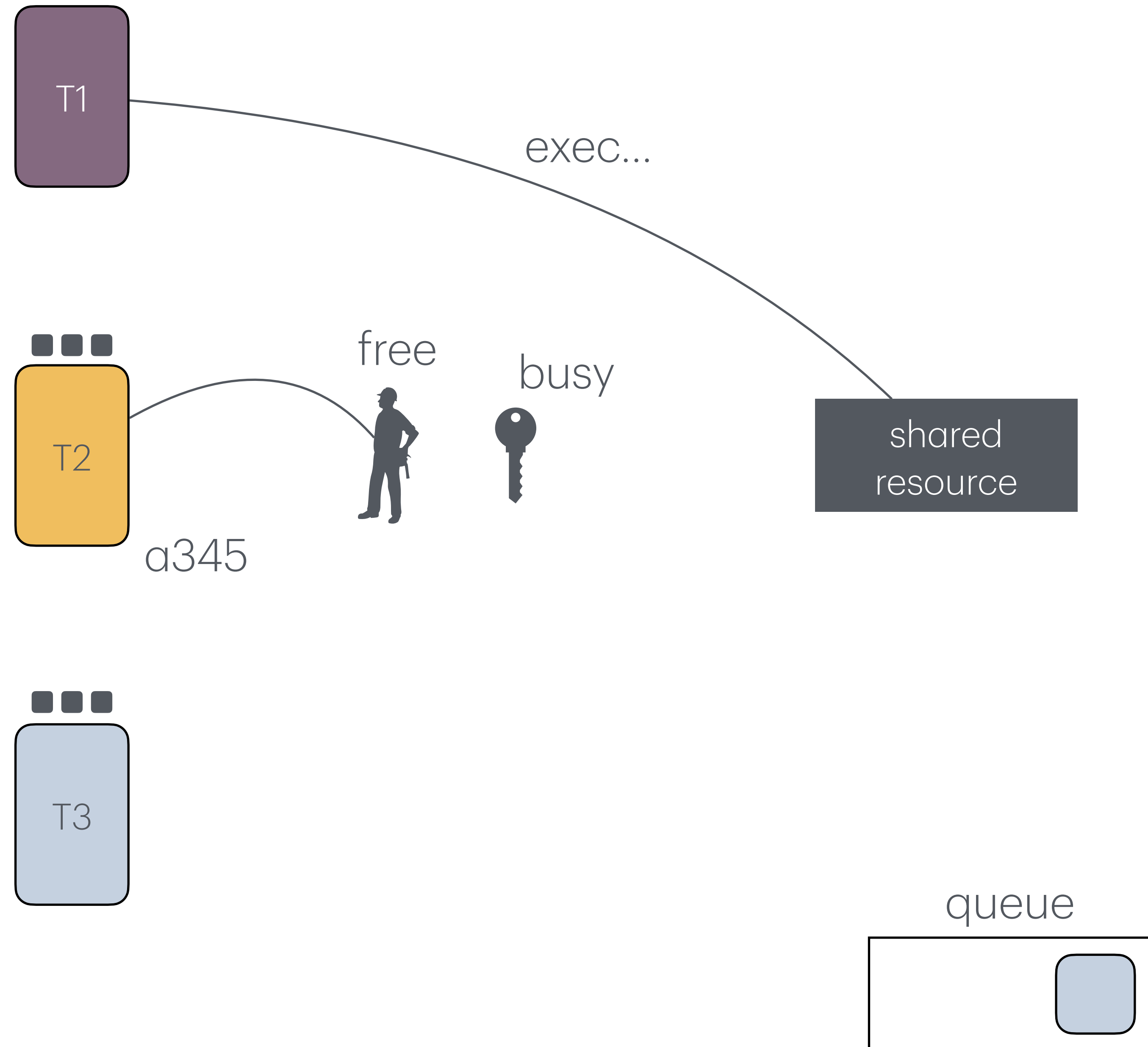
```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {   // (2)  
        take_thread_off_wait_queue();   // (3)  
        place_thread_on_ready_queue();  // (4)  
    } else {  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



Timestep: 7

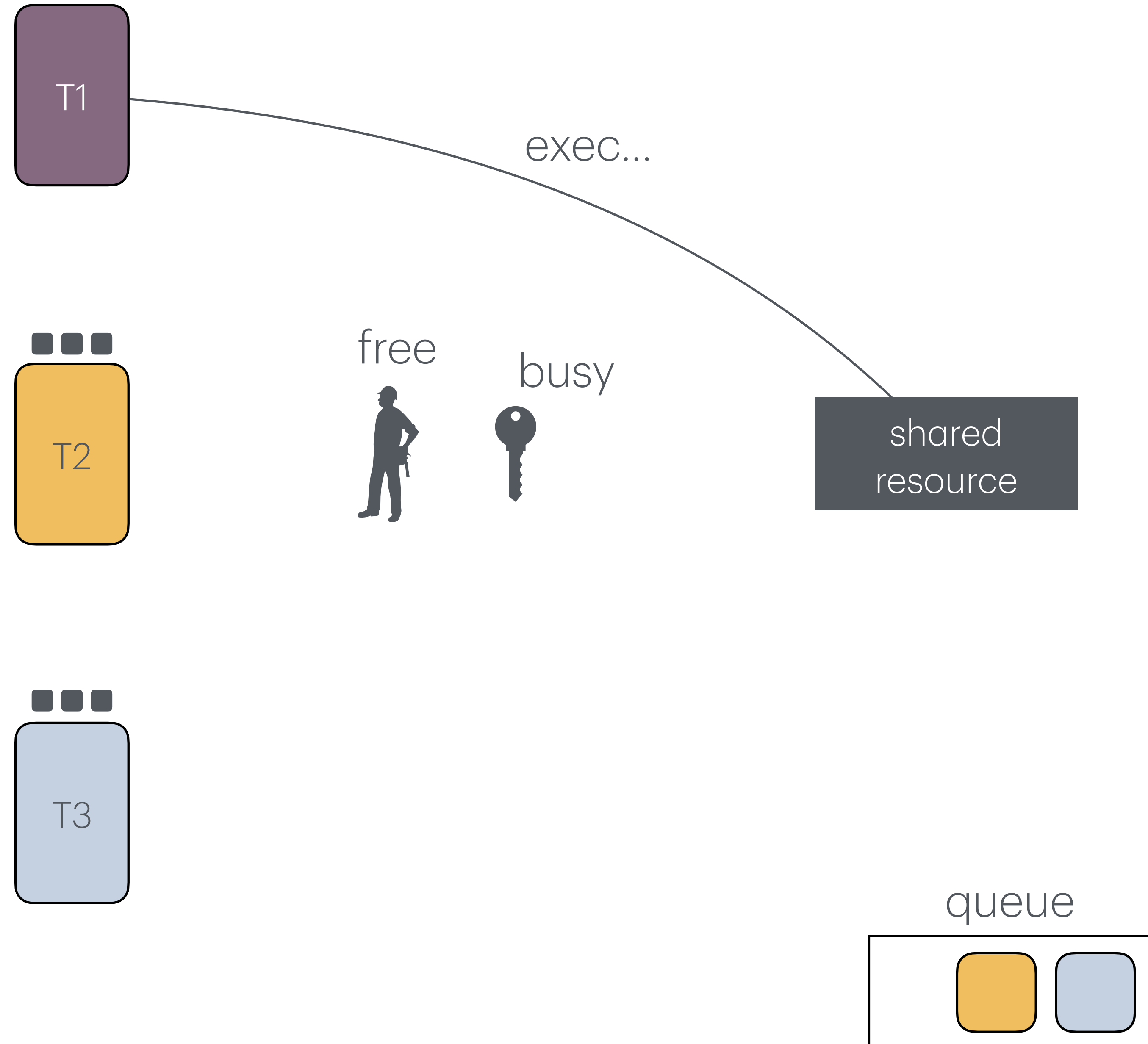
```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                 // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {  // (2)  
        take_thread_off_wait_queue();  // (3)  
        place_thread_on_ready_queue(); // (4)  
    } else {  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



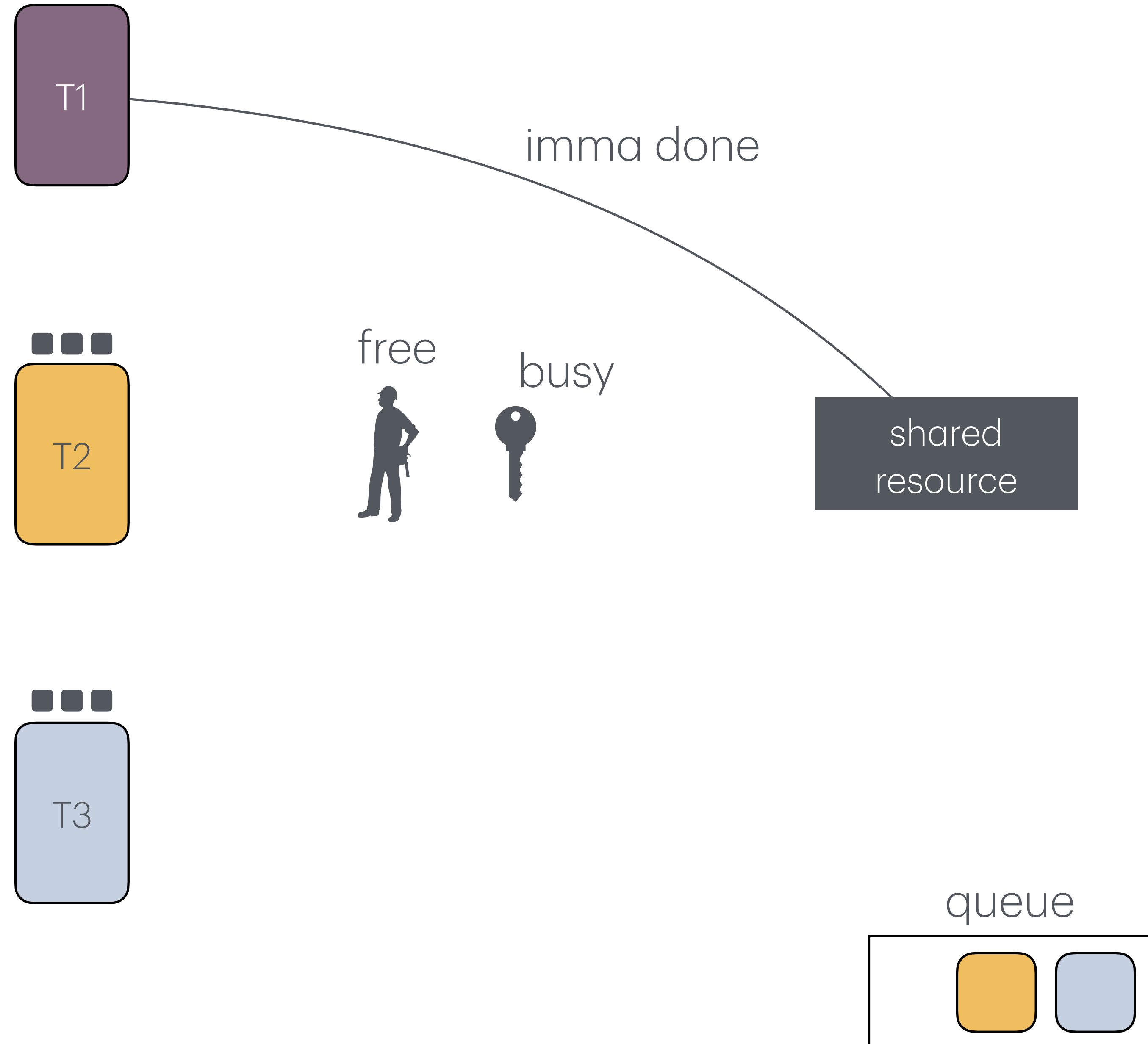
```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                  // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {   // (2)  
        take_thread_off_wait_queue();   // (3)  
        place_thread_on_ready_queue();  // (4)  
    } else {  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



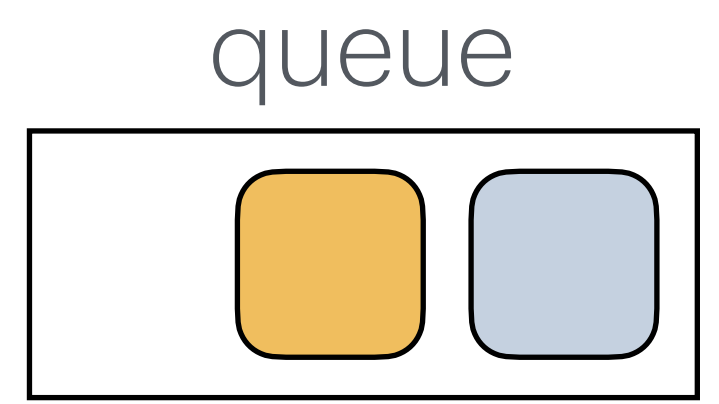
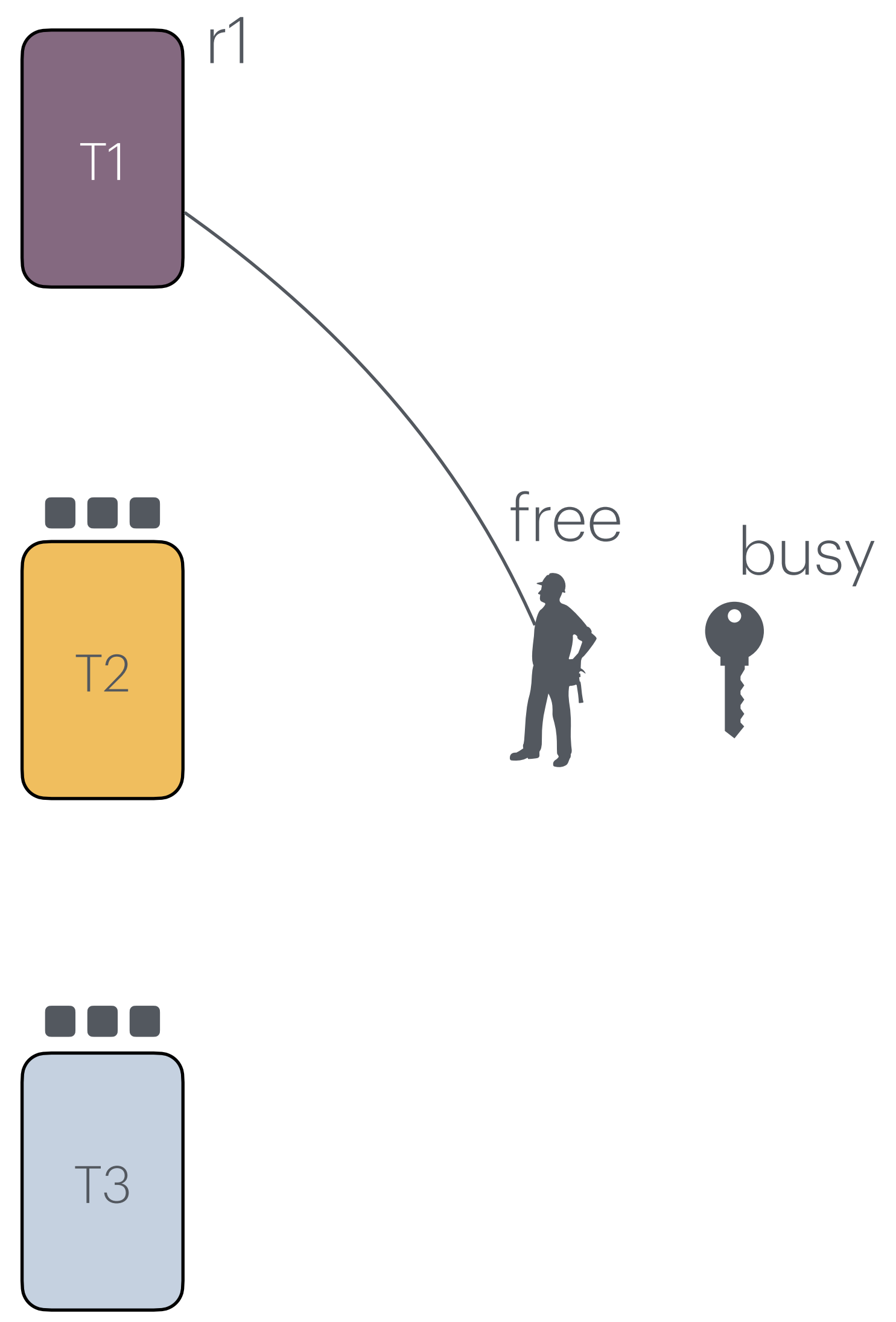
```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {               // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                 // (5)  
    } else {                           // (6)  
        *lock = BUSY;                  // (7)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {  // (2)  
        take_thread_off_wait_queue();  // (3)  
        place_thread_on_ready_queue(); // (4)  
    } else {                           // (5)  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



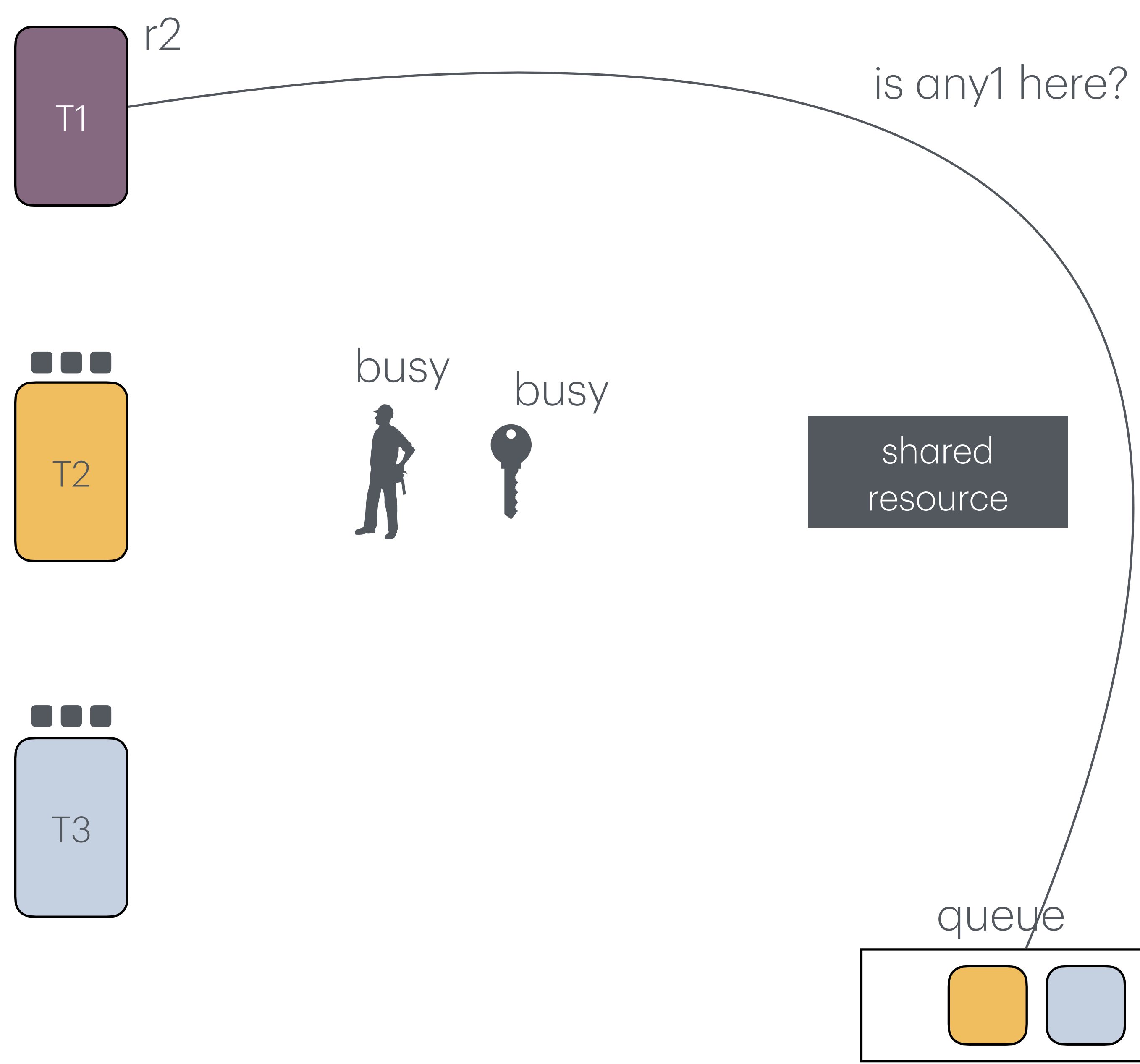
```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                 // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {   // (2)  
        take_thread_off_wait_queue();   // (3)  
        place_thread_on_ready_queue();  // (4)  
    } else {  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {               // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                 // (5)  
    } else {  
        *lock = BUSY;                 // (6)  
        guard = FREE;                 // (7)  
    }  
}
```

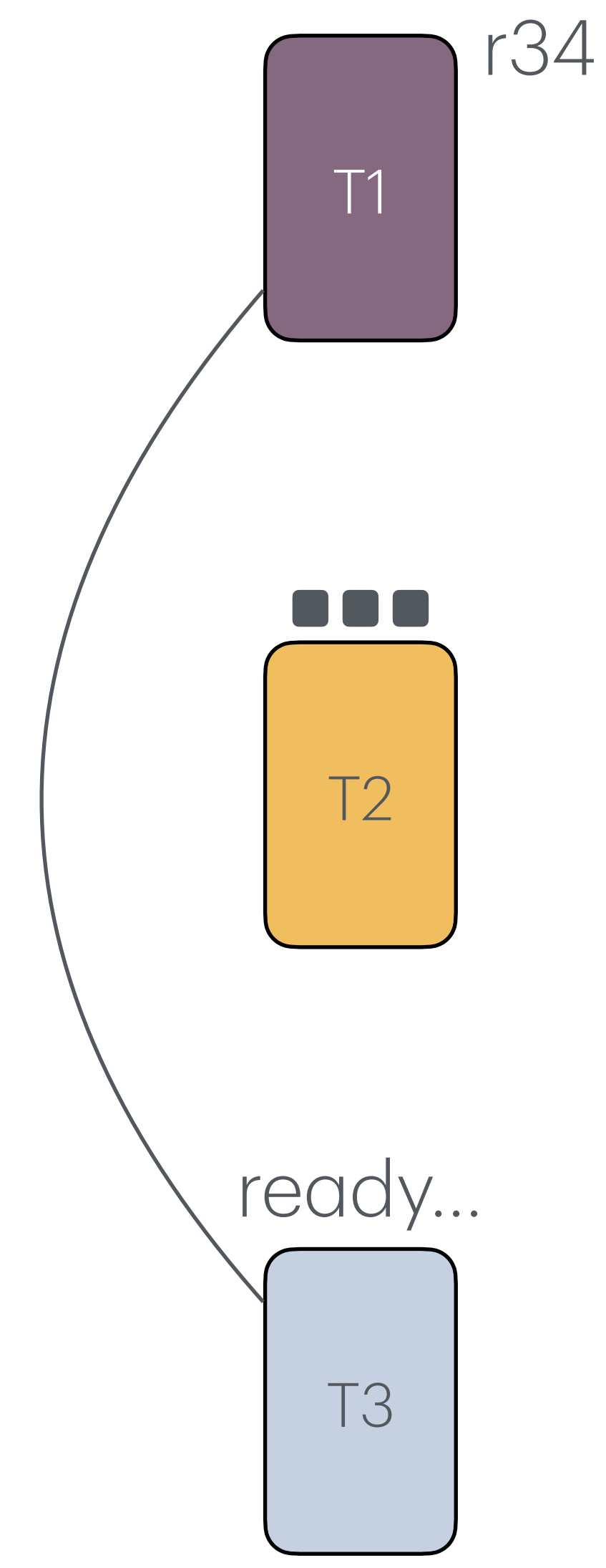
```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {   // (2)  
        take_thread_off_wait_queue();  // (3)  
        place_thread_on_ready_queue(); // (4)  
    } else {  
        *lock = FREE;                 // (5)  
    }  
    guard = FREE;                     // (6)  
}
```



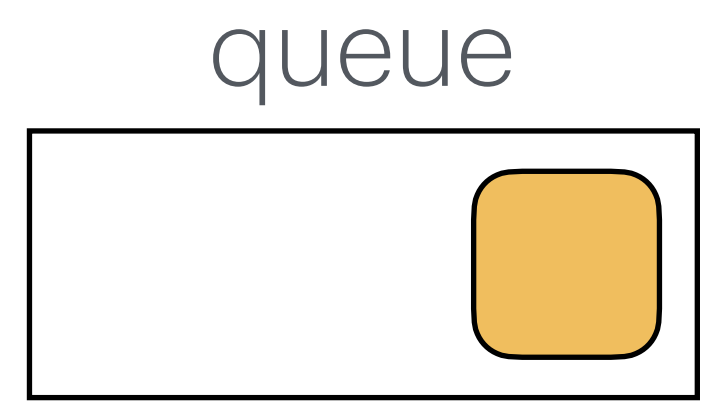

```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                 // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {   // (2)  
        take_thread_off_wait_queue();  // (3)  
        place_thread_on_ready_queue(); // (4)  
    } else {  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```

hey you
wakey wakey



shared
resource




```

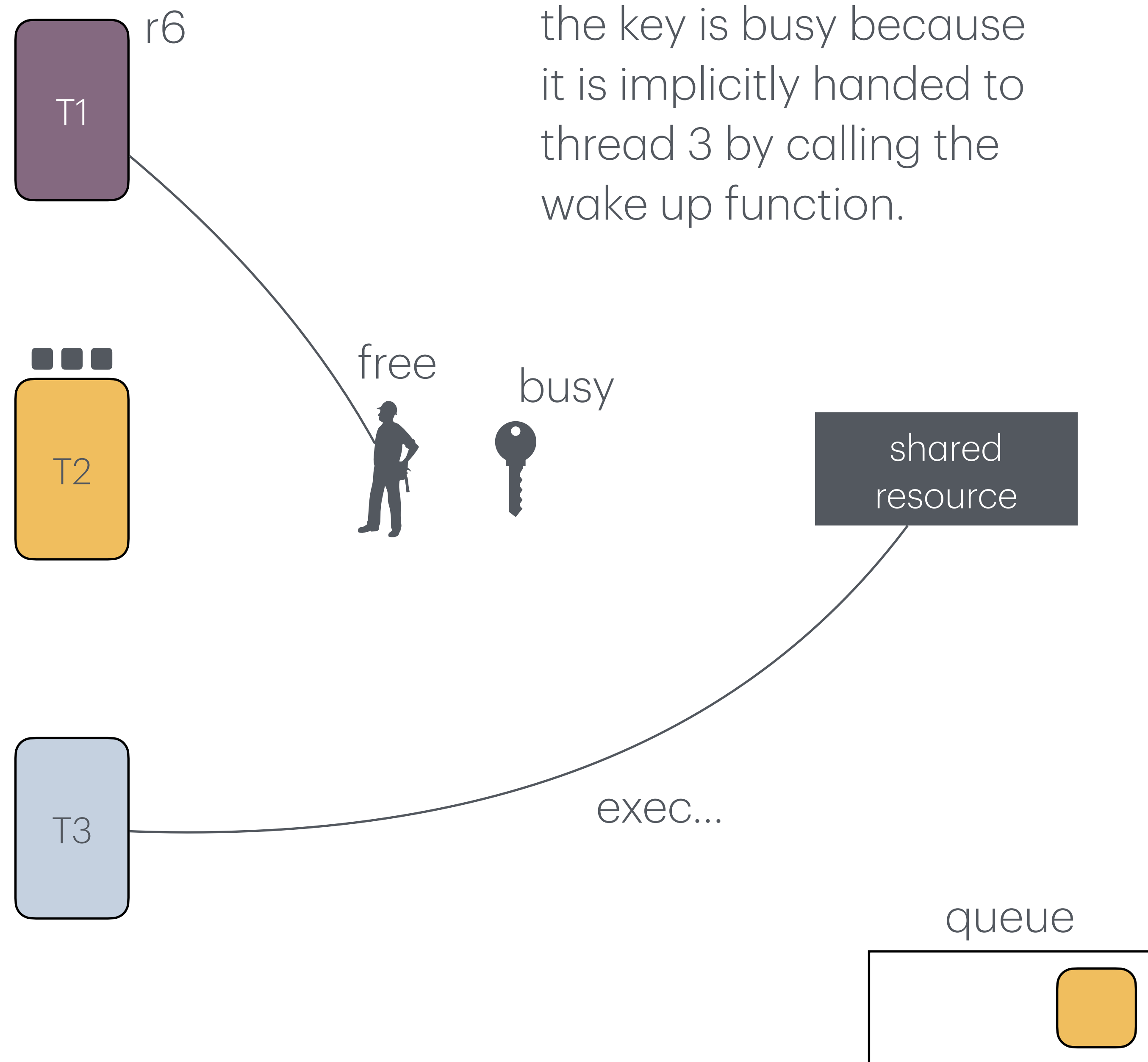
void acquire(int* lock) {
    while (test&set(guard));           // (1)
    if (*lock == BUSY) {               // (2)
        put_thread_on_wait_queue();    // (3)
        go_to_sleep();                 // (4)
        guard = FREE;                 // (5)
    } else {
        *lock = BUSY;                  // (6)
        guard = FREE;                  // (7)
    }
}

```

```

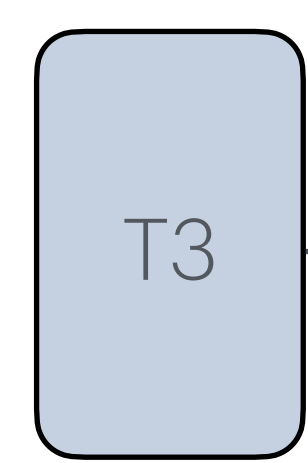
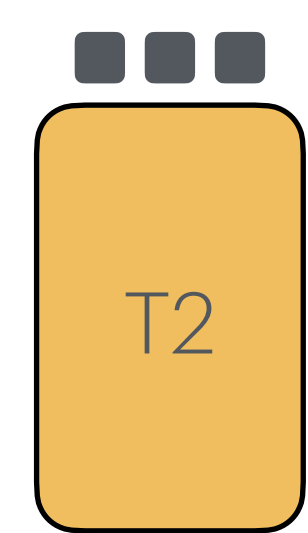
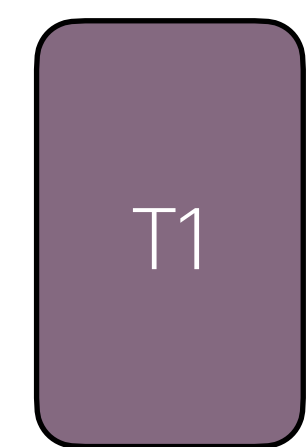
void release(int* lock) {
    while (test&set(guard));           // (1)
    if (any_thread_on_wait_queue()) {  // (2)
        take_thread_off_wait_queue();  // (3)
        place_thread_on_ready_queue(); // (4)
    } else {
        *lock = FREE;                  // (5)
    }
    guard = FREE;                      // (6)
}

```



```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {               // (2)  
        put_thread_on_wait_queue();   // (3)  
        go_to_sleep();                // (4)  
        guard = FREE;                // (5)  
    } else {                           // (6)  
        *lock = BUSY;                 // (7)  
        guard = FREE;                 // (7)  
    }  
}
```

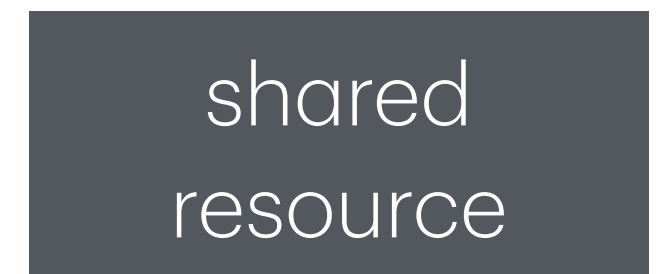
```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {  // (2)  
        take_thread_off_wait_queue(); // (3)  
        place_thread_on_ready_queue(); // (4)  
    } else {                           // (5)  
        *lock = FREE;                 // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



free

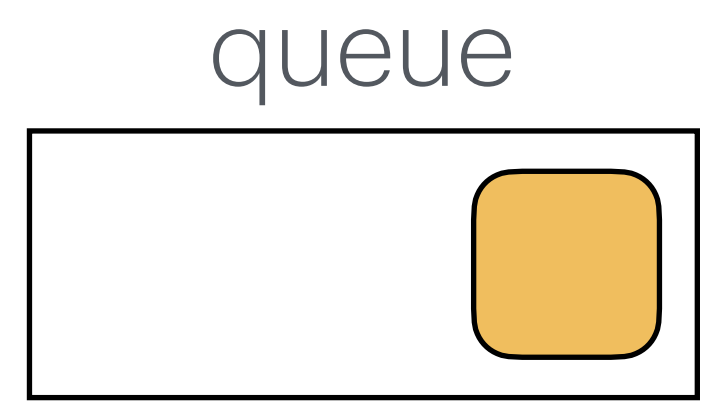


busy



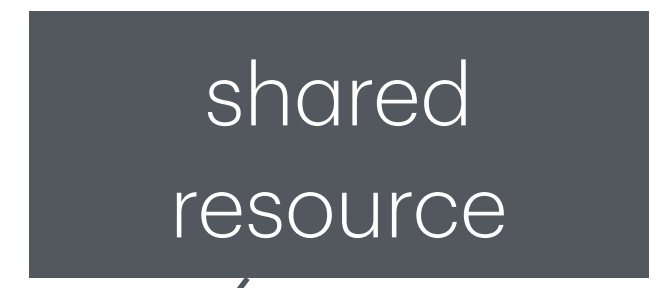
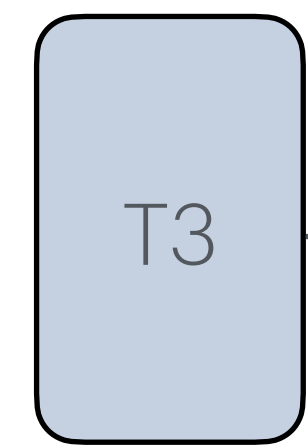
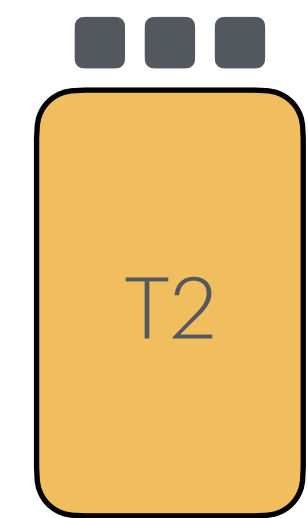
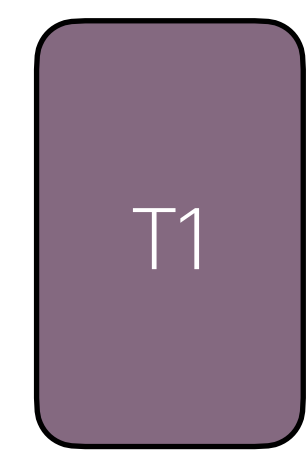
shared
resource

exec...

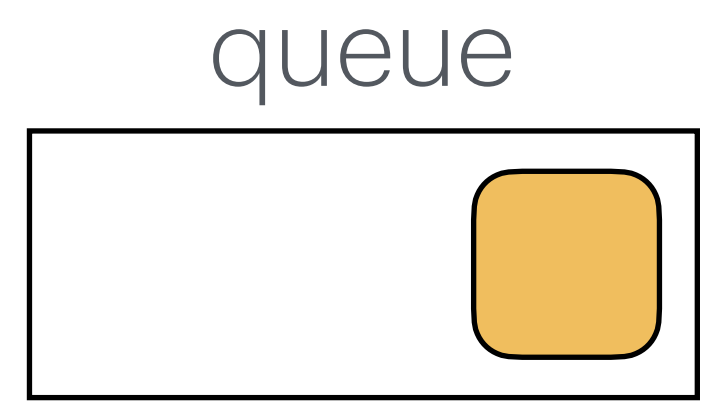


```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                  // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {   // (2)  
        take_thread_off_wait_queue();  // (3)  
        place_thread_on_ready_queue(); // (4)  
    } else {  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```

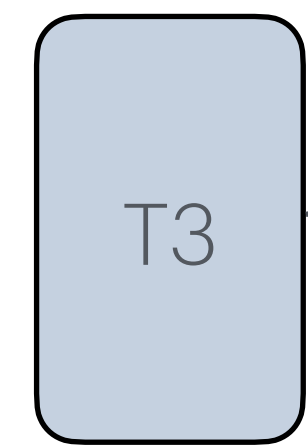
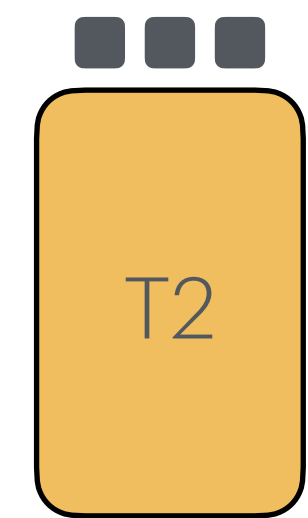
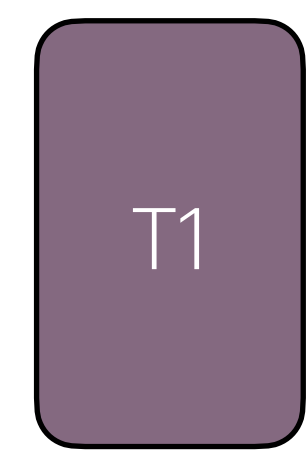


imma done



```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                 // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

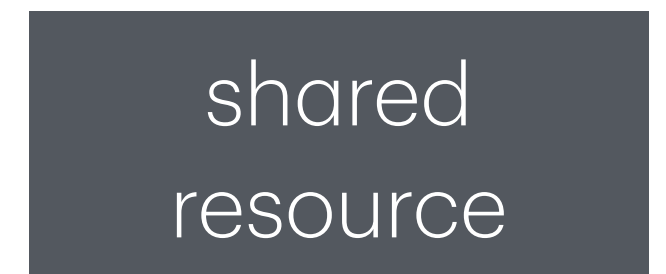
```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {   // (2)  
        take_thread_off_wait_queue();   // (3)  
        place_thread_on_ready_queue();  // (4)  
    } else {  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



r1

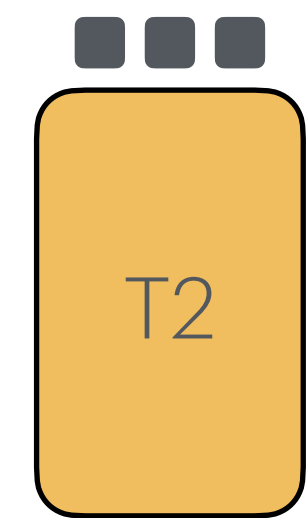
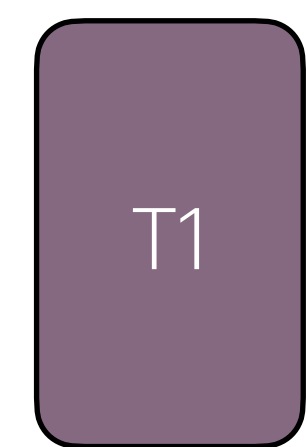


busy

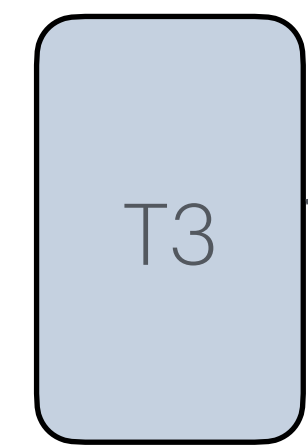


```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                 // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {  // (2)  
        take_thread_off_wait_queue();  // (3)  
        place_thread_on_ready_queue(); // (4)  
    } else {  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



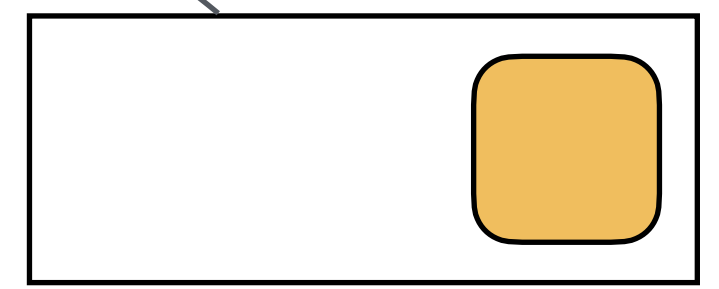
shared
resource



r2

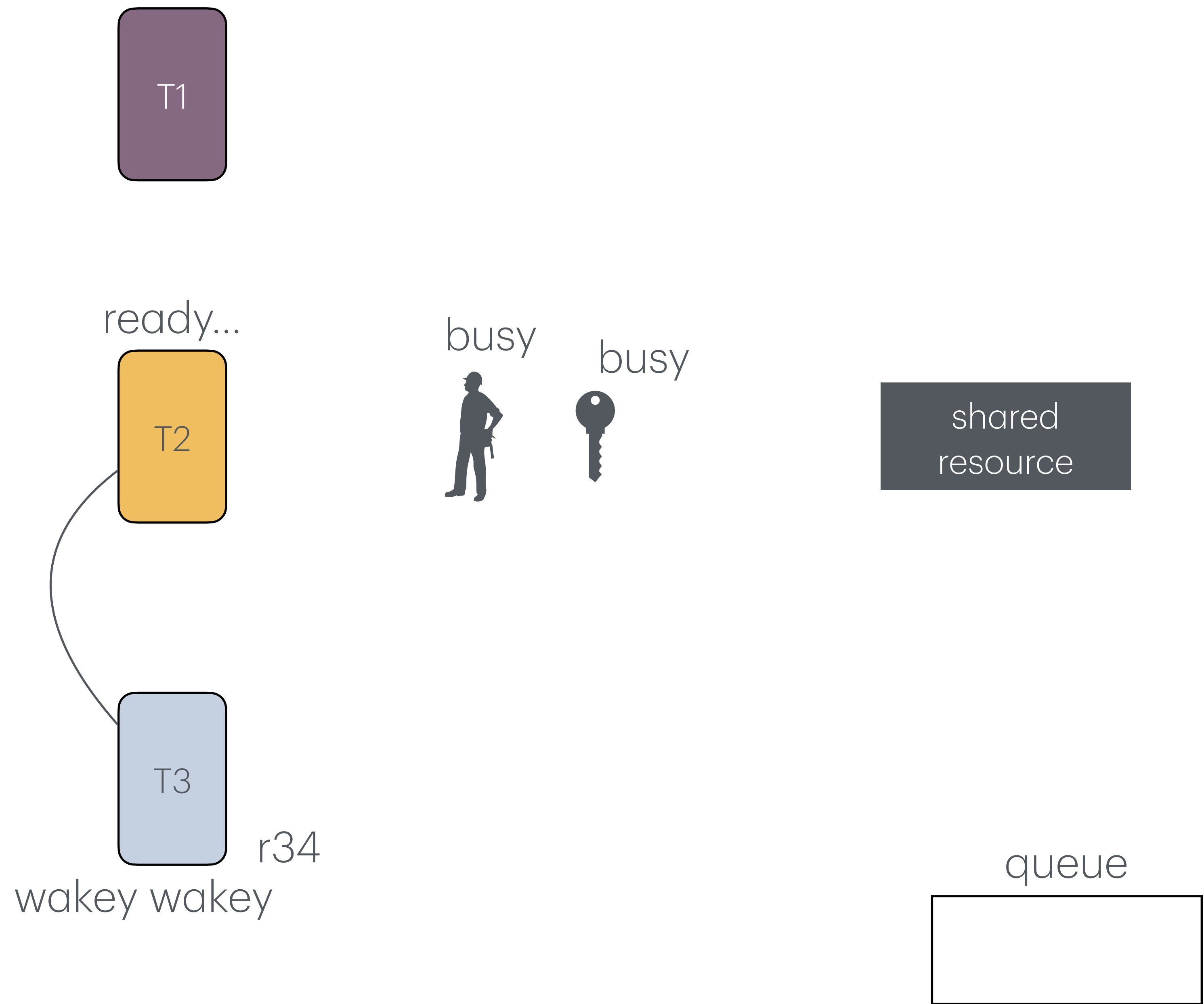
is any1 here?

queue



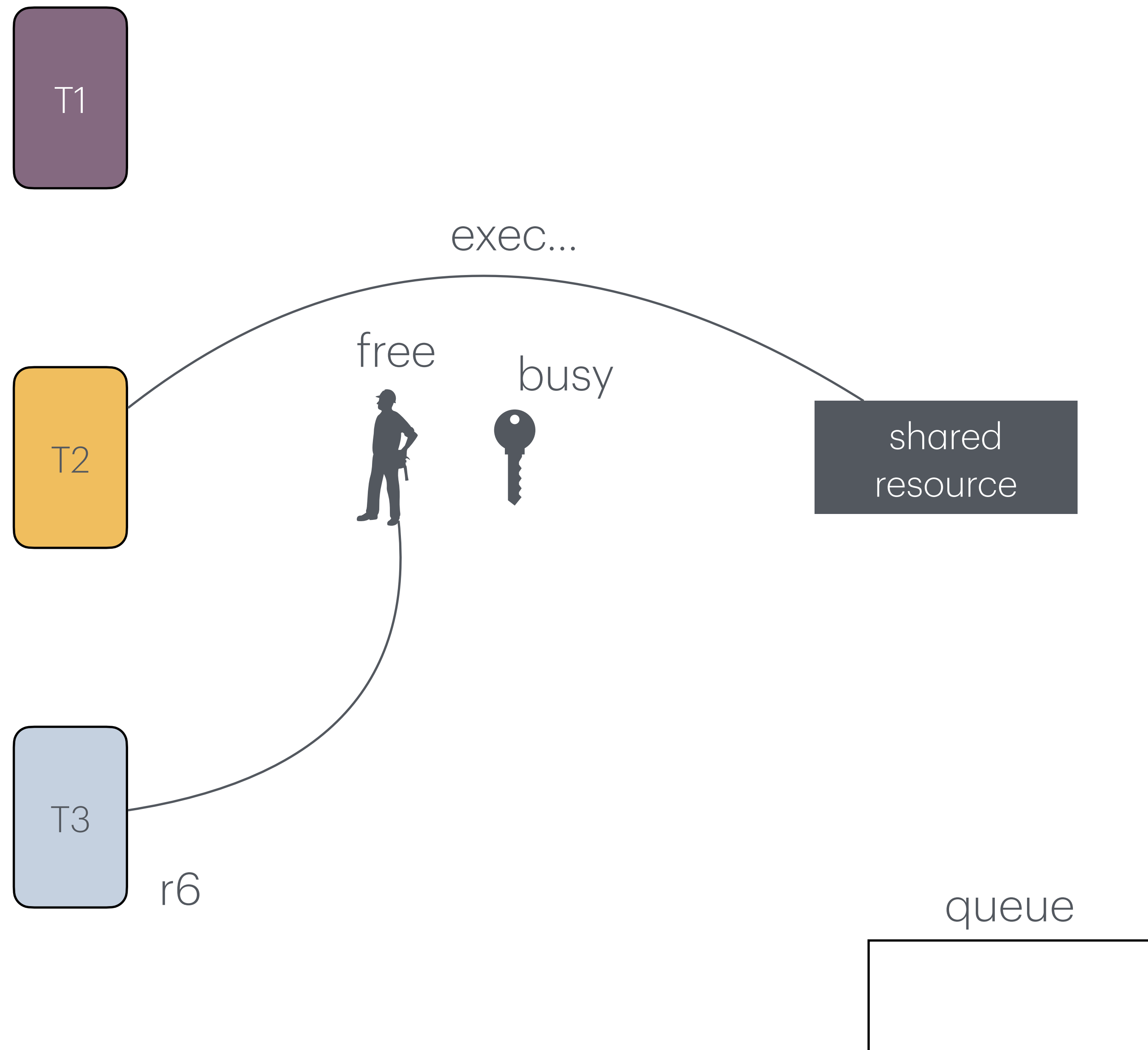

```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                 // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {  // (2)  
        take_thread_off_wait_queue(); // (3)  
        place_thread_on_ready_queue(); // (4)  
    } else {  
        *lock = FREE;                 // (5)  
    }  
    guard = FREE;                     // (6)  
}
```




```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {               // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                 // (5)  
    } else {  
        *lock = BUSY;                 // (6)  
        guard = FREE;                 // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {  // (2)  
        take_thread_off_wait_queue();  // (3)  
        place_thread_on_ready_queue(); // (4)  
    } else {  
        *lock = FREE;                 // (5)  
    }  
    guard = FREE;                     // (6)  
}
```



```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {               // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                 // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {   // (2)  
        take_thread_off_wait_queue();   // (3)  
        place_thread_on_ready_queue();  // (4)  
    } else {  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



exec...

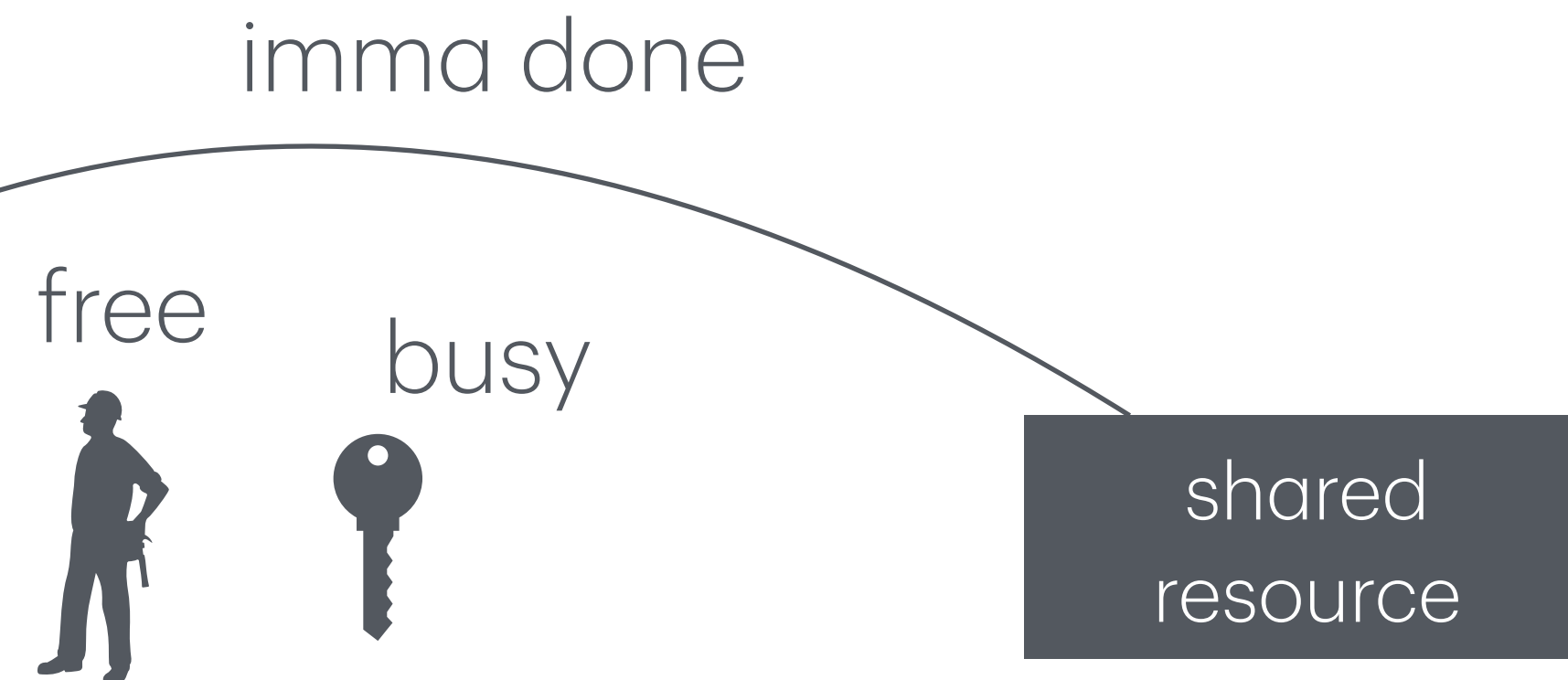
shared
resource

queue



```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {               // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                 // (5)  
    } else {  
        *lock = BUSY;                 // (6)  
        guard = FREE;                 // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {   // (2)  
        take_thread_off_wait_queue();   // (3)  
        place_thread_on_ready_queue();  // (4)  
    } else {  
        *lock = FREE;                 // (5)  
    }  
    guard = FREE;                     // (6)  
}
```

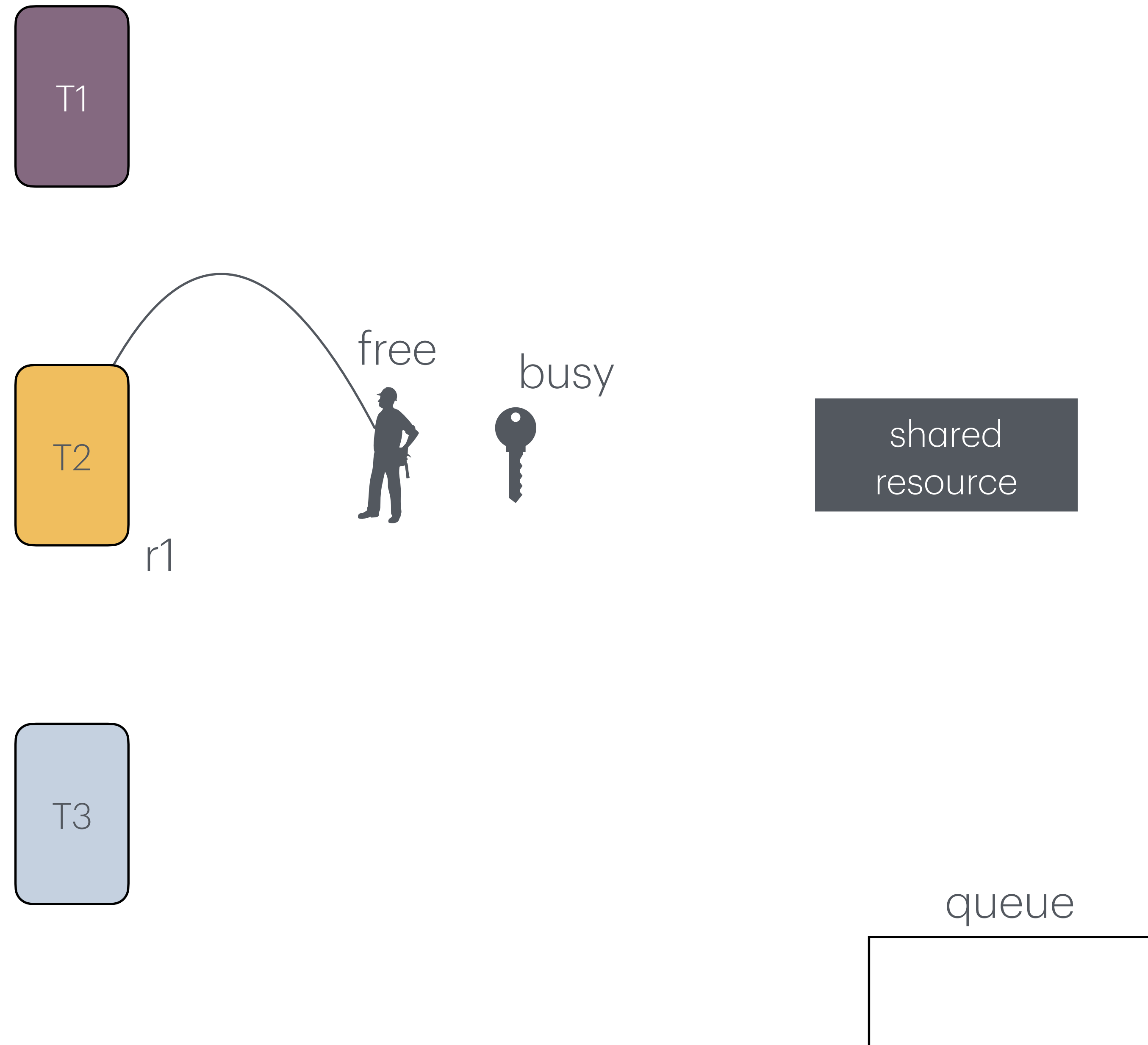


queue



```
void acquire(int* lock) {
    while (test&set(guard));           // (1)
    if (*lock == BUSY) {               // (2)
        put_thread_on_wait_queue();    // (3)
        go_to_sleep();                 // (4)
        guard = FREE;                  // (5)
    } else {
        *lock = BUSY;                  // (6)
        guard = FREE;                  // (7)
    }
}
```

```
void release(int* lock) {
    while (test&set(guard));           // (1)
    if (any_thread_on_wait_queue()) {  // (2)
        take_thread_off_wait_queue();  // (3)
        place_thread_on_ready_queue(); // (4)
    } else {
        *lock = FREE;                  // (5)
    }
    guard = FREE;                      // (6)
}
```



```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                  // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {   // (2)  
        take_thread_off_wait_queue();   // (3)  
        place_thread_on_ready_queue();  // (4)  
    } else {  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



free



busy

shared
resource

r2

is any1 here?

queue

no


```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {               // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                 // (5)  
    } else {  
        *lock = BUSY;                 // (6)  
        guard = FREE;                 // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {  // (2)  
        take_thread_off_wait_queue();  // (3)  
        place_thread_on_ready_queue(); // (4)  
    } else {  
        *lock = FREE;                 // (5)  
    }  
    guard = FREE;                     // (6)  
}
```



r5

free



free

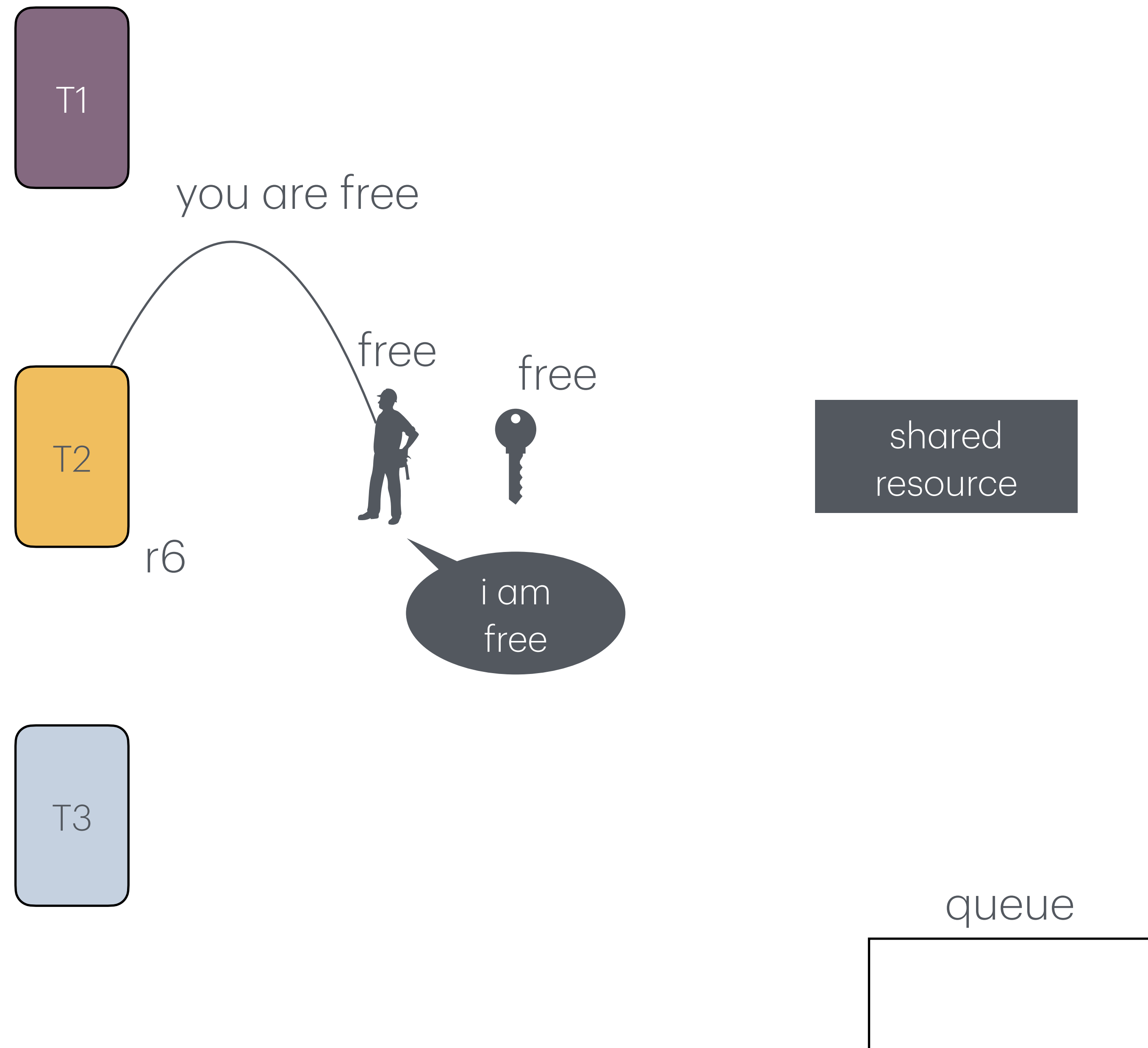
shared
resource

queue



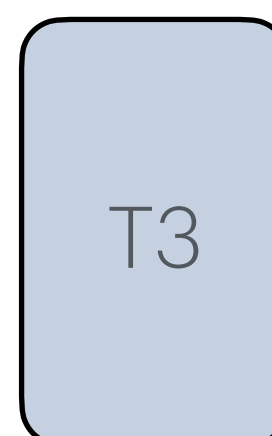
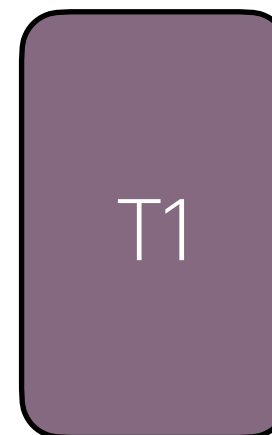

```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                 // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {   // (2)  
        take_thread_off_wait_queue();   // (3)  
        place_thread_on_ready_queue();  // (4)  
    } else {  
        *lock = FREE;                  // (5)  
    }  
    guard = FREE;                      // (6)  
}
```



```
void acquire(int* lock) {  
    while (test&set(guard));           // (1)  
    if (*lock == BUSY) {                // (2)  
        put_thread_on_wait_queue();    // (3)  
        go_to_sleep();                 // (4)  
        guard = FREE;                  // (5)  
    } else {  
        *lock = BUSY;                  // (6)  
        guard = FREE;                  // (7)  
    }  
}
```

```
void release(int* lock) {  
    while (test&set(guard));           // (1)  
    if (any_thread_on_wait_queue()) {   // (2)  
        take_thread_off_wait_queue();   // (3)  
        place_thread_on_ready_queue();  // (4)  
    } else {  
        *lock = FREE;                   // (5)  
    }  
    guard = FREE;                       // (6)  
}
```



free



free

shared
resource

queue

