



Grand Central Dispatch





Grand Central Dispatch

Why do we need it?

- Blocking syscalls
- Take advantage of multicore hardware



Read from disk

- Caller's thread is blocked on user-space
- Context is switched to continue code execution by kernel
- Kernel schedules next read from disk
- When the data is copied to RAM, execution is returned to the blocked thread
- If failed, error is returned



Why do we care again?

- Don't ever block main thread!!!



main thread is responsive for UI manipulations

DispatchQueue

- Can execute closures one by one

```
DispatchQueue.main.async {  
    // code to be executed on the main thread  
}
```

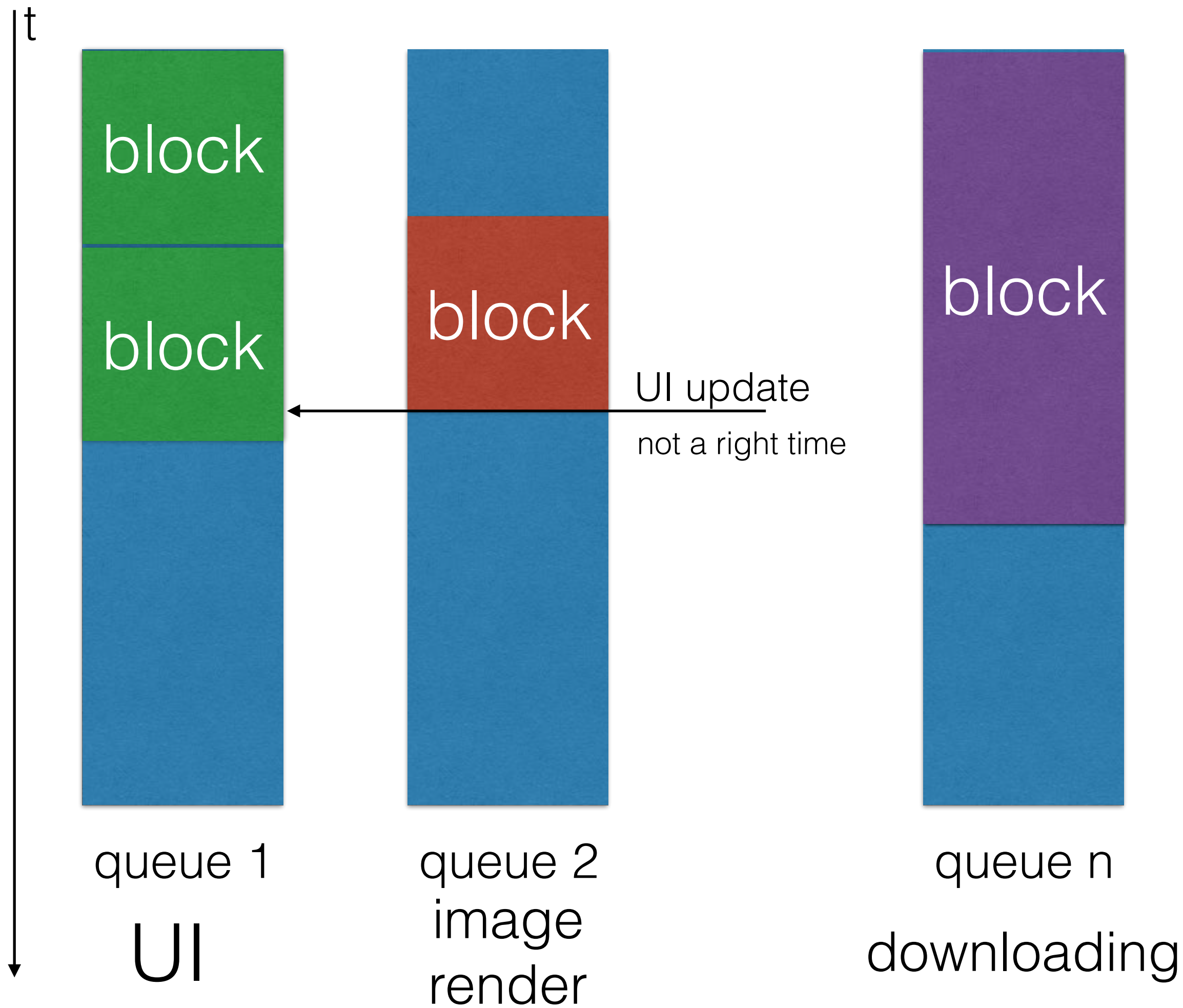
- Can be created

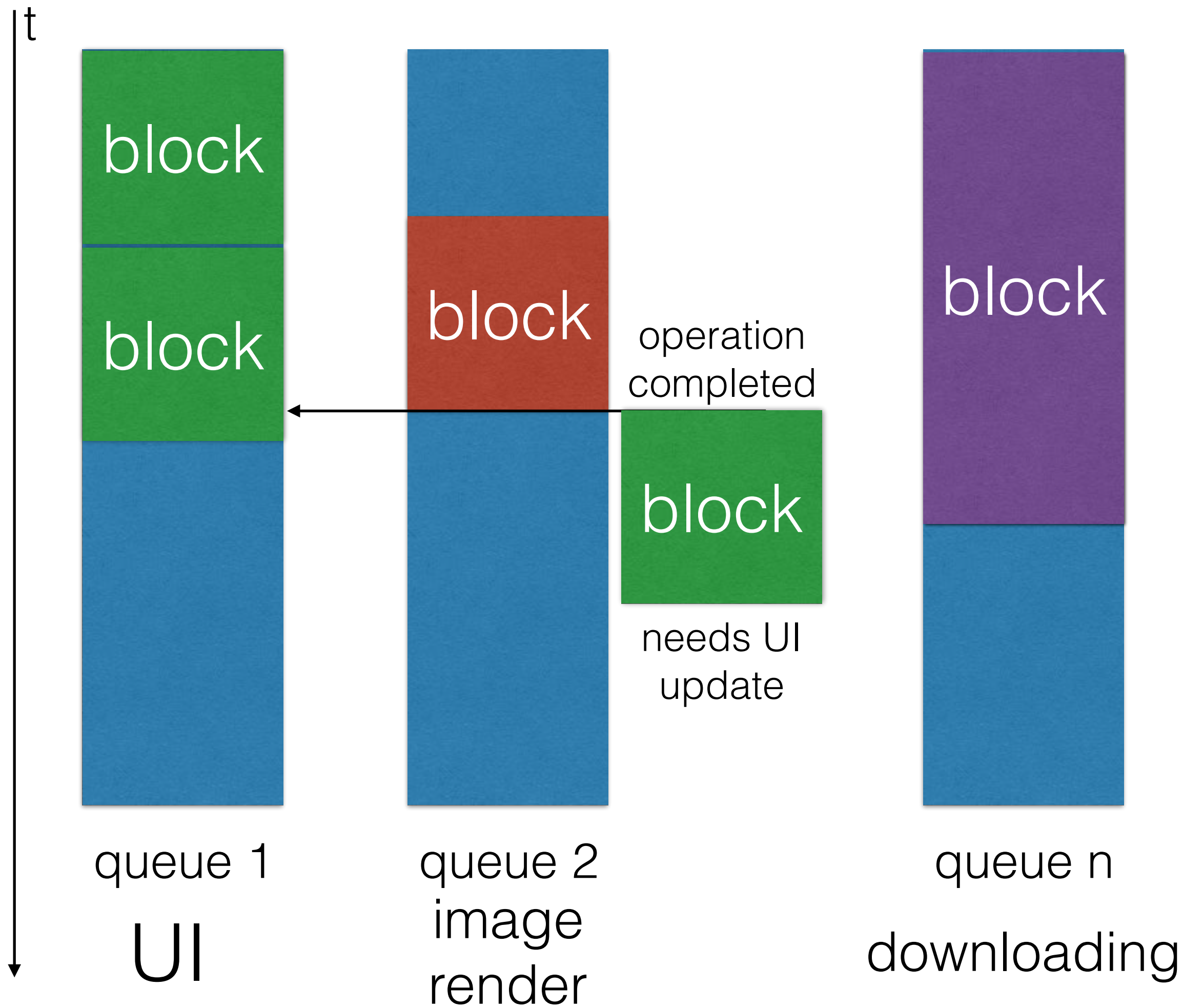
```
let queue = DispatchQueue(label: "My loading queue")
```

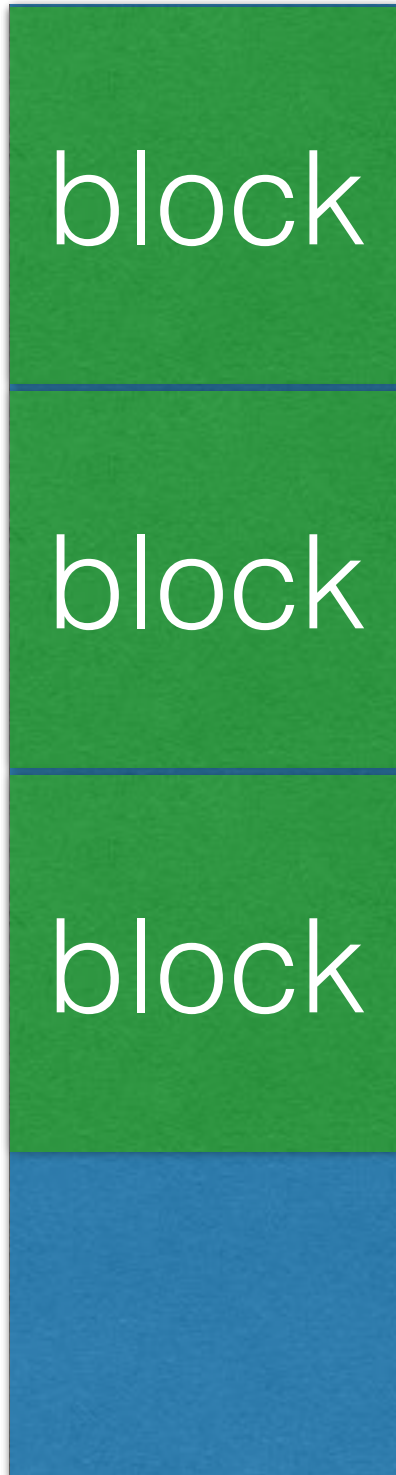
more after example

DispatchQueue by example

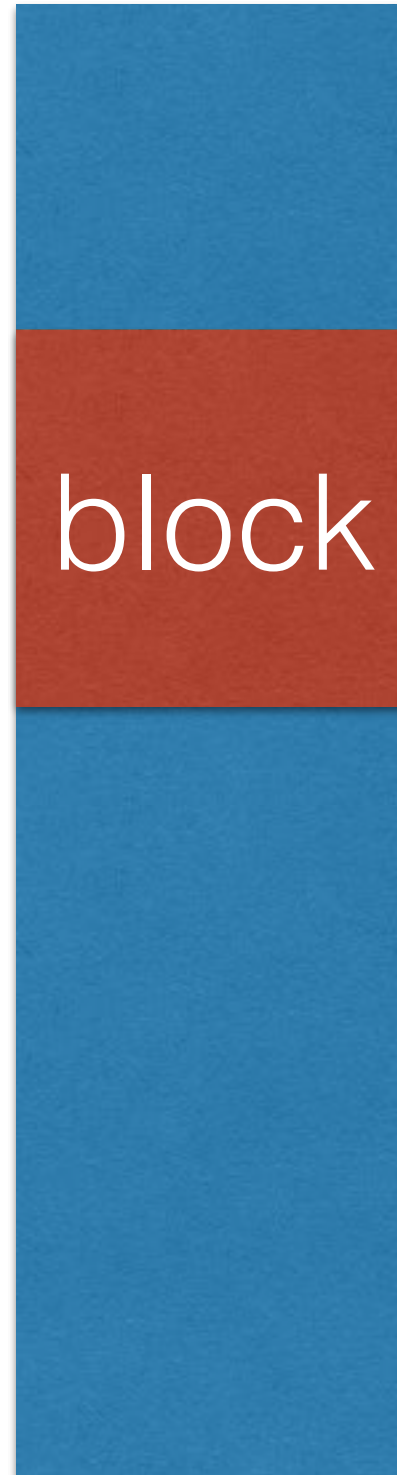
- Download image from internet
- Resize it
- Show inside UIImageView







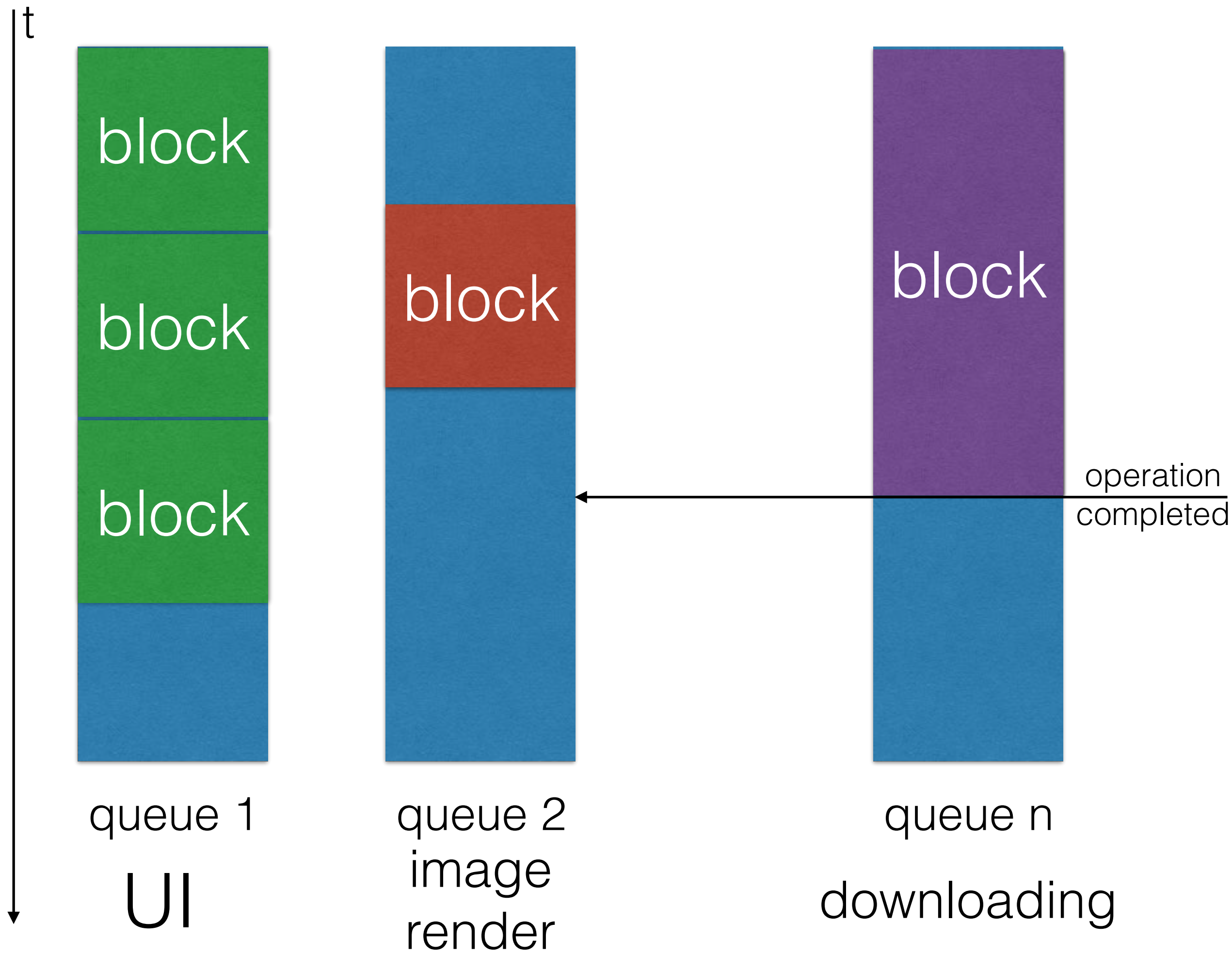
queue 1
UI

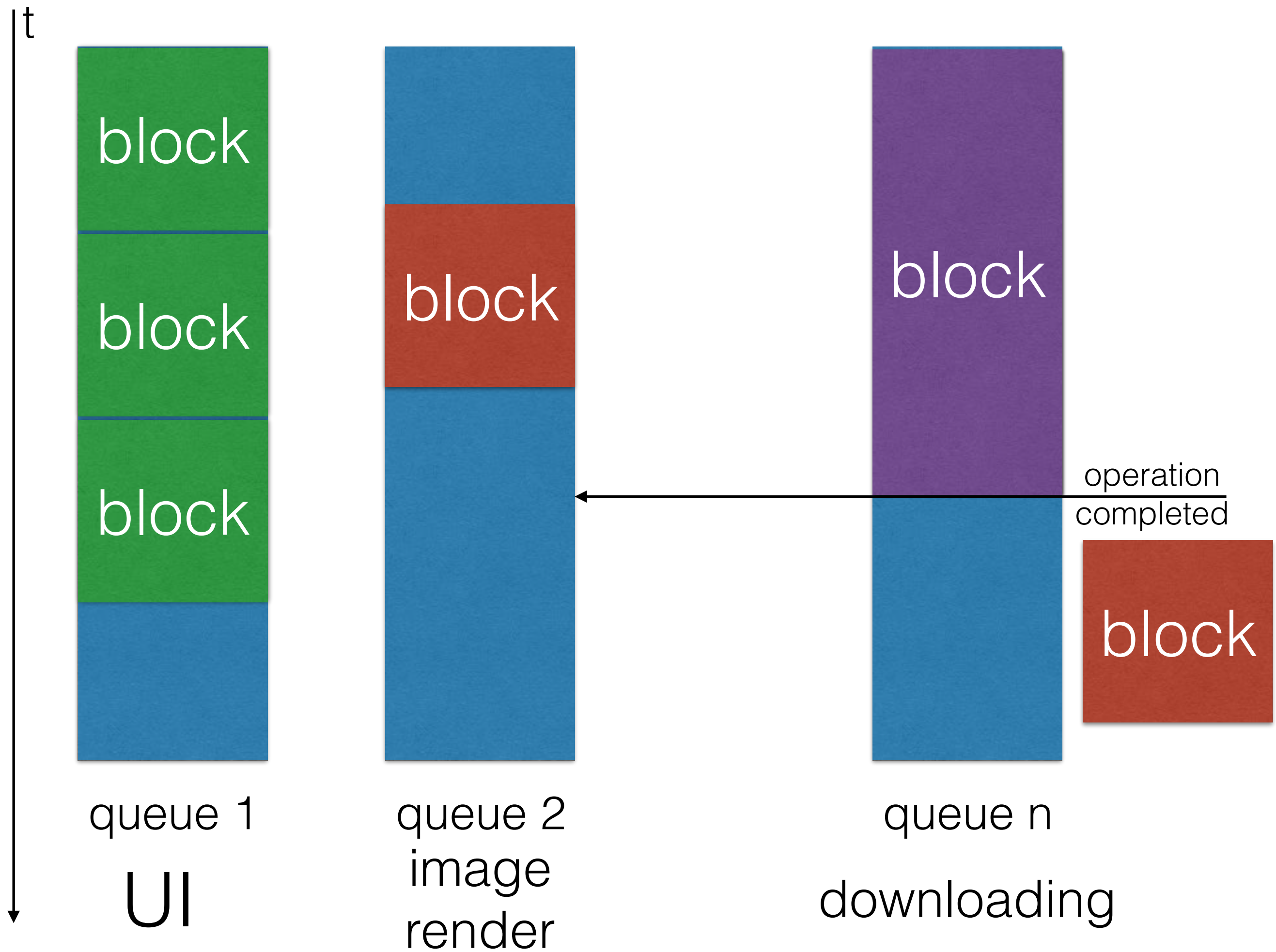


queue 2
image
render

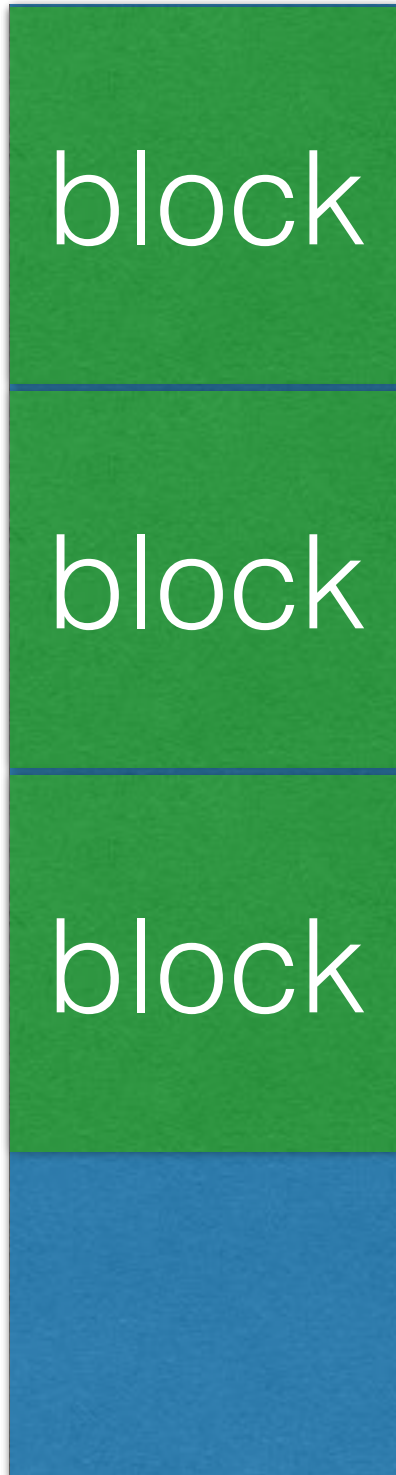


queue n
downloading

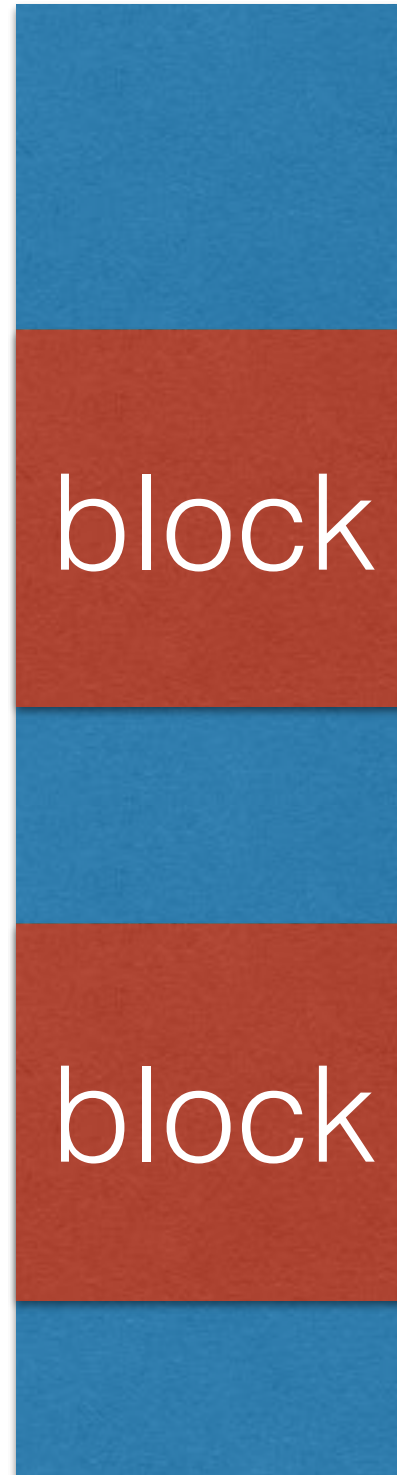




t



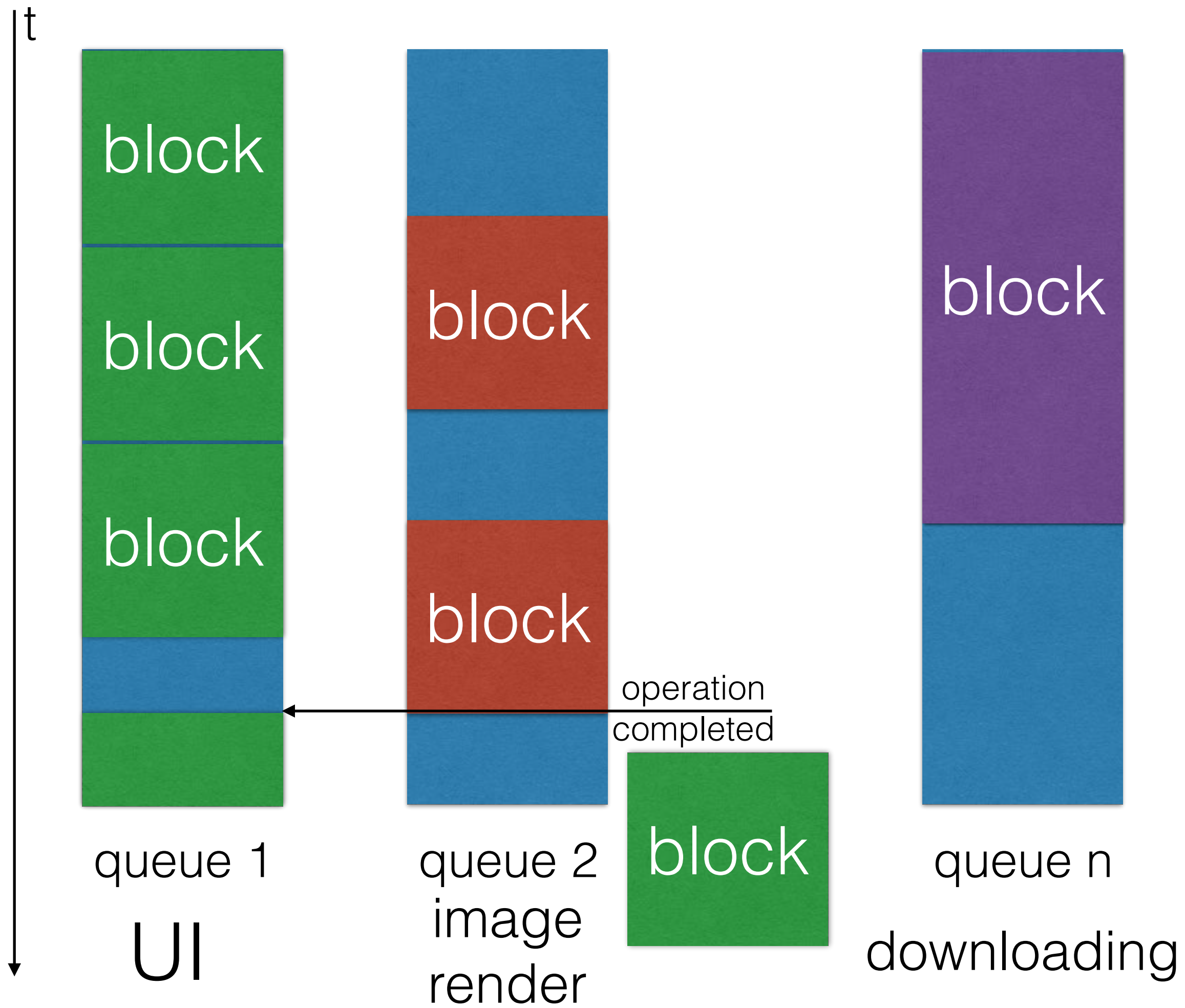
queue 1
UI



queue 2
image
render



queue n
downloading



Patterns

- Don't block main queue
- Create queue for each type of work(downloading, resizing)
- Be accurate when modifying different properties from different queue(synchronization)

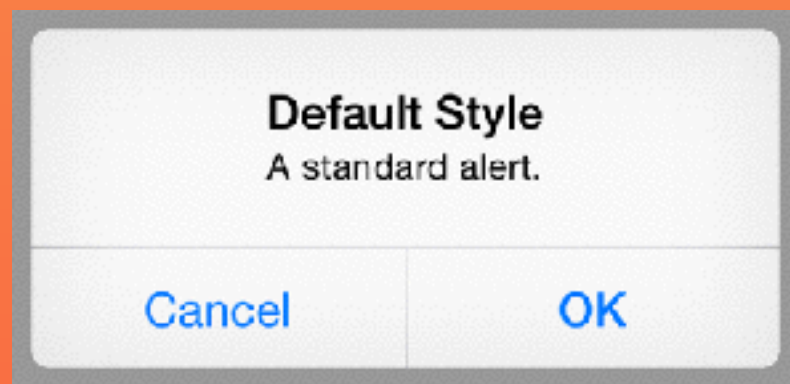
demo

ErrorHandling

- special Error protocol
- you can mark functions that can execute with errors by
throws

```
do {  
    try FileManager.default.removeItem(at: url)  
} catch let error {  
    print("failed with error \(error)")  
}
```

UIAlertController



Alert



Action sheet

```
let alert = UIAlertController(title: "Title", message: "Message", preferredStyle: .alert)
let action = UIAlertAction(title: "OK", style: .default) { action in
    // this will be called when user presses action
}
alert.addAction(action)
```

More about queue

- qos(Quality of service):

background, utility, default, userInitiated, userInteractive

cpu time →

- You can global queue with specified qos:

```
DispatchQueue.global(qos: .background)
```

Operation & OperationQueue

Object Oriented API for GCD

Operation

- Piece of work
- has status isCancelled, isExecuting, isFinished
- dependencies

reduces number of macaronies :)

OperationQueue

- convenient OOP wrapper
- can add/cancel operations, `cancelAllOperations`