Performance

iOS 7.0 capable iPhones

	iPhone 4	iPhone 4S	iPhone 5/5C	iPhone 5S
CPU	< 1 GHz	800 MHz	1.3 GHz	1.3 GHz (64-bit)
RAM	512 MB	512 MB	1 GB RAM	1 GB RAM
Cores	1	2	2	2
Battery	1,420 mAh	1,430 mAh	~1510 mAh	~1570 mAh

iOS 7.0 capable iPads

	iPad 2	iPad 3rd gen	iPad 4th gen	iPad Air
CPU	800 MHz	1.0 GHz	1.4 GHz	1.4 GHz (64-bit)
RAM	512 MB	1 GB	1 GB	1 GB
Cores	2	2	2	2
Battery	6,944 mAh	11,560 mAh	11,560 mAh	8,827 mAh

Performance

- Responsiveness
- Memory
- Battery
- Disk usage
- Bandwidth usage

Optimization

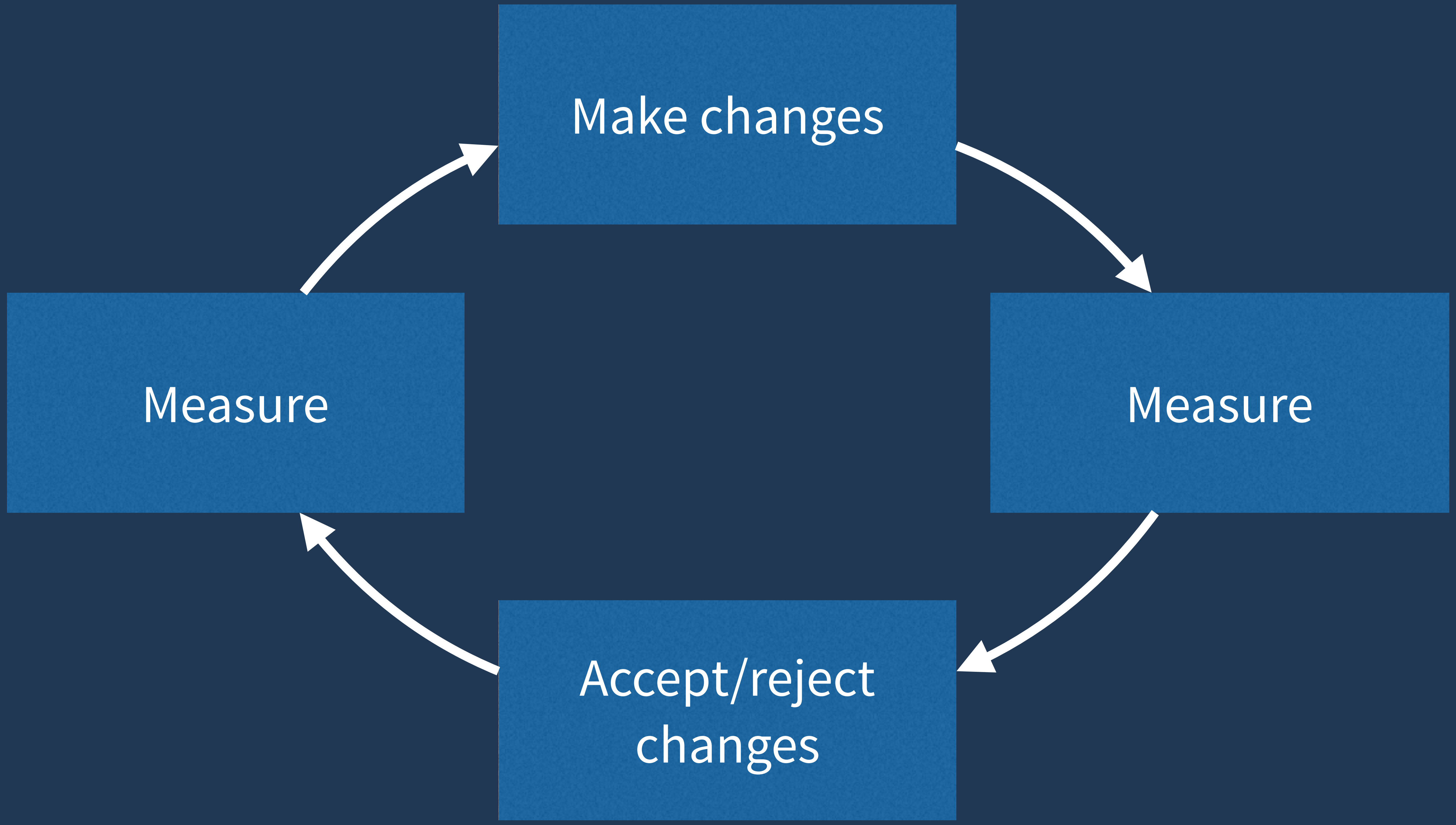
Premature Optimization

60 FPS

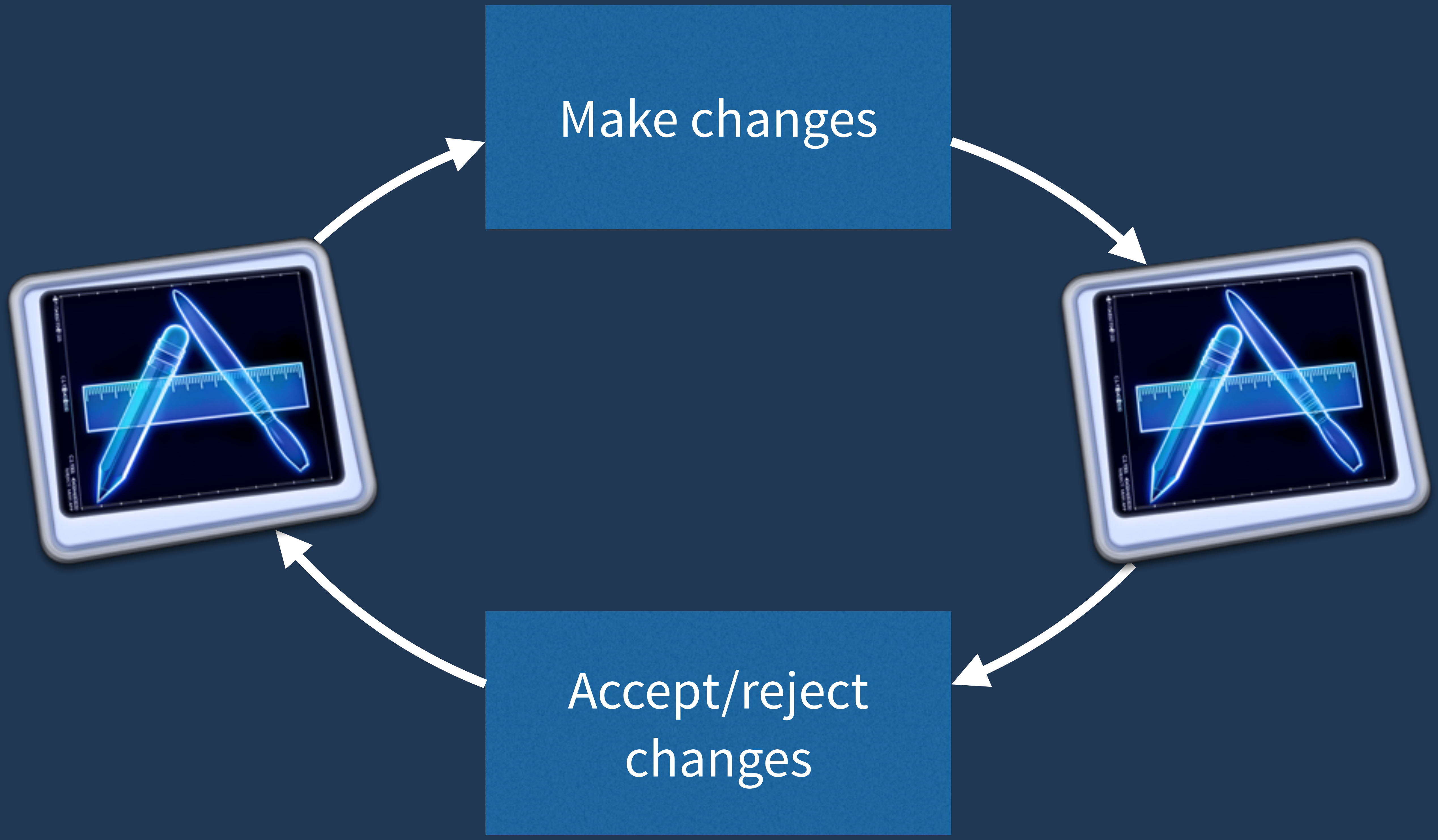
One frame every 16.6 ms, *at bare minimum,*
including everything else happening on the main thread.

Keep expensive operations off the main thread

Optimization



Instruments



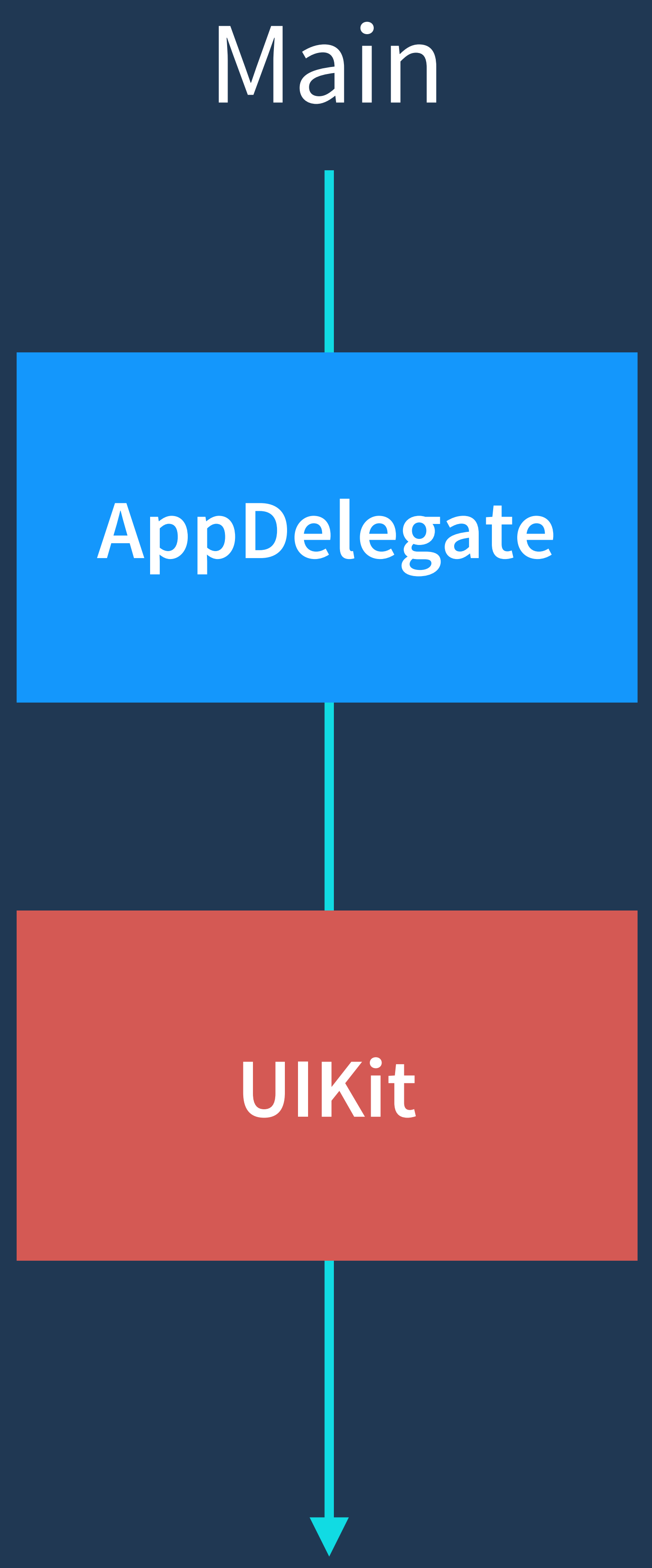
Instruments

- Device or Simulator
- Deferred mode
- Templates
- Custom instruments
- Zombie detection!

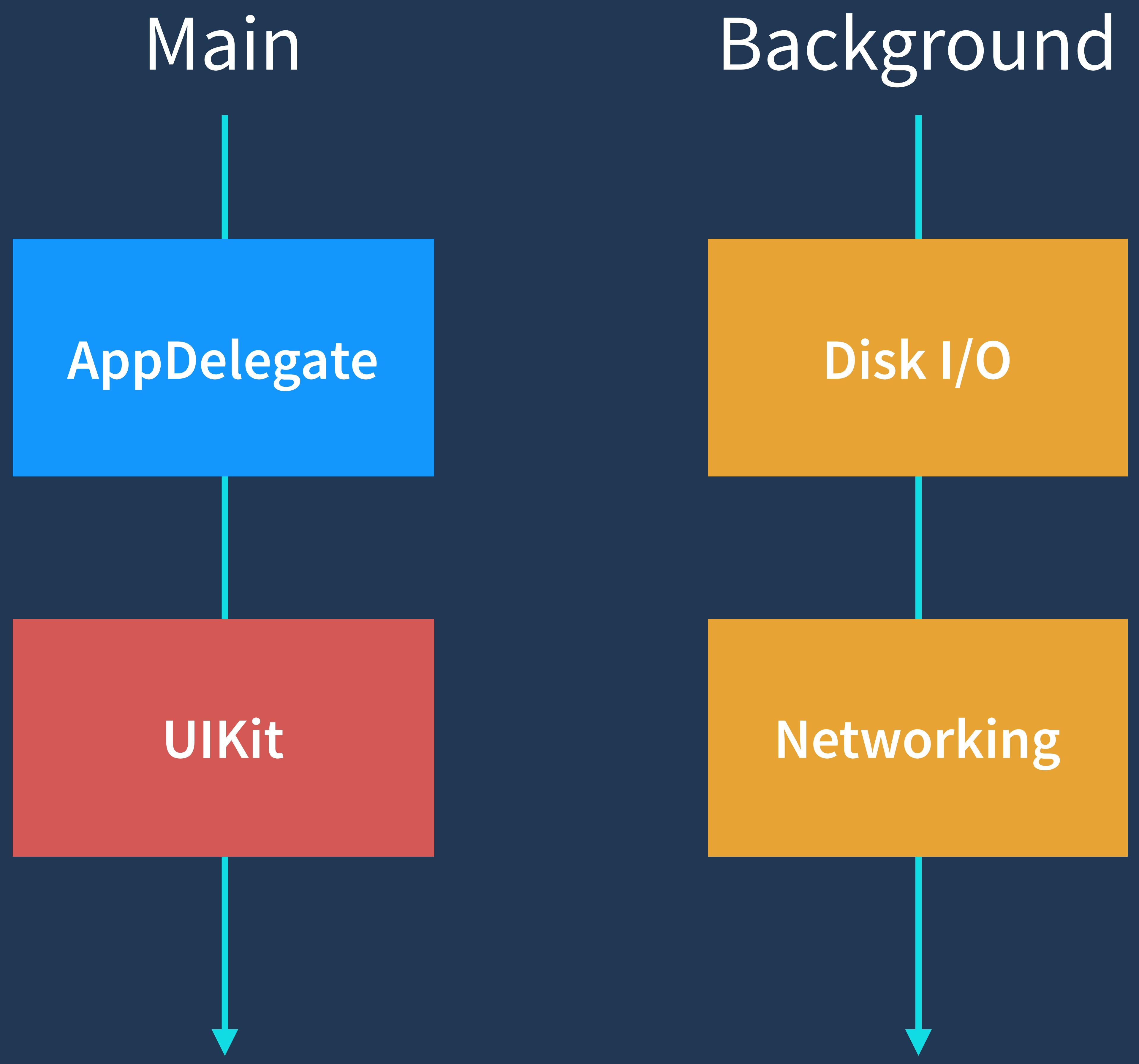
Primes Demo

Concurrency

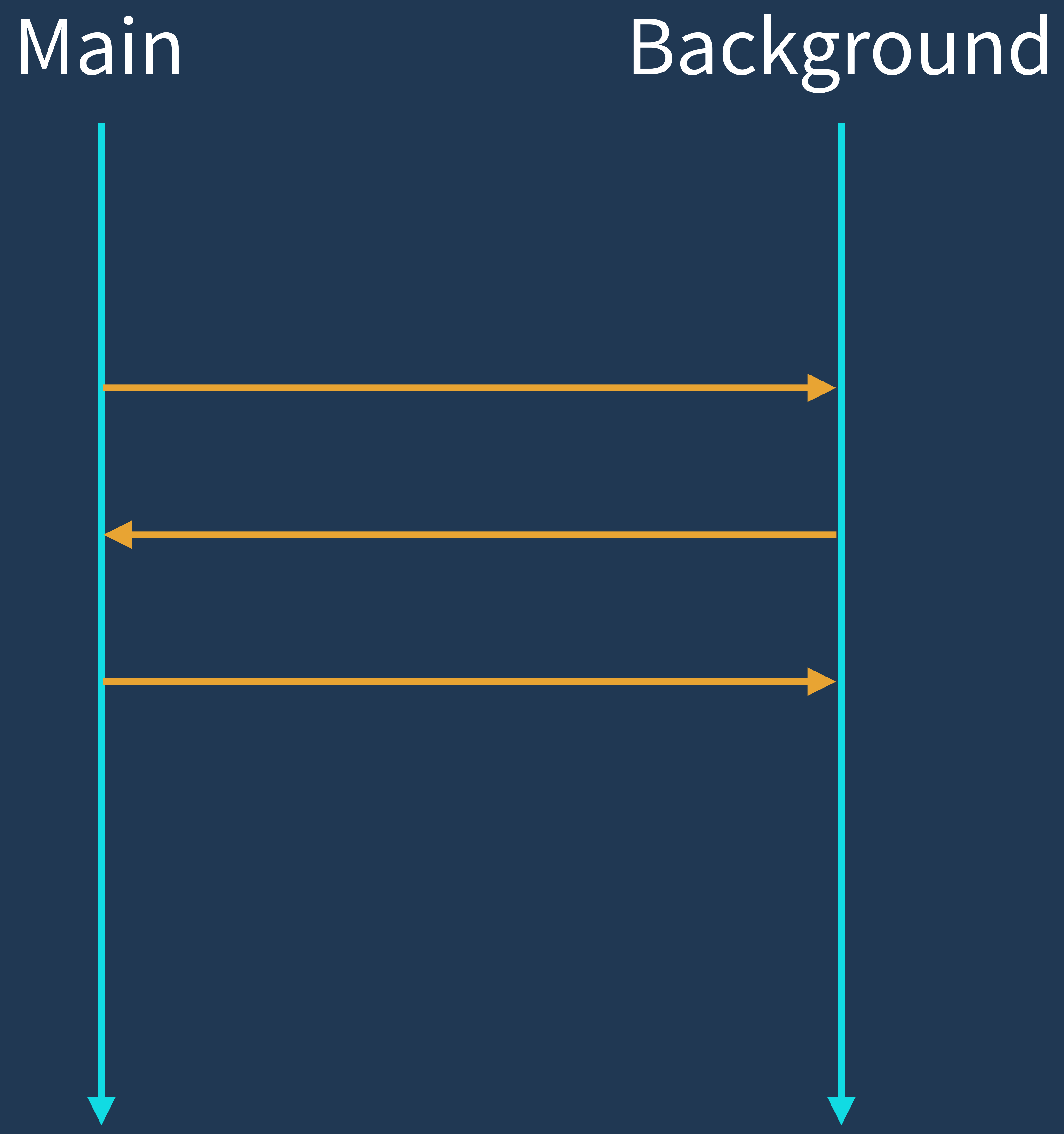
Threads



Threads



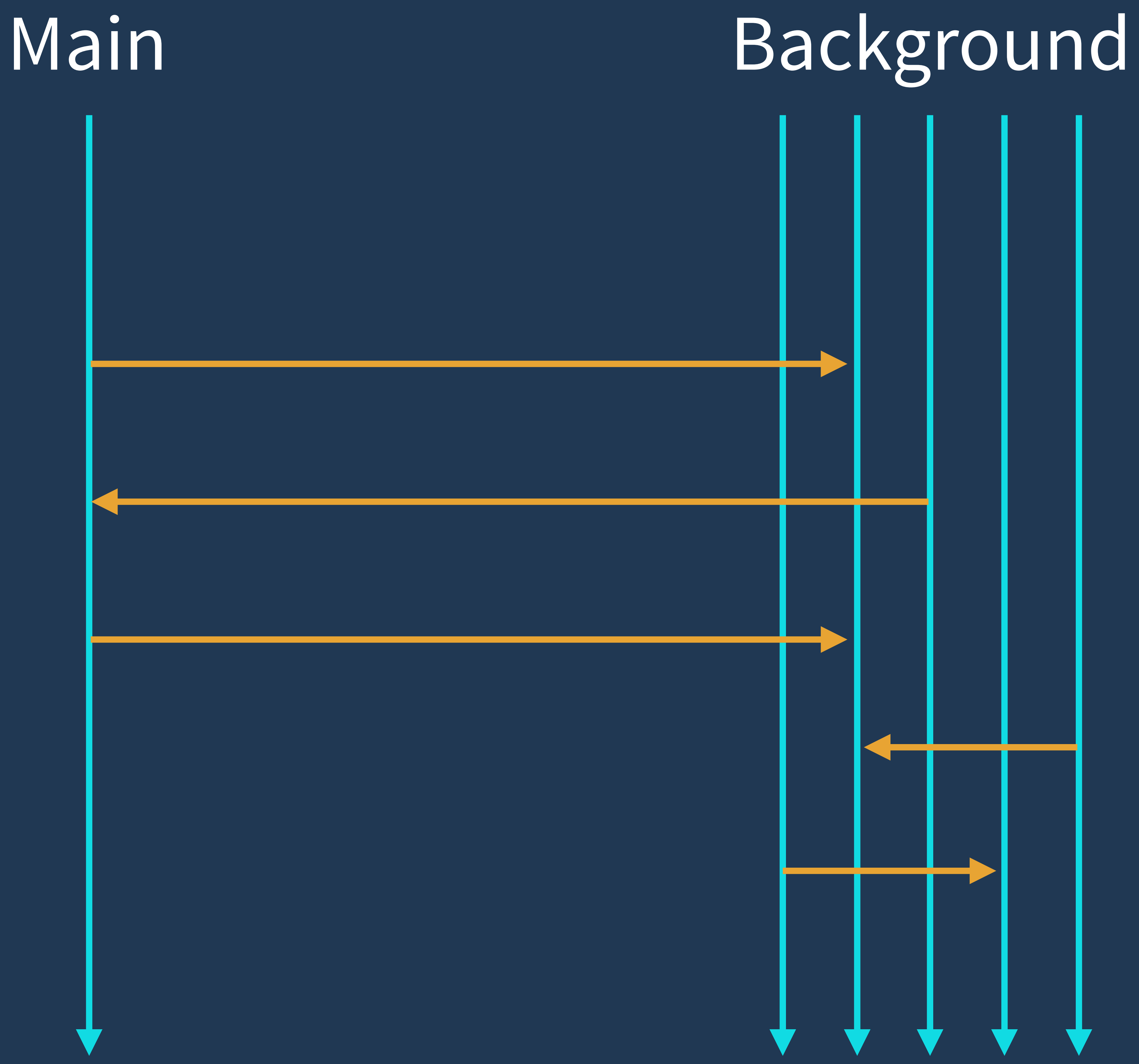
Threads



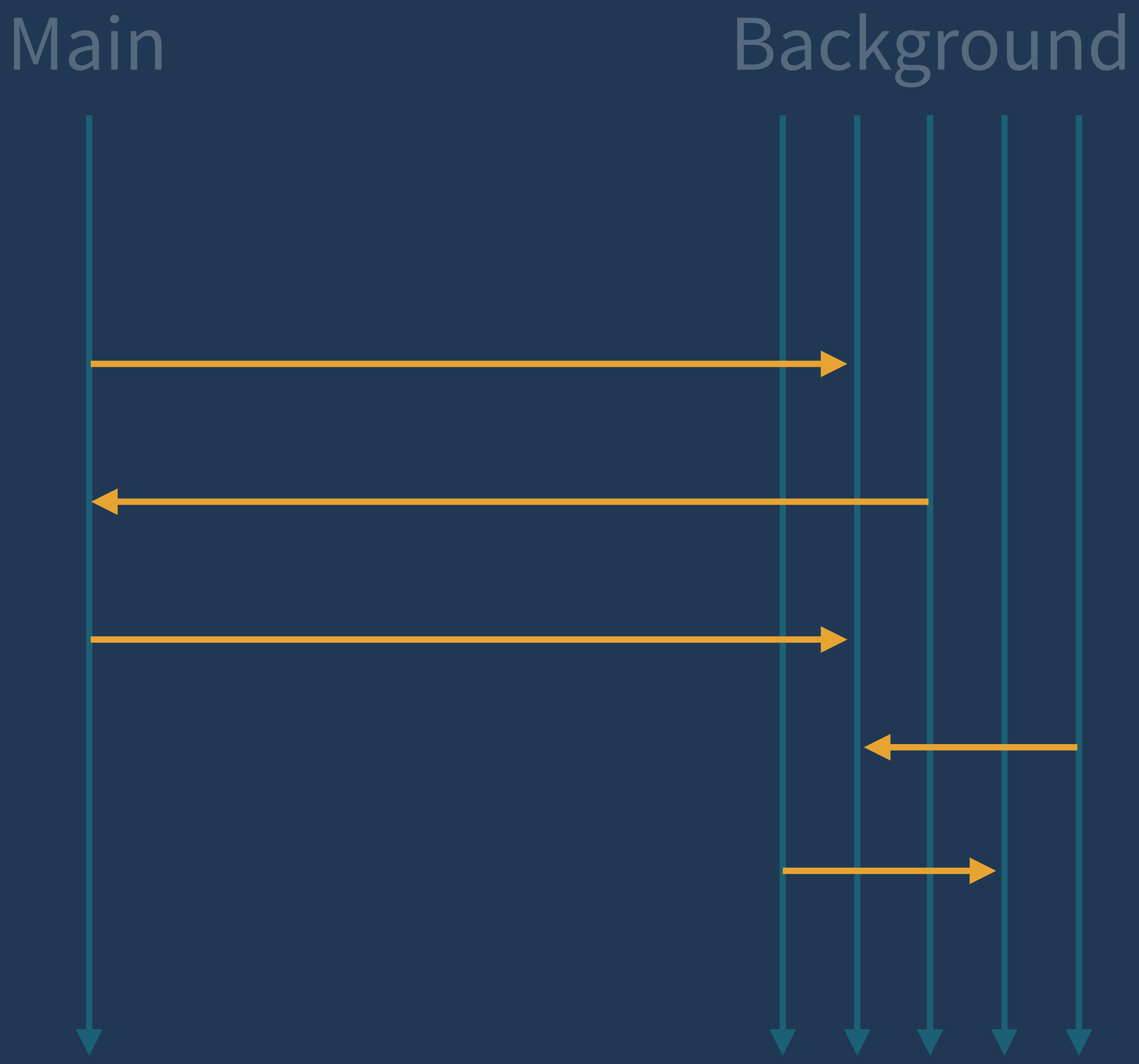
NSThread

```
[NSThread detachNewThreadSelector:@selector(myThreadMainMethod:)
                   toTarget:self withObject:nil];
```

Threads



Threads



Threads

- @synchronized
- NSLock
- Low-level (pthread, OSSpinLock)

Threads suck

Migrating away from threads

- Simplifies concurrent code
- Prevents locking, improves performance
- Platform independent regardless of CPUs

Migrating away from threads

- Use asynchronous callbacks
- Protect shared memory with single threaded queues
- Avoid locking

Migrating away from threads

Grand Central Dispatch

Operation Queues

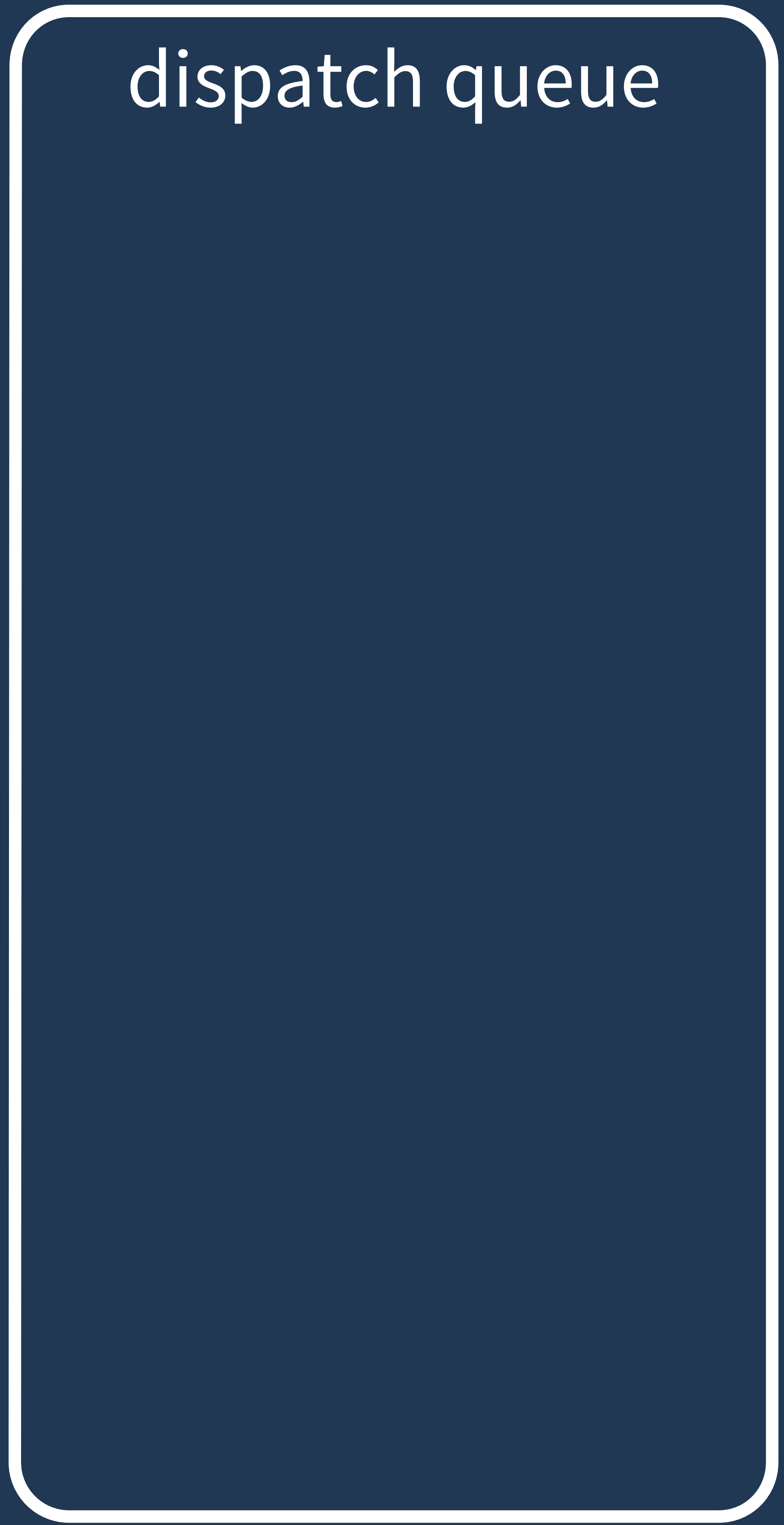
Grand Central Dispatch

Grand Central Dispatch

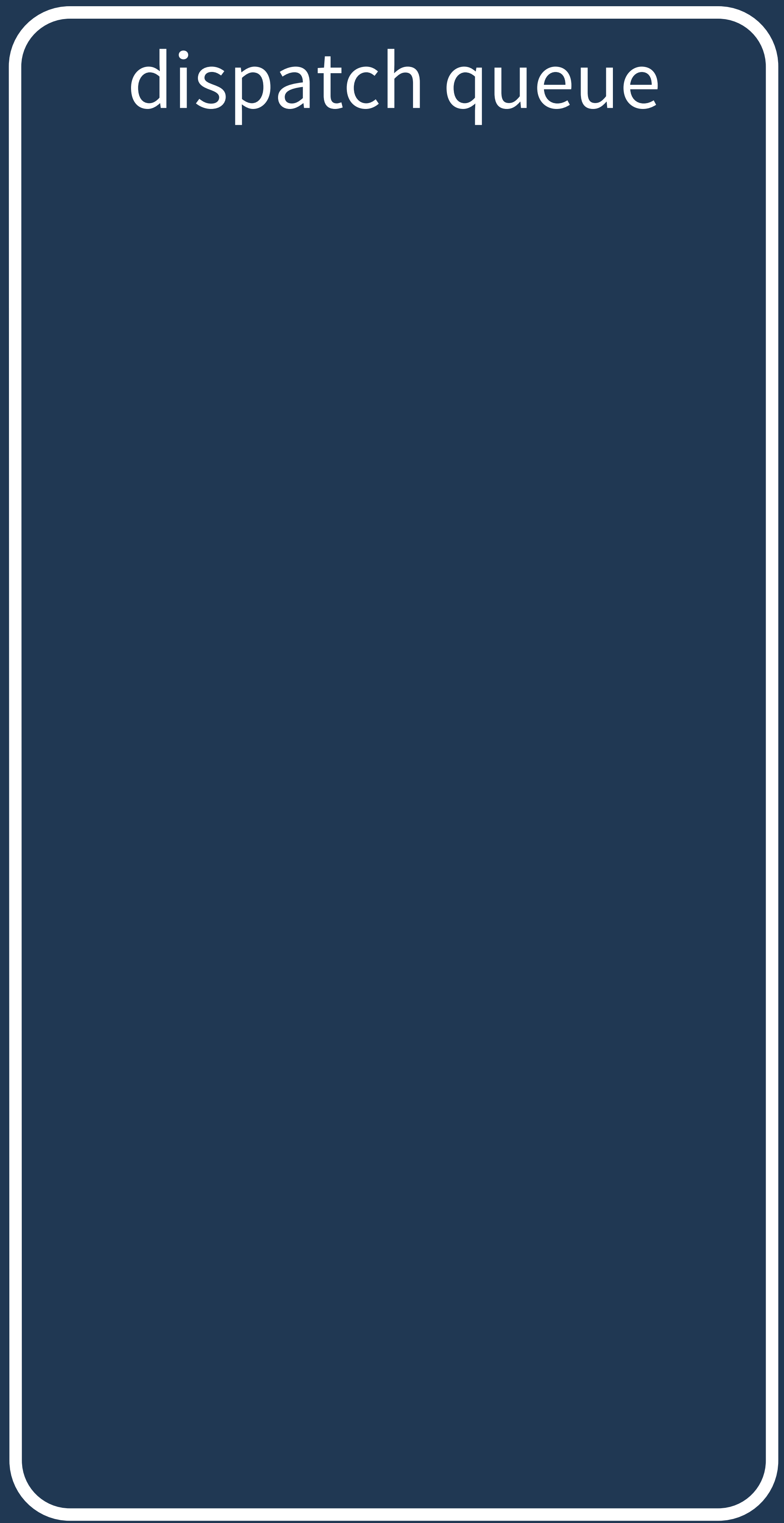
- Simple, but powerful
- C-based API
- Blocks
- One-off fire and forget tasks
- Composing your own concurrency system

Dispatch Queues

- A pointer, not an object
- Serial or concurrent
- Accepts blocks

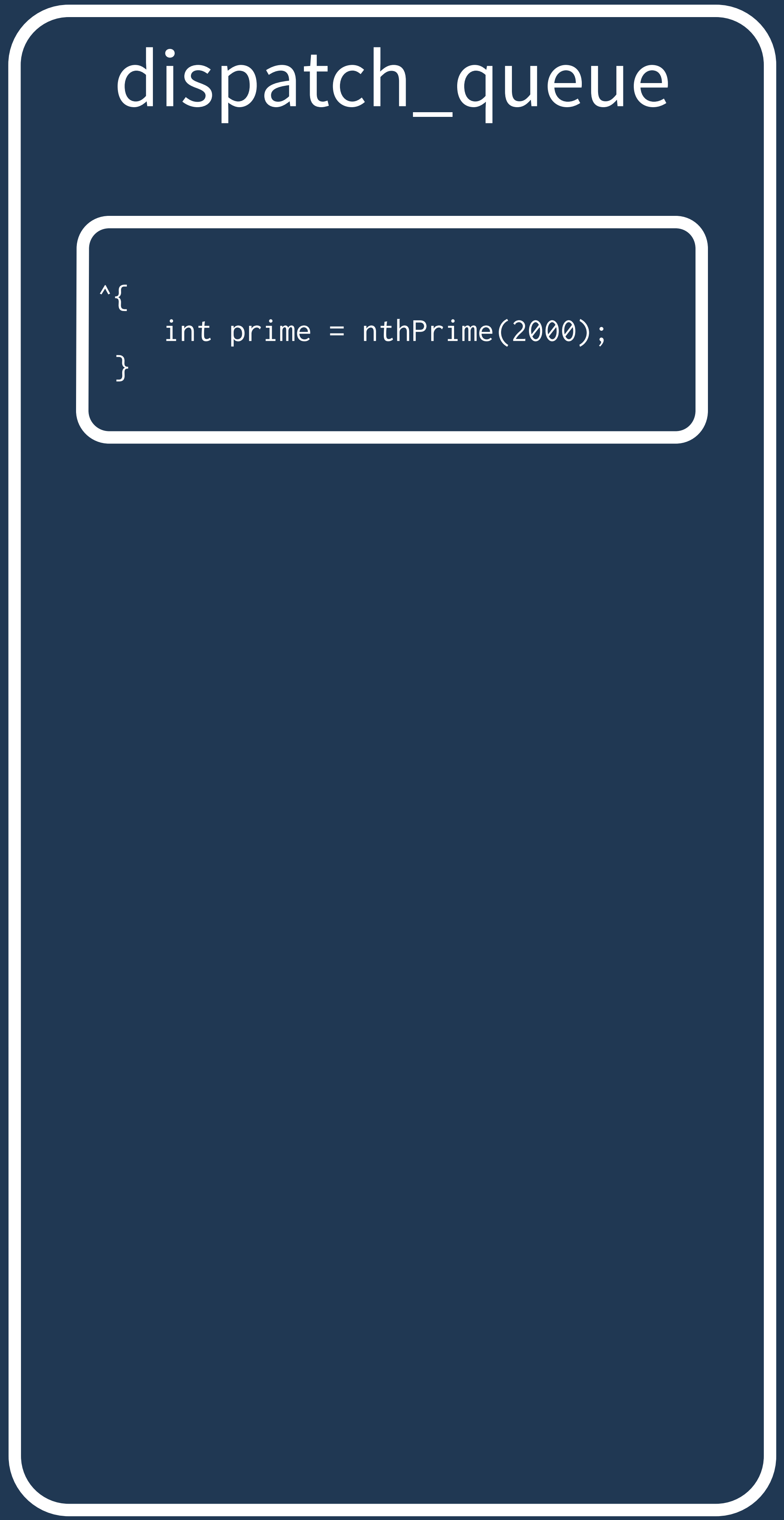


Dispatch Queues



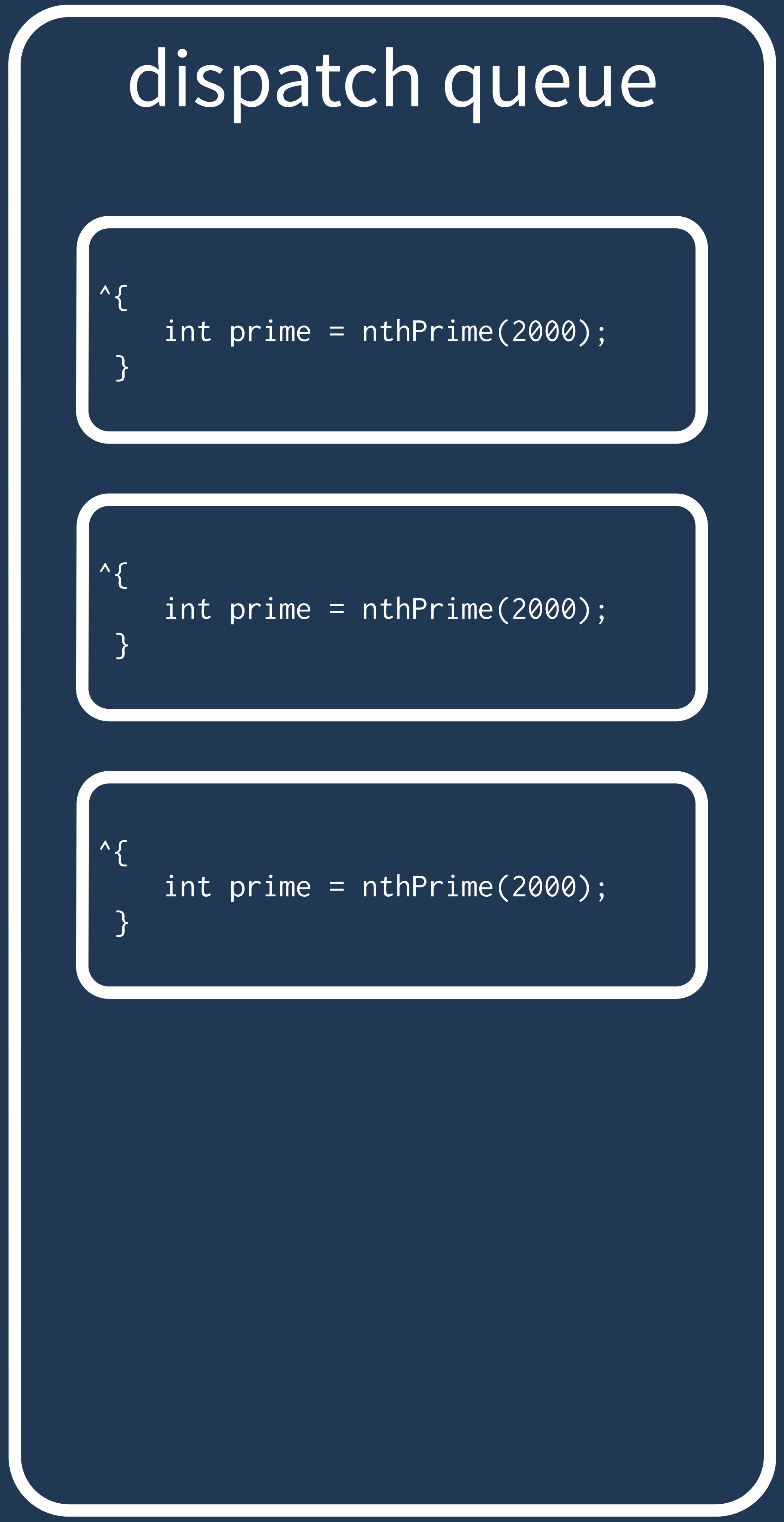
```
^{
    int prime = nthPrime(2000);
 }
```

Dispatch Queues



```
^{
    int prime = nthPrime(2000);
 }
```

Dispatch Queues



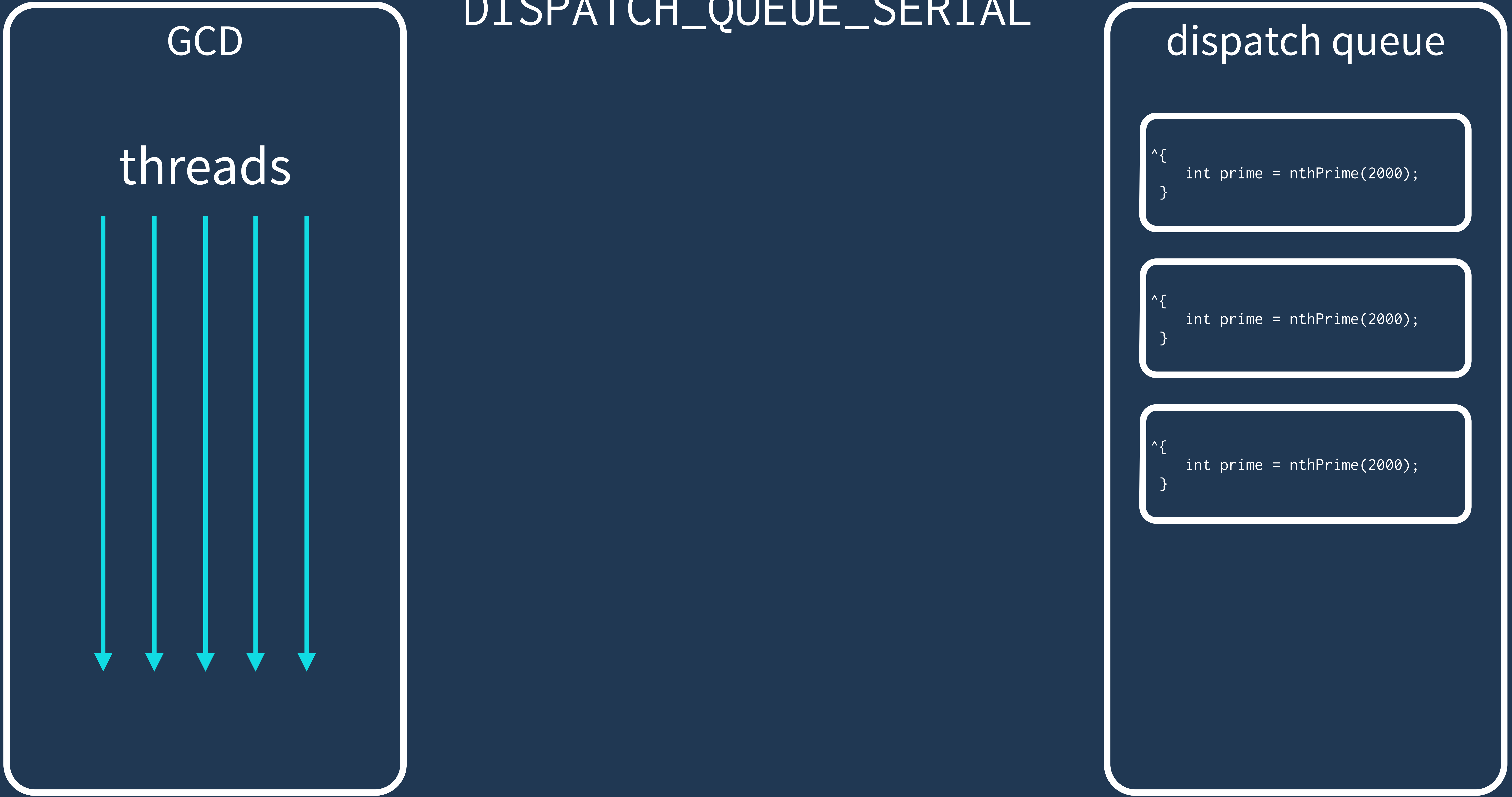
```
^{
    int prime = nthPrime(2000);
 }
```

```
^{
    int prime = nthPrime(2000);
 }
```

```
^{
    int prime = nthPrime(2000);
 }
```

Dispatch Queues

DISPATCH_QUEUE_SERIAL



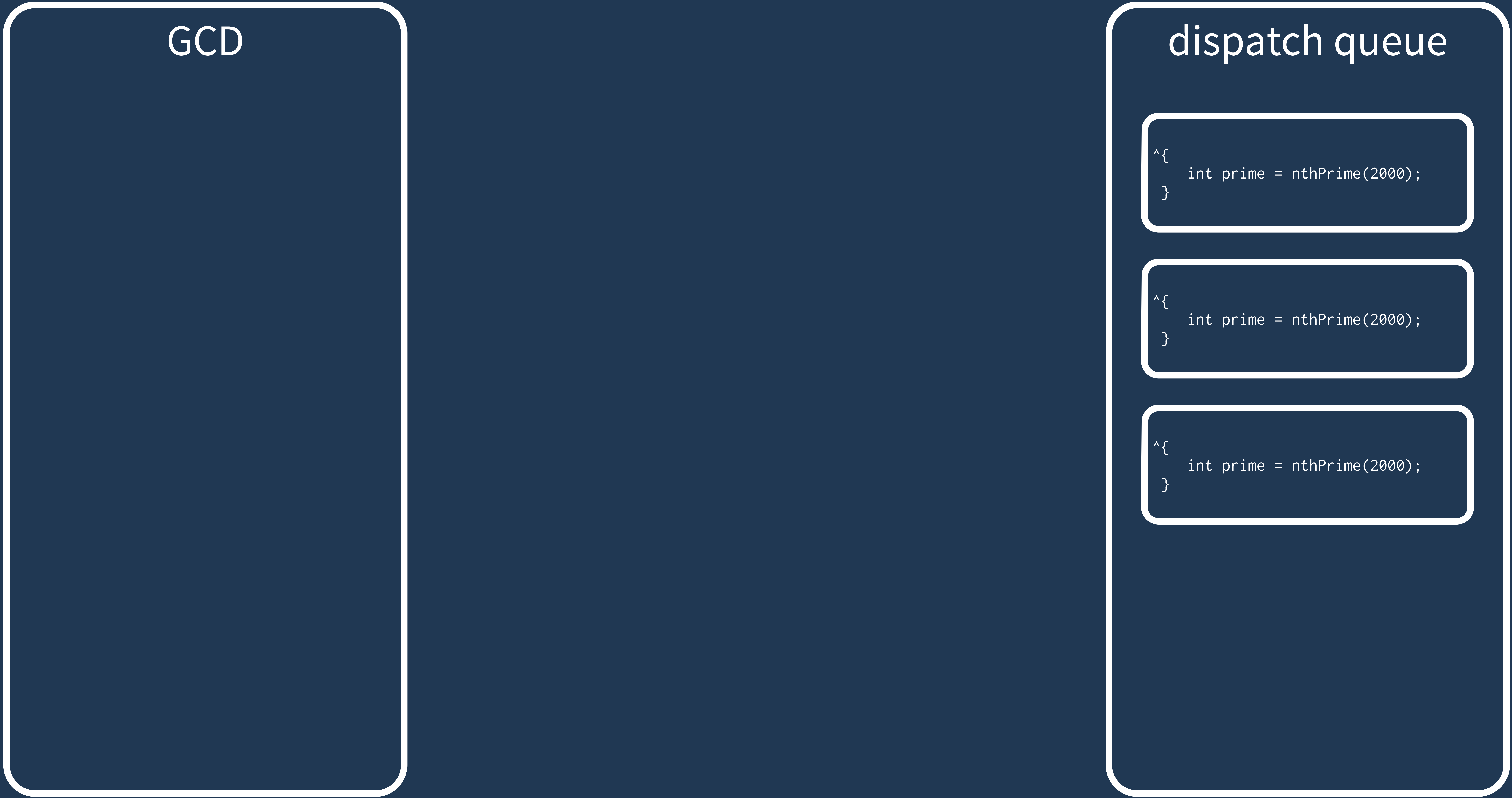
```
^{
    int prime = nthPrime(2000);
 }
```

```
^{
    int prime = nthPrime(2000);
 }
```

```
^{
    int prime = nthPrime(2000);
 }
```

Dispatch Queues

DISPATCH_QUEUE_CONCURRENT



```
^{
    int prime = nthPrime(2000);
 }
```

```
^{
    int prime = nthPrime(2000);
 }
```

```
^{
    int prime = nthPrime(2000);
 }
```

Global Queues

Concurrent

```
dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0);
```

```
DISPATCH_QUEUE_PRIORITY_HIGH
DISPATCH_QUEUE_PRIORITY_DEFAULT
DISPATCH_QUEUE_PRIORITY_LOW
DISPATCH_QUEUE_PRIORITY_BACKGROUND
```

Global Queues

Serial

```
dispatch_get_main_queue();
```

```
DISPATCH_QUEUE_PRIORITY_HIGH
Runs on main, like a boss!
```

Roll your own queues

```
dispatch_queue_create("MySerialQueue", DISPATCH_QUEUE_SERIAL);

dispatch_queue_create("MyConcurrentQueue", DISPATCH_QUEUE_CONCURRENT);
```

Submitting a block

```
// Next line returns immediately
dispatch_async(queue, ^{
   NSLog(@“Aww yeah! On a queue!”);
}


// Next line blocks
dispatch_sync(queue, ^{
   NSLog(@“Aww yeah! On a queue!”);
}
```

Operation Queues

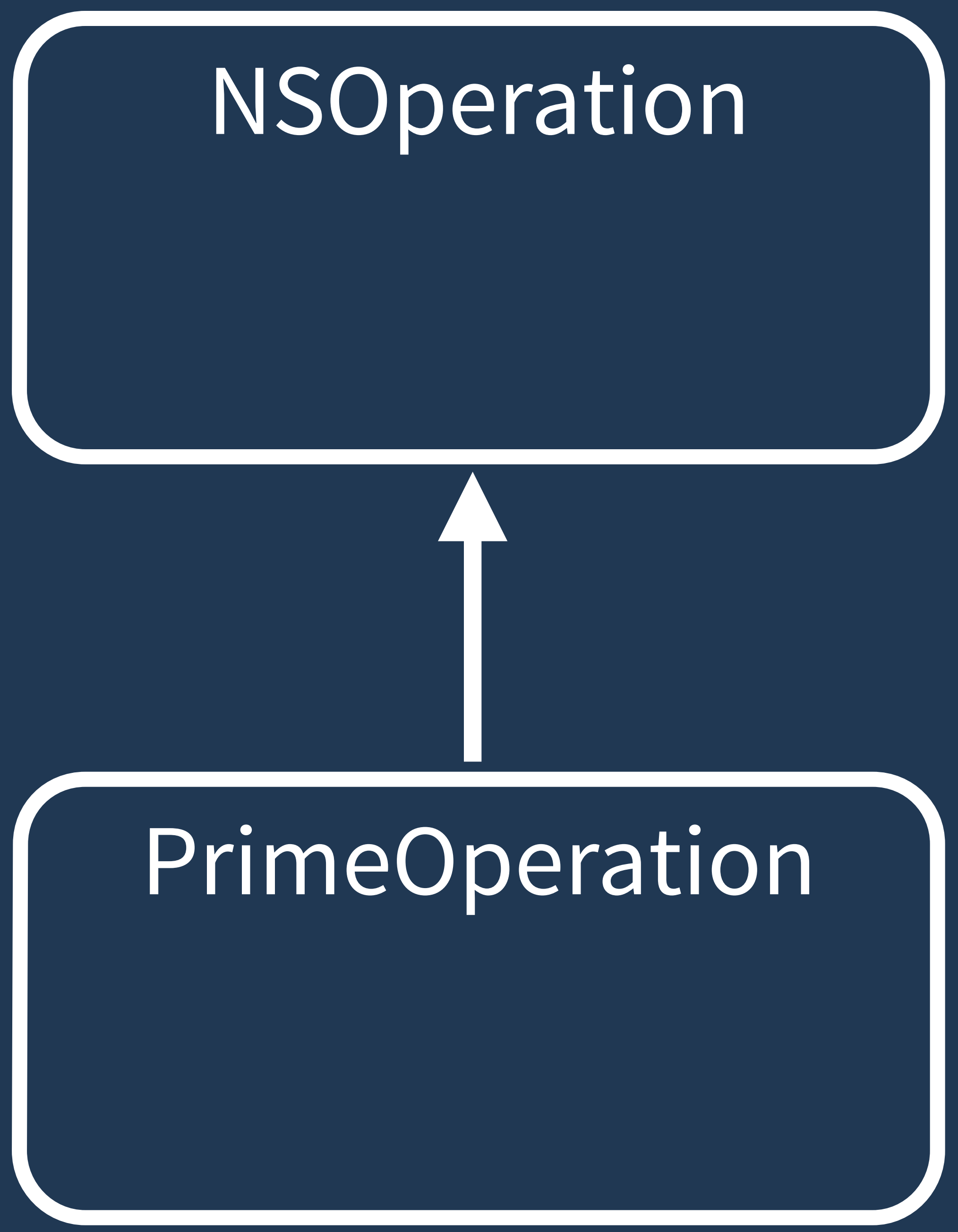
Operation Queues

- Higher order Objective-C based API
- Supports dependencies
- Pausing and cancellation

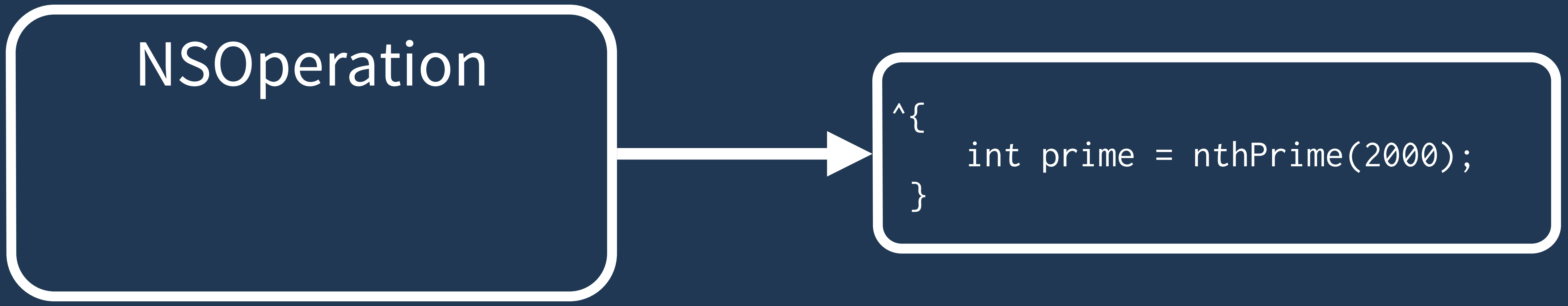
Operation

NSOperation

Operation



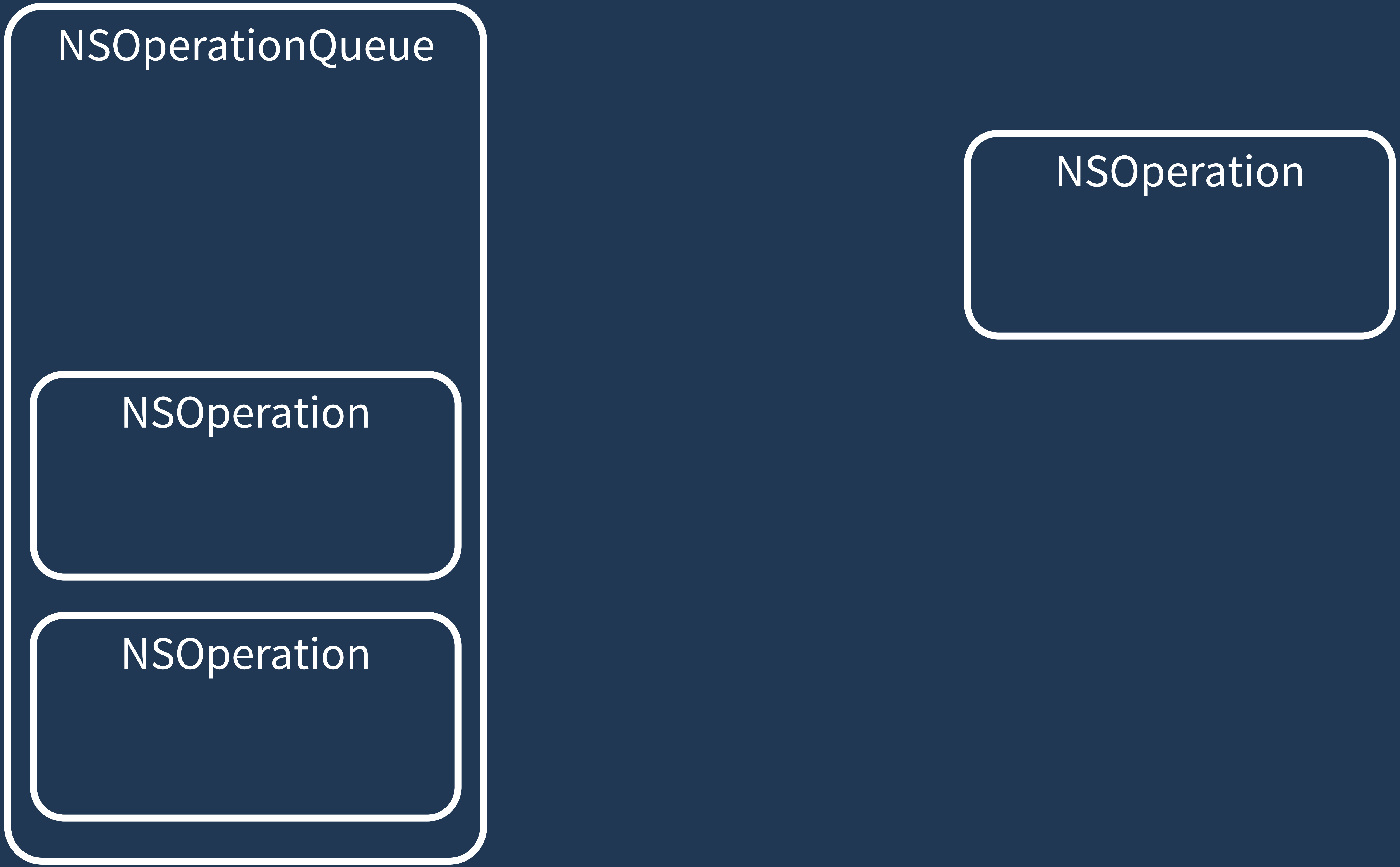
Operation



```
^{
    int prime = nthPrime(2000);
 }
```

```
NSBlockOperation* theOp = [NSBlockOperation blockOperationWithBlock: ^{
     int prime = nthPrime(2000);
}];
```

Operation Queues



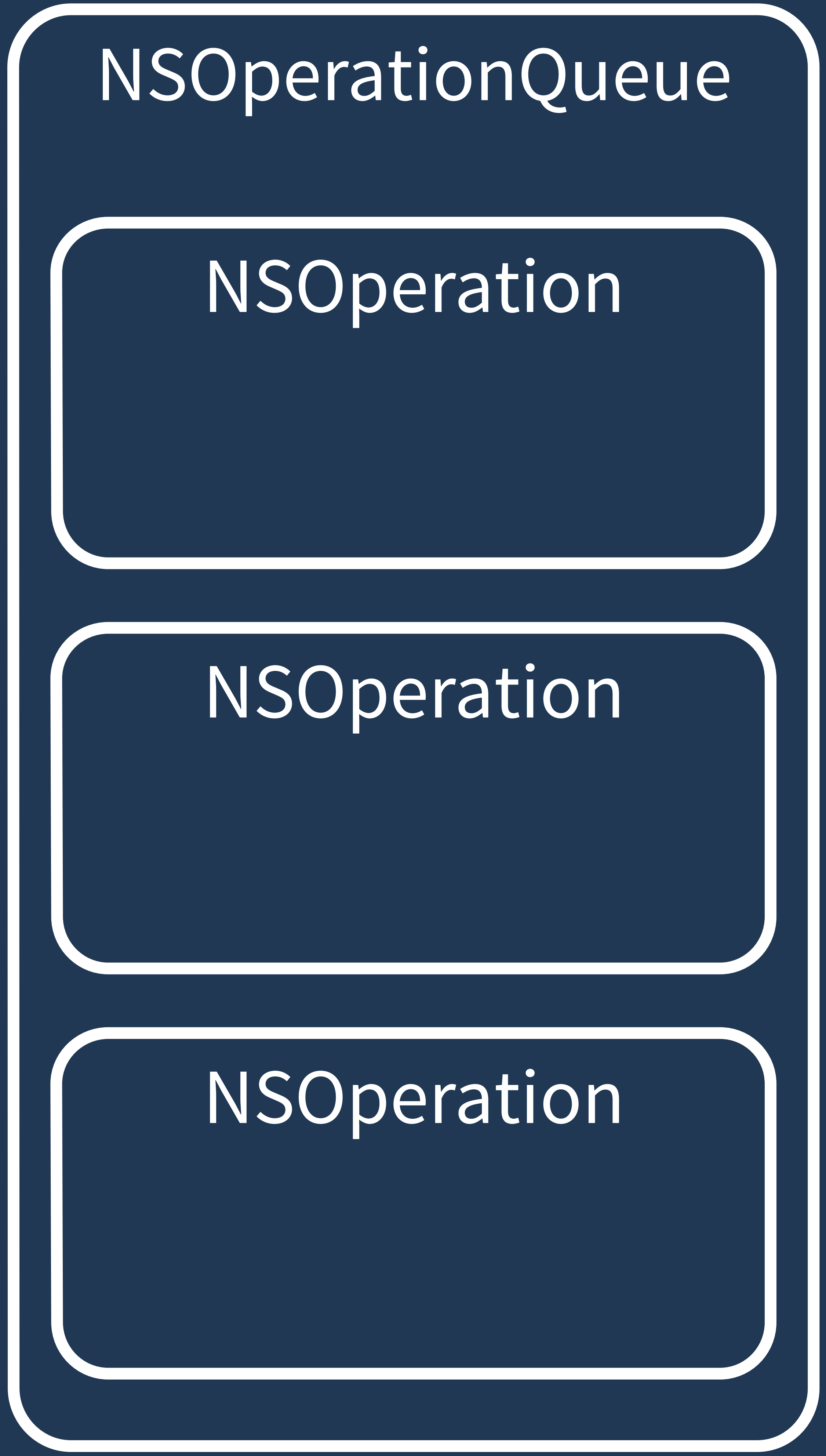
Operation Queues

Just like GCD queues:

- Can use blocks
- Either serial or concurrent
- Concurrent queues autoscale

Additionally:

- Cancelable
- Supports dependencies
- Task based priorities



Performance Lab