Going Native

JEE to iOS

You

Enterprise Java

iOS development

Me

Abe White

Architecture

JEE

I have yet to see any problem, however complicated, which, when looked at in the right way, did not become still more complicated.

- Poul Anderson

Architecture

JEE

JSP

Servlet

JAX

EJB

Logic

Entity

Store

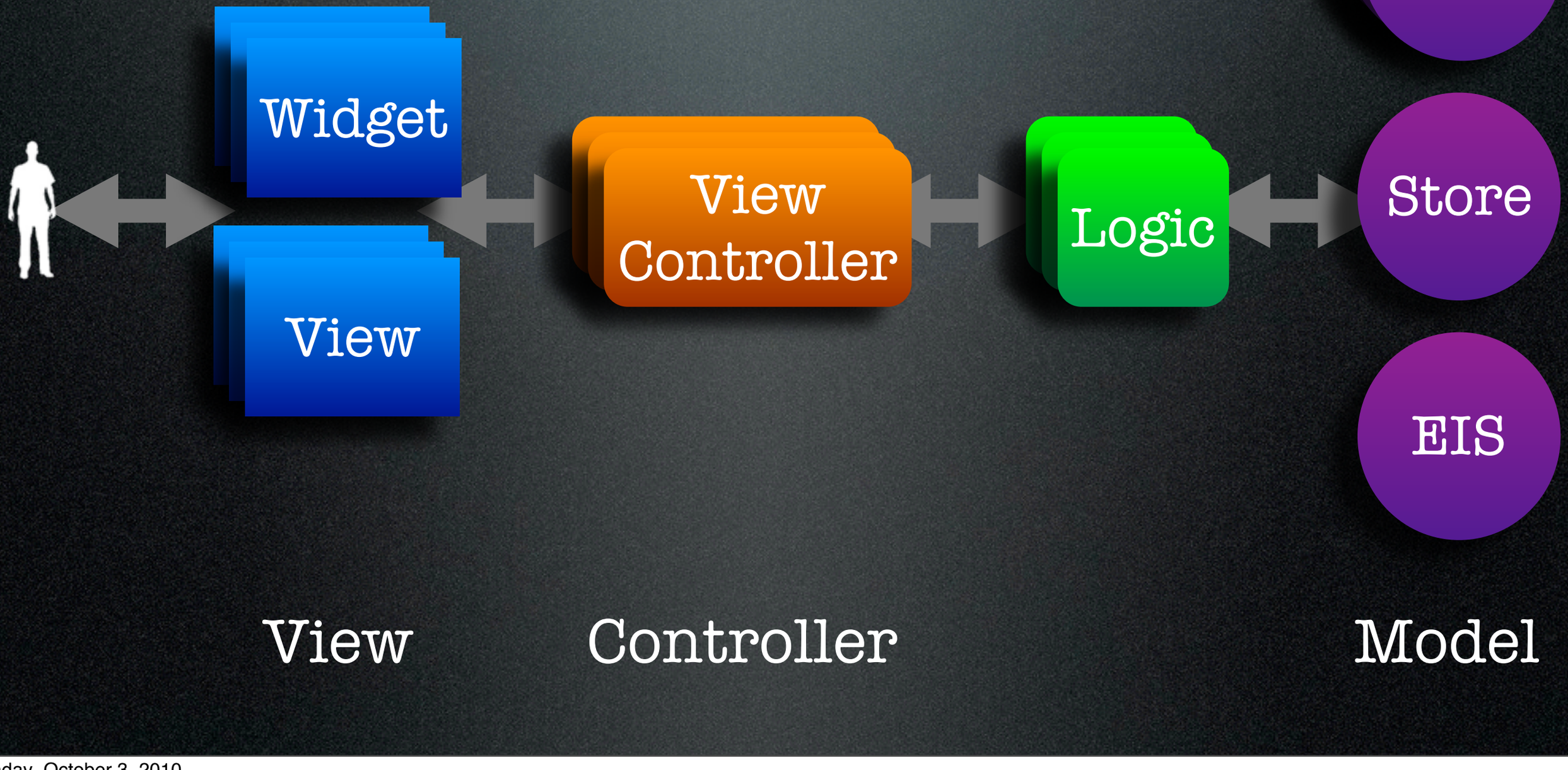
Presentation

Business

Integration

Architecture

iOS



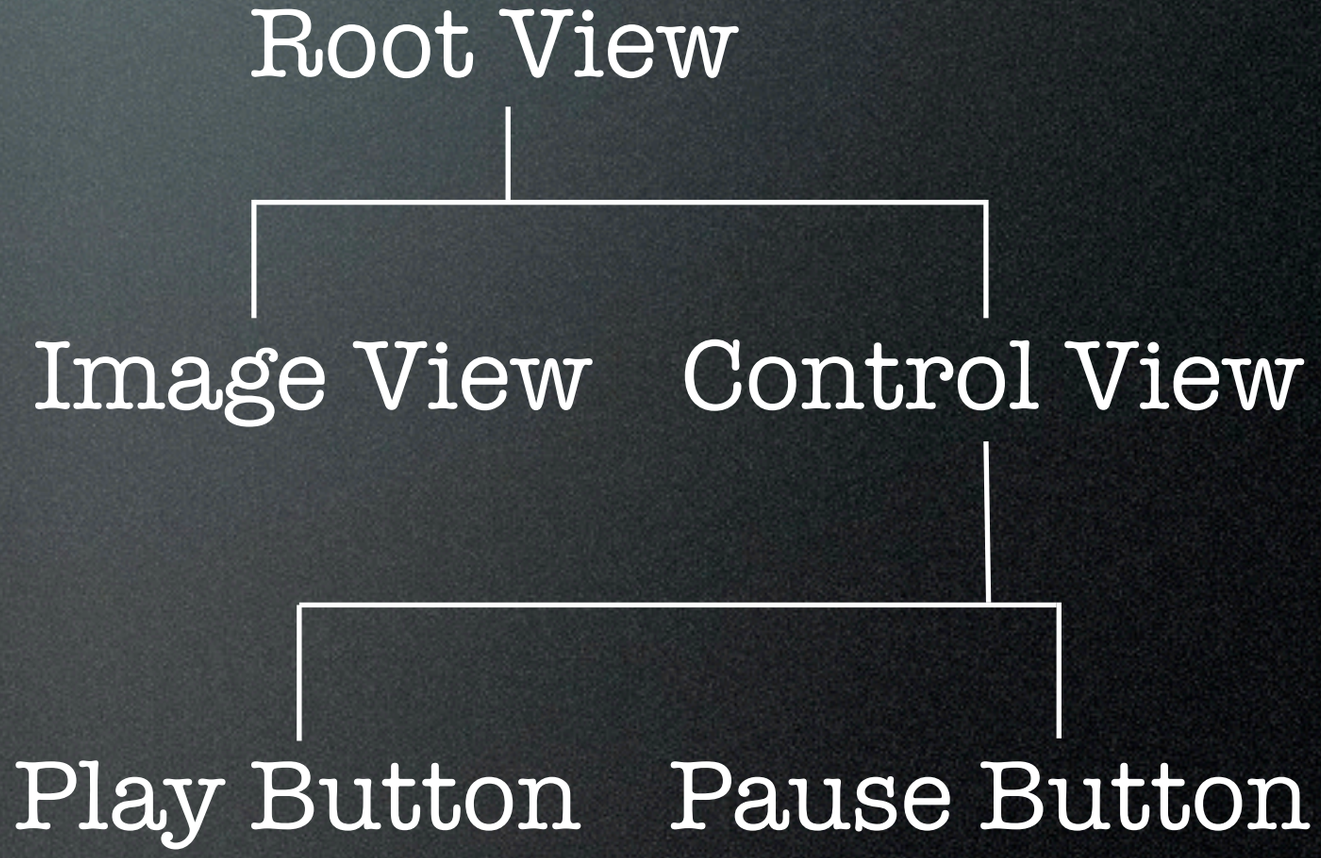
M Model

V View

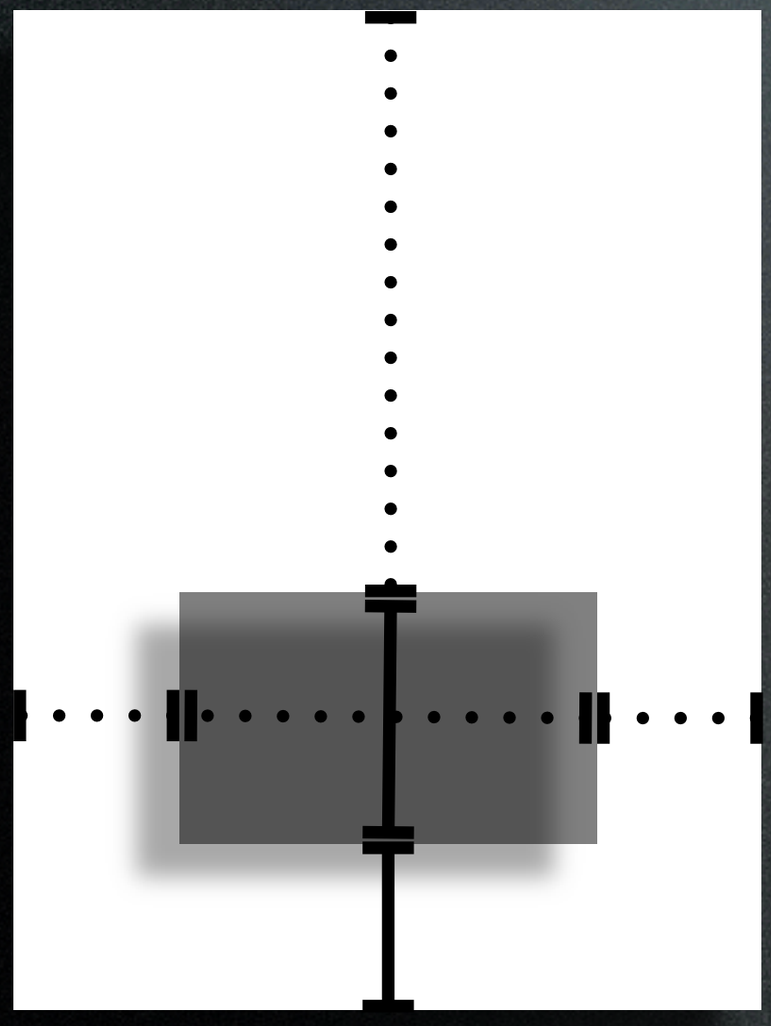
C Controller

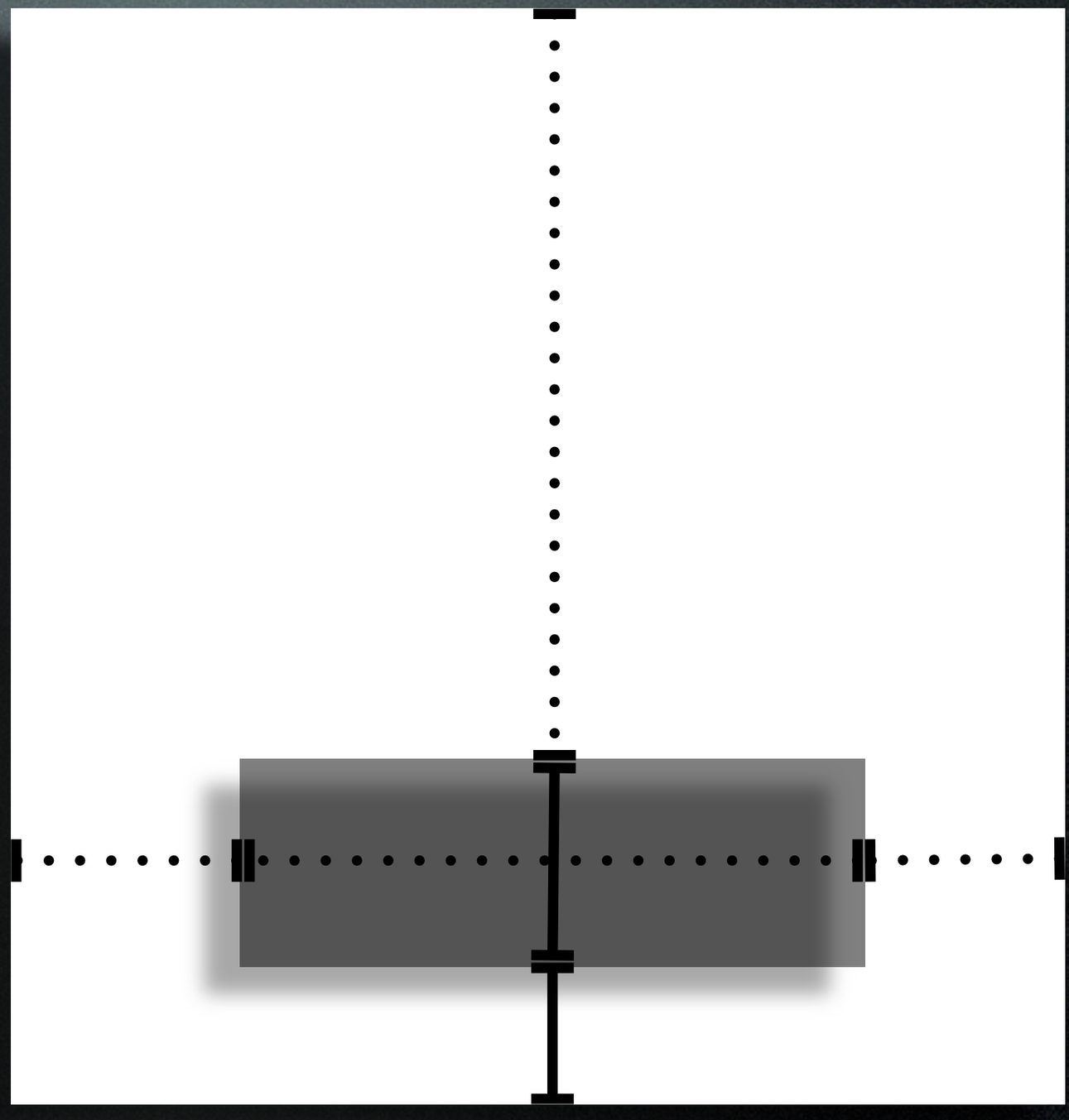
V

View Tree



V Layout





C

ObjC Syntax

```
int width = view.getWidth();
int height = view.getHeight();
view.setSize(10, 20);
```

```
int width = [view width];
int height = [view height];
[view setWidth:10 height:20];
```

C Listeners vs Actions

```
button.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(Event e) {
        doSomething();
    }
});

[button addTarget:self
    action:@selector(doSomething)
    forControlEvents:TouchUpInside];
```

C Notifications

C Notifications

```
[[NotificationCenter defaultCenter]
   postNotificationName:PersonDidDeleteNotification
   object:person];
                                        . . .
[[NotificationCenter defaultCenter] addObserver:self
   selector:@selector(personDidDelete:)
   name:PersonDidDeleteNotification
   object:nil];
                                        . . .
- (void)personDidDelete:(Notification*)notification {
   Person *person = [notification object];
   ...
}
```

C Delegates

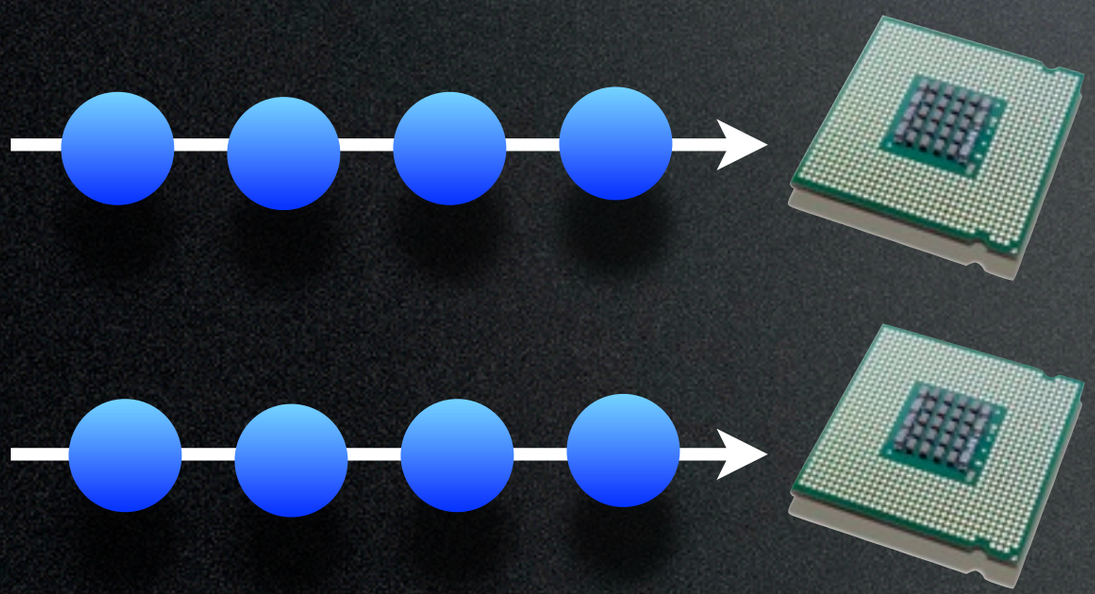
```
@protocol ListViewDelegate

- (int)numberOfItemsInListView:(ListView *)list;
- (ListItem *)listView:(ListView *)list itemAtIndex:(int)i;
...

@optional

- (void)listView:(ListView*)list didSelectItemAtIndex:(int)i;
...

@end
```

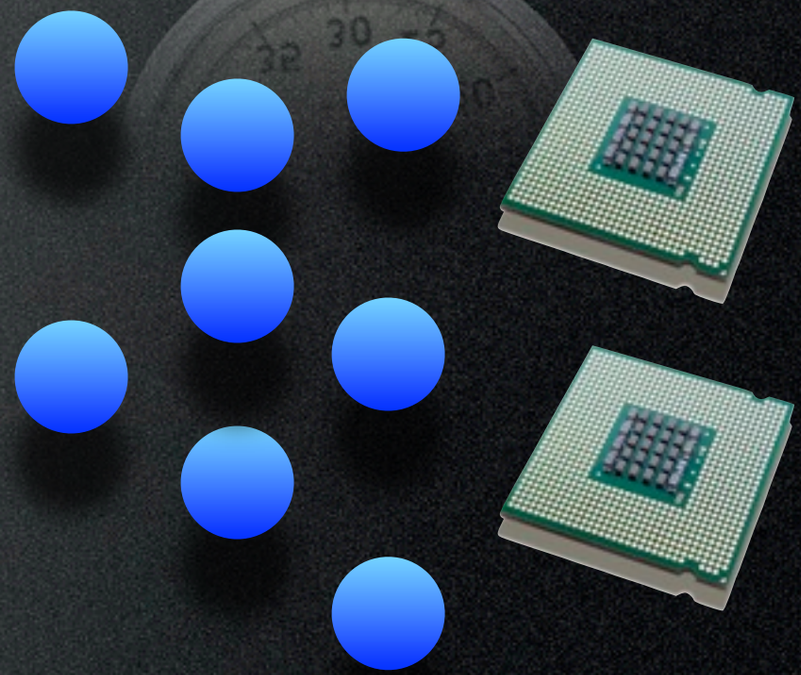
Concurrency

Any sufficiently advanced bug is indistinguishable from a feature.

- Bruce Brown

Concurrency



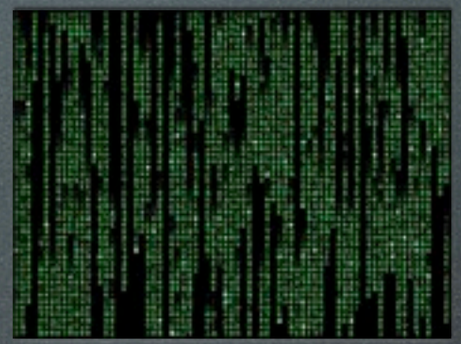


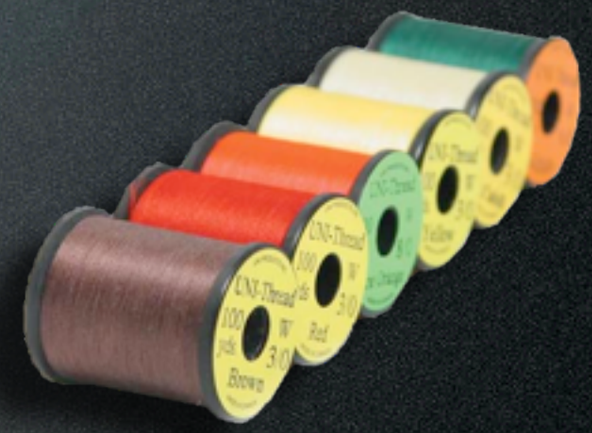
Concurrency

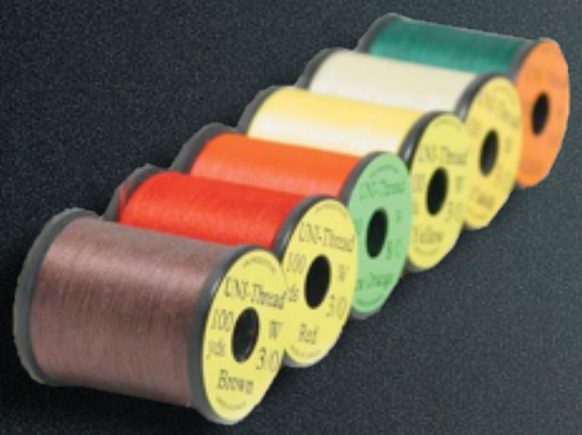
JEE

Concurrency

iOS







Concurrency

iOS

```
Data *data = ...;
[self performSelectorInBackground:@selector(processData:)
   withObject:data];
```

. . .

```
- (void)processData:(Data*)data {
   id result = ...;
   [self performSelectorOnMainThread:@selector(display:)
      withObject:result
      waitUntilDone:NO];
}
- (void)display:(id)result {
   [view display:result];
   [view show];
}
```

Concurrency

iOS

```
[self performSelector:@selector(hideControls)
   withObject:nil
   afterDelay:3.0];
```

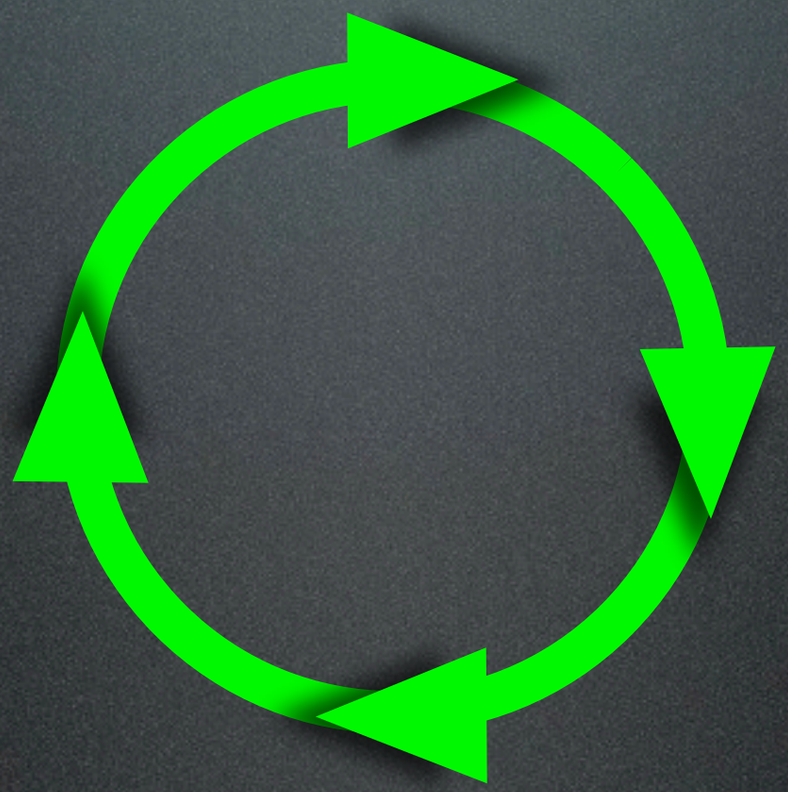
. . .

```
[Object cancelPreviousPerformRequestsWithTarget:self
   selector:@selector(hideControls)
   object:nil];
```

Concurrency

iOS

Controller



TouchEvent

Update Views

Idle

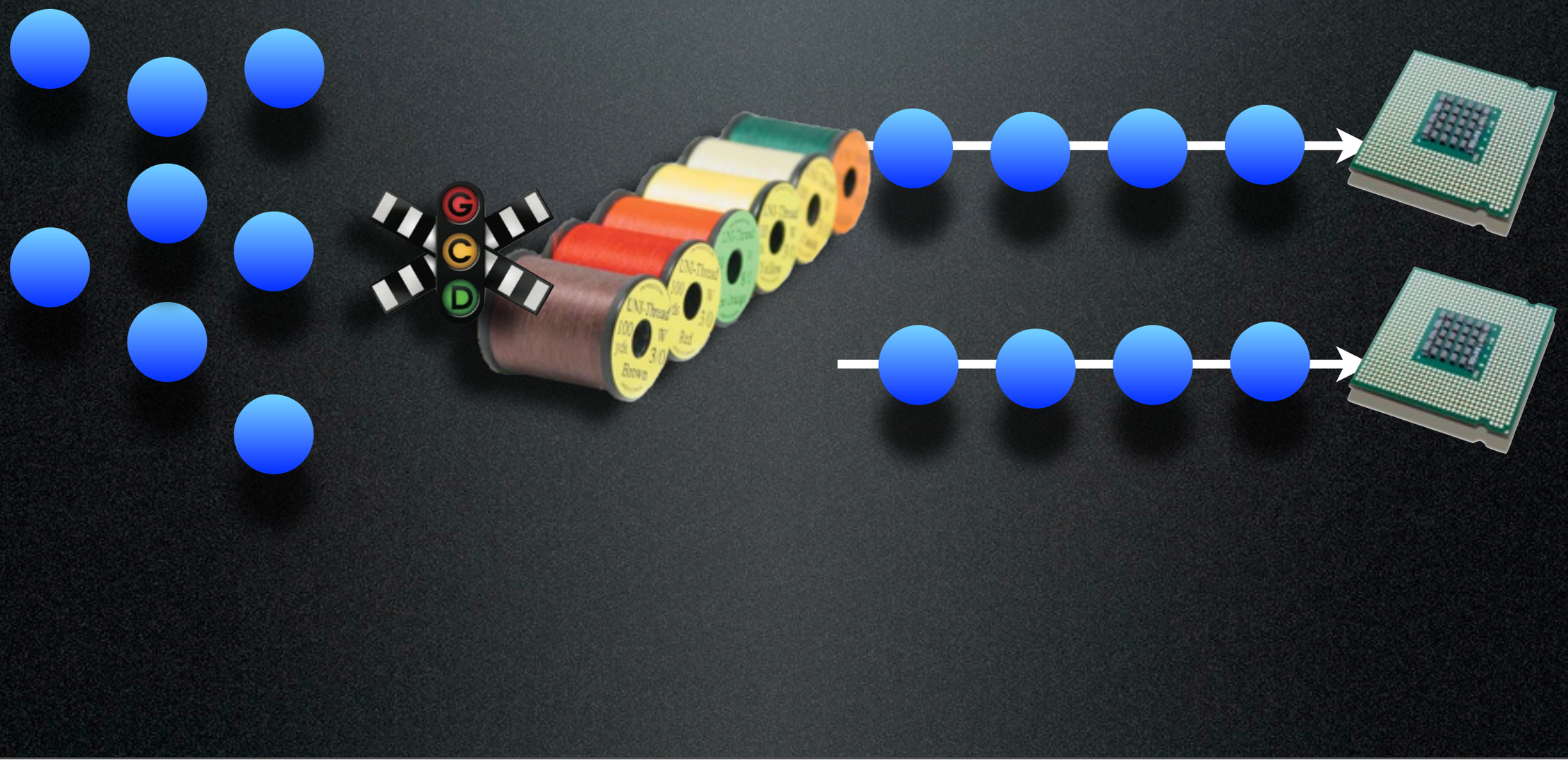
Concurrency

iOS

```
[busyIndicator show];
[self performSelector:@selector(processData:)
   withObject:data
   afterDelay:0.0];
```

Concurrency

iOS



Concurrency

iOS

Queueing Model

C-level API

Blocks

Concurrency

iOS

Blocks

```
float (^myBlock)(int, int)  = ^(int x, int y) {
    return x / (float)y;
};

float result = myBlock(1, 2);
```

Concurrency

iOS

Blocks

```
int y = 2;

float (^myBlock)(int) = ^(int x) {
    return x / (float)y;
};

float result = myBlock(1);
```

Concurrency

iOS

Blocks

```
(float (^)(int)) divideByY(int y) {
    return ^(int x) {
        return x / (float)y;
    };
}

float (^divideBy2)(int) = divideByY(2);
float result = divideBy2(1);
```

Concurrency

iOS

```
Data *data = ...;
id result = [self processData:data];
[view display:result];
[view show];
```

Concurrency

iOS

```
Data *data = ...;
dispatch_async(dispatch_get_global_queue(0,0), ^{
    id result = [self processData:data];
    dispatch_async(dispatch_get_main_queue(), ^{
        [view display:result];
        [view show];
    });
});
```

Concurrency

iOS

```
Cache *cache = ...;
dispatch_queue_t cacheQ = dispatch_queue_create(
  “com.xyz.cache”, NULL);
                                        . . .
Record *record = ...;
dispatch_async(cacheQ, ^{
  [cache addRecord:record];
});

                                        . . .
__block Record *result;
dispatch_sync(cacheQ, ^{
  result = [cache recordForKey:key];
});
```

Memory

As you get older three things happen. The first is your memory goes, and I can't remember the other two...

- Norman Wisdom

Memory

JEE

Garbage Collected

Memory-Sensitive Datastructures

Memory

iOS

Reference Counted

Low Memory Notifications

Memory

Reference Counted iOS

alloc = create +1

retain = increase reference count +1

release = decrease reference count -1

autorelease = release later -1

Memory

Reference Counted iOS

1. Retain what you need later

2. Release what you alloc or retain

3. Autorelease what you return

Memory

Reference Counted iOS

```
@interface Person {
   String *_firstName;
   String *_lastName;
}
...
@property String *firstName;
@property String *lastName;
- (String *)fullName;
...
@end
```

Memory

Reference Counted iOS

```
- (void)setFirstName:(String*)newFirst {
    [newFirst retain];
    [_firstName release];
    _firstName = newFirst;
}
```

Memory

Reference Counted iOS

```
- (String *)fullName {
    String *full = [[String alloc] initWithFormat:
      “%s %s”, _firstName, _lastName];
    return [full autorelease];
}
```

Memory

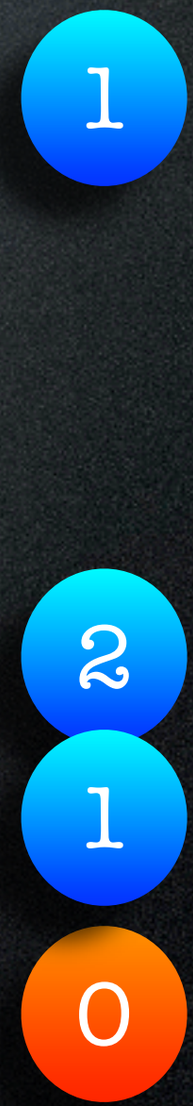
Reference Counted iOS

```
- (void)dealloc {
    [_firstName release];
    [_lastName release];
    [super dealloc];
}
```

Memory

Reference Counted iOS

```
Person *person = [[Person alloc] init];
person.firstName = “Abe”;
person.lastName = “White”;
String *fullName = [person fullName];
Log(fullName);
[list add:person];
[person release];

                                . . .

[list remove:person];
```



Memory

Memory Notifications iOS

```
[[NotificationCenter defaultCenter] addObserver:self
   selector:@selector(memoryWarning:)
   name:MemoryWarningNotification
   object:nil];

                                    ...
- (void)memoryWarning:(Notification*)notification {
   [cache clear];
   ...
}
```

Memory

Memory Notifications iOS

```
- (void)viewDidUnload {
    [view release];
    view = nil;
    ...
    [super viewDidUnload];
}
```

Thank You

Fortunately, the second-to-last bug has just been fixed.

- Ray Simard