



# Kotlin Coroutines

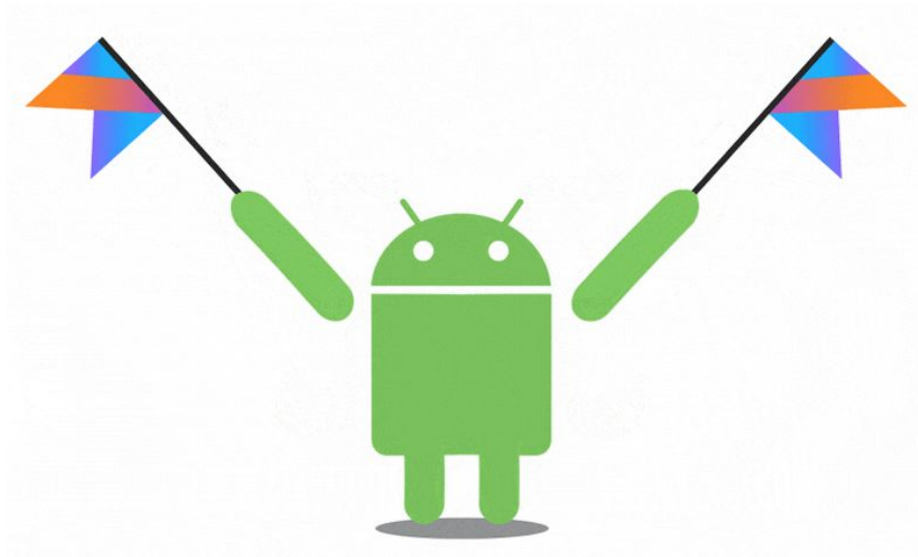
Victor Kropp

@kropp • victor.kropp.name



10.01.2018

 Kotlin

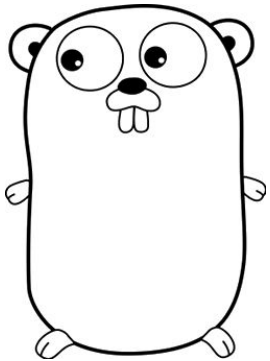




# Asynchronous programming

`async/await`

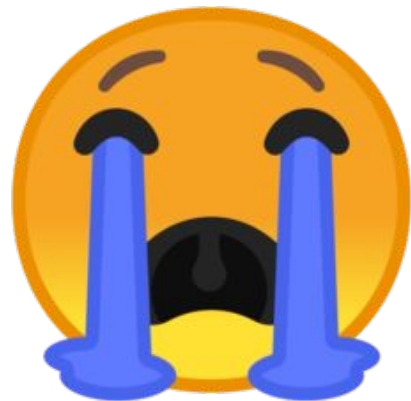
# Asynchronous programming



goroutines

# Callback Hell

```
        });  
    });  
});  
});  
});  
});  
});  
});  
});  
});
```



# The problem

Time consuming operation  
Freezes application

```
val bitmap = processBitmap("image.png")  
mImageView.setImageBitmap(bitmap)
```

# Worker threads

```
new Thread(new Runnable() {  
    public void run() {  
        final Bitmap bitmap = processBitMap("image.png");  
        mImageView.post(new Runnable() {  
            public void run() {  
                mImageView.setImageBitmap(bitmap);  
            }  
        });  
    }  
}).start();
```

Background thread  
OK!

# Worker threads

```
new Thread(new Runnable() {  
    public void run() {  
        final Bitmap bitmap = processBitMap("image.png");  
        mImageView.post(new Runnable() {  
            public void run() {  
                mImageView.setImageBitmap(bitmap);  
            }  
        });  
    }  
}).start();
```

Threads are expensive



# Worker threads

```
new Thread(new Runnable() {  
    public void run() {  
        final Bitmap bitmap = processBitMap("image.png");  
        mImageView.post(new Runnable() {  
            public void run() {  
                mImageView.setImageBitmap(bitmap);  
            }  
        });  
    }  
}).start();
```

Object allocations

# Callback style

```
fun processBitMap(name: String,  
                 callback: (Throwable?, Bitmap?) -> Unit)
```

# Future

```
fun processBitMap(name: String): Future<Bitmap>
```

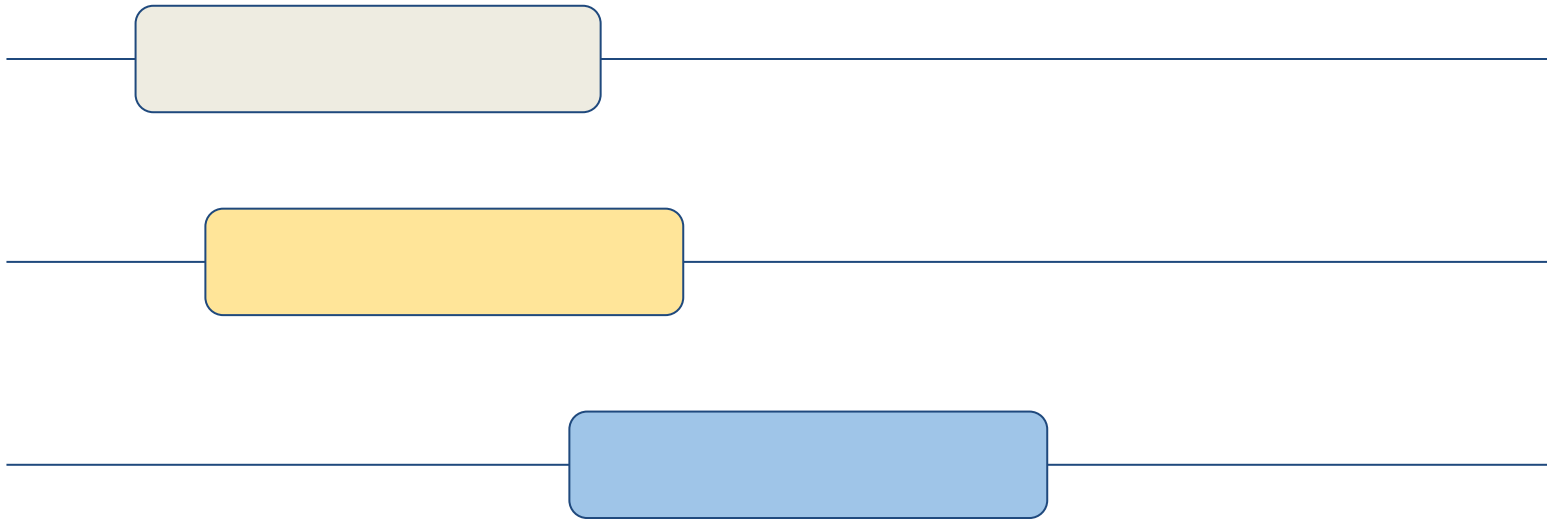
# Kotlin way

```
suspend fun processBitMap(name: String): Bitmap
```

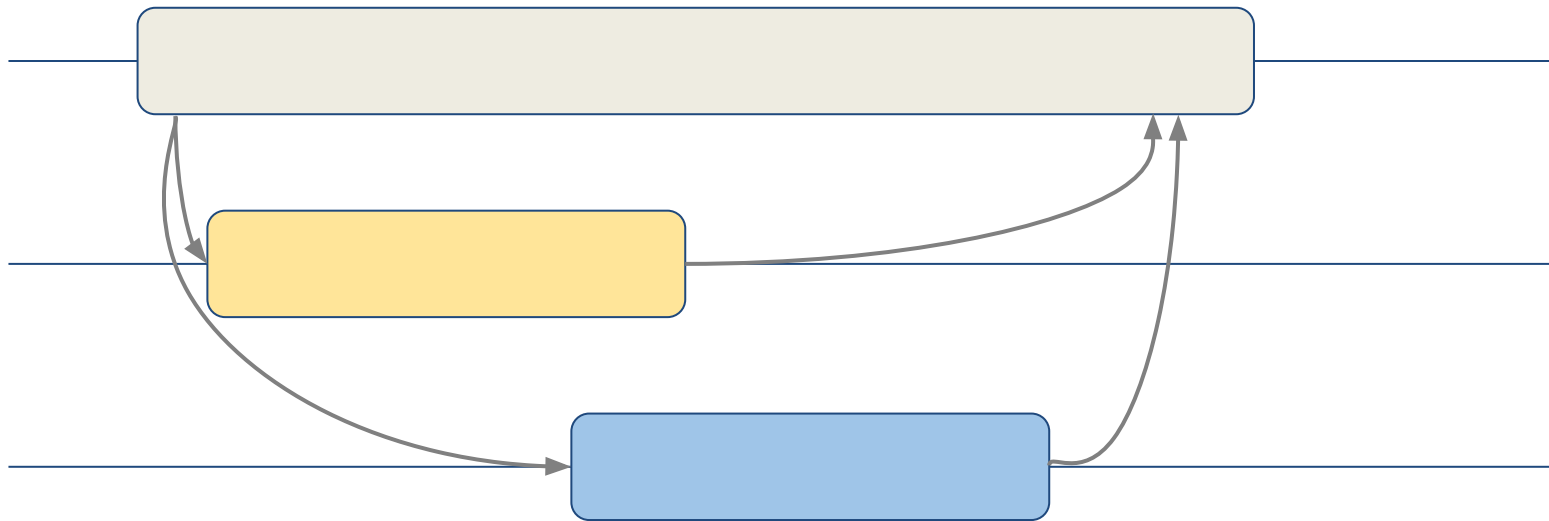
# Kotlin way

```
val bitmap = processBitmap("image.png")  
mImageView.setImageBitmap(bitmap)
```

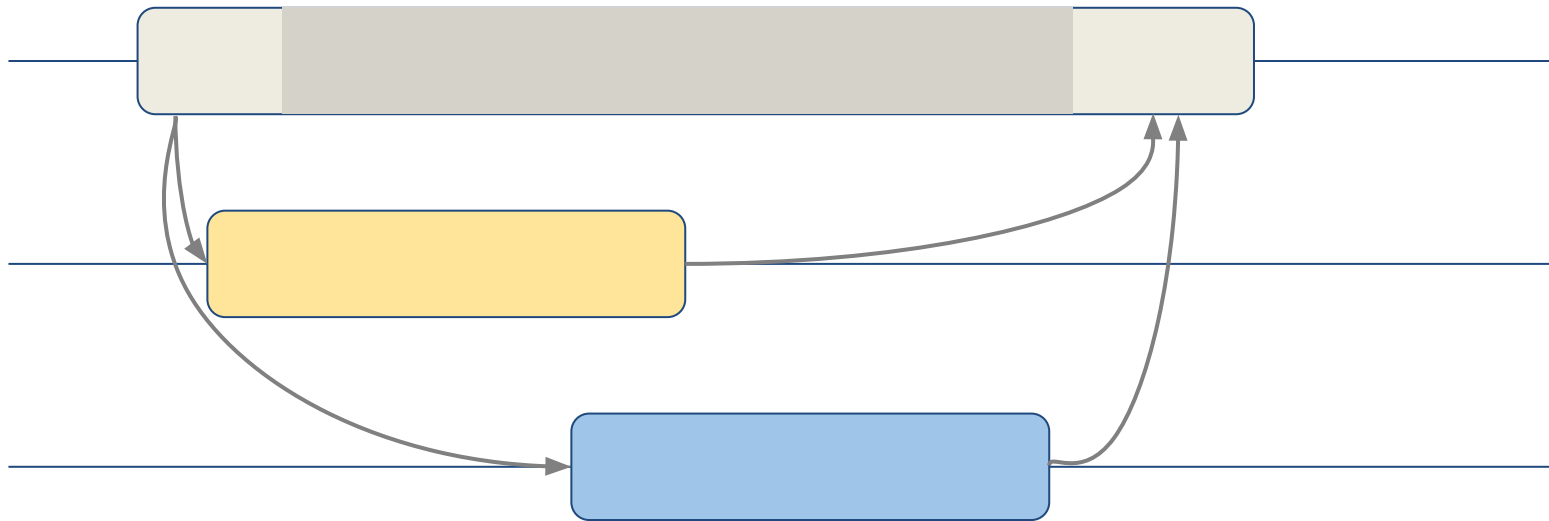
# Threads vs Coroutines



# Threads vs Coroutines

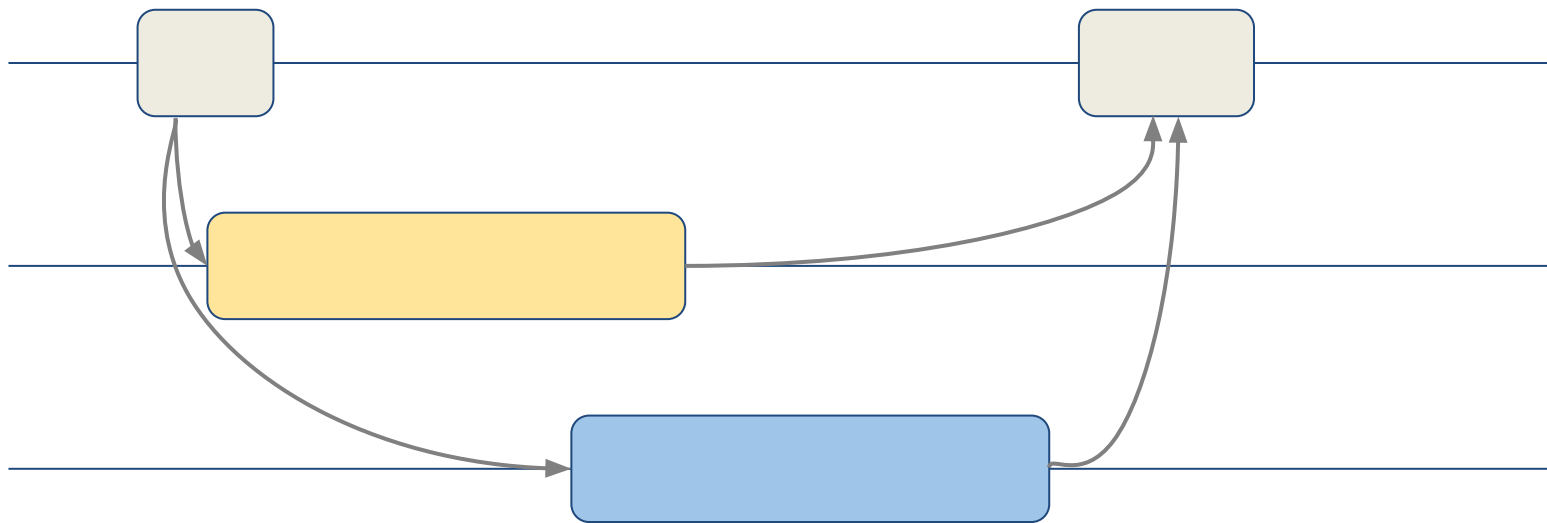


# Threads vs Coroutines

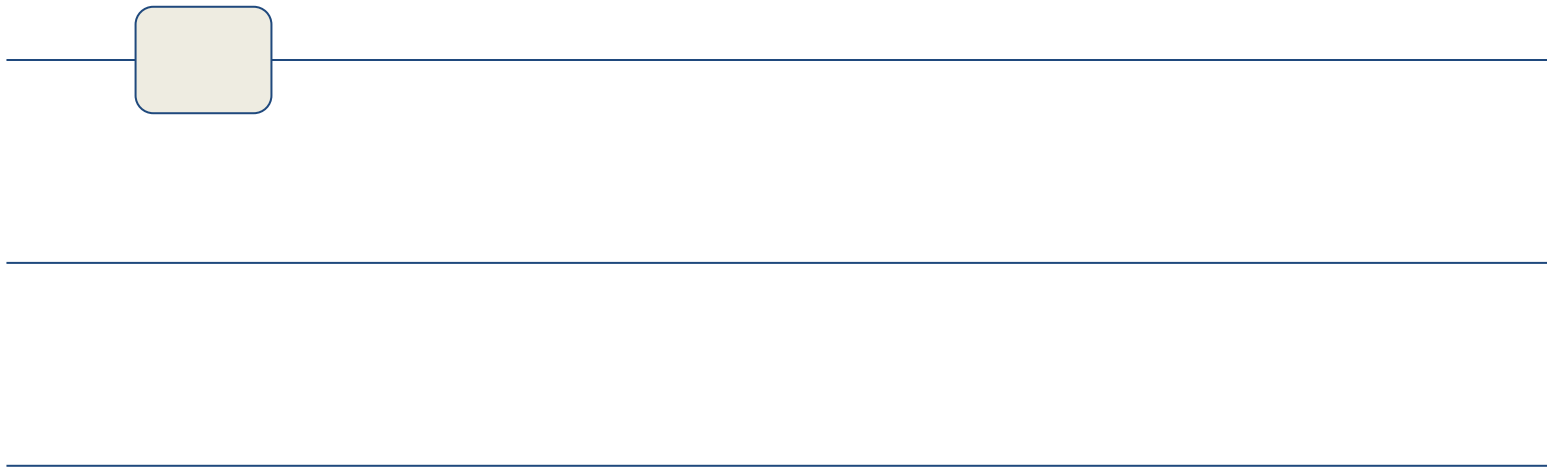




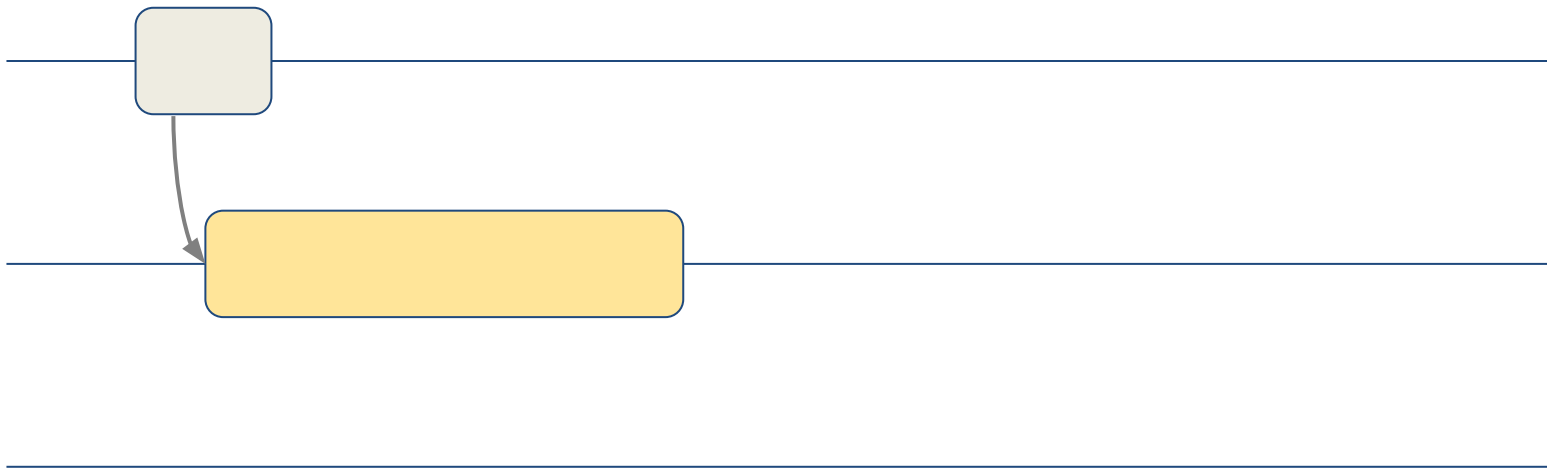
# Threads vs Coroutines



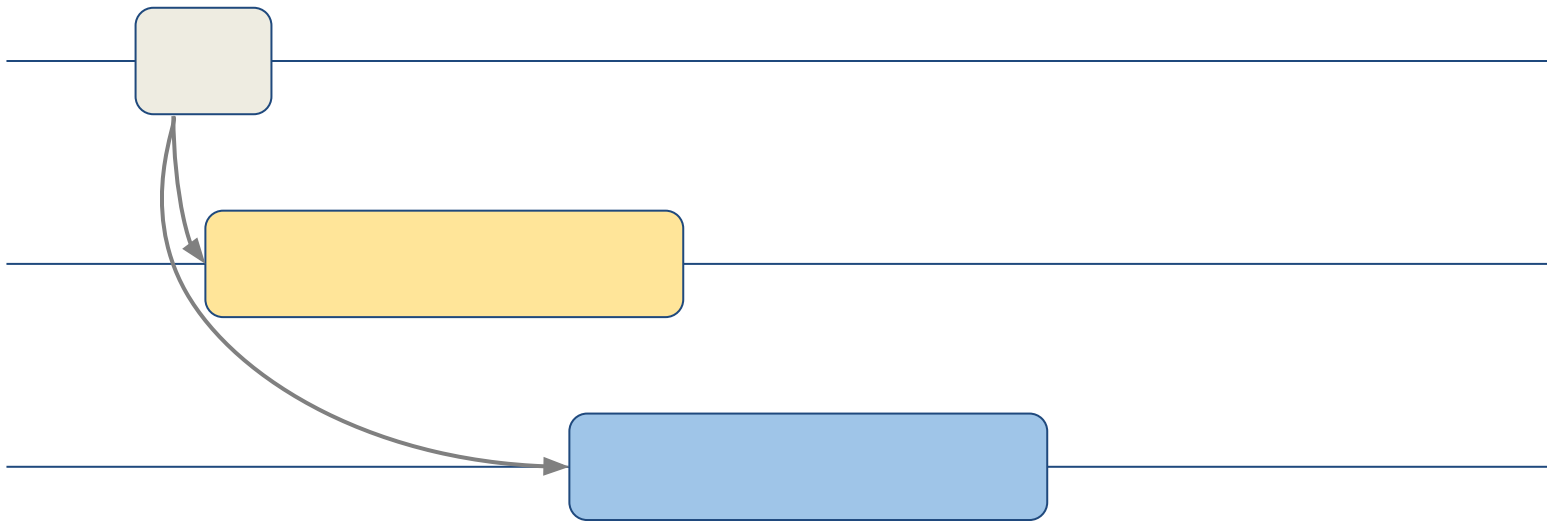
# Threads vs Coroutines



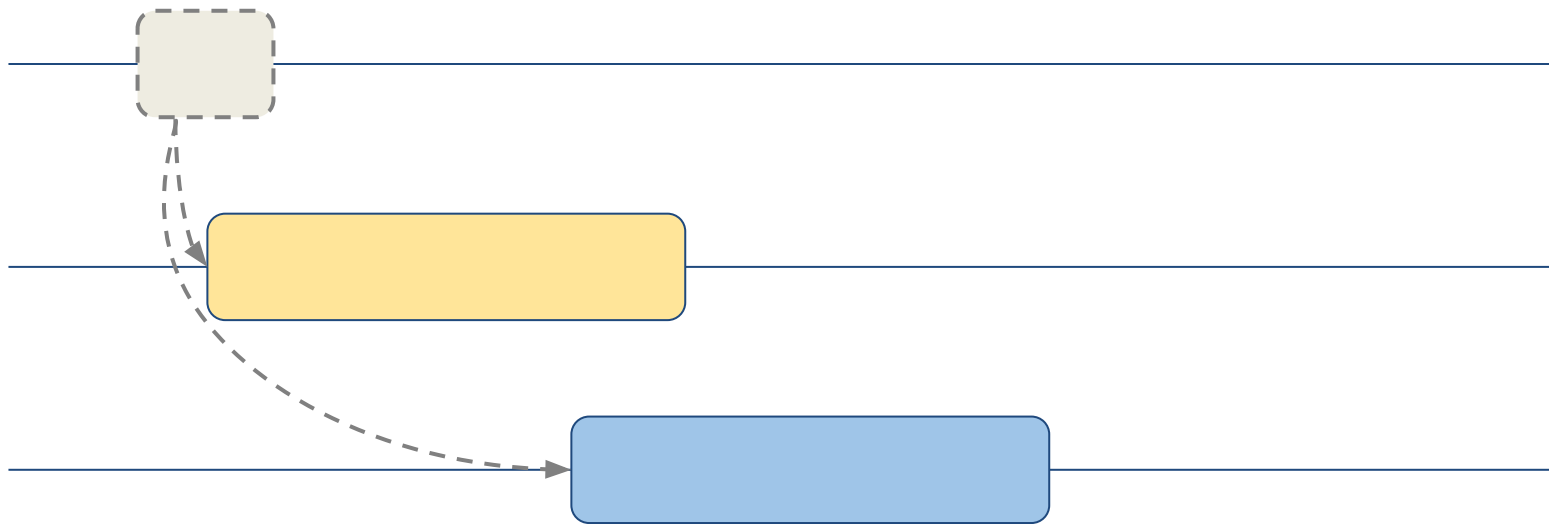
# Threads vs Coroutines



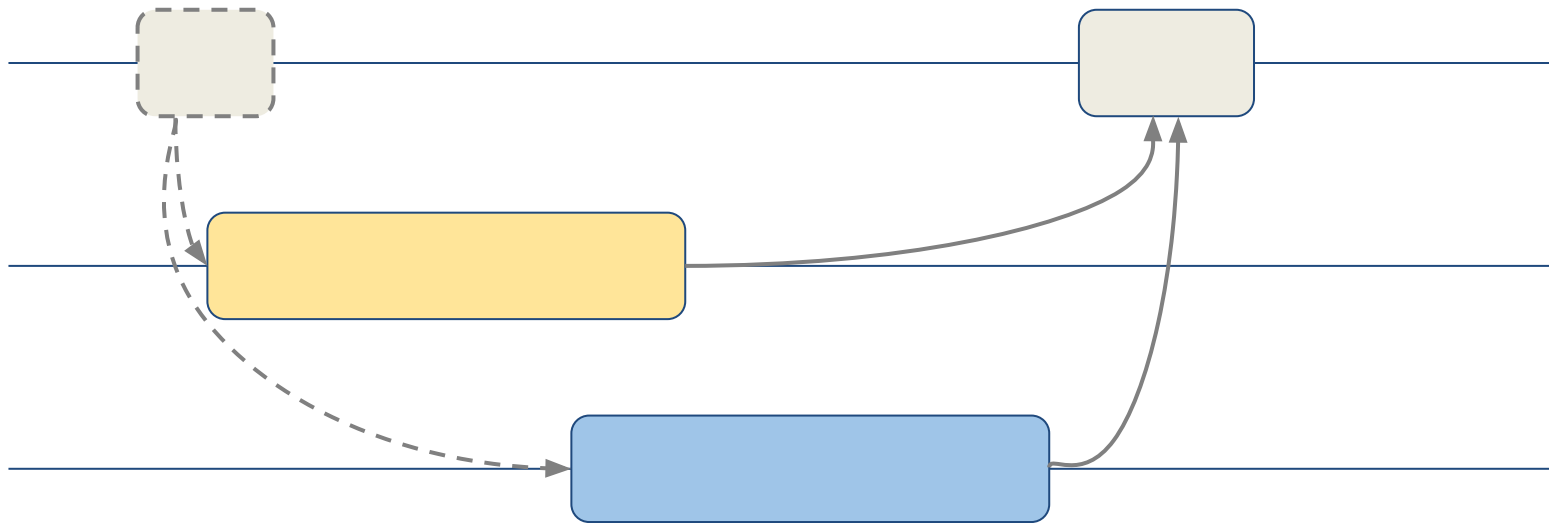
# Threads vs Coroutines



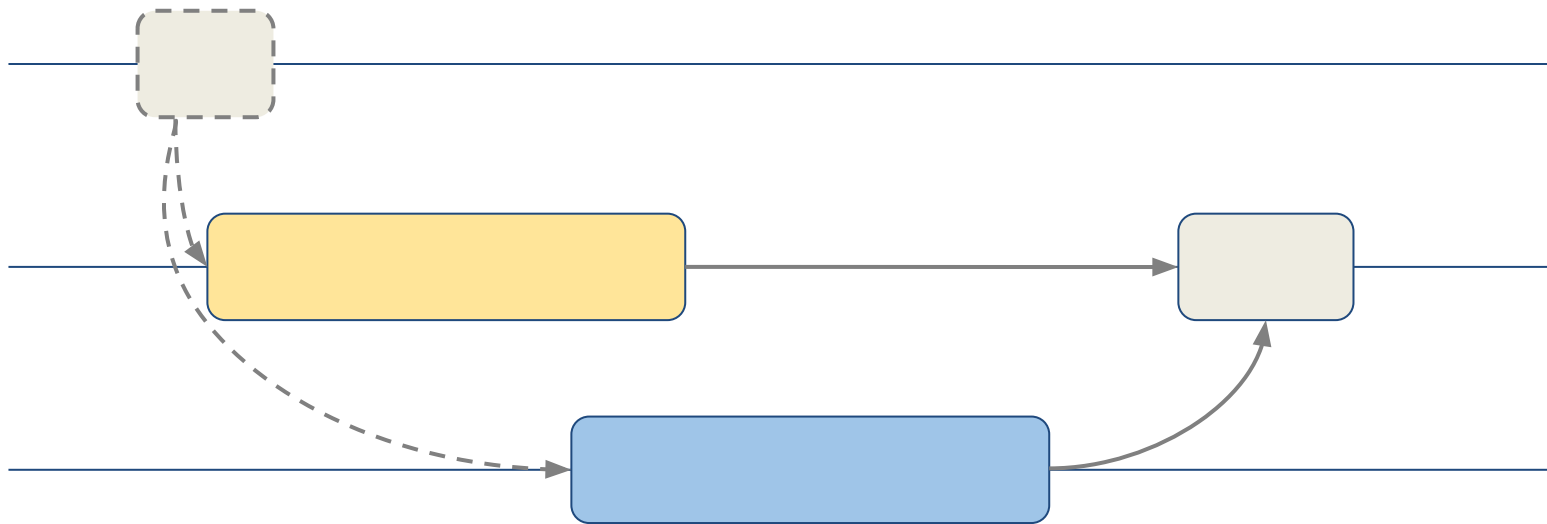
# Threads vs Coroutines



# Threads vs Coroutines



# Threads vs Coroutines



# suspend

```
fun applyImageFilter() {  
    val bitmap = processBitMap("image.png")  
    mImageView.setImageBitmap(bitmap)  
}
```

suspendable function must be called  
from another suspendable function



# suspend

```
suspend fun applyImageFilter() {  
    val bitmap = processBitMap("image.png")  
    mImageView.setImageBitmap(bitmap)  
}
```

suspendable function must be called  
from another suspendable function

# Coroutine builders

```
suspend fun applyImageFilter() {  
    val bitmap = processBitmap("image.png")  
    mImageView.setImageBitmap(bitmap)  
}
```

```
async {  
    applyImageFilter()  
}
```

# Coroutine builders

```
async(CommonPool) {  
    applyImageFilter()  
}
```

# Coroutine builders

```
async(CommonPool) {  
    applyImageFilter()  
}
```

```
async(UI) {  
    applyImageFilter()  
}
```

# Integration

```
suspend fun applyImageFilter() {  
    val bitmap = processBitMap("image.png").await()  
    mImageView.setImageBitmap(bitmap)  
}
```

# Loops

```
suspend fun applyImageFilter() {  
    for (i in 0..10) {  
        val bitmap = processBitmap("image$i.png").await()  
        mImageViews[i].setImageBitmap(bitmap)  
    }  
}
```

# Exception handling

```
suspend fun applyImageFilter() {  
    val bitmap = processBitMap("image.png").await()  
    mImageView.setImageBitmap(bitmap)  
}
```

# Exception handling

```
suspend fun applyImageFilter() {  
    try {  
        val bitmap = processBitmap("image.png").await()  
        mImageView.setImageBitmap(bitmap)  
    } catch(t: Throwable) {  
        ...  
    }  
}
```



# Demo

Live coding

# Links

`kotlinx.coroutines`

<https://github.com/Kotlin/kotlinx.coroutines>

Coroutines Design Document (KEEP)

<https://github.com/Kotlin/kotlin-coroutines/blob/master/kotlin-coroutines-informal.md>

Coroutines by example

<https://github.com/Kotlin/kotlinx.coroutines/blob/master/coroutines-guide.md>

Kotlin Coroutine (Experimental) Adapter

<https://github.com/JakeWharton/retrofit2-kotlin-coroutines-adapter>

# Thank you!

Victor Kropp

@kropp

victor.kropp.name

# Questions?

Victor Kropp

@kropp

victor.kropp.name