WorkManager Classes

- Worker
- WorkRequest
- WorkManager

CoroutineWorker class example

```
class BlurWorker(
    ctx: Context,
    params: WorkerParameters
) : CoroutineWorker(ctx, params) {

    override suspend fun doWork(): Result {
        // Long running background task
    }
}
```

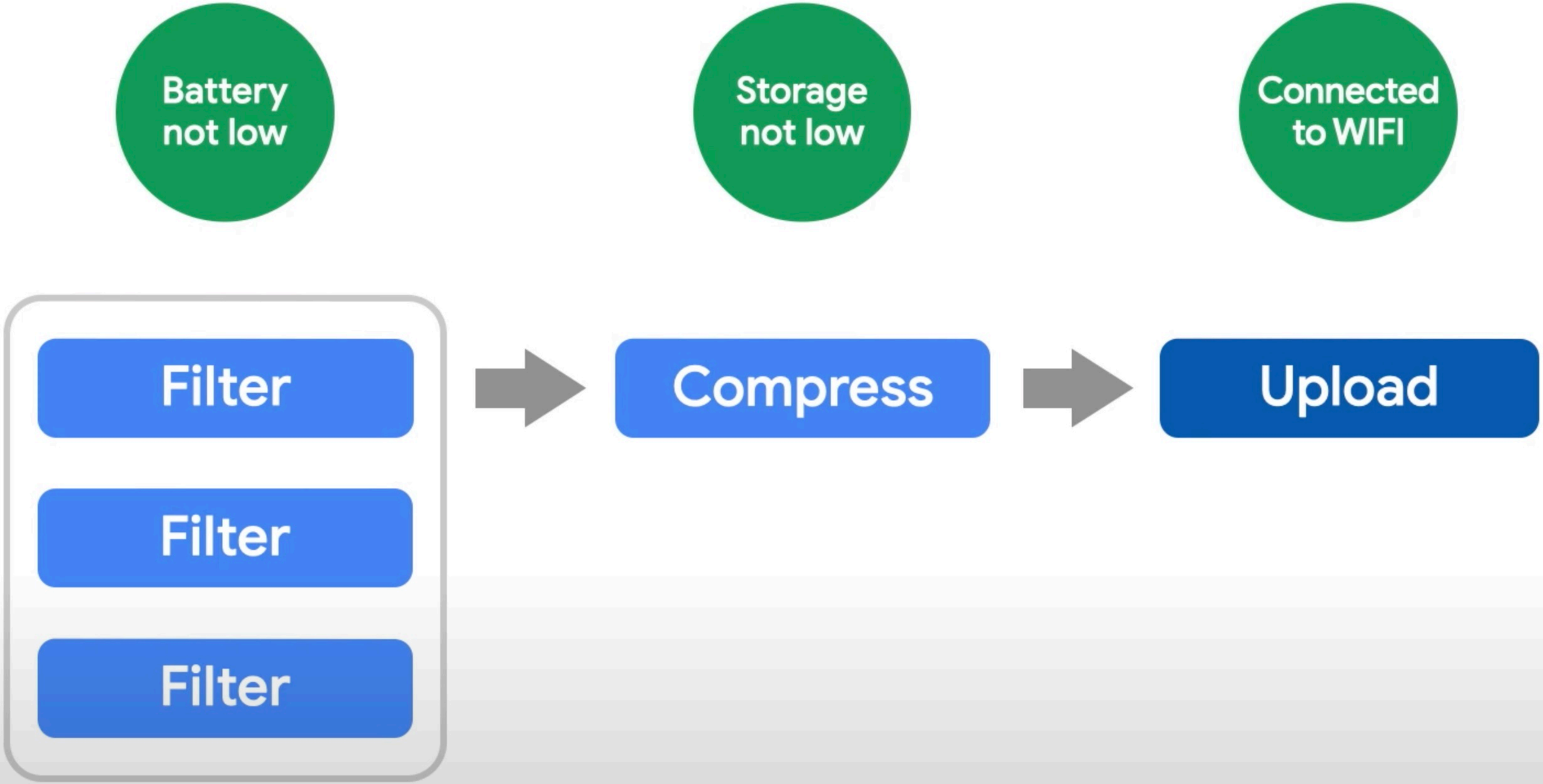
WorkRequest class example

```
// Create charging constraint
val chargingConstraint = Constraints.Builder()
    .setRequiresCharging(true)
    .build()

// Create a WorkRequest to blur the image
val blurWorkRequest = OneTimeWorkRequestBuilder<BlurWorker>()
    .setConstraints(chargingConstraint)
    .build()
workManager.enqueue(blurWorkRequest)
```

WorkRequest class example

```
WorkManager wm = WorkManager.getInstance(Context)
wm.enqueue(OneTimeWorkRequest.from(BlurWorker::class.java))
```


Battery not low
Storage not low
Connected to WIFI
Filter
Filter
Filter
Compress
Upload

Chaining Workers sample code

```
WorkManager.getInstance(myContext)
    // Candidates to run in parallel
    .beginWith(listOf(imageFilter1, imageFilter2, imageFilter3))
    // Dependent work (only runs after all previous work in chain)
    .then(compress)
    .then(upload)
    // Call enqueue to kick things off
    .enqueue()
```

WorkManager Summary

- Observe work request status
- Backoff policy
- Periodic WorkRequest
- Tagged work