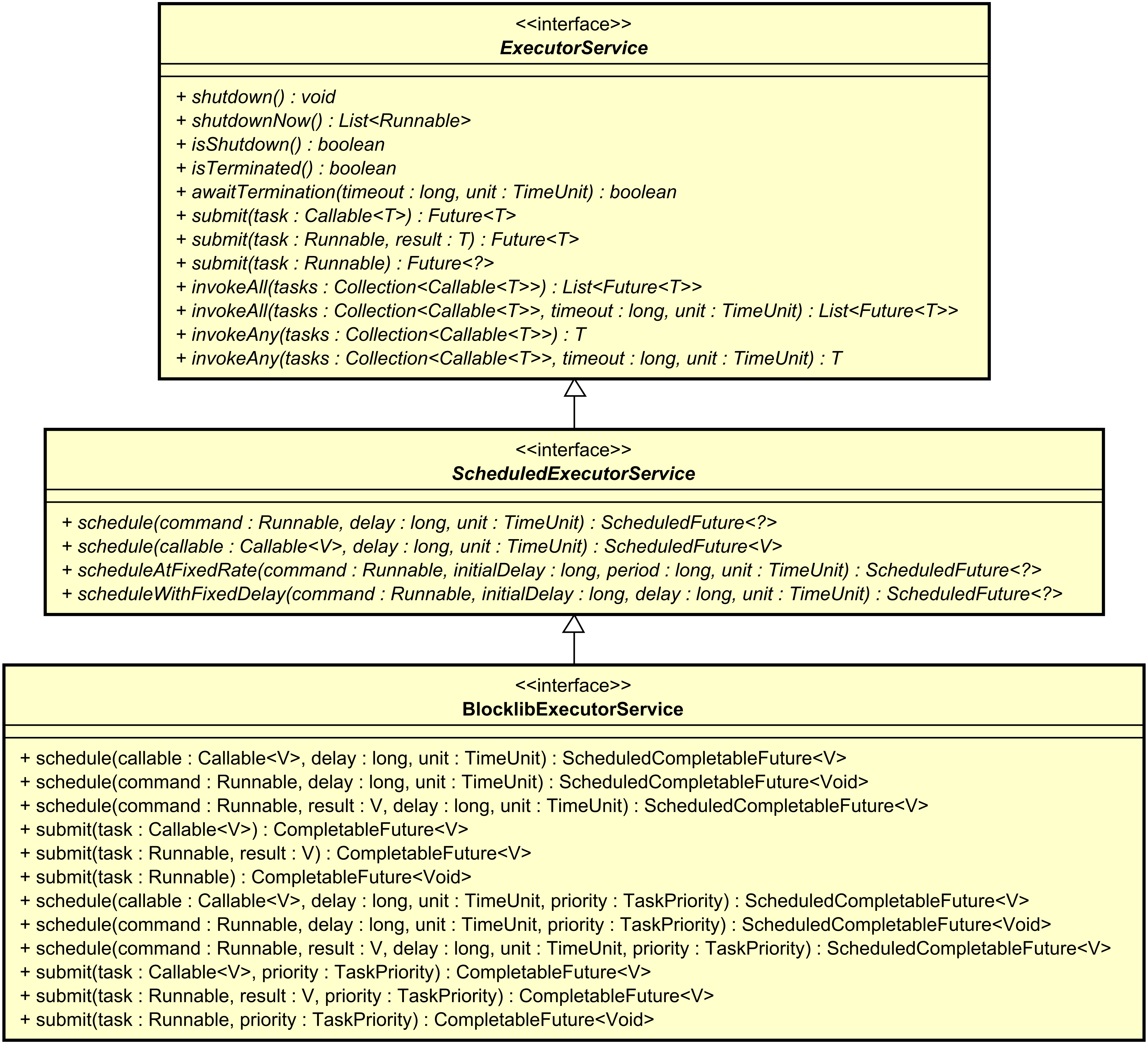

<<interface>>
ExecutorService
+ *shutdown() : void*
+ *shutdownNow() : List<Runnable>*
+ *isShutdown() : boolean*
+ *isTerminated() : boolean*
+ *awaitTermination(timeout : long, unit : TimeUnit) : boolean*
+ *submit(task : Callable<T>) : Future<T>*
+ *submit(task : Runnable, result : T) : Future<T>*
+ *submit(task : Runnable) : Future<?>*
+ *invokeAll(tasks : Collection<Callable<T>>) : List<Future<T>>*
+ *invokeAll(tasks : Collection<Callable<T>>, timeout : long, unit : TimeUnit) : List<Future<T>>*
+ *invokeAny(tasks : Collection<Callable<T>>) : T*
+ *invokeAny(tasks : Collection<Callable<T>>, timeout : long, unit : TimeUnit) : T*
<<interface>>
ScheduledExecutorService
+ *schedule(command : Runnable, delay : long, unit : TimeUnit) : ScheduledFuture<?>*
+ *schedule(callable : Callable<V>, delay : long, unit : TimeUnit) : ScheduledFuture<V>*
+ *scheduleAtFixedRate(command : Runnable, initialDelay : long, period : long, unit : TimeUnit) : ScheduledFuture<?>*
+ *scheduleWithFixedDelay(command : Runnable, initialDelay : long, delay : long, unit : TimeUnit) : ScheduledFuture<?>*
<<interface>>
BlocklibExecutorService
+ schedule(callable : Callable<V>, delay : long, unit : TimeUnit) : ScheduledCompletableFuture<V>
+ schedule(command : Runnable, delay : long, unit : TimeUnit) : ScheduledCompletableFuture<Void>
+ schedule(command : Runnable, result : V, delay : long, unit : TimeUnit) : ScheduledCompletableFuture<V>
+ submit(task : Callable<V>) : CompletableFuture<V>
+ submit(task : Runnable, result : V) : CompletableFuture<V>
+ submit(task : Runnable) : CompletableFuture<Void>
+ schedule(callable : Callable<V>, delay : long, unit : TimeUnit, priority : TaskPriority) : ScheduledCompletableFuture<V>
+ schedule(command : Runnable, delay : long, unit : TimeUnit, priority : TaskPriority) : ScheduledCompletableFuture<Void>
+ schedule(command : Runnable, result : V, delay : long, unit : TimeUnit, priority : TaskPriority) : ScheduledCompletableFuture<V>
+ submit(task : Callable<V>, priority : TaskPriority) : CompletableFuture<V>
+ submit(task : Runnable, result : V, priority : TaskPriority) : CompletableFuture<V>
+ submit(task : Runnable, priority : TaskPriority) : CompletableFuture<Void>