Android Comparision Chart : AsyncTask Vs Service

	AsyncTask	Service
Purpose ?	It enables proper and easy use of the UI thread. This class allows performing background operations and publishing results on the UI thread without having to manipulate threads and/or handlers. It should ideally be used for short operations (a few seconds at the most.)	A service can run in the background to perform work even while the user is in a different application
It is an application component?	No	Yes
How to invoke?	The task instance must be created on the UI thread. execute(Params...) must be invoked on the UI thread. The task can be executed only once (an exception will be thrown if a second execution is attempted.)	Services can be started with Context.startService() and Context.bindService().
How to stop?	A task can be cancelled at any time by invoking `cancel(boolean)` After invoking this method, `onCancelled(Object)`, instead of `onPostExecute(Object)` will be invoked after `doInBackground(Object[])` returns	`stopSelf()` or `stopService()`, the system destroys the service as soon as possible.
Is a thread?	Yes	No
Runs on UI Thread?	No	Yes
Whether other application can access?	No	Yes, Global access to a service can be enforced when it is declared in its manifest's tag.
How task execution happens?	The task can be executed only once (an exception will be thrown if a second execution is attempted.)	If this service is not already running, it will be instantiated and started (creating a process for it if needed); if it is running then it remains running.