A few years back, I played a game called "**Watch_Dogs**" where the main character "**Aiden Pearce**" is part of a "**hacker**" group working to free the city of criminals and punish the ones responsible for the death of his sister's daughter. He uses his mobile to hack into cameras, traffic lights, fork lifts, steam pipes, power grid, bridges and much more. I really loved this idea, and technologically it all seems feasible but is indeed an engineering challenge to make such a system simple enough.

There exist a number of the Internet of Things platforms, and I must admit my limited knowledge, I do think these systems are currently centralized (running on AWS / Google). My objective is to build a platform that enable "**peers**" to share data and enable "**distributed**" code execution. My hope is to eventually cater to use cases such as AI based border patrol, organic agriculture and waste minimization through "**distributed computing**". Currently I aim to develop a "**device manager**" that enables any Linux machine, such as "**high-end server**", or "**raspberry pi**" to act as a code execution platform. Once a single device can be managed through a standard HTTP REST API, the next stage can to enable devices to easily "**look for**" and communicate with each other.

Currently, it is possible to use the "**device manager**" to deploy any "**python**", "**node**" code, or serve a "**tensorflow model**" (for AI). Once deployed, it can be managed and any number of instances (processes) can be executed. It currently supports monitoring and logging, file system downloads and a few other functions all though a REST API. A dashboard style UI design is currently ongoing.