OpenC2 Orchestration and Worker Implementation

https://github.com/StephenOTT/OpenC2-Orchestrator https://github.com/StephenOTT/OpenC2-Worker

Services are similar to OpenC2 Actuators, but with a additional abstraction: Services live within a Worker, but a Service can perform many different types of Actions depending on its configurations. A Worker is the equivalent to a Actuator, but a Worker has the additional benefit of more flexible use cases. OpenC2-Services Service Catalogue Pull Services to Send into Workers Workflows are the steps that the Orchestrator will follow. Steps may be sequential or parallel commands and data to sent to Workers, waiting for responses from Workers, dealing with incidents and errors, or to perform other interactions internal to the Orchestrator or with other systems (such as reporting) **22-Workflows** Workflow Catalogue Pull Shareable Workflows **Open(** Orchestrator implements a TCP Server OpenC2-Orchestrator allowing connections with one or more Workers. Orchestrator implements a BPM Engine, Orchestrator allowing orchestration to benefit from BPMN Workflows (Shareable Workflows). Orchestrator can push C2 commands and can push new Services to the Worker all in real-time. Workers do not need to be re-deployed. The Worker can utilize the Service as a persistent Service or the Worker can delete the service upon task or workflow completion **TCP Connections** Network Server Device OpenC2-Worker Worker Worker Worker Worker (installed on a existing (installed on a device) (as a Server) (IoT, Desktop, etc) server) Workers can be scaled and run in High Assurance / Clusters as needed Multiple Workers can be run within the same context, and allow eventbus communications. This allows further refinement of sub-orchestrators within a specific context using only workers

A Worker implements a TCP Client allowing connection to one or more Orchestrators to receive commands from.

TCP Client can be swapped out for any other Message broker such as a Event Bus, Message Queue, HTTP, etc

Message contents to/from Orchestrator and Worker is JSON payload. It is up to the orchestrator, worker, and workflows to define exact data needs within the message.