Analysis of Alternatives

Virtualization

- Xen
  - Only x86 virtualization (requires additional simulator processes in each VM)
  - High overhead
- Docker
  - Sufficient features
  - Low overhead
    - Run-time overhead is barely higher than that of a "bare metal" Linux process.
- Bare metal simulator processes
  - Lacks orchestration features
  - Not feasible for some devices
- Mixed solutions
  - Possible (but likely has little advantage over Docker)

Gateway/network

- Xen virtual networks
  - Only useful in a Xen environment
- Docker
  - Built-in network virtualization
  - Should be easy to build a virtual gateway container
- VyOS (gateway)
  - Probably requires a full x86 VM, which means we'd need more than Docker.
- Raw Linux bridge interfaces
  - More effort to manage

PLC

- OpenPLC
  - Only major open-source option
- Proprietary software
  - Probably not worth the cost

Control device

- Xen VM
  - Closest to "real hardware" user experience but higher overhead
- Docker image
  - Lower overhead
  - Probably sufficient features

Relay simulator

- PLC program
  - "Piggybacks" on another implemented device (saves effort)
- C++ program
  - Probably more flexible and efficient

Transition to AWS

Overall structure

We can keep using Docker containers, but since our container repo won't be on the local network, we'll need to give more thought to the size of container images, which is largely dependent on the Linux distribution being used as the base.

- Debian
  - Provides a very standard userspace and good package selection
  - Relatively large images (~100MB)
- Alpine Linux
  - Much smaller images (can be under 10MB)
  - Potentially smaller package selection

Conclusion: We'll probably use Alpine for most images and use two-stage builds to keep our build tools out of the final images.

Networking

The AWS container-focused services (i.e. ECS) would *mostly* run our containers just fine, but networking configuration would be tricky because there's no simple way to add our own bridged container network to ECS.

- Set up a tool like Weave Net on ECS
  - Lets us use the ECS Web interface
    - Nice for single containers but tedious for numerous containers running on distinct hosts
  - Increases the complexity of our ECS setup (can't use "stock" ECS machine images)
- Run Docker in swarm mode on plain EC2 VMs
  - More control
  - More manual setup
    - Can be automated with a script and/or custom machine images
    - Easier to migrate to a standalone deployment

Conclusion: It seems best to use "plain" EC2 virtual machines and manage Docker ourselves instead of relying on the ECS interface.