Small Workplace Automation & Remote Monitoring (SWARM) IT Module

- SWARM is program designed to add control and visibility to small and isolated building HVAC systems that are not connected to the central building automation system. This includes offices, classrooms, athletics spaces, and some research spaces. The scope does not include BSL2+ or other spaces with high safety concerns regarding the HVAC system.

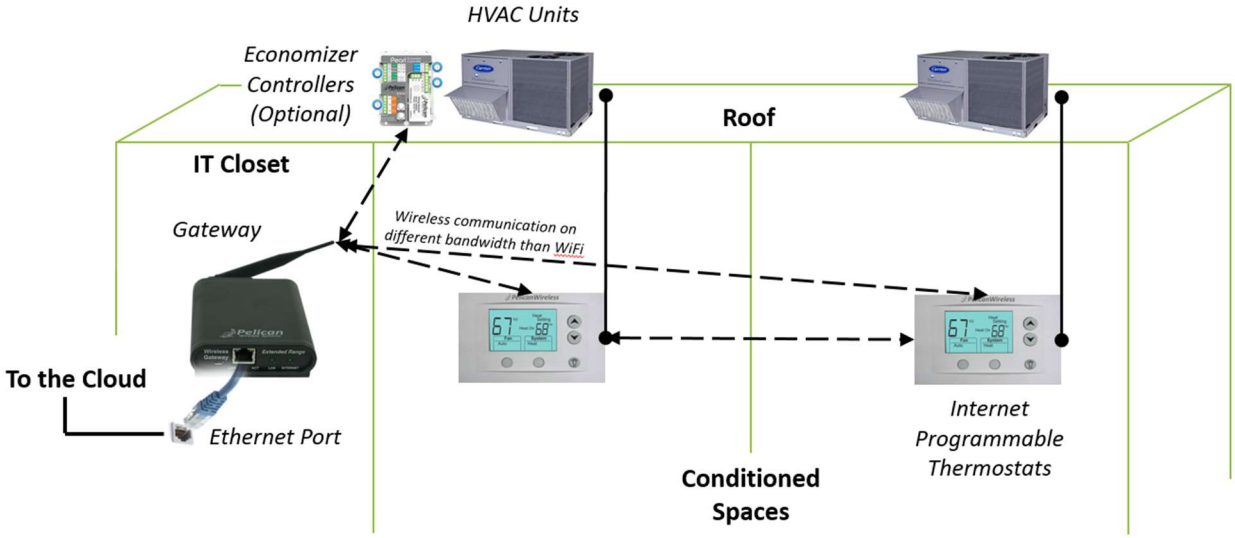

Figure 1: Physical Layout of SWARM

- SWARM requires a 1-to-1 replacement of old thermostats with internet programmable thermostats. The thermostats and associated web client must have the following capabilities:
- **IT capabilities:**
  - Ability to connect to a centralized web-based scheduling interface
  - Ability to assign building operators remote access to individual thermostat controls
  - Ability to assign permission levels to different user types (building occupant, operator, administrator)
  - Use web API capabilities to pull historical temperature and operating data into on-campus data systems
  - Ability to group thermostats by cluster/building
  - If thermostats use wireless communications, they should not rely on WiFi and should use a different frequency band than WiFi. Web-enabled thermostats cannot typically connect to the eduroam network, which requires a username & password.

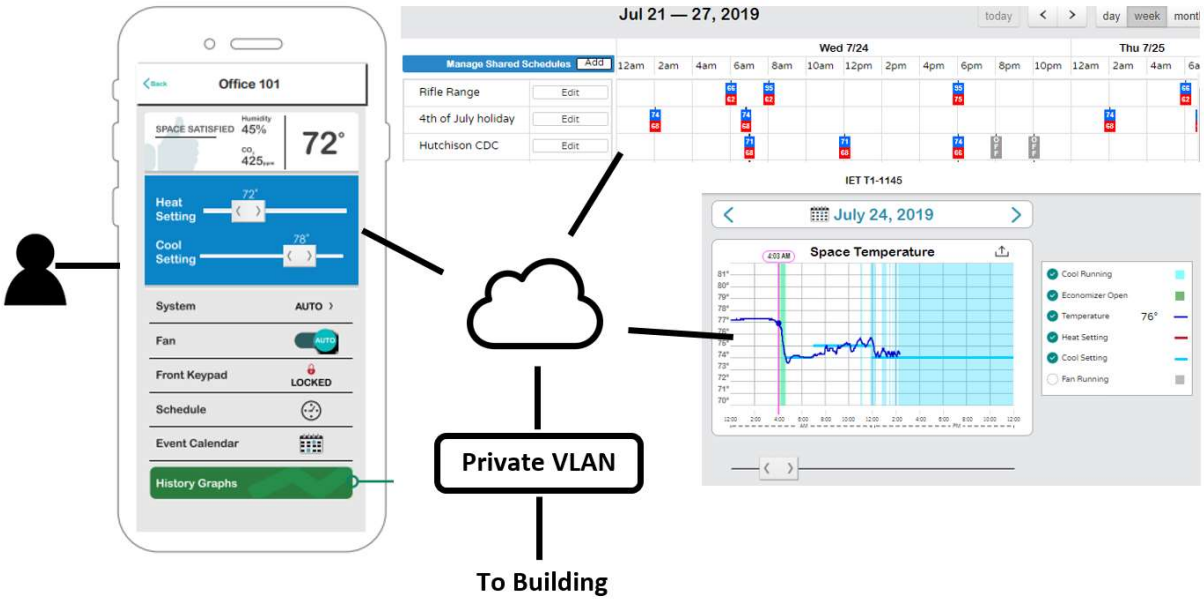

Figure 2: Virtual Layout of SWARM

Case study – IT security for the SWARM project at UC Davis

- To ensure that SWARM complies with UC Davis campus security requirements, the SWARM thermostats at UC Davis are configured in the following way:
    - Individual thermostats are not assigned an IP address and communicate to the web via one gateway per building or per cluster of buildings. To communicate with the gateway, thermostats use a wireless mesh network on a frequency band that does not overlap with the Wi-Fi frequency band.
    - Gateway uses a wired Ethernet network connection. Cellular connections are acceptable for temporary applications.
    - Proxy server for the gateway-to-web server connection
    - Wireless mesh network (not Wi-Fi bandwidth) for thermostat-to-gateway connection
    - Thermostats can ping server to ask for changes (avoids outside system making changes on campus network)
    - Ability to set static IP address for gateways
    - Gateway shall be communicating out to a fixed external IP address so that outbound traffic can be limited to that IP address only for security purposes. IP address ranges (e.g. from AWS) are not acceptable.
    - The gateway should be in a locked enclosure. In most cases, it will also be in an IT or electrical closet to avoid being tampered with by the public.
    - When in a more public space such as a lecture hall, thermostats should be enclosed and locked.
- UC Davis uses Pelican Wireless technology, which meets our campus security requirements. Lowell Valiant, UC Davis Systems Architect, and the UC Davis SWARM team made the following notes about Pelican Wireless's technology and security:

- Pelican uses IEEE 802.15.4 for the mesh network. Low traffic, narrow range. Does not interfere with the Wi-Fi network (different frequency range).
- The gateway (GW400) acts as the 'transfer point' between the mesh network and the cloud server.
- For each client site, Pelican sets up a virtualized, dedicated server (in the cloud) to which all the gateways from the site will connect.
- Pelican will provide 1 IP address and 1 port number. This prevents the gateways from communicating to other sources if they are compromised.
- Cloud instances are hosted in SOC 2 data centers. There are 4 data centers across the country. Pelican's IT provider maintains the hardware; the hardware is dedicated to Pelican, Pelican administers the servers.
- Traffic coming out of the Pelican gateway is not web traffic, only TCP. It is possible to set up a proxy to monitor that TCP traffic.
- 2 types of traffic:
    - Data logging going from the gateway to the cloud server - raw data stream, unintelligible - AES-encrypted
    - Commands from the server (schedule, setpoint, etc) to the thermostats - AES-encrypted also – use a shared key that exists on the device.
- Each instance has its own key server. Keys are generated dynamically for a server and rotated. Keys are changing and are specific to the virtual server, not shared across multiple servers. The gateway forwards the traffic and only worries about TCP security. The thermostat handles the decryption. This is basically end-to-end encryption.
- Private networks for gateway and source-netting: not a problem at all for Pelican. All the gateway cares about is that its packets get delivered to the server.
- The instance has a unique URL which is used for Pelican's GUI web app, which can also be used for doing Restful API. All of that access is https.
- All devices are freely upgraded and updated. Updates are pushed to the gateways. Every customer runs the latest software. Critical bugs are pushed immediately, new features can be pushed on request.
- Pelican does not use OpenSSL, so they are not vulnerable to that. AES-standard encryption. Not running Linux, not open source. Thin architecture. Servers are linux-based, they run SSL and stay up to date as far as kernels and such. Devices on site run Pelican's own software. Nobody can log on to these devices.