Blue Light Hero: Security Architecture

Control Board Firmware, Network Transport, and Cloud Platform

Pedestal Pro

Draft v0.9 — for internal review

Table of Contents

About this document

This paper describes how the Pedestal Pro Blue Light Hero (BLH) system is built, how a BLH station behaves once it is connected to a customer's network, and what testing and controls stand behind it. It is written for the network and security teams who are asked to approve a BLH installation.

The first section is a summary for facilities, procurement, and security leadership. The sections that follow are written for the engineer who will actually evaluate the device. An appendix contains a pre-install checklist your network team can act on directly.

Questions this document does not answer, or answers in a way you would like to verify independently, should go to **bluelight@pedestalpro.com**. We will respond to any specific technical question about the system.

Part 1 — Executive Summary

What a Blue Light Hero station is

A BLH station is an emergency call station: a physical blue light tower or wall station that a person in distress can activate to summon help. Pedestal Pro has added a control board to that station so it can be monitored and managed remotely — so that a campus safety office

knows a station is online and working *before* someone needs it, rather than finding out afterward.

The control board is not a computer in the ordinary sense. It is a pair of small microcontrollers with no operating system, no user shell, no file system exposed to the network, and no ability to run arbitrary software. It does one job: report station status upward and accept a narrow set of configuration and command messages in return.

The control board is not part of the emergency call. When a person presses the call button, the call is placed by the station's own emergency communication equipment. It does not pass through the control board, the network module, the Pedestal Pro platform, or the internet connection described anywhere in this document. These are separate systems that happen to share an enclosure. The control board can report that an alarm occurred; it cannot place, route, intercept, record, or interrupt the call itself. Section 2.5 states this in full.

The question this document answers

Universities and large institutions do not install unknown network devices without review. When Pedestal Pro submitted a Rev 3 board to a university partner, that review took long enough to delay the installation. That is a reasonable thing for a customer to do, and this paper exists to make it faster — by putting the information a reviewer needs in one place, up front, instead of requiring them to discover it through bench testing.

Five things a network reviewer should know

1. The station makes one outbound connection and accepts none. The board opens a single outbound TCP connection to the Pedestal Pro cloud platform. It runs no server, listens on no port, and cannot be reached from your network or the public internet. Nothing on your network can connect *to* it, which removes the most common class of IoT compromise.

2. It does not talk to anything else on your network. The board does not scan, discover, broadcast, or route. Apart from DHCP and DNS at startup, it exchanges traffic with exactly one destination. It cannot be used as a pivot into your network because it has no mechanism for reaching anything on it.

3. Cellular removes the question entirely. BLH stations can be configured with a cellular module instead of an Ethernet module. In that configuration the station never touches your network at all — no cable, no IP address, no firewall rule, no IT ticket. Section 5 covers this option.

4. It carries very little data, and none of it is personal. Measured cellular usage is approximately 2 MB per station per day. The payload is station status, diagnostics, alarm events, and lighting configuration. The board does not capture, store, or transmit audio, video, or any information about individuals.

5. It has been tested by people other than us, and we will show you the results. The management platform underwent a two-week manual penetration test by a three-person

team from Cobalt in early 2026, against the production application and API, using credentialed accounts at three privilege levels. Six findings were reported — one high, two medium, three low, none critical. **All six were remediated and verified fixed**, and Cobalt's final report records every one as resolved. Section 8.1 gives the full breakdown, including the categories that were tested and produced no findings. The board separately carries UL listing for electrical and enclosure safety. Dependencies are continuously scanned and the codebase carries an automated test suite.

What we are not claiming

We think it is more useful to be exact about this than to imply more than we have.

The UL listing that appears on the enclosure covers electrical and environmental safety — that the unit is watertight and will not become a fire or shock hazard. **It is not a cybersecurity certification and we do not present it as one.**

Pedestal Pro has not pursued UL 2900 or IEC 62443 certification for the control board. Those programs are expensive and slow, and on review their penetration-testing requirements are largely oriented toward whether the device itself keeps functioning under attack rather than whether the device endangers the network it sits on — which is the question our customers actually ask. We concluded that a specific, verifiable description of the device's network behavior, backed by independent testing of the platform, answers that question better than a certificate would. If a certification is a hard procurement requirement for your institution, tell us; that changes our calculus and we would like to know.

What we ask of a reviewing IT team

Read Appendix A. It lists exactly what the station needs from your network — an IP address and one outbound destination — and what it will never ask for. In most cases that is the whole integration. If your environment blocks all outbound traffic by default, Appendix A tells your firewall team precisely what to allow and why.

Part 2 — Technical Detail

1. System architecture

The system has four layers. Each boundary between them is authenticated.

Layer	Component	Role
Station	Control PIC microcontroller	Drives lights, alarm, sensors, real-time clock, physical call button
Station	MQTT PIC	Owns all network communication; orchestrates

Layer	Component	Role
	microcontroller	firmware updates for both processors
Station	Network module	Wired Ethernet module *or* cellular module (mutually exclusive; same board socket)
Transport	Outbound MQTT session	Single persistent connection from station to IoT platform
Cloud	ThingsBoard IoT platform	Device registry, per-device credentials, telemetry store, OTA package distribution
Cloud	Pedestal Pro management application	Web application at the Pedestal Pro platform URL: users, roles, scheduling, alarms, reporting
Client	Browser	HTTPS to the management application only; never speaks to a station directly

Two properties of this shape matter for a security review:

The station has exactly one peer. It does not accept commands from a browser, from the customer's network, or from any address other than its configured platform endpoint over its own authenticated session.

A user's browser never reaches the station. An operator pressing "trigger test alarm" in the web interface is writing to the cloud platform; the station picks that up on its next poll. There is no path from a workstation on your network to a control board, which means a compromised workstation cannot command a station directly.

2. The control board on your network

2.1 What the hardware is

The board carries two 8-bit PIC microcontrollers with on the order of tens of kilobytes of program flash and roughly 16 KB of RAM. There is no general-purpose operating system, no Linux, no shell, no package manager, no USB host stack, no removable media, and no embedded web interface or configuration portal. The firmware is a fixed compiled image; there is no interpreter and no facility for loading or executing code delivered by any means other than the signed-version update path described in Section 4.

This is worth stating plainly because most published IoT security incidents depend on capabilities this device does not have: a default-credentialed web admin page, a telnet or SSH service, a busybox shell, or an exposed UPnP or ONVIF stack. None of these exist on the board.

2.2 Network behavior

Once powered and connected, a station:

- obtains an address by DHCP, or uses a static configuration if you prefer — IP address, subnet mask, gateway, and DNS server can all be assigned;
- resolves the platform hostname by DNS;
- opens one outbound TCP connection to the IoT platform endpoint and keeps it open;
- publishes telemetry and subscribes for configuration and command messages on that connection;
- re-establishes that connection if it drops.

It does not:

- listen on any TCP or UDP port, or run any service;
- accept inbound connections from any source;
- perform network scanning, host discovery, or port probing;
- issue broadcast or multicast traffic beyond what DHCP requires;
- act as a router, bridge, proxy, or relay between network segments;
- connect to any host other than its configured platform endpoint;
- store, request, or transmit customer/user credentials or directory information.

Because the connection is outbound and originated by the station, a standard stateful firewall requires no inbound rule and no port forwarding. In a default-deny outbound environment, one egress rule is required; Appendix A specifies it.

2.3 Identification and inventory

Every board is identified and labeled during manufacture, test, and burn-in, so a unit on your network can be traced to a specific build.

Wired stations. The Ethernet module carries a factory-applied MAC address on a white label. All addresses begin with the prefix `00:08:DC`, the module vendor's registered OUI, which means BLH stations are straightforward to recognize and group in your network inventory. A green dot applied next to the MAC label indicates the module has been programmed with its configuration and tested end-to-end against the IoT platform before shipment.

Cellular stations. The cellular module has no MAC address; it is identified by the IMEI embedded in its firmware. The 15-digit IMEI is printed on a label above the SIM holder, with the SIM's ICCID beneath it, and a green dot on the SIM holder indicates the module has been configured and connection-tested.

Both. A label on the back of the terminal block carries a Julian lot code (`yyddd` — two-digit year plus day of year) and an internal reference tying the unit to its build record. That record holds the firmware version installed on each of the two processors at manufacture.

MAC addresses can be supplied in advance of shipment if your network uses MAC-based authorization or a NAC registration process. Ask us and we will send them with the shipping notification.

802.1X is not supported. The network module does not act as an 802.1X supplicant, so a BLH station cannot authenticate to a port that requires it. If your access layer enforces 802.1X, a station needs either a MAC Authentication Bypass exception or a port configured outside the 802.1X policy — or, if neither is acceptable to your team, the cellular configuration avoids the question entirely by not using your access layer at all.

2.4 Bandwidth and load

Measured usage on a station running normal status reporting is approximately **2 MB per day**, on a reporting cadence of roughly 15–30 seconds. The station is not a meaningful consumer of bandwidth and does not generate bursty traffic; firmware updates, which are infrequent, are the only exception and are chunked to accommodate the microcontroller's limited RAM.

2.5 The emergency call path is separate from the control board

This is the single most important thing to understand about the product, and it is stated here without qualification.

The control board does not interface with the emergency call function, and an emergency call does not pass through the control board. They are two separate systems that share an enclosure.

When a person presses the call button on a BLH station, the call is placed by the station's own emergency communication equipment, over that equipment's own path, to campus police, EMS, or whichever responders the site has configured — exactly as legacy units do. The call does not traverse the control board, the Ethernet or cellular module, the Pedestal Pro platform, the internet connection described in this document, or any customer network.

It follows that the control board:

- cannot place, route, intercept, record, delay, or interrupt an emergency call;
- cannot prevent a call from being placed, whatever state the board is in;
- is not a dependency of the call function in any mode of operation.

A control board that is powered off, disconnected from the network, misconfigured, or physically removed has **no effect** on the station's ability to summon help. The board's role is to report that an alarm occurred so that the event appears in monitoring and history, and to let staff verify remotely that a station is healthy before someone needs it. That is the entirety of its relationship to the emergency function.

2.6 Behavior on failure

If the network connection is lost, the station continues to operate locally, and — per Section 2.5 — the emergency call function is unaffected regardless. Remote monitoring degrades; the safety function does not. A station that loses connectivity is shown as offline in the management platform, which is itself a monitored condition.

3. Wired Ethernet deployment

In the wired configuration the board uses a hardware Ethernet controller module. The module provides the TCP/IP stack; it is a fixed-function part, not a general-purpose computer, and it exposes no management interface of its own.

The module vendor does not publish a security white paper or equivalent statement for the part, so there is no third-party document for us to cite here. The behavior described in Section 2.2 is what we have verified ourselves and what you can verify with a packet capture on the station's port — which we encourage, because it is the fastest way to confirm everything this document claims about network behavior.

The relevant network facts are those in Section 2.2. From the customer's perspective, the station occupies one address and one outbound flow.

4. Firmware and updates

4.1 Two processors, one update path

The Control PIC drives station hardware. The MQTT PIC owns the network. The MQTT PIC always updates first, because it is the processor that performs downloads; once it is current, it checks whether the Control PIC also needs an update and, if so, retrieves and stages that image as well.

4.2 How an update reaches a station

1. A Pedestal Pro administrator uploads a compiled firmware image through the management application. This route is restricted to Pedestal Pro administrators, enforced by both route middleware and an authorization policy, and is rate-limited.
2. The application computes a SHA-256 hash of the image and registers it with the IoT platform as an OTA package, which validates the hash on upload.
3. Version information is published to the station as configuration data.
4. The MQTT PIC reads that version information on its normal polling cycle (roughly every ten minutes), compares it against its own compiled version, and — only on a mismatch — initiates a download.
5. The image is retrieved in chunks **over the station's existing outbound connection**. No new port is opened, no inbound connection is accepted, and the station does not connect to any additional host.
6. The image is written to external EEPROM. The processor reboots into its bootloader, which verifies a CRC16 supplied by the compiler build before committing the image to program flash.
7. If the update is interrupted — power loss, connectivity loss — the CRC check fails and the bootloader retries on the next boot rather than running a partial image.

Two independent integrity checks are therefore applied: a SHA-256 hash at the platform layer, verifying the file in transit and at rest, and a hardware-specific CRC16 at the bootloader layer, verifying what actually lands in flash.

4.3 A design decision worth surfacing

An earlier design had stations connect to a dedicated TCP daemon on a separate port to fetch firmware. That design was abandoned in favor of the existing outbound session. One of the stated reasons was that opening an additional listening port was flagged as a security concern during penetration testing. We mention it because it illustrates how the constraint has been applied in practice: the device's outbound-only posture was preserved even where doing so meant rebuilding a working subsystem.

4.4 What the update path cannot do

A station will not accept firmware from an arbitrary source. It has no update mechanism reachable from the local network, no USB or serial update path exposed in the field, and no ability to be redirected to a different update host by anything on the customer's network.

On image validation, to be precise: the bootloader downloads the image into external memory and verifies it against a CRC published alongside it. If the computed CRC does not match, the image is never written to the microcontroller's program flash and the processor continues running its existing firmware. This protects against a corrupted or truncated download, and it means an interrupted update can never leave a station running a partial image.

It is not a cryptographic signature, and we do not describe it as one. The microcontrollers in this hardware revision cannot verify a signed image without an additional secure-element chip and a revised bootloader. What establishes that an image is genuine is therefore the update path itself: firmware is published only by a Pedestal Pro administrator through the authenticated management platform, distributed only by the IoT platform, and retrieved by the station only over its own authenticated session. The CRC confirms the image arrived intact; the authenticated path is what establishes it came from us.

A reader who has also read Section 6.2 will see the consequence: for a wired station, an attacker in a privileged position on the network path between the station and the internet is the case these controls are weakest against. That is one more reason we recommend segment isolation for wired deployments, and it is one more thing the cellular configuration removes entirely.

Firmware signing is on the roadmap for a hardware revision that can support it.

5. Cellular deployment: removing the network question

A BLH station can be configured with a cellular module in place of the Ethernet module. It is a direct substitution on the same board — the same socket, the same firmware, and, in field testing, a connection established immediately on power-up with no configuration.

For a customer's IT organization, this configuration means:

- the station is **never connected to the customer network**, at any point;
- no IP address, DHCP reservation, VLAN assignment, or firewall rule is required;
- no network security review of the device is required, because the device is not on the network to be reviewed;
- no Ethernet cabling has to be run to the station location, which for outdoor towers can be a significant part of the installation work.

This is the strongest available answer to "can this device compromise our network," because the device has no adjacency to it. The station is isolated from customer infrastructure by construction rather than by configuration.

The cellular module also integrates GPS, which the station uses for location reporting.

Signal availability is not a practical constraint for this product, because blue light stations are installed where people are — campuses, parking structures, transit areas — which is where cellular coverage exists.

> **Editorial note for the draft:** Jack has prepared a separate paper on cellular security and the air-gap argument, drawing on prior deployments where a customer migrated an installed base entirely to cellular for network-security reasons. That material should be merged into this section, or published alongside this paper and cross-referenced here. This section is deliberately short pending that merge.

6. Station-to-cloud communication

6.1 Credentials

Each station is provisioned through the IoT platform's device provisioning flow and issued its **own** access token. Tokens are per-device: there is no shared secret across the fleet, and compromise of one station's credential does not yield access to another station or to another customer's stations.

Device credentials are stored encrypted at rest in the management application's database using Laravel's encrypted attribute casting, keyed by the application encryption key. They are not written to logs and are not exposed through the user interface or API.

6.2 Transport

The management application communicates with the IoT platform over HTTPS with certificate validation enabled, using short-lived access tokens with automatic refresh; long-lived static tokens are explicitly deprecated in the codebase in favor of this flow. Browser traffic to the management application is HTTPS only, with HSTS asserted.

The station-to-platform link is different, and we would rather you read it here than discover it with a packet capture.

The control board connects to the IoT platform using MQTT on port 1883. That connection is not encrypted. The 8-bit microcontroller and Ethernet module in the current hardware revision cannot terminate a TLS session; adding TLS would require either different hardware on the board or a TLS-terminating service between the station and the platform. Neither is in the shipping product.

What this does and does not mean:

What is exposed. An attacker positioned to observe traffic on the path between a station and the internet can read that station's telemetry — its online status, supply voltage, temperature, alarm events — and can read the per-device access token presented when the session opens. With that token, they could publish false telemetry for that one station or receive the configuration messages sent to it.

What is not exposed. The payload contains no personal data, no customer credentials, no directory information, and nothing about any other device on the network. The token is scoped to a single station: it grants nothing on any other station, on any other customer's stations, or in the management application. And — most importantly — **the emergency call function is entirely unaffected.** Per Section 2.5, calls do not pass through the control board, so no compromise of this link can block, intercept, record, or interfere with a call for help. The realistic worst case for a compromised station token is a false alarm or false status on that one station, which is visible and correctable from the management platform.

What removes the exposure completely. In the cellular configuration the traffic never touches the customer's network, so there is no position on that network from which to observe it. For a customer who considers unencrypted device telemetry unacceptable — a defensible position — cellular is the answer, and it requires no compromise elsewhere.

What we recommend for wired deployments. Place stations on an isolated VLAN or IoT segment with egress restricted to the platform endpoint, as described in Appendix A. This is good practice for any IoT device, and here it also reduces the set of network positions from which the link could be observed to those an attacker would have to reach first.

TLS on this link is under evaluation for a future hardware revision. We would rather say that plainly than imply protection the current board does not provide.

Production firmware connects to mqtt.thingsboard.cloud on port 1883. That single destination is the whole of the station's outbound network requirement; Appendix A states it in the form your firewall team will want.

6.3 What is sent

Uplink from station to platform: online/offline status, alarm events with type and severity, and diagnostic readings such as supply voltage, current draw, and temperature. Downlink from platform to station: configuration writes to the station's EEPROM configuration space (alarm duration, lighting behavior, brightness, schedules, clock synchronization) and a small set of command messages (trigger test alarm, set lighting, test photocell).

Configuration downlinks are encoded as fixed-format frames addressed to a defined configuration memory map. The address space is bounded and documented internally; the station will not act on addresses outside it. There is no mechanism in this channel for delivering executable code — firmware moves only through the path in Section 4.

6.4 What is not sent

The station does not capture, transmit, or store audio or video. It does not identify individuals. It does not collect information about other devices on the customer's network — it has no capability to observe them. And per Section 2.5, no emergency call content passes through the control board or the platform at any point.

7. The management platform

7.1 Stack

The management application is a Laravel 12 / PHP 8.2 application with a Vue 3 single-page front end, backed by MySQL, integrating with ThingsBoard as the IoT platform.

7.2 Authentication

- **Public registration is disabled.** The system is invitation-only; accounts are created by Pedestal Pro administrators or by a customer's own administrator. There is no self-service signup path to attack.
- Passwords are hashed with bcrypt.
- Login is rate-limited per account and source address, locking out after repeated failures.
- Invitation acceptance and email verification endpoints are separately rate-limited to prevent token enumeration, and verification links are cryptographically signed.
- Deactivated users are rejected at the middleware layer on **every request**, not merely at login, so revoking access takes effect immediately rather than at session expiry. An active administrative impersonation session involving a deactivated account is terminated and returned to the administrator's own session. This control was independently verified during penetration testing (Section 8.1).
- Sessions are server-side and encrypted; session cookies are HttpOnly, SameSite, and marked Secure in production.

7.3 Authorization and tenant isolation

The platform is multi-tenant. Users belong to exactly one customer organization, and devices are owned by an organization. Authorization is enforced in depth:

- **Roles** carry named permission sets (administrator, dispatcher, technician, shift manager, operations manager, viewer, and others).
- **Middleware** rejects requests lacking the specific permission a route requires.
- **Policies** — eight of them, covering devices, users, roles, groups, presets, schedules, delegations, and firmware — are evaluated per object, so authorization depends on the specific record being acted on, not merely on the route.
- **Per-device access** can be granted or withheld for an individual user through an explicit access matrix, narrowing access below what the user's role would otherwise permit.
- **Delegation** allows a user to temporarily grant a subset of their authority to another user, with an expiry and an explicit revocation path.
- **Administrative impersonation** is recorded to a dedicated audit log capturing the administrator, the target user, the action, the source IP address, and a timestamp.

Cross-tenant access is prevented by ownership checks in the policy layer rather than by interface hiding: a request for another organization's device fails authorization regardless of how it is constructed.

This architecture has been adversarially tested. The penetration test described in Section 8.1 was conducted with working credentials at three privilege levels specifically to attempt escalation between them; the authorization weaknesses it identified were corrected and re-verified.

7.4 Application hardening

- Security headers are applied globally: `X-Frame-Options`, `X-Content-Type-Options: nosniff`, `Referrer-Policy`, a restrictive `Permissions-Policy` disabling camera and microphone access, and HSTS on secure connections.
- CSRF protection is enforced across the application.
- Rate limiting is applied granularly — roughly ninety route groups carry explicit limits, tightening to a few requests per minute on the most sensitive operations such as firmware push.
- The inbound webhook endpoint from the IoT platform is authenticated by a shared secret compared in constant time, and returns a service-unavailable response rather than accepting traffic if the secret is unconfigured.
- Secrets are held in environment configuration, not in source control.

8. Testing and assurance

8.1 Third-party penetration testing

The Blue Light Hero platform was penetration tested by **Cobalt**, an independent security testing provider.

Engagement

Provider	Cobalt Labs, Inc.
Target	The Blue Light Hero web application and API, **production environment**
Period	February 19 – March 5, 2026
Team	Three testers — one lead, two pentesters
Method	Primarily **manual** assessment, tool-assisted
Standards	OWASP Top 10, OWASP Application Security Verification Standard (ASVS), OWASP API Security Top 10, CVE databases
Access	Credentialed testing at three privilege levels — super admin, customer admin, and viewer
Report status	Final

Two aspects of this scope are worth drawing out, because they determine how much the result is worth.

It was tested in production, not in a sanitized demo. The assessment ran against the live application, so the result describes the system a customer actually receives.

It was credentialed, not black-box. Testers were given working accounts at three different privilege levels and asked to break out of them. This is the harder test and the more informative one: an unauthenticated scan cannot find a privilege-escalation or tenant-isolation flaw, because it never gets far enough in to try. For a multi-tenant platform where one customer's operators must never reach another customer's stations, this is the assessment that matters.

Results

Cobalt reported six findings:

Severity	Reported	Open	Resolved
Critical	0	0	0
High	1	**0**	1
Medium	2	**0**	2
Low	3	**0**	3
Informational	0	0	0
Total	**6**	**0**	**6**

All six findings were remediated and verified. Cobalt's post-test remediation record lists every finding in a `Fixed` state, with resolution dates between March 4 and March 15, 2026. No finding was left open, deferred, or accepted as residual risk.

The findings clustered in access control, configuration, error handling, and input validation. Each was corrected at the source rather than mitigated at the perimeter, and the corrections are visible in the platform's current authorization architecture as described in Section 7.3 — most directly in the per-object authorization policies and in the middleware that rejects deactivated accounts on every request rather than only at login.

We publish these numbers deliberately. A vendor claiming a penetration test with nothing found is either describing a very narrow test or asking you to take their word for it. A test that finds real issues in real code, and a vendor that fixes all of them and can show you the dates, is the more useful signal.

Areas tested with no findings

Cobalt's coverage report records the following control categories as assessed against ASVS requirements with **no findings**:

Category	Requirements assessed
Authentication	Credential recovery, authenticator lifecycle, password security, general authenticator and one-time-verifier requirements
Session Management	Logout and timeout, cookie-based session management, session binding, defenses against session exploits
Stored Cryptography	Algorithm requirements
Data Protection	Sensitive private data, client-side data protection
File and Resources	Upload, download, storage, execution, and **SSRF protection** requirements
Communications	Communications security requirements
API and Web Service	Generic and RESTful web service verification requirements

Obtaining the report

Cobalt's full report contains the complete finding detail, methodology, and remediation record. Where our agreements allow, we can share it — or a written summary of it — with customers and prospective customers under a mutual non-disclosure agreement. Ask us and we will tell you what we are able to provide for your review.

[RECOMMENDED — add regression tests for the pen test findings. The corrections are in place, but the automated suite does not currently carry a test that specifically fails if the privilege-escalation or inactive-account behavior regresses. Adding those would let this section say the fixes are locked in by CI, which is a stronger claim than "we fixed it," and it protects the fixes through future refactors.]

8.2 Continuous dependency scanning

Third-party dependencies are continuously monitored by Snyk, with upgrade pull requests raised automatically against the repository and a documented scanning policy in the codebase. Vulnerabilities in the open-source supply chain are the most common route into a modern web application; they are tracked here as a standing process rather than a point-in-time audit.

8.3 Automated testing

The codebase carries an automated test suite spanning authentication, password and invitation flows, delegation, device ownership and synchronization, device downlink behavior, configuration frame encoding and decoding, scheduling, and lighting control. Several of these tests exist specifically to lock down security behavior — device ownership on synchronization, and delegation lifecycle among them — so that a regression in an access-control rule fails the build.

8.4 Structured beta testing

Ahead of general release, the system ran a structured beta program against a written test rubric covering every functional area of the platform, with findings tracked in a formal bug and feature tracker and retested after remediation. Field testing was conducted with a university partner on live stations.

8.5 What UL listing does and does not cover

The BLH enclosure carries UL listing. That listing addresses electrical and environmental safety — enclosure integrity, water ingress, and fire and shock hazard. **It makes no statement about cybersecurity**, and we do not present it as doing so. Its presence on the unit is relevant to your facilities and electrical inspectors; it is not relevant to your network security review, and any vendor implying otherwise should be questioned.

9. Shared responsibility

Pedestal Pro is responsible for the security of the control board firmware and its update path, the security of the management platform and its hosting, per-device credential issuance and protection, platform monitoring and patching, dependency management, and notifying customers of security-relevant defects.

The customer is responsible for managing their own user accounts and promptly deactivating departed staff, choosing appropriate roles rather than granting administrator access broadly, protecting the credentials of their own users, physical security of the station, and any network controls they choose to apply to the wired configuration.

Jointly: deciding between wired and cellular deployment. We are glad to make that recommendation with your network team rather than for them.

10. Reporting a security issue

Pedestal Pro welcomes reports of suspected vulnerabilities in the BLH platform or control board and will not pursue action against good-faith researchers who report responsibly.

Report a suspected vulnerability to **bluelight@pedestalpro.com**. We will acknowledge your report within 5 business days.

Appendix A — Pre-install checklist for network teams

This appendix is also published as a standalone one-page document.

If the station is cellular

Nothing is required from your network. The station does not connect to it.

If the station is wired Ethernet

What the station needs

Requirement	Detail
Physical	One standard Ethernet drop at the station location
Addressing	One DHCP lease per station, **or** a static IP, subnet mask, gateway and DNS server if your policy requires it
DNS	Resolution of the IoT platform hostname
Outbound	One outbound TCP connection to the IoT platform, **MQTT on port 1883** — mqtt.thingsboard.cloud
Inbound	None. No port forwarding, no NAT rule, no inbound firewall rule
Bandwidth	Approximately 2 MB per station per day
802.1X	**Not supported.** A station cannot authenticate to a port requiring it — see below

What the station does not need and will not use

- No inbound access of any kind
- No access to internal systems, file shares, directory services, or other hosts
- No customer/user credentials from your directory, and no account on your network
- No static public IP address
- No VPN
- No open ports on the station itself — it runs no services

Recommended controls, if you want them

The station's behavior is narrow enough that you can constrain it tightly without affecting function:

- Place stations on an isolated VLAN or an IoT segment with no route to internal networks. The station never needs to reach anything internal.
- Restrict egress from that segment to the platform endpoint only. The station will not attempt any other destination.
- If your environment uses MAC-based authorization, request MAC addresses in advance of shipment; each board is serialized and its MAC recorded at manufacture.
- Monitor the segment if you wish. Traffic should consist of DHCP, DNS, and a single persistent outbound session per station, and nothing else. Anything else is worth a call to us.

Before installation day, confirm with your network team

1. Is outbound traffic from the intended VLAN permitted by default, or is an explicit egress rule required?
2. **Does the intended switch port require 802.1X?** Stations do not support it. You will need a MAC Authentication Bypass exception or a port outside the 802.1X policy.
3. Is DHCP available on the intended segment, or should we ship stations pre-configured with static addressing?
4. Is there a MAC registration or NAC process that needs addresses in advance? We can supply them with the shipping notification; wired station MACs begin `00:08:DC`.
5. **Is unencrypted device telemetry acceptable on this segment?** Station-to-platform MQTT is not encrypted (Section 6.2). If it is not acceptable, the cellular configuration keeps that traffic off your network entirely.

If the answer to any of these makes wired installation slow or unacceptable, the cellular configuration removes all five questions.

Appendix B — Document control

Field	Value
Version	0.9 — draft for internal review
Prepared by	Taj Calles (software, platform, firmware update path)
Contributing	Jack Heiser (hardware, cellular), Trent McElhaney (network firmware, bootloader)
Review	Gavin Johnson, Bruce Kutsche
Intended publication	Pedestal Pro website, at product release