Atlas
Remoticom Device Management

REMOTICOM

Remoticom Atlas

Atlas.

Atlas is a data-model-driven platform for managing Remoticom device fleets in complex, real-world environments. It is designed for deployments with limited bandwidth, low-power devices, and gateway- or mesh-based connectivity. By defining device capabilities in versioned data models rather than custom code, Atlas provides consistent control and long-term extensibility across evolving fleets. The platform combines efficient, secure, data-optimized device communication with standard web APIs and streaming interfaces, enabling direct integration into customer applications and platforms. The Atlas frontend supports day-to-day operations, provisioning, and troubleshooting for operational teams.

Atlas is designed to manage Remoticom device fleets in environments where connectivity, power, and network topology impose real constraints. Typical deployments involve limited bandwidth, battery-powered devices, and installations where devices are reachable only through gateways or mesh networks. Atlas is built specifically for these conditions, avoiding assumptions of always-on connectivity or direct IP access.

The platform is based on a data-model-driven architecture. Device capabilities are defined in versioned data models that describe available resources, supported operations, validation rules, and access permissions. These models act as the authoritative description of device behavior and are interpreted by the platform at runtime. By separating device behavior from platform logic, Atlas ensures consistent control across heterogeneous fleets and allows new device variants or firmware versions to be introduced without changes to the core system.

Atlas treats gateway- and mesh-connected devices as first-class devices. From an integration and operational perspective, devices behave identically regardless of how they are connected. The platform manages routing, topology awareness, and addressing internally, allowing control logic and integrations to remain stable even as network conditions or gateway assignments change over time.

The web frontend is included to support provisioning, monitoring, troubleshooting, and support workflows. Atlas functions both as a complete operational platform and as an embedded device-management backend.

Integration into customer systems is a primary design goal. Atlas exposes standard web APIs, streaming interfaces, and event-forwarding mechanisms that allow customers to embed device management directly into their own applications, backends, or operational platforms. Long-lived streaming connections reduce connection overhead and support bidirectional interaction, while webhook delivery enables simpler event-driven integrations. Asynchronous command handling ensures that device latency or intermittent connectivity does not block upstream systems.

Access control and accountability are applied consistently across users, devices, and integrations. Atlas supports hierarchical organizations with role-based access control, ensuring that access is scoped correctly while retaining service-provider oversight for support and fleet recovery. All actions and forwarded data are logged and auditable, supporting traceability, compliance, and long-term operational insight.

ADVANTAGES.

- ✓ Data-model-driven architecture
- ✓ Built for constrained deployments
- ✓ Transparent gateway and mesh support
- ✓ Role based access control and accountability
- ✓ Firmware deployment manager
- ✓ Integrated planning manager
- ✓ Advanced observe engine
- ✓ Flexible forwarding endpoints

Data-Model-Driven Architecture

Atlas is built around versioned data models that describe device capabilities, resources, permissions, and validation rules. Device behavior is defined declaratively rather than embedded in device-specific application code, ensuring consistent behavior across heterogeneous fleets. While model- or resource-specific logic can be introduced when necessary, Atlas is designed to minimize such coupling in favor of maintainable, model-driven definitions. This approach allows new device types or firmware versions to be introduced without modifying the core platform, while model versioning enables multiple device generations to coexist safely within the same deployment.

Built for Constrained Deployments

The platform is designed for environments with limited bandwidth, intermittent connectivity, and battery-powered devices. Atlas assumes that devices may be offline, slow to respond, or reachable only periodically. All device interactions are asynchronous by design, preventing device latency from blocking upstream systems and ensuring reliable operation under real-world conditions.

Transparent Gateway and Mesh Support

Atlas ensures that devices operating in Bluetooth Mesh networks and devices with direct connectivity are addressed through the same API and operational workflows. From an integration and operational perspective, devices behave identically regardless of how they are connected. Routing, gateway selection, and mesh topology handling are managed internally by the platform, allowing automation logic and customer integrations to remain stable as network structure changes.

Atlas is designed to operate large-scale Bluetooth Mesh networks as part of a unified device fleet. Individual mesh nodes are managed at the device level, while the platform maintains awareness of gateways, hop relationships, and reachability. Changes in mesh topology, such as node additions, removals, or gateway reassignment, are absorbed automatically without manual reconfiguration.

Bluetooth Mesh networks are treated as self-managing communication layers. Temporary link degradation or node unavailability is handled as a normal operating condition, with commands and state updates routed through available gateways and mesh paths. Semi-automatic provisioning workflows support efficient installation and onboarding while abstracting mesh complexity away from operators and integrators.

RBAC and Accountability

Atlas enforces role-based access control across organizations, projects, devices, and operations. Hierarchical organizations allow multi-tenant deployments while preserving strict isolation. A root service-provider role retains oversight capabilities to support fleets in the field without risking loss of control. All actions, configuration changes, and forwarded data are logged and auditable, enabling traceability, compliance reporting, and forensic analysis.

Firmware Deployment Manager

Atlas provides a centralized firmware deployment manager to control device software updates across fleets. Firmware versions are managed as explicit artifacts with defined compatibility and access rules, allowing updates to be staged and targeted to specific devices or projects. Firmware updates are deployed independently from operational traffic, ensuring that routine device control and telemetry are not disrupted during rollout. This approach supports gradual deployment strategies and simplifies recovery in the event of issues.

Fleet & Project Organization

Atlas supports hierarchical project management to organize devices by site, function, or deployment phase. Projects define operational scope, visibility, and deployment targets, allowing large fleets to be managed safely and efficiently.

Project boundaries integrate with access control and deployment workflows, ensuring that planning changes, firmware updates, and operational actions are applied only to the intended subset of devices.

Planning Manager

Atlas includes an integrated planning manager used to define scenes and schedules, Planning artifacts are versioned, validated, and deployed through the same control pipeline as other device operations. This ensures that automation logic is consistent, auditable, and repeatable across large fleets.

Planning is treated as a first-class configuration object rather than ad-hoc device state. This allows operators to stage changes, track revisions, and roll out behavioral updates in a controlled manner without manual device interaction.

Observability & Telemetry

Atlas includes a built-in observe engine that enables devices to publish telemetry and state changes using device-defined subscriptions. Instead of polling, devices push updates based on configured observe definitions, reducing bandwidth usage and backend load.

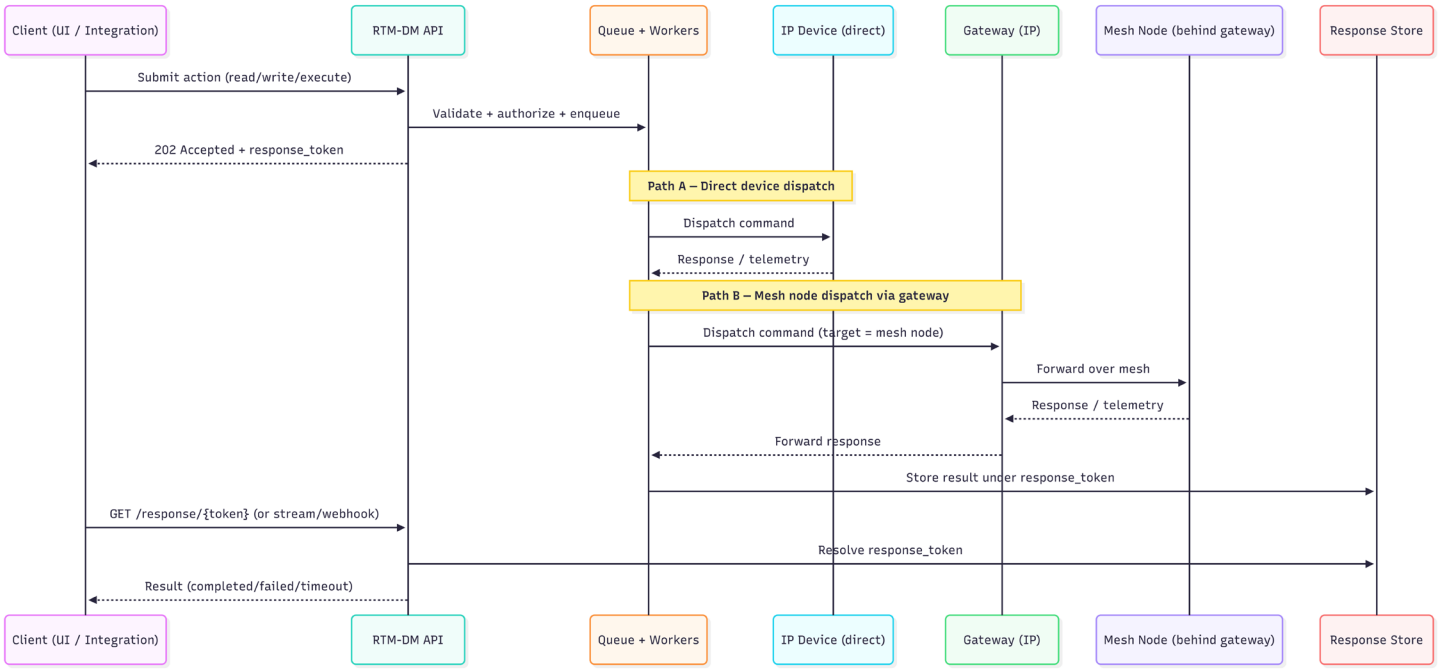
Observations can be forwarded in real time, selectively persisted, or both, depending on configuration. This allows Atlas to support monitoring, alerting, and analytics use cases without enforcing a single data retention model.

Integration & Data Forwarding

Atlas supports multiple forwarding mechanisms to integrate device data into customer systems, including webhooks, WebSocket connections, HTTP/2 streaming, and Server-Sent Events (SSE). This allows customers to choose the integration pattern that best fits their architecture and operational needs. Long-lived streaming connections are preferred for high-volume or real-time integrations, minimizing connection overhead and enabling bidirectional interaction.

Command lifecycle with response token

Explains how upstream systems stay non-blocking while actions complete later via token, stream, or webhook.



Bluetooth Mesh Topology

Atlas exchanges commands and responses with mesh nodes via a gateway, with bidirectional communication across the Bluetooth Mesh network.