Tech Note
Afero Cloud Notify

Cloud-to-Cloud Messaging

12 Jan 17 - Version 1.0

1. Introduction

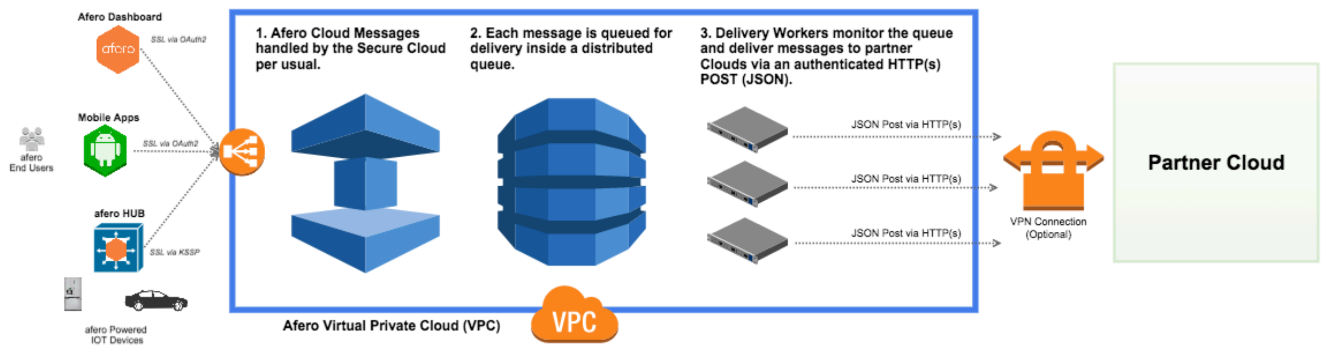
The Afero Cloud Notify feature provides real-time push messaging of attribute updates from the Afero Cloud to remote Cloud infrastructure managed by partners and/or developers. This allows remote clouds to extend the functionality of the Afero Cloud with custom functionality of their own, integrating into both new and existing business processes. Remote clouds will receive a well-defined, structured message relating information about an attribute change (such as turning on a light, a new sensor reading, etc.) immediately after that message has been fully processed by the Afero platform.

This “push” design eliminates the need for partners/developers to repeatedly poll the Afero API for information, dramatically lowering the overhead on both sides while reducing the time it takes for remote clouds to hear about changes on Afero devices.

2. Basic System Overview

Each time a device attribute change occurs on the Afero platform, a message communicating the change is generated. The message lifecycle proceeds as follows:

1. The message is securely processed by the Afero Cloud, as usual.
2. Once processed, the message is placed in a highly-scalable, distributed queue, ready for delivery.
3. A pool of delivery workers constantly monitors the queue for new messages to deliver. When a new message is placed in the queue, a worker immediately pulls it out and delivers it to the remote cloud for processing.


Afero Dashboard
SSL via OAuth2
Mobile Apps
SSL via OAuth2
afero End Users
afero HUB
SSL via KSSP
afero Powered IOT Devices
1. Afero Cloud Messages handled by the Secure Cloud per usual.
2. Each message is queued for delivery inside a distributed queue.
3. Delivery Workers monitor the queue and deliver messages to partner Clouds via an authenticated HTTP(s) POST (JSON).
JSON Post via HTTP(s)
JSON Post via HTTP(s)
JSON Post via HTTP(s)
VPN Connection (Optional)
Partner Cloud
Afero Virtual Private Cloud (VPC)
VPC

3. Remote Cloud Requirements

For a remote partner/developer cloud to listen for Afero attribute updates, the partner must provide Afero with an HTTP(s) endpoint written in the language and platform of their choosing. For example, partners may provide Afero with an an endpoint such as `https://api.mycompany.com/notify`. The implementation of this endpoint is up to the discretion of the developer, provided it is:

- SSL protected (For development purposes, self-signed certificates are fine, although we suggest a full commercial certificate for production deployments.)
- Able to consume an HTTP POST request containing a JSON payload
- Able to return an HTTP 2xx status code in response to the message delivery
- Be protected under Basic Authentication, at the very least

That’s it!

4. Message Payload Details

Each message delivered by Afero Notify looks like the following:

PAYLOAD EXAMPLE	FIELD DEFINITIONS

`{`
` "id": "c9bd4aaf-2163-4ca4-a56a-`
`a61c6817a05f",`
` "deviceId": "12345678abcdefg",`
` "utcTimestamp": 1458155516000,`
` "attributeId": 1024,`
` "attributeData": "01",`
` "attributeDataConverted": "1",`
` "attributeDataType": "UINT8",`
` "extendedData": {`
`     "serial_number": "00000000123456",`
`     "lot_number": "234532423423",`
`     "customField": "some value"`
` }`
`}`
	• **id:** The UUID (universally unique identifier) value that can be used to uniquely identify a particular message.
	• **deviceId:** The unique ID given to any Afero chip, uniquely identifying an Afero powered device.
	• **utcTimestamp:** A long value, in milliseconds, indicating the time in UTC that the message was processed by the Afero Cloud.
	• **attributeId:** Defines the numbered attribute, as assigned by Afero Profile Editor (APE), for the attribute that was updated.
	• **attributeData**: A string containing the hexadecimal representation of the attribute value at the time it was updated.
	• **attributeDataType**: The datatype of the attribute, as defined in APE.
	• **attributeDataConverted**: For convenience, a string containing the result of converting the **attributeData** value based on the **attributeDataType**.
	• **extendedData:** A flexible, free-form map of name/value pairs that the partner/developer defines at device provisioning time (or potentially after).
	This information can contain details that make sense to the partner, like serial numbers or other information that will help them resolve this device within their internal systems. This is optional information that may or may not be sent, depending on the needs of the partner.

5. Security and Authentication

5.1 SSL

Afero requires that messages sent by the Afero Cloud be sent over a secured channel. As such, remote endpoints must provide an SSL encrypted endpoint so that messages can be delivered over HTTPs. Self-signed certificates are OK (especially during development), but Afero strongly suggests using a commercial certificate (we may decide to enforce this requirement in the future).

5.2 Authentication

Remote clouds must protect their endpoints with Basic Authentication, and provide Afero with authentication credentials. These credentials will be sent in the headers of each Afero message delivered.

5.3 Remote Firewalls

To further protect their endpoints, partners should protect the endpoints via firewall rules, preventing access from any source IP except for the externally-facing IP addresses from the Afero Cloud network:

US Cloud	*Japan Cloud*
52.10.129.197	**52.196.125.126**
35.164.125.32	**52.68.17.150**
35.167.118.140	

5.4 VPN Delivery

Partners may choose to protect their endpoint behind a Virtual Private Network (VPN), rather than expose their endpoint to the open internet. Afero can work with these partners to authenticate to this VPN for message delivery. In the case of VPN delivery, partners may choose to relax either the Authentication or Firewall security measures at their discretion (although we suggest using a multi-layered approach to security in this context).

The "ideal" remote endpoint will be SSL protected with a commercial certificate, require Authentication upon each request, be firewalled to only whitelist the Afero IP address, and be delivered over VPN.

6. Customizing the Afero Message Request

Partners may provide Afero with custom "headers" to be added to each request and/or custom "query parameters" to be added to each request. Under most circumstances this will not be necessary, but if required, it can be provided.

For example, some remote clouds may be consuming messages from multiple sources and wish to route traffic from Afero differently from other traffic. The ability to tack on custom headers and/or query parameters to each request can help with this situation.

7. Failure Handling

In the event a problem occurs during message delivery (for example, during a maintenance on the remote cloud, or during an outage event on the remote cloud), Afero will save failed messages for up to 24 hours. After 24 hours, message failures will be purged from the Afero Cloud.

Partners can optionally choose to query our API for failed messages periodically or after downtime events by making an authenticated API request like so:

HTTP GET:
https://api.afero.io/v1/partners/{partnerId}/pushFailures

```
[{
    "partnerPushFailureId": "a58b7f53-f6ad-4099-
8d8e-c39ed76b0145",
    "partnerId": "366b3280-f0e1-4bd8-94ef-
8258fc469890",
    "createdTimestamp": 1458155516000,
    "errorMessage": "503 Service Unavailable",
    "partnerPushMessage": {
        "id": "c9bd4aaf-2163-4ca4-a56a-
a61c6817a05f",
        "deviceId": "12345678abcdefg",
        "utcTimestamp": 1458155516000,
        "attributeId": 1,
        "attributeData": "01",
        "attributeDataConverted": "1",
        "attributeDataType": "UINT8",
        "extendedData": {
            "serial_number": "00000000123456",
            "lot_number": "234532423423"
        }
    }
}]
```

FIELD DEFINITIONS

- **partnerPushFailureId:** The UUID (universally unique identifier) value that can be used to uniquely identify a message.
- **partnerId:** The partners' specific unique ID assigned to them by the Afero platform.
- **createdTimestamp:** The UTC timestamp (in milliseconds) when the failure took place.
- **errorMessage**: The message, if any, the Afero Cloud detected when trying to send the message.
- **partnerPushMessage**: A full copy of the message the Afero Cloud initially tried to send, as defined above.

After processing a failure message, partners can purge that specific message from the failure queue by executing the following request:

```
HTTP DELETE: https://api.afero.io/v1/partners/{partnerId}/pushFailures/{partnerPushFailureId}
```

Note: Neither the querying for, nor the explicit delete of, failure messages is required. Afero will purge these messages automatically after 24 hours. However, in certain circumstances partner clouds may wish to query for and handle these messages after the fact for auditing and so forth. This is a convenience feature and remote clouds are not required to use this capability.