Arrowhead Gatekeeper Core Service 4.1.3 Release Notes

In Arrowhead version 4.1.3, the whole codebase of the Arrowhead framework has been rewritten using Spring Boot technology. We believe the new version is better both in code quality, performance and robustness. It is easier to use, maintain and improve.
However, this means that Arrowhead version 4.1.3 is NOT compatible with providers and consumers written for version 4.1.2 because the changes affect not just the backend but the public interface of the core services, too.
This document describes the new public API of the Arrowhead Gatekeeper Core Service 4.1.3 and shows the key differences between the new interface and the old one.

Client Services

These services can be used by providers and consumers of the Arrowhead Cloud.

Uri: /gatekeeper/echo
Type: GET

Returns a “Got it” message with the purpose of testing the core service availability.

4.1.2 version: GET /gatekeeper/mgmt
It was only available for system operator of the local cloud.

Management Services

These services can only be used by the system operator of the local cloud. All date fields contain the text representation of a UTC timestamp.

Uri: /gatekeeper/mgmt/clouds
Type: GET
Query params:

- *page* – zero-based page index (optional),
- *item_per_page* – maximum number of items returned (optional),
- *sort_field* – sort field (optional, default: id, possible values: id, operator, name, createdAt, updatedAt),
- *direction* – direction of sorting (optional, default: ASC, possible values: ASC or DESC)

Returns a page of cloud records. If page and item_per_page are not defined, returns all records.

Returned JSON structure:

```
{
  "count": 0,
  "data": [
    {
      "id": 0,
      "name": "string",
      "operator": "string",
      "authenticationInfo": "string",
      "secure": true,
      "neighbor": true,
      "ownCloud": false,
      "gatekeeperRelays": [
        {
          "id": 0,
          "address": "string",
          "port": 0,
          "type": "GATEKEEPER_RELAY",
          "exclusive": false,
          "secure": true,
          "createdAt": "string",
          "updatedAt": "string"
        }
      ],
      "gatewayRelays": [
        {
          "id": 0,
          "address": "string",
          "port": 0,
          "type": "GATEWAY_RELAY",
          "exclusive": true,
          "secure": true,
          "createdAt": "string",
          "updatedAt": "string"
        }
      ],
      "createdAt": "string",
      "updatedAt": "string"
    }
  ]
}
```

Gatekeepers of different clouds does not communicate directly each other any more in version 4.1.3. Instead, they uses relays to send and receive messages, so every cloud has a list a relays which can be used to communicate with that cloud's gatekeeper. Consumers and providers in different clouds may have to communicate each other through relays too such as before (we refered relays as brokers in the previous version). Clouds can define exclusive relays for consumer-provider communication to avoid using public relays.

4.1.2 version: GET /gatekeeper/mgmt/neighborhood
This version returned only neighbour clouds and did not provide paging feature. Also, the returning objects were very different: it contains information to direct access the cloud's gatekeeper and did not contain time information.

Uri: /gatekeeper/mgmt/clouds
Type: POST

Creates cloud records and returns the newly created records.

Input JSON structure:

```
[
  {
    "operator": "string",
    "name": "string",
    "authenticationInfo": "string",
    "neighbor": true,
    "secure": true,
    "gatekeeperRelayIds": [
      0
    ],
    "gatewayRelayIds": [
      0
    ]
  }
]
```

Returned JSON structure:

```
{
  "count": 0,
  "data": [
    {
      "id": 0,
      "name": "string",
      "operator": "string",
      "authenticationInfo": "string",
      "secure": true,
      "neighbor": true,
      "ownCloud": false,
      "gatekeeperRelays": [
        {
          "id": 0,
          "address": "string",
          "port": 0,
          "type": "GATEKEEPER_RELAY",
          "exclusive": false,
          "secure": true,
          "createdAt": "string",
          "updatedAt": "string"
        }
      ],
      "gatewayRelays": [
        {
          "id": 0,
          "address": "string",
          "port": 0,
          "type": "GATEWAY_RELAY",
          "exclusive": true,
          "secure": true,
          "createdAt": "string",
          "updatedAt": "string"
        }
      ],
      "createdAt": "string",
      "updatedAt": "string"
    }
  ]
}
```

4.1.2 version: POST /gatekeeper/mgmt/neighborhood
This version required gatekeeper direct access informations instead of relay ids.

Uri: /gatekeeper/mgmt/clouds/{id}
Type: GET

Returns the cloud record specified by the id path parameter.

Returned JSON structure:

```
{
  "id": 0,
  "name": "string",
  "operator": "string",
  "authenticationInfo": "string",
  "secure": true,
  "neighbor": true,
  "ownCloud": false,
  "gatekeeperRelays": [
    {
      "id": 0,
      "address": "string",
      "port": 0,
      "type": "GATEKEEPER_RELAY",
      "exclusive": false,
      "secure": true,
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "gatewayRelays": [
    {
      "id": 0,
      "address": "string",
      "port": 0,
      "type": "GATEWAY_RELAY",
      "exclusive": true,
      "secure": true,
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: Not existed.

Uri: /gatekeeper/mgmt/clouds/{id}
Type: PUT

Updates the cloud specified by id path parameter and returns the updated record.

Input JSON:

```
 {
  "operator": "string",
  "name": "string",
  "authenticationInfo": "string",
  "neighbor": true,
  "secure": true,
  "gatekeeperRelayIds": [
    0
  ],
  "gatewayRelayIds": [
    0
  ]
}
```

Returned JSON structure:

```
{
  "id": 0,
  "name": "string",
  "operator": "string",
  "authenticationInfo": "string",
  "secure": true,
  "neighbor": true,
  "ownCloud": false,
  "gatekeeperRelays": [
    {
      "id": 0,
      "address": "string",
      "port": 0,
      "type": "GATEKEEPER_RELAY",
      "exclusive": false,
      "secure": true,
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "gatewayRelays": [
    {
      "id": 0,
      "address": "string",
      "port": 0,
      "type": "GATEWAY_RELAY",
      "exclusive": true,
      "secure": true,
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: PUT /gatekeeper/mgmt/neighborhood
This version used operator and name to identify which cloud will be updated. And the input and output JSON structure was different.

Uri: /gatekeeper/mgmt/clouds/{id}
Type: DELETE

Removes the cloud record specified by the id path parameter.

4.1.2 version: Not existed.

Uri: /gatekeeper/mgmt/clouds/assign
Type: POST

Assigns gatekeeper and/or exclusive gateway relays to a cloud and returns the updated cloud.

Input JSON:

```
{
  "cloudId": 0,
  "gatekeeperRelayIds": [
    0
  ],
  "gatewayRelayIds": [
    0
  ]
}
```

Returned JSON structure:

```
{
  "id": 0,
  "name": "string",
  "operator": "string",
  "authenticationInfo": "string",
  "secure": true,
  "neighbor": true,
  "ownCloud": false,
  "gatekeeperRelays": [
    {
      "id": 0,
      "address": "string",
      "port": 0,
      "type": "GATEKEEPER_RELAY",
      "exclusive": false,
      "secure": true,
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "gatewayRelays": [
    {
      "id": 0,
      "address": "string",
      "port": 0,
      "type": "GATEWAY_RELAY",
      "exclusive": true,
      "secure": true,
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: Not existed.

Uri: /gatekeeper/mgmt/relays
Type: GET
Query params:

- *page* – zero-based page index (optional),
- *item_per_page* – maximum number of items returned (optional),
- *sort_field* – sort field (optional, default: id, possible values: id, address. port, createdAt, updatedAt),
- *direction* – direction of sorting (optional, default: ASC, possible values: ASC or DESC)

Returns a page of relay records. If page and item_per_page are not defined, returns all records.

Returned JSON structure:

```
{
  "count": 0,
  "data": [
    {
      "id": 0,
      "address": "string",
      "port": 0,
      "type": "GATEKEEPER_RELAY",
      "exclusive": false,
      "secure": true,
      "createdAt": "string",
      "updatedAt": "string"
    }
  ]
}
```

4.1.2 version: GET /gatekeeper/mgmt/brokers
This version always returned all records in an array of JSON objects. The objects did not contain any time information. In this version we used brokers only to provide communication channel between consumer and provider in different clouds with no direct access. Gatekeepers communicates directly, so there was no need for broker type. Also, exclusive broker concept did not exist at that time.

Uri: /gatekeeper/mgmt/relays
Type: POST

Creates records records and returns the newly created records.

Input JSON structure:

```
[
  {
    "address": "string",
    "port": 0,
    "type": "string"
    "exclusive": true,
    "secure": true,
  }
]
```

Possible values for type are:

- GATEKEEPER_RELAY,
- GATEWAY_RELAY and
- GENERAL_RELAY (if you want to use a relay for both purposes)

Only GATEWAY_RELAY can be exclusive.

Returned JSON structure:

```
{
  "count": 0,
  "data": [
    {
      "id": 0,
      "address": "string",
      "port": 0,
      "type": "GATEWAY_RELAY",
      "exclusive": true,
      "secure": true,
      "createdAt": "string",
      "updatedAt": "string"
    }
  ]
}
```

4.1.2 version: POST /gatekeeper/mgmt/brokers
This version used different input JSON: no type and exclusive flag. Also, the returning JSON did not contain time information.

Uri: /gatekeeper/mgmt/relays/address/{address}/port/{port}
Type: GET

Returns the relay record specified by address and port path parameters.

Returned JSON structure:

```
{
  "id": 0,
  "address": "string",
  "port": 0,
  "type": "GATEWAY_RELAY",
  "exclusive": true,
  "secure": true,
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: GET /gatekeeper/mgmt/brokers/address/{address}/port/{port}
The returning JSON did not contain type, exclusive flag nor any time information.

Uri: /gatekeeper/mgmt/relays/{id}
Type: GET

Returns the relay record specified by the id path parameter.

Returned JSON structure:

```
{
  "id": 0,
  "address": "string",
  "port": 0,
  "type": "GATEWAY_RELAY",
  "exclusive": true,
  "secure": true,
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: Not existed.

Uri: /gatekeeper/mgmt/relays/{id}
Type: PUT

Updates the relay specified by id path parameter and returns the updated record.

Input JSON:

```
{
  "address": "string",
  "port": 0,
  "type": "string"
  "exclusive": true,
  "secure": true,
}
```

Possible values for type are:

- GATEKEEPER_RELAY,
- GATEWAY_RELAY and
- GENERAL_RELAY (if you want to use a relay for both purposes)

Only GATEWAY_RELAY can be exclusive.

Returned JSON structure:

```
{
  "id": 0,
  "address": "string",
  "port": 0,
  "type": "GATEWAY_RELAY",
  "exclusive": true,
  "secure": true,
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: PUT /gatekeeper/mgmt/brokers/{brokerId}
This version used different input JSON: no type and exclusive flag. Also, the returning JSON did not contain time information.

Uri: /gatekeeper/mgmt/relays/{id}
Type: DELETE

Removes the relay record specified by the id path parameter. It is not allowed to remove the only gatekeeper relay of a cloud.

4.1.2 version: DELETE /gatekeeper/mgmt/broker/{brokerId}
Basically is the same than the new version.

The following services are no longer exist:

- GET /gatekeeper/mgmt/neighborhood/operator/{operator}/cloudname/{cloudName}
- DELETE /gatekeeper/mgmt/neighborhood/operator/{operator}/cloudname/{cloudName}
- GET /gatekeeper/mgmt/brokers/brokername/{brokerName}
- GET /getekeeper/mgmt/brokers/address/{address}

Private Services

These services can only be used by other core services, therefore they are not part of the public API.