Arrowhead Service Registry 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 Service Registry 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. All date fields contain the text representation of a UTC timestamp.

Uri: /serviceregistry/echo
Type: GET

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

4.1.2 version: GET /serviceregistry

Uri: /serviceregistry/query
Type: POST

Returns Service Registry data that fits the input specification.

Input JSON structure:

```
{
  "serviceDefinitionRequirement": "string",
  "interfaceRequirements": [
    "string"
  ],
  "securityRequirements": [
    "NOT_SECURE"
  ],
  "metadataRequirements": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  },
  "versionRequirement": 0,
  "maxVersionRequirement": 0,
  "minVersionRequirement": 0,
  "pingProviders": true
}
```

Mandatory fields: “serviceDefinitionRequirement”

Valid interface name pattern: protocol-“SECURE” or “INSECURE”-format
(e.g.: HTTPS-SECURE-JSON)

Possible values for “securityRequirements” are:

- NOT_SECURE
- CERTIFICATE
- TOKEN
- not defined, if you don’t want to filter on security type

Clients are allowed to query ONLY for the public core system services.

Returned JSON structure:

```
{
  "serviceQueryData": [
    {
      "id": 0,
      "serviceDefinition": {
        "id": 0,
        "serviceDefinition": "string",
        "createdAt": "string",
        "updatedAt": "string"
      },
      "provider": {
        "id": 0,
        "systemName": "string",
        "address": "string",
        "port": 0,
        "authenticationInfo": "string",
        "createdAt": "string",
        "updatedAt": "string"
      },
      "serviceUri": "string",
      "endOfValidity": "string",
      "secure": "NOT_SECURE",
      "metadata": {
        "additionalProp1": "string",
        "additionalProp2": "string",
        "additionalProp3": "string"
      },
      "version": 0,
      "interfaces": [
        {
          "id": 0,
          "interfaceName": "string",
          "createdAt": "string",
          "updatedAt": "string"
        }
      ],
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "unfilteredHits": 0
}
```

4.1.2 version: PUT /serviceregistry /query
This version always returned the records in an array of JSON objects. The response did not contain any information about the unfiltered hits and the objects did not contain any modification related time stamp information. Interfaces and metadata were bound to the service definition and security type

was not defined. Service Registry object did contain an unnecessary “udp” flag beside the interface definition.

Uri: /serviceregistry/register
Type: POST

Registers a service. A provider is allowed to register only its own services. It means that provider system name and certificate common name must match for successful registration.

Input JSON structure:

```
{
  "serviceDefinition": "string",
  "providerSystem": {
    "systemName": "string",
    "address": "string",
    "port": 0,
    "authenticationInfo": "string"
  },
  "serviceUri": "string",
  "endOfValidity": "string",
  "secure": "NOT_SECURE",
  "metadata": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  },
  "version": 0,
  "interfaces": [
    "string"
  ]
}
```

Mandatory fields: “serviceDefinition”, “providerSystem”, “serviceUri”, “interfaces”

Valid interface name pattern: protocol-“SECURE” or “INSECURE”-format
(e.g.: HTTPS-SECURE-JSON)

Possible values for “secure” are:

- NOT_SECURE (used value when field not defined)
- CERTIFICATE
- TOKEN

Returned JSON structure:

```
{
  "id": 0,
  "serviceDefinition": {
    "id": 0,
    "serviceDefinition": "string",
    "createdAt": "string",
    "updatedAt": "string"
  },
  "provider": {
    "id": 0,
    "systemName": "string",
    "address": "string",
    "port": 0,
    "authenticationInfo": "string",
    "createdAt": "string",
    "updatedAt": "string"
  },
  "serviceUri": "string",
  "endOfValidity": "string",
  "secure": "NOT_SECURE",
  "metadata": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  },
  "version": 0,
  "interfaces": [
    {
      "id": 0,
      "interfaceName": "string",
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: POST /serviceregistry /register
In this version interfaces and metadata were bound to the service definition and security type was not defined at all. The response object did not contain any modification related time stamp information. Service Registry object did contain an unnecessary “udp” flag beside the interface definition.

Uri: /serviceregistry/unregister
Type: DELETE
Query params:

- *service_definition* – name of service to be removed
- *system_name* – name of provider
- *address* – address where the service was available from
- *port* – port where the service was available from

Removes a registered service. A provider is allowed to unregister only its own services. It means that provider system name and certificate common name must match for successful unregistration.

4.1.2 version: PUT / serviceregistry/remove
In this version the input was a JSON object with many unnecessary information.

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: /serviceregistry/mgmt
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, createdAt, updatedAt),
- *direction* – direction of sorting (optional, default: ASC, possible values: ASC or

DESC)

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

Returned JSON structure:

```
{
  "data": [
    {
      "id": 0,
      "serviceDefinition": {
        "id": 0,
        "serviceDefinition": "string",
        "createdAt": "string",
        "updatedAt": "string"
      },
      "provider": {
        "id": 0,
        "systemName": "string",
        "address": "string",
        "port": 0,
        "authenticationInfo": "string",
        "createdAt": "string",
        "updatedAt": "string"
      },
      "serviceUri": "string",
      "endOfValidity": "string",
      "secure": "NOT_SECURE",
      "metadata": {
        "additionalProp1": "string",
        "additionalProp2": "string",
        "additionalProp3": "string"
      },
      "version": 0,
      "interfaces": [
        {
          "id": 0,
          "interfaceName": "string",
          "createdAt": "string",
          "updatedAt": "string"
        }
      ],
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "count": 0
}
```

4.1.2 version: GET /serviceregistry/mgmt/all

This version always returned the records in an array of JSON objects. The objects did not contain any modification related time stamp information. Interfaces and metadata were bound to the service definition and security type was not defined. Service Registry object did contain an unnecessary “udp” flag beside the interface definition.

Uri: /serviceregistry/mgmt/{id}
Type: GET

Returns the service registry record specified by the id path parameter.

Returned JSON structure:

```
{
  "id": 0,
  "serviceDefinition": {
    "id": 0,
    "serviceDefinition": "string",
    "createdAt": "string",
    "updatedAt": "string"
  },
  "provider": {
    "id": 0,
    "systemName": "string",
    "address": "string",
    "port": 0,
    "createdAt": "string",
    "authenticationInfo": "string",
    "updatedAt": "string"
  },
  "serviceUri": "string",
  "endOfValidity": "string",
  "metadata": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  },
  "secure": "NOT_SECURE",
  "version": 0,
  "interfaces": [
    {
      "id": 0,
      "interfaceName": "string",
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: GET /serviceregistry/mgmt/id/{id}
In this version interfaces and metadata were bound to the service definition and security type was not defined at all. The response object did not contain any modification related time stamp information. Service Registry object did contain an unnecessary “udp” flag beside the interface definition.

Uri: /serviceregistry/mgmt/servicedef/{serviceDefinition}
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, createdAt, updatedAt),
- *direction* – direction of sorting (optional, default: ASC, possible values: ASC or DESC)

Returns a page of service registry records specified by the serviceDefinition path parameter. If page and item_per_page are not defined, returns all records.

Returned JSON structure:

```
{
  "data": [
    {
      "id": 0,
      "serviceDefinition": {
        "id": 0,
        "serviceDefinition": "string",
        "createdAt": "string",
        "updatedAt": "string"
      },
      "provider": {
        "id": 0,
        "systemName": "string",
        "address": "string",
        "port": 0,
        "authenticationInfo": "string",
        "createdAt": "string",
        "updatedAt": "string"
      },
      "serviceUri": "string",
      "endOfValidity": "string",
      "secure": "NOT_SECURE",
      "metadata": {
        "additionalProp1": "string",
        "additionalProp2": "string",
        "additionalProp3": "string"
      },
      "version": 0,
      "interfaces": [
        {
          "id": 0,
          "interfaceName": "string",
          "createdAt": "string",
          "updatedAt": "string"
        }
      ],
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "count": 0
}
```

4.1.2 version: GET /serviceregistry/mgmt/servicedef/{serviceDefinition}
This version always returned the records in an array of JSON objects. The objects did not contain any modification related time stamp information. Interfaces and metadata were bound to the service definition and security type was not defined. Service Registry object did contain an unnecessary “udp” flag beside the interface definition.

Uri: /serviceregistry/mgmt/grouped
Type: GET

Return all service registry entries grouped by three different purpose:

- autoCompleteData
- servicesGroupedByServiceDefinitionAndInterface
- servicesGroupedBySystems

```
{
  "autoCompleteData": {
    "interfaceList": [
      {
        "id": 0,
        "value": "string"
      }
    ],
    "serviceList": [
      {
        "id": 0,
        "value": "string"
      }
    ],
    "systemList": [
      {
        "id": 0,
        "systemName": "string",
        "address": "string",
        "port": 0,
        "authenticationInfo": "string",
        "createdAt": "string",
        "updatedAt": "string"
      }
    ]
  },
  "servicesGroupedByServiceDefinitionAndInterface": [
    {
      "serviceDefinitionId": 0,
      "serviceDefinition": "string",
      "interfaceName": "string",
      "providerServices": [
        {
          "id": 0,
          "serviceDefinition": {
            "id": 0,
            "serviceDefinition": "string",
            "createdAt": "string",
            "updatedAt": "string"
          },
          "provider": {
            "id": 0,
            "systemName": "string",
            "address": "string",
            "port": 0,
            "authenticationInfo": "string",
            "createdAt": "string",
            "updatedAt": "string"
          },
          "serviceUri": "string",
          "endOfValidity": "string",
          "secure": "NOT_SECURE",
          "metadata": {
            "additionalProp1": "string",
            "additionalProp2": "string",
            "additionalProp3": "string"
          },
          "version": 0,
          "interfaces": [
            {
              "id": 0,
              "interfaceName": "string",
              "createdAt": "string",
              "updatedAt": "string"
            }
          ],
          "createdAt": "string",
          "updatedAt": "string"
        }
      ],
    }
  ],
  "servicesGroupedBySystems": [
    {
      "systemId": 0,
      "systemName": "string",
      "address": "string",
      "port": 0,
      "services": [
        {
          "id": 0,
          "serviceDefinition": {
            "id": 0,
            "serviceDefinition": "string",
            "createdAt": "string",
            "updatedAt": "string"
          },
          "provider": {
            "id": 0,
            "systemName": "string",
            "address": "string",
            "port": 0,
            "authenticationInfo": "string",
            "createdAt": "string",
            "updatedAt": "string"
          },
          "serviceUri": "string",
          "endOfValidity": "string",
          "secure": "NOT_SECURE",
          "metadata": {
            "additionalProp1": "string",
            "additionalProp2": "string",
            "additionalProp3": "string"
          },
          "version": 0,
          "interfaces": [
            {
              "id": 0,
              "interfaceName": "string",
              "createdAt": "string",
              "updatedAt": "string"
            }
          ],
          "createdAt": "string",
          "updatedAt": "string"
        }
      ]
    }
  ]
}
```

4.1.2 version: Not existed

Uri: /serviceregistry/mgmt
Type: POST

Creates service registry record and returns the newly created record.

Input JSON structure:

```
{
  "serviceDefinition": "string",
  "providerSystem": {
    "systemName": "string",
    "address": "string",
    "port": 0,
    "authenticationInfo": "string"
  },
  "serviceUri": "string",
  "endOfValidity": "string",
  "secure": "NOT_SECURE",
  "metadata": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  },
  "version": 0,
  "interfaces": [
    "string"
  ]
}
```

Mandatory fields: “serviceDefinition”, “providerSystem”, “serviceUri”, “interfaces”

Valid interface name pattern: protocol-“SECURE” or “INSECURE”-format
(e.g.: HTTPS-SECURE-JSON)

Possible values for “secure” are:

- NOT_SECURE (used value when field not defined)
- CERTIFICATE
- TOKEN

Returned JSON structure:

```
{
  "id": 0,
  "serviceDefinition": {
    "id": 0,
    "serviceDefinition": "string",
    "createdAt": "string",
    "updatedAt": "string"
  },
  "provider": {
    "id": 0,
    "systemName": "string",
    "address": "string",
    "port": 0,
    "createdAt": "string",
    "authenticationInfo": "string",
    "updatedAt": "string"
  },
  "serviceUri": "string",
  "endOfValidity": "string",
  "metadata": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  },
  "secure": "NOT_SECURE",
  "version": 0,
  "interfaces": [
    {
      "id": 0,
      "interfaceName": "string",
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: POST /serviceregistry/support/register
It was available for clients as well, not only for the system operator of the local cloud. Interfaces and metadata were bound to the service definition and security type was not defined at all. The response object did not contain any modification related time stamp information. Service Registry object did contain an unnecessary “udp” flag beside the interface definition.

Uri: /serviceregistry/mgmt/{id}
Type: PUT

Updates and returns the modified service registry record specified by the id path parameter. Not defined fields are going to be updated to “null” value.

Input JSON structure:

```
{
  "serviceDefinition": "string",
  "providerSystem": {
    "systemName": "string",
    "address": "string",
    "port": 0,
    "authenticationInfo": "string"
  },
  "serviceUri": "string",
  "endOfValidity": "string",
  "secure": "NOT_SECURE",
  "metadata": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  },
  "version": 0,
  "interfaces": [
    "string"
  ]
}
```

Mandatory fields: “serviceDefinition”, “providerSystem”, “serviceUri”, “interfaces”

Valid interface name pattern: protocol-“SECURE” or “INSECURE”-format
(e.g.: HTTPS-SECURE-JSON)

Possible values for “secure” are:

- NOT_SECURE (used value when field not defined)
- CERTIFICATE
- TOKEN

Returned JSON structure:

```
{
  "id": 0,
  "serviceDefinition": {
    "id": 0,
    "serviceDefinition": "string",
    "createdAt": "string",
    "updatedAt": "string"
  },
  "provider": {
    "id": 0,
    "systemName": "string",
    "address": "string",
    "port": 0,
    "createdAt": "string",
    "authenticationInfo": "string",
    "updatedAt": "string"
  },
  "serviceUri": "string",
  "endOfValidity": "string",
  "metadata": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  },
  "secure": "NOT_SECURE",
  "version": 0,
  "interfaces": [
    {
      "id": 0,
      "interfaceName": "string",
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: PUT /serviceregistry/mgmt/update/{id}
In this version interfaces and metadata were bound to the service definition and security type was not defined at all. The response object did not contain any modification related time stamp information. Service Registry object did contain an unnecessary “udp” flag beside the interface definition.

Uri: /serviceregistry/mgmt/{id}
Type: PATCH

Updates and returns the modified service registry record specified by the id path parameter. Not defined fields are going to be NOT updated.

Input JSON structure:

```
{
  "serviceDefinition": "string",
  "providerSystem": {
    "systemName": "string",
    "address": "string",
    "port": 0,
    "authenticationInfo": "string"
  },
  "serviceUri": "string",
  "endOfValidity": "string",
  "secure": "NOT_SECURE",
  "metadata": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  },
  "version": 0,
  "interfaces": [
    "string"
  ]
}
```

Valid interface name pattern: protocol-“SECURE” or “INSECURE”-format
(e.g.: HTTPS-SECURE-JSON)

Possible values for “secure” are:

- NOT_SECURE
- CERTIFICATE
- TOKEN

Returned JSON structure:

```
{
  "id": 0,
  "serviceDefinition": {
    "id": 0,
    "serviceDefinition": "string",
    "createdAt": "string",
    "updatedAt": "string"
  },
  "provider": {
    "id": 0,
    "systemName": "string",
    "address": "string",
    "port": 0,
    "createdAt": "string",
    "authenticationInfo": "string",
    "updatedAt": "string"
  },
  "serviceUri": "string",
  "endOfValidity": "string",
  "metadata": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  },
  "secure": "NOT_SECURE",
  "version": 0,
  "interfaces": [
    {
      "id": 0,
      "interfaceName": "string",
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: Not existed

Uri: /serviceregistry/mgmt/{id}
Type: DELETE

Remove the service registry record specified by the id path parameter.

4.1.2 version: DELETE /serviceregistry/mgmt/{entryId}
This version did return Http 404 (not found), when record was not found by id.

Uri: /serviceregistry/mgmt/services
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, createdAt, updatedAt),
- *direction* – direction of sorting (optional, default: ASC, possible values: ASC or DESC)

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

Returned JSON structure:

```
{
  "data": [
    {
      "id": 0,
      "serviceDefinition": "string",
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "count": 0
}
```

4.1.2 version: GET /mgmt/services
This version always returned the records in an array of JSON objects. The objects did not contain any modification related time stamp information. Interfaces and metadata were part of the service definition entity.

Uri: /serviceregistry/mgmt/services/{id}
Type: GET

Returns the service definition record specified by the id path parameter.

Returned JSON structure:

```
{
  "id": 0,
  "serviceDefinition": "string",
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: GET /mgmt/services/{serviceId}
The response object did not contain any modification related time stamp information. Interfaces and metadata were part of the service definition entity.

Uri: /serviceregistry/mgmt/services
Type: POST

Creates service definition record and returns the newly created record.
Input JSON structure:

```
{
  "serviceDefinition": "string"
}
```

Returned JSON structure:

```
{
  "id": 0,
  "serviceDefinition": "string",
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: POST /mgmt/services
In this version interfaces and metadata were part of the service definition entity. The response object did not contain any modification related time stamp information.

Uri: /serviceregistry/mgmt/services/{id}
Type: PUT

Updates and returns the modified service definition record specified by the id path parameter.

Input JSON structure:

```
{
  "serviceDefinition": "string"
}
```

Returned JSON structure:

```
{
  "id": 0,
  "serviceDefinition": "string",
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: PUT /mgmt/services/{serviceId}
The response object did not contain any modification related time stamp information. Interfaces and metadata were part of the service definition entity.

Uri: /serviceregistry/mgmt/services/{id}
Type: PATCH

Updates and returns the modified service definition record specified by the id path parameter.

Input JSON structure:

```
{
  "serviceDefinition": "string"
}
```

Returned JSON structure:

```
{
  "id": 0,
  "serviceDefinition": "string",
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: Not existed

Uri: /serviceregistry/mgmt/services/{id}
Type: DELETE

Removes the service definition record specified by the id path parameter.

4.1.2 version: DELETE /mgmt/services/{serviceId}
This version did return Http 404 (not found), when record was not found by id.

Uri: /serviceregistry/mgmt/systems
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, systemName, address, port, createdAt, updatedAt),
- *direction* – direction of sorting (optional, default: ASC, possible values: ASC or DESC)

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

Returned JSON structure:

```
{
  "data": [
    {
      "id": 0,
      "systemName": "string",
      "address": "string",
      "port": 0,
      "authenticationInfo": "string",
      "createdAt": "string",
      "updatedAt": "string"
    }
  ],
  "count": 0
}
```

4.1.2 version: GET /mgmt/systems

This version always returned the records in an array of JSON objects. The objects did not contain any modification related time stamp information.

Uri: /serviceregistry/mgmt/systems/{id}
Type: GET

Returns the system record specified by the id path parameter.

Returned JSON structure:

```
{
  "id": 0,
  "systemName": "string",
  "address": "string",
  "port": 0,
  "authenticationInfo": "string",
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: GET /mgmt/systems/{systemId}
In this version the response object did not contain any modification related time stamp information.

Uri: /serviceregistry/mgmt/systems
Type: POST

Creates system record and returns the newly created record.

Input JSON structure:

```
{
  "systemName": "string",
  "address": "string",
  "port": 0,
  "authenticationInfo": "string"
}
```

Mandatory fields: “systemName”, “address”, “port”

Returned JSON structure:

```
{
  "id": 0,
  "systemName": "string",
  "address": "string",
  "port": 0,
  "authenticationInfo": "string",
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: POST /mgmt/systems
In this version the response object did not contain any modification related time stamp information.

Uri: /serviceregistry/mgmt/systems/{id}
Type: PUT

Updates and returns the modified system record specified by the id path parameter. Not defined fields are going to be updated to “null” value.

Input JSON structure:

```
{
  "systemName": "string",
  "address": "string",
  "port": 0,
  "authenticationInfo": "string"
}
```

Mandatory fields: “systemName”, “address”, “port”

Returned JSON structure:

```
{
  "id": 0,
  "systemName": "string",
  "address": "string",
  "port": 0,
  "authenticationInfo": "string",
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: PUT /mgmt/systems/{systemId}
In this version the response object did not contain any modification related time stamp information.

Uri: /serviceregistry/mgmt/systems/{id}
Type: PATCH

Updates and returns the modified system record specified by the id path parameter. Not defined fields are going to be NOT updated.

Input JSON structure:

```
{
  "systemName": "string",
  "address": "string",
  "port": 0,
  "authenticationInfo": "string"
}
```

Returned JSON structure:

```
{
  "id": 0,
  "systemName": "string",
  "address": "string",
  "port": 0,
  "authenticationInfo": "string",
  "createdAt": "string",
  "updatedAt": "string"
}
```

4.1.2 version: Not existed

Uri: /serviceregistry/mgmt/systems/{id}
Type: DELETE

Removes the system record specified by the id path parameter.

4.1.2 version: DELETE /mgmt/systems/{systemId}
This version did return Http 404 (not found), when record was not found by id.

The following services are no longer exist:

- PUT /mgmt/services
- PUT /mgmgt/systems
- GET /serviceregistry/mgmt
- GET /serviceregistry/mgmt/systemId/{systemId}
- GET /serviceregistry/mgmt/serviceId/{serviceId}/providers
- PUT /serviceregistry/mgmt/query
- PUT /serviceregistry/mgmt/subscriptions/{id}
- PUT /serviceregistry/support/remove
- DELETE /serviceregistry/mgmt/all

Private Services

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