**Data Push Platform Interface Specification**

**Project name： Data Push Platform**

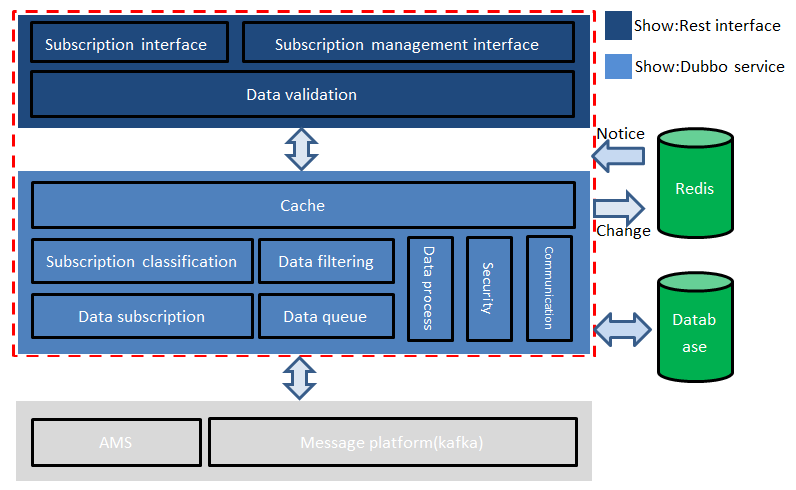
**Version： V1.0**

**Date： July 04, 2017**

Document revision record

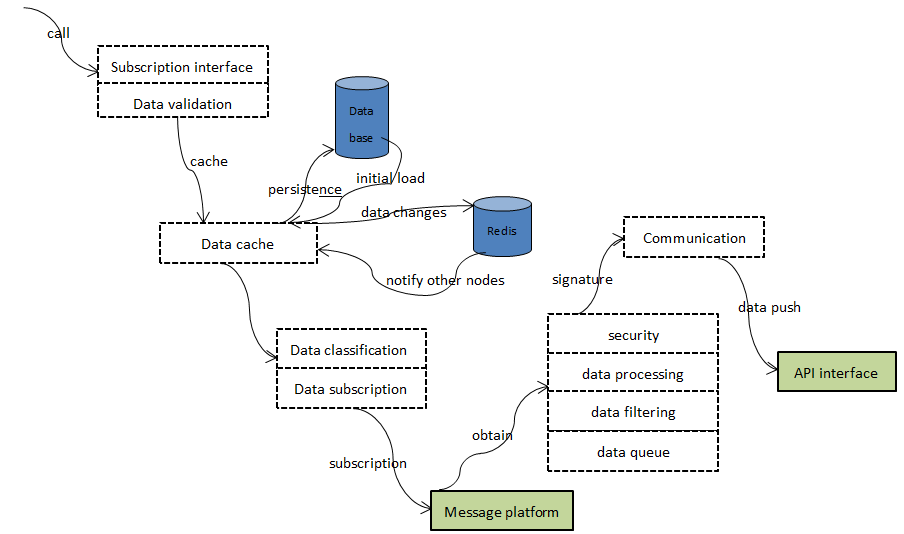
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Standard revision history record | | | | | | | | | |
| Version | Date | | Reviser | Revision page | | | Revise the content | | |
| V1.0 | 2017-07-04 | | Yang Junku |  | | | new | | |
|  |  | |  |  | | |  | | |
|  |  | |  |  | | |  | | |
|  |  | |  |  | | |  | | |
| 会 签 | | | | | | | | | |
| Management body | | Signed advice | | | | Signature | | | Date |
|  | |  | | | |  | | |  |
|  | |  | | | |  | | |  |
|  | |  | | | |  | | |  |
| Create | |  | | | Review | | |  | |
| Audit | |  | | | Approval | | |  | |

## Technology Architecture



**Description:**The red box is the system scope of data push, and the external system dependent outside the red box.

Based on the above technical architecture, the following figure shows the implementation principle and the interaction process of each module.



## Receive push interface requirements

### Access method

The interactive interface with data push is unified into REST interface based on HTTP.

Utf-8 coding;

POST to push data.

### Input parameters

The parameters identified in this document

|  |  |
| --- | --- |
| Type | Description |
| BODY | Refers to passing parameters through the Http request Body |
| HEAD | Refers to passing parameters through the Http request Header |

### Public fields

A common field describes the common fields that are included in each request and should be allowed. These fields are not specified and are not repeated in the business class interface message definition.

Verify sign must be transmitted.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **The input parameters** | | | | | | | | |
| **Parameter names** | **Type** | | **location** | **Required or not** | | **Description** | | |
| sign | String | | Header | Required | | The string to be signed is:systemId+systemKey +timestamp；  systemId:Cloud application ID, within 40 characters,Haier U+ cloud platform globally unique;  systemKey:The systemKey applied to the application cannot be sent in clear text on the Haigeek network.  Timestamp:Unix timestamp(YYMMDDhhmmss). | | |
| timestamp | String | | Header | Required | | Unix timestamp | | |
| Content-Type | String | | Header | Required | | The data format type of the client and hardware access interaction, the JSON format interaction is set to : application/json; charset=utf-8 | | |
| **Output parameters** | | | | | | | | |
| **Parameter names** | | **Type** | | | **location** | | **Required or not** | **Description** |
| resultCode | | String | | | Body | | Required | Return code |
| resultInfo | | String | | | Body | | Not required | Return information |

## Body description:

Haier currently pushes the data format of the three-party cloud (on and off line, state, alarm and big data). Format description:

### Device status report(on and off line)

#### Data format:

{"online": "1", "deviceId": "DC330D0112FB","wifiType":"00000000000000000008080000000041410"}

### Device status report

#### Data format:

{"status":{"resn1RunningStatus":"True","targetTemperature":"40","resn2CycleStatus":"True","heatingStatus":"1","onOffStatus":"True","resn1Temperature":"40","resn1TimeHH":"04","currentTemperature":"26","nightMode":"False","resn2TimeHH":"00","lowerTemperature":"29","runningMode":"0","resn1TimeMM":"00","resn2RunningStatus":"False","volume":"60","scene":"7","resn2TimeMM":"00","resn2Temperature":"75","resn1CycleStatus":"True","resnMode":"0","upperTemperature":"0"},"deviceId":"0007A8D02632","wifiType":"00000000000000000008080000000041410"}

### Device alarm report

#### Data format:

{ "faults": ["tempSensorErr", "humiditySensorErr"],"deviceId": "DC330D0112FB", "wifiType":"00000000000000000008080000000041410"}

### Device big data format report

#### Data format

{"bigdata":{"workTime":"4380","useGasL":"58816"},"deviceId":"0007A8D02632","wifiType":"00000000000000000008080000000041410"}

## Return code list

Only the successful return code is agreed, and the failure or error code is determined by a third party according to its own situation to facilitate locating the problem.

|  |  |
| --- | --- |
| **resultCode(Return code 5 digits)** | **resultInfo** |
| 00000 | Successful |
|  |  |
|  |  |
|  |  |
|  |  |