

物联网设备 MQTT 对接协议

版本	1.4
编号	0001
项目	物联网开发平台
部门	软件部门
主导	韦昌南 蒙广波
日期	2022.12.14

修订历史记录

版本	修订人	日期	描述	审核人	发布日期
1.0	蒙广波	2022.09.26	初版		
1.1	蒙广波	2022.10.13	添加模组免预注册		
1.2	蒙广波	2022.11.24	模组免预注册兼容自带 deviceName		
1.3	蒙广波	2022.12.01	添加云柯猫眼模组支持		
1.4	蒙广波	2022.12.14	数据传输加密		

1 协议说明

1.1 Topic 结构

MQTT 是通过 Topic 的订阅和发布来通信的。考虑到良好的扩展和架构，层级结构如下：

业务 / 操作 / 子操作（N 个） / 资源 / \${productId} / \${deviceName}

层级	描述	是否必须
----	----	------

业务	业务范围，当前有 iot,lock	是
操作	标识当前 Topic 的操作	是
子操作	主操作下面的子操作，可以有多个层级的子操作	否
资源	表示操作要操作的数据类型，比如 alarm	否
productId	设备产品标识，创建产品的时候生成，每个产品唯一	是
deviceName	设备唯一的标识，相当于 SN 号	是

例子：设备上报透传信息 `iot/transfer/up/cateye-lock/anzh0102191101000001`

1.2 传输数据结构

1.2.1 数据例子

数据传输采用 JSON 方式，注意的是文件上传不是严格的 JSON 格式，是自定义的格式。格式分为数据加密和数据不加密两类，分配服务器需要数据加密，业务服务器不需要数据加密（服务器分类请看 1.4.1）。

加密的数据格式例子如下：

以下例子为一型一密免预注册请求

```
iot/register/product/request/${productId}/${deviceName}{  
  "id": "123456",  
  "version": 1,  
  "time": 1521234567000  
  "code": 0  
  "data": "dsfdsa21fa31f5dsa4fsd5a64fdas64fsd54"  
}
```

其中 data 为数据字段，值为数据加密后的字符数据。

解密后的值为：

```
{  
  "type" : 0,  
  "productId" : "YTL-WB01"  
  "deviceName" : "anzh0102191101000001"  
}
```

数据加密方式为 AES，详细说明请看 1.2.1 数据加解密

不加密的数据格式例子如下：

以下例子为事件调用回调结果

```
iot/thing/cmd/up/${productId}/${deviceName}
{
  "id": "123456",
  "version": 1,
  "time": 1521234567000
  "code": 0
  "data": {
    "output" : { "result" : 0}
  }
}
```

字段说明：

名字	类型	必填	说明	备注
id	string	是	消息 ID 号。String 类型的数字，取值范围 0~4294967295，且每个消息 ID 在当前设备中具有唯一性。对于配对的指令，比如 ntp 时间的请求和返回，id 要保持一致。	
version	int	是	版本号，数据协议版本号，扩展用	目前填 1
time	long	是	13 位的时间戳长整型	
code	int	否	结果码，设备数据上传下发成对出现的时候，返回的数据需要填写。具体含义见 1.3 统一结果码	
data	json 加密	否	存放通信数据，有下面三种情况： 1 数据为对象类型，表示为 JSONObject，具体定义要根据接口而定。 2 数据为数组类型，表示为 JSONArray，表示结果是一系列值，值的类型要根据接口而	需要通过 AES 加密

1.2.1 数据加解密

加解密方式：AES

加解密模式：CBC

填充方式：pkcs7padding

偏移量 IV： 16 字节，时间戳字段（time）转字符，不够 16 字节在前面添加"a"。比如例子时间戳 1521234567000，转成字符串后"1521234567000"，不够 16 字节前面补"a"，补位后的偏移量为"aaa1521234567000"。

加密密钥：长度 128bit（16 字节）。加密密钥根据不同的指令选择产品密钥，芯片密钥，设备密钥（指令都有说明使用哪种密钥）。如果选择的密钥长度超过 16 字节，截取前面 16 字节。如果选择的密钥低于 16 字节，请在前面补"a"。

1.3 统一结果码

码值	描述
0	成功
1	通用失败
2	配网 token 过期
3	密钥错误
4	请求超时
5	数据解析错误
6	请求参数错误
7	请求数据校验失败
8	请求上传文件失败
9	文件上传成功
10	文件下载完成
11	文件数据异或校验失败
12	未按门铃

13	按门铃时间超时
14	base64 数据解码错误
15	重新获取业务服务器地址
16	固件无需升级
17	固件校验错误
18	文件长度偏移量不匹配
19	文件上传会话 ID 不匹配
20	设备已禁用
21	暂时无法分配业务 IP，可能服务器已经超负荷

1.4 服务器说明

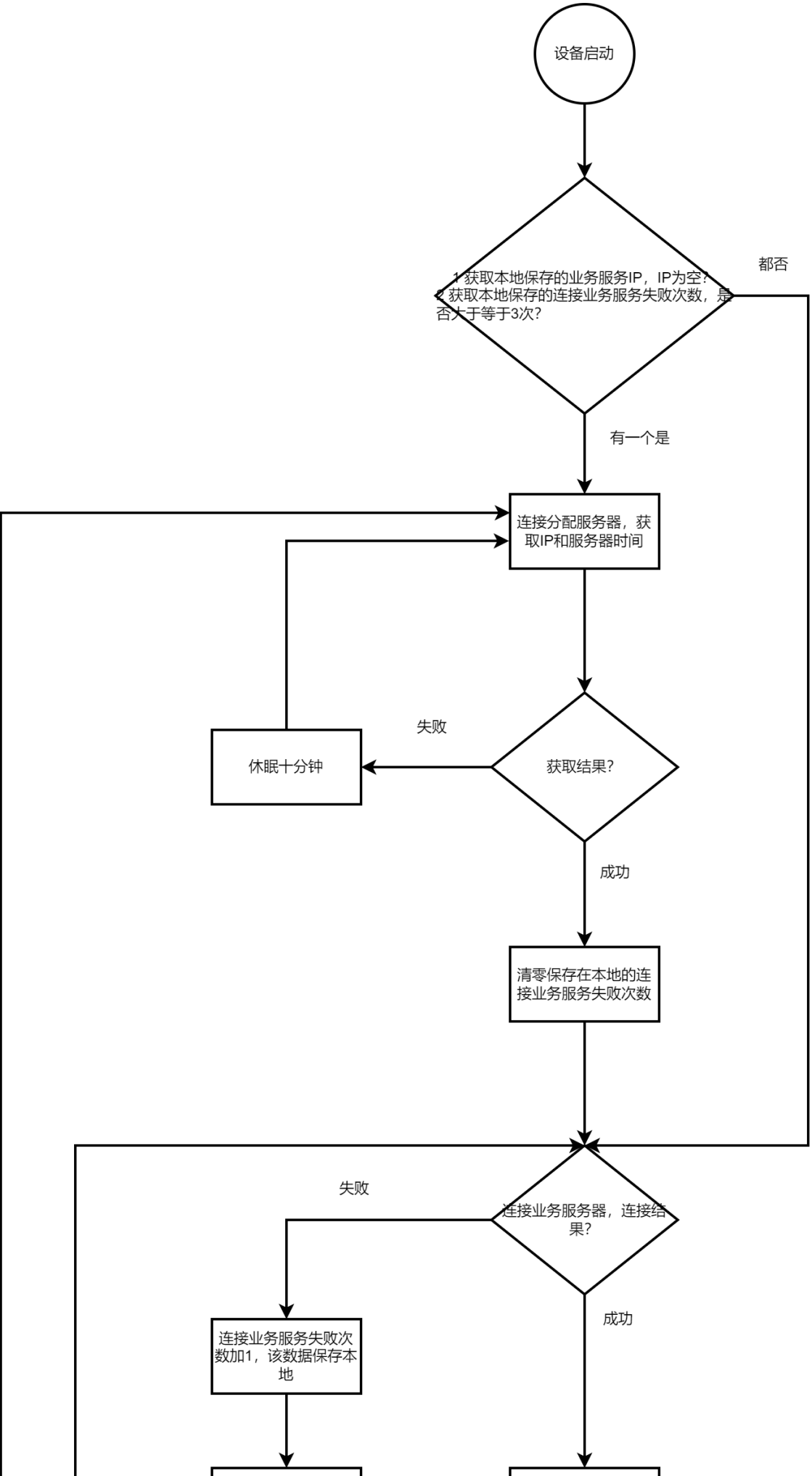
1.4.1 服务器分类

服务器分为两种，**分配服务器**和**业务服务器**。

分配服务器是为设备分配链接业务服务器的 IP 和端口。业务服务器主要处理设备的具体业务。

分配服务器域名：mqtt.iot.yuntianzn.net，端口：8883

设备连接 MQTT broker 推荐流程：



1.4.2 服务器链接认证

物联网平台创建产品时会生成 `productId` 和 `productSecret`，在该产品下创建设备会获取到 `deviceName` 和 `deviceSecret`。如果设备通过平台认证的 WIFI 模组进行数据传输，芯片内部会包含芯片认证信息的 `chipKey`，`chipSecret`，平台颁发的授权码 `authCode`（WIFI 模组的授权码在出厂的时候单独烧录，做到每块模组的授权码唯一性）。设备链接到物联网平台需使用以上信息进行身份认证。针对不同的设备类型和安全等级，平台提供以下三种设备链接认证方案：

（1）**一机一密**：每台设备生产的时候直接烧录自己的设备证书（`productId`、`deviceName` 和 `deviceSecret`）。

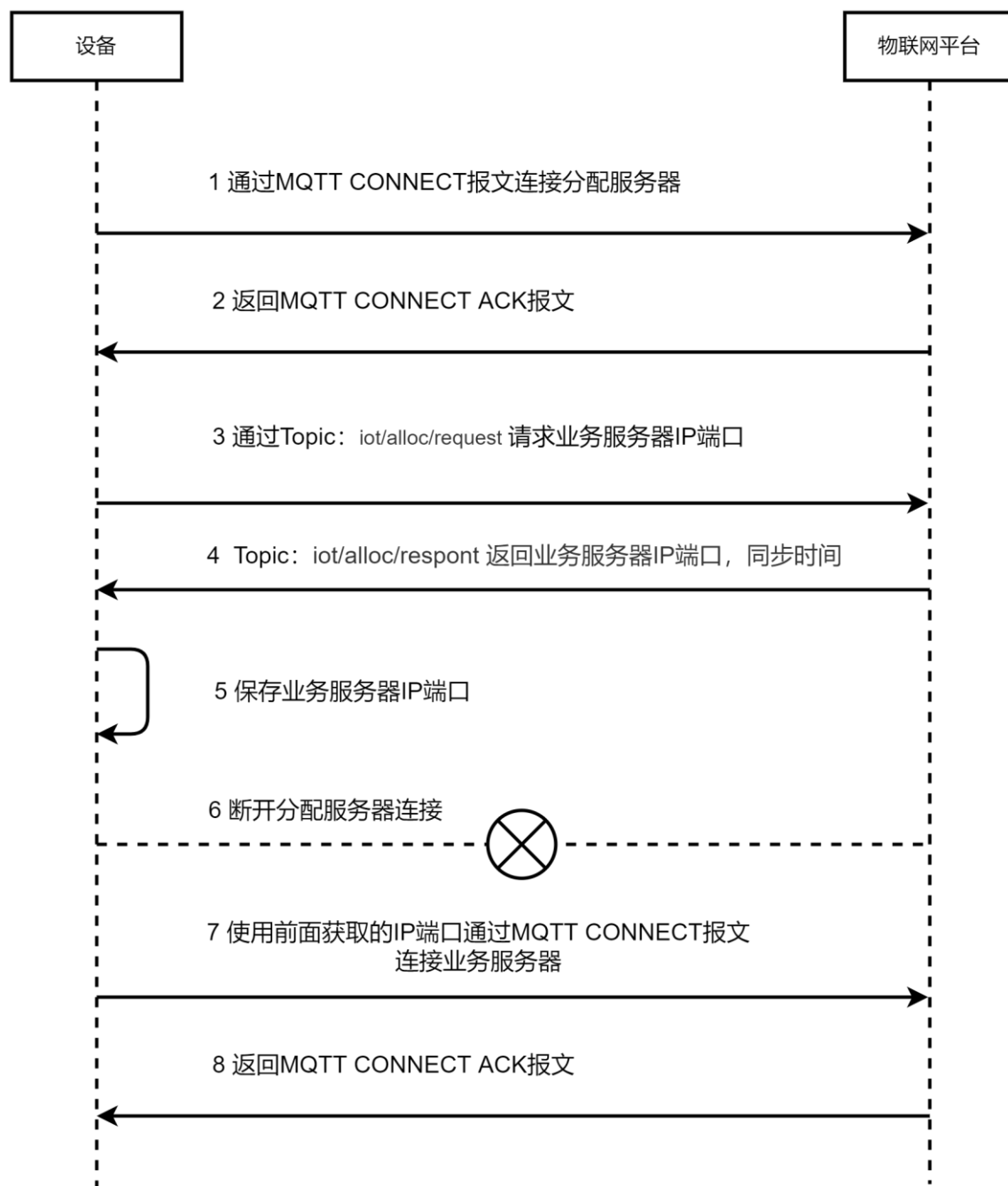
（2）**一型一密预注册**：同一产品下设备烧录相同产品证书（`deviceName`，`productId` 和 `productSecret`）。在物联网平台开通产品的动态注册功能后设备通过动态注册获取 `deviceSecret`。

（3）**模组免预注册**：物联网平台认证的模组，携带模组证书（`chipKey`，`chipSecret`）和授权码（`authCode`）向物联网平台动态注册获取 `deviceName` 和 `deviceSecret`。注意，该模式下设备尚未添加，在注册的过程动态添加。

1.4.2.1 一机一密

该方式下已经拿到了设备密钥 `deviceSecret`，所以只需要直接连接分配服务器获取业务服务器 IP 端口即可。

流程如下：



1 连接分配服务器，发送 CONNECT 报文，报文中包含动态分配服务器认证参数，请求建立连接。认证参数如下：

clientId，客户端 ID：deviceName

username，连接用户名：version + "&" + connectMethod + "&" + signMethod + "&" + random + "&" + productId + "&" + deviceName

password，连接密码：Base64 (MD5(content + "&" + deviceSecret))

参数	说明	备注
productId	产品 ID，产品唯一识别，在创建产品的时候生成	
productSecret	产品密钥，在创建产品的时候生成	
deviceName	设备 ID，设备唯一识别，添加设备的时候生成	
deviceSecret	设备密钥，设备校验使用，添加设备的时候生成	
random	自定义随机数。16 位随机数字和字母组合	
signMethod	签名算法。后续会支持三种：md5、sha1、sha256。目前只支持 md5。	填 "md5"
connectMethod	连接方式，"device"一机一秘连接，"product"一型一秘连接， "chip"模组连接。分配服务器三种连接方式都支持，业务服务器只支持"device"。	填 "device"
version	链接协议版本	填"1"
content	<p>content 的值是提交给服务器的必需参数和值。</p> <p>(version , connectMethod , signMethod , random , productId , deviceName) 按照字母 ASC 码顺序排序、拼接（无拼接符号）的字符串。比如：</p> <p>version : "1"</p> <p>connectMethod : "device"</p> <p>signMethod : "md5"</p> <p>random: "1234567890123456"</p> <p>productId : "YTL-WB01"</p> <p>deviceName : "anzh0102191101000000"</p> <p>那么 content 等于：</p> <p>"connectMethoddeviceNameanzh0102191101000000"</p>	

	productIdYTL- WB01random1234567890123456signMethodmd5version1"	
MD5	哈希 MD5 算法，加密内容文本使用 utf8 编码	
Base64	Base64 编码	

2 MQTT 服务通过 CONNECT ACK 返回认证结果，如果失败则 MQTT 服务器连接失败。

3 设备发起获取业务服务器 IP 端口请求。

4 MQTT 服务端返回业务服务器 IP，端口，服务器时间戳。

5 设备端保存返回的业务服务器 IP 端口和时间戳。

6 设备断开分配服务器连接。

7 连接业务服务器。认证参数如下：

clientId，客户端 ID：deviceName

username，连接用户名：version + "&" + connectMethod + "&" + signMethod + "&" + random + "&" + productId + "&" + deviceName + "&" + timestamp

password，连接密码：Base64 (MD5(content + "&" + deviceSecret))

参数	说明	备注
productId	产品 ID，产品唯一识别，在创建产品的时候生成	
productSecret	产品密钥，在创建产品的时候生成	
deviceId	设备 ID，设备唯一识别，添加设备的时候生成	
deviceSecret	设备密钥，设备校验使用，添加设备的时候生成	
random	您自定义随机数。16 位随机数字和字母组合	

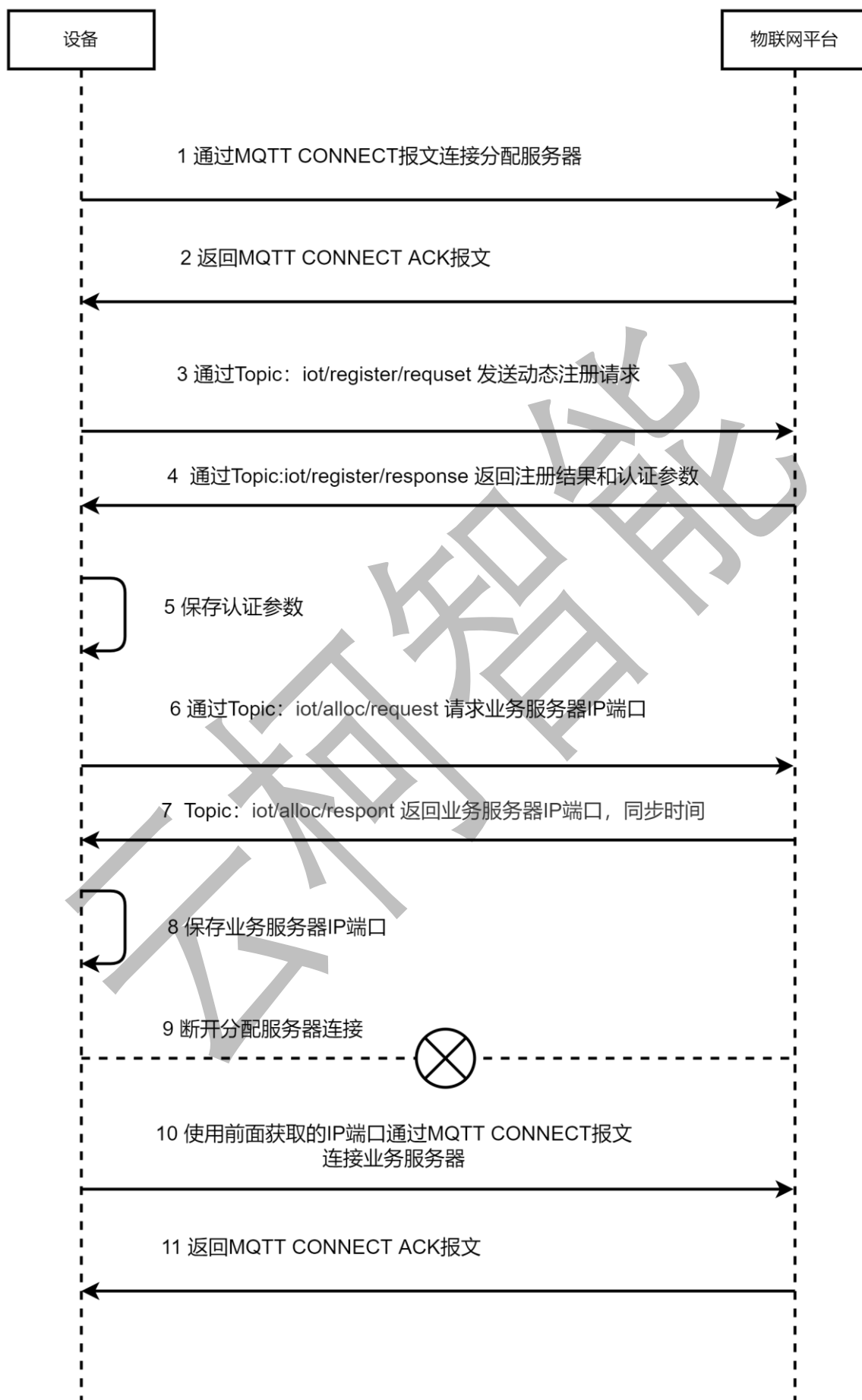
signMethod	签名算法。后续会支持三种：md5、sha1、sha256。目前只支持md5，填“md5”。	填 "md5"
connectMethod	连接方式，“device”一机一秘连接，“product”一型一秘连接，“chip”模组连接。分配服务器三种连接方式都支持，业务服务器只支持“device”。	填 "device"
timestamp	时间戳，13 位整数。如果时间戳和服务器时间前后相差 5 分钟以上，则服务器拒绝连接。所以如果连接业务服务器失败，请重新连接分配服务器获取服务器时间戳或者使用 http 协议获取时间戳（协议见物联网设备 HTTP 协议）。	
content	<p>content 的值是提交给服务器的必需参数和值。</p> <p>(version , connectMethod , signMethod , random , productId , deviceName) 按照字母 ASC 码顺序排序、拼接（无拼接符号）的字符串。比如：</p> <p>version : "1"</p> <p>connectMethod : "device"</p> <p>signMethod : "md5"</p> <p>random: "1234567890123456"</p> <p>productId : "YTL-WB01"</p> <p>deviceName : "anzh0102191101000000"</p> <p>timestamp : "1671075531322"</p> <p>那么 content 等于：</p> <p>"connectMethoddeviceNameanzh0102191101000000productIdYTL-WB01random1234567890123456signMethodmd5timestamp1671075531322version1"</p>	

MD5	哈希 MD5 算法，加密内容文本必须 utf8 编码	
Base64	Base64 编码	

8 返回业务服务连接结果

1.4.2.2 一型一密预注册

要支持这种方式，需要在物联网后台开启动态注册功能。流程如下：



1 连接分配服务器，发送 CONNECT 报文，报文中包含动态分配服务器认证参数，请求建立连接。认证参数如下：

clientId，客户端 ID：deviceName

username，连接用户名：version + "&" + connectMethod + "&" + signMethod + "&" + random + "&" + productId + "&" + deviceName

password，连接密码：Base64 (MD5(content + "&" + productSecret))

参数	说明	备注
productId	产品 ID，产品唯一识别，在创建产品的时候生成	
productSecret	产品密钥，在创建产品的时候生成	
deviceName	设备 ID，设备唯一识别，添加设备的时候生成	
deviceSecret	设备密钥，设备校验使用，添加设备的时候生成	
random	您自定义随机数。16 位随机数字和字母组合	
signMethod	签名算法。后续会支持三种：md5、sha1、sha256。目前只支持 md5。	填 "md5"
connectMethod	连接方式，"device"一机一秘连接，"product"一型一秘连接，"chip"模组连接。分配服务器三种连接方式都支持，业务服务器只支持"device"。	填 "product"
version	链接协议版本	填 "1"
content	content 的值是提交给服务器的必需参数和值。 (version , connectMethod , signMethod , random , productId , deviceName) 按照字母 ASC 码顺序排序、拼接（无拼接符号）的字符串。比如： version : "1"	

	<p>connectMethod : "device"</p> <p>signMethod : "md5"</p> <p>random: "1234567890123456"</p> <p>productId : "YTL-WB01"</p> <p>deviceName : "anzh0102191101000000"</p> <p>那么 content 等于 :</p> <p>"connectMethoddeviceNameanzh0102191101000000 0productIdYTL- WB01random1234567890123456signMethodmd5version1"</p>	
MD5	哈希 MD5 算法，加密内容文本必须 utf8 编码	
Base64	Base64 编码	

2 MQTT 服务通过 CONNECT ACK 返回认证结果，如果失败则 MQTT 服务器连接失败。

3 设备发起动态注册请求，参数请参考动态注册指令。

4 MQTT 服务端返回设备密钥。

5 设备端保存返回的设备密钥 deviceSecret。

6 设备发起获取业务服务器 IP 端口请求。

7 MQTT 服务端返回业务服务器 IP，端口和服务器时间戳

8 设备端保存返回的业务服务器 IP，端口和时间戳。

9 设备断开分配服务器连接。

10 连接业务服务器。认证参数如下：

clientId，客户端 ID : deviceName

username，连接用户名 : version + "&" + connectMethod + "&" + signMethod + "&" + random + "&" + productId + "&" + deviceName + "&" + timestamp

password，连接密码 : Base64 (MD5(content + "&" + deviceSecret))

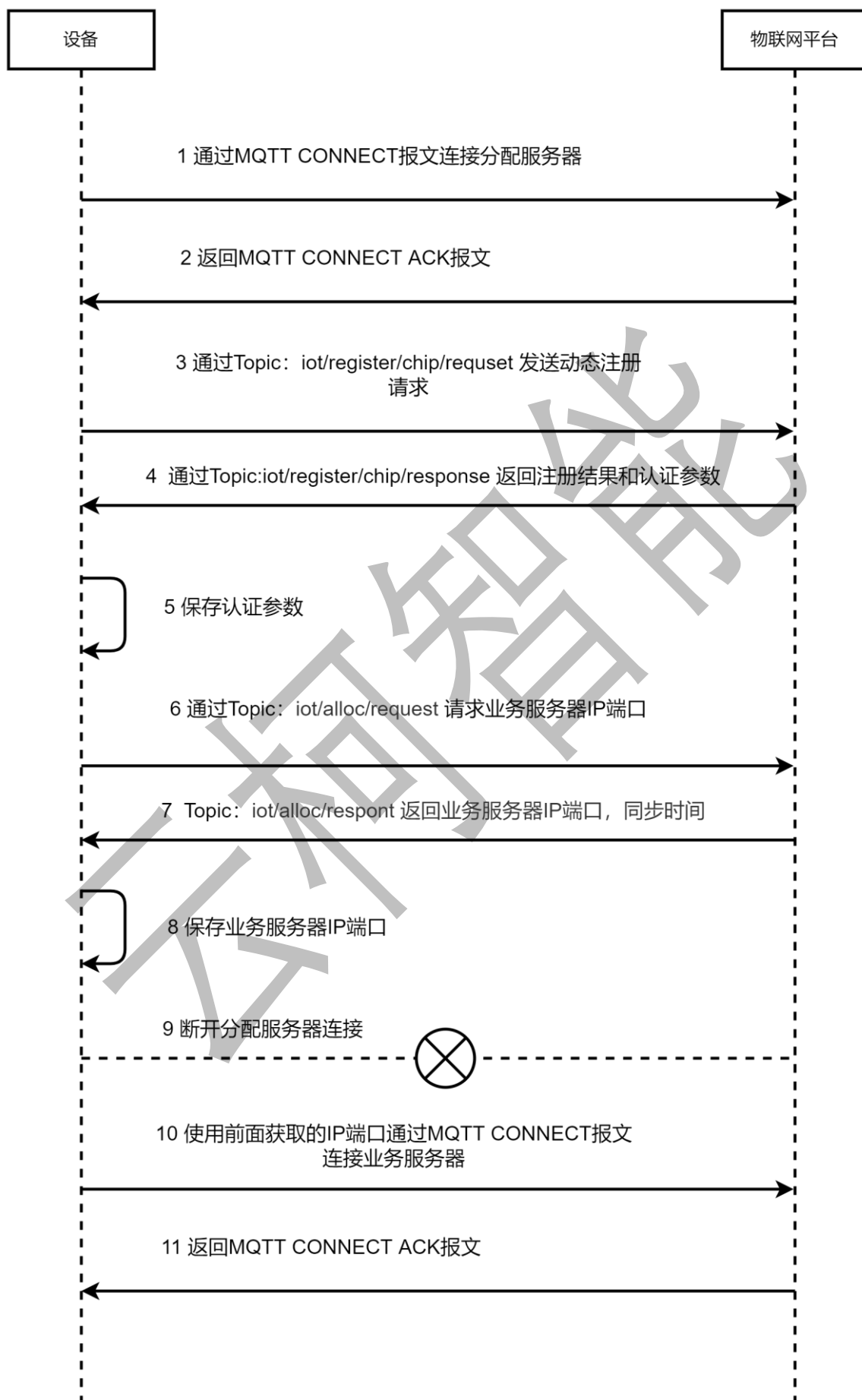
参数	说明	备注
productId	产品 ID，产品唯一识别，在创建产品的时候生成	
productSecret	产品密钥，在创建产品的时候生成	
deviceId	设备 ID，设备唯一识别，添加设备的时候生成	
deviceSecret	设备密钥，设备校验使用，添加设备的时候生成	
random	您自定义随机数。16 位随机数字和字母组合	
signMethod	签名算法。后续会支持三种：md5、sha1、sha256。目前只支持 md5，填“md5”。	填 "md5"
connectMethod	连接方式，“device”一机一秘连接，“product”一型一秘连接，“chip”模组连接。分配服务器三种连接方式都支持，业务服务器只支持“device”。	填 "device"
version	链接协议版本	填 "1"
timestamp	时间戳，13 位整数。如果时间戳和服务端时间前后相差 5 分钟以上，则服务器拒绝连接。所以如果连接业务服务器失败，请重新连接分配服务器获取服务器时间戳或者使用 http 协议获取时间戳（协议见物联网设备 HTTP 协议）。	
content	content 的值是提交给服务器的必需参数和值。 (version , connectMethod , signMethod , random , productId , deviceId) 按照字母 ASC 码顺序排序、拼接（无拼接符号）的字符串。比如：	

	<div>version : "1"</div> <div>connectMethod : "device"</div> <div>signMethod : "md5"</div> <div>random: "1234567890123456"</div> <div>productId : "YTL-WB01"</div> <div>deviceName : "anzh0102191101000000"</div> <div>timestamp : "1671075531322"</div> <div>那么 content 等于 :</div> <div>"connectMethoddeviceNameanzh0102191101000000pr oductIdYTL- WB01random1234567890123456signMethodmd5timestamp1 671075531322version1"</div>	
MD5	哈希 MD5 算法，加密内容文本必须 utf8 编码	
Base64	Base64 编码	

11 返回业务服务连接结果

1.4.2.3 模组免预注册

流程如下：



1 连接分配服务器，发送 CONNECT 报文，报文中包含动态分配服务器认证参数，请求建立连接。认证参数如下：

clientId，客户端 ID：authCode

username，连接用户名：version + "&" + connectMethod + "&" + signMethod + "&" + random + "&" + chipKey + "&" + authCode

password，连接密码：Base64 (MD5(content + "&" + chipSecret))

参数	说明	备注
productId	产品 ID，产品唯一识别，在创建产品的时候生成	
productSecret	产品密钥，在创建产品的时候生成	
deviceId	设备 ID，设备唯一识别，添加设备的时候生成	
deviceSecret	设备密钥，设备校验使用，添加设备的时候生成	
chipKey	设备 WIIF 模组标识，在创建芯片模组的时候生成	
chipSecret	设备 WIIF 模组密钥，在创建芯片模组的时候生成，16 位的字母和数字组成	
authCode	授权码，物联网平台申请生成，32 位的 UUID，每一台设备生产的时候会烧录到设备端	
random	您自定义随机数。16 位随机数字和字母组合	

m		
signMethod	签名算法。后续会支持三种：md5、sha1、sha256。目前只支持 md5。	填 "md5"
connectMethod	连接方式，"device"一机一秘连接，"product"一型一秘连接，"chip"模组连接。分配服务器三种连接方式都支持，业务服务器只支持"device"。	填 "chip"
version	链接协议版本	填 "1"
content	<p>content 的值是提交给服务器的必需参数和值。</p> <p>(version , connectMethod , signMethod , random , productId , deviceName) 按照字母 ASC 码顺序排序、拼接（无拼接符号）的字符串。比如：</p> <p>version : "1"</p> <p>connectMethod : "device"</p> <p>signMethod : "md5"</p> <p>random: "1234567890123456"</p> <p>chipKey : "AC7916AB"</p> <p>authCode : "12345678901234567890123456789012"</p> <p>那么 content 等于：</p> <p>"authCode12345678901234567890123456789012chipKeyAC7916ABconnectMethoddevicerandom1234567890123456signMethodmd5version1"</p>	
MD5	哈希 MD5 算法，加密内容文本必须 utf8 编码	

Base64	Base64 编码	
--------	-----------	--

2 MQTT 服务通过 CONNECT ACK 返回认证结果，如果失败则 MQTT 服务器连接失败。

3 设备发起动态注册请求，参数请参考动态注册指令。

4 MQTT 服务端返回设备号 deviceName 和设备密钥 deviceSecret。

5 设备端保存返回的设备号和密钥。

6 设备发起获取业务服务器 IP 端口请求。

7 MQTT 服务端返回业务服务器 IP 端口。

8 设备端保存返回的业务服务器 IP 端口。

9 设备断开分配服务器连接。

10 连接业务服务器。认证参数如下：

clientId，客户端 ID：deviceName

username，连接用户名：version + "&" + connectMethod + "&" + signMethod + "&" + random + "&" + productId + "&" + deviceName + "&" + timestamp

password，连接密码：Base64 (MD5(content + "&" + deviceSecret))

参数	说明	备注
productId	产品 ID，产品唯一识别，在创建产品的时候生成	
productSecret	产品密钥，在创建产品的时候生成	
deviceName	设备 ID，设备唯一识别，添加设备的时候生成	
deviceSecret	设备密钥，设备校验使用，添加设备的时候生成	
random	您自定义随机数。16 位随机数字和字母组合	

signMethod	签名算法。后续会支持三种：md5、sha1、sha256。目前只支持md5，填“md5”。	填 "md5"
connectMethod	连接方式，“device”一机一秘连接，“product”一型一秘连接，“chip”模组连接。分配服务器三种连接方式都支持，业务服务器只支持“device”。	填 "device"
version	链接协议版本	填"1"
timestamp	时间戳，13 位整数。如果时间戳和服务端时间前后相差 5 分钟以上，则服务器拒绝连接。所以如果连接业务服务器失败，请重新连接分配服务器获取服务器时间戳或者使用 http 协议获取时间戳（协议见物联网设备 HTTP 协议）。	
content	<p>content 的值是提交给服务器的必需参数和值。</p> <p>（version，connectMethod，signMethod，random，productId，deviceName）按照字母 ASC 码顺序排序、拼接（无拼接符号）的字符串。比如：</p> <pre> version : "1" connectMethod : "device" signMethod : "md5" random: "1234567890123456" productId : "YTL-WB01" deviceName : "anzh0102191101000000" timestamp : "1671075531322" </pre> <p>那么 content 等于：</p> <pre> "connectMethoddeviceNameanzh0102191101000000productIdYTL-WB01random1234567890123456signMethodmd5timestamp1 </pre>	

	671075531322version1"	
MD5	哈希 MD5 算法，加密内容文本必须 utf8 编码	
Base64	Base64 编码	

11 返回业务服务连接结果

1.4.3 MQTT 协议规范

（1）支持版本：

目前物联网平台支持 MQTT 标准协议接入，兼容 5.0、3.1.1 和 3.1 版本协议，具体的协议请参见 MQTT 5.0、MQTT 3.1.1 和 MQTT 3.1 协议文档。

（2）与标准 MQTT 的区别：

支持 MQTT 的 PUB、SUB、PING、PONG、CONNECT、DISCONNECT 和 UNSUB 等报文。

支持 clean session。**请填写为支持。**

不支持 will、retain msg。

支持 QoS 0、QoS 1，不支持 QoS 2。**请尽量都用 QoS0**

不支持 SUB QoS，消息 QoS 以发送方（PUB）指定为准。

（3）连接参数

CientID，Username，Password 请参考**服务器连接认证**填写。

（4）安全等级

TCP 通道 TLS 直连模式：安全级别高。

TCP 直连模式（数据不加密）：不安全，明文传输。

目前由于嵌入式设备的资源限制，目前的版本**只支持 TCP 直连模式。**

（5）MQTT 保活

设备端在保活时间间隔内，至少需要发送一次报文，包括 ping 请求。

连接保活时间的取值范围为 30 秒~1200 秒。建议取值 300 秒以上。

从物联网平台发送 CONNACK 响应 CONNECT 消息时，开始心跳计时。收到 PUBLISH、SUBSCRIBE、PING 或 PUBACK 消息时，会重置计时器。物联网平台每隔 30 秒定时检测一次设备的保活心跳，设备上线时间点距离最新定时检测时间点的时间，是定时检测的等待时间。定义最大超时时间为：保活心跳时间*1.5+定时检测的等待时间。超过最大超时时间未收到设备消息，服务器会自动断开连接。

带有休眠省电性质的设备，建议取值 30 秒，减少因为频繁断开重连导致 socket 的 WaitTime 过长。

（6）MQTT 域名

分配服务器域名(www.smartke.net/iot/mqtt),端口(1883)。

业务服务器域名端口由分配服务器返回。

2 功能指令

2.1 动态注册

注册的流程请参考 1.4.2 服务器连接认证。

2.1.1 一型一密预注册

描述：设备注册。

Topic	iot/register/product/request/\${productId}/\${deviceName}
方向	设备 ==> 云端

版本	1
备注	只支持分配服务器。 数据加密使用 productSecret。

data 数据内容：

字段	必填	类型	描述	备注
type	是	int	动态注册方式： 0 预注册 1 免预注册 免预注册指的是事先不需要在物联网平台注册添加设备。动态根据传输的 DeviceName 在平台生成该设备。安全性要求必须 TLS。目前不支持该方式。	填 0
productId	是	string	产品标识	
deviceName	否	string	设备唯一标识	

服务器返回：

Topic	iot/register/product/response/\${ProductId}/\${DeviceName}
方向	云端 ==> 设备
版本	1
备注	只支持分配服务器。 数据 id 字段要和请求一致。 数据加密使用 productSecret。

data 数据内容：

字段	必填	类型	描述	备注
deviceName	是	string	设备唯一标识	
productId	是	string	产品标识	

deviceSecret	是	字符串	设备密钥	
--------------	---	-----	------	--

结果码 code：

码值	描述	备注
0	成功	
1	失败	
n	其他失败	1.3 统一结果码表

2.1.2 模组免预注册

描述：设备注册。

Topic	iot/register/chip/request/\${chipKey}/\${authCode}
方向	设备 ==> 云端
版本	1
备注	只支持分配服务器。 topic 的后面连个层级分别是 chipKey 和 authCode。 数据加密使用 chipSecret。

data 数据内容：

字段	必填	类型	描述	备注
productId	是	string	产品标识	
deviceName	否	string	设备唯一标识，如果注册的时候没有带上该项，则物联网平台会随机生成 deviceName 返回。	

服务器返回：

Topic	iot/register/chip/response/\${chipKey}/\${authCode}
方向	云端 ==> 设备
版本	1

备注	<p>只支持分配服务器。</p> <p>数据 id 字段要和请求一致。</p> <p>请注意 topic 的后面连个层级分别是 chipKey 和 authCode。</p> <p>数据加密使用 chipSecret</p>
----	--

data 数据内容：

字段	必填	类型	描述	备注
deviceName	是	string	设备唯一标识	
productId	是	string	产品标识	
deviceSecret	是	字符串	设备密钥	

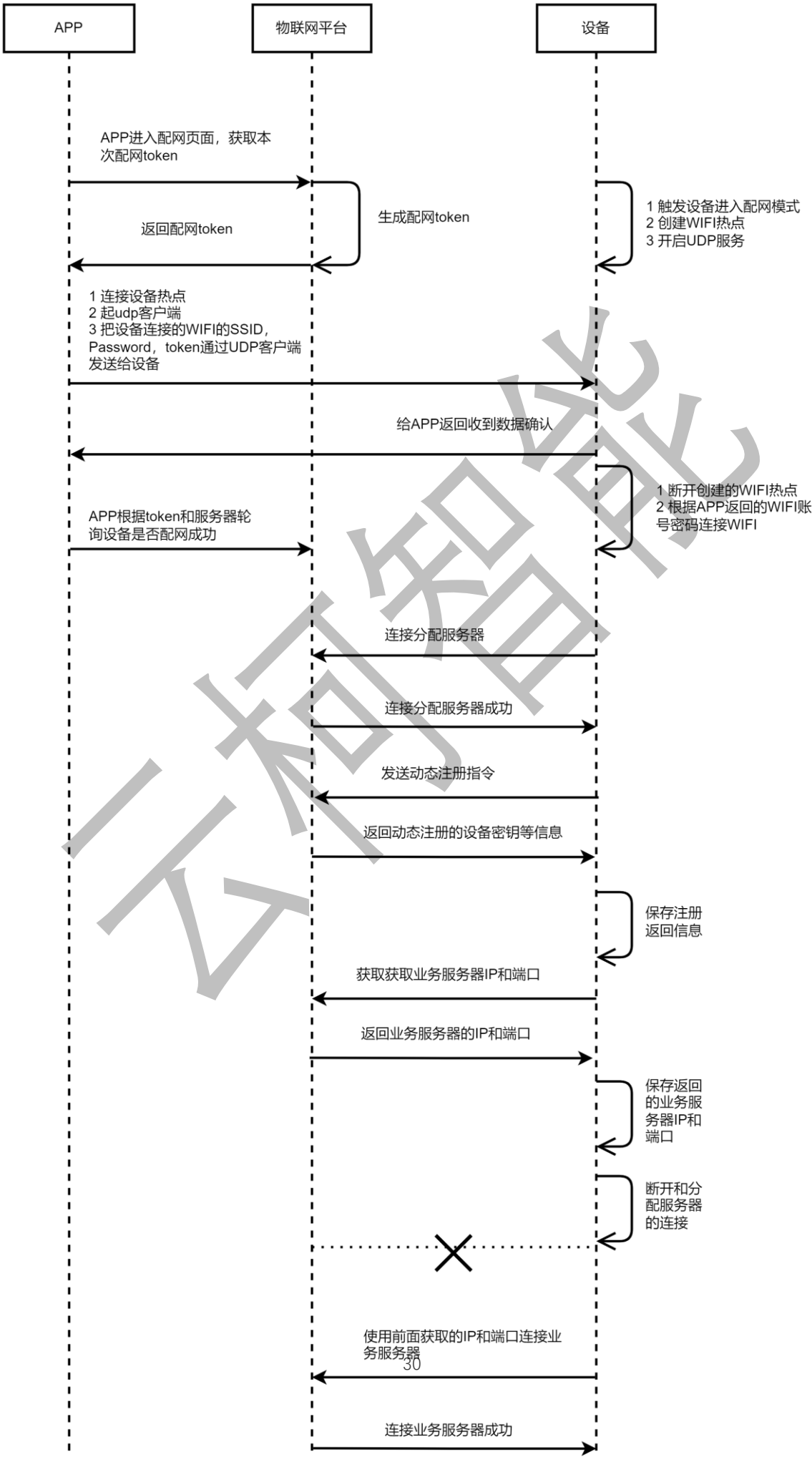
结果码 code：

码值	描述	备注
0	成功	
1	失败	
n	其他失败	1.3 统一结果码表

2.2 配网绑定

2.1.2 猫眼模组 + 第三方锁板（涂鸦）

流程图：



提交配网指令：

Topic	iot/pair/request/\${productId}/\${deviceName}
方向	设备 ==> 云端
版本	1
备注	只支持业务服务器。

data 数据内容：

字段	必填	类型	描述	备注
token	是	string	本次配网操作的 token	
deviceName	是	string	设备 id	
productId	是	string	产品 id	
versions	否	json 数组	选填，设备版本信息，例如： <pre>[{ "version": "1.1.12", "module": "WIFI" }, { "version": "1.1.0", "module": "CPU" },]</pre> 其中： module：模块类型 "WIFI" WIFI 模组， "MCU" MCU 模块， "EXPAND" 拓展模块， "OTHER" 第三方模块，	

			version：版本号版本号采用 xx.yy.zz 点分表示，范围支持 0.0.0 到 99.99.99。例如：1.0.0	
--	--	--	---	--

配网回执：

Topic	iot/pair/response/\${productId}/\${deviceName}
方向	云端 ==> 设备
版本	1
备注	只支持业务服务器。

data 数据内容：

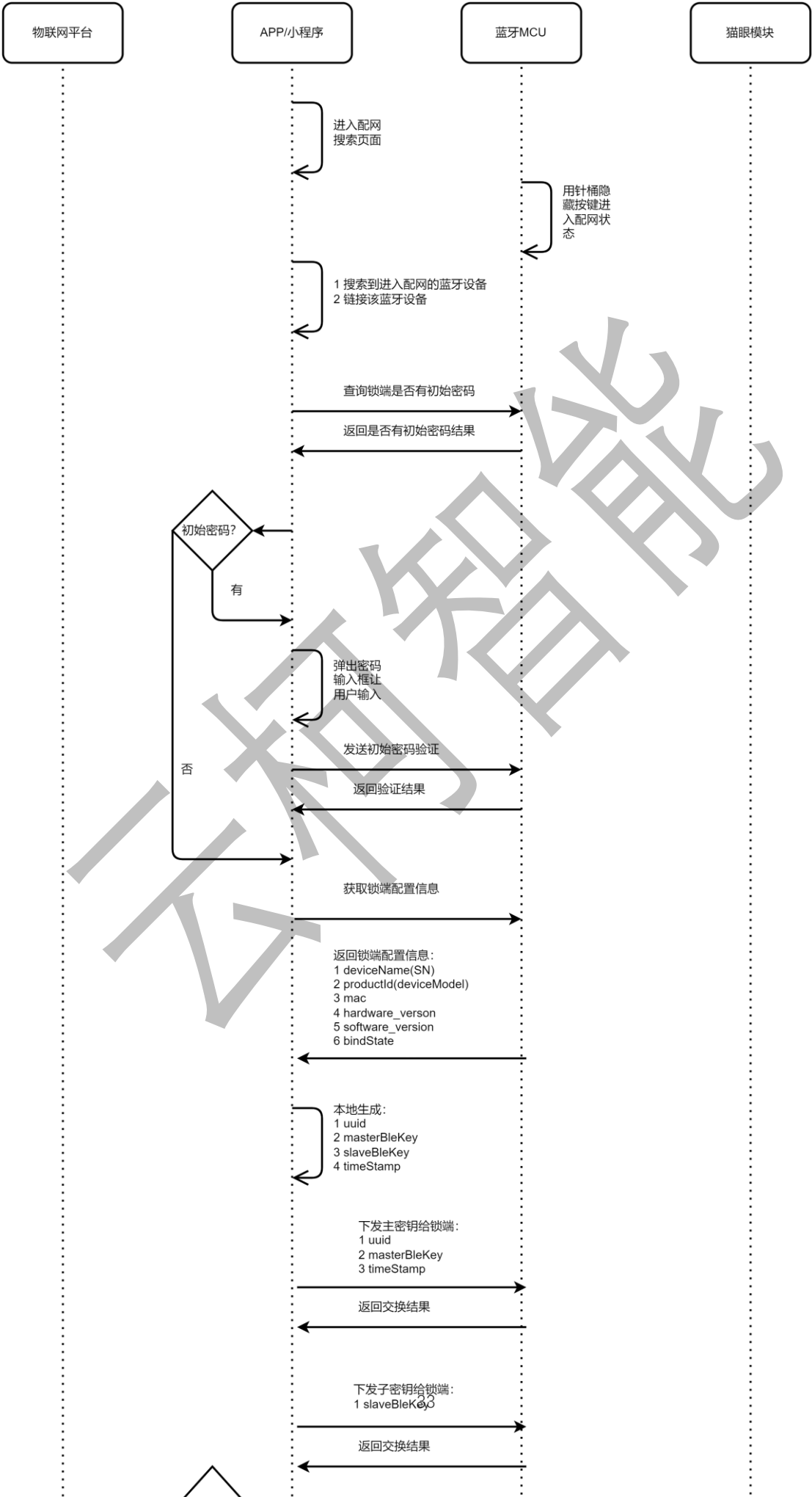
字段	必填	类型	描述	备注

结果码 code：

码值	描述	备注
0	成功	
1	失败	
n	其他失败	1.3 统一结果码表

2.2.2 猫眼模组 + 云柯蓝牙锁板

流程图：



2.3 获取业务服务 IP 和服务时间

描述：设备数据量剧增到一定量，单台服务器将不负重荷，为了分担压力，会部署多台服务器。设备需要调用该指令和分配服务器获取自己需要连接的业务服务器 IP 和端口。该接口同时返回服务器的时间戳。

推荐更新策略：请求获取到分配的业务服务器 IP 后，后续业务都连该 IP。如果连不上分配的 IP 服务器，则对比上次请求的时间，如果超过一定时间（推荐一天），重新和分配服务器获取新的业务服务器 IP。

2.3.1 一机一密 & 一型一密

Topic	iot/alloc/request/\${productId}/\${deviceName}
方向	设备 ==> 云端
版本	1
备注	只支持分配服务器。 数据加密使用 deviceSecret。

data 数据内容：

字段	必填	类型	描述	备注
deviceSendTime	是	long	设备发送时间，13 位（毫秒）	
deviceType	否	int	设备信息 0 休眠设备 1 低功耗不休眠设备	不填则默认 0

响应：

Topic	iot/alloc/response/\${productId}/\${deviceName}
方向	云端 ==> 设备
版本	1

备注	只支持分配服务器。 数据加密使用 deviceSecret。 数据 id 字段要和请求一致。
----	--

data 数据内容：

字段	必填	类型	描述	备注
ip	是	string	"106.52.172.119:1883"	目前默认
deviceSendTime	是	long	设备发送时间，13 位（毫秒）	
serverRecvTime	是	long	服务器收到时间，13 位（毫秒）	
serverSendTime	是	long	服务器发送时间，13 位（毫秒）	

结果码 code：

码值	描述	备注
0	成功	
1	失败	
21	暂时无法分配业务 IP，可能服务器已经超负荷	请设备休眠一段时间再请求
n	其他失败	

2.3.2 模组免预注册

Topic	iot/alloc/chip/request/\${chipKey}/\${authCode}
方向	设备 ==> 云端
版本	1
备注	只支持分配服务器。 数据加密使用 chipSecret。

data 数据内容：

字段	必填	类型	描述	备注
----	----	----	----	----

deviceSendTime	是	long	设备发送时间，13 位（毫秒）	
deviceType	否	int	设备信息 0 休眠设备 1 低功耗不休眠设备	不填则默认 0

响应：

Topic	iot/alloc/chip/response/\${chipKey}/\${authCode}
方向	云端 ==> 设备
版本	1
备注	只支持分配服务器。 数据加密使用 chipSecret。 数据 id 字段要和请求一致。

data 数据内容：

字段	必填	类型	描述	备注
ip	是	string	"106.52.172.119:1883"	目前默认
deviceSendTime	是	long	设备发送时间，13 位（毫秒）	
serverRecvTime	是	long	服务器收到时间，13 位（毫秒）	
serverSendTime	是	long	服务器发送时间，13 位（毫秒）	

结果码 code：

码值	描述	备注
0	成功	
1	失败	
21	暂时无法分配业务 IP，可能服务器已经超负荷	请设备休眠一段时间再请求
n	其他失败	

2.4 获取服务器时间

描述：同步设备时间。物联网平台提供的 NTP 服务，借鉴 NTP 协议原理，将物联网平台作为 NTP 服务器。高精度度的时间校正流程如下：

1. 设备端通过指定 Topic 向物联网平台发送消息，会携带发送时间 **deviceSendTime**。
2. 物联网平台接收设备端消息后，回复消息中，会增加接收消息的时间 **serverRecvTime** 和回复消息的时间 **serverSendTime**。
3. 设备端接收到物联网平台回复，会根据本地时间，给出接收回复的时间 **deviceRecvTime**。
4. 根据以上出现的 4 个时间，计算设备端与物联网平台的时间差，得出设备端获取到的服务端当前的精确时间。设备上的精确时间为：

$$(\text{serverRecvTime} + \text{serverSendTime} + \text{deviceRecvTime} - \text{deviceSendTime}) / 2$$

2.4.1 一机一密

Topic	iot/ntp/device/request/\${productId}/\${deviceName}
方向	设备 ==> 云端
版本	1
备注	同时支持分配服务器和业务服务器。 分配服务器数据加密使用 deviceSecret 。

data 数据内容：

字段	必填	类型	描述	备注
deviceSendTime	是	long	设备发送时间，13 位（毫秒）	

服务器返回：

Topic	iot/ntp/device/response/\${ProductId}/\${DeviceName}
方向	云端 ==> 设备

版本	1
备注	同时支持分配服务器和业务服务器。 分配服务器数据加密使用 deviceSecret。 数据 id 字段要和请求一致。

data 数据内容：

字段	必填	类型	描述	备注
deviceSendTime	是	long	设备发送时间，13 位（毫秒）	
serverRecvTime	是	long	服务器收到时间，13 位（毫秒）	
serverSendTime	是	long	服务器发送时间，13 位（毫秒）	

结果码 code：

码值	描述	备注
0	成功	
1	失败	
n	其他失败	1.3 统一结果码表

2.4.2 一型一密

Topic	iot/ntp/product/request/\${productId}/\${deviceName}
方向	设备 ==> 云端
版本	1
备注	同时支持分配服务器和业务服务器。 分配服务器数据加密使用 productSecret。

data 数据内容：

字段	必填	类型	描述	备注
deviceSendTime	是	long	设备发送时间，13 位（毫秒）	

服务器返回：

Topic	iot/ntp/product/response/\${ProductId}/\${DeviceName}
方向	云端 ==> 设备
版本	1
备注	同时支持分配服务器和业务服务器。 分配服务器数据加密使用 productSecret。 数据 id 字段要和请求一致。

data 数据内容：

字段	必填	类型	描述	备注
deviceSendTime	是	long	设备发送时间，13 位（毫秒）	
serverRecvTime	是	long	服务器收到时间，13 位（毫秒）	
serverSendTime	是	long	服务器发送时间，13 位（毫秒）	

结果码 code：

码值	描述	备注
0	成功	
1	失败	
n	其他失败	1.3 统一结果码表

2.4.3 模组免预注册

Topic	iot/ntp/chip/request/\${chipKey}/\${authCode}
方向	设备 ==> 云端
版本	1
备注	同时支持分配服务器和业务服务器。 分配服务器数据加密使用 chipSecret。

data 数据内容：

字段	必填	类型	描述	备注
----	----	----	----	----

deviceSendTime	是	long	设备发送时间，13 位（毫秒）	
----------------	---	------	-----------------	--

服务器返回：

Topic	iot/ntp/chip/response/\${chipKey}/\${authCode}
方向	云端 ==> 设备
版本	1
备注	同时支持分配服务器和业务服务器。 分配服务器加密使用 chipSecret。 数据 id 字段要和请求一致。

data 数据内容：

字段	必填	类型	描述	备注
deviceSendTime	是	long	设备发送时间，13 位（毫秒）	
serverRecvTime	是	long	服务器收到时间，13 位（毫秒）	
serverSendTime	是	long	服务器发送时间，13 位（毫秒）	

结果码 code：

码值	描述	备注
0	成功	
1	失败	
n	其他失败	1.3 统一结果码表

2.5 设备文件上传

2.5.1 设备请求上传文件

描述：设备请求上传文件到云端，步骤分为两个，先上传文件初始化信息，然后再上传文件片段。单个文件大小不能超过 16MB。

Topic	iot/file/upload/init/\${productId}/\${deviceName}
-------	---

方向	设备 ==> 云端
版本	1
备注	只支持业务服务器。

数据内容：

字段	必填	类型	描述	备注
file_name	是	string	<p>设备上传文件的名称。限制如下：</p> <p>1 支持数字、英文字母、下划线（_）和英文句点（.）。</p> <p>2 首字符仅支持数字和英文字母。</p> <p>3 长度不超过 100 字节。</p>	
file_size	是	int	<p>上传文件大小，单位字节。单个文件大小不超过 16 MB。取值为-1 时，表示文件大小未知。文件上传完成时，需在上传文件分片的消息中，指定参数 complete，具体说明，请参见设备上传文件分片。</p>	
conflict	否	string	<p>物联网平台对设备上传同名文件的处理策略。非必填参数，默认为 overwrite。使用说明：</p> <p>1 若物联网平台不存在同名文件，直接创建文件的上传任务。</p> <p>2 若物联网平台已存在同名文件，则按照设置的策略，进行如下处理：</p> <p>overwrite：覆盖模式。先删除同名文件，再创建文件的上传任务。</p> <p>append：文件追加模式。若同名文件上传未完成，设备端可根据物联网平台返回的文件信息，继续上传文件。</p> <p>若同名文件上传已完成，创建文件上传任务会失败。设备端可修改文件名称或通过覆盖模式（overwrite）重新请求上传文件。</p> <p>reject：拒绝模式。物联网平台拒绝同名文件上传的</p>	

			<p>请求，并返回文件已存在的错误码。</p> <p>注意 从创建文件上传任务成功开始计时，如果 10 分钟内设备端未完成上传，物联网平台云端会自动删除该文件上传任务。</p>	
fic_mode	否	string	<p>文件的完整性校验模式，目前可取值"BCC"，字节异或校验。非必传参数。</p> <p>若不传入，在文件上传完成后不校验文件完整性。</p> <p>若传入，与 fic_value 同时传入，根据校验模式和校验值校验文件完整性。</p> <p>注意 若 file_size 值为-1，不支持设置 fic_mode 和 fic_value。</p>	
fic_value	否	string	<p>文件的完整性校验值，是 16 位的 Hex 格式编码的字符串。非必传参数。若传入，与 fic_mode 同时传入。</p>	
init_uid	否	string	<p>自定义的设备请求上传文件的任务唯一 ID，同一上传任务请求必须对应相同的唯一 ID。限制如下：</p> <ol style="list-style-type: none"> 1 支持数字、英文字母、短划线 (-)、下划线 (_) 和英文句点 (.)。 2 首字符仅支持数字和英文字母。 3 长度不超过 16 个字符。 <p>该参数可选：</p> <p>传入：用于设备请求上传文件的消息，在重发场景下的幂等性处理。您需确保同一上传文件任务的重试消息使用相同的 init_uid。在同一设备下，该参数相同的两个上传请求消息，会被物联网平台云端识别为消息重试，返回相同的文件请求上传的响应消息。</p> <p>在同一设备下，若重试的时间间隔超过 10 分钟，则重试的请求会识别为新的文件上传任务请求。此时若请求上传文件成功，则返回新的 upload_id。</p> <p>不传入：物联网平台的云端识别当前请求为新的文件上传请求。</p>	
service_id	是	string	<p>文件上传相关的业务 ID。该 ID 需要在物联网平台控制台预先定义。业务如下：</p>	

			"capture": 抓拍上传图片	
--	--	--	-------------------	--

远端返回：

Topic	iot/file/upload/init_reply/\${productId}/\${deviceName}
方向	云端 ==> 设备
版本	1
备注	只支持业务服务器。 id 字段要和请求一致。

数据内容：

字段	必填	类型	描述	备注
upload_id	否	string	如果创建上传任务成功，本项存在。本次上传文件任务的标识 ID。后续上传文件分片时，需要传递该文件标识 ID。	
file_name	否	string	如果创建上传任务成功，本项存在。设备上传的文件名称。	
file_offset	否	int	仅当请求参数 conflict 为 append，且物联网平台云端存在未完成上传的文件时，返回的已上传文件的大小，单位为字节。	

结果码 code：

码值	描述	备注
0	成功	
1	失败	
n	其他失败	1.3 统一结果码表

2.5.2 设备上传文件分片

文件传输和服务器约定好数据后，开始分片上传数据。

Topic	iot/file/upload/send/\${productId}/\${deviceName}
方向	设备 ==> 云端
版本	1
备注	只支持业务服务器。

数据内容：

请注意，由于文件数据各种字符都存在，所以 json 格式不支持。而且如果把文件用 base64 编码，则文件体积会大了三分之一。所以该指令的数据分为两大部分来传输。一部分是数据信息部分，仍采用 json 格式。第二部分采用二进制格式，保存的是文件数据分片。大格式如下：

请求 Header 数据格式		文件分片	文件分片校验值
Header 长度 (2 字节)	header 数据 (n 字节)	文件分片数据 (m 字节)	分片数据异或校验值 (1 字节)
指定 header 数据的字节 n，大端格式	数据信息部分 (JSON 格式)	字节个数 m 由数据信息部分的 slice_size 决定	校验分片完整性

其中，header 数据的格式样例如下：

```

{
  "id": "123456",
  "version": 1
  "time": 1521234567000
  "data":
  {
    "upload_id": "13676289291"
    "slice_offset": "1234"
    "slice_size": "512"
    "is_complete": false
  }
}

```

字段	必填	类型	描述	备注
upload_id	是	string	设备请求上传文件时返回的文件上传任务标识 ID。	
slice_offset	是	int	已上传文件分片的总大小，单位为字节。	
slice_size	是	int	当前上传文件分片的大小，单位为字节。 非最后一个分片时，分片大小范围为 256 B~131072 B。 最后一个文件分片时，若上传的文件大小已知，则分片大小范围为 1 B~131072 B；若上传的文件大小未知，则分片大小范围为 0 B~131072 B。	
is_complete	否	Boolean	<p>仅当设备请求上传文件中 file_size 为-1，即文件大小未知时，该参数有效，表示当前分片是否是文件的最后一个分片。</p> <p>true: 是。此时，物联网平台的云端会校验已上传文件大小是否超过 16 MB:</p> <p>未超过：若文件大小大于 0，则文件上传成功。若文件大小为 0，则返回文件不能为空的错误信息且删除文件上传任务。</p> <p>超过：返回文件大小超过 16 MB 的错误码，并删除已上传的文件。详细说明，请参见设备上传文件相关错误码。</p>	

字段	必填	类型	描述	备注
upload_id	是	string	本次上传文件任务的标识 ID。后续上传文件分片时，需要传递该文件标识 ID。	
slice_offset	是	int	已上传文件分片的总大小，单位为字节。	
slice_size	是	int	当前上传文件分片的大小，单位为字节。 非最后一个分片时，分片大小范围为 256 B~131072 B。 最后一个文件分片时，若上传的文件大小已知，则分片大小范围为 1 B~131072 B；若上传的文件大小未知，则分片大小范围为 0 B~131072 B。	
complete	否	Boolean	当上传了最后一个分片数据后，文件上传完成，返回该参数，值为 true。若设备请求上传文件的请求消息中 fileSize 值大于 0，即文件大小已知时，若已上传的文件大小与设备请求上传文件时的文件大小相同，文件被识别为上传完成。 若设备请求上传的请求消息中 file_size 值为 -1，即文件大小未知时，若文件分片上传请求中 isComplete 存在且值为 true，文件被识别为上传完成。	

结果码 code：

码值	描述	备注
0	成功	
1	失败	
n	其他失败	1.3 统一结果码表

2.5.3 设备取消上传文件

描述：设备取消上传文件到云端的任务

Topic	iot/file/upload/cancel/\${productId}/\${deviceName}
方向	设备 ==> 云端
版本	1
备注	只支持业务服务器。

数据内容：

字段	必填	类型	描述	备注
upload_id	否	string	设备请求上传文件时返回的文件上传任务标识 ID。	

远端返回：

Topic	iot/file/upload/cancel_reply/\${productId}/\${deviceName}
方向	云端 ==> 设备
版本	1
备注	只支持业务服务器。 id 字段要和请求一致。

数据内容：

字段	必填	类型	描述	备注
upload_id	否	string	取消的文件上传任务标识 ID。	

结果码 code：

码值	描述	备注
0	成功	
1	失败	
n	其他失败	1.3 统一结果码表

2.6 设备文件下载

描述：设备端下载文件，请务必使用 http/https 协议下载文件。暂不支持 mqtt 下载。

2.7 设备上报属性

描述：设备上报设备属性，属性必须物模型定义过的，比如开锁方式，电量，开门日志，告警信息等都属于属性

Topic	iot/thing/property/ post /\${productId}/\${deviceName}
方向	设备 ==> 云端
版本	1
备注	只支持业务服务器。

data 数据内容：

字段	必填	类型	描述	备注
properties	是	object	<p>例子 1：</p> <pre> "properties" : { "battery_percentage" : { "value": 99, "time": 1524448722000 }, "battery_level" : { "value": 2, "time": 1524448722000 }, } </pre> <p>例子 2：</p>	

		<pre>"properties" : { "battery_percentage" : 99 , "battery_level" : 2 }</pre> <p>例子 3：透传音视频属性</p> <pre>{ "properties":{ "p2p_info":{ "tx_device_name":"xxx", "tx_product_id":"xxx" } } }</pre> <p>例子中设备上报了的两个属性 battery_percentage（电量百分比）和 battery_level（电量水平）两个属性。</p> <p>两个例子区别：</p> <p>属性上报时间戳，类型为 UTC 毫秒级时间。该 参数为可选字段。根据您的业务场景决定消息中 是否带时间戳。如果消息频繁，需根据时间戳判 断消息顺序，建议消息中带有时间戳。</p> <ul style="list-style-type: none">● 若上传 time，物联网平台的云端保存上传 的时间作为属性上报时间。	
--	--	---	--

			<ul style="list-style-type: none"> ● 若不上传 time，物联网平台的云端自动生成属性上报时间并保存。 	

设备收到确认：

Topic	iot/thing/property/post_reply/\${productId}/\${deviceName}
方向	云端 ==> 设备
版本	1
备注	只支持业务服务器。 数据 id 字段要和请求一致。

data 数据内容：

字段	必填	类型	描述	备注

结果码 code：

码值	描述	备注
0	成功	
1	失败	
n	其他失败	1.3 统一结果码表

2.8 设备读取属性

描述：设备读取平台保存的属性值，属性必须物模型定义过的，比如开锁方式，电量，开门日志，告警信息等都属于属性

Topic	iot/thing/property/get/\${productId}/\${deviceName}
方向	设备 ==> 云端

版本	1
备注	只支持业务服务器。

data 数据内容：

字段	必填	类型	描述	备注
properties	是	array	"properties" : ["battery_percentage" , "battery_level" ,]	battery_percentage 和 battery_level 都在物模 型中有定义

设备收到确认：

Topic	iot/thing/ property /get_reply/\${productId}/\${deviceName}
方向	云端 ==> 设备
版本	1
备注	只支持业务服务器。 数据 id 字段要和请求一致。

data 数据内容：

字段	必填	类型	描述	备注
properties	否	object	如果获取成功，该项有值： "properties" : { "battery_percentage" : 99 , "battery_level" : 2 }	

结果码 code：

码值	描述	备注
0	成功	

1	失败	
n	其他失败	1.3 统一结果码表

2.9 设备上报事件

描述：设备上报事件，事件必须物模型定义过的，比如电量低，解锁错误过多,可疑人在门口晃动等。

Topic	iot/thing/event/post/\${productId}/\${deviceName}
方向	设备 ==> 云端
版本	1
备注	只支持业务服务器。

data 数据内容：

字段	必填	类型	描述	备注
	是	object	门铃例子: <pre>{ "bell": { "params": { "info": { "tx_product_id": "123456", "tx_device_name": "789456" } }, "time": 1524448722000 } }</pre> 其中：	

			<p>bell 物模型中事件的标识。</p> <p>params 物模型中定义的事件输出数据。</p> <p>info 事件输出一个参数的标识，物模型中定义。</p> <p>time 为可选字段。根据您的业务场景决定消息中是否带时间戳。如果不带时间戳，平台会以收到的时间做为事件的时间戳。</p>	
--	--	--	--	--

设备收到确认：

Topic	iot/thing/event/post_reply/\${productId}/\${deviceName}
方向	云端 ==> 设备
版本	1
备注	只支持业务服务器。 数据 id 字段要和请求一致。

data 数据内容：

字段	必填	类型	描述	备注

结果码 code：

码值	描述	备注
0	成功	
1	失败	
n	其他失败	1.3 统一结果码表

2.10 云端功能调用

描述：业务服务直接通过物联网平台异步下发功能命令给设备，比如可以下发给设备模块拍照的功能。物模型的功能分为异步调用和同步调用。异步调用只发送功能命令后直接返回，不等待设备处理结果。设备处理结果可以通过事件上报的方式异步返回。物联网平台会把事件回执通过 MQ 队列的方式返回给业务服务端。

Topic	iot/thing/cmd/ down /\${productId}/\${deviceName}
方向	云端 ==> 设备
版本	1
备注	只支持业务服务器。

data 数据内容

字段	必填	类型	描述	备注
根据物模型调用的功能标识符	是	object	<p>例子：</p> <pre>{ "snap": { "input" : { "file_name": "123.jpeg", "service_id": "product.Lm3US5SR61TcRTDN.snap" } } }</pre> <p>例子为给设备下发抓拍功能异步调用，数据解析如下：</p> <p>id：功能标识，物模型定义。</p> <p>input：输入参数</p> <p>file_name：抓拍图片文件名，物模型定义。</p> <p>service_id：抓拍的业务 id，物模型定义。</p>	

设备收到确认：

Topic	iot/thing/cmd/ up /\${productId}/\${deviceName}
方向	设备 ==> 云端
版本	1
备注	<p>只支持业务服务器。</p> <p>数据 id 字段要和请求一致。</p>

data 数据内容：

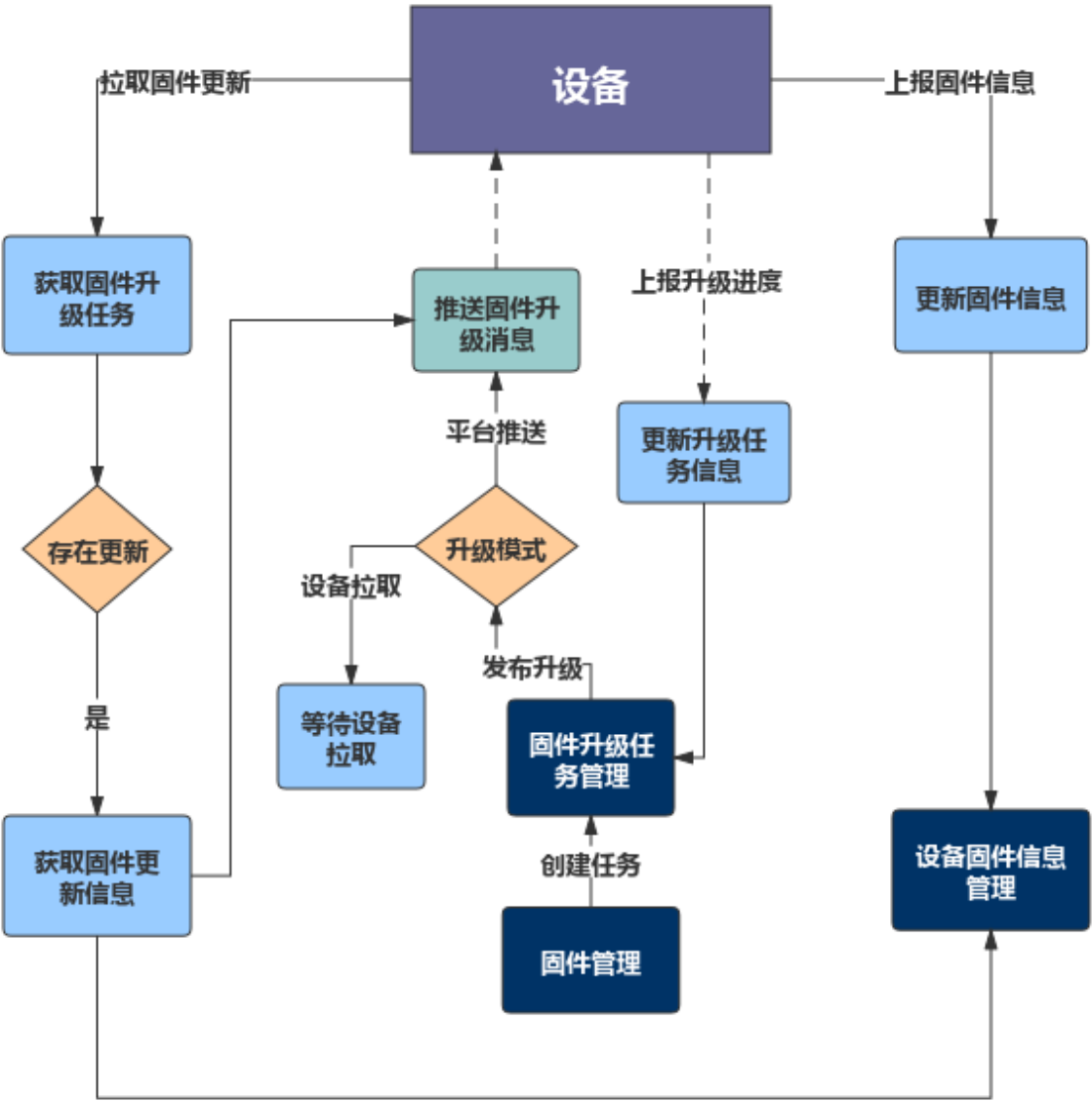
字段	必填	类型	描述	备注
output	否	object	根据物模型定义，如果功能调用有返回参数，则该项有值： "output":{ result: 0 //0 成功，1 失败，2 密钥不存在，3 密钥错误 }	

结果码 code：

码值	描述	备注
0	成功	
1	失败	
n	其他失败	1.3 统一结果码表

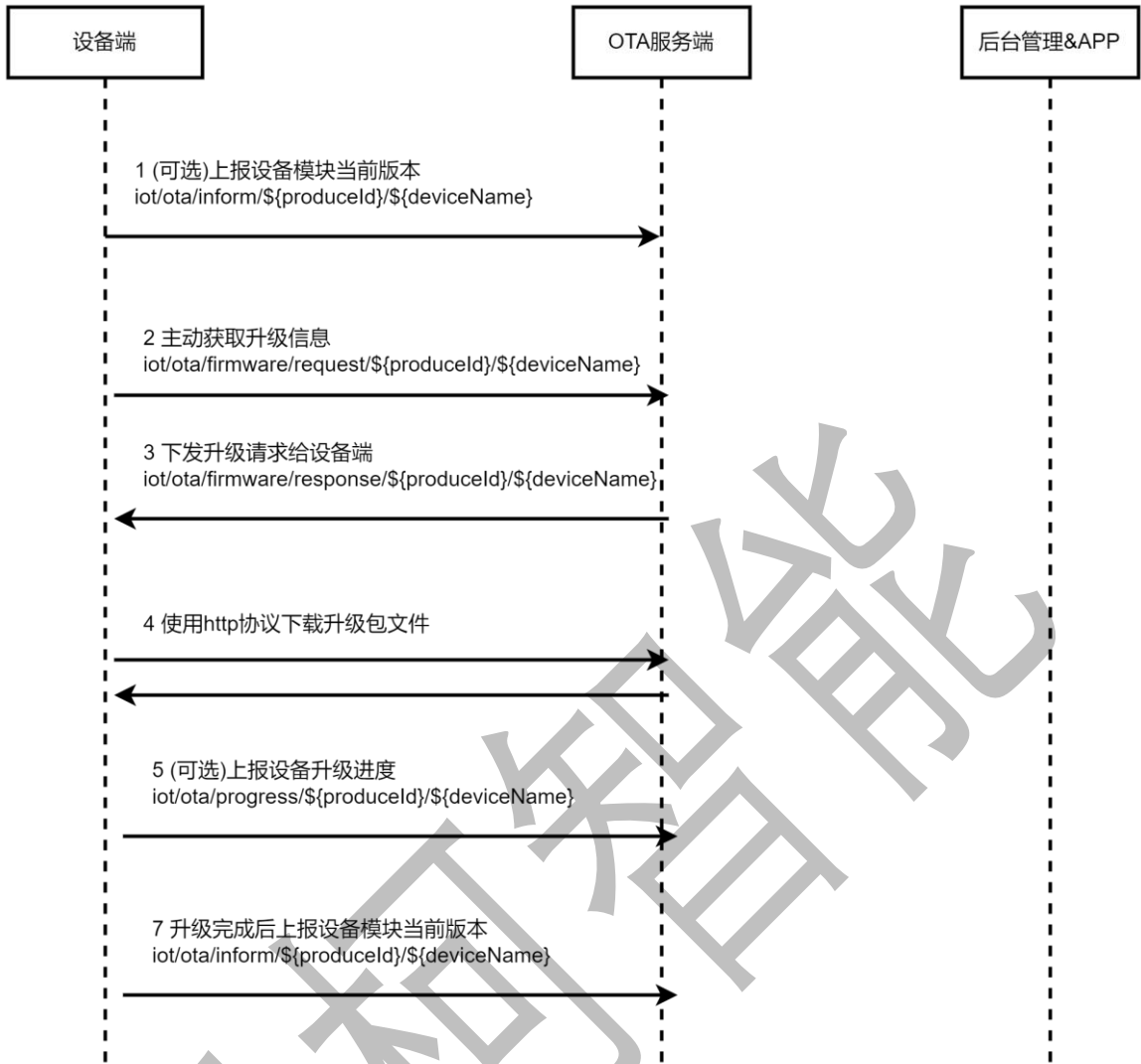
2.11 OTA

描述：设备端固件升级，升级启动分为主动拉取或者平台推送。

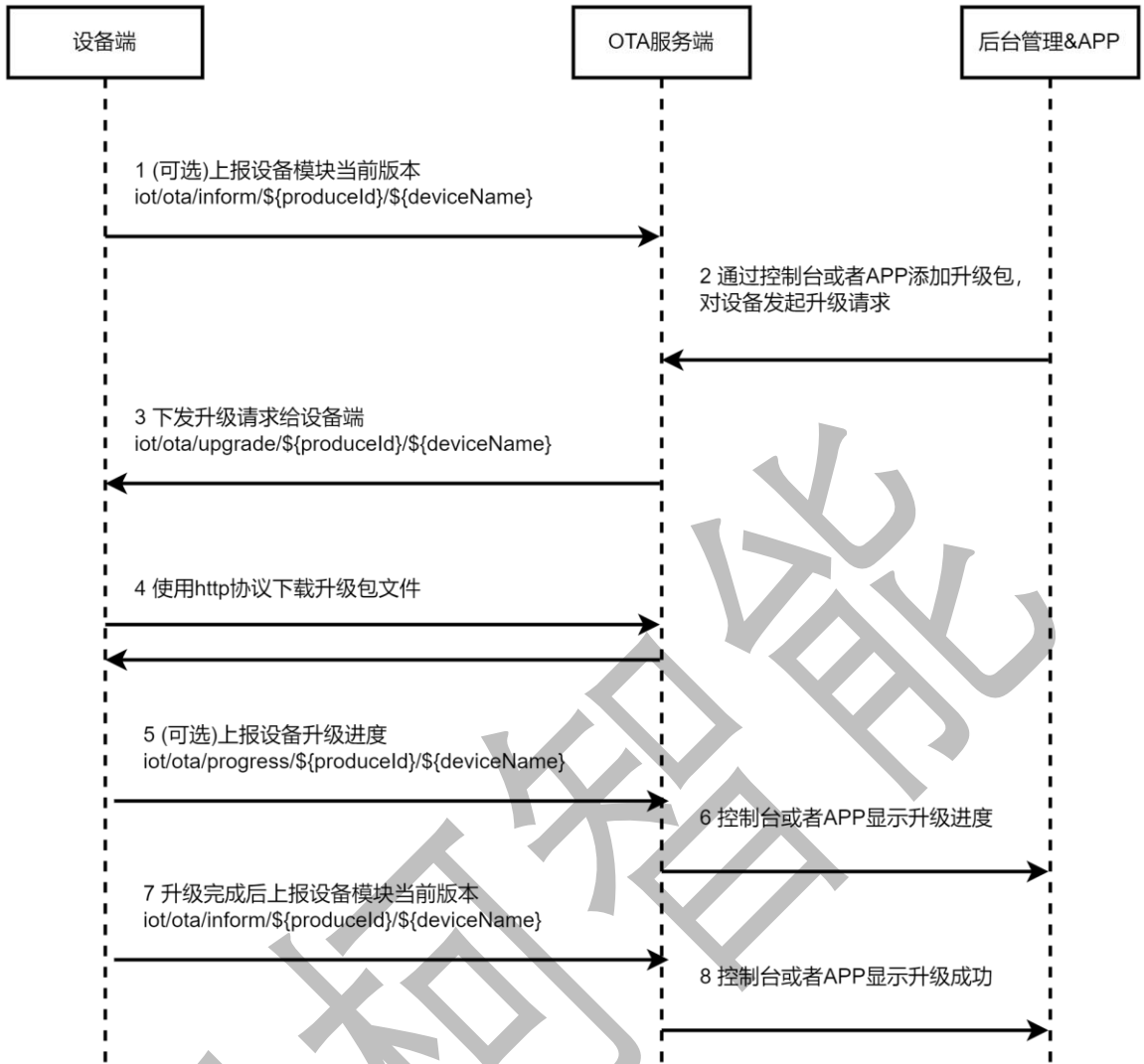


主动拉取需要设备端在自己空闲或者定时去拉取需要升级的升级固件信息后自动升级。平台拉取是指 APP 发送升级信息给设备端，如果设备离线，则推送失败。

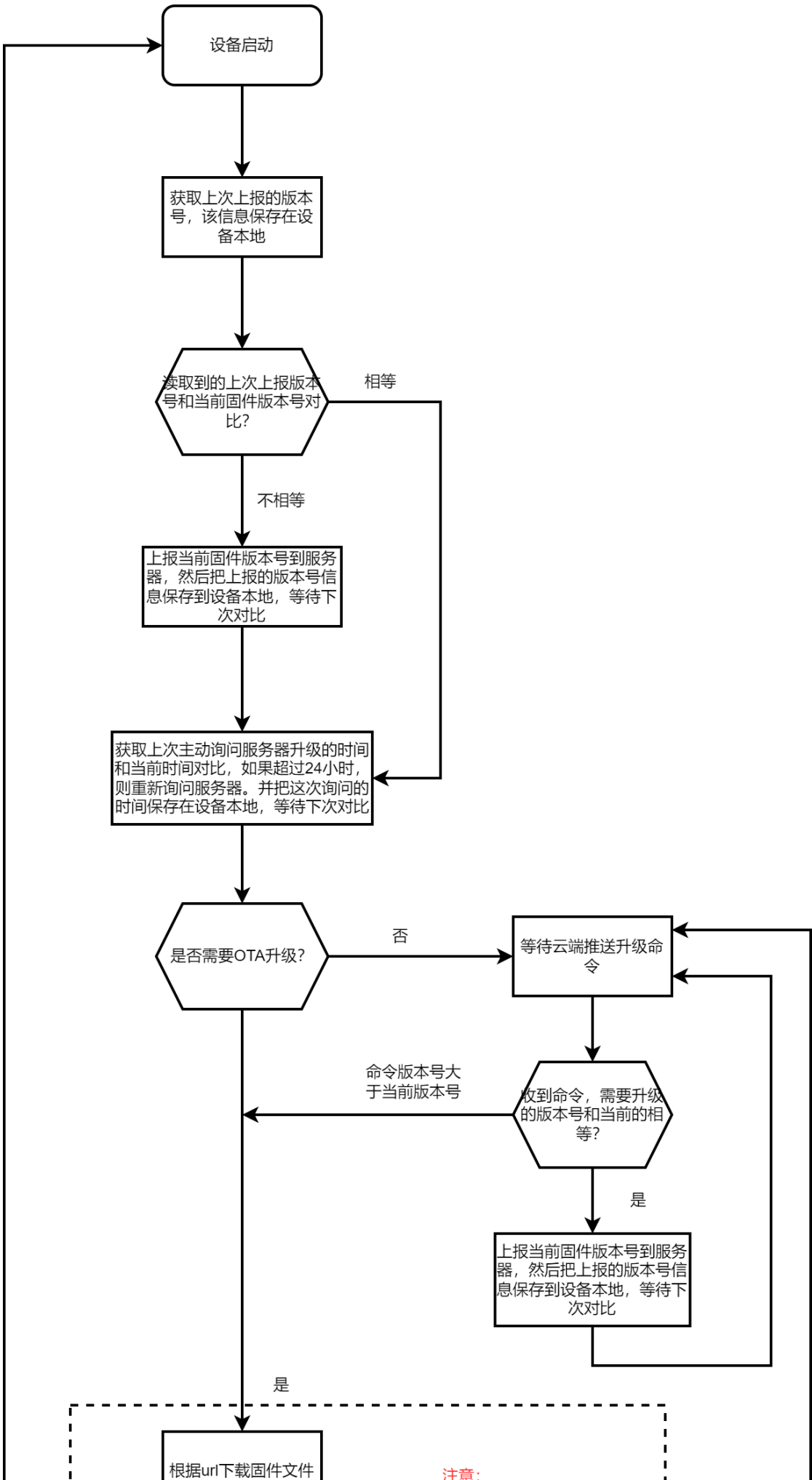
主动拉取流程图：



平台推送流程图：



设备端 OTA 大致流程图：



2.12.1 上报版本信息

描述：设备上报版本信息，在两种情况下发送。第一种，固件升级完成后务必上报升级后的版本信息告知云端，云端以这个版本信息为依据判断升级成功。第二种，可以在需要的时候选择定时或者某些条件触发后发送，这种情况下请勿过于频繁发送。

Topic	iot/ota/inform/\${productId}/\${deviceName}
方向	设备 ==> 云端
版本	1
备注	只支持业务服务器。

data 数据内容：

字段	必填	类型	描述	备注
模块列表信息	是	json 数组	<p>例子：</p> <pre>[{ \"version\": \"1.1.12\", \"module\": \"WIFI\" }, { \"version\": \"1.1.0\", \"module\": \"CPU\" }]</pre> <p>其中：</p> <p>module：模块类型</p> <p>“WIFI” WIFI 模组，</p> <p>“MCU” MCU 模块，</p> <p>“EXPAND” 拓展模块，</p> <p>“OTHER” 第三方模块</p> <p>version：版本号版本号采用 xx.yy.zz 点分表示，</p>	

			范围支持 0.0.0 到 99.99.99。例如：1.0.0	
--	--	--	--------------------------------	--

2.12.2 请求固件升级

描述：设备端主动和云端请求升级信息。如果设备的固件可升级任务处于非待确认和非成功状态，会把该升级任务下发。主动升级请定时一定时间（比如 1-2 天），太过频繁会加重服务器负担。

Topic	iot/ota/firmware/request/\${productId}/\${deviceName}
方向	设备 ==> 云端
版本	1
备注	只支持业务服务器。

data 数据内容：

字段	必填	类型	描述	备注
version	是	string	当前设备版本号，只支持数字和英文点号（.）	
module	是	string	模块类型 "WIFI" WIFI 模组， "MCU" MCU 模块， "EXPAND" 拓展模块， "OTHER" 第三方模块	

设备收到确认：

Topic	iot/ota/firmware/response/\${productId}/\${deviceName}
方向	云端 ==> 设备
版本	1
备注	只支持业务服务器。 数据 id 字段要和请求一致。

data 数据内容：

字段	必填	类型	描述	备注
----	----	----	----	----

version	否	string	版本号	
size	否	long	固件文件大小，单位字节	
url	否	string	固件文件下载地址	
verify_method	否	string	固件文件校验算法	
verify_value	否	string	固件文件校验值	
module	是	string	模块类型 "WIFI" WIFI 模组， "MCU" MCU 模块， "EXPAND" 拓展模块， "OTHER" 第三方模块	
extra	否	string	推送给设备的自定义键值 KV 信息	
tip	否	string	推送给设备的升级版本说明	

结果码 code：

码值	描述	备注
0	成功	
1	失败	
n	其他失败	1.3 统一结果码表

2.12.3 推送下发升级命令

描述：物联网平台推送给设备端升级要求。

Topic	iot/ota/upgrade/down/\${productId}/\${deviceName}
方向	云端 ==> 设备
版本	1
备注	只支持业务服务器。

data 数据内容：

字段	必填	类型	描述	备注
----	----	----	----	----

version	是	string	版本号	
size	否	long	固件文件大小，单位字节，如果云端获取不到文件大小，该项不返回。	
url	否	string	固件文件下载地址	
verify_method	否	string	固件文件校验算法，如果不支持不返回	
verify_value	否	string	固件文件校验值，如果不支持不返回	
module	是	string	模块类型 "WIFI" WIFI 模组， "MCU" MCU 模块， "EXPAND" 拓展模块， "OTHER" 第三方模块	
extra	否	string	推送给设备的自定义信息	
tip	否	string	推送给设备的自定义信息	

2.12.4 上报升级进度

描述：设备上上报升级进度，有的设备不支持该功能则不需要调用过程，只要上报本次升级的最后结果即可。**注意，升级成功务必使用 `iot/ota/inform/${productId}/${deviceName}`**

上报版本。

Topic	iot/ota/progress/\${productId}/\${deviceName}
方向	云端 ==> 设备
版本	1
备注	只支持业务服务器。

data 数据内容：

字段	必填	类型	描述	备注
----	----	----	----	----

version	是	string	版本号	
module	是	string	模块类型 "WIFI" WIFI 模组, "MCU" MCU 模块, "EXPAND" 拓展模块, "OTHER" 第三方模块	
status	是	int	请参考 升级状态描述	
status_extra	否	int	请参考 升级状态描述	

升级状态描述：

status	值	status_extra
升级中	3	0-100, 百分比
升级失败	5	0 通用失败 1 下载失败 2 校验失败 3 烧写失败 4 未知失败
升级取消	6	

2.12 门锁设备数据透传上报

描述：WIIF 模块直接把门锁的数据透传到云端，由云端来解析处理

Topic	lock/transfer/up/\${productId}/\${deviceName}
方向	设备 ==> 云端
版本	1
备注	只支持业务服务器。

data 数据内容：

字段	必填	类型	描述	备注
protocol	是	int	协议类型： 0：几围协议 1：涂鸦协议 2：云柯协议	
cmd	是	base64 编码后的字符串	设备要的透传数据，格式见相应的门锁协议文档	

2.13 门锁设备数据透传下发

描述：云端直接根据门锁芯片协议通过 wifi 透传下发指令给门锁芯片

Topic	lock/transfer/down/\${productId}/\${deviceName}
方向	云端 ==> 设备
版本	1
备注	只支持业务服务器。

data 数据内容：

字段	必填	类型	描述	备注
protocol	是	int	协议类型： 0x00：几围协议 0x01：涂鸦协议 0x02：云柯协议	
cmd	是	base64 编码后的 byte 数组	数据格式见门锁协议文档	

3 物模型

3.1 物模型支持的数据类型

数据类型	说明	示例
int	32 位整型。	10
long	64 位整数	12345678901
float	单精度浮点型。	1.1
double	双精度浮点型。	1.23
text	字符串，对应的数据长度不能超过 10240 字节。	"hello world!"
date	时间戳。格式为 String 类型的 UTC 时间戳，单位：毫秒。	"1635839462000"
bool	布尔型。采用 0（false）或 1（true）来定义布尔值，且 0 和 1 为 int32 类型。	0 表示关、1 表示开。
enum	枚举型。定义枚举项的参数值和参数描述，参数值必须为整数。	整数 0 表示红色，整数 1 表示蓝色，整数 2 表示绿色。
struct	JSON 对象。定义一个 JSON 结构体，结构体内元素类型支持 int32、float、double、text、date、bool 和 enum，不支持结构体嵌套。	<code>{ "name": "Tom", "age": 10 }</code>
array	数组。需声明数组内的元素类型、数组元素个数。元素类型支持 int32、float、double、text 或 struct，需确保同一个数	[1, 2, 3, 4, 5, 6]

	组元素类型相同。元素个数限制为1~512个。	
--	------------------------	--

云柯智能