



从 0 到 1 搭建十亿级包裹 API Versioning



杨利航

AfterShip
高级研发工程师

这是 API Versioning 吗?



/api/v1



目录

Why

01

What

02

How

03

Takeaway

04

Part 1/4

Why

1. SaaS 产品的特点
2. API 的重要性
3. 我们 API 面临的挑战



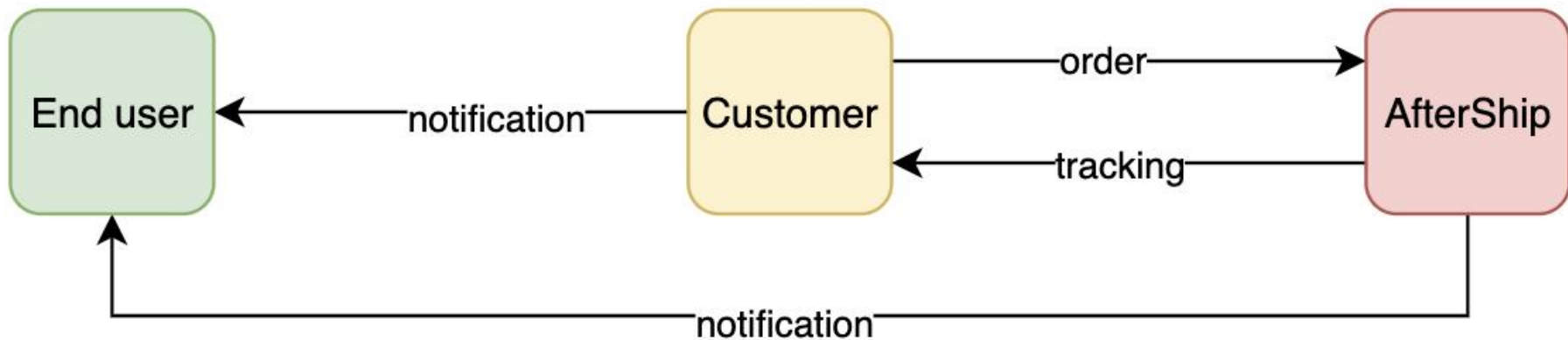
1.1 SaaS 产品的特点

SaaS 产品提供某一领域的解决方案的软件服务，本质是提供订阅服务。



1.1 SaaS 产品的特点

AfterShip 属于国际 SaaS 的 B2B2C 产品



1.1 SaaS 产品的特点

SaaS 2B 和 2C 的差异

- 产品定位
- 稳定性影响
- 稳定性承诺

国内 SaaS 和国际 SaaS 的技术规划的差异

- 专注和极致
- 规范和标准
- 安全合规
- 全球化

1.2 API 的重要性

API 也是产品的一部分，而不仅仅是技术。对于 SaaS 产品，提供 API 服务几乎是与企业级用户合作的必须项，特别是在海外。

- **自动化** - 通过 API，企业可以利用 SaaS 产品的功能来构建自动化流程和 workflows。
- **定制化** - 通过 API，企业可以开发自己的应用程序、插件或集成其他工具，以满足特定的业务需求。
- **数据集成** - 通过 API，企业可以灵活地与 SaaS 提供商实现数据的共享和交换。

1.2.1 标准与规范

RESTful

- Request
- Response
- Query
- ...

/books

GET	/books	Lists all the books in the database
DELETE	/books/{bookId}	Deletes a book based on their id
POST	/books	Creates a Book
PUT	/books/{bookId}	Method to update a book
GET	/books/{bookId}	Retrieves a book based on their id

安全性

- AES
- RSA
- OAuth2.0
- ...

1. API Key

Required Headers

2. AES

Required Headers

Calculating the signatu...

3. RSA

Required Headers

Calculating the signatu...

4. Legacy API Keys

全球化

- 多时区
- 多语言
- 多国家/地区
- 多币种
- ...

Country / Region

Do's

```
{ "country_region": "USA", "country_region_name": "United States of America" }
```

Don'ts

```
{ "country_id": "USA" }  
{ "country": "United States of America" }  
{ "country": "USA" }
```

Currency

Do's

```
Generic usage, only 0 OR 2 decimal digits  
{ "amount": "23.52", "currency": "USD" }  
{ "amount": "23", "currency": "USD" }  
Billing related usage support up to 10 decimal digits  
{ "amount": "23.0123456789", "currency": "USD" }
```

Don'ts

```
{ "amount": "USD23.52" if do not merge }  
{ "amount": 23.5 if not a number }  
{ "currency": "USD" }
```

Phone Number - E164

Do's

```
{ "phone": { "country_code": "86", "number": "91238123" } }  
{ "phone2": { "country_code": "86", "number": "15951238123" if area code also here } }  
{ "mobile": { "country_code": "86", "number": "15951238123" if no chunk code } }  
{ "fax": { "country_code": null, "number": "91238123" } }
```

Don'ts

```
{ "phone": { "country_code": "86", "number": "91238123" } }  
{ "phone2": { "country_code": "86", "number": "15951238123" } }  
{ "mobile": { "country_code": "86", "number": "15951238123" } }  
{ "fax": { "country_code": null, "number": "91238123" } }
```

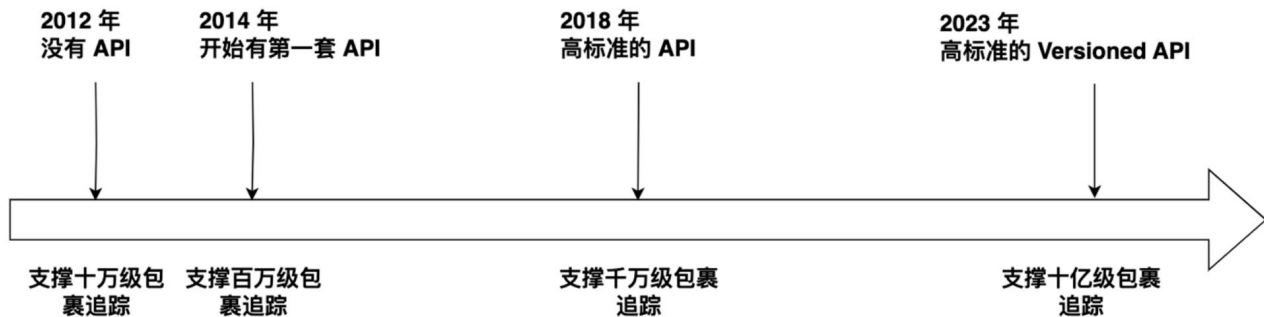
Do's

```
{ "created_at": "2020-01-01T12:00:00Z" }  
{ "created_at": "2020-01-01T12:00:00+08:00" }
```

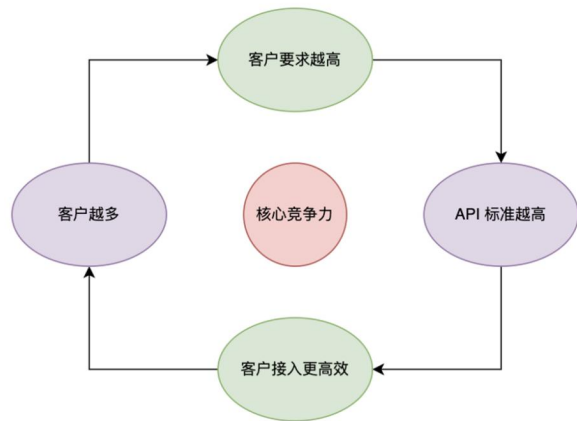
Don'ts

```
{ "created_at": "20200101 12:00:00" }  
{ "created_at": "12:00:00" }
```

1.2.2 AfterShip API 的演进路线



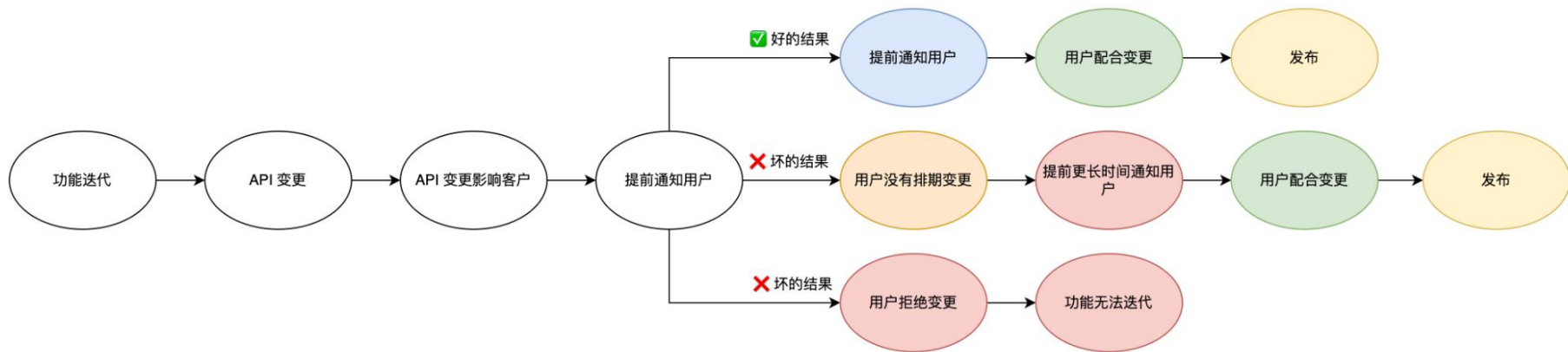
业务量从十万级到十亿级，从没有 API 到高标准的 API。不是因为 API 做好了所以业务增长了，而是良好的 API 设计才能支撑业务增长。



1.3 我们 API 面临的挑战

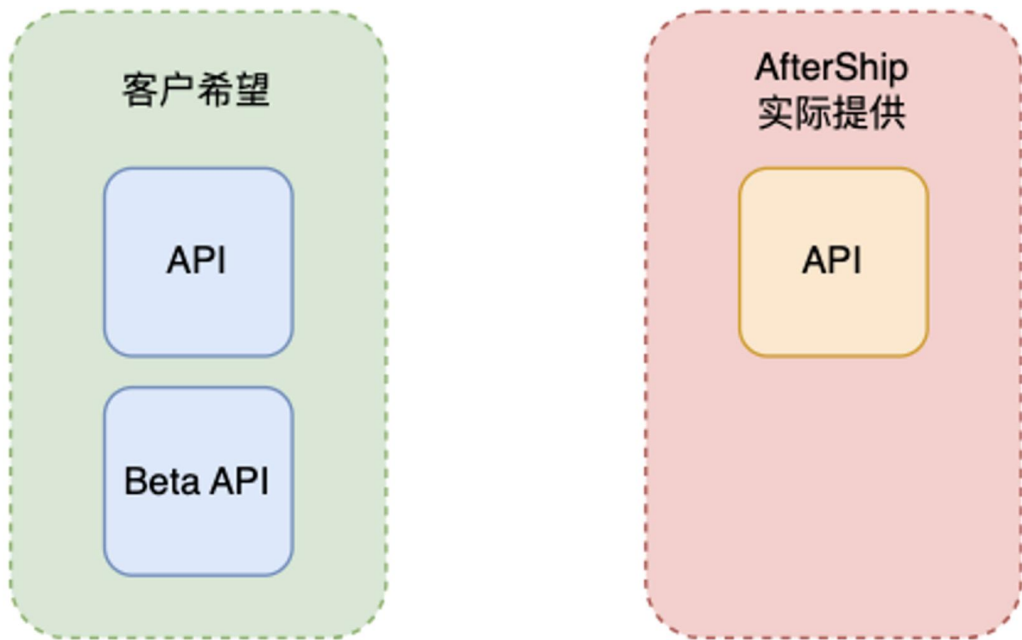
挑战一、变更不规律，打乱客户更新计划

向后兼容 - API 的稳定性要求避免破坏性的更改。在进行更新或修改时，应保持对现有功能的支持，以防止影响已部署的应用程序或客户端。



1.3 API 变更面临的挑战

挑战二、新功能缺少调试环境



Part 2/4

What

1. 本质
2. 实现效果

2.1 本质

我们以为的 **API Versioning**

v1, v2, v3 就是版本化..

```
curl --location --request POST 'https://api.aftership.com/v4/trackings' \  
--header 'as-api-key: your aftership api key' \  
--header 'Content-Type: application/json' \  
--data-raw '{  
  "tracking": {  
    "tracking_number": "9405511202575421535949",  
    "slug": "usps"  
  }  
'
```

2.1 本质

向后兼容 - API 的稳定性要求避免破坏性的更改。在进行更新或修改时，应保持对现有功能的支持，以防止影响已部署的应用程序或客户端。

REST API / About the REST API /

Breaking changes

Learn about breaking changes that were introduced in each REST API version.

About breaking changes in the REST API [🔗](#)

The GitHub REST API is versioned. The API version name is based on the date when the API version was released. For example, the API version `2022-11-28` was released on Mon, 28 Nov 2022.

Any breaking changes will be released in a new API version. Breaking changes are changes that can potentially break an integration. Breaking changes include:

- removing an entire operation
- removing or renaming a parameter
- removing or renaming a response field
- adding a new required parameter
- making a previously optional parameter required
- changing the type of a parameter or response field
- removing enum values
- adding a new validation rule to an existing parameter

只有做到对当前版本 **API 不发布任何 breaking changes** 才能向后兼容。

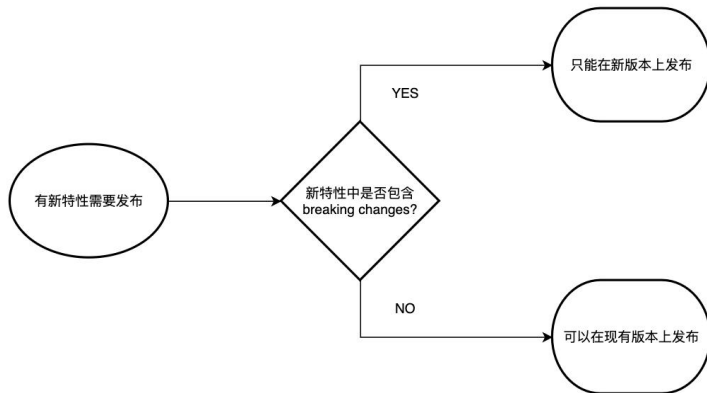
2.1 本质

实际上的 API Versioning

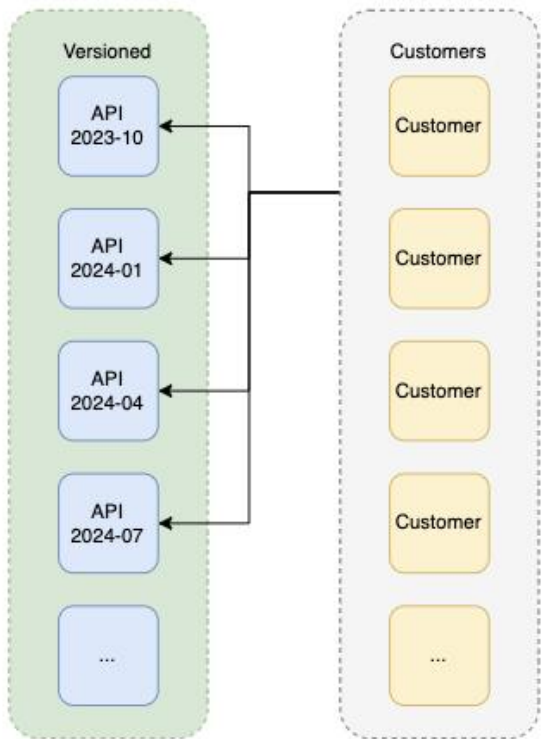
Versioning

When backwards-incompatible changes are made to the API, we release a new, dated version. The current version is **2023-10-16**. Learn more about [API upgrades and backwards compatibility](#). For information on all API updates, view our [API changelog](#).

1. API 有版本概念
2. 同一个版本的每次 API 变更都能向后兼容
3. 不能向后兼容的发布只会发生在新版本中



2.2 实现效果



一个 **Versioned** 的 **API**，将提供多套稳定版本的 **API** 在线上，每套 **API** 有独立的逻辑、实例、资源，各版本之间互不影响。客户端可以自由选择调用哪个版本。

Part 3/4

How

1. Support Policy
2. API Version
3. Webhook Version
4. 公开文档
5. 系统架构
6. 版本维护



3.1 Support Policy

企业	业务特点	API 版本发布周期	API 版本支持时长
Shopify	提供电子商务解决方案，业务快速增长，功能迭代快。	三个月定期发布	12 个月
Stripe	提供在线支付解决方案的公司，业务相对成熟，功能很少迭代。	不定期发布，发布频率低	永久，有一定约束，如新用户只能使用最新版本

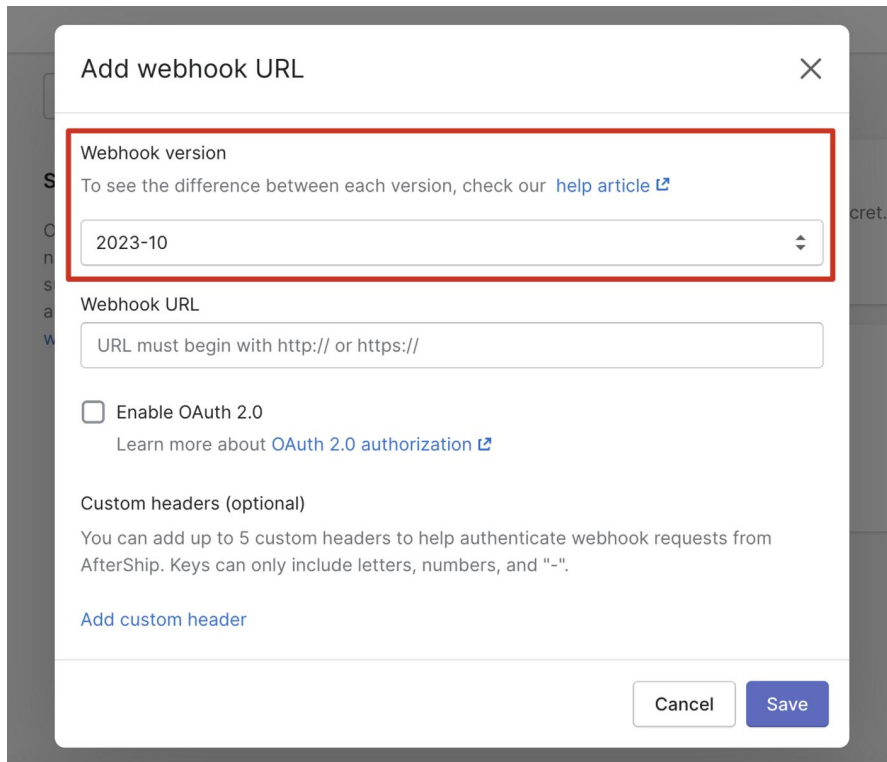
每新增一个版本都会带来额外的维护成本，所以版本发布频率高的企业，版本支持时间较短；版本发布频率低的企业，版本支持时间较长。企业应根据自身的业务特性选择合适的 **Support Policy**。

3.2 API version

v1/v2/v3 这种数值类型的版本号，没有生命周期，无法约束废弃时间。以日期作为 **API Version** 对于生命周期以及兼容性可以有更加清晰的定义。

```
GET 'https://api.aftership.com/tracking/2023-10/trackings'  
--  
HTTP/2 200  
date: Mon, 11 Dec 2023 14:29:43 GMT  
content-type: application/json; charset=utf-8  
surrogate-control: no-store  
cache-control: no-store, no-cache, must-revalidate, proxy-revalidate  
pragma: no-cache  
expires: 0  
as-api-version: 2023-10  
...|
```

3.3 Webhook version



Add webhook URL

Webhook version

To see the difference between each version, check our [help article](#)

2023-10

Webhook URL

URL must begin with http:// or https://

☐ Enable OAuth 2.0

[Learn more about OAuth 2.0 authorization](#)

Custom headers (optional)

You can add up to 5 custom headers to help authenticate webhook requests from AfterShip. Keys can only include letters, numbers, and "-".

[Add custom header](#)

Cancel Save

```
host: webhook.example.com
content-length: 12345
accept: application/json
content-type: application/json
accept-encoding: gzip
user-agent: AfterShipTrackingWebhook/4.0
as-webhook-version: 2023-10
...
```

3.4 公开文档

[Home](#)[AfterShip Tracking](#)[AfterShip Returns](#)[AfterShip Shipping](#)[AfterShip Protection](#)[More ▾](#)

✓ 2023-10

legacy



Rate Limit

Body Envelope

Request Errors

Try API Endpoints

CHANGELOG

Changelog

Migration Guide

REFERENCE

 API Overview

Trackings ▾

Get trackings GET

Create a tracking POST

Get a tracking by ID GET

Update a tracking by ID PUT

Delete a tracking by ID DELETE

Migration from legacy API to 2023-10

The following guide will help you to migrate the API and webhook from the legacy version to the `2023-10` version.



Currently SDKs are not supported for versioned APIs. Please stay tuned for further updates.

Breaking changes in version 2023-10

API Versioning changes

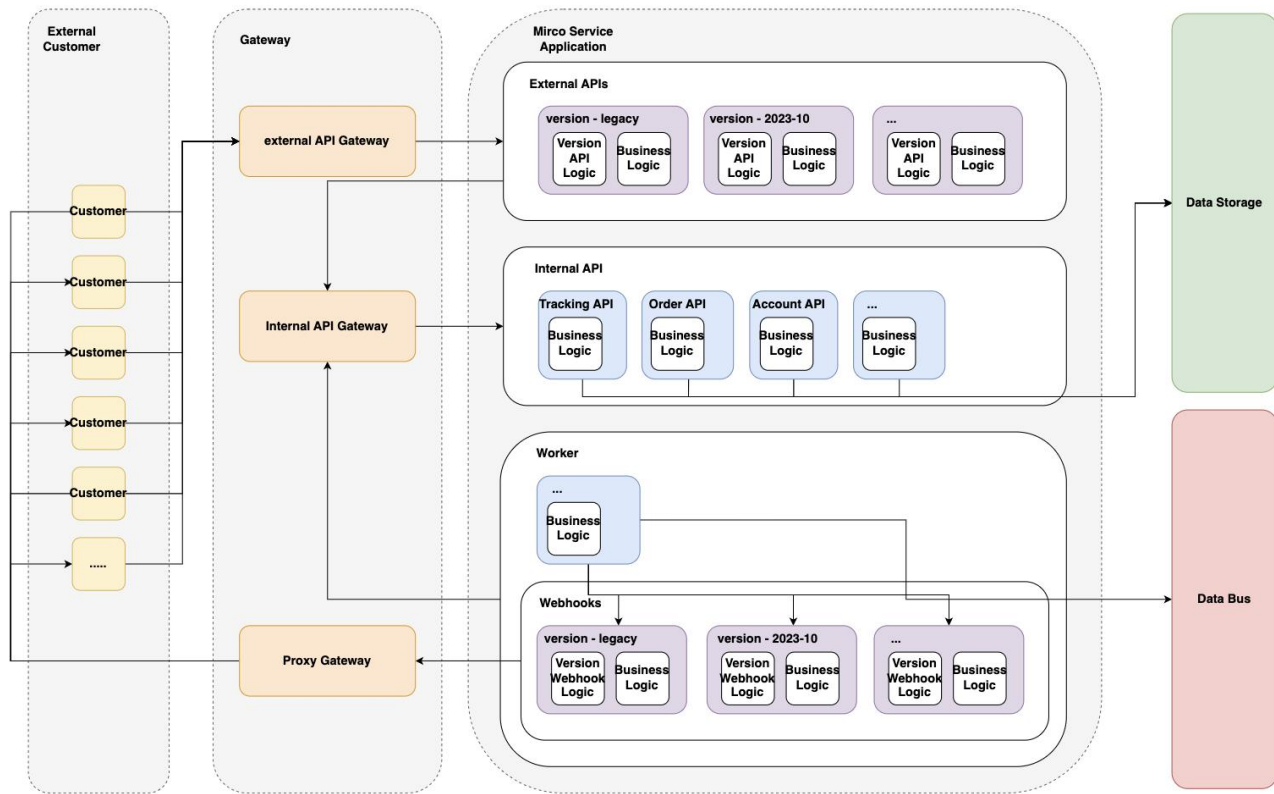
- API

What's the difference?

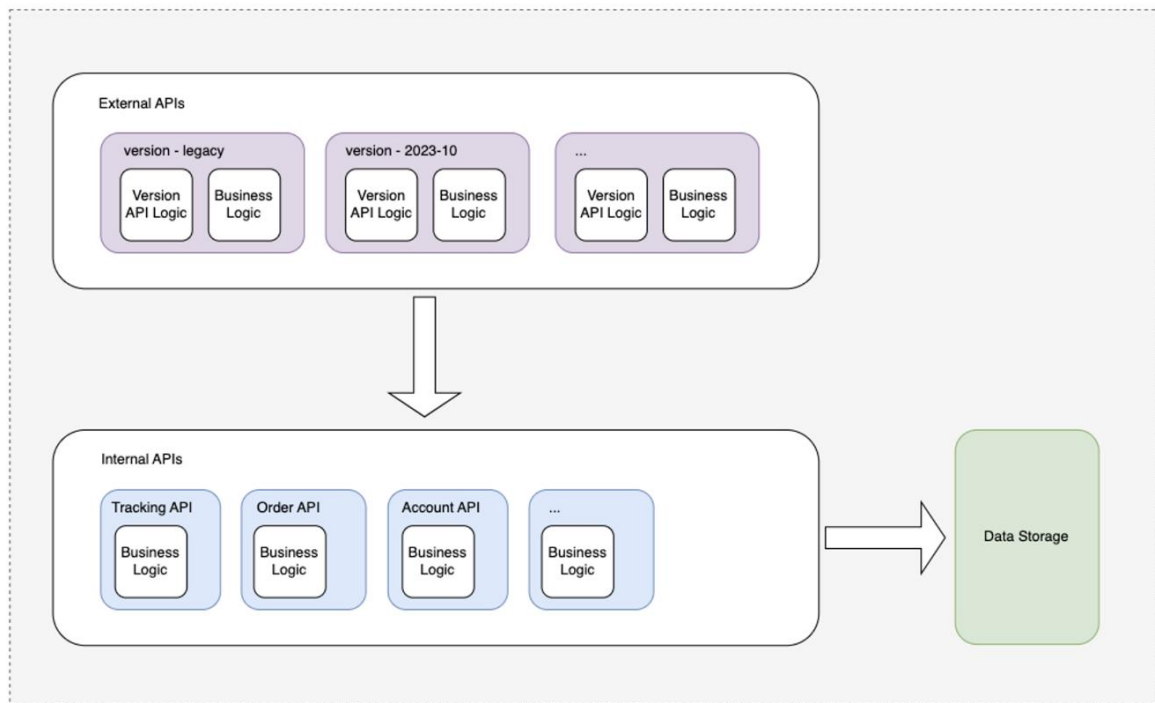
CHANGE	BEFORE	AFTER
Base URL	<code>https://api.aftership.com/v4</code>	<code>https://api.aftership.com/tracking/2023-10</code>
New response header	-	New response header key <code>as-api-version</code> Example value <code>2023-10</code>



3.5.1 系统架构概览

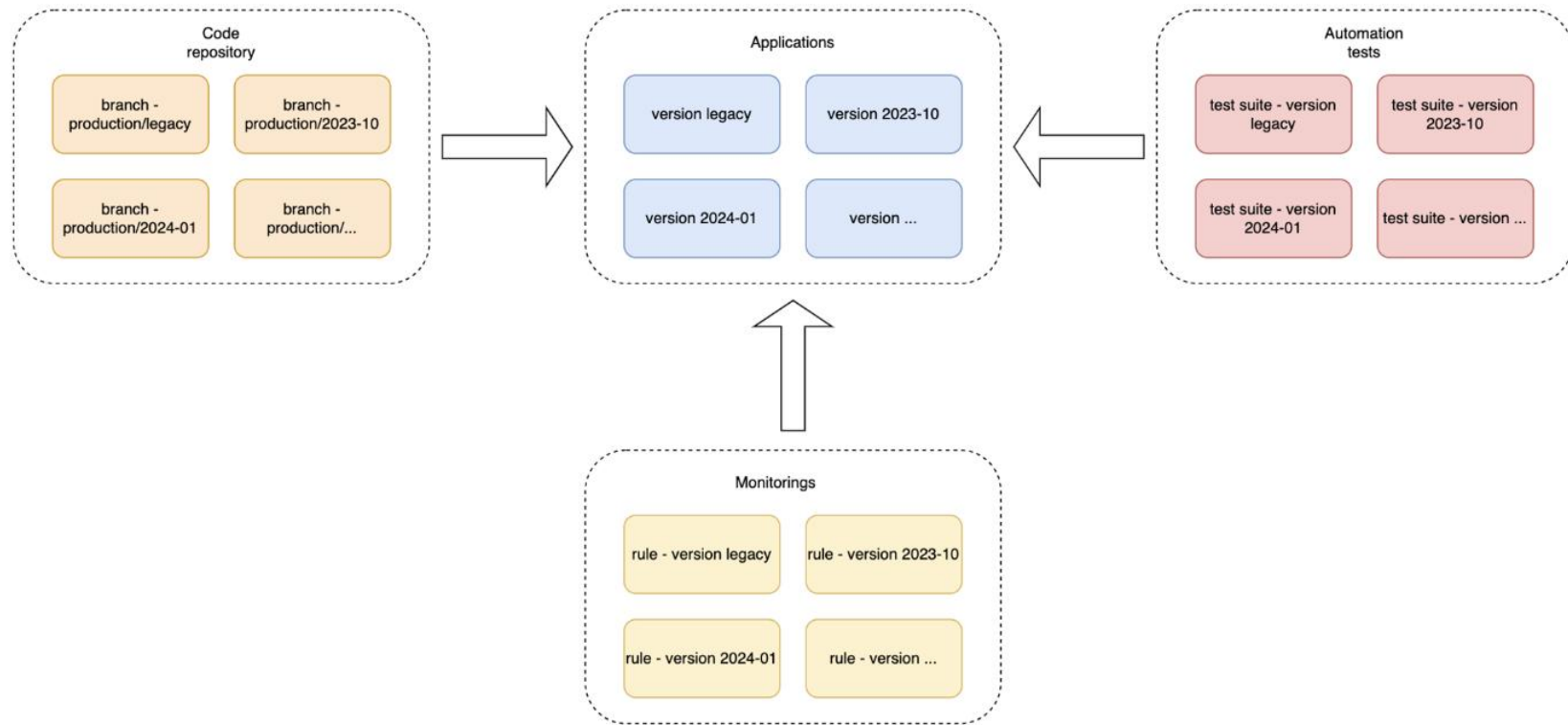


3.5.2 系统设计的 trade-off



我们只在应用层做版本化，
每个版本维护的业务逻辑实际
上是对能力层的兼容。

3.6 版本维护



Part 4/4

Takeaway

Takeaway

1. API 不仅仅是技术，也是一款产品。如果想要和企业用户合作，API 是必需项。在一开始就应该以高标准、高要求规划我们的 API。
2. API Versioning 的本质是最大化解决向后兼容问题，已存在的 API 不会出现向后不兼容的变更，不能向后兼容的功能只会在新版本中发布。
3. 根据我们业务迭代的速度，决定是按固定周期发布版本还是按需发布新版本。比如我们平均每三个月发布一个新功能，可以考虑以 3 个月为周期发布版本，每个版本保留 12 个月，那我们同时最多只需要维护 4 个版本的 API。
4. 建议在应用层实现 Versioning，不建议在能力层实现。虽然可以做到非常极致，但是业务复杂后维护成本将相当可怕，我们必须做取舍。



