

# 基于现有数据接口测试策略

第三届中国移动互联网测试开

主办方: TesterHome

用蜕变理论来识别数据及用例及自动化断言

邱化峰 张雅君

## MTSC2018

第四届中国移动互联网测试开发大会

TesterHome

01

数据的来源

02

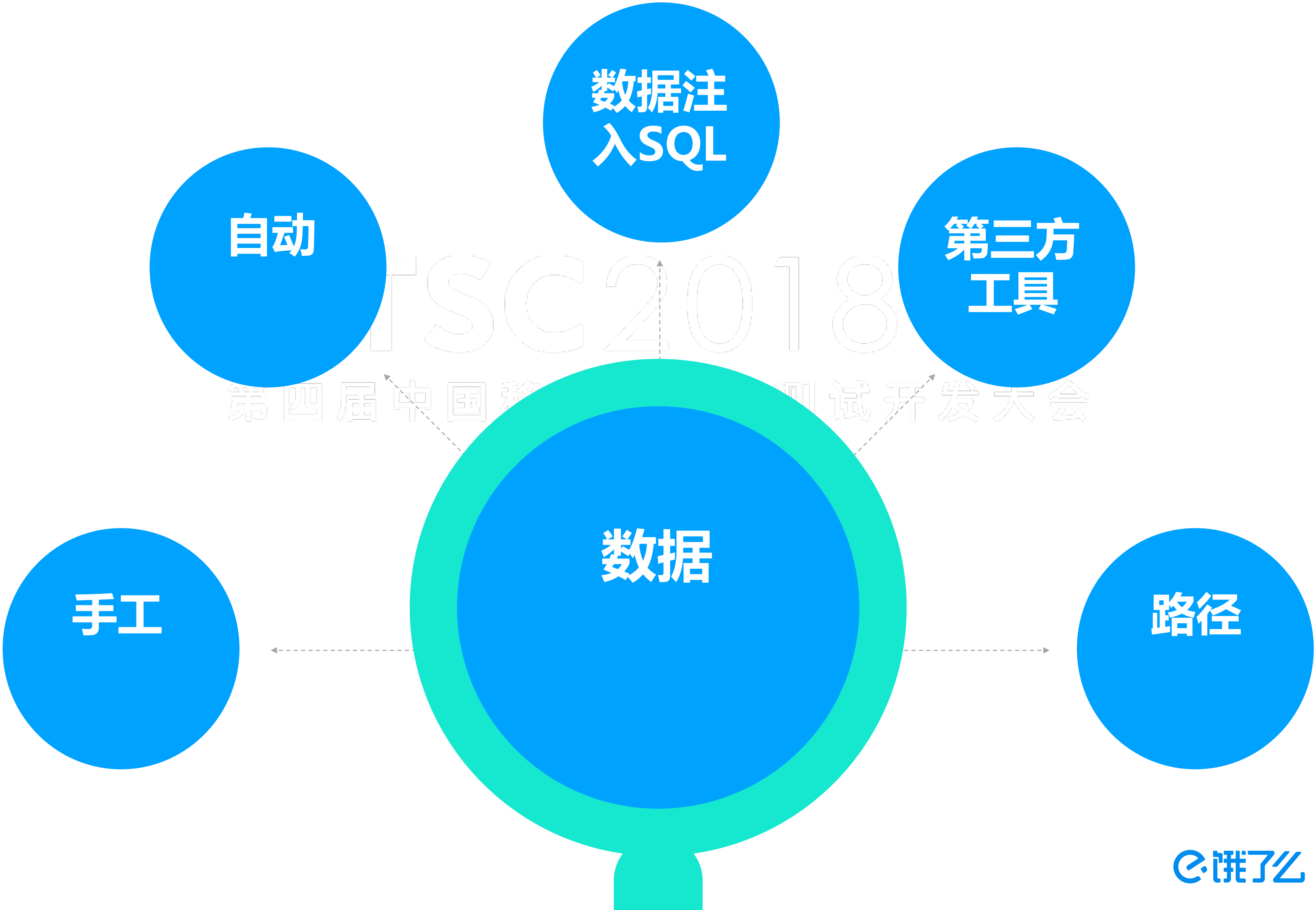
数据生成的策略

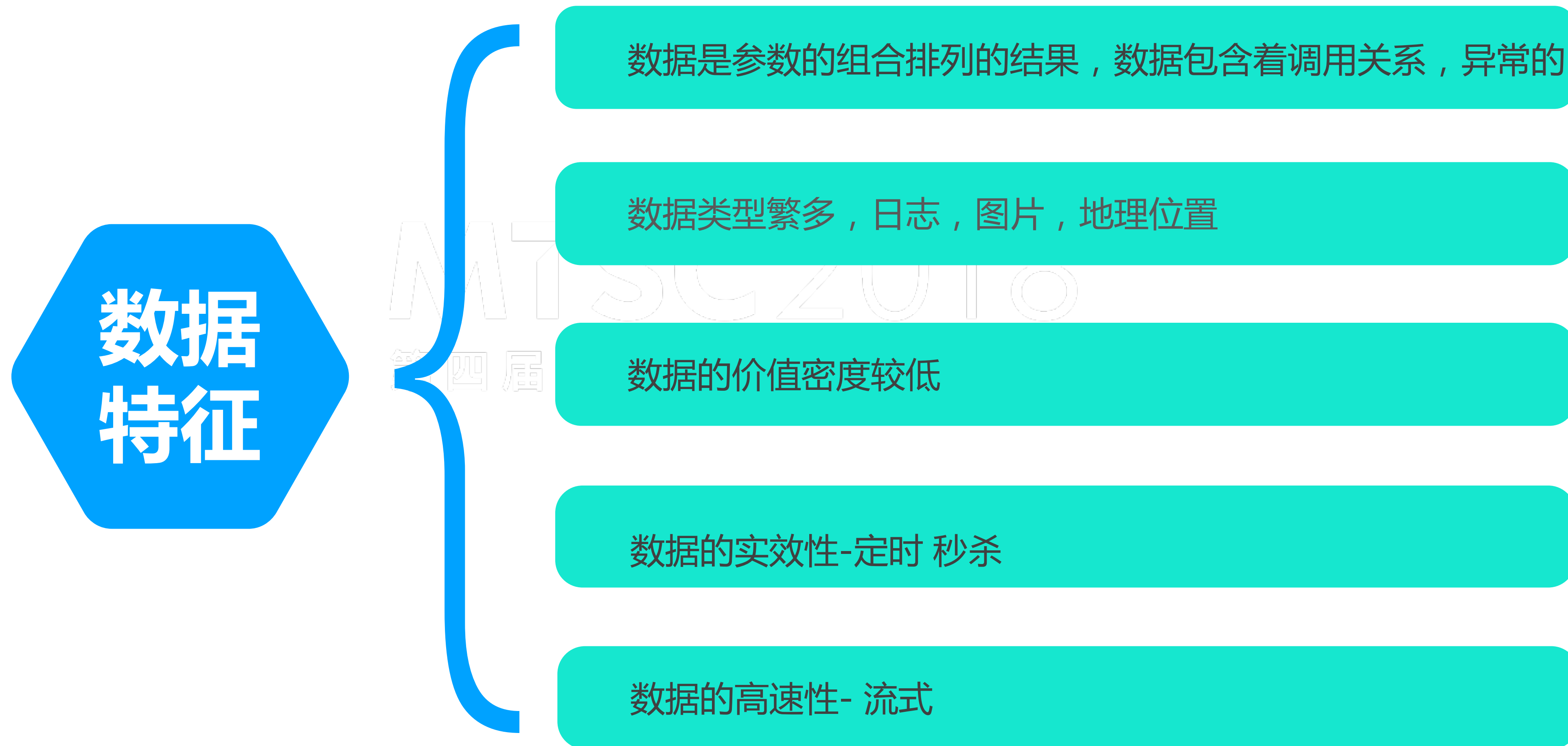
03

数据的有效性验证

# 01

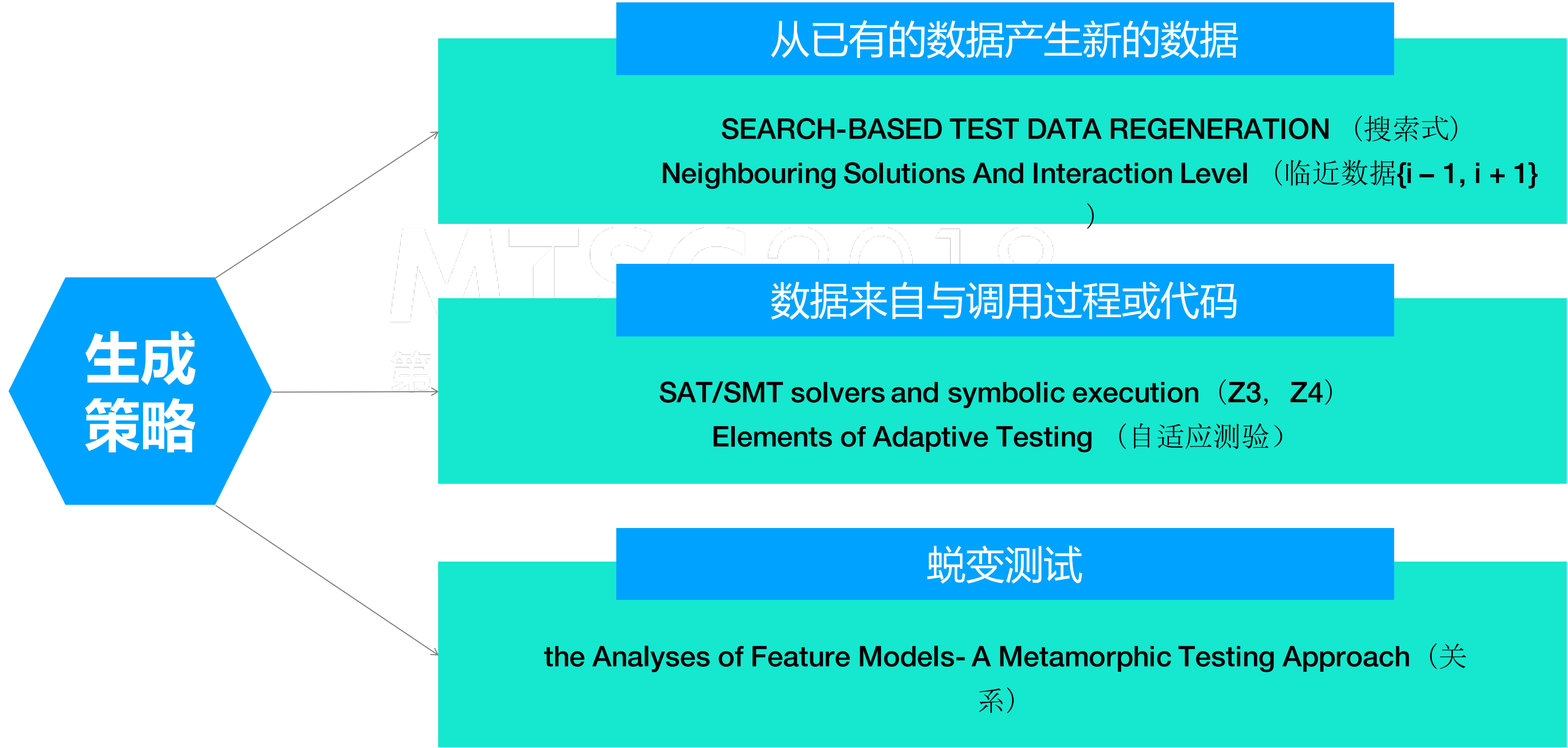
## 数据的来源



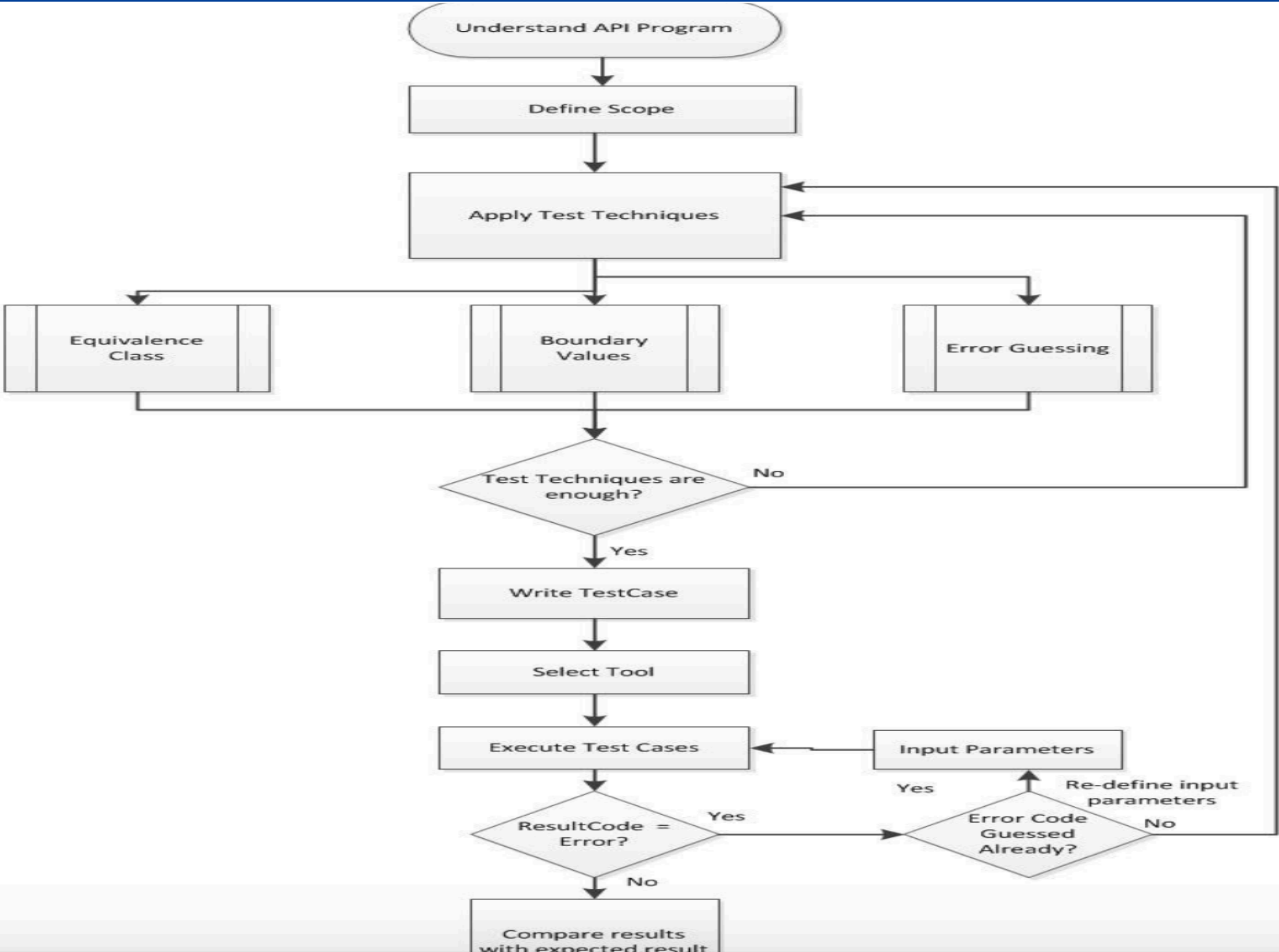


# 02

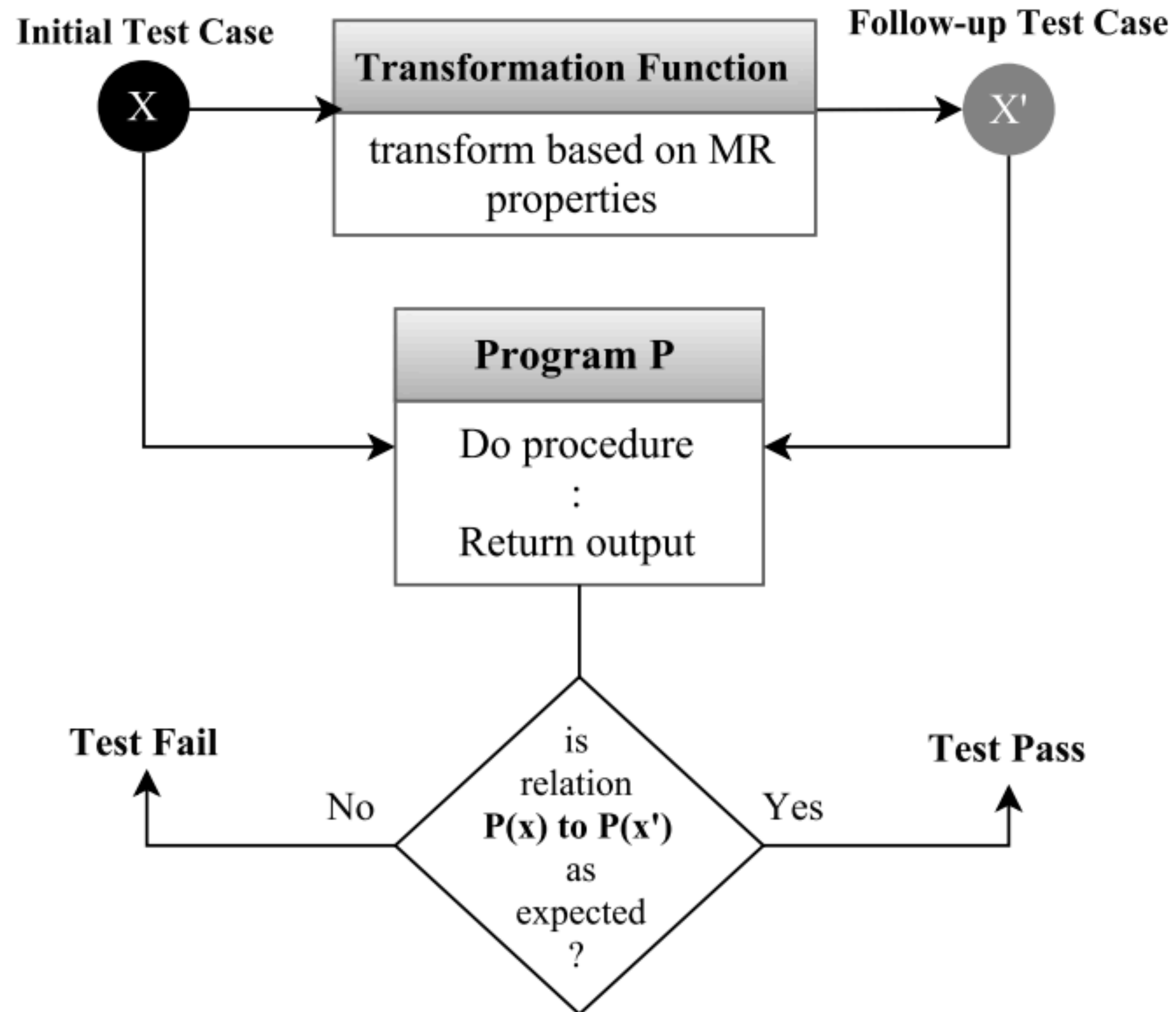
## 数据生成的策略







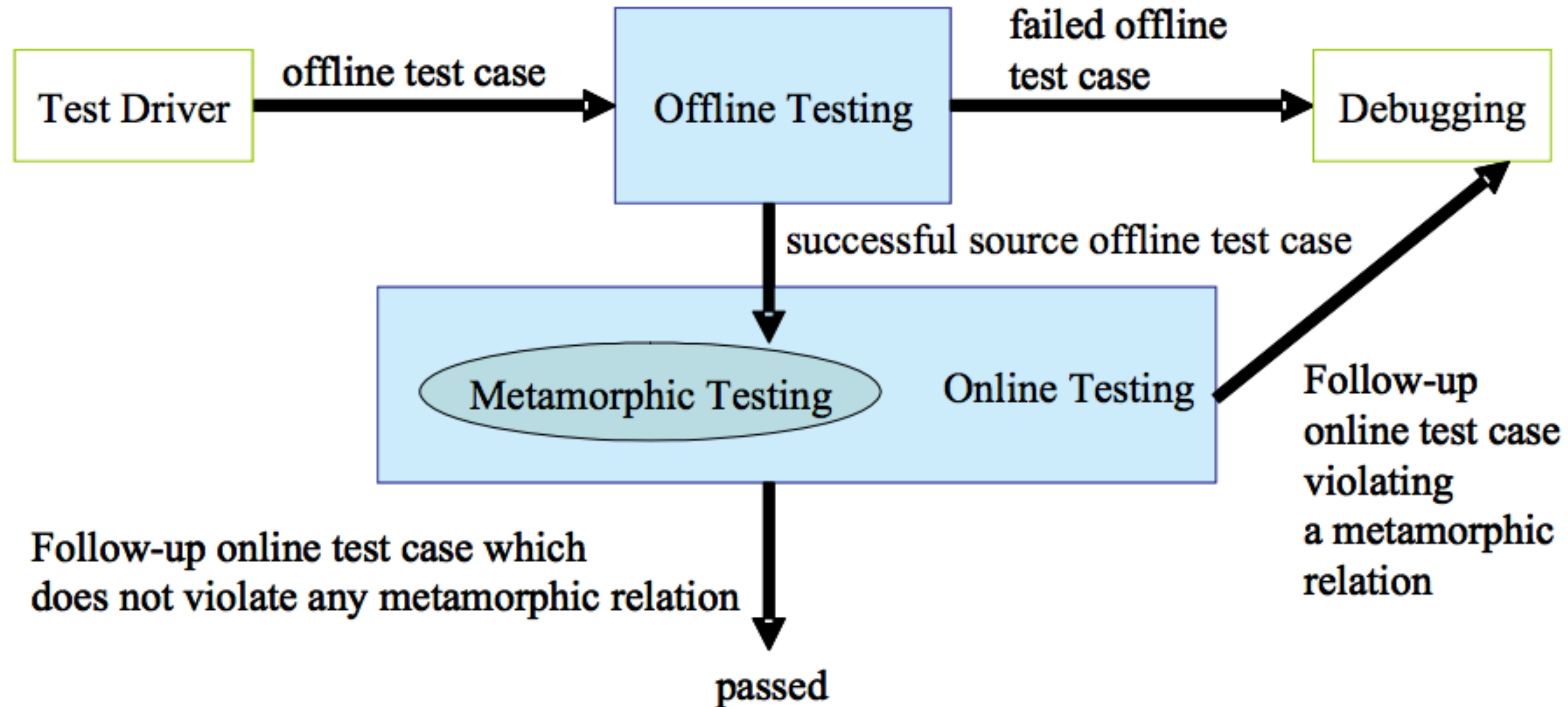


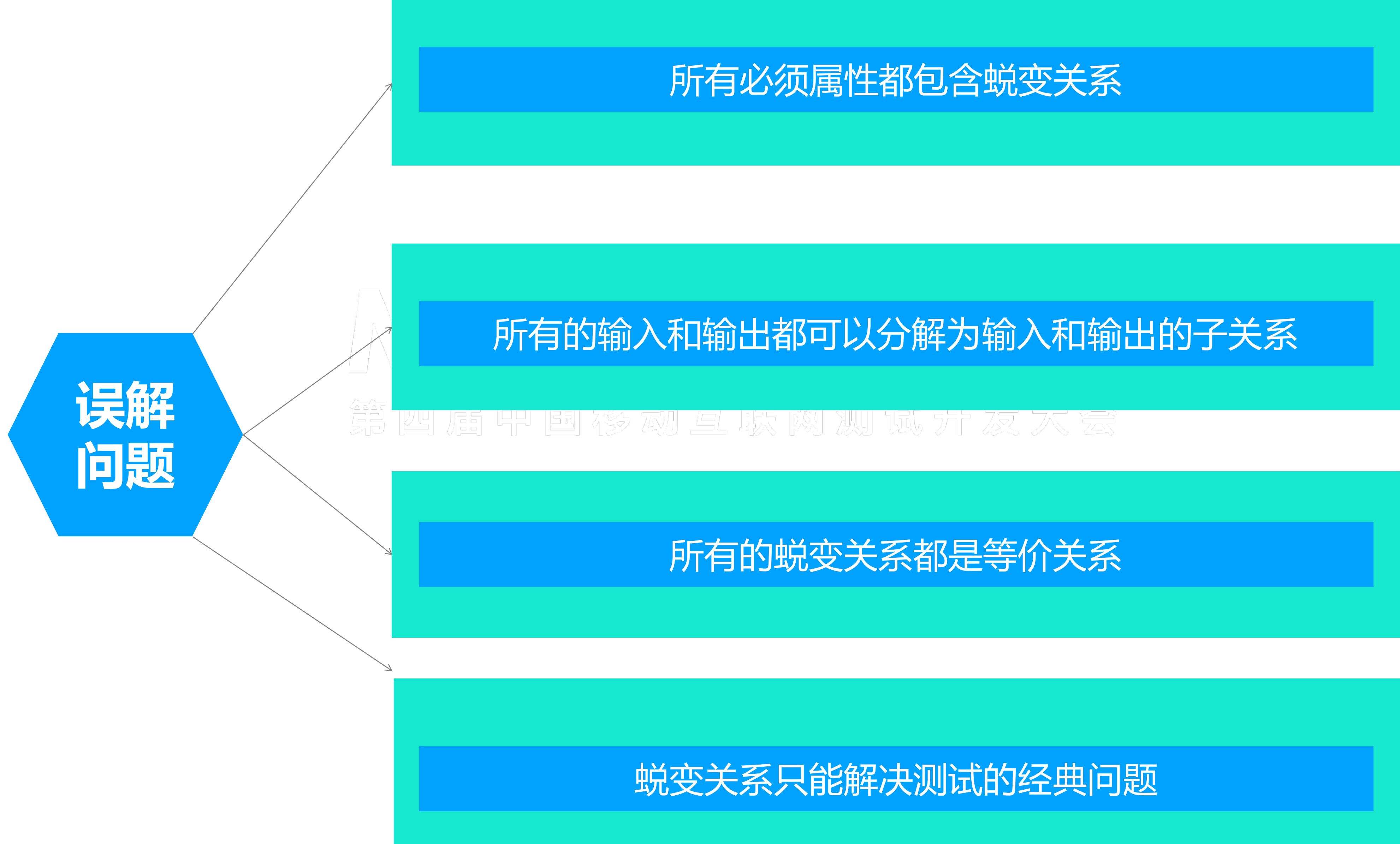


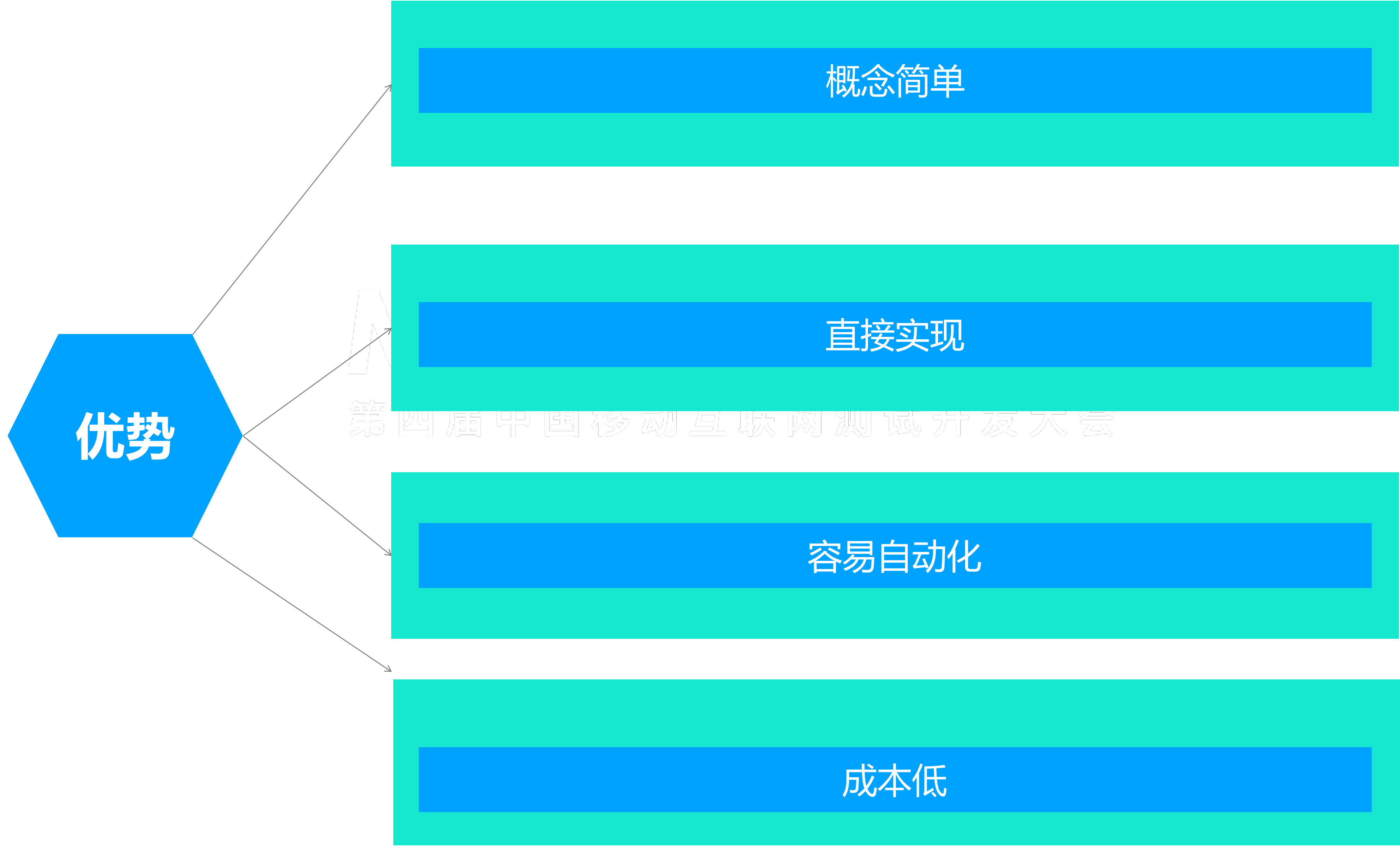
将蜕变测试应用于微服务测试以解决oracle问题

通过变形测试机制，选择用例，交叉验证测试结果

在测试环境下建立测试用例，应用于线上测试







蜕变测试的理论及其演进

更容易的被测试人员接受

新的覆盖范围和标准

如何应用于用户测试

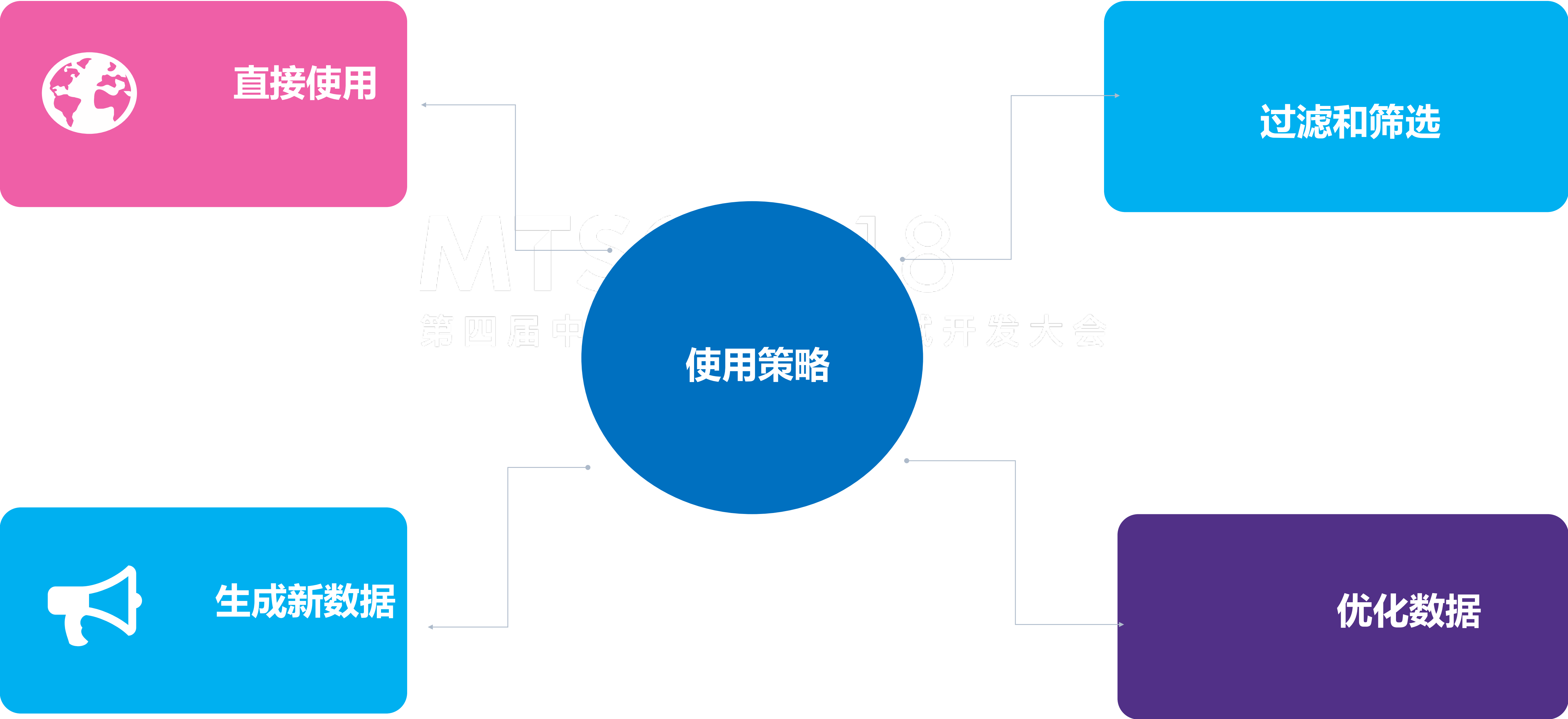
应用于大数据

应用于云计算和众包

应用于敏捷开发







### 录制

一：真实的请求调用

二：收集调用的相关信息

三：持久化所需要的数据和内容

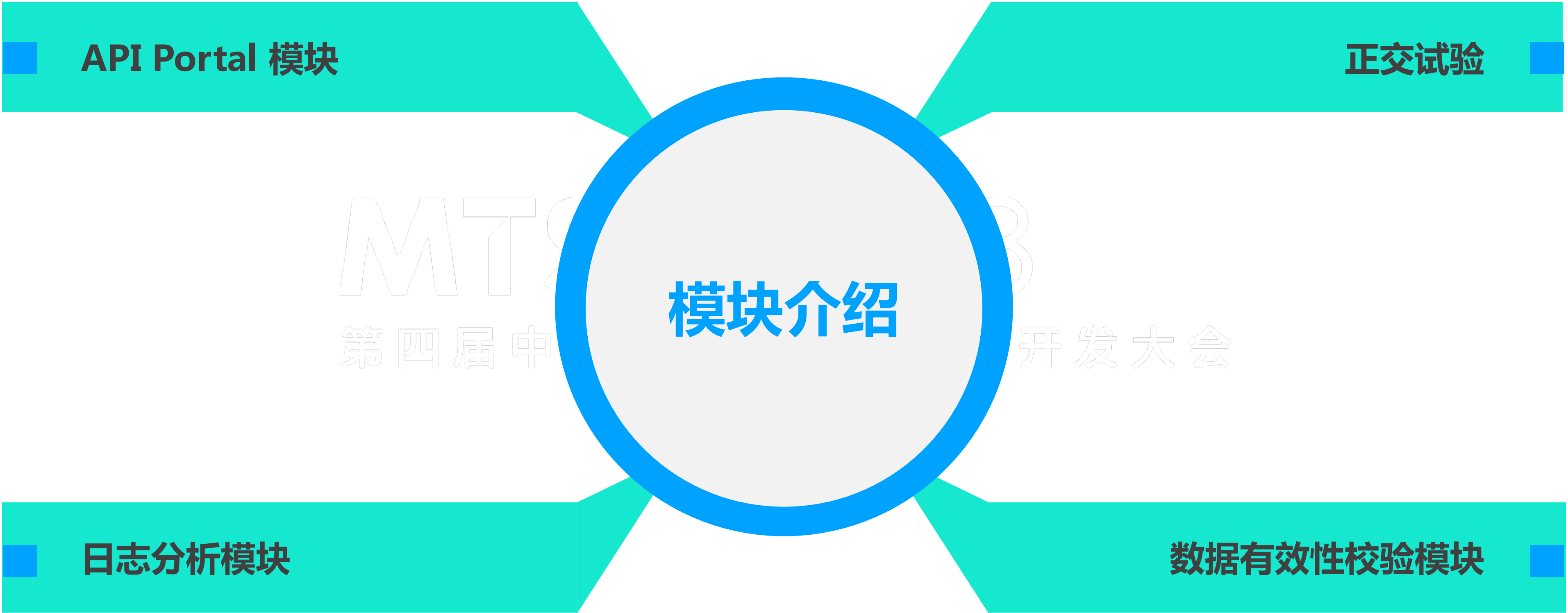
### 回放

一：收集方法的信息（方法名，参数）

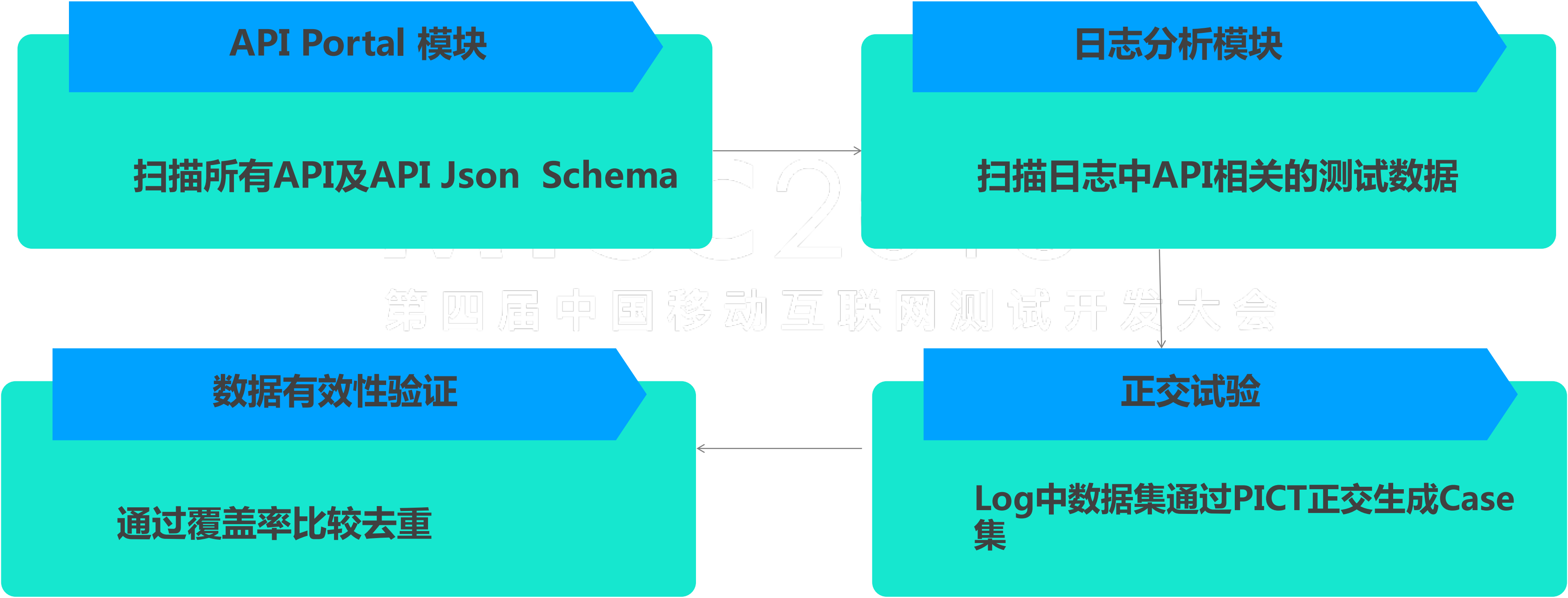
二：请求的方法名由之前录制的请求；返回之前请求返回的数据

三：如果没有命中，是否需要从新录





。



## API Portal 模块

扫描source code 获取接口信息

API信息

```
public interface INotifyService {  
    PageRepDto<EnterpriseListRepDto> getEnterpriseInfoList(EnterpriseListRepDto queryEnterpriseReqDtoPageQuery)
```

### API Portal 模块

#### API 信息

```
public class EnterpriseListRepDto {  
  
    private Long id;  
  
    private String cityId;  
  
    private String cityName;  
  
    private String enterpriseName;  
  
    private Integer start;  
  
    private Integer sex;  
  
    private String pinyin_name;  
  
    private Integer walle_id;  
  
    private Integer certificate_type;  
  
}
```

#### API Json Schema

```
{  
  "required": [  
    "enterpriseListRep"  
  ],  
  "type": "object",  
  "properties": {  
    "enterpriseListRep": {  
      "type": "object",  
      "properties": {  
        "cityId": {  
          "title": "",  
          "type": "integer",  
          "description": "",  
          "format": "int32"  
        },  
        "cityName": {  
          "title": "",  
          "type": "string",  
          "description": ""  
        },  
        "enterpriseName": {  
          "title": "",  
          "type": "string",  
          "description": ""  
        }  
      }  
    }  
  }  
}
```



### 日志分析模块

```
public class EnterpriseListRepDto {  
  
    private Long id;  
  
    private String cityId;  
  
    private String cityName;  
  
    private String enterpriseName;  
  
    private Integer start;  
  
    private Integer sex;  
  
    private String pinyin_name;  
  
    private Integer walle_id;  
  
    private Integer certificate_type;  
  
}
```

```
cityId: null^92  
start: 0  
walle_id: 2b5432439^289ib12578^0^2ib4064998^1000ib057246^100005ib3558  
sex: 1^2  
enterpriseName: #^饿了么  
certificate_type: 0  
cityName: #^北京^null  
id: 20209^22478^227449^19025^229305^200241^162497  
pinyin_name: ybl^lj^zyk^cqn^zmy^hxm^zhx
```

正交试验

cityId	start	walle_id	sex	enterpriseName	certificate_type	cityName	id	pinyin_name
92	0	2ib4064998	1	#	0	null	22478	zyk
null	0	100005ib3558	2	饿了么	0	北京	227449	zhx
92	0	1000ib057246	2	饿了么	0	#	22478	cqn
null	0	289ib12578	1	#	0	北京	162497	cqn
null	0	2b5432439	2	#	0	#	229305	zyk
92	0	01	饿了么	0	null	162497	zhx	
92	0	289ib12578	2	饿了么	0	null	20209	ybl
92	0	100005ib3558	1	饿了么	0	#	162497	zyk
null	0	1000ib057246	1	#	0	北京	200241	zyk
null	0	2ib4064998	2	饿了么	0	北京	19025	zmy
null	0	01	#	0	null	19025	zyk	
92	0	2b5432439	2	饿了么	0	null	200241	zmy
null	0	2ib4064998	1	#	0	#	227449	ybl
92	0	2b5432439	1	#	0	北京	229305	zhx
null	0	100005ib3558	1	#	0	#	20209	zmy
92	0	1000ib057246	2	饿了么	0	null	227449	zmy
null	0	02	饿了么	0	null	227449	cqn	
92	0	2ib4064998	2	饿了么	0	#	20209	zhx
null	0	100005ib3558	1	饿了么	0	null	229305	zmy
null	0	01	饿了么	0	北京	20209	hxm	
null	0	01	饿了么	0	北京	229305	ybl	
92	0	289ib12578	1	饿了么	0	#	19025	ybl
92	0	289ib12578	1	饿了么	0	#	227449	zyk
null	0	01	#	0	#	200241	cqn	
null	0	289ib12578	2	饿了么	0	北京	22478	lj
92	0	100005ib3558	1	#	0	北京	200241	ybl



### 覆盖率去重

cityId	start	walle_id	sex	enterpriseName	certificate_type	cityName	id	pinyin_name
92	0	2ib4064998	1	# 0	null	22478	zyk	
null	0	100005ib3558	2	饿了么 0	北京	227449	zhx	
92	0	1000ib057246	2	饿了么 0	#	22478	cqn	
null	0	289ib12578	1	# 0	北京	162497	cqn	
92	0	2b5432439	2	# 0	null	22478	ybl	
null	0	1000ib057246	1	饿了么 0	北京	19025	zhx	
null	0	100005ib3558	2	饿了么 0	null	22478	hxm	
92	0	1000ib057246	2	# 0	null	162497	ybl	
~								

一次使用即失效的数据

业务强依赖的数据

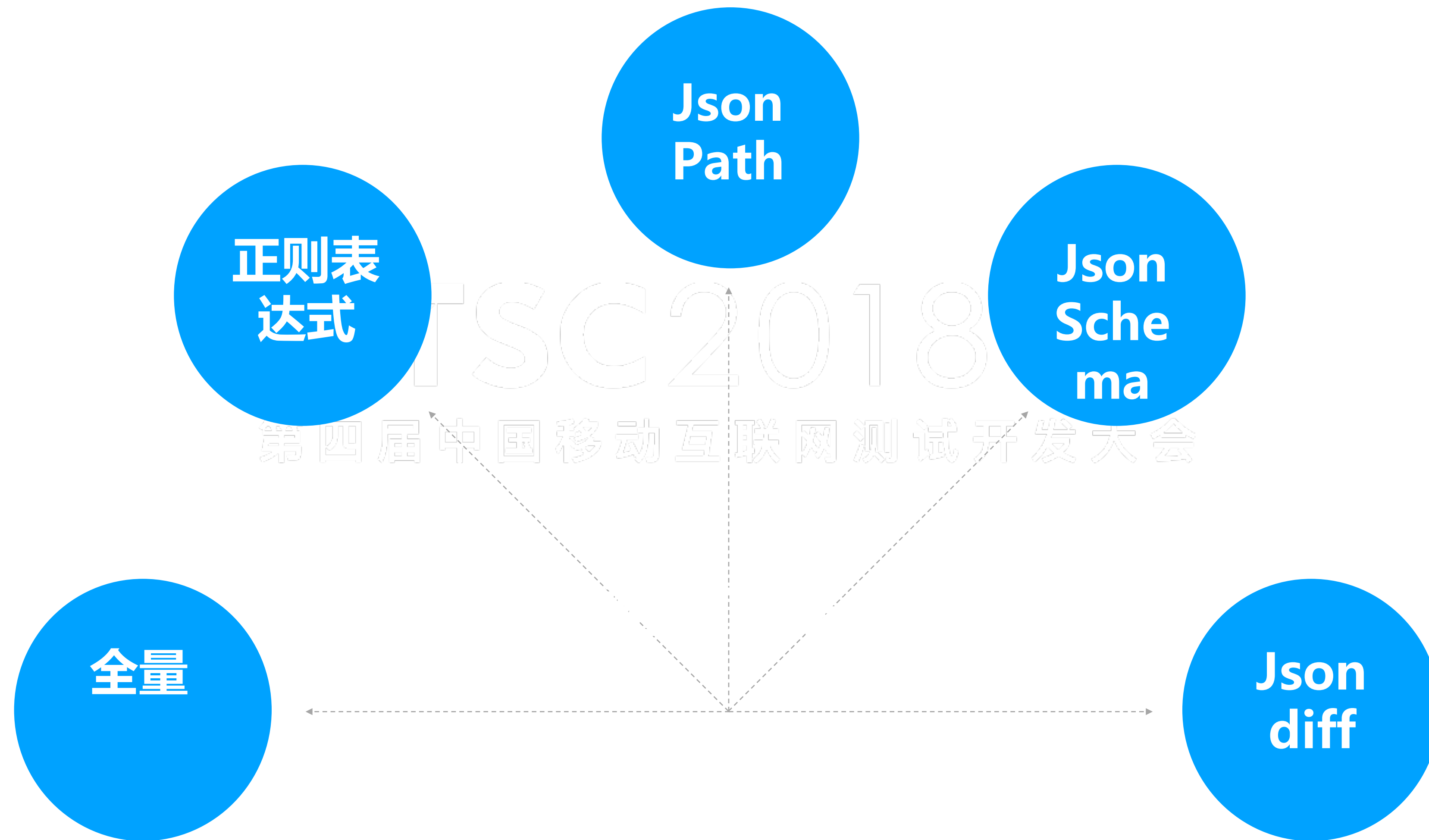
逻辑简单的API

	args数量	PICT生产/去除有效数据	LOG中Case数量	PICTCase生成数量	PICT去重Case数量
	1	-6	9	9	3
e	2	-4	9	9	5
	2	0	81	9	9
ce	3	-9	14	14	5
e	3	-5	9	36	4
ce	3	-11	14	14	3
ce	3	-9	12	12	3
e	4	+1	9	81	10
ce	4	-1	4	6	3
ce	5	-2	5	9	3
ce	5	0	4	6	4
ce	6	-3	6	9	3
e	6	+4	4	17	8
ce	6	+2	11	42	13
e	7	+5	7	49	12
ce	13	+5	8	99	13
ce	15	+1	2	8	3
ce	15	-5	8	31	3
ce	19	0	2	2	2
ce	22	+2	4	10	6
ce	22	-3	6	8	3
ce	24	-9	11	80	2
ce	25	-1	11	13	10
ce	26	-19	21	25	2
ce	26	-7	15	31	8
ce	32	-7	13	36	6

args数量居中，PICT可以生成较多有效的数据; args数量较少或较多时，去重作用明显

# 03

## 数据的有效性验证





**coverage**

**Mutation  
Score**

谢谢大家

MTSC2018

第四届中国移动互联网测试开发大会

TesterHome