分词器简介

Analyzer

职责

```
{
  "tokens" : [
    {
      "token" : "test",
      "start_offset" : 0,
      "end_offset" : 4,
      "type" : "<ALPHANUM>",
      "position" : 0
    },
    {
      "token" : "case",
      "start_offset" : 5,
      "end_offset" : 9,
      "type" : "<ALPHANUM>",
      "position" : 1
    }
  ]
}
```

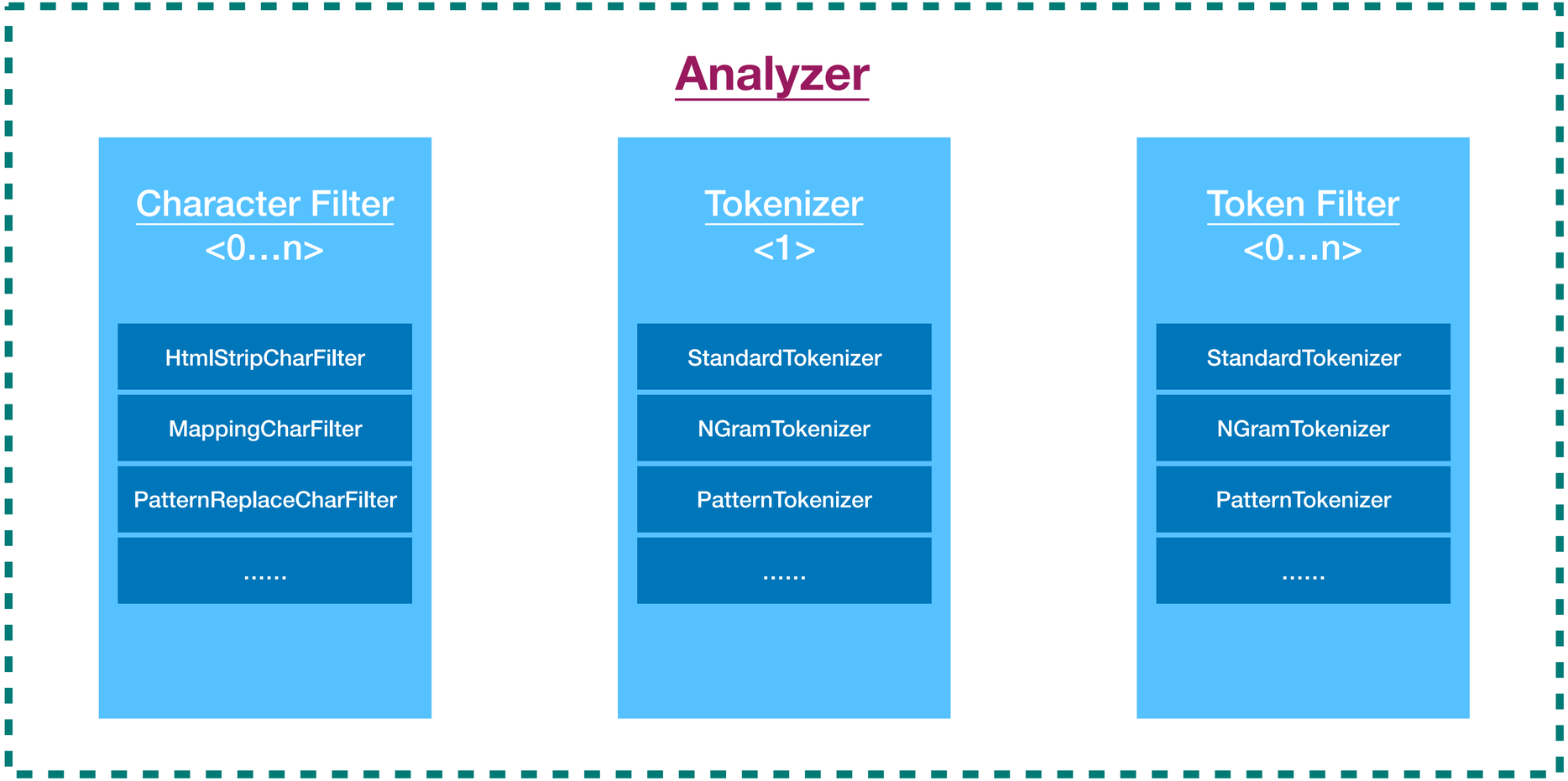
字符转换

按照分词规则切分单词

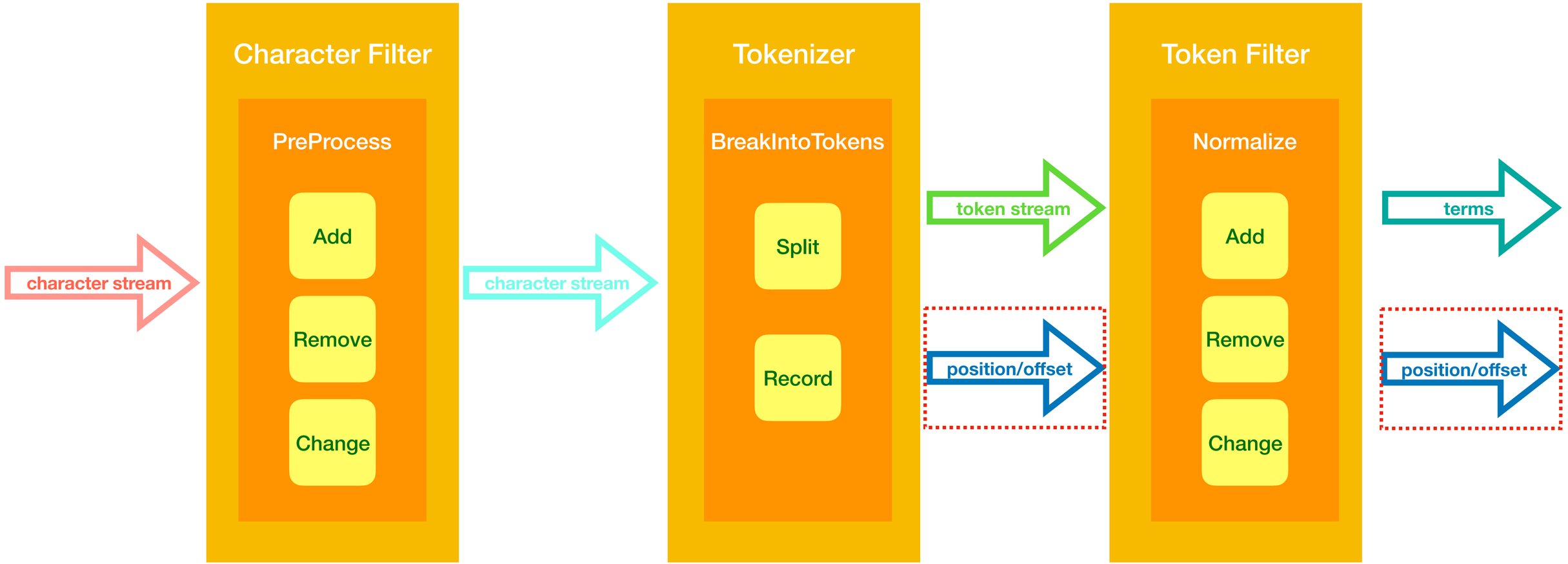
记录词间相对位置

记录单词原始偏移

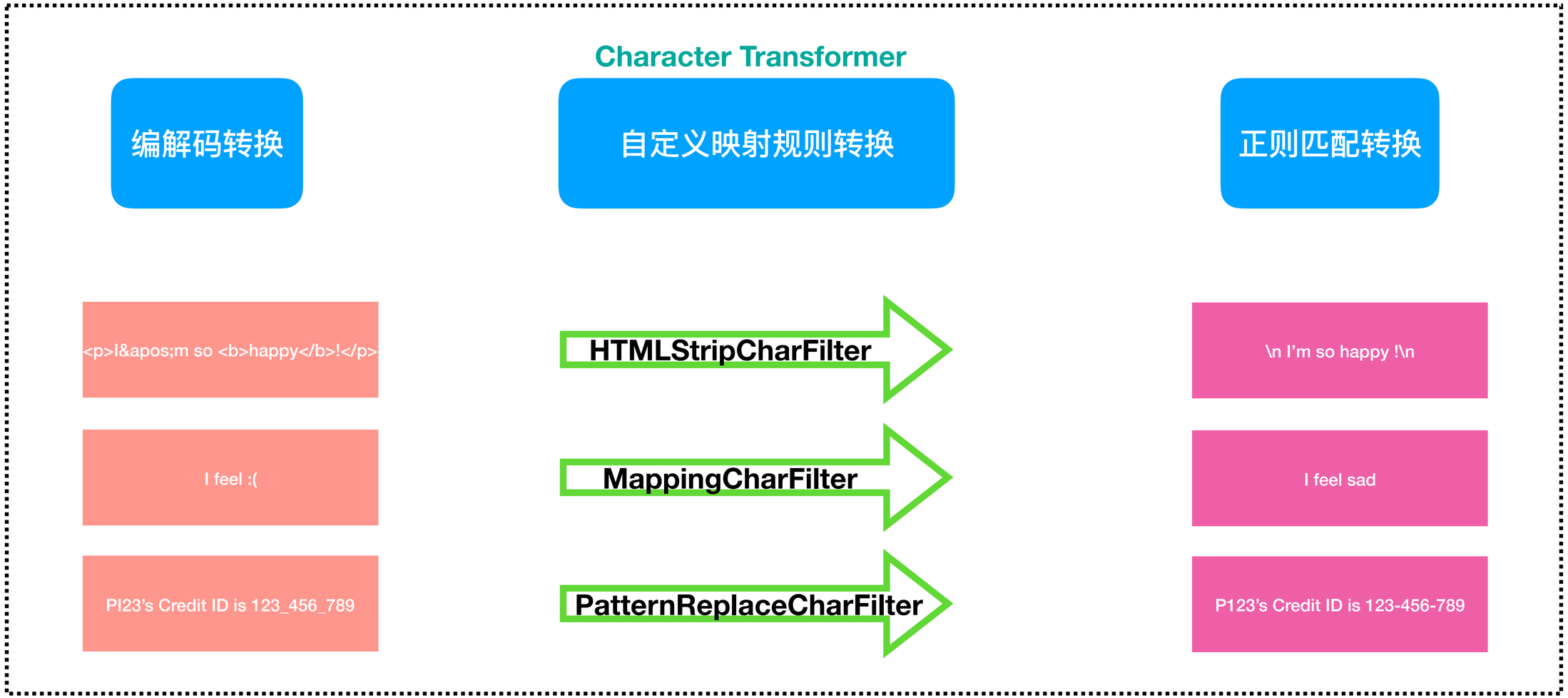
组成元素


Analyzer
Character Filter
<0…n>
HtmlStripCharFilter
MappingCharFilter
PatternReplaceCharFilter
.....
Tokenizer
<1>
StandardTokenizer
NGramTokenizer
PatternTokenizer
.....
Token Filter
<0…n>
StandardTokenizer
NGramTokenizer
PatternTokenizer
.....

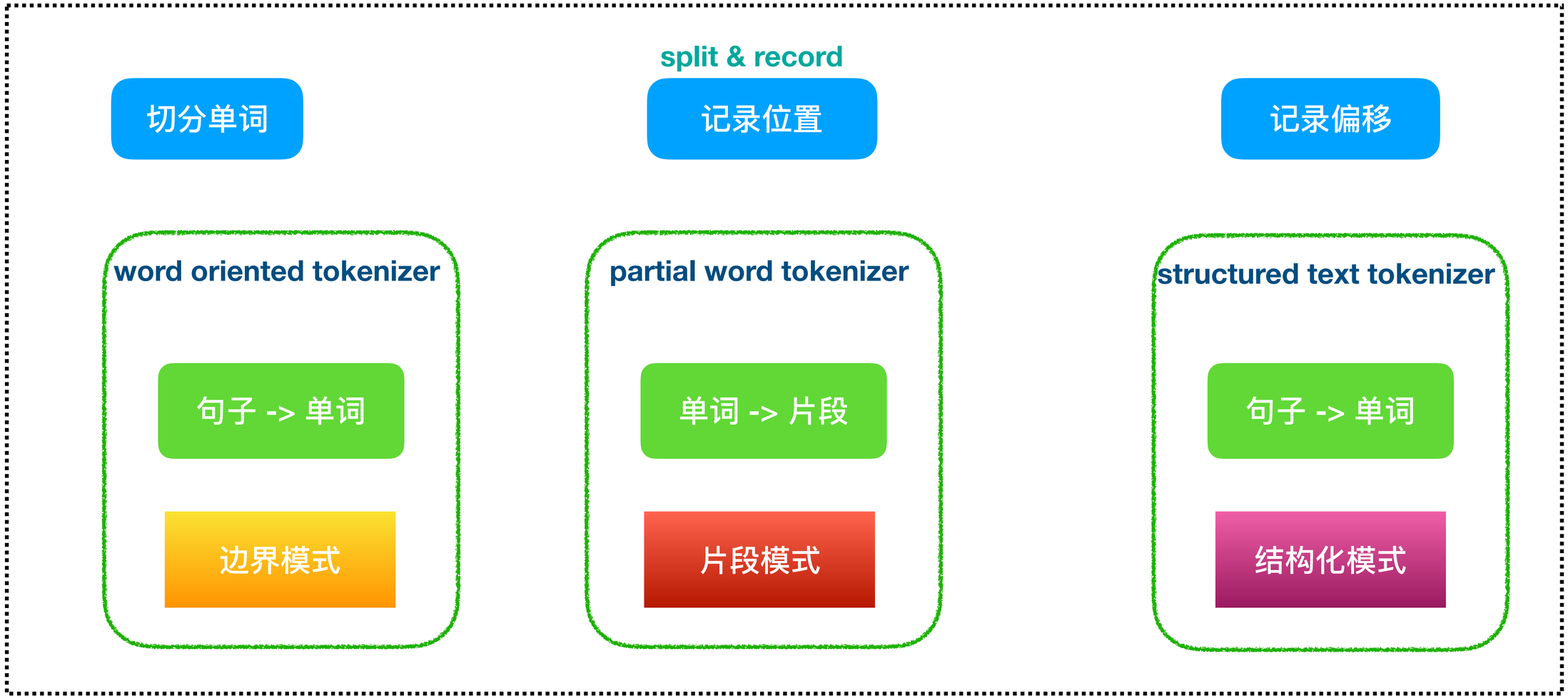
处理流程


Character Filter
PreProcess
Add
Remove
Change
Tokenizer
BreakIntoTokens
Split
Record
Token Filter
Normalize
Add
Remove
Change
character stream
character stream
token stream
position/offset
terms
position/offset

元素1：CharacterFilter



元素2：Tokenizer



元素2-1： WordOrientedTokenizer

The 2 QUICK Brown-Foxes jumped over the lazy dog's bone.

Standard Tokenizer

Letter Tokenizer

Lowercase Tokenizer

WhiteSpace Tokenizer

UAX URL Email Tokenizer

Classic Tokenizer

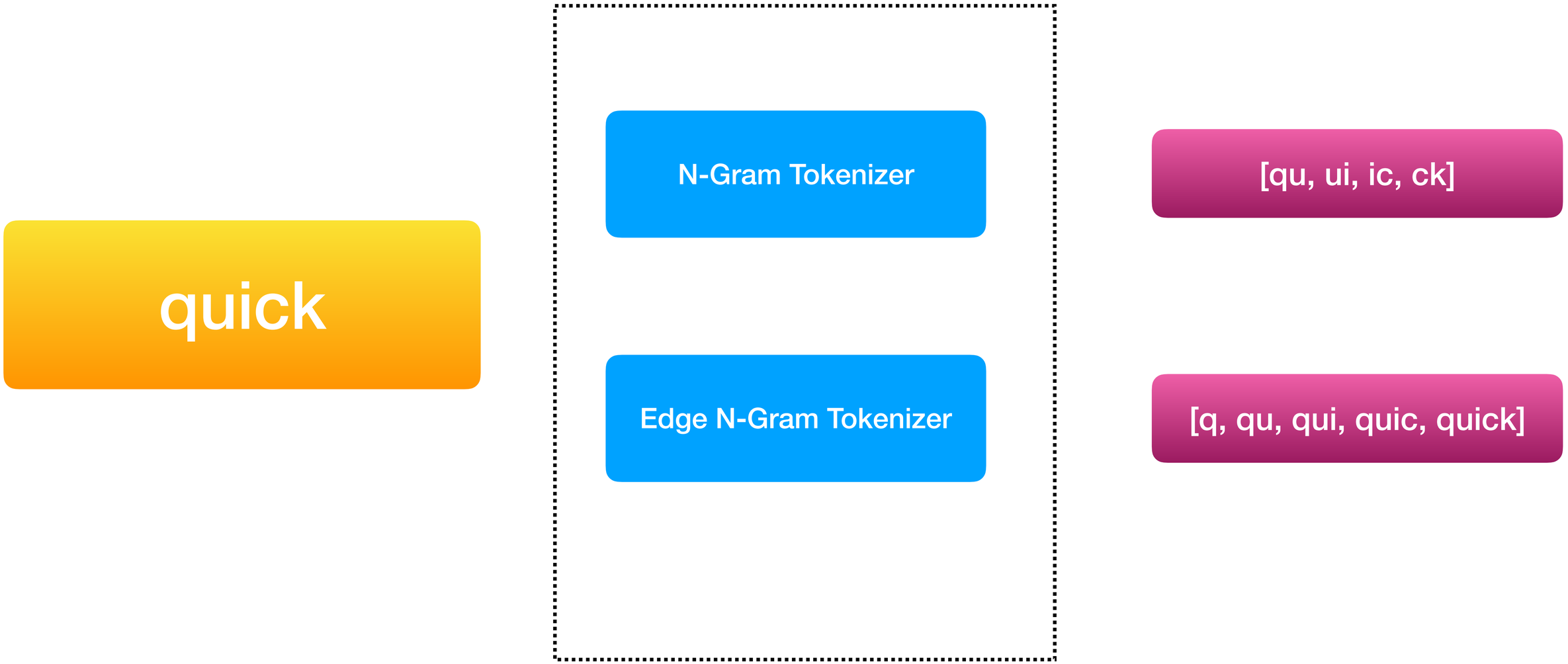
Thai Tokenizer

[The, QUICK, Brown, Foxes, jumped, over, the, lazy, dog, s, bone]

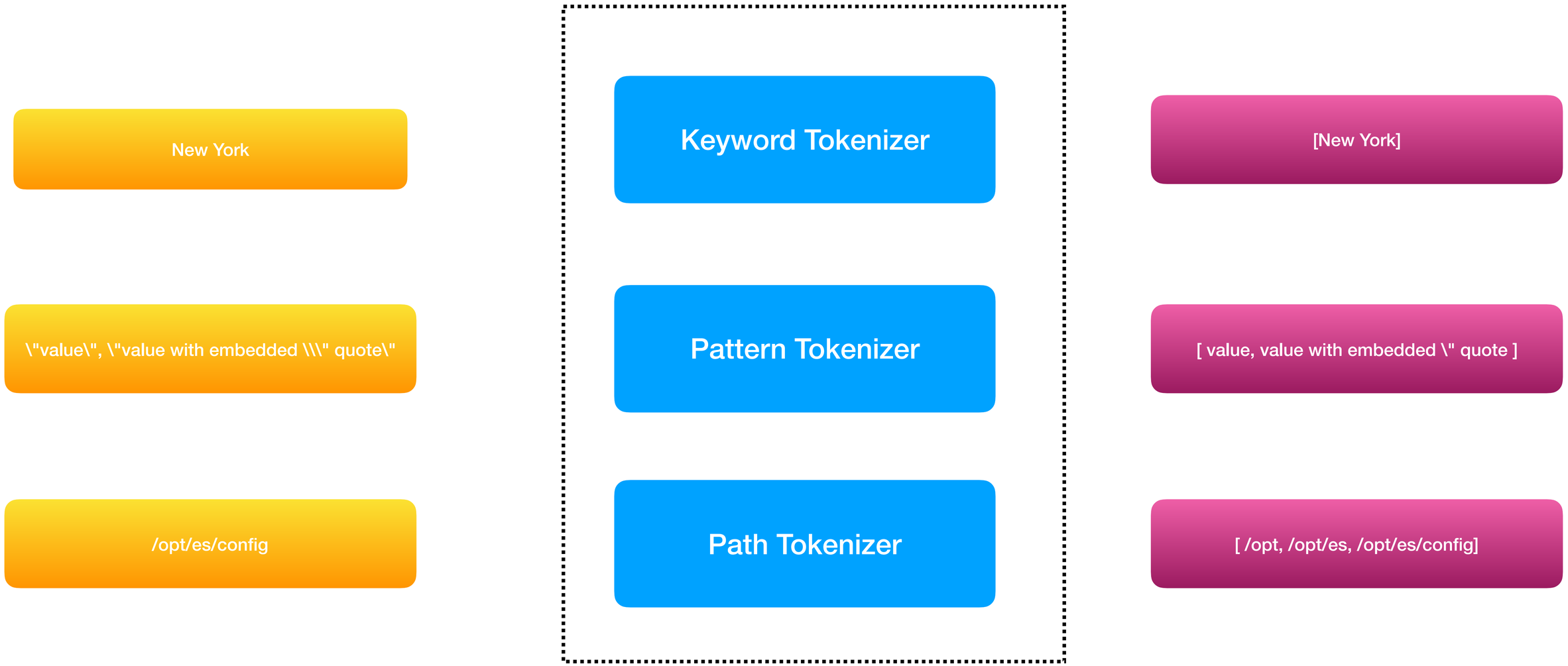
[the, quick, brown, foxes, jumped, over, the, lazy, dog, s, bone]

[The, 2, QUICK, Brown-Foxes, jumped, over, the, lazy, dog's, bone.]

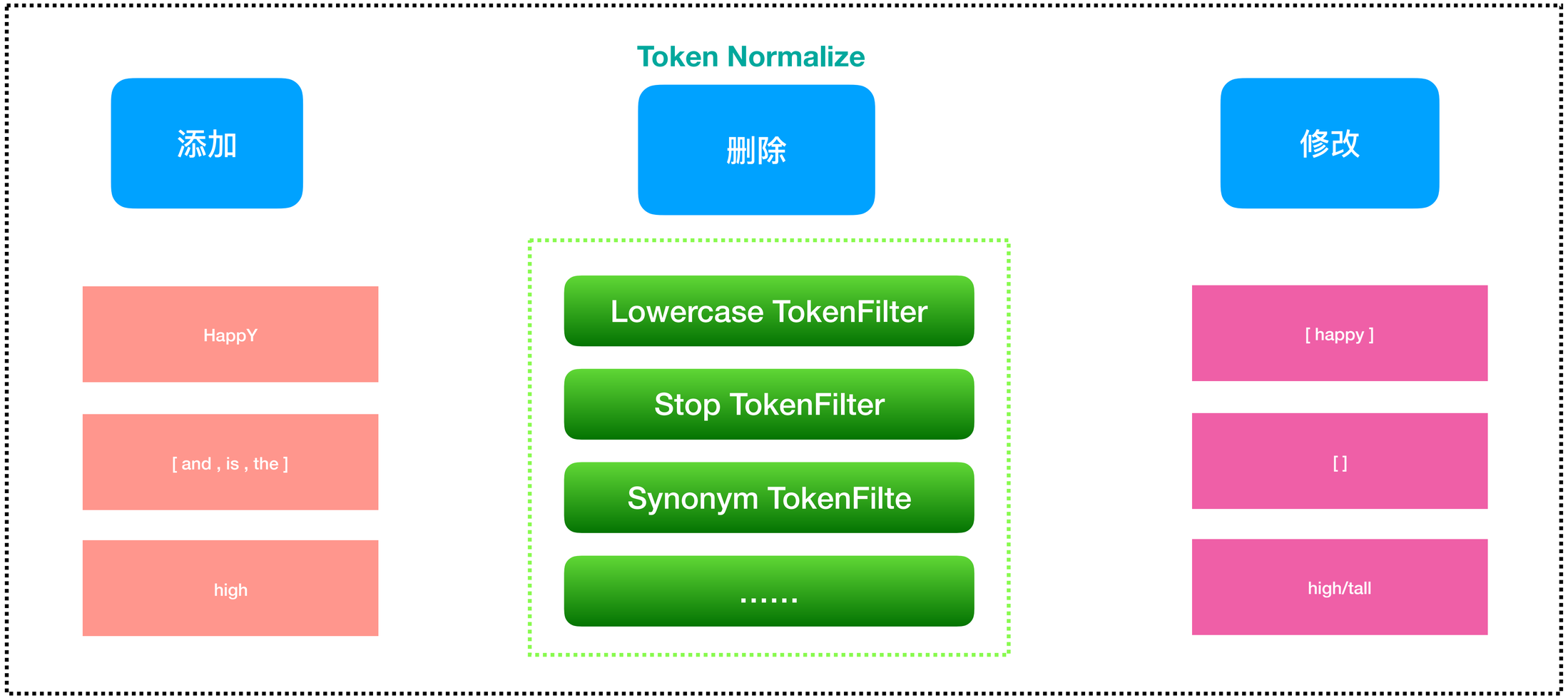
元素2-2：PartialWordTokenizer



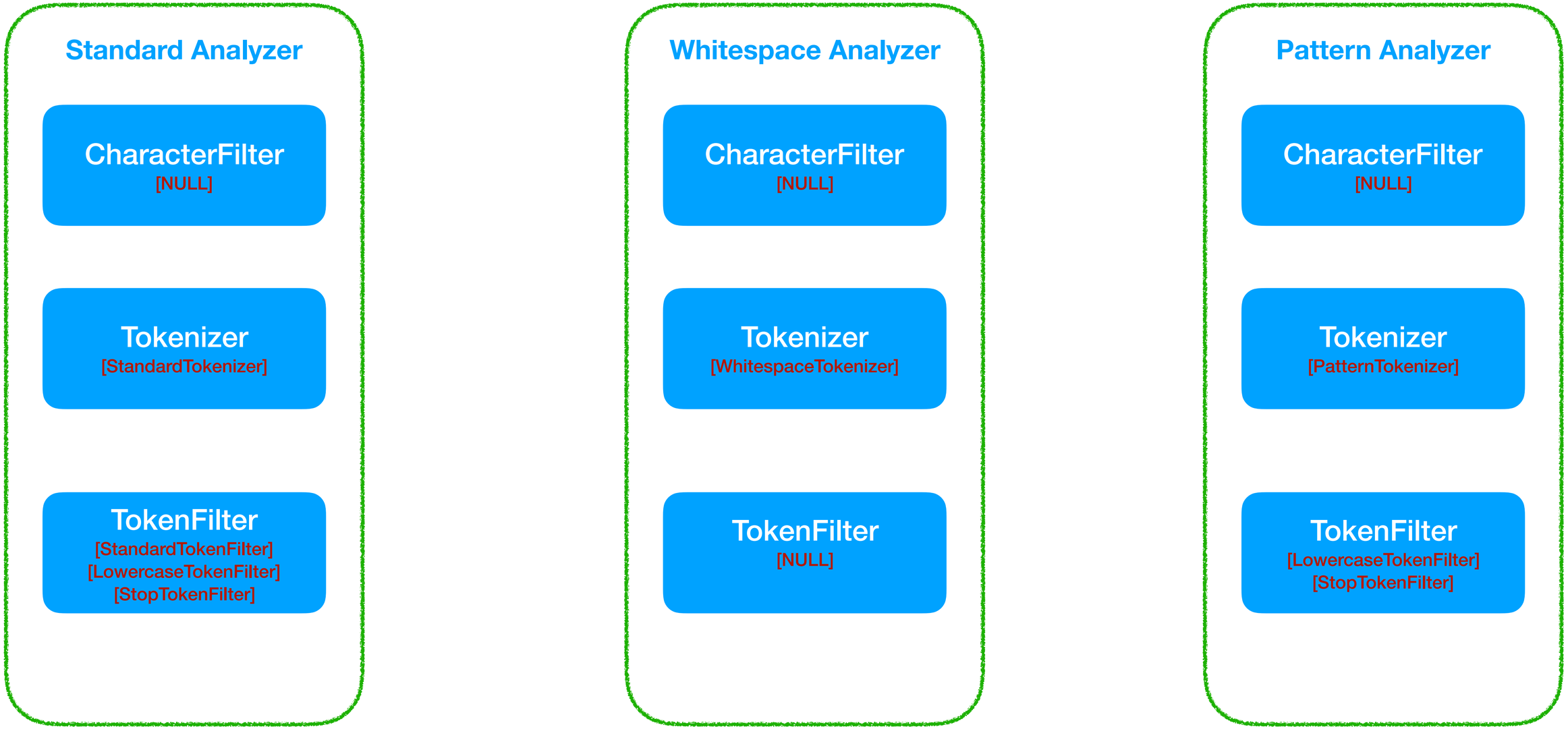
元素2-3：StructedTextTokenizer



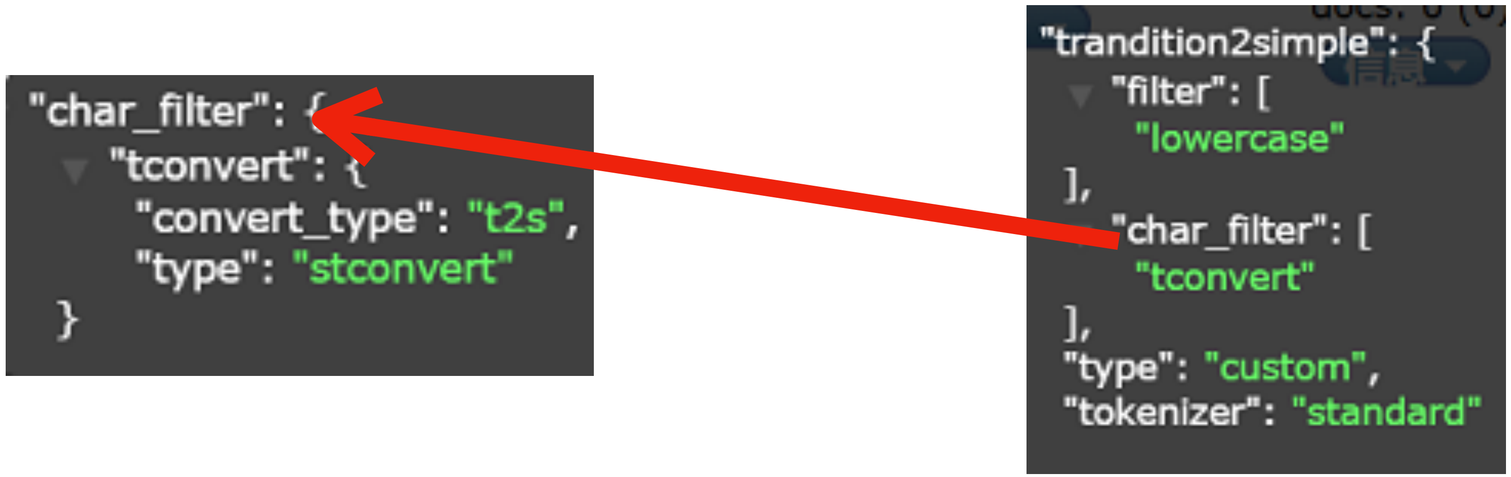
元素3：TokenFilter



内置分词器


Standard Analyzer
CharacterFilter
[NULL]
Tokenizer
[StandardTokenizer]
TokenFilter
[StandardTokenFilter]
[LowercaseTokenFilter]
[StopTokenFilter]
Whitespace Analyzer
CharacterFilter
[NULL]
Tokenizer
[WhitespaceTokenizer]
TokenFilter
[NULL]
Pattern Analyzer
CharacterFilter
[NULL]
Tokenizer
[PatternTokenizer]
TokenFilter
[LowercaseTokenFilter]
[StopTokenFilter]

自定义配置分词器


"char_filter": {
"tconvert": {
"convert_type": "t2s",
"type": "stconvert"
}
"trandition2simple": {
"filter": [
"lowercase"
],
"char_filter": [
"tconvert"
],
"type": "custom",
"tokenizer": "standard"

思考

1: CharacterFilter VS TokenFilter

2：在搜索阶段使用的分词器与索引阶段使用的分词器是否应该保持一致

3：高亮是依托于分词器的哪一部分输出实现的

4：filter中为何会存在“add”操作

5：分词过程是否只存储单词/位置/偏移就足够了

如何开发一个分词器

分析源码：

扩展点-org.elasticsearch.index.analysis

- Analyzer
- AnalyzerProvider
- TokenFilter
- TokenFilterFactory
- Tokenizer
- TokenizerFactory

* CharTermAttribute
* OffsetAttribute
* createComponent
* incrementToken

注册点-org.elasticsearch.plugin.analysis

- Plugin
- AnalysisPlugin

殊途同归：<扩展+注册> vs <spring ioc>

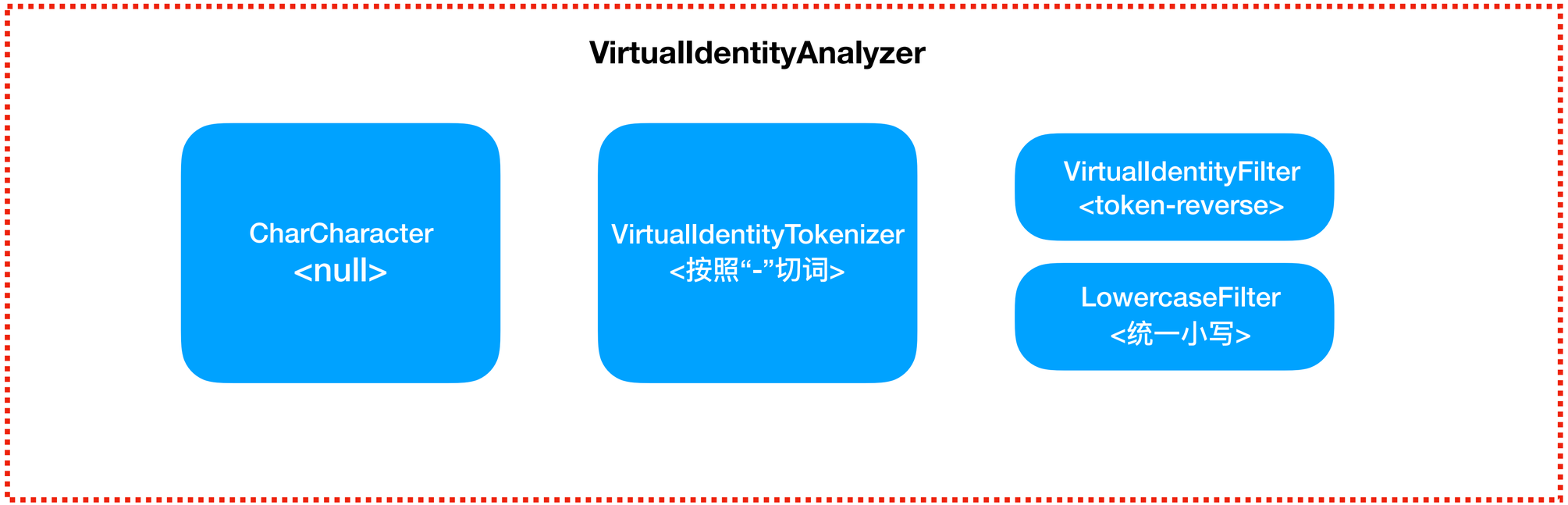
esplugin开发流程/licence/lucence-api/testcase/...

虚拟场景

情形描述：一批虚拟身份数据，需要查询具有某一特征的对象，其中某一虚拟社区的人具有相同的身份后缀

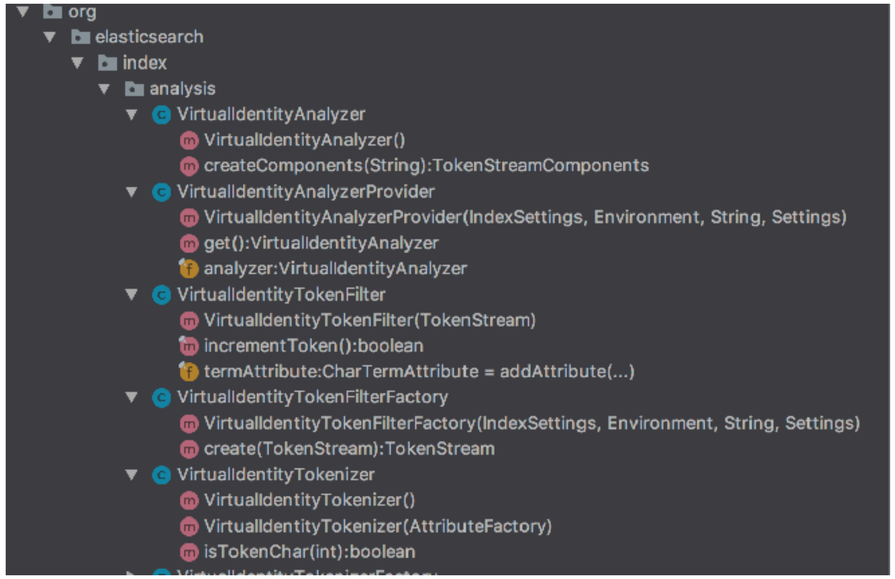
样例： xxxgoV-yyyaRmy-zZzsinacom

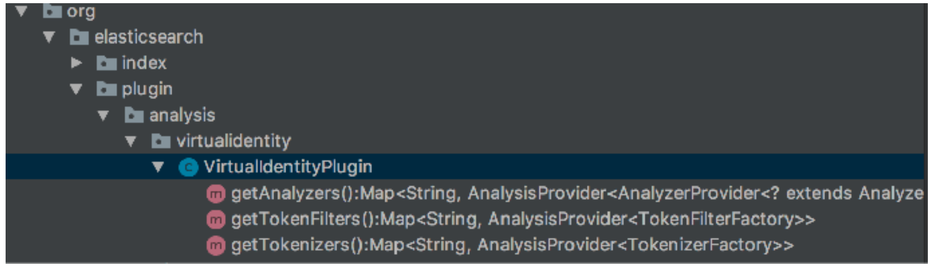
分析：一人具有多重身份，以”-”分割。找到如何条件的人需要根据后缀匹配。比如gov类的代表政府机关。

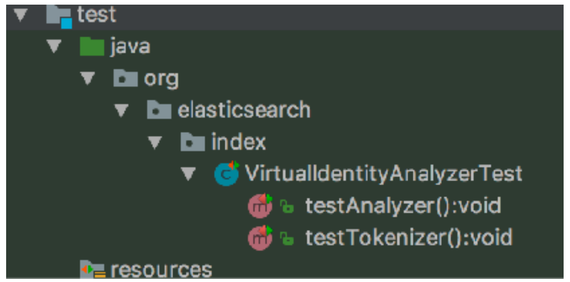


虚拟场景

代码结构


org
elasticsearch
index
analysis
VirtualIdentityAnalyzer
VirtualIdentityAnalyzer()
createComponents(String):TokenStreamComponents
VirtualIdentityAnalyzerProvider
VirtualIdentityAnalyzerProvider(IndexSettings, Environment, String, Settings)
get():VirtualIdentityAnalyzer
analyzer:VirtualIdentityAnalyzer
VirtualIdentityTokenFilter
VirtualIdentityTokenFilter(TokenStream)
incrementToken():boolean
termAttribute:CharTermAttribute = addAttribute(...)
VirtualIdentityTokenFilterFactory
VirtualIdentityTokenFilterFactory(IndexSettings, Environment, String, Settings)
create(TokenStream):TokenStream
VirtualIdentityTokenizer
VirtualIdentityTokenizer()
VirtualIdentityTokenizer(AttributeFactory)
isTokenChar(int):boolean


org
elasticsearch
index
plugin
analysis
virtualidentity
VirtualIdentityPlugin
getAnalyzers():Map<String, AnalysisProvider<AnalyzerProvider<? extends Analyze
getTokenFilters():Map<String, AnalysisProvider<TokenFilterFactory>>
getTokenizers():Map<String, AnalysisProvider<TokenizerFactory>>


test
java
org
elasticsearch
index
VirtualIdentityAnalyzerTest
testAnalyzer():void
testTokenizer():void
resources

分词效果

```
[magneto@jingkai logs]$ curl -XGET 192.168.50.194:9200/_analyze?pretty -d '{"analyzer":"virtualidentity","text":"sinAcomcn-yahOocomCN"}'
{
  "tokens" : [
    {
      "token" : "ncmocanis",
      "start_offset" : 0,
      "end_offset" : 9,
      "type" : "word",
      "position" : 0
    },
    {
      "token" : "ncmocoohay",
      "start_offset" : 10,
      "end_offset" : 20,
      "type" : "word",
      "position" : 1
    }
  ]
}
```