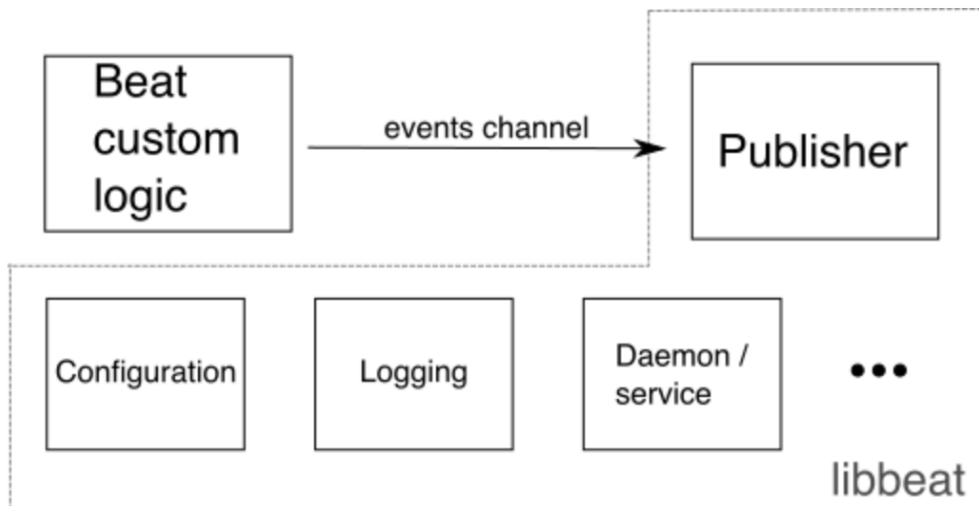


Beats 主要由两部分组成：

1. 数据抓取组件
2. 数据发布组件



数据发布组件为不同 beats 的公共部分，不同的 Beats 主要是在数据抓取部分实现不同的逻辑。官方提供了标准的 Beats 接口，至少实现 Run 和 stop：

```
1 type Beater interface {
2     // The main event loop. This method should block until signalled to stop
    by an
3     // invocation of the Stop() method.
4     Run(b *Beat) error
5
6     // Stop is invoked to signal that the Run method should finish its execu
    tion.
7     // It will be invoked at most once.
8     Stop()
9 }
```

为了实现 Beater 结构，一个 beat 对象通常需要包含：

`done`: Channel used by the `Run()` method to stop when the `Stop()` method is called.

`config`: Configuration options for the Beat

`client`: Publisher that takes care of sending the events to the defined output.

```
1 type Countbeat struct {
2     done chan struct{}
```

```

3  config config.Config
4  client publisher.Client
5  ...
6  }
7
8  func (bt *Countbeat) Run(b *beat.Beat) error {
9  ...
10 }
11
12 func (bt *Countbeat) Stop() {
13 ...
14 }

```

每个 beat 对象同样需要实现 New()

New() Creates the Beat object

Run() Contains the main application loop that captures data and sends it to the defined output using the publisher

Stop() Contains logic that is called when the Beat is signaled to stop