Game Event Processing System

Scenario

You are building a simple game server that processes different types of in-game events (combat, chat, trading, achievements).

The system must:

- Ensure only one server config exists
- Create events without hard-coding concrete classes
- Process events using pluggable behaviors
- Store event data immutably and safely
- Be adaptable by allowing us to add new event types without modifying existing processing code

Hints:

- Think of our recent topics.
- Events can be stored in a collection. Java List works well to begin with. Maybe other collections are useful but they can make it more complicated.
- We have not talked about streams, but they are good use for processing and filtering these events.
- Server configuration exists once
- Event creation decides what to produce
- One-off behavior created at use time
- Stateless logic for filtering & transforming of events
- See the conceptual design on the next page
- This is not meant to be multithreaded. Do not get to hung up thinking about EventHandler as something related to threading.

Conceptual Design

```
+-------------------+
|   Configuration   |
|-------------------|
| • Exists once     |
| • Holds server    |
|   settings        |
| • Globally        |
|   accessible      |
+----------+--------+
           |
           v
+-------------------+          +-------------------+
|  Event Creation   | ----->   |    Event Data     |
|-------------------|          |-------------------|
| • Given raw input |          | • Immutable       |
|   (type, player,  |          | • Represents one  |
|   value)          |          |   occurrence      |
| • Decides what    |          | • No behavior     |
|   kind of event   |          +-------------------+
|   to produce      |
+----------+--------+
           |
           v
+-------------------+
|  Event Processing |
|-------------------|
| • Receives events |
| • Applies rules   |
| • Selects logic   |
|   based on event  |
|   type or data    |
+----------+--------+
           |
           v
+-------------------+          +-------------------+
|  Event Behavior   | <---->   |  One-Off Behavior |
|-------------------|          |-------------------|
| • Defines what it |          | • Inline logic    |
|   means to handle |          | • Created at use  |
|   an event        |          |   time            |
| • Can vary at     |          | • No reuse needed |
|   runtime         |          +-------------------+
+----------+--------+
           |
           v
+-------------------+
| Event Queries /   |
| Transformations   |
|-------------------|
| • Filter events   |
| • Aggregate data  |
| • Produce output  |
| • Stateless logic |
+-------------------+
```