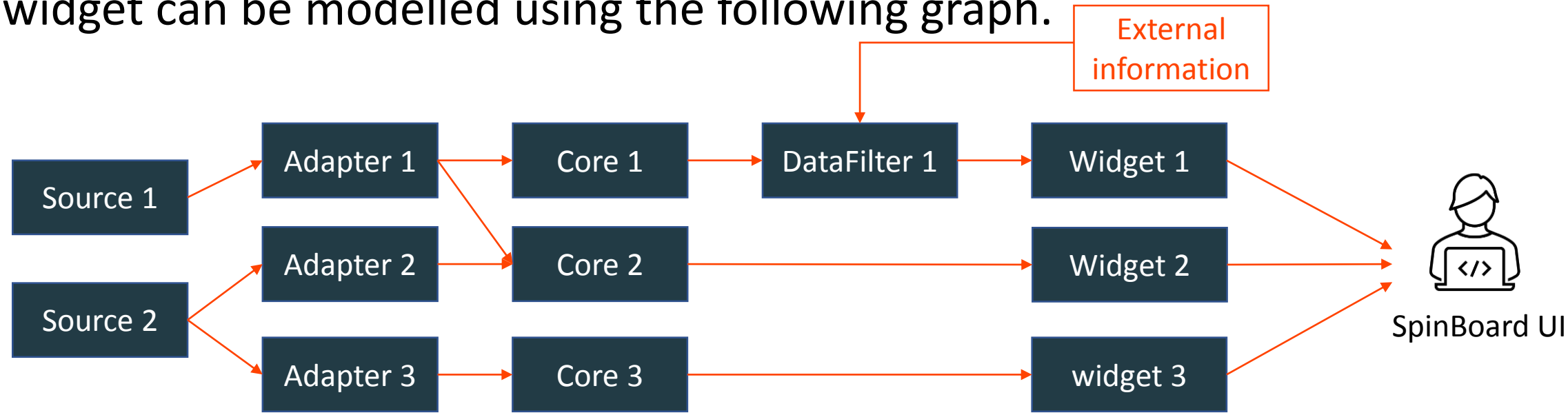
SpinBoard Custom Adapter Development

SpinBoard Architecture

The chain structure of the backend

A general setup containing two different sources and three different widget can be modelled using the following graph.



- Widget 1 gets the data from Source 1 through Adapter 1 with a data filter layer.
- Widget 2 contains data from Sources 1 and 2 through respectively Adapters 1 and 2
- Widget 3 gets the data from Source 2

SpinBoard Architecture

The adapter as a stateless mapping component

The goal of an adapter is to map the data from an external source to an internal representation.

The tool provides a general Interface receiving a generic EventMessage containing the data from a source and returning a list of Rows.

```
public interface Adapter {
    List<Row> baseAdapt(EventMessage<?> message);
}
```

Every customer can implement a set of adapters based on the widget requirements:

```
private List<Row> buildRow(EventMessage<KafkaMessage> message) {
    Map<Long, Cell> cells = new HashMap<Long, Cell>();

    cells.put(NAME, CellBuilder.builder().withDisplayValue(message.getName()).build());
    cells.put(ID, CellBuilder.builder().withDisplayValue(message.getId()).build());
    cells.put(PRICE, CellBuilder.builder().withDisplayValue(message.getPrice()).build());

    Row finalRow = newRowBuilder().withId(message.getId()).withAction(INSERT).withCells(cells).build();

    return Lists.newArrayList(finalRow);
}
```

For every KafkaMessage a row containing three cells is created

Then the resulting row is returned to the tool for the next step of the process

Thank You!

Contact us at:

Info@t-spin.com

Visit us at:

https://www.t-spin.com