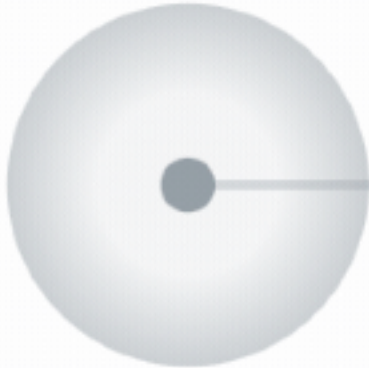






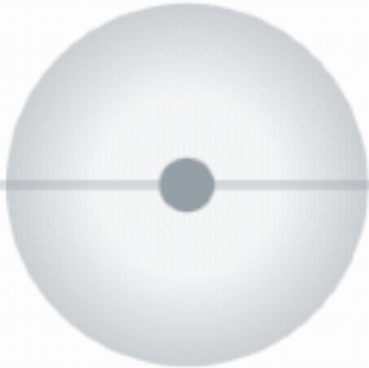
implementing the chart with

Customer



1

Active carts



1

Checking out



0

Purchased

← Open custom settings



```
const ordersRunningDots = useSprings(  
  newOrders.length,  
  newOrders.map((item, index) => ({
```

```
const radiusSprings = useSprings(  
  data.length,  
  data.map((dataPoint, index) => ({  
    radiusScale: radiusScale(dataPoint, count)  })))
```











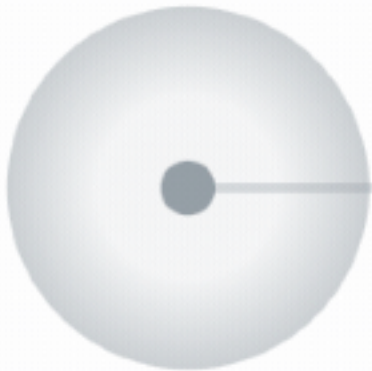
```
<CustomerBehaviorCanvas
```

```
radiusSprings= {bucketsRadius}
```

```
ordersRunningDots= {ordersRunningDots}
```

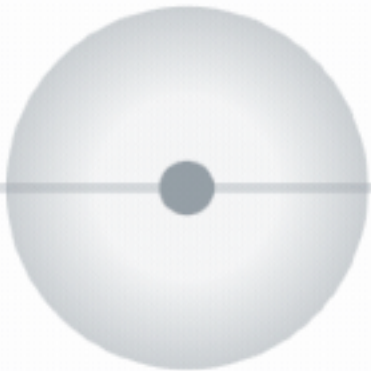
```
>
```

Customer  
.....



1

Active carts



1

Checking out

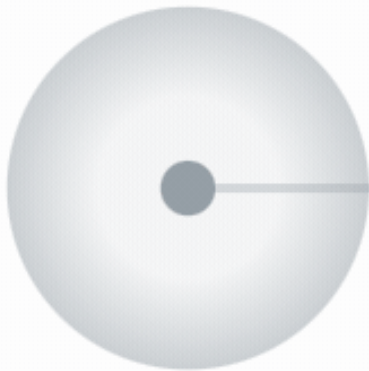


0

Purchased

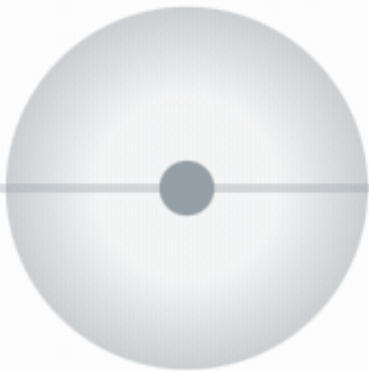
← Open custom settings

Customer  
.....



1

Active carts



1

Checking out



0

Purchased

← Open custom settings

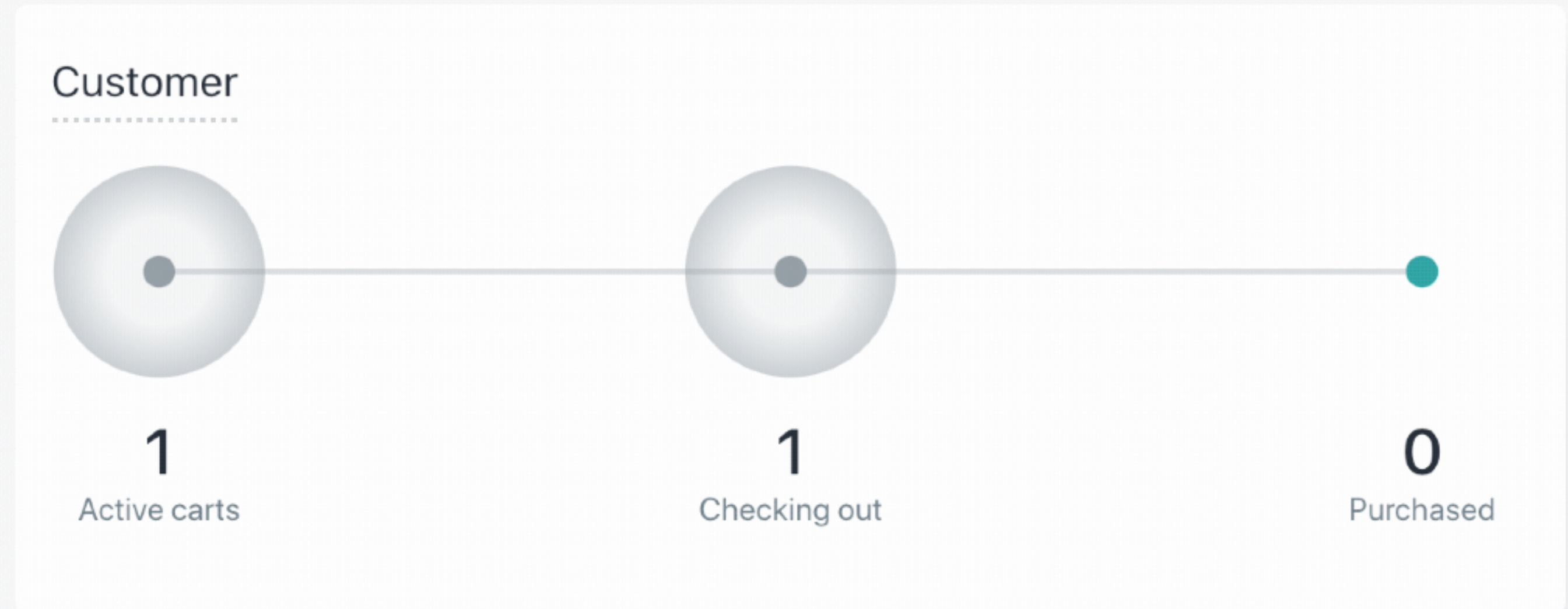


# Re-implementing the chart with Canvas

```
const radiusSprings = useSprings(  
  // ...  
);
```

```
const ordersRunningDots = useSprings(  
  // ...  
);
```

```
<CustomerBehaviorCanvas  
  radiusSprings={bucketsRadius}  
  ordersRunningDots={ordersRunningDots}  
>
```





# Re-implementing the chart with Canvas

```
const drawCircle = ({ context, x, y, radius, fillColor }) => {  
  context.beginPath();  
  context.arc(x, y, radius, 0, Math.PI * 2, false);  
  context.fillStyle = fillColor;  
  context.fill();  
};
```

```
const drawRect = ({ context, x, y, width, height, fillColor }) => {  
  context.fillStyle = fillColor;  
  context.fillRect(x, y, width, height);  
};
```

```
const clearCanvas = () => {  
  context.clearRect(0, 0, width, height);  
};
```