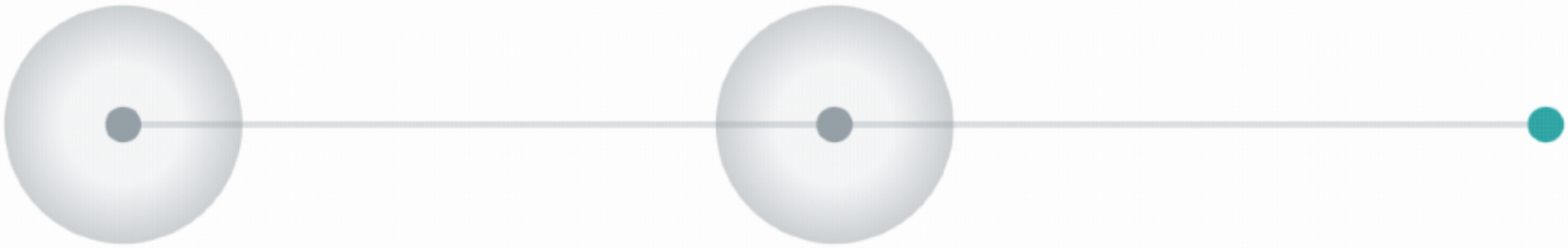
Customer



1
Active carts

1
Checking out

0
Purchased

Open custom settings

```
useEffect(() => {
  if (orders.count > prevOrders.count) {

    let increment = totalOrders.count - prevTotalOrders.count;
    increment = clamp(increment, 1, maxCirclesPerBucket);

    const newOrders = Array(increment)
      .fill({
        // "Checking out" bucket position:
        fromCx:  xScale(1),
        // "Purchased" bucket position:
        toCx: xScale(2),
        // how big the radius should be based
        // on the increment size:
        radius: getRunningDotRadius(increment)
      });

    setNewOrders(newOrders);
  }
}, [data]);
```

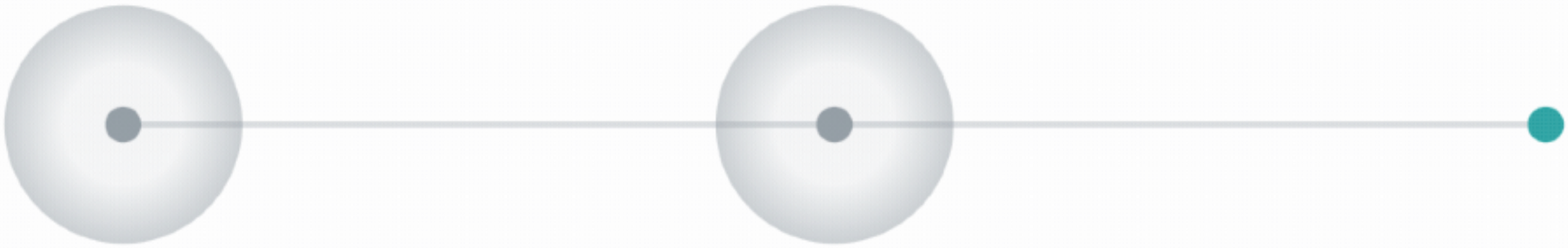
Run dots,

```
const newOrders = Array();

console.log(newOrders)   []
```

```
const getRunningDotRadius = (count) ⇒
  count > bigIncrement ? runningDotRadiusBig : runningDotRadius;
```

Customer



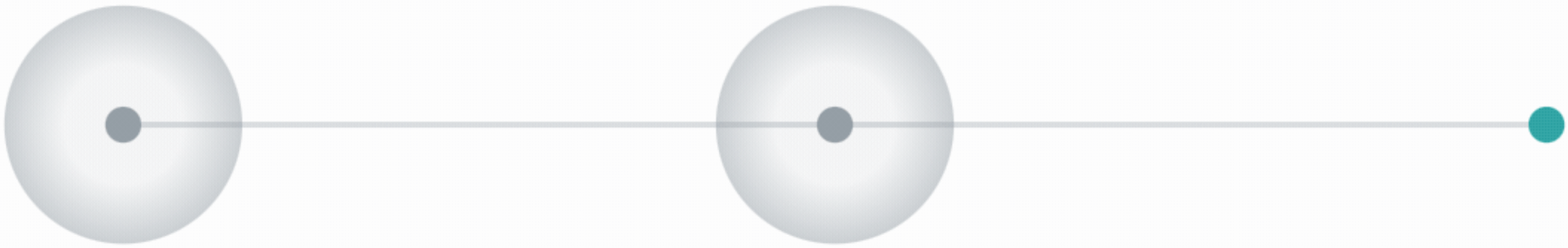
1
Active carts

1
Checking out

0
Purchased

Open custom settings

Customer



1
Active carts

1
Checking out

0
Purchased

Open custom settings

```
const newOrders = Array();

console.log(newOrders)   []
```

```
const ordersRunningDots = useSprings(
  newOrders.length,
  newOrders.map((item, index) => ({
    delay: index * springDelay,
    from: {
      cx: item.fromCx,
      fill: rgba(gray, 1),
      radius: item.radius,
    },
    to: {
      cx: item.toCx,
      fill: rgba(green, 1),
      radius: item.radius,
    },
  }))
);
```