Concurrent Programming Using The Disruptor

Trisha Gee, Developer at LMAX
@trisha_gee
mechanitis.blogspot.com

The Disruptor?

What I'm covering

- Overview of the Disruptor
- Create your own!
- Turn it up to Eleven
- Q&A

What is it?

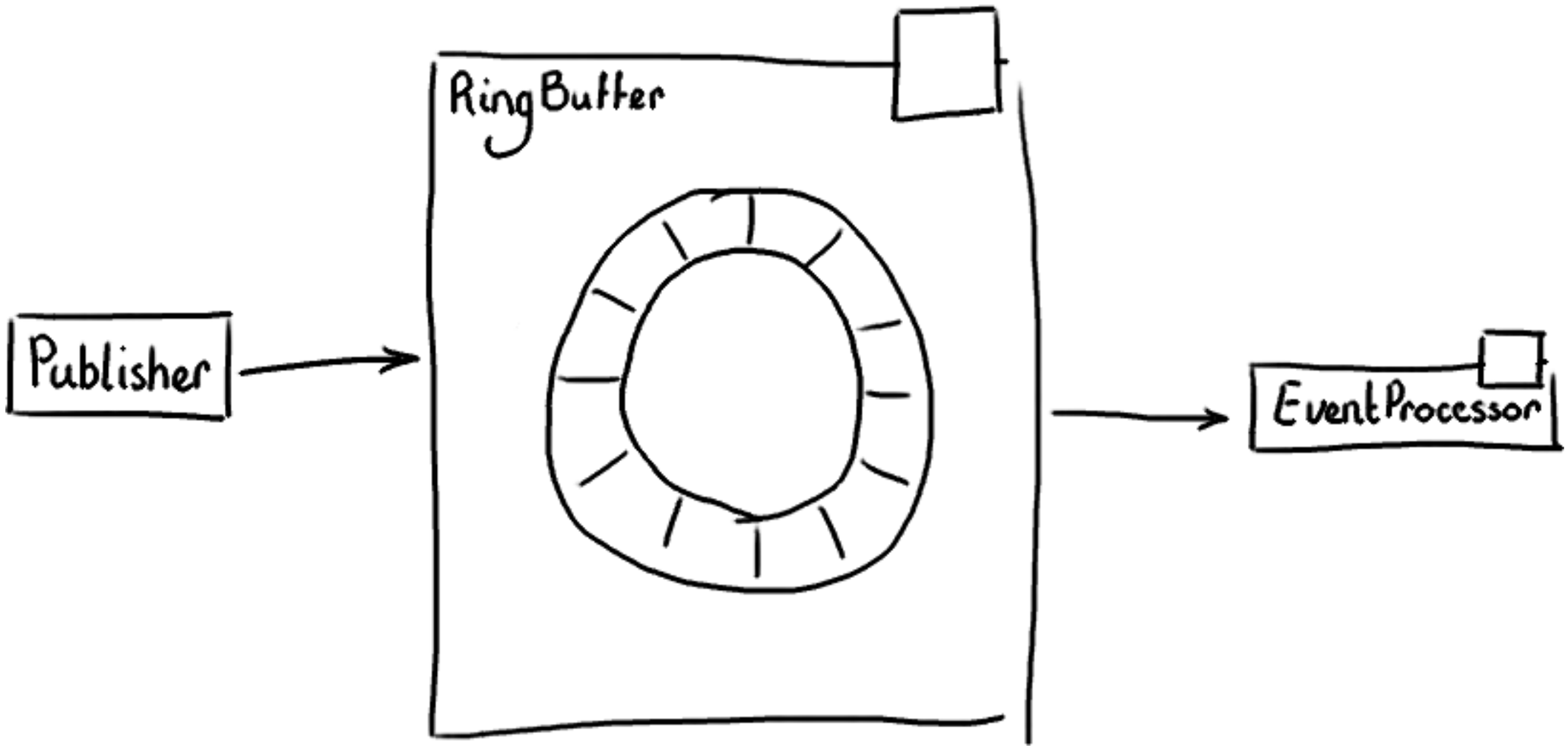
- Data structure and work flow with no contention.
- Very fast message passing.
- Allows you to go truly parallel.

Why Open Source it?

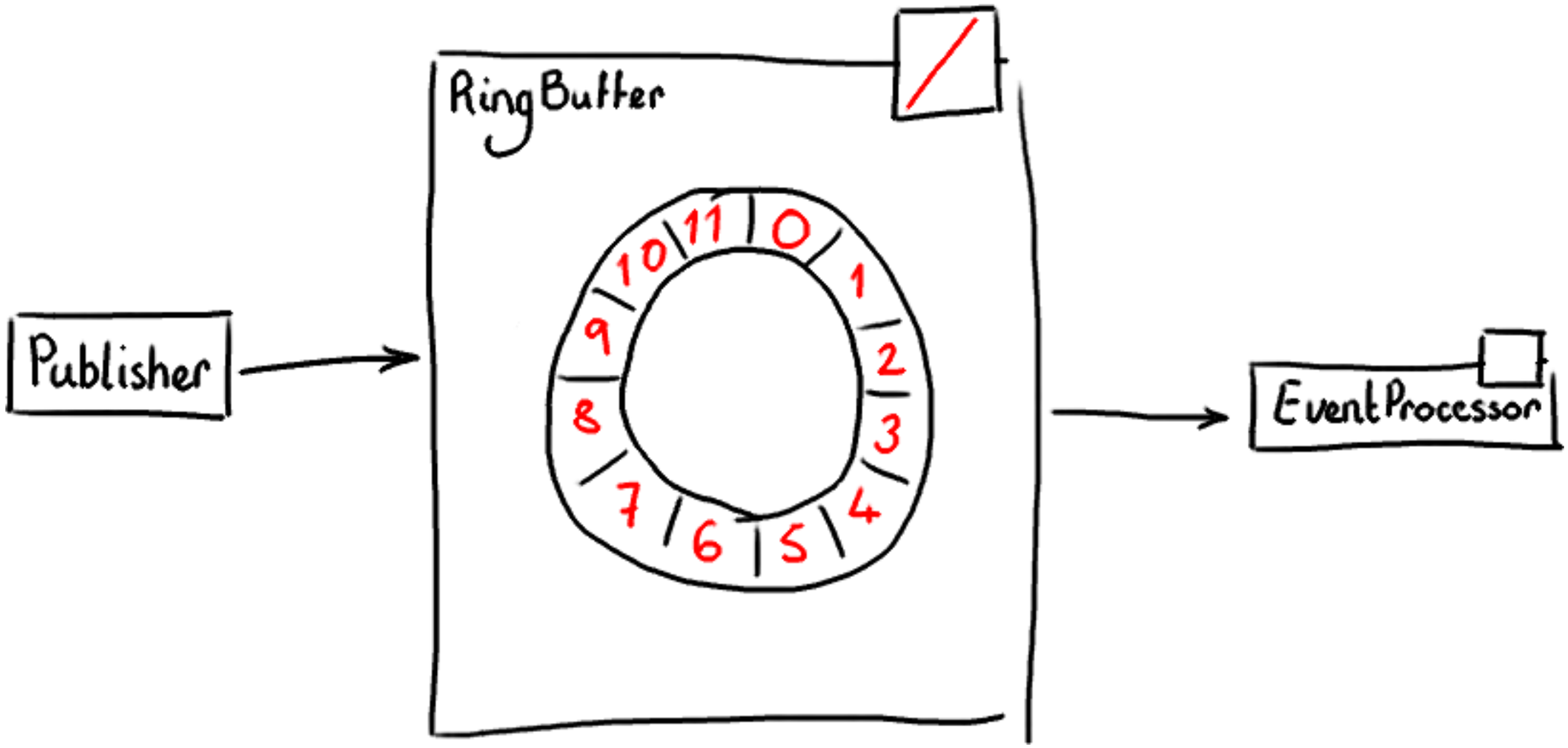
- We did something new that we wanted to share
- We were a young company and wanted to showcase our skills

So...?

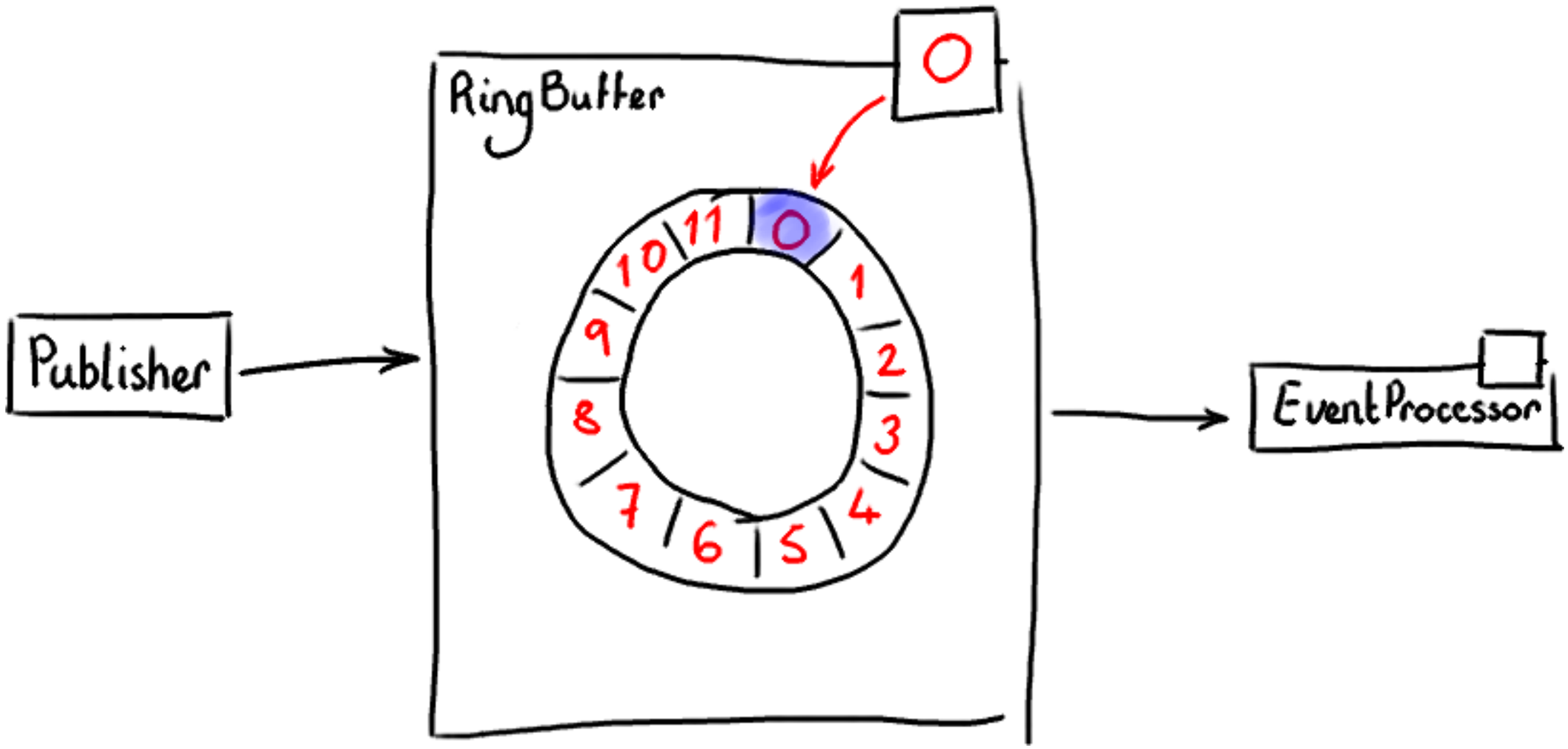
The Magic RingBuffer



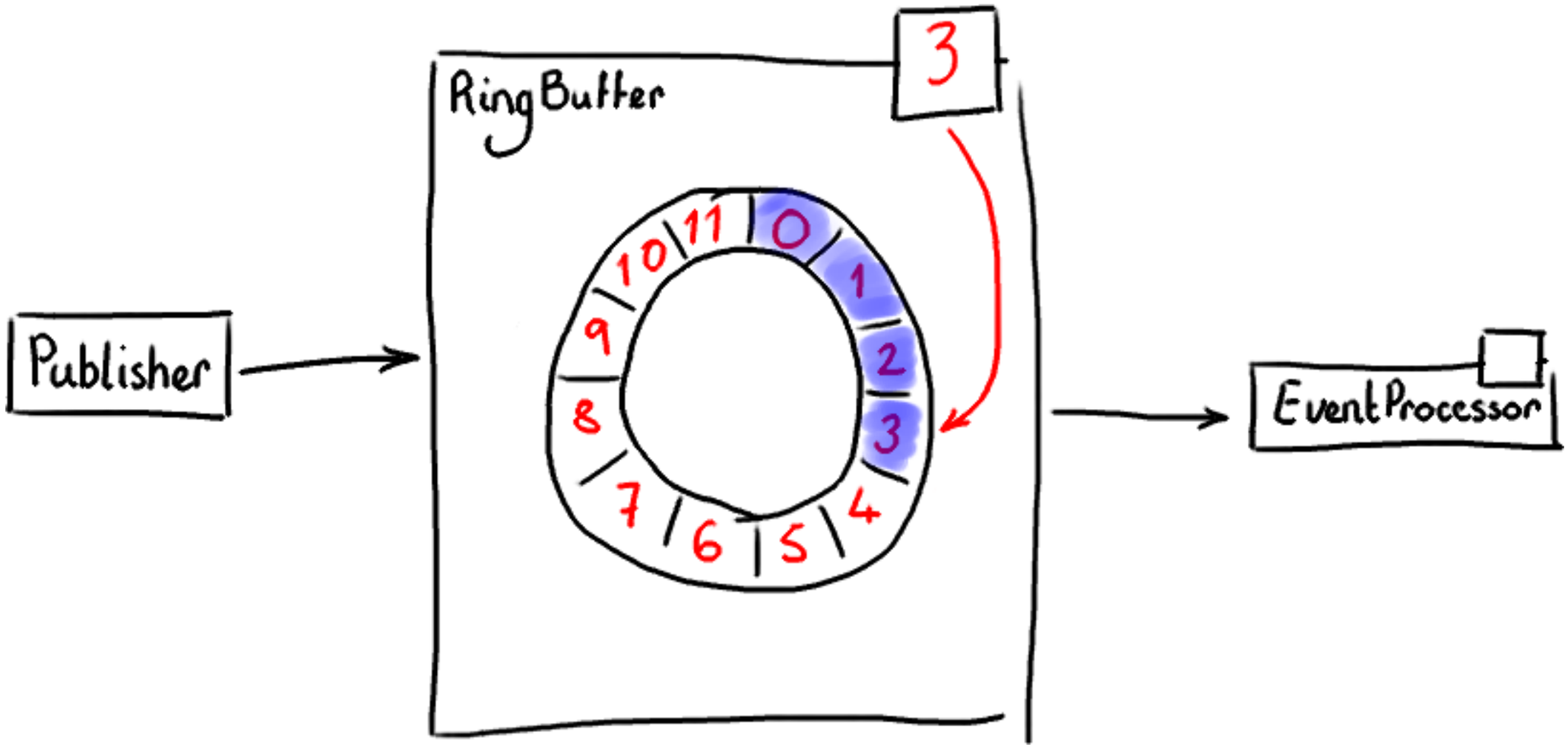
The Magic RingBuffer



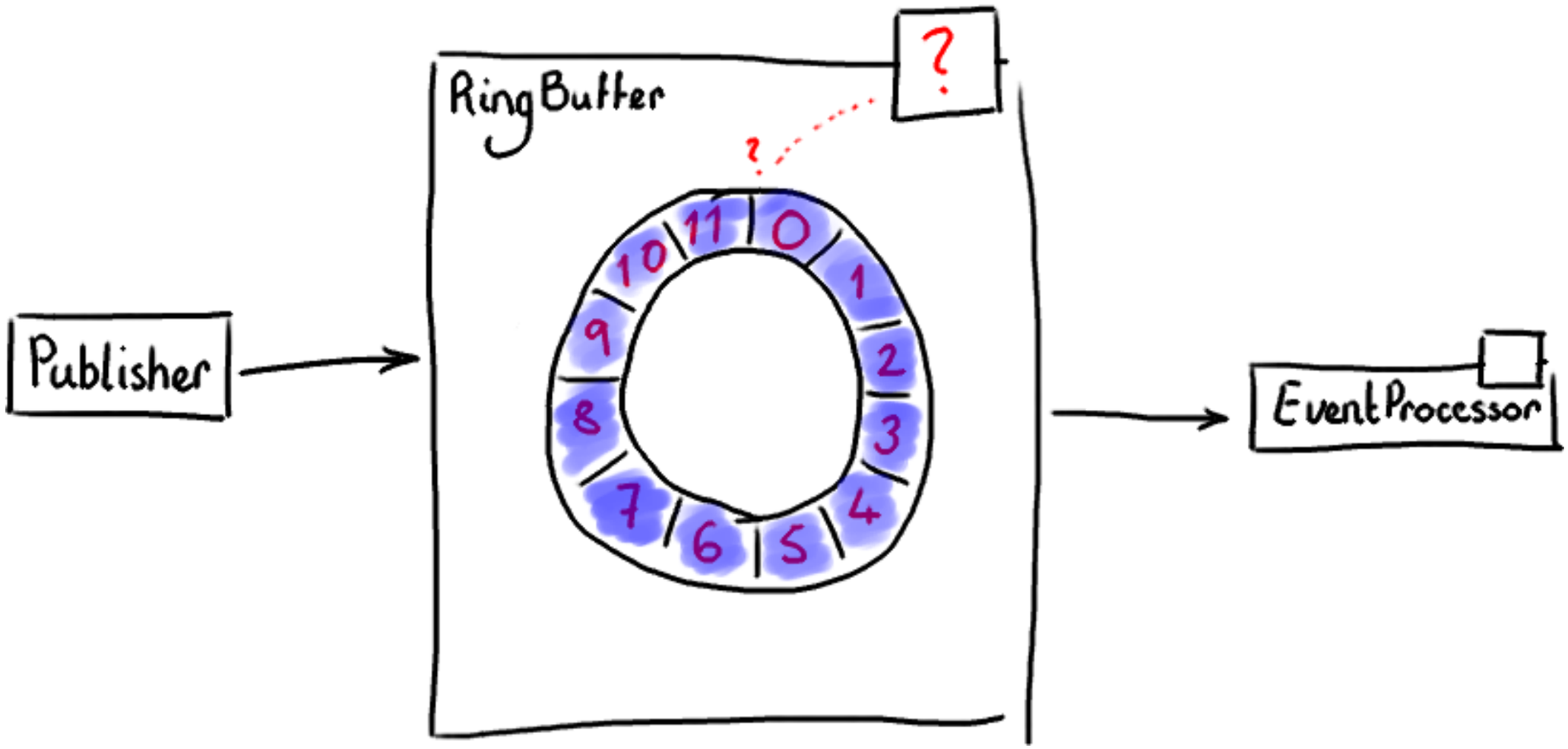
The Magic RingBuffer



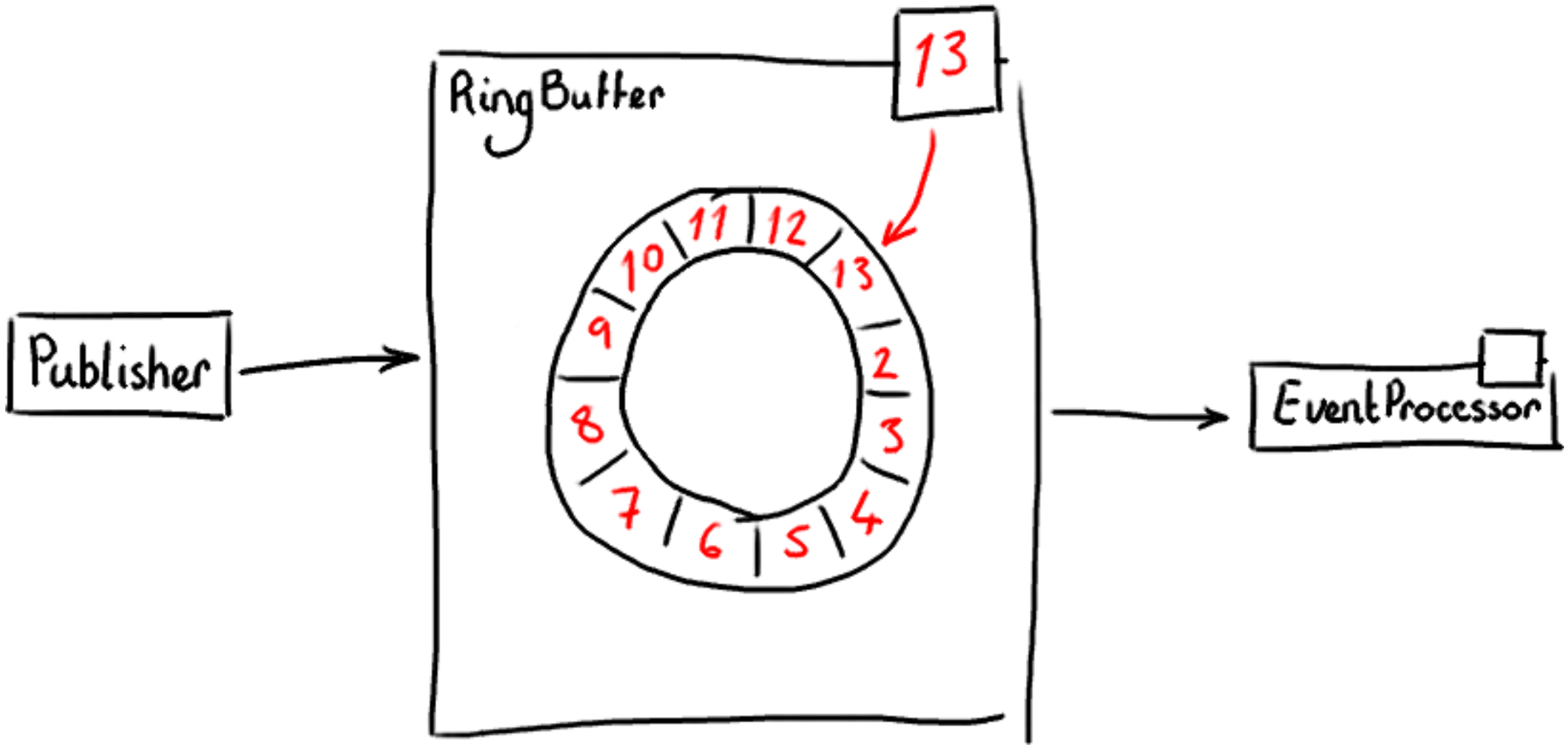
The Magic RingBuffer



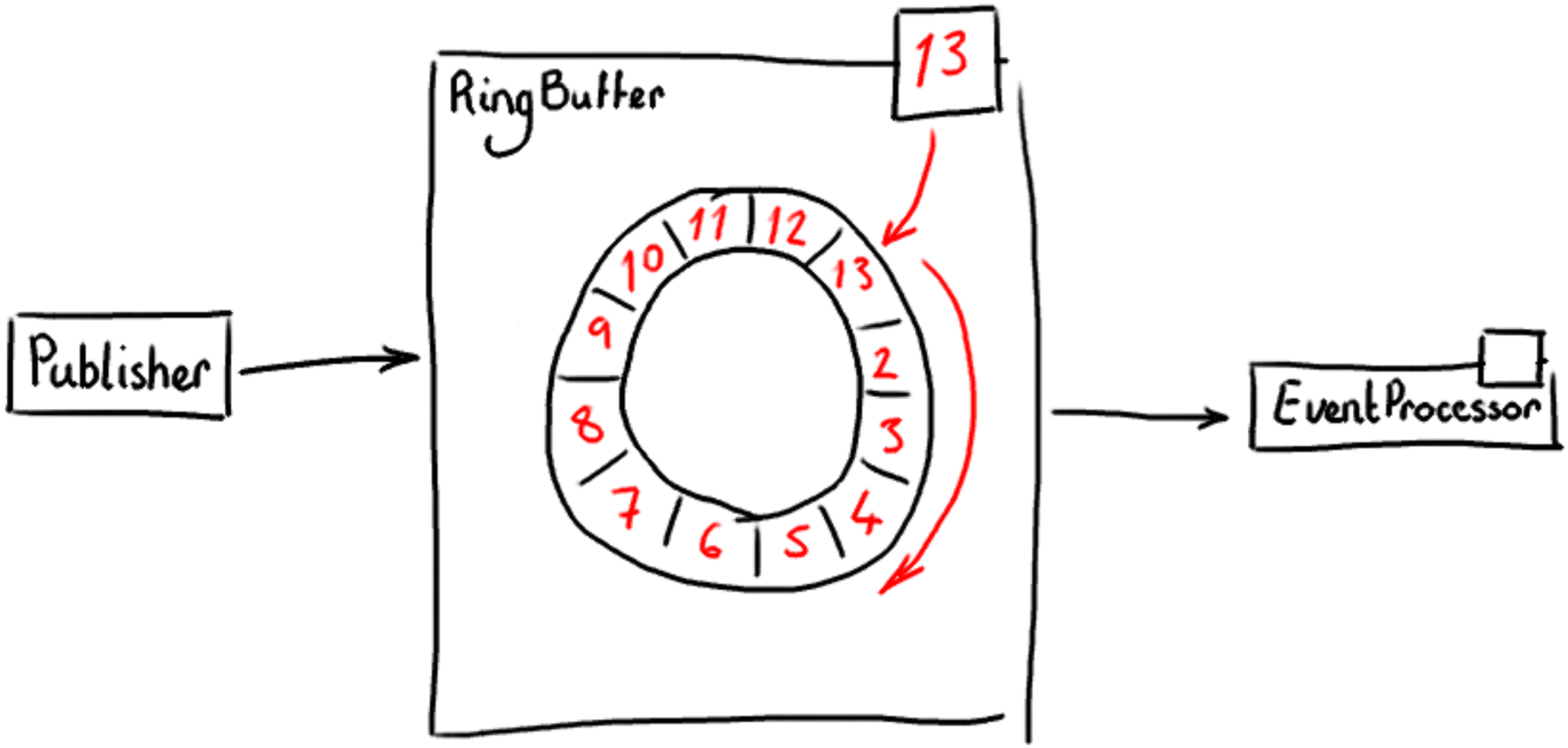
The Magic RingBuffer



The Magic RingBuffer



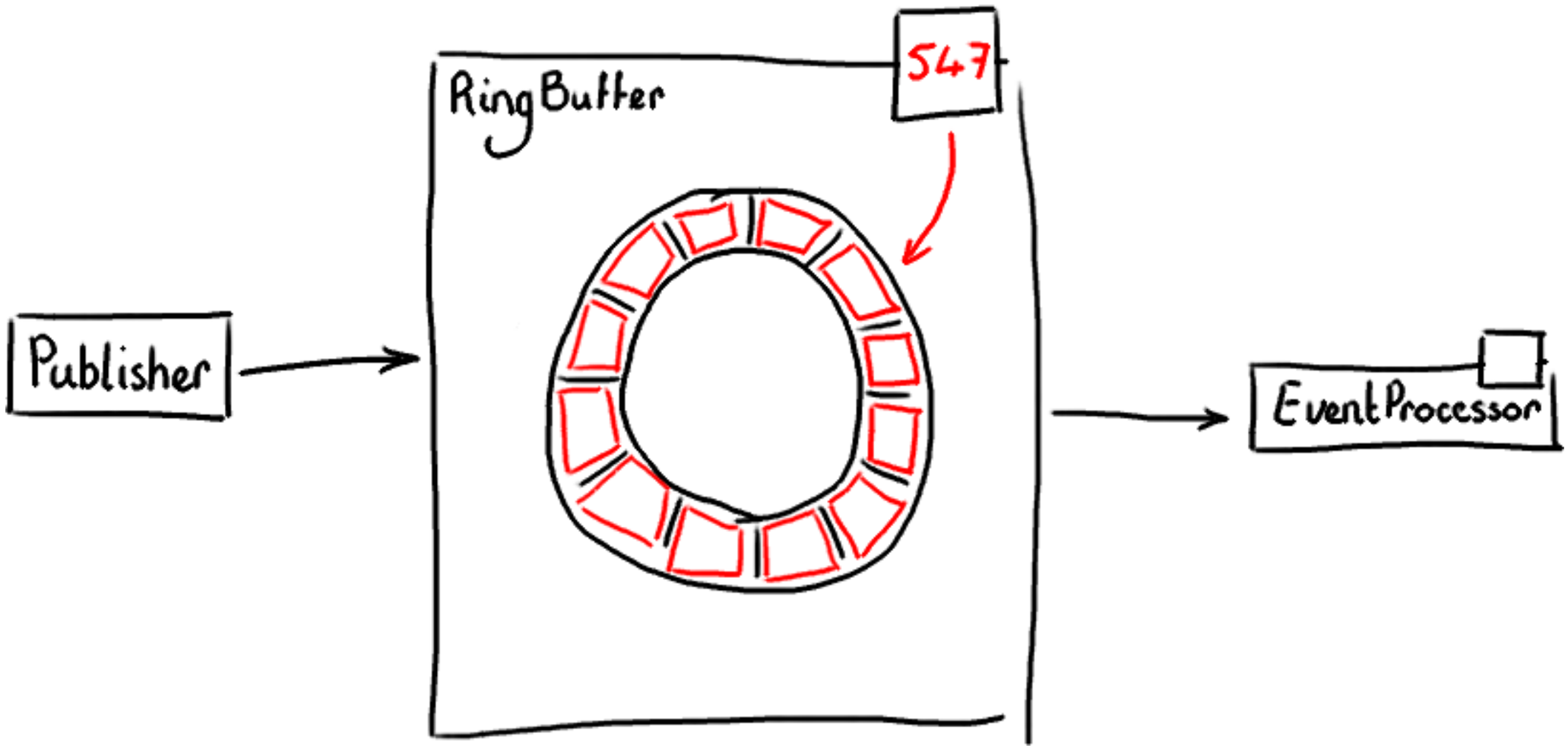
The Magic RingBuffer



Creating a RingBuffer

```
final RingBuffer<SimpleEvent> ringBuffer =
    new RingBuffer<SimpleEvent>(SimpleEvent.EVENT_FACTORY,
                                RING_BUFFER_SIZE);
```

The Events are Buckets



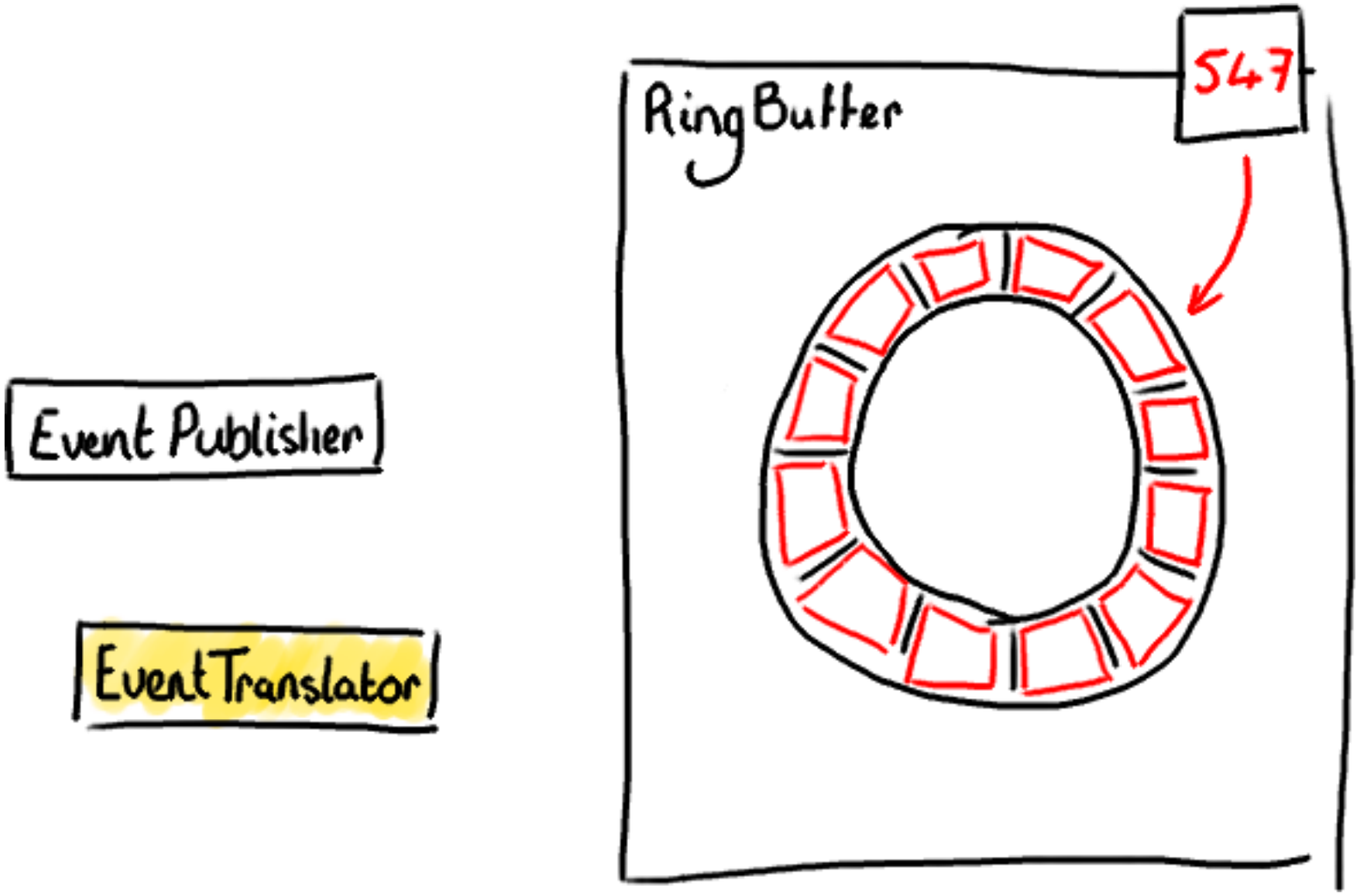
Great! I want one!

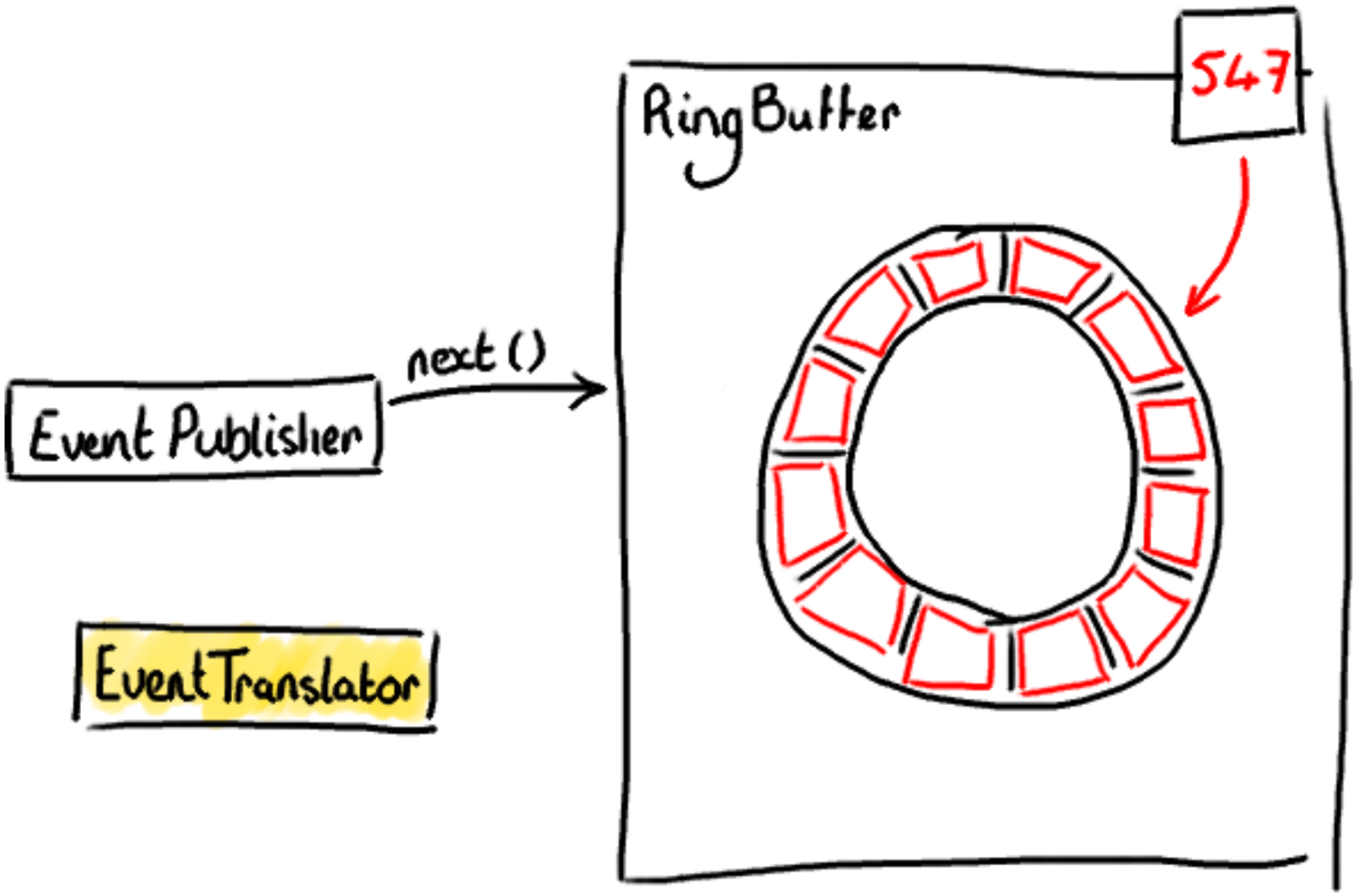
```
public class SimpleEvent {
    public static final EventFactory<SimpleEvent> EVENT_FACTORY =
        new SimpleEventFactory();

    private String value;


    private static class SimpleEventFactory implements EventFactory<SimpleEvent> {
        @Override
        public SimpleEvent newInstance() {
            return new SimpleEvent();
        }
    }
}
```

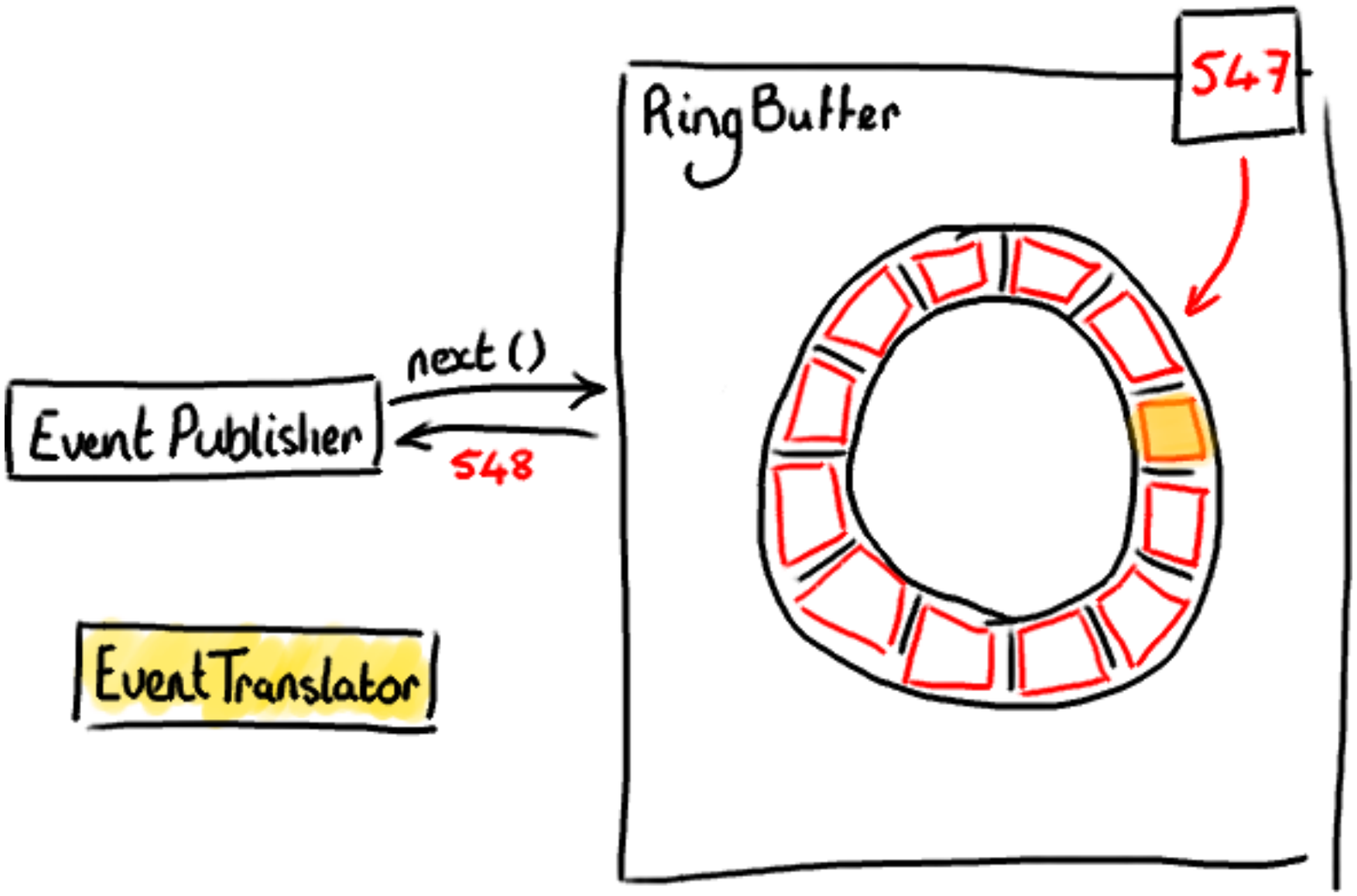
I've got a RingBuffer!

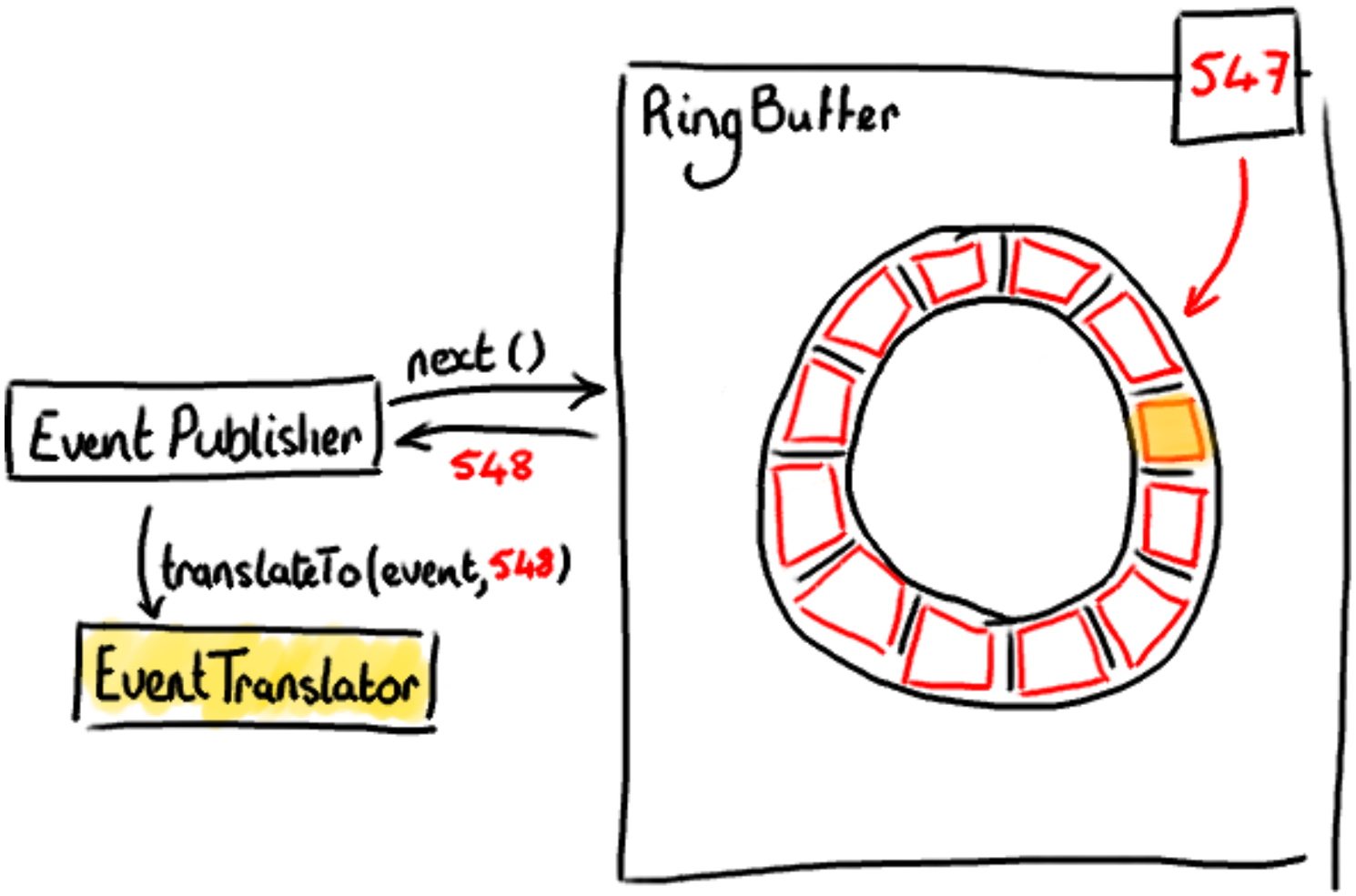
- Erm.... how do I poke things into it?

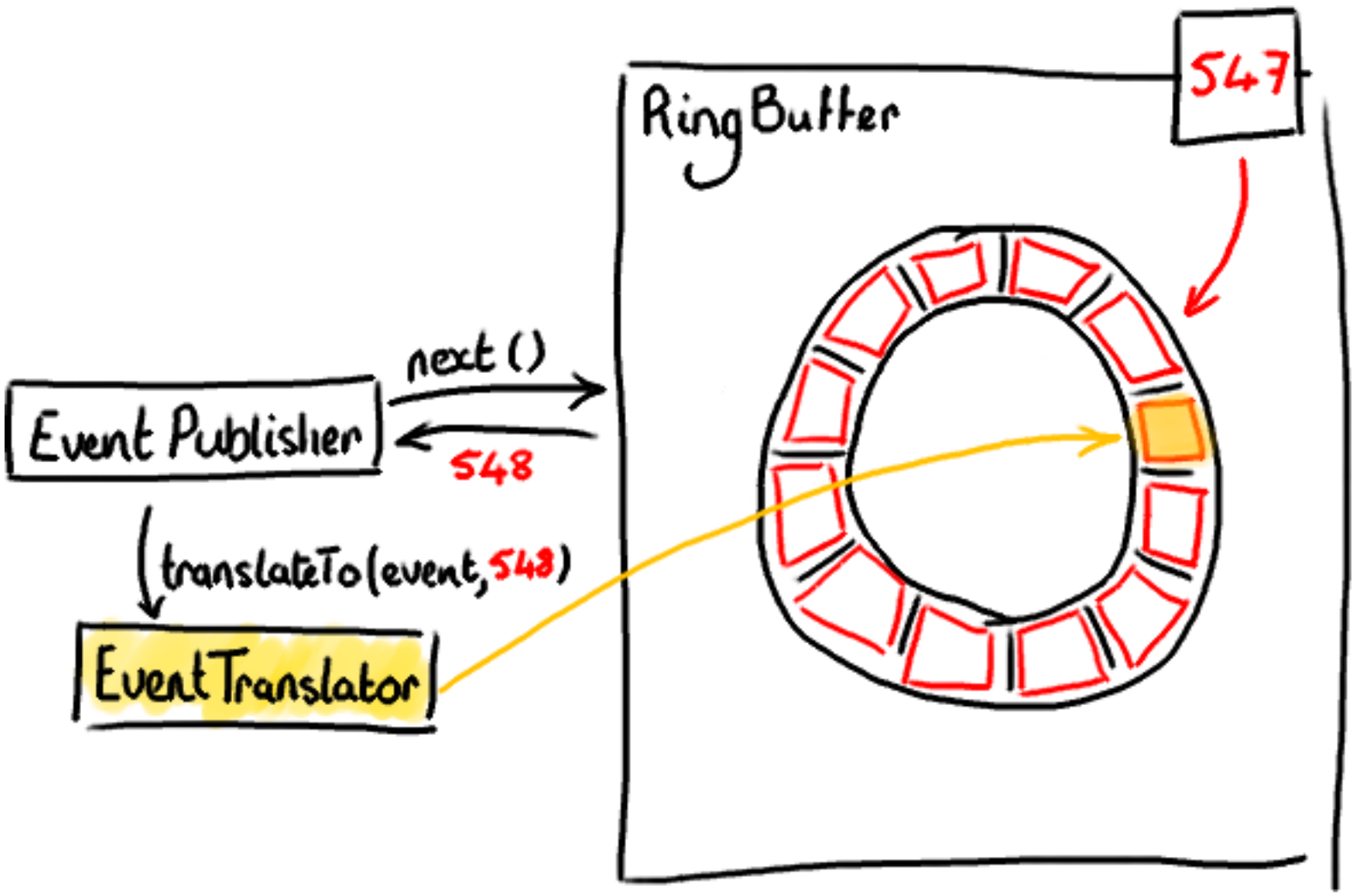
The Publisher

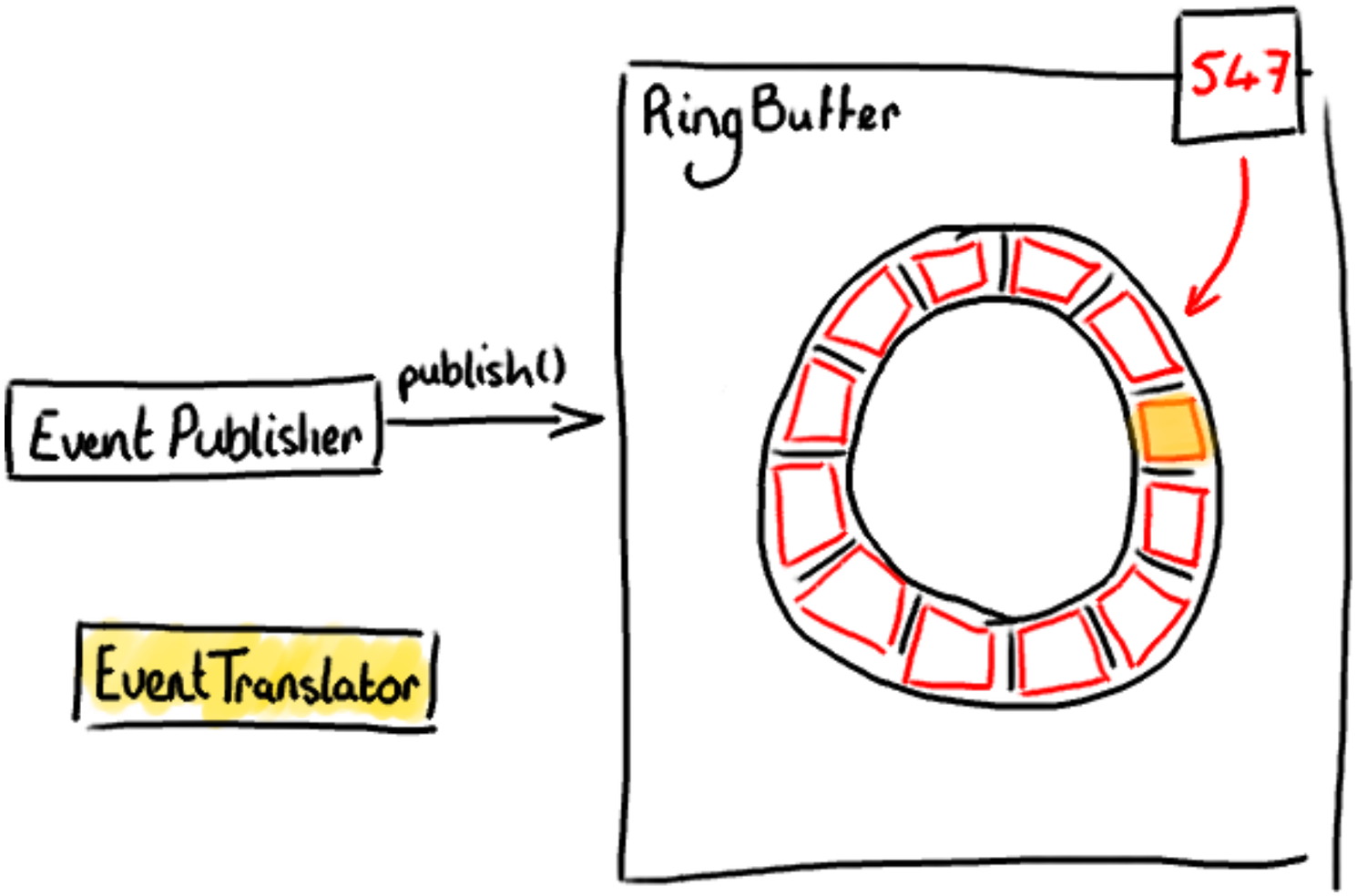


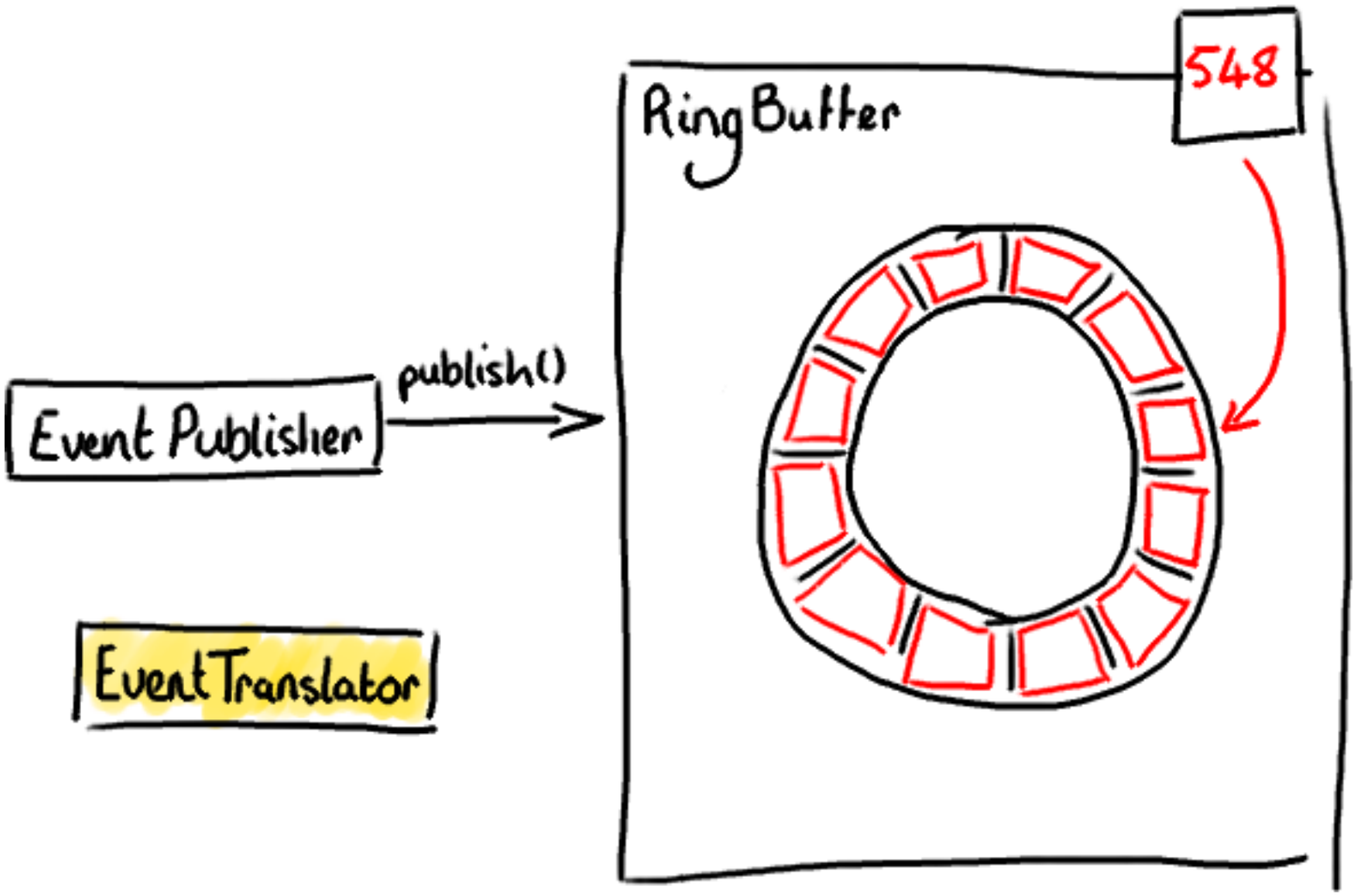

RingButter
547
next()
Event Publisher
EventTranslator


RingButter
547
next()
Event Publisher
548
EventTranslator


RingButter
547
next()
548
Event Publisher
translateTo(event, 548)
EventTranslator


RingButter
547
next()
548
Event Publisher
translateTo(event, 548)
EventTranslator


RingButter
547
publish()
Event Publisher
EventTranslator


RingButter
548
publish()
Event Publisher
EventTranslator

What do I do?

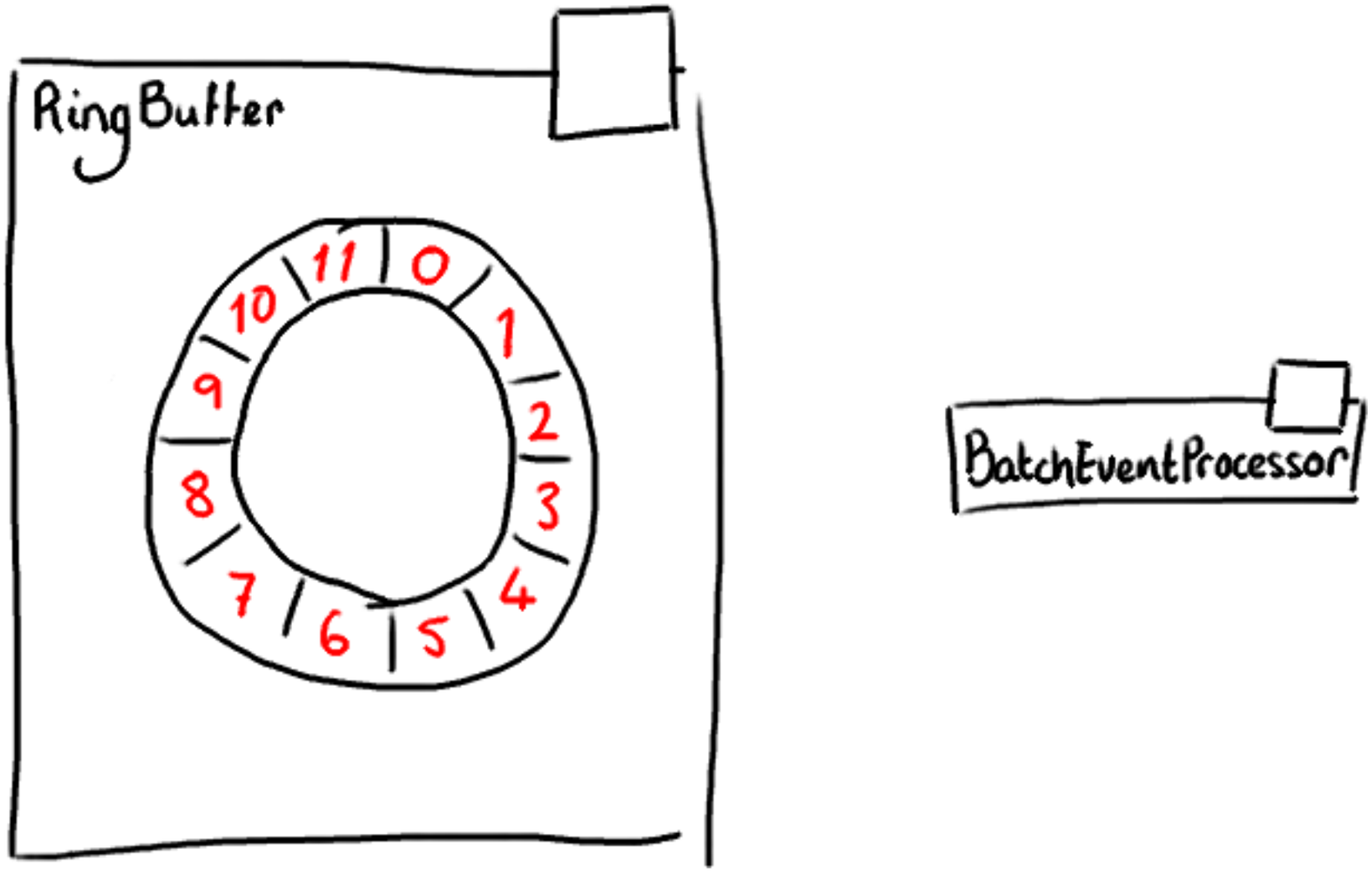
```
public class SimpleEventTranslator implements
    EventTranslator<SimpleEvent>
```

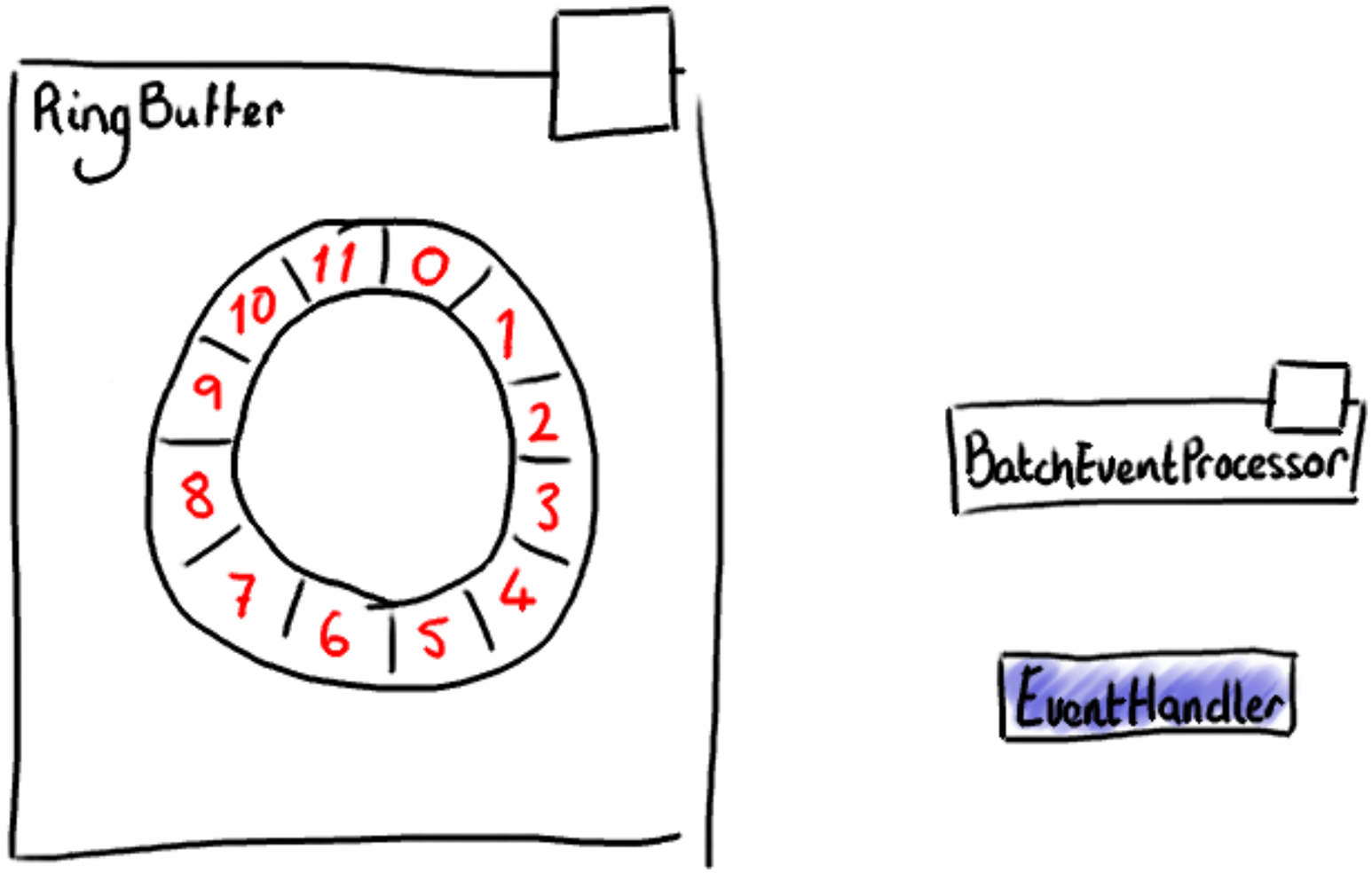
```
SimpleEventTranslator translator = new SimpleEventTranslator();

EventPublisher<SimpleEvent> publisher =
    new EventPublisher<SimpleEvent>(ringBuffer);

// poke your translator here
// ...and when you're done...
publisher.publishEvent(translator);
```

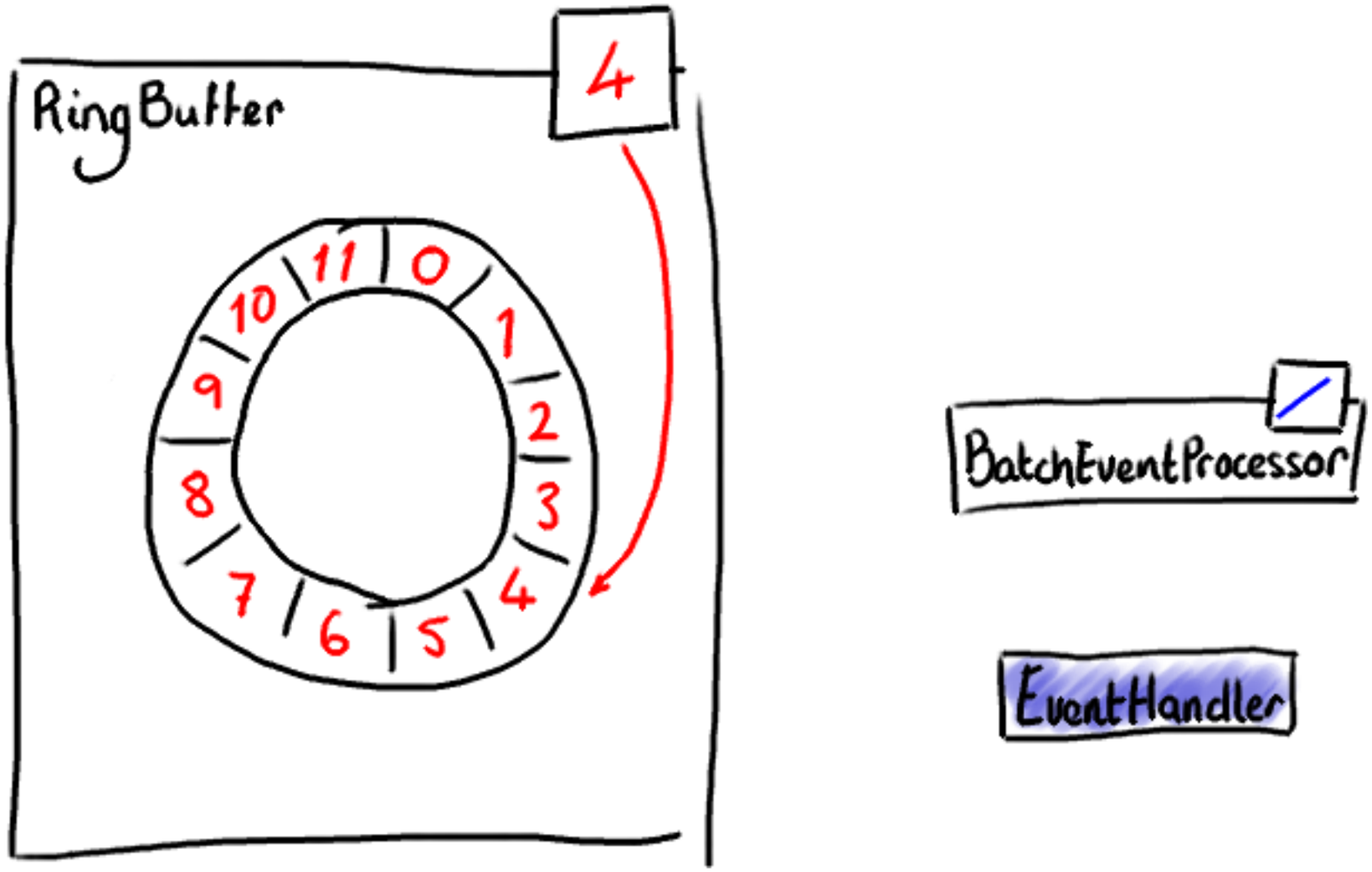
...so now I want to read

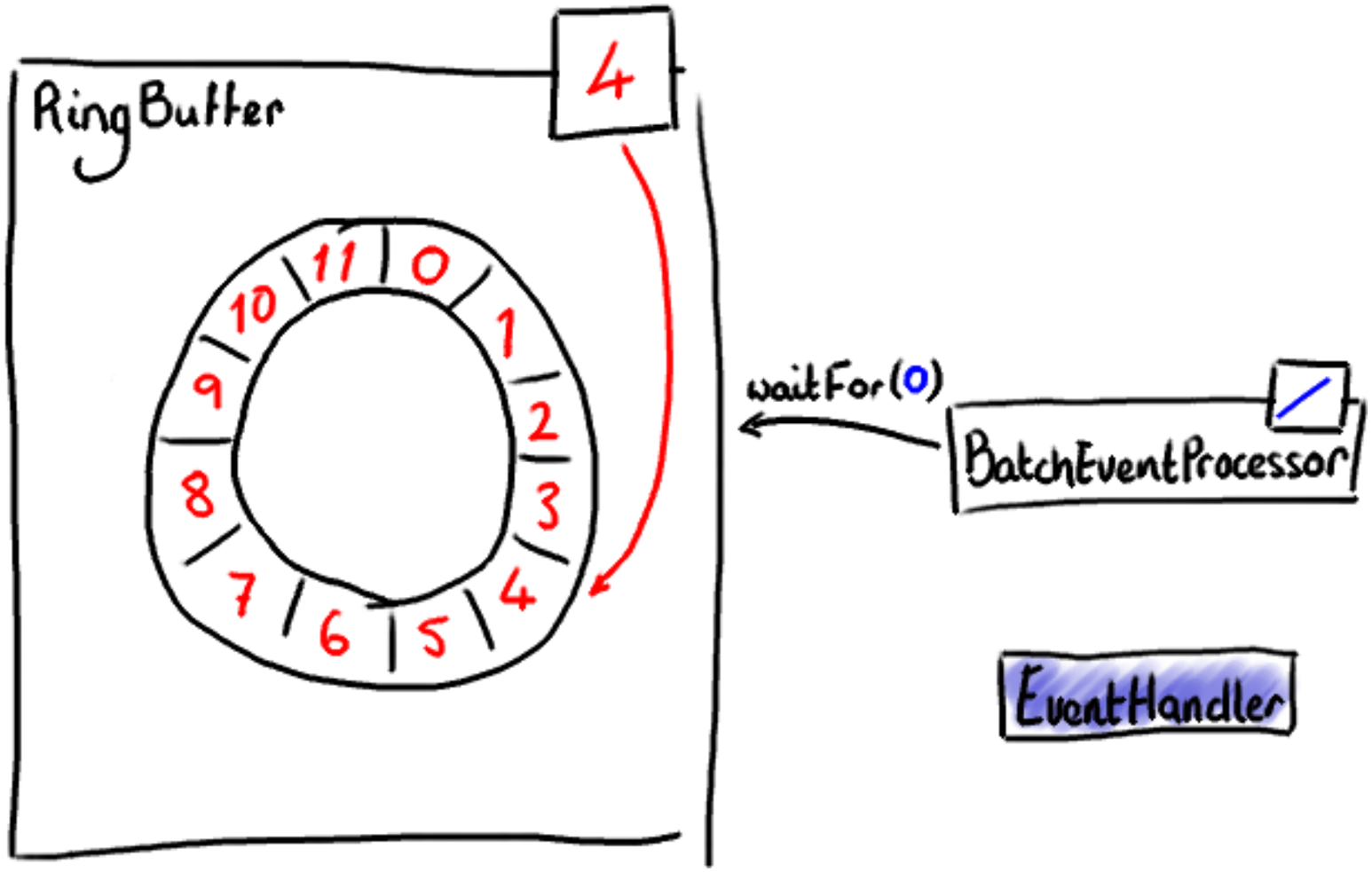
- The Disruptor provides nice batching behaviour for free

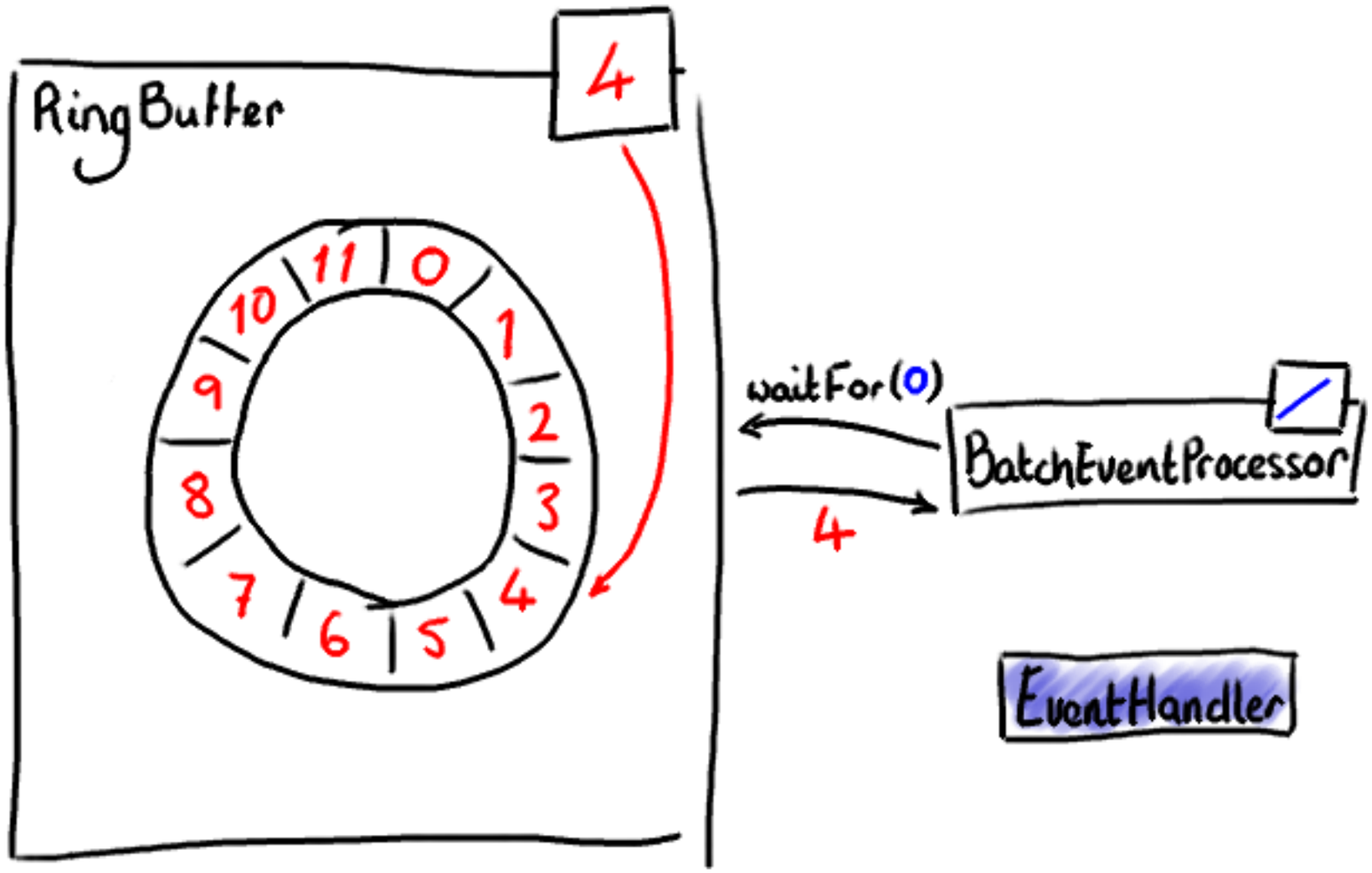
BatchEventProcessor

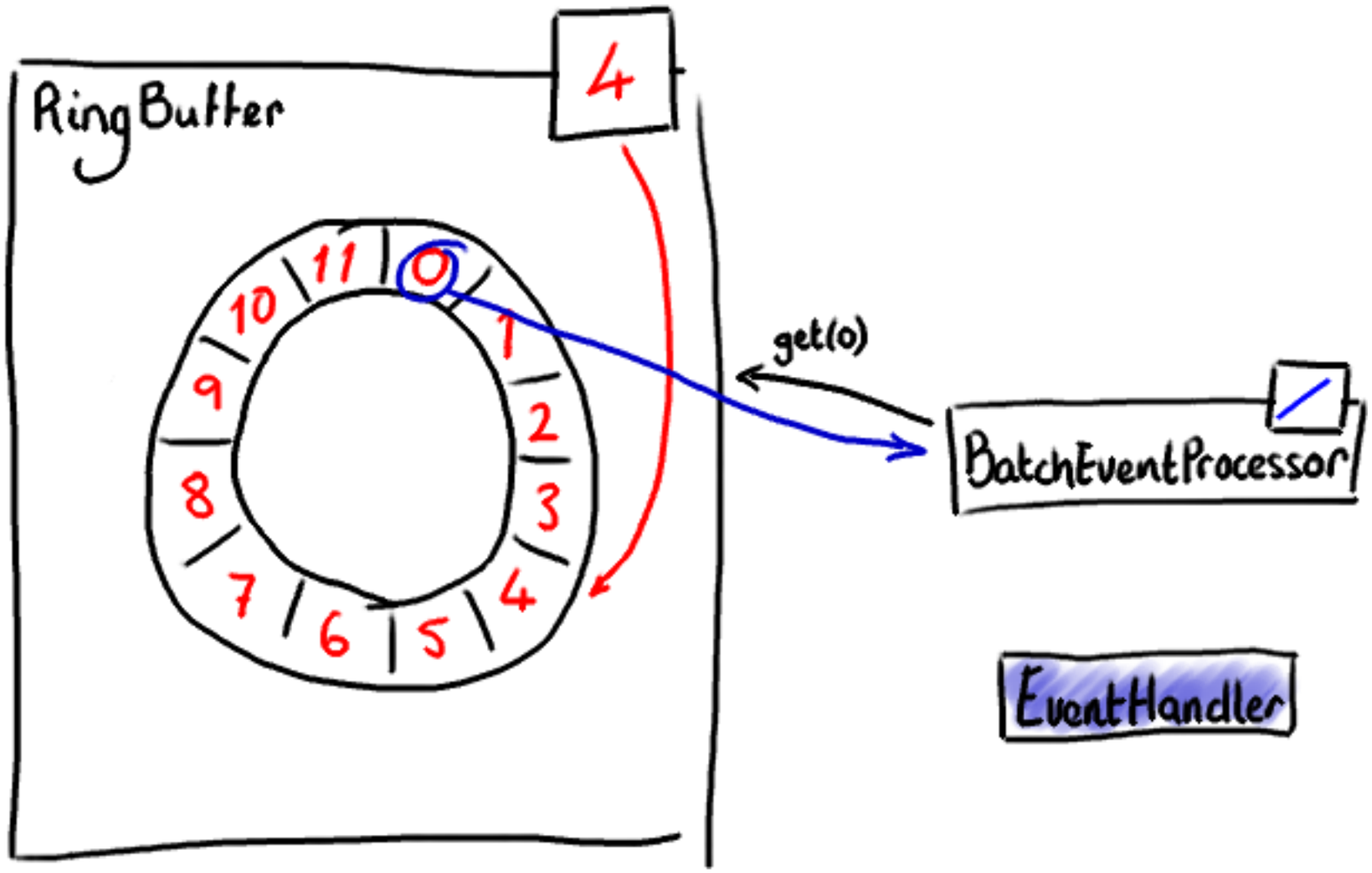


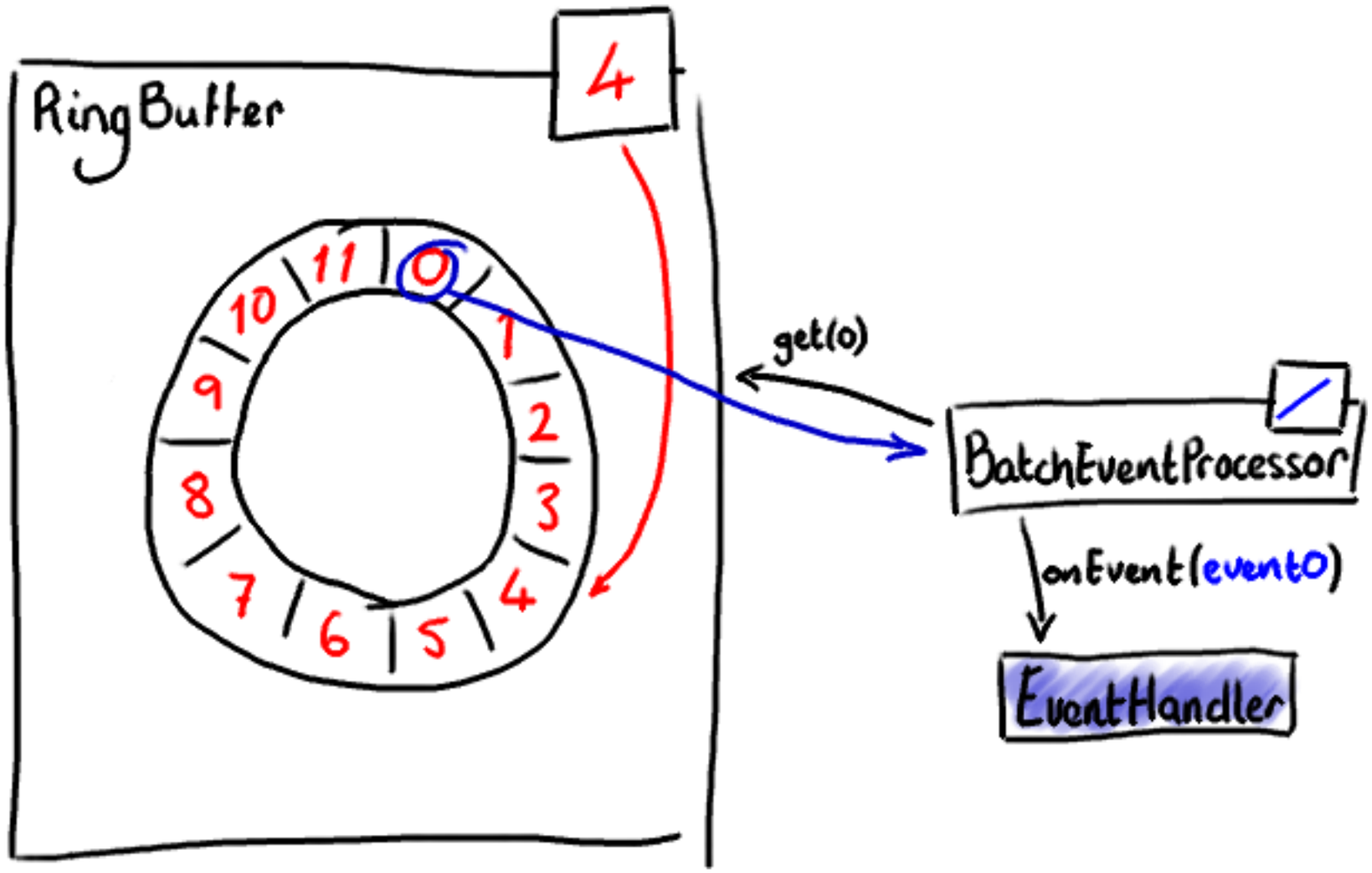

RingButter
11
0
1
2
3
4
5
6
7
8
9
10
BatchEventProcessor
EventHandler

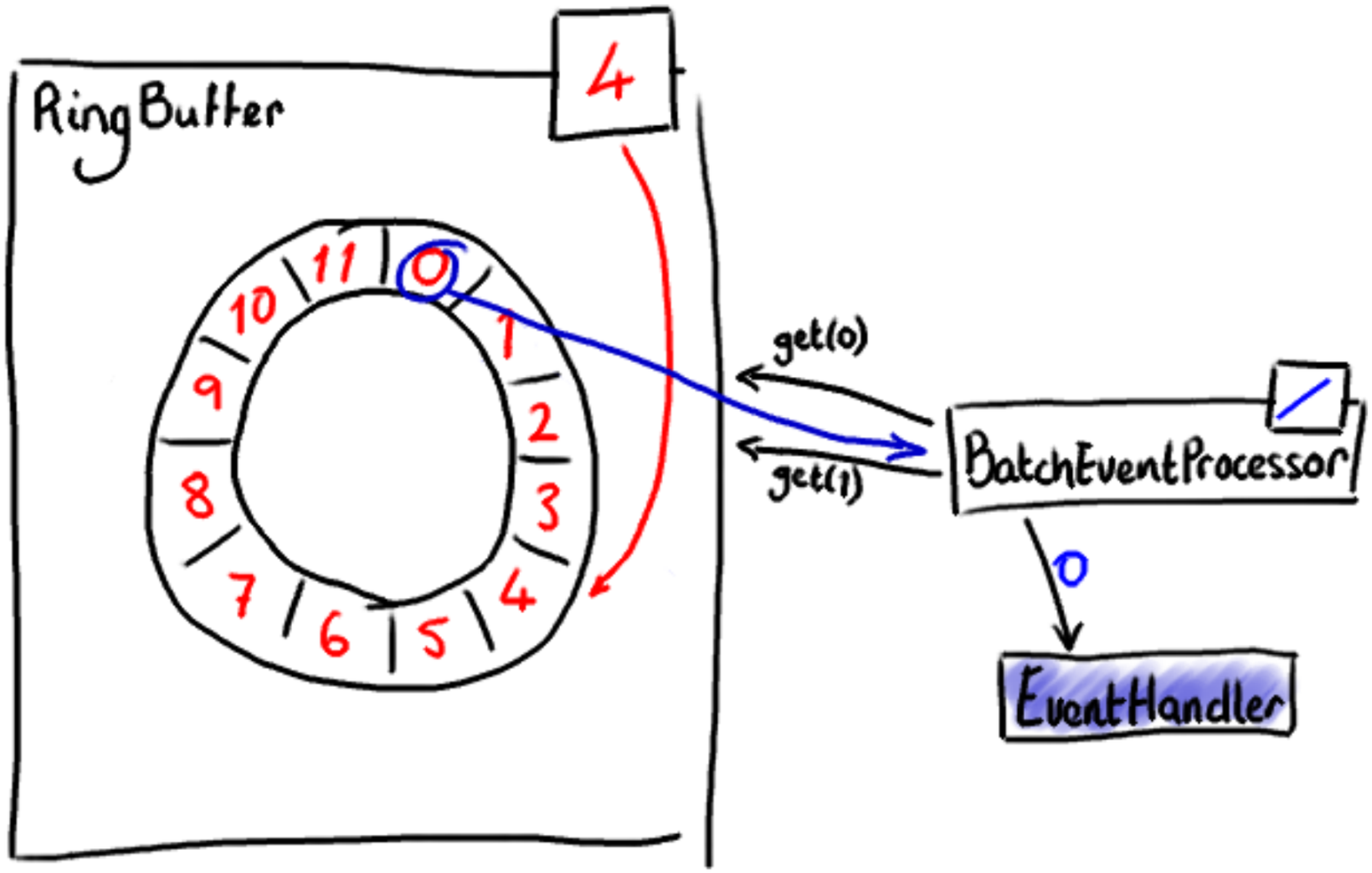

RingButter
4
11
0
1
2
3
4
5
6
7
8
9
10
BatchEventProcessor
EventHandler

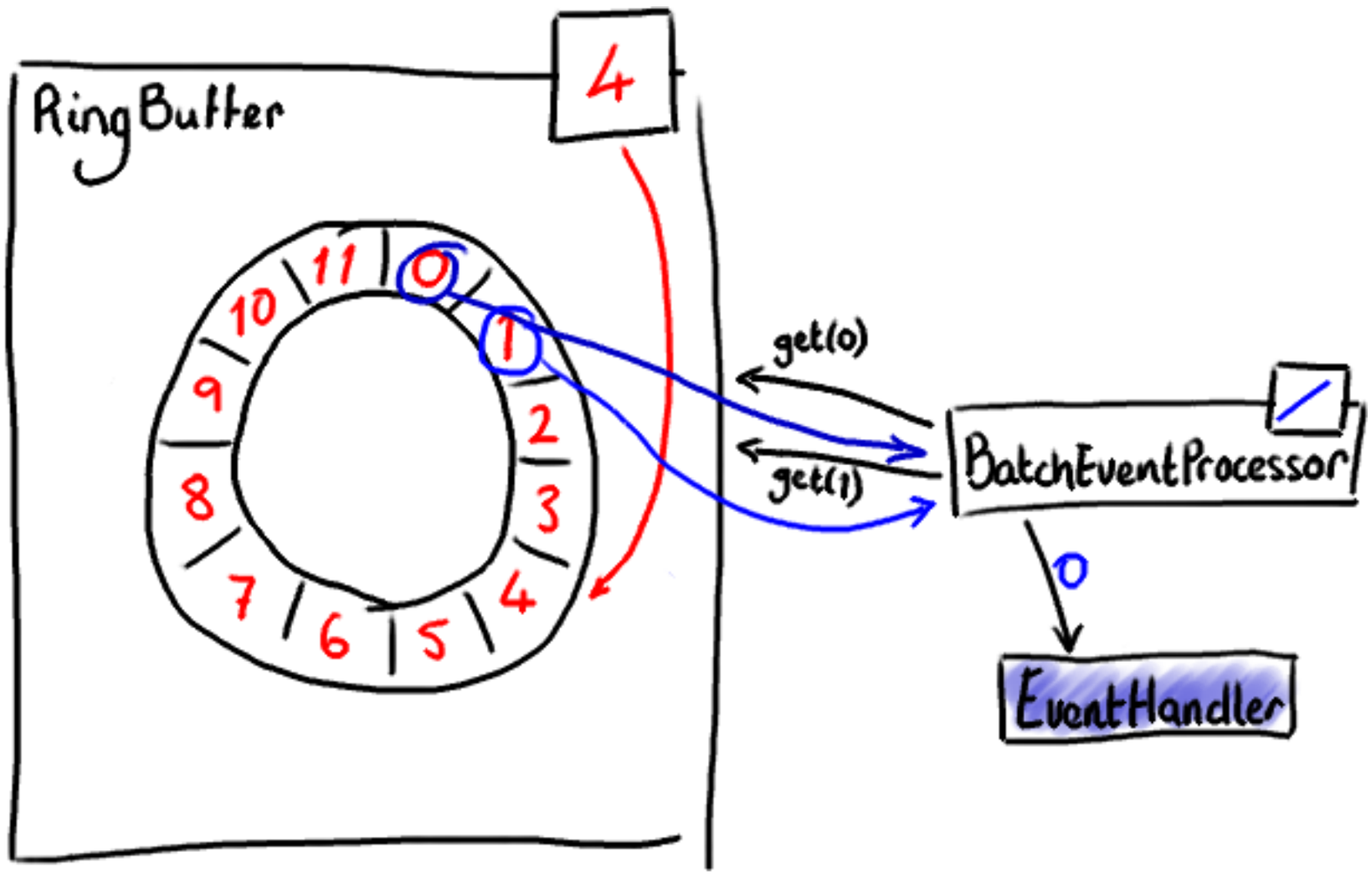

RingButter
4
11
0
1
2
3
4
5
6
7
8
9
10
waitFor(0)
BatchEventProcessor
EventHandler

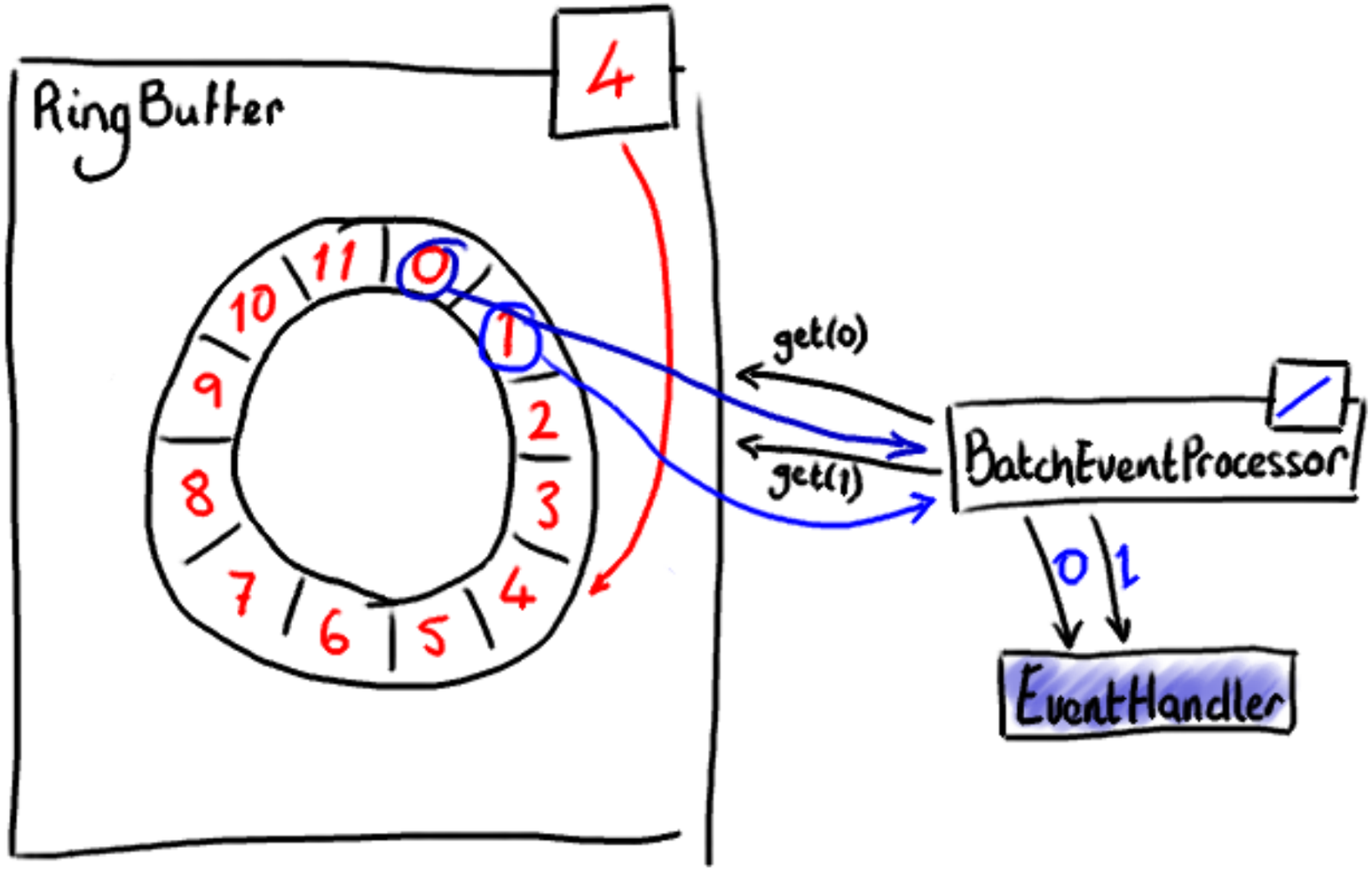

RingButter
4
11
0
1
2
3
4
5
6
7
8
9
10
waitFor(0)
BatchEventProcessor
4
EventHandler

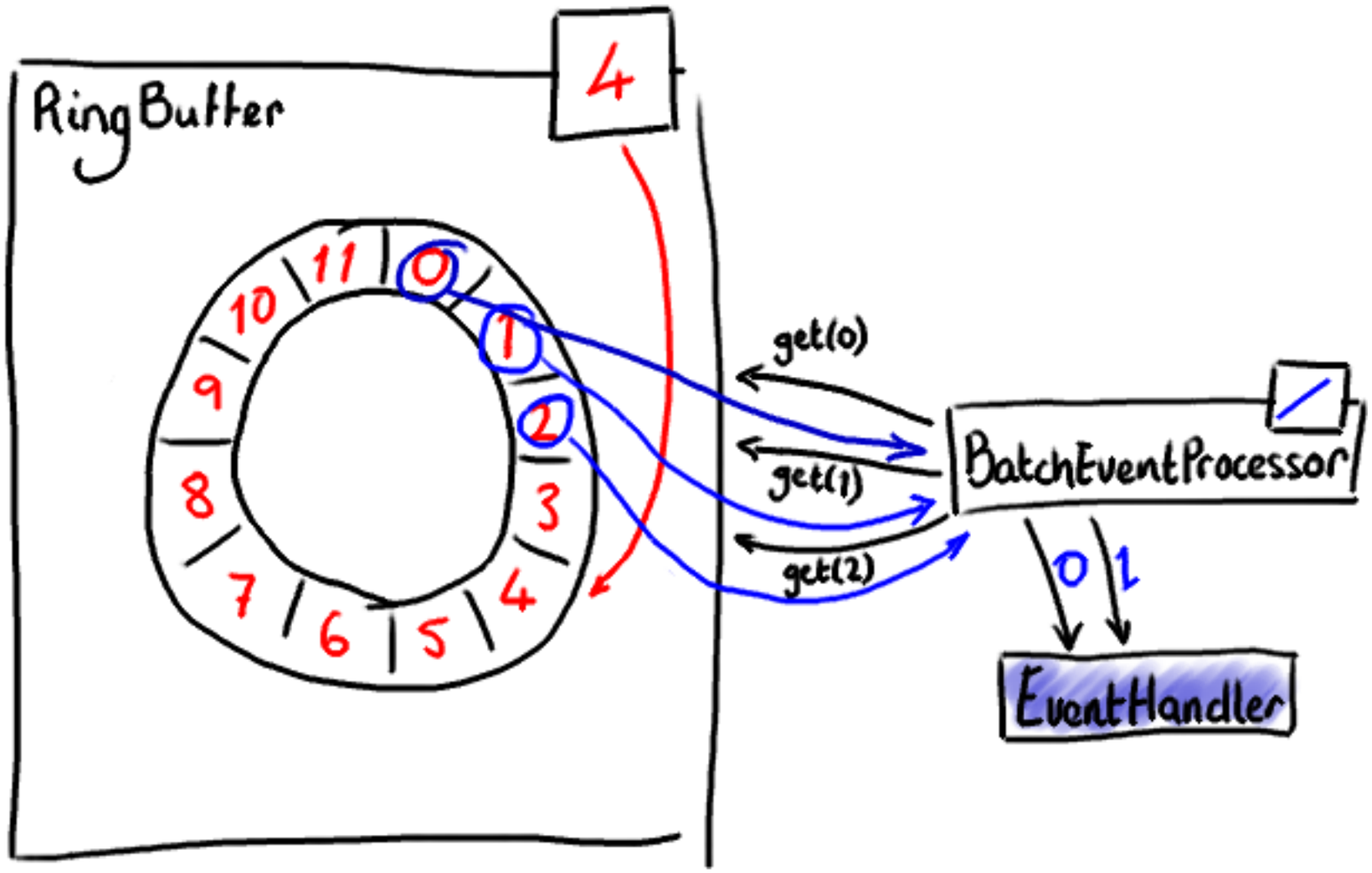

RingButter
4
11
0
1
2
3
4
5
6
7
8
9
10
get(0)
BatchEventProcessor
EventHandler

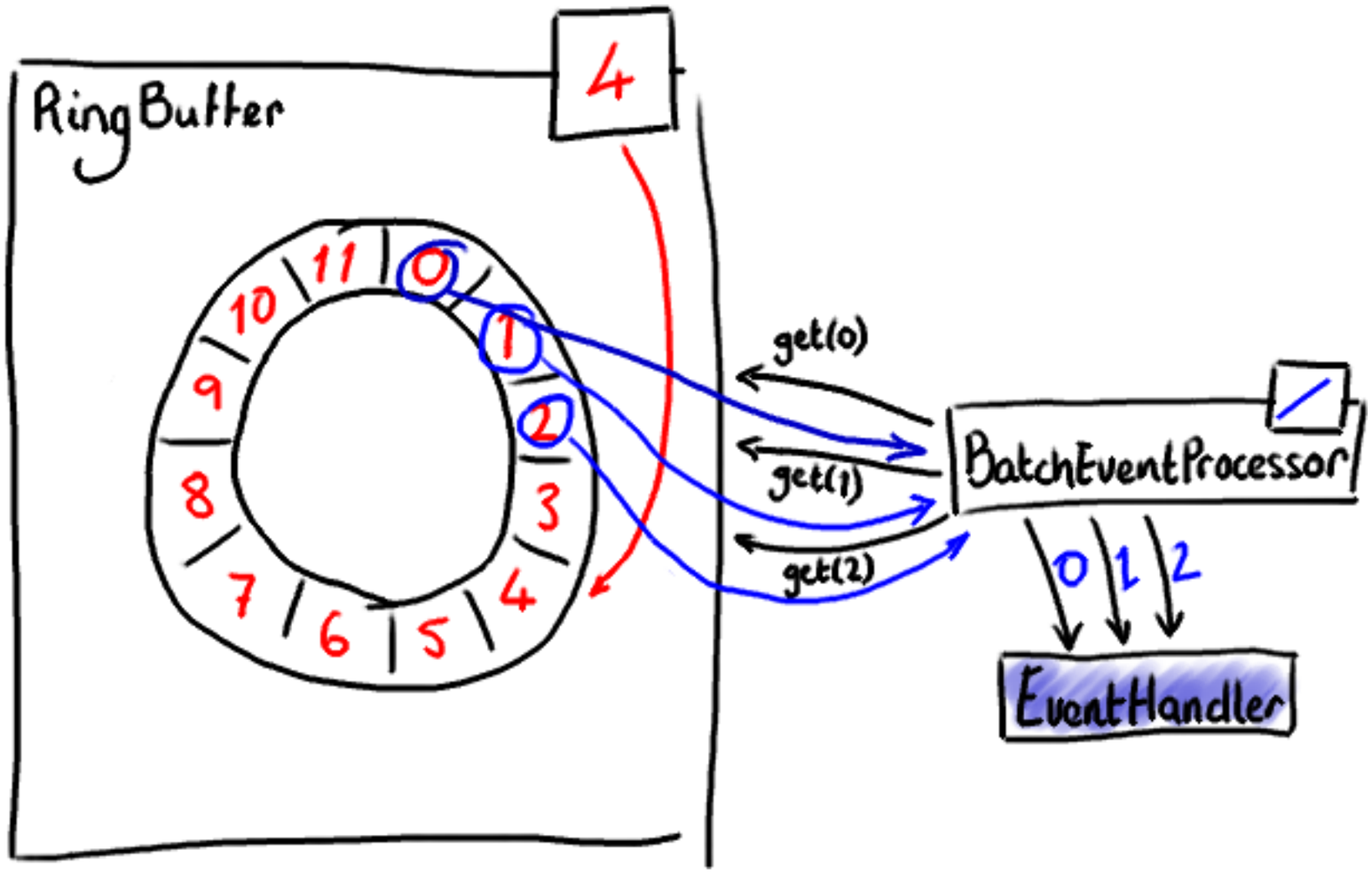

RingBuffer
4
11
0
1
2
3
4
5
6
7
8
9
10
get(0)
BatchEventProcessor
onEvent(event0)
EventHandler

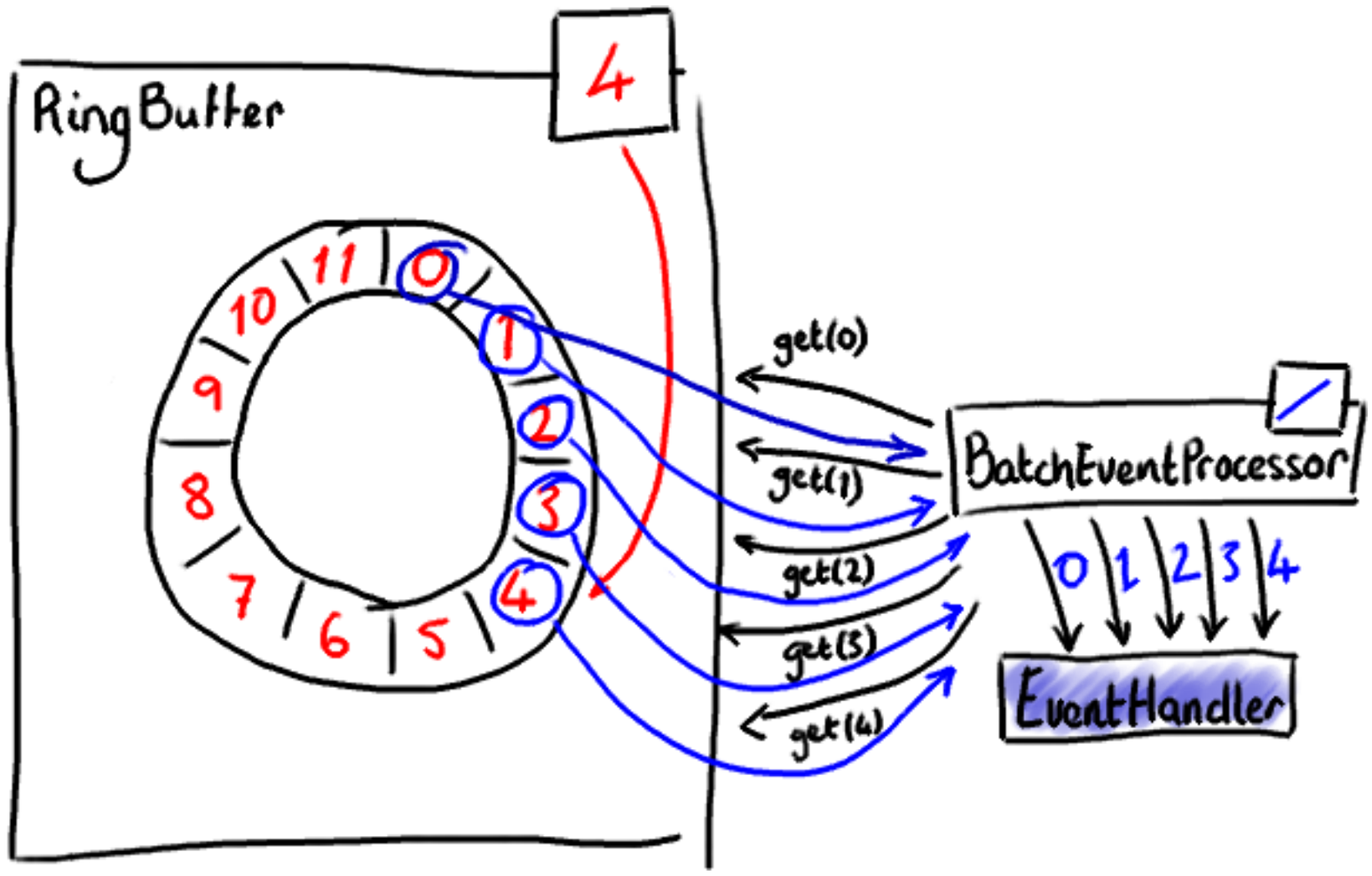

RingBuffer
4
11
0
1
2
3
4
5
6
7
8
9
10
get(0)
get(1)
BatchEventProcessor
0
EventHandler

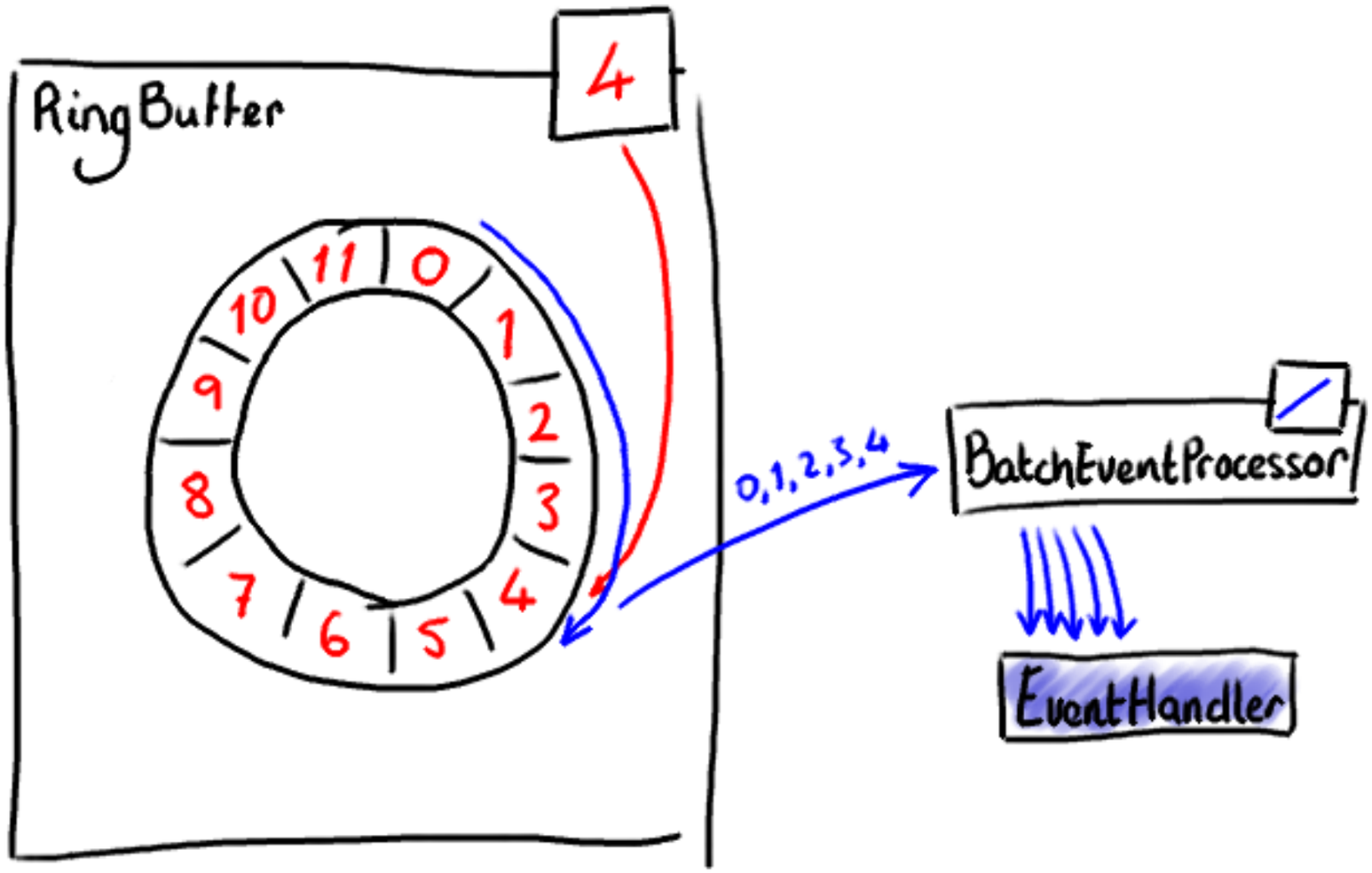

RingBuffer
4
11
0
1
2
3
4
5
6
7
8
9
10
get(0)
get(1)
BatchEventProcessor
0
EventHandler

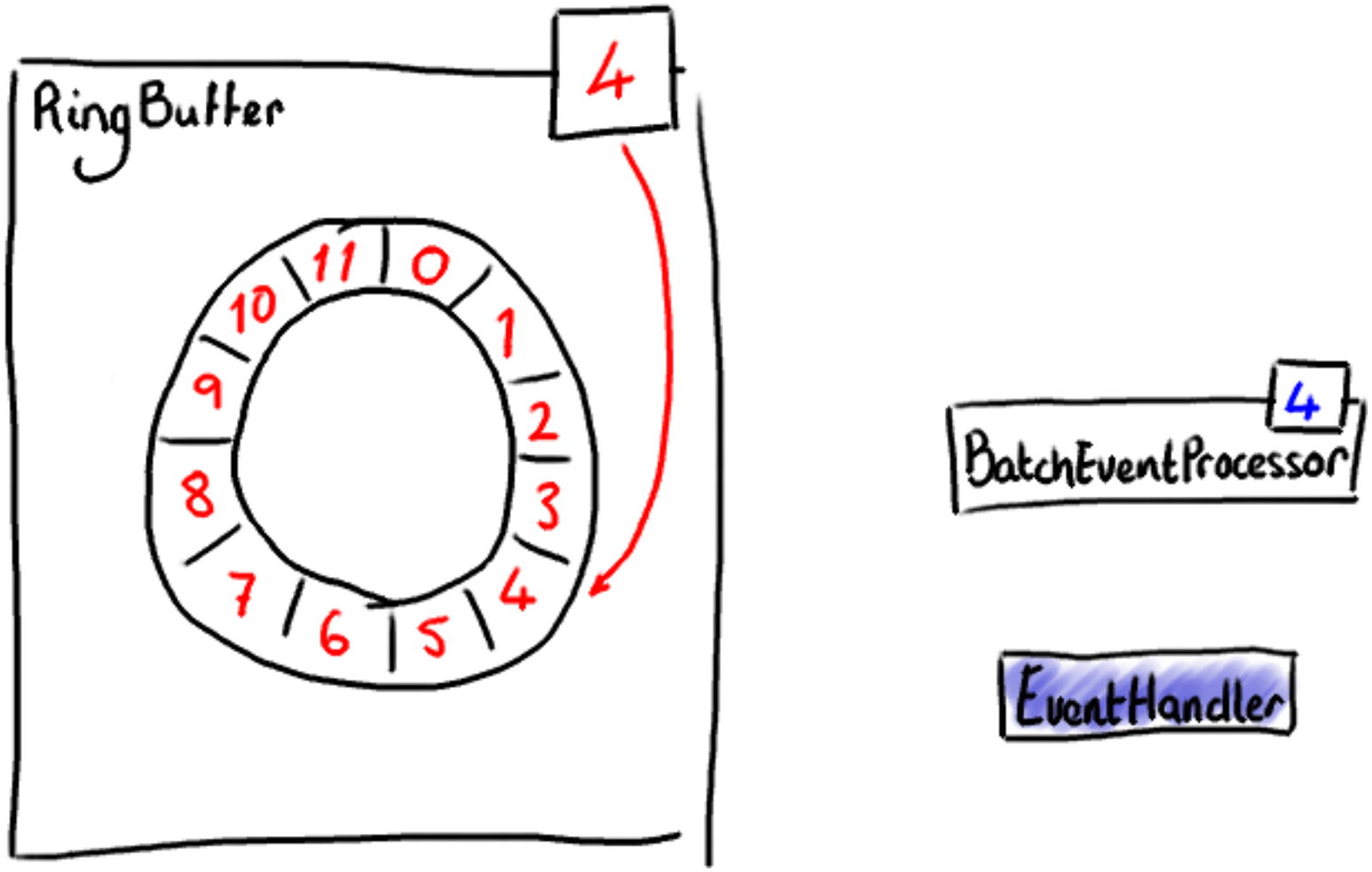

RingButter
4
11
0
10
1
9
2
8
3
7
6
5
4
get(0)
get(1)
BatchEventProcessor
0
1
EventHandler


RingBuffer
4
11
0
1
2
3
4
5
6
7
8
9
10
get(0)
get(1)
get(2)
BatchEventProcessor
0
1
EventHandler


RingBuffer
4
11
0
1
2
3
4
5
6
7
8
9
10
get(0)
get(1)
get(2)
BatchEventProcessor
0
1
2
EventHandler


RingButter
4
11
0
10
1
9
2
8
3
7
6
5
4
get(0)
get(1)
get(2)
get(3)
get(4)
BatchEventProcessor
0
1
2
3
4
EventHandler


RingBuffer
4
11
0
1
2
3
4
5
6
7
8
9
10
0,1,2,3,4
BatchEventProcessor
EventHandler

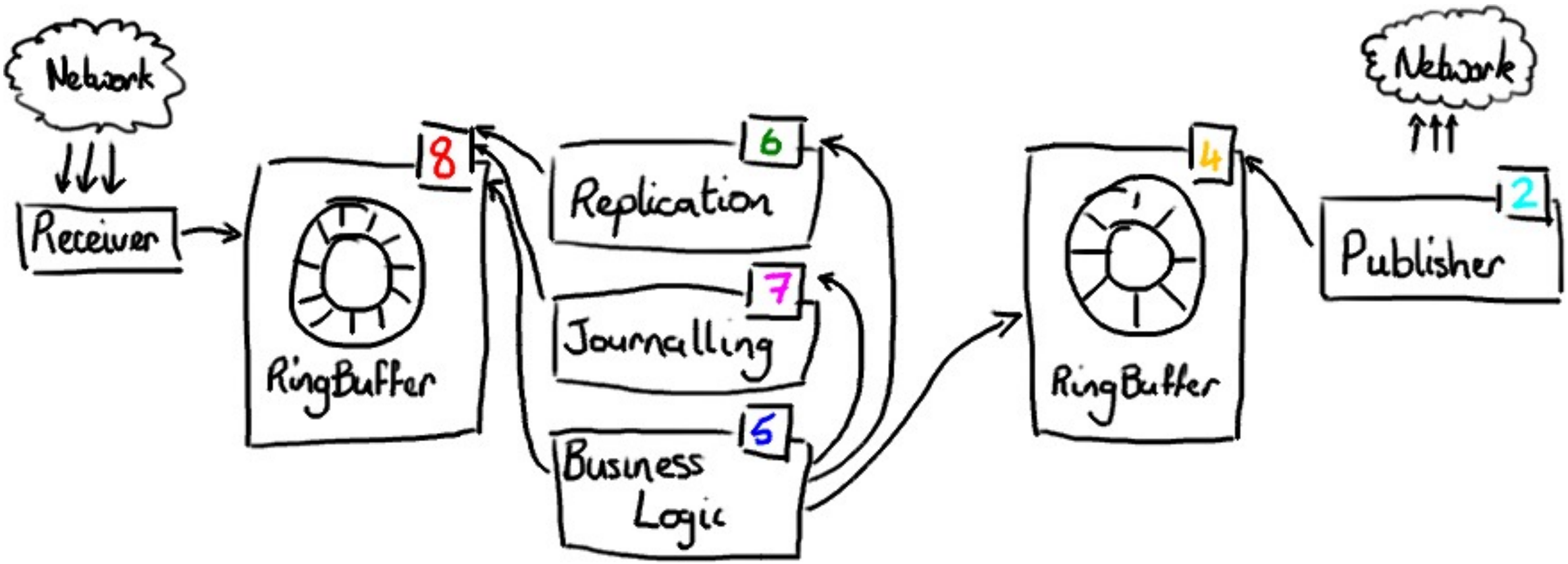

RingBuffer
4
11
0
1
2
3
4
5
6
7
8
9
10
4
BatchEventProcessor
EventHandler

...and all you need is...

```
public class SimpleEventHandler implements EventHandler<SimpleEvent>
{

    @Override
    public void onEvent(final SimpleEvent event,
                        final long sequence,
                        final boolean endOfBatch) throws Exception {
        // do stuff
    }
}
```

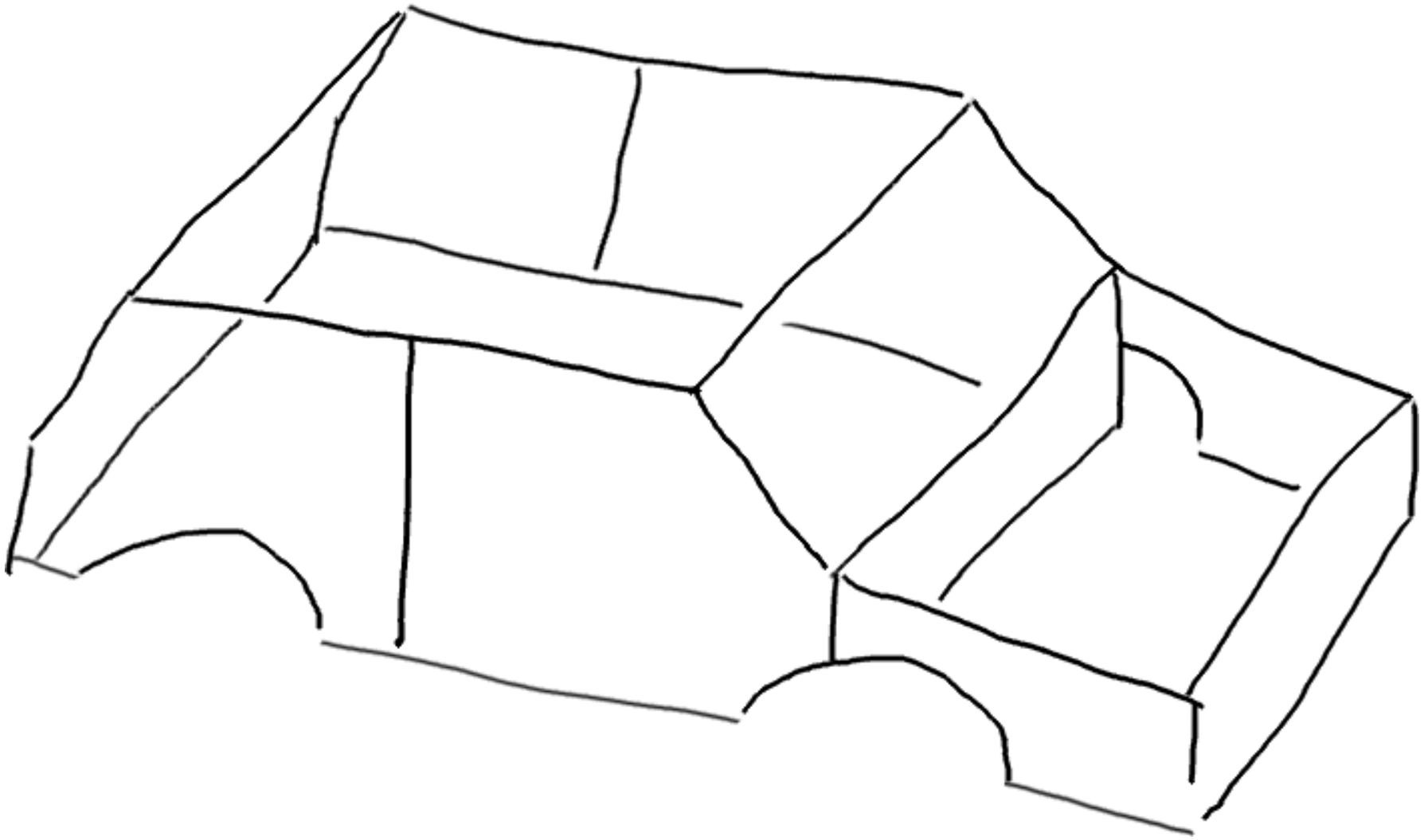
Shiny. So what?

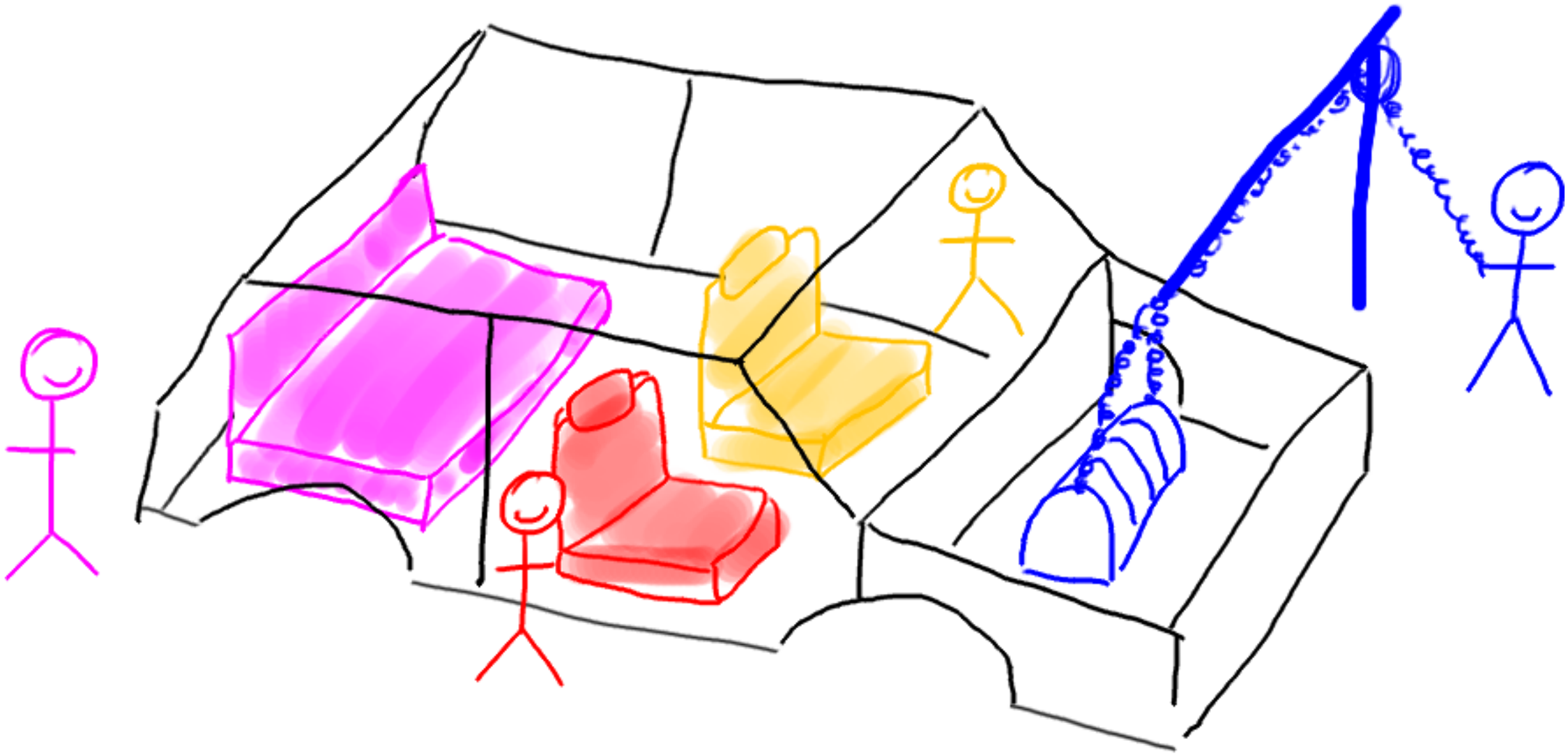
Let's go parallel



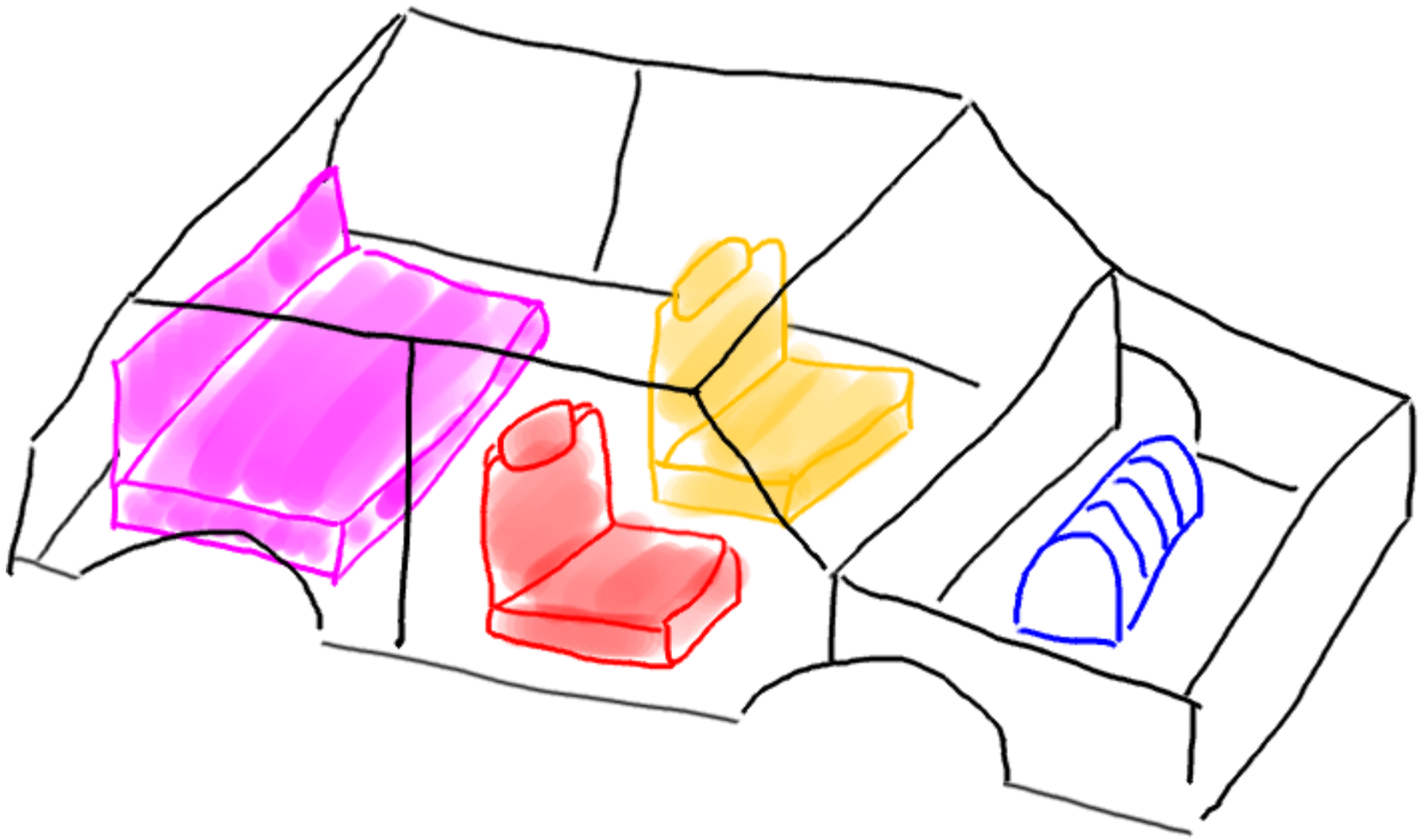
And now for something different...

Remember Henry Ford?

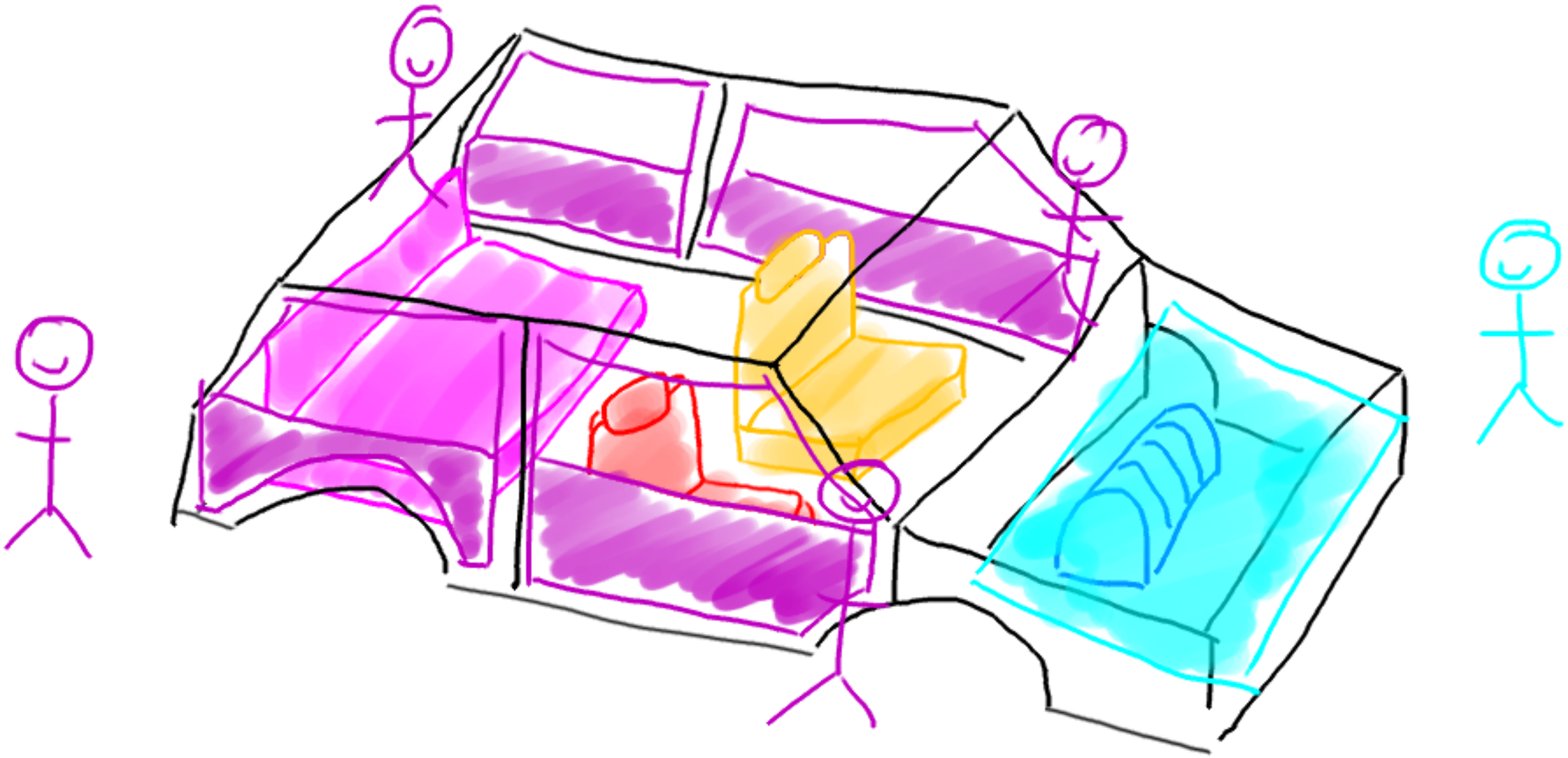




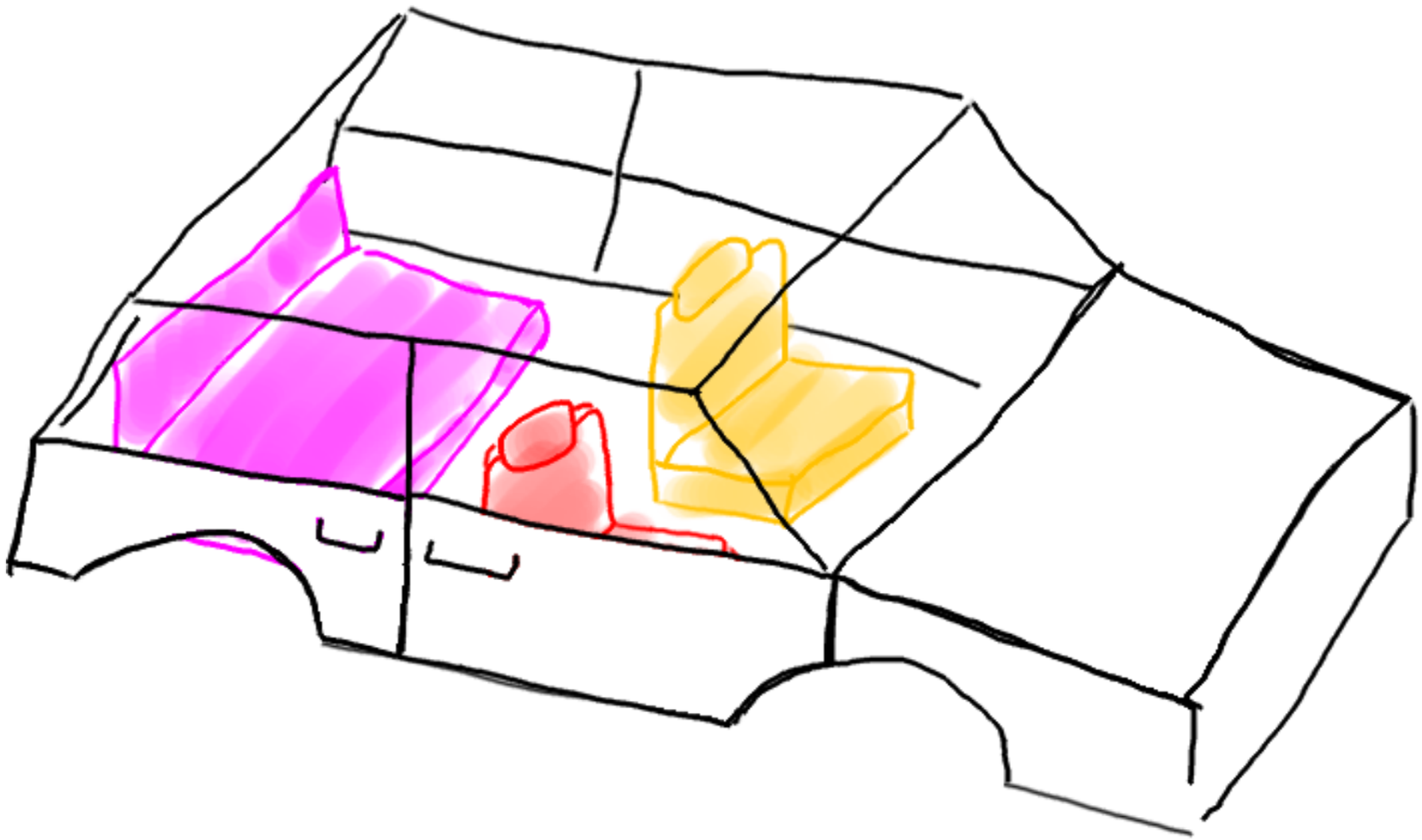
*Not to Scale



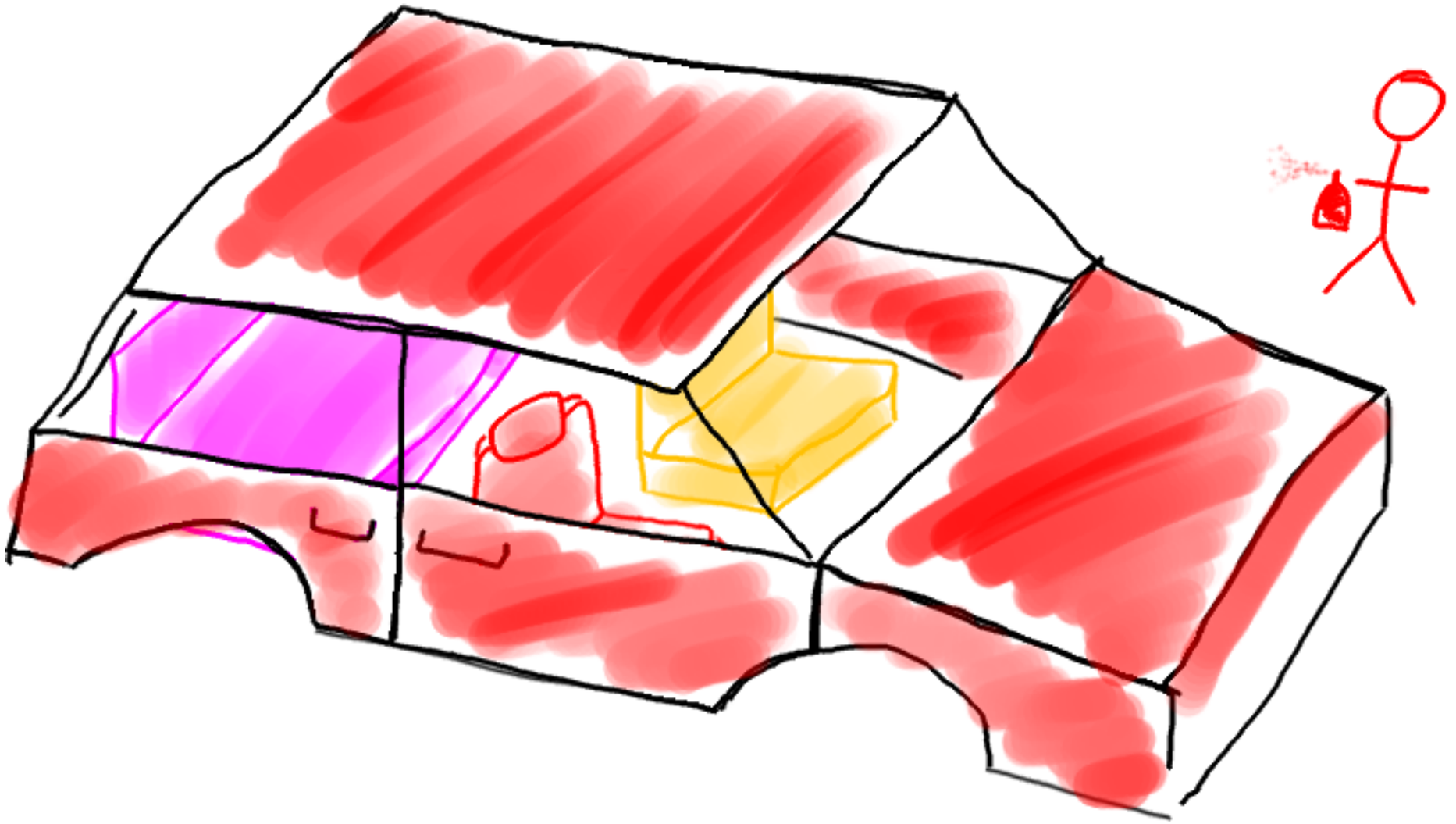
*Not to Scale



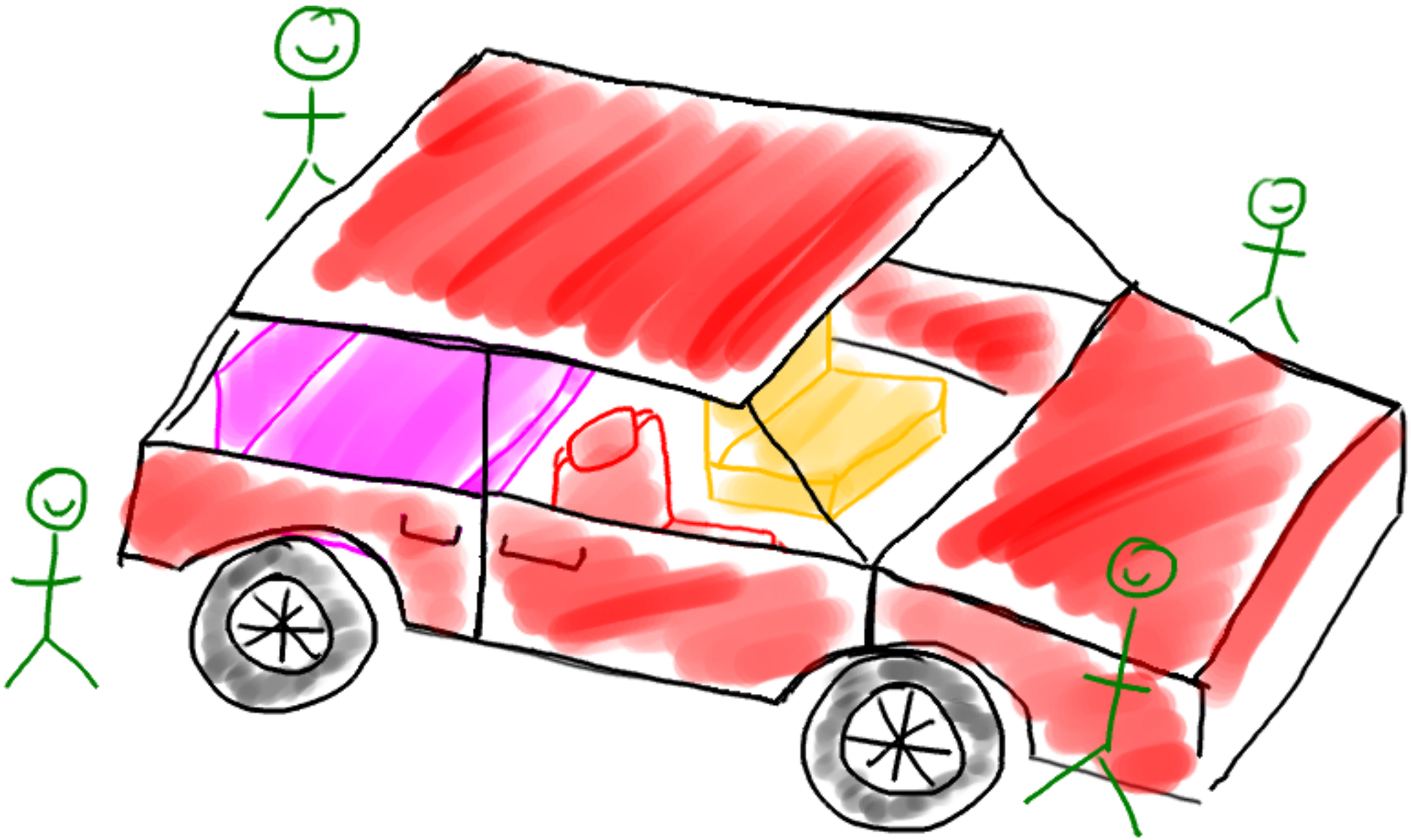
*Not to Scale



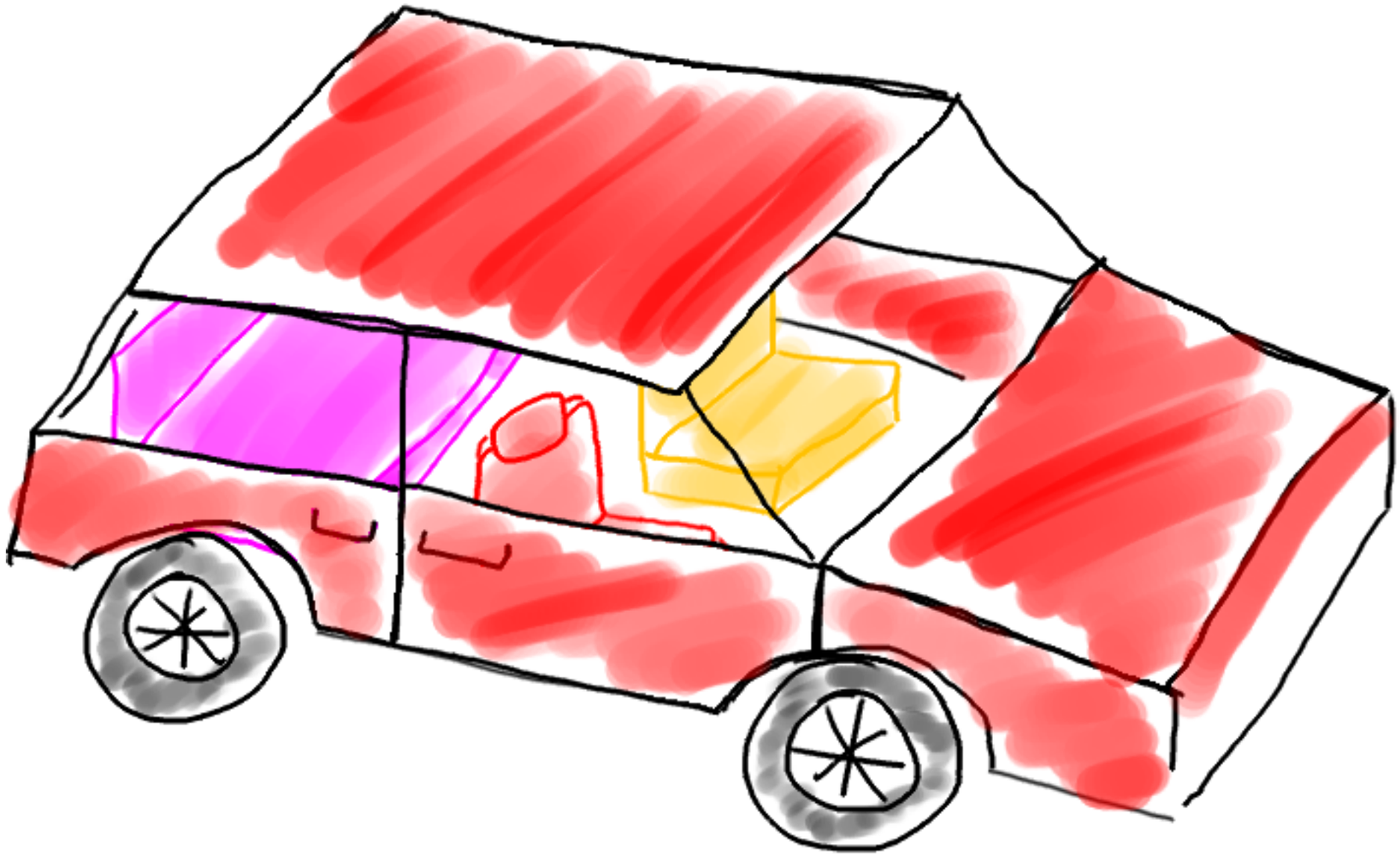
*Not to Scale



*Not to Scale

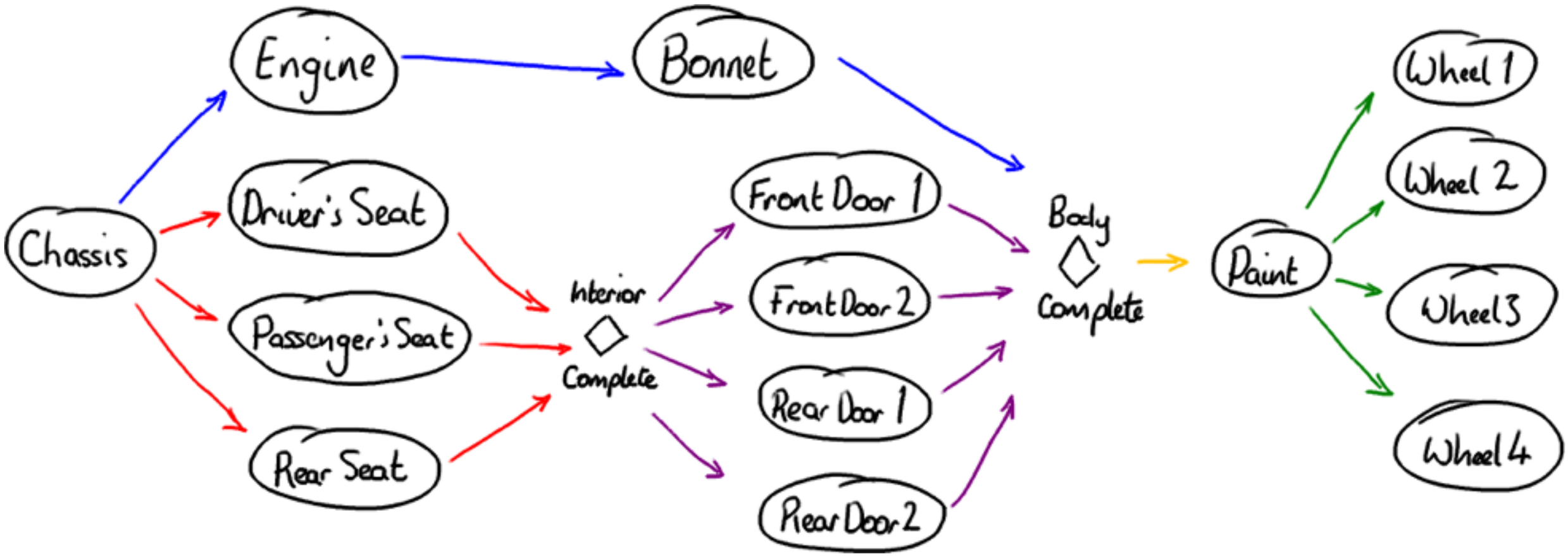


*Not to Scale

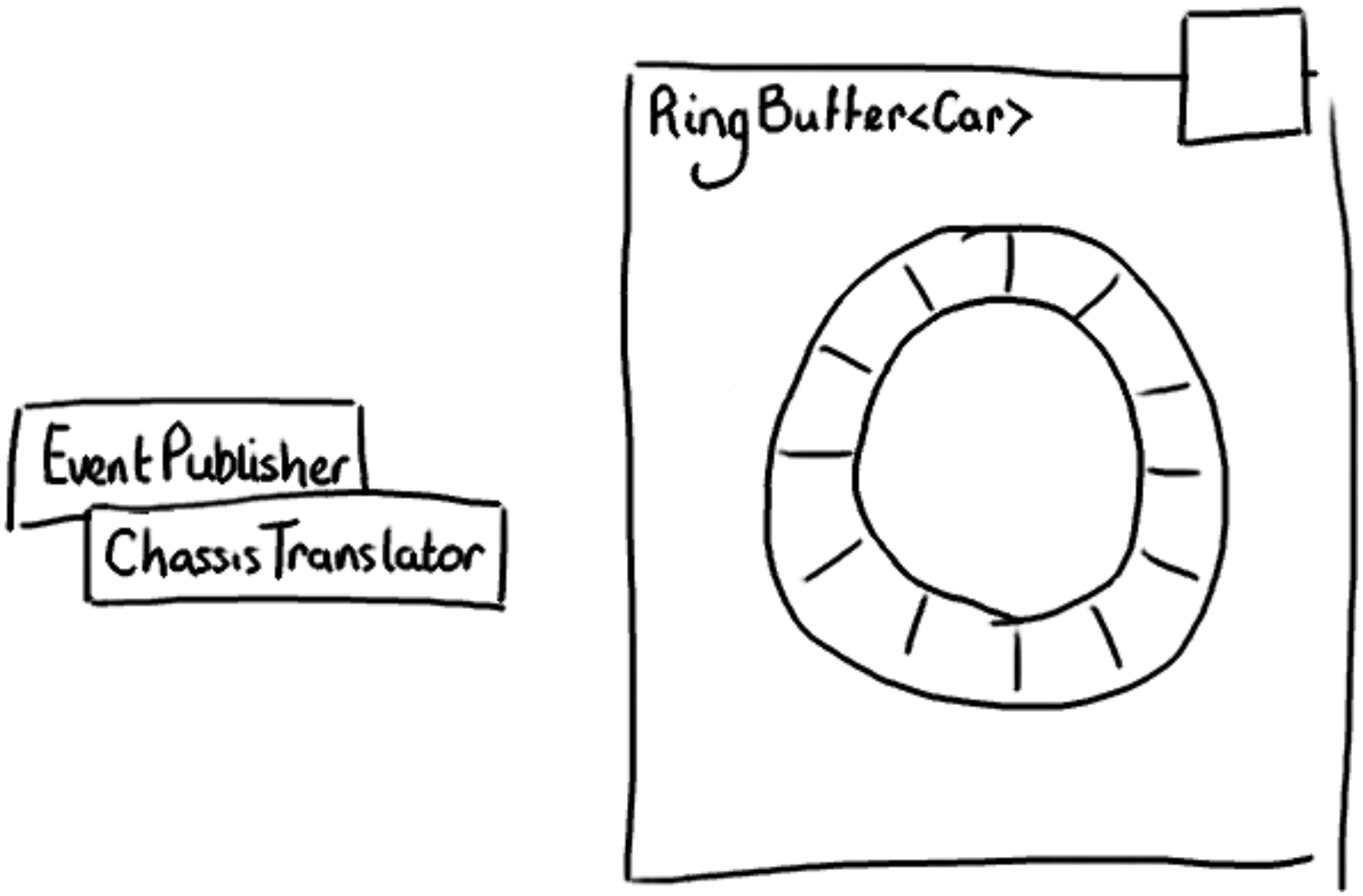


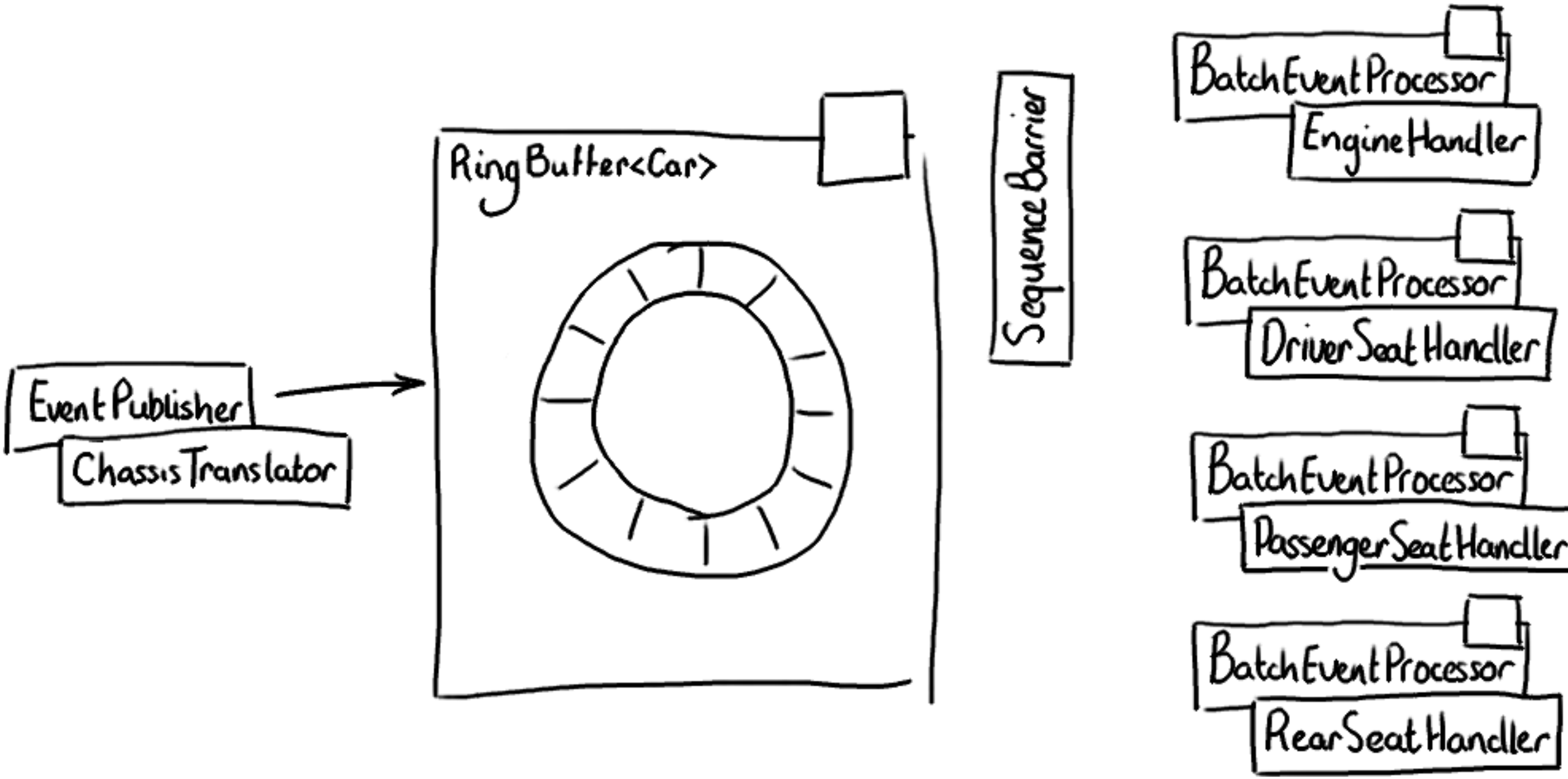
*Not to Scale

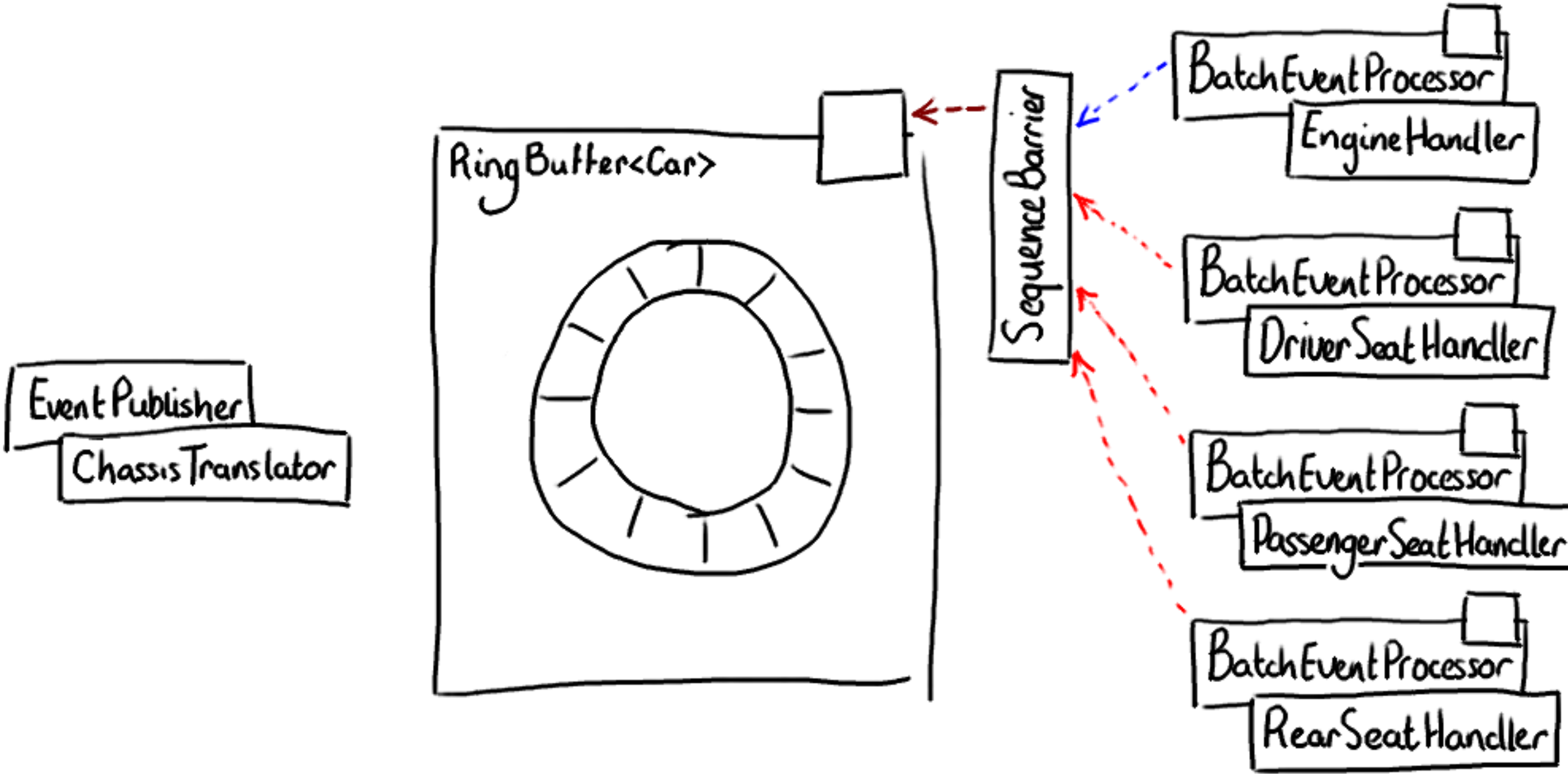
Complex workflow...

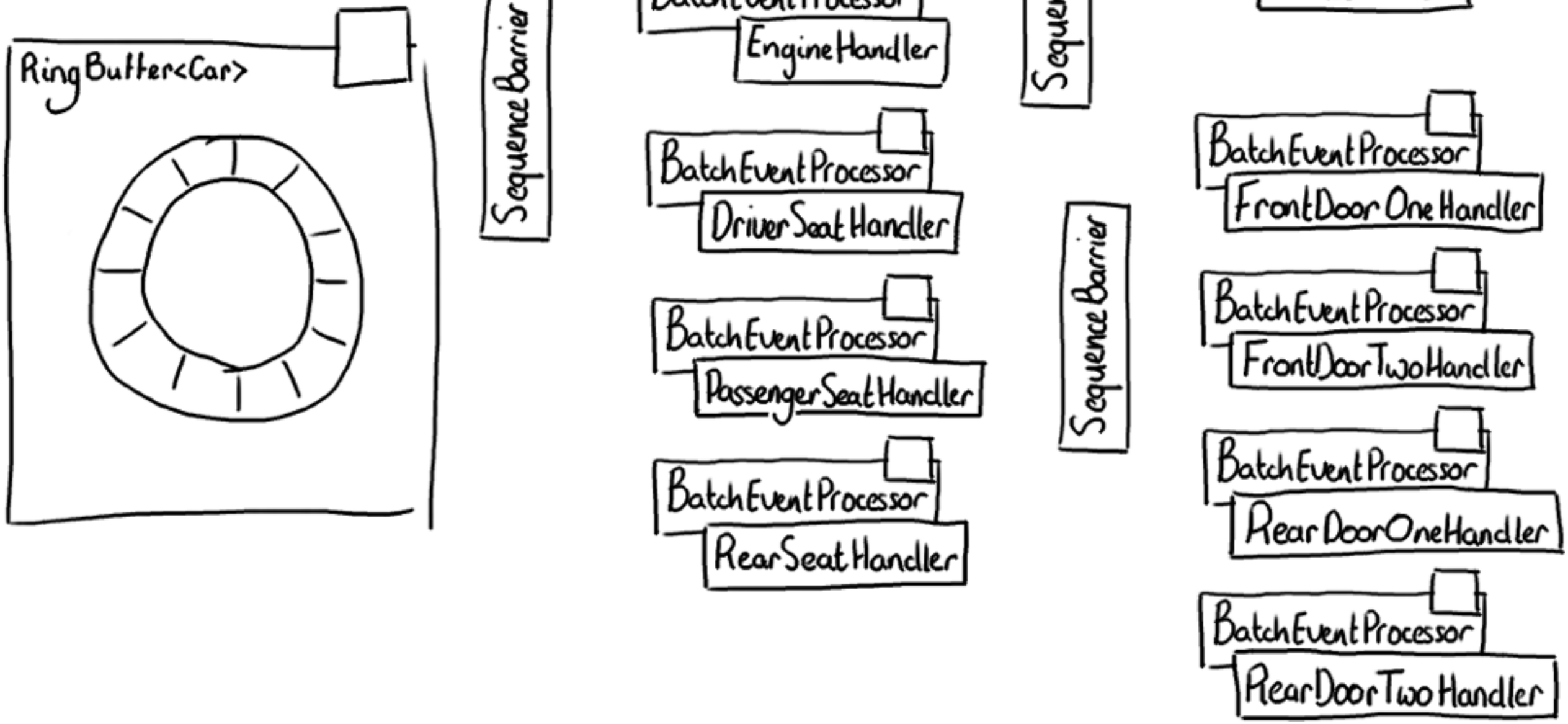


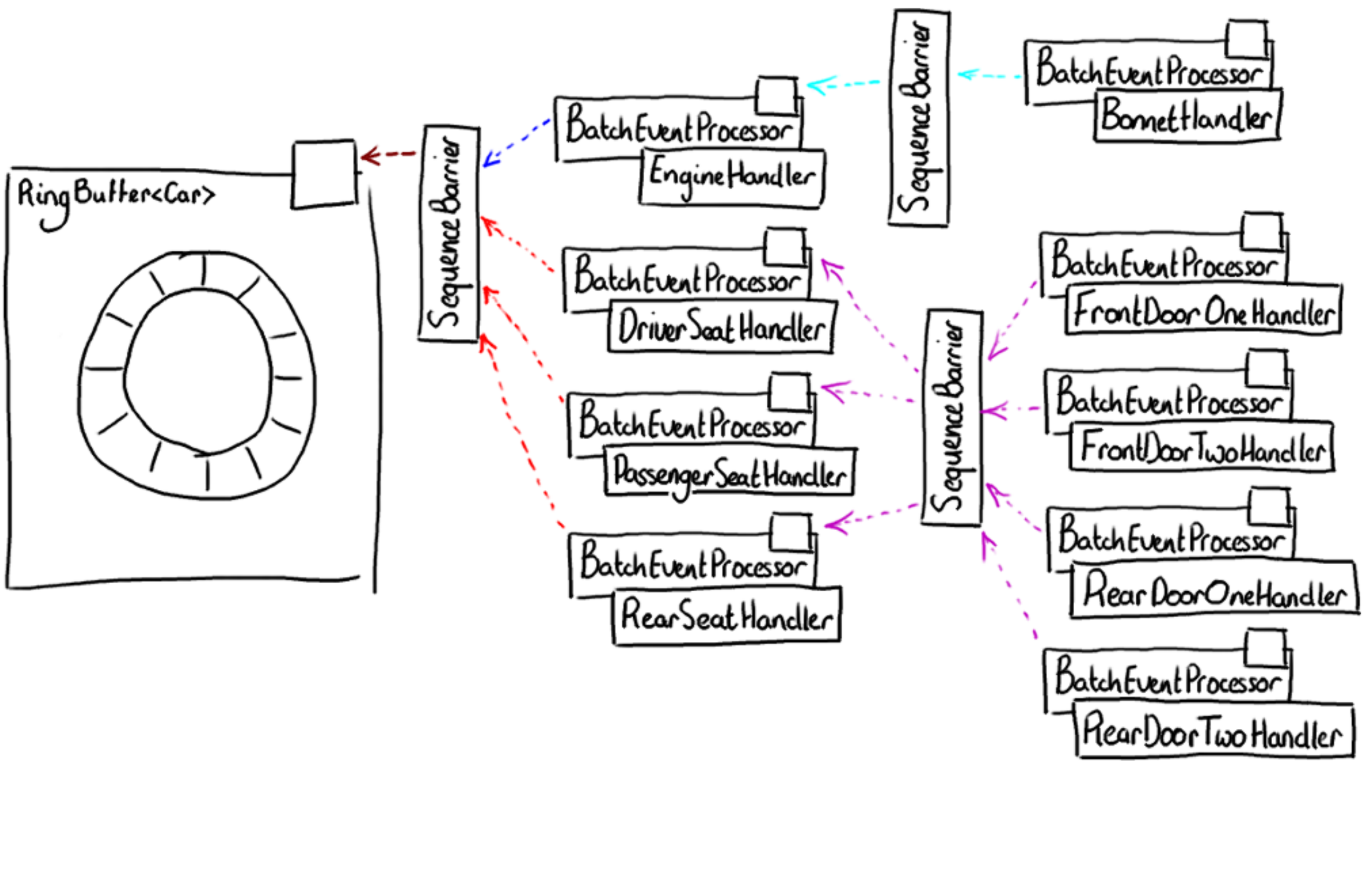
What on Earth has this got to do with RingBuffers?!

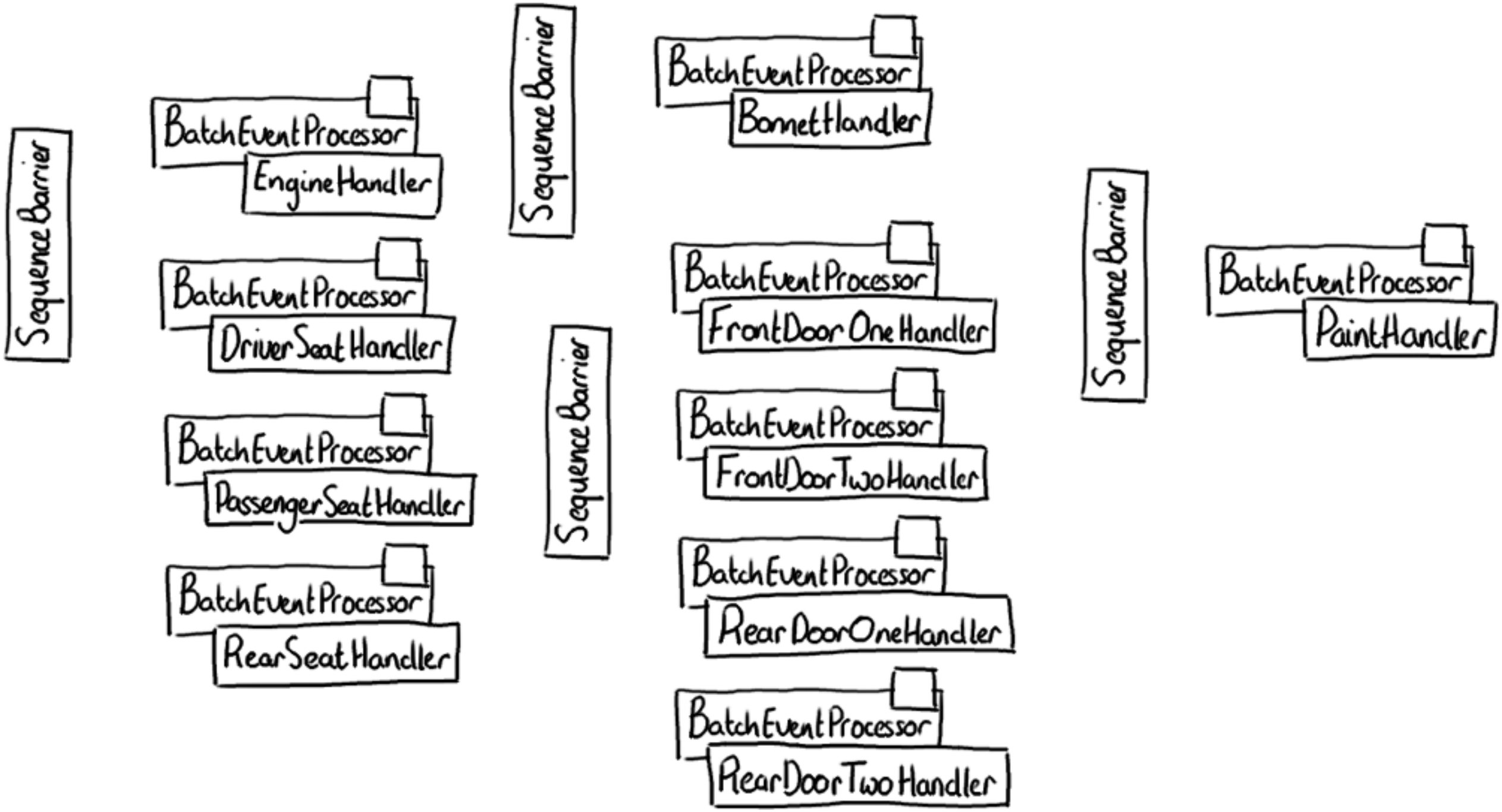

RingButter<Car>
EventPublisher
ChassisTranslator

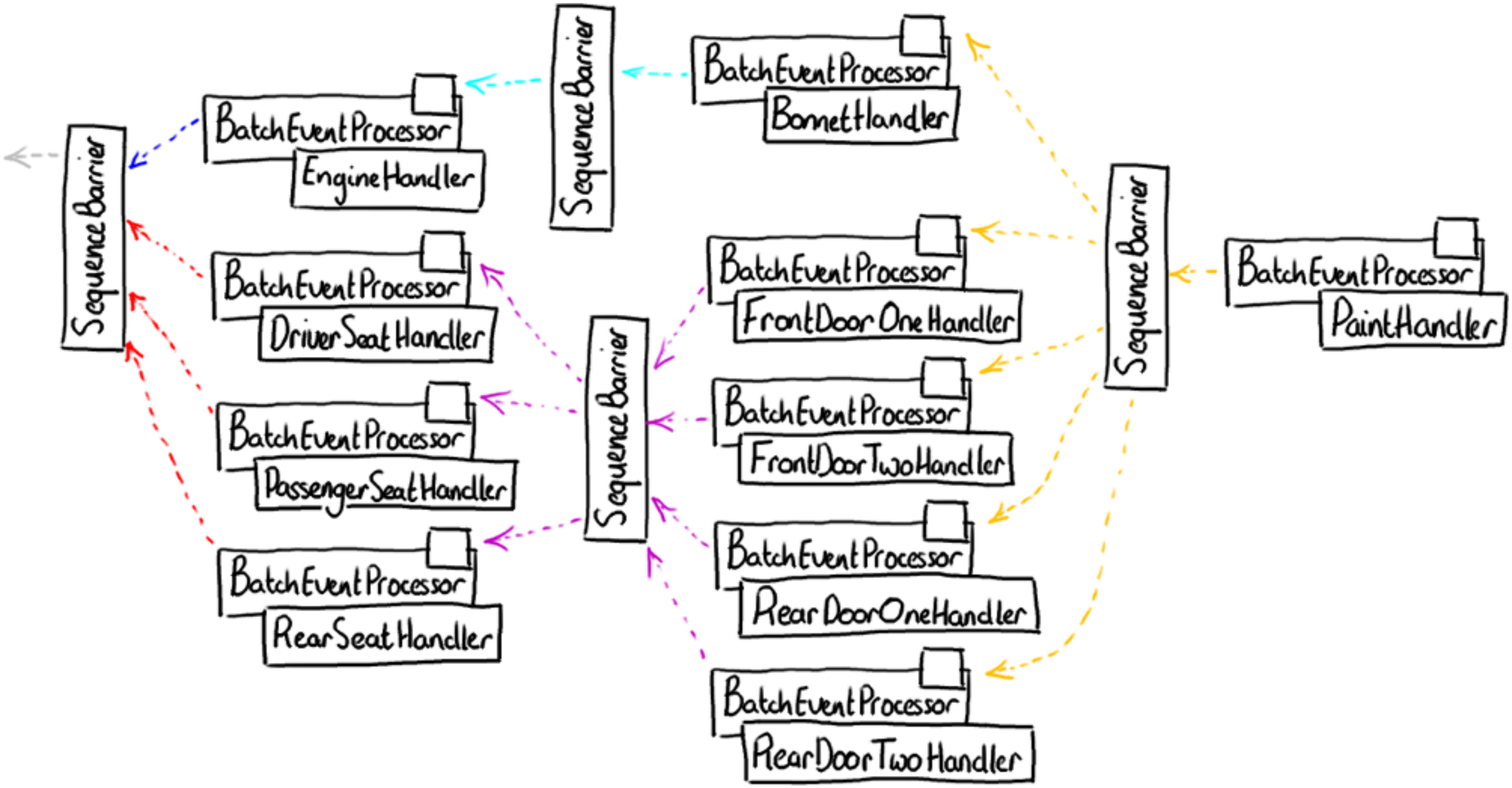

RingButter<Car>
SequenceBarrier
BatchEventProcessor
EngineHandler
BatchEventProcessor
DriverSeatHandler
EventPublisher
ChassisTranslator
BatchEventProcessor
PassengerSeatHandler
BatchEventProcessor
RearSeatHandler

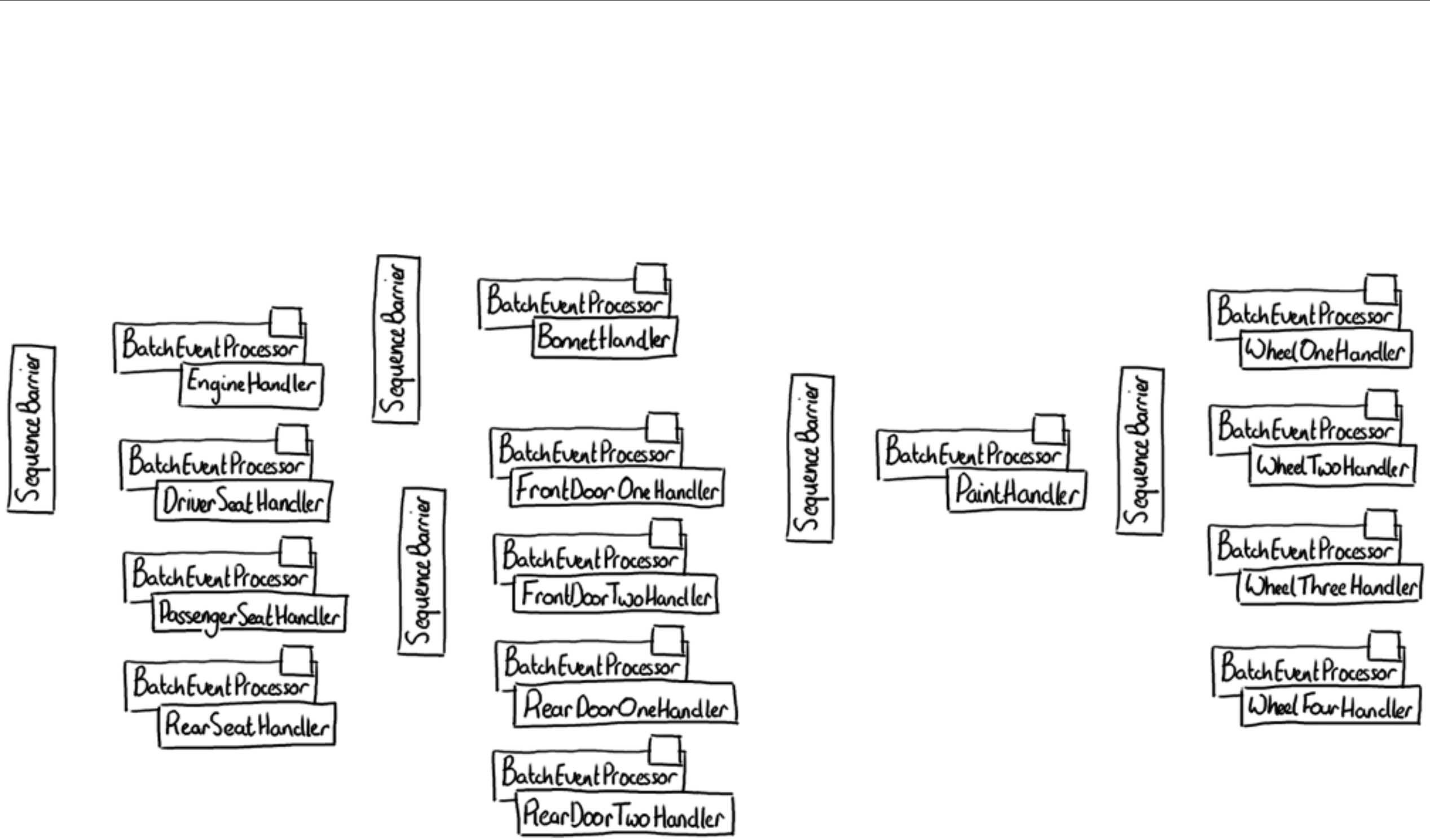

EventPublisher
ChassisTranslator
RingBuffer<Car>
SequenceBarrier
BatchEventProcessor
EngineHandler
BatchEventProcessor
DriverSeatHandler
BatchEventProcessor
PassengerSeatHandler
BatchEventProcessor
RearSeatHandler

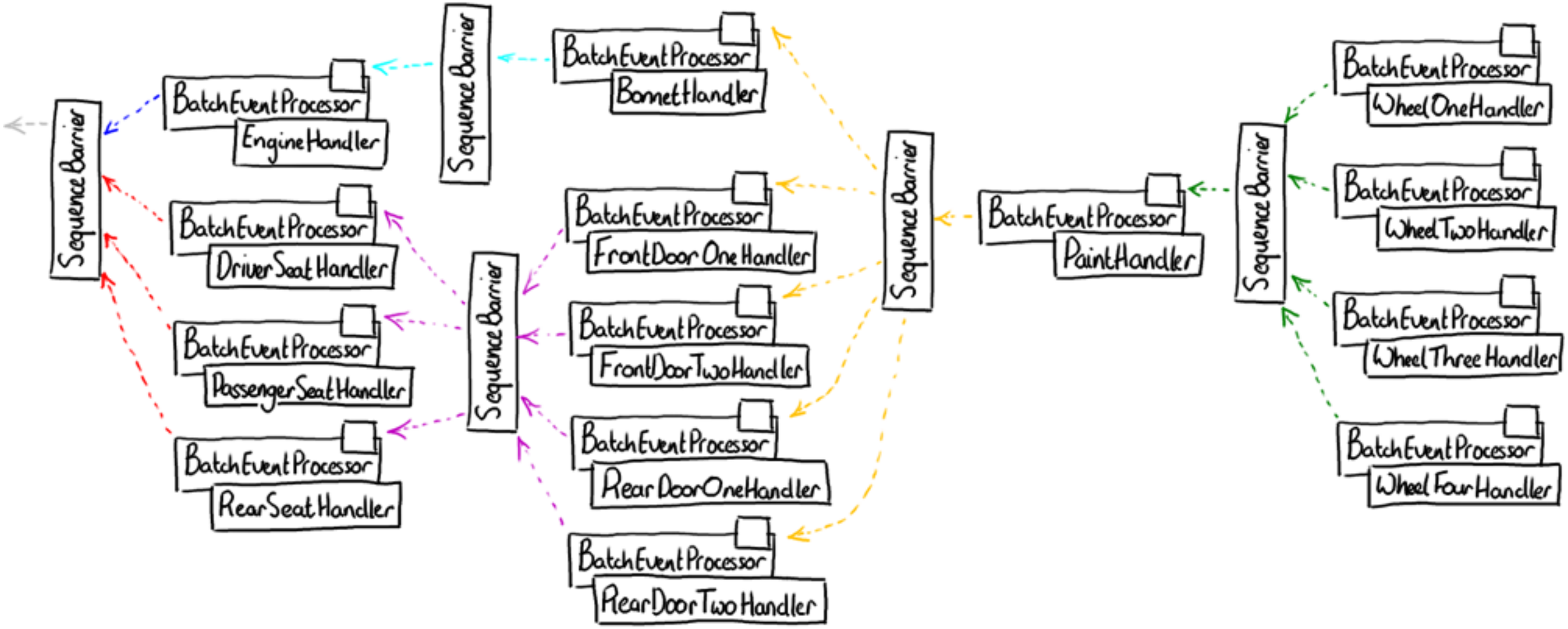

RingButter<Car>
SequenceBarrier
BatchEventProcessor
EngineHandler
BatchEventProcessor
DriverSeatHandler
BatchEventProcessor
PassengerSeatHandler
BatchEventProcessor
RearSeatHandler
SequenceBarrier
SequenceBarrier
BatchEventProcessor
BonnetHandler
BatchEventProcessor
FrontDoorOneHandler
BatchEventProcessor
FrontDoorTwoHandler
BatchEventProcessor
RearDoorOneHandler
BatchEventProcessor
RearDoorTwoHandler

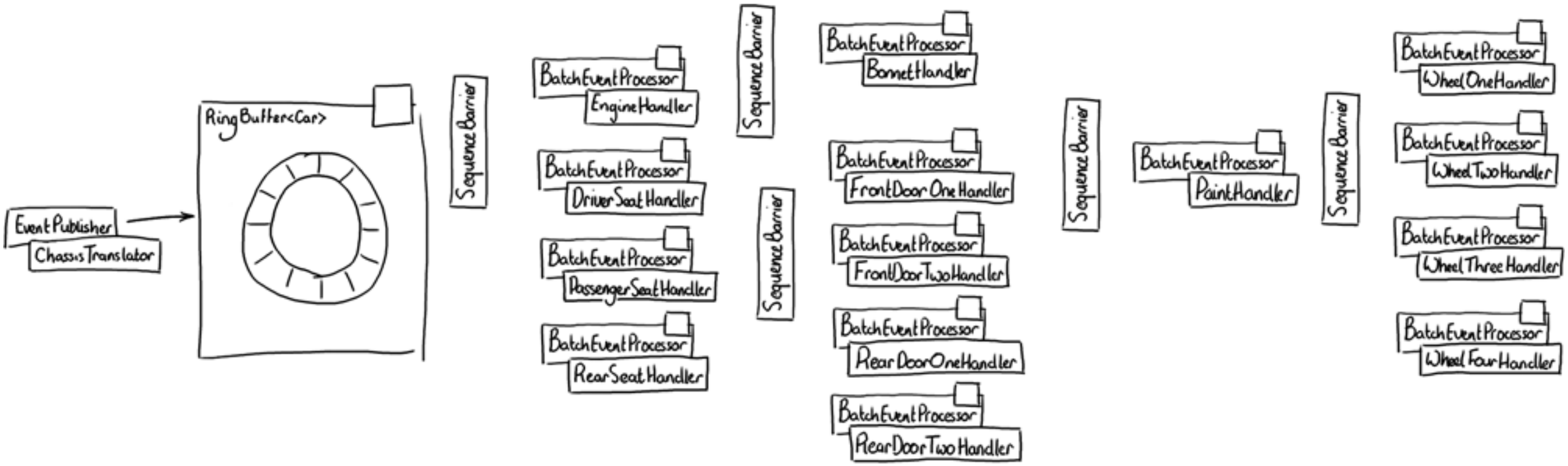

RingButter<Car>
SequenceBarrier
BatchEventProcessor
EngineHandler
SequenceBarrier
BatchEventProcessor
BonnetHandler
BatchEventProcessor
DriverSeatHandler
BatchEventProcessor
PassengerSeatHandler
BatchEventProcessor
RearSeatHandler
SequenceBarrier
BatchEventProcessor
FrontDoorOneHandler
BatchEventProcessor
FrontDoorTwoHandler
BatchEventProcessor
RearDoorOneHandler
BatchEventProcessor
RearDoorTwoHandler

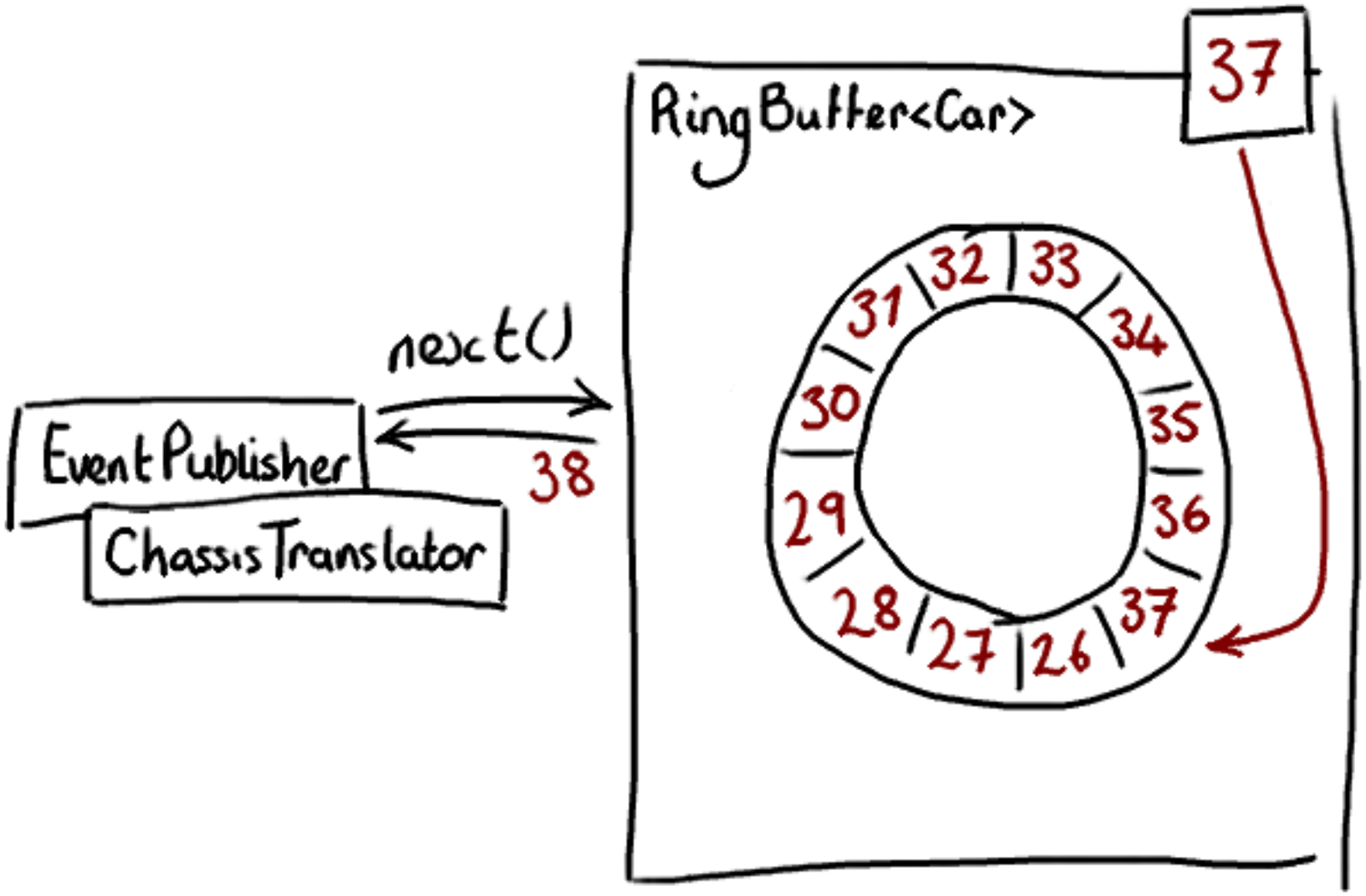

Sequence Barrier
BatchEventProcessor
EngineHandler
BatchEventProcessor
DriverSeatHandler
BatchEventProcessor
PassengerSeatHandler
BatchEventProcessor
RearSeatHandler
Sequence Barrier
Sequence Barrier
BatchEventProcessor
BonnetHandler
BatchEventProcessor
FrontDoorOneHandler
BatchEventProcessor
FrontDoorTwoHandler
BatchEventProcessor
RearDoorOneHandler
BatchEventProcessor
RearDoorTwoHandler
Sequence Barrier
BatchEventProcessor
PaintHandler

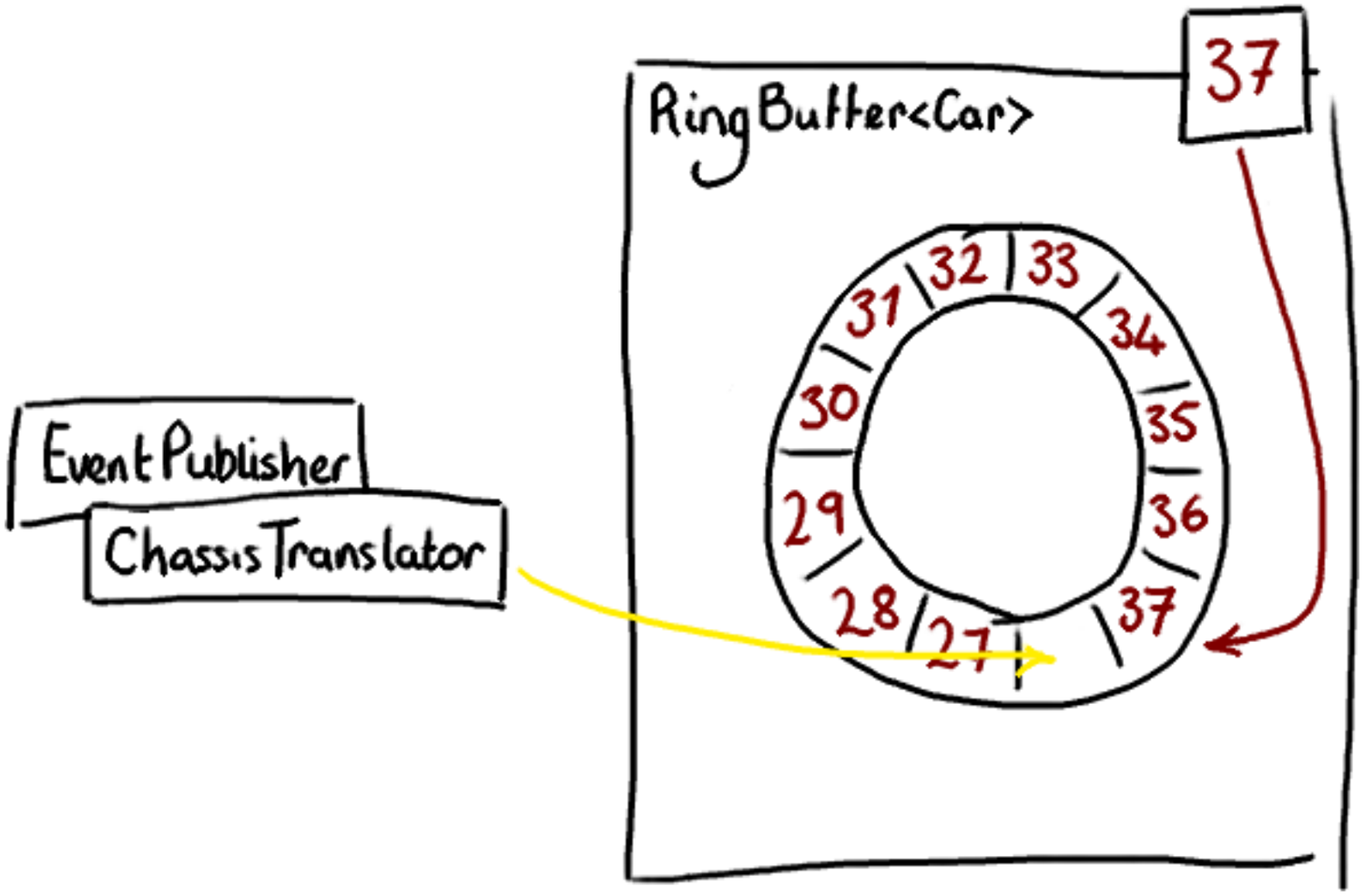

Sequence Barrier
BatchEventProcessor
EngineHandler
BatchEventProcessor
DriverSeatHandler
BatchEventProcessor
PassengerSeatHandler
BatchEventProcessor
RearSeatHandler
Sequence Barrier
BatchEventProcessor
BonnetHandler
BatchEventProcessor
FrontDoorOneHandler
BatchEventProcessor
FrontDoorTwoHandler
BatchEventProcessor
RearDoorOneHandler
BatchEventProcessor
RearDoorTwoHandler
Sequence Barrier
BatchEventProcessor
PaintHandler

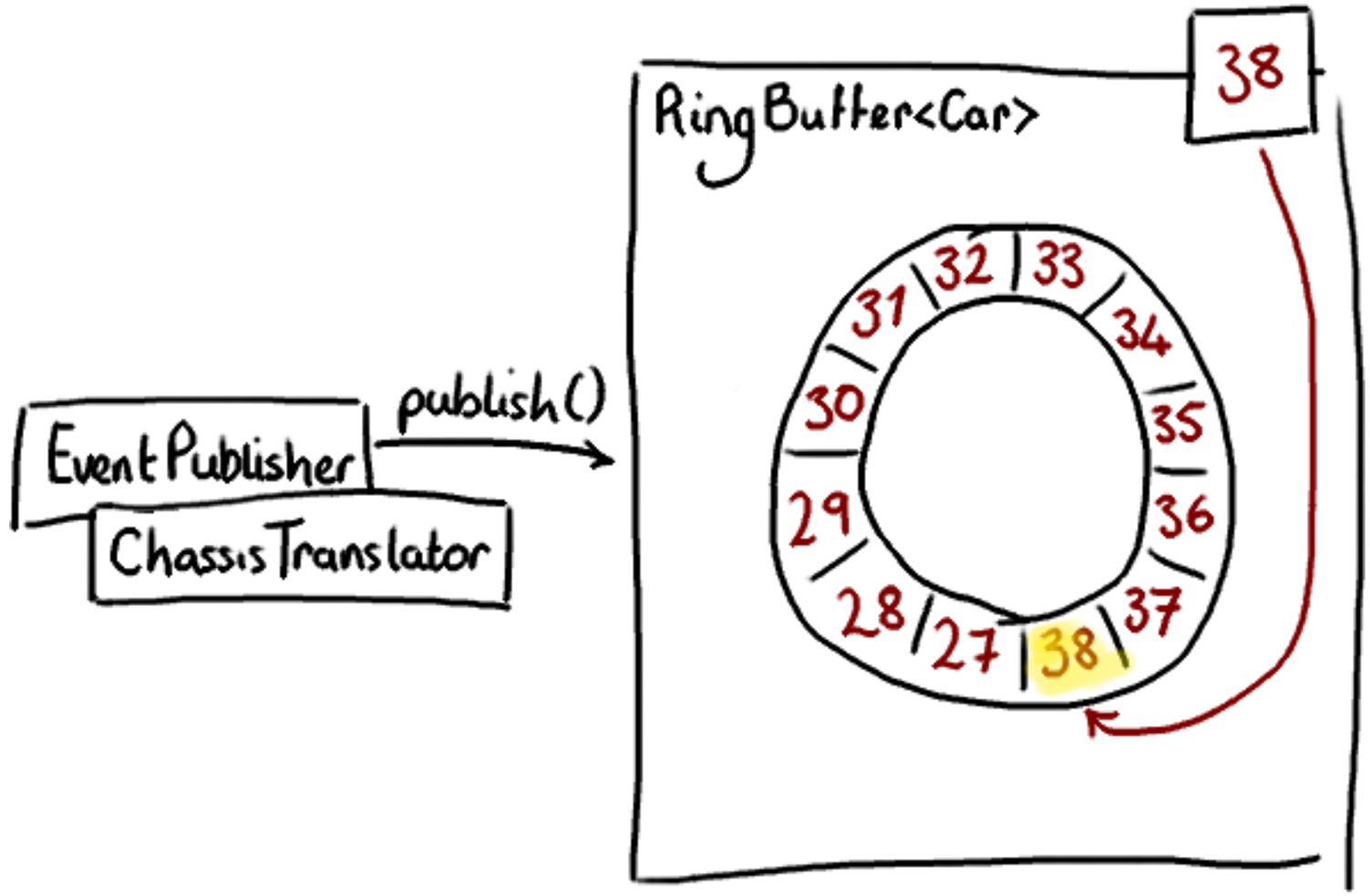

Sequence Barrier
BatchEventProcessor
EngineHandler
BatchEventProcessor
DriverSeatHandler
BatchEventProcessor
PassengerSeatHandler
BatchEventProcessor
RearSeatHandler
Sequence Barrier
Sequence Barrier
BatchEventProcessor
BonnetHandler
BatchEventProcessor
FrontDoorOneHandler
BatchEventProcessor
FrontDoorTwoHandler
BatchEventProcessor
RearDoorOneHandler
BatchEventProcessor
RearDoorTwoHandler
Sequence Barrier
BatchEventProcessor
PaintHandler
Sequence Barrier
BatchEventProcessor
WheelOneHandler
BatchEventProcessor
WheelTwoHandler
BatchEventProcessor
WheelThreeHandler
BatchEventProcessor
WheelFourHandler

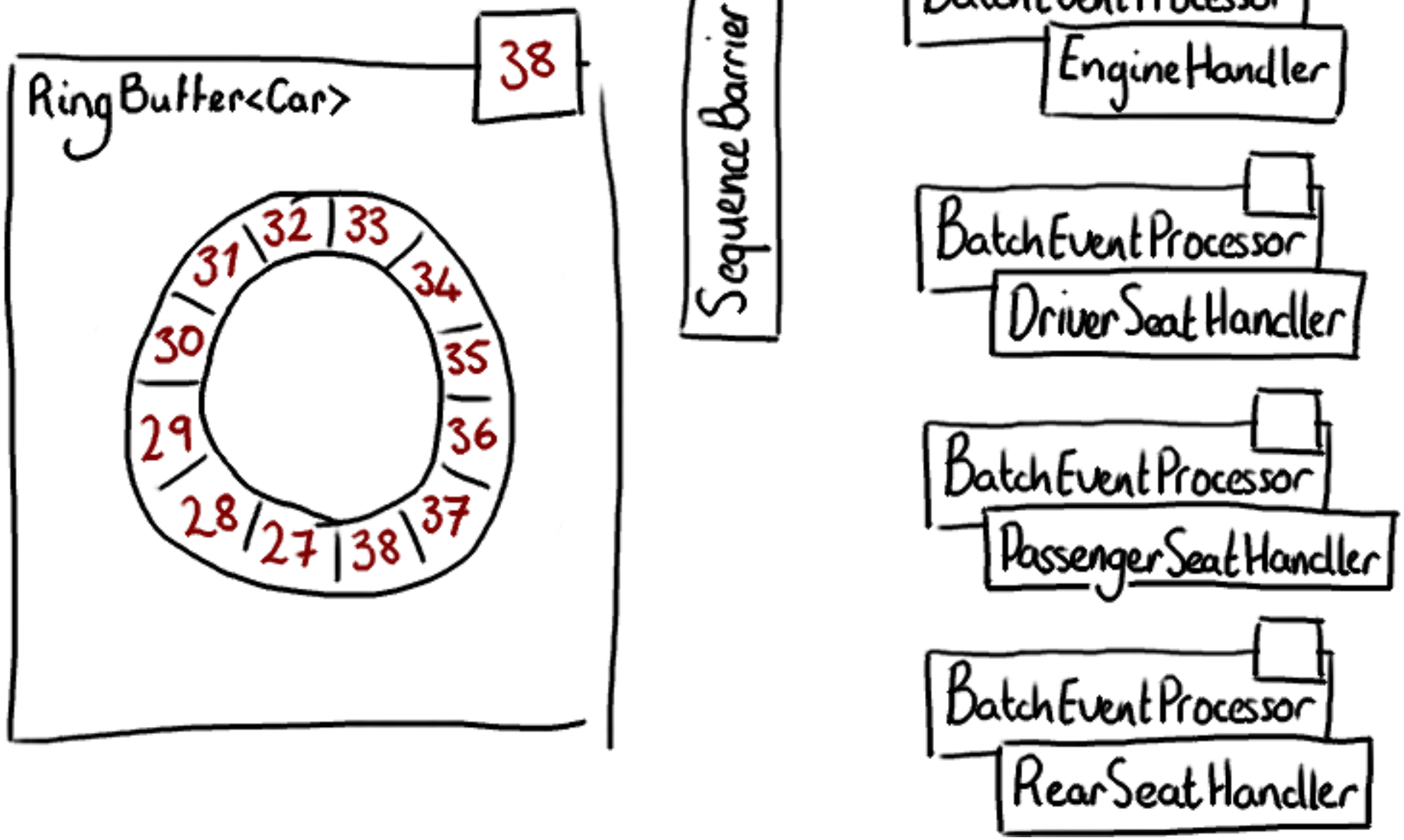

Sequence Barrier
BatchEventProcessor
EngineHandler
BatchEventProcessor
DriverSeatHandler
BatchEventProcessor
PassengerSeatHandler
BatchEventProcessor
RearSeatHandler
Sequence Barrier
BatchEventProcessor
BonnetHandler
Sequence Barrier
BatchEventProcessor
FrontDoorOneHandler
BatchEventProcessor
FrontDoorTwoHandler
BatchEventProcessor
RearDoorOneHandler
BatchEventProcessor
RearDoorTwoHandler
Sequence Barrier
BatchEventProcessor
PaintHandler
Sequence Barrier
BatchEventProcessor
WheelOneHandler
BatchEventProcessor
WheelTwoHandler
BatchEventProcessor
WheelThreeHandler
BatchEventProcessor
WheelFourHandler

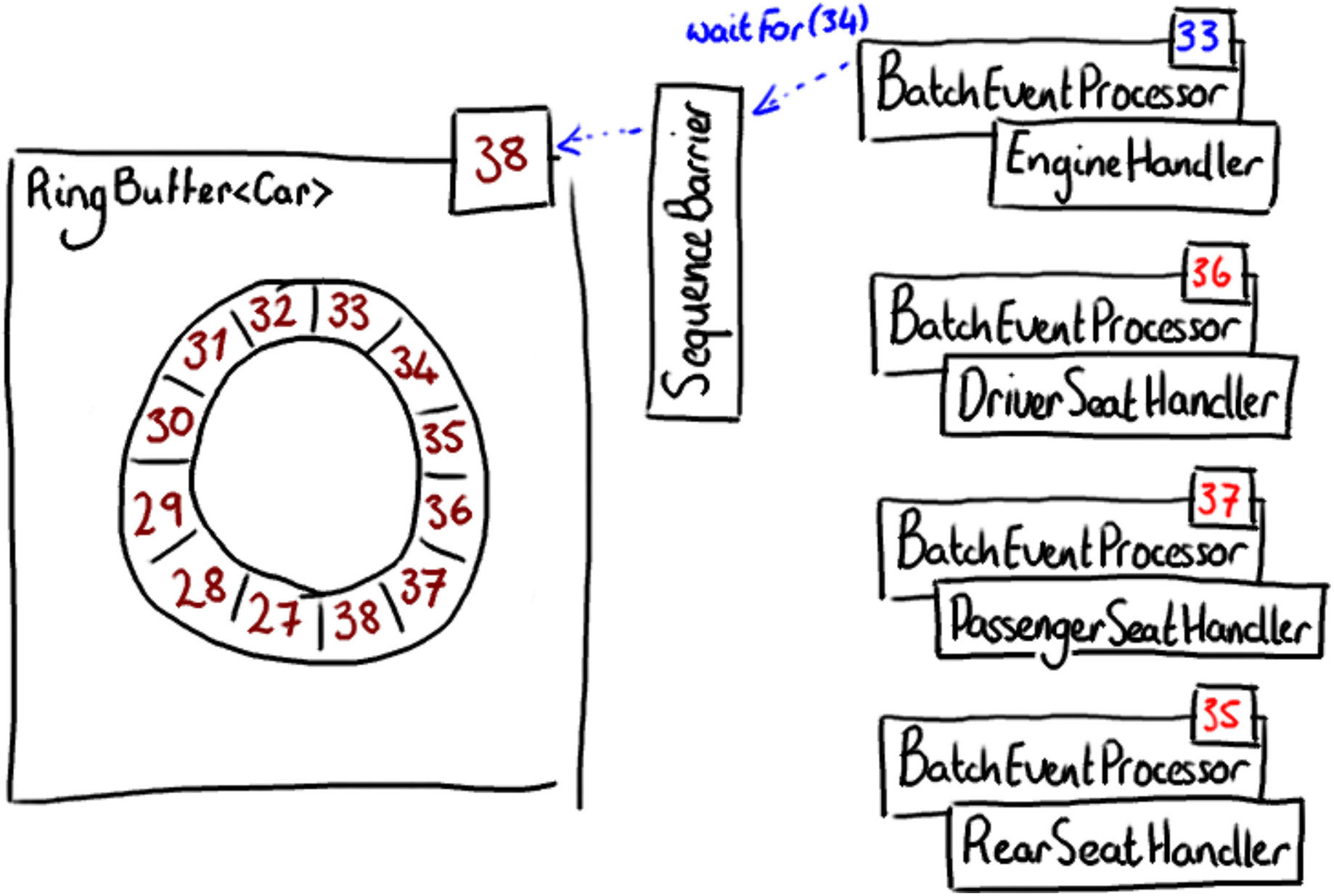

Event Publisher
Chassis Translator
RingBuffer<Car>
Sequence Barrier
BatchEventProcessor
EngineHandler
BatchEventProcessor
DriverSeatHandler
BatchEventProcessor
PassengerSeatHandler
BatchEventProcessor
RearSeatHandler
Sequence Barrier
Sequence Barrier
BatchEventProcessor
BonnetHandler
BatchEventProcessor
FrontDoorOneHandler
BatchEventProcessor
FrontDoorTwoHandler
BatchEventProcessor
RearDoorOneHandler
BatchEventProcessor
RearDoorTwoHandler
Sequence Barrier
BatchEventProcessor
PaintHandler
Sequence Barrier
BatchEventProcessor
WheelOneHandler
BatchEventProcessor
WheelTwoHandler
BatchEventProcessor
WheelThreeHandler
BatchEventProcessor
WheelFourHandler

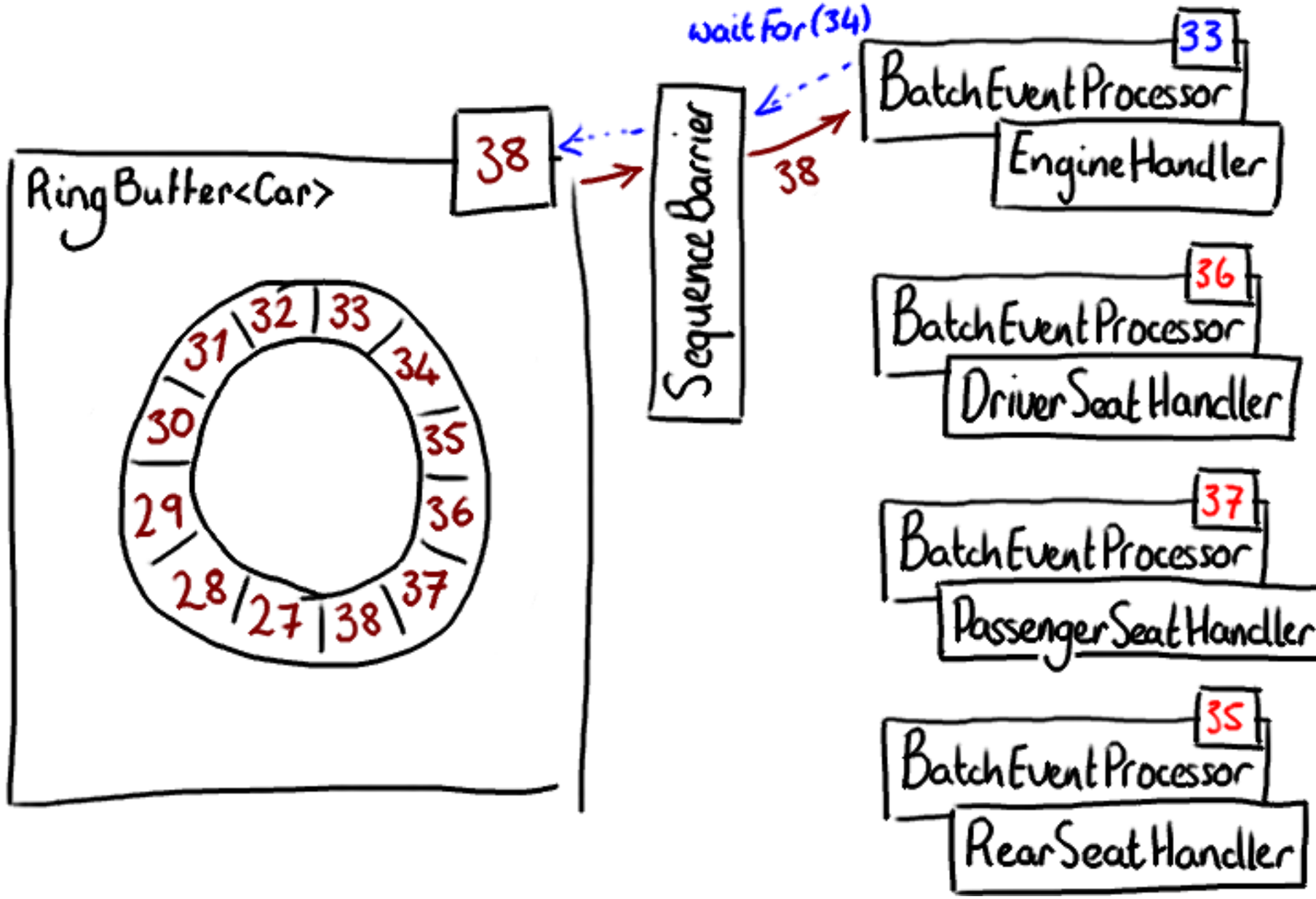

RingButter<Car>
37
next()
38
EventPublisher
ChassisTranslator
32
33
31
34
30
35
29
36
28
27
26
37

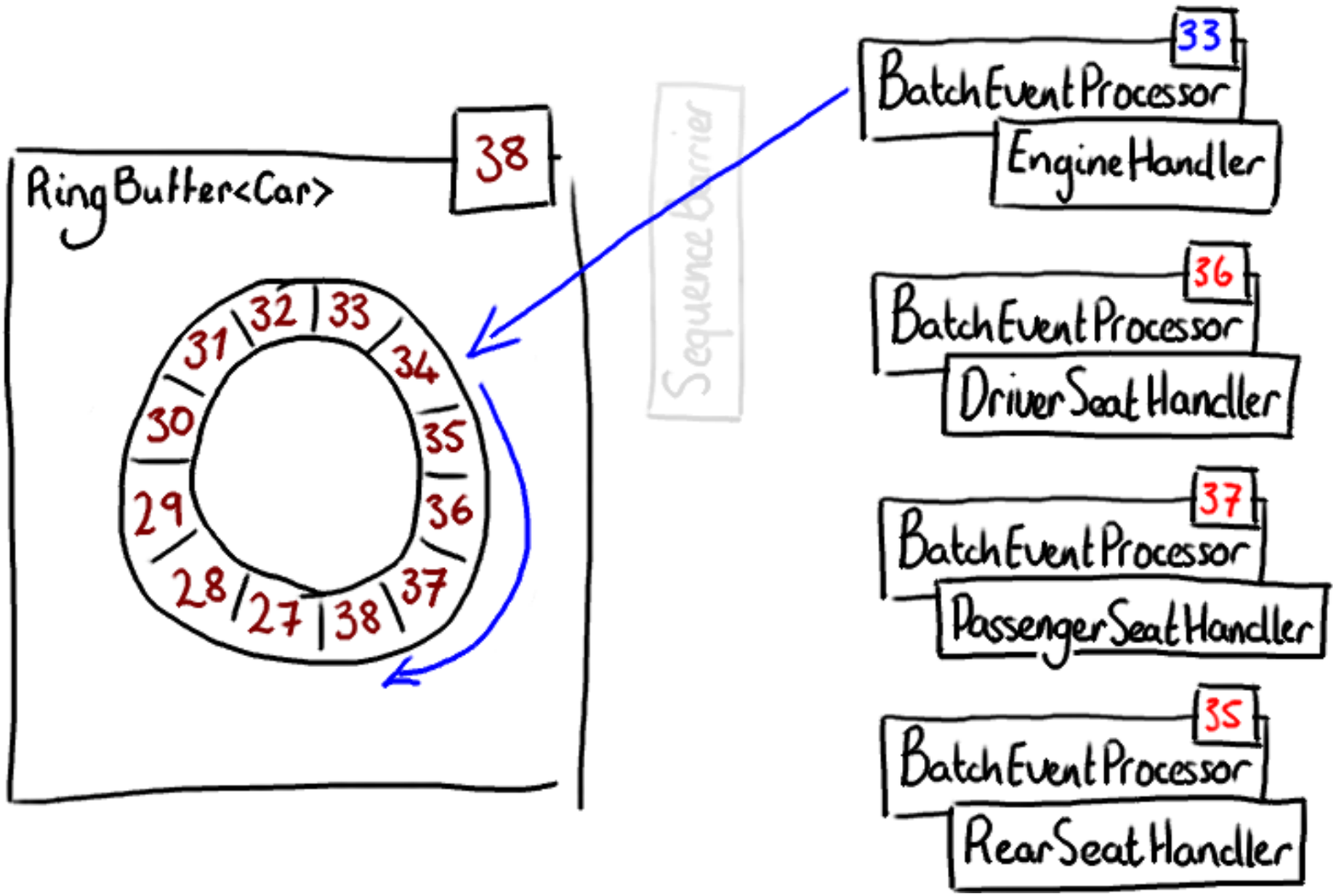

RingButter<Car>
37
32
33
31
34
30
35
29
36
28
27
37
EventPublisher
ChassisTranslator

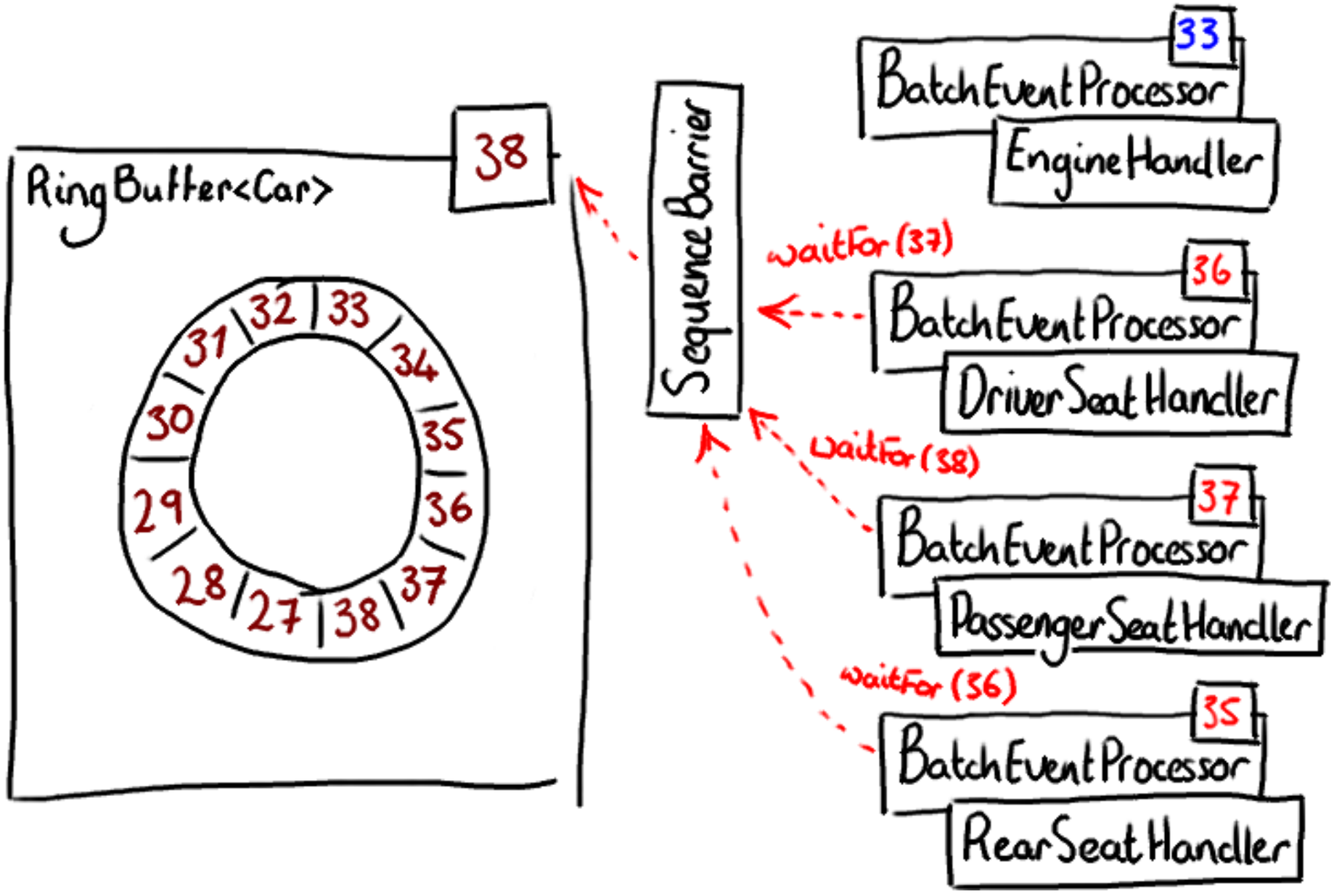

RingButter<Car>
38
publish()
EventPublisher
ChassisTranslator
31
32
33
34
30
35
29
36
28
27
38
37

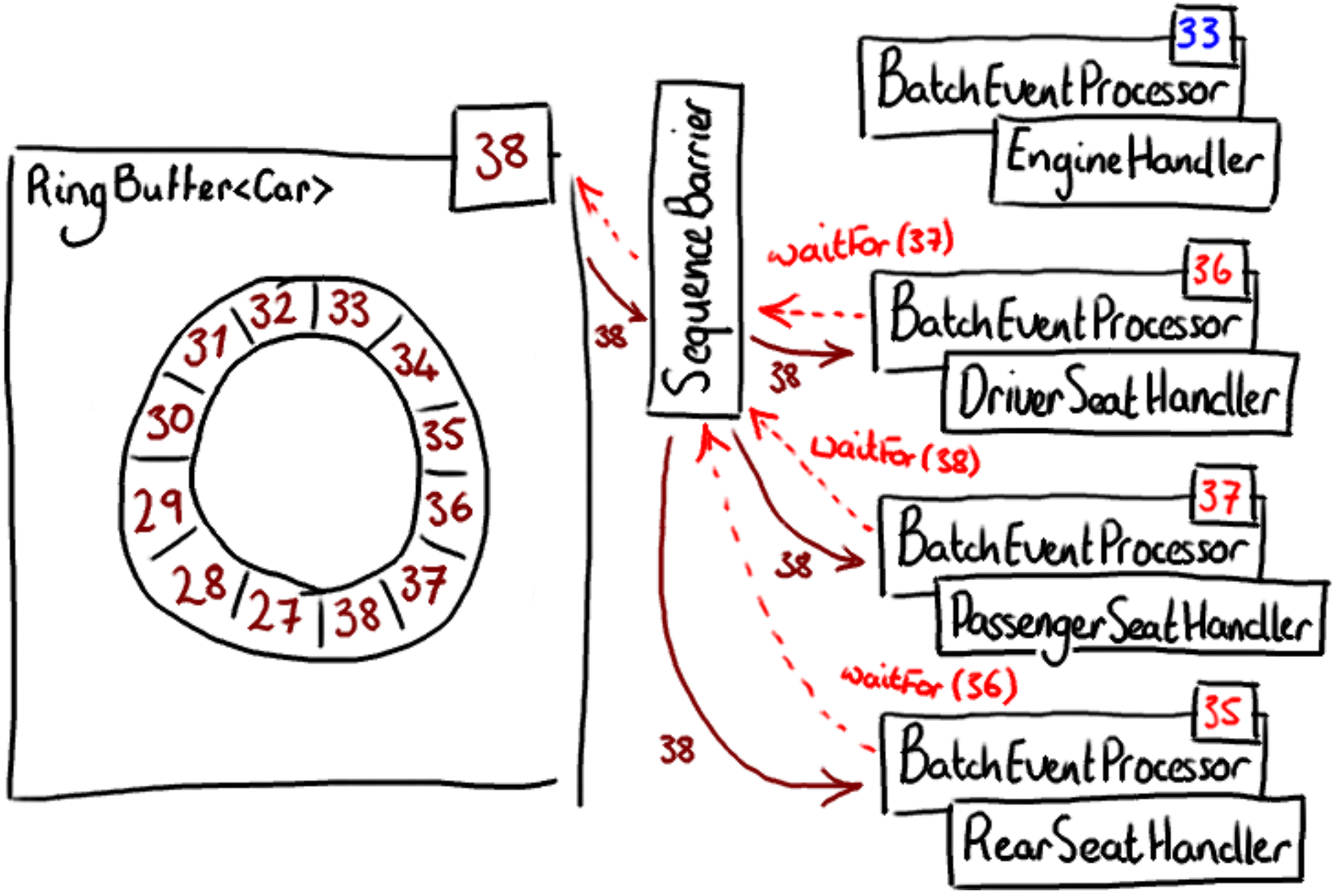

RingButter<Car>
38
30
31
32
33
34
35
29
36
28
27
38
37
SequenceBarrier
BatchEventProcessor
EngineHandler
BatchEventProcessor
DriverSeatHandler
BatchEventProcessor
PassengerSeatHandler
BatchEventProcessor
RearSeatHandler

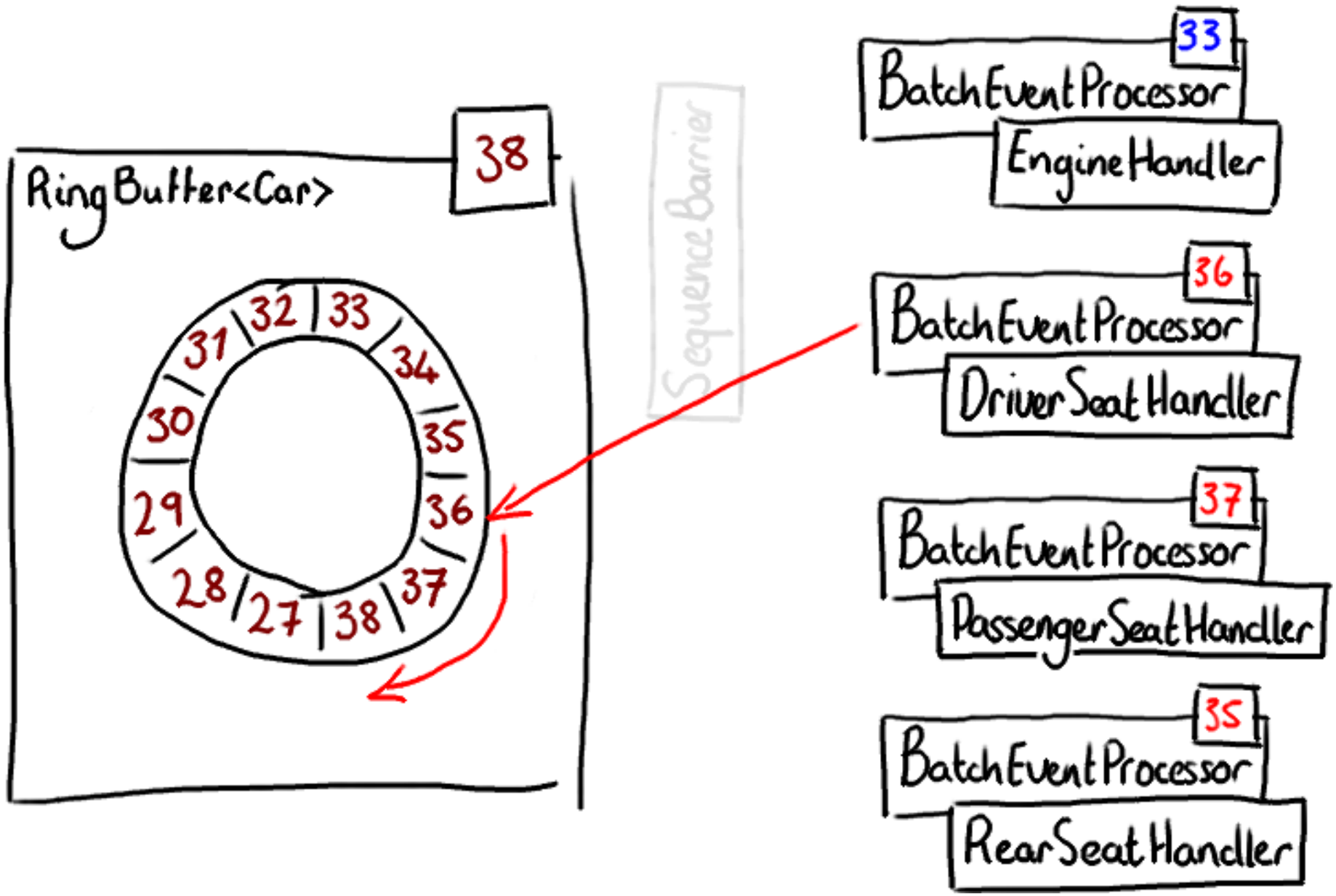

wait For (34)
33
BatchEventProcessor
EngineHandler
38
RingButter<Car>
SequenceBarrier
36
BatchEventProcessor
DriverSeatHandler
37
BatchEventProcessor
PassengerSeatHandler
35
BatchEventProcessor
RearSeatHandler
30
31
32
33
34
35
29
36
28
27
38
37

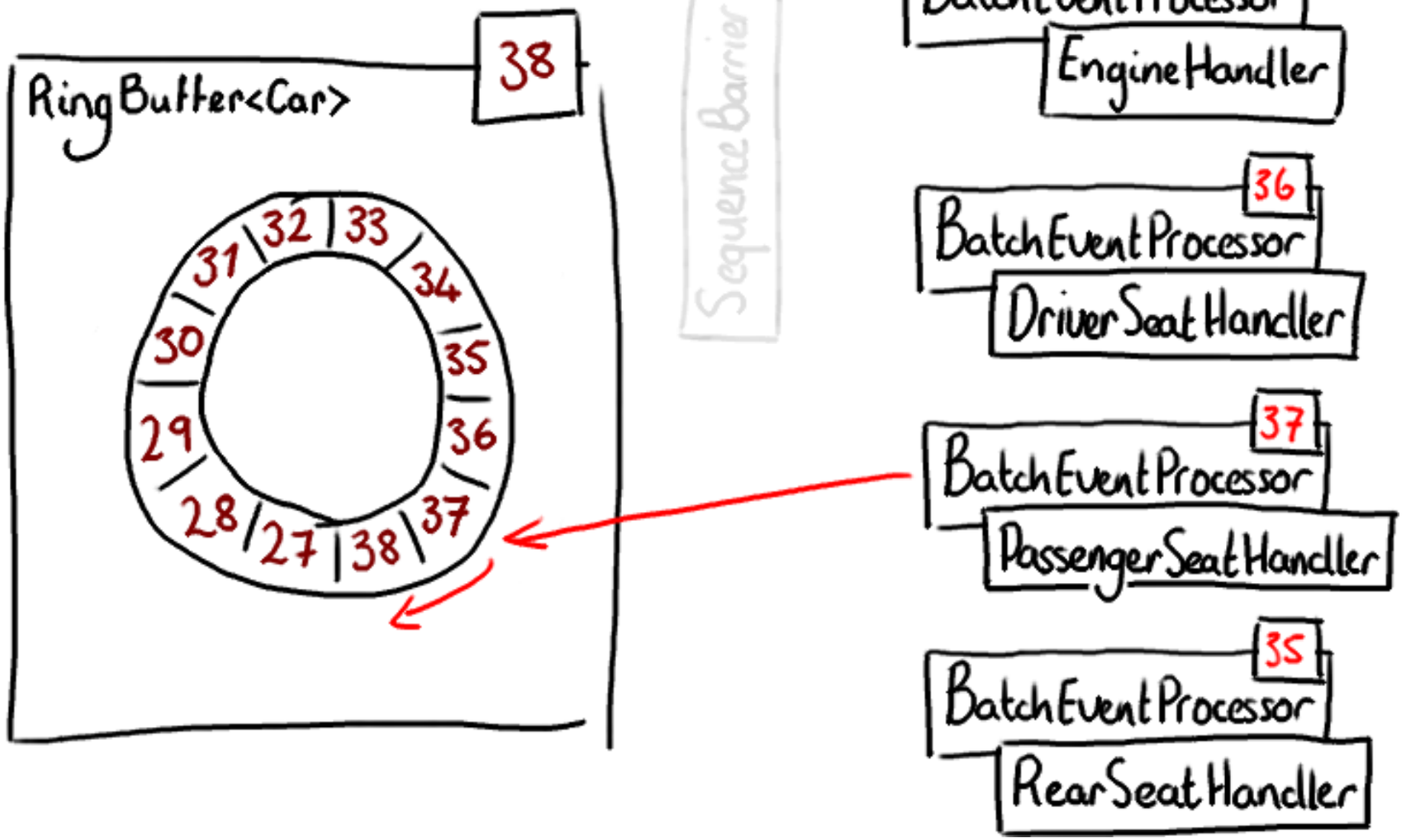

RingButter<Car>
38
30
31
32
33
34
35
36
37
29
28
27
38
SequenceBarrier
waitFor(34)
38
33
BatchEventProcessor
EngineHandler
36
BatchEventProcessor
DriverSeatHandler
37
BatchEventProcessor
PassengerSeatHandler
35
BatchEventProcessor
RearSeatHandler

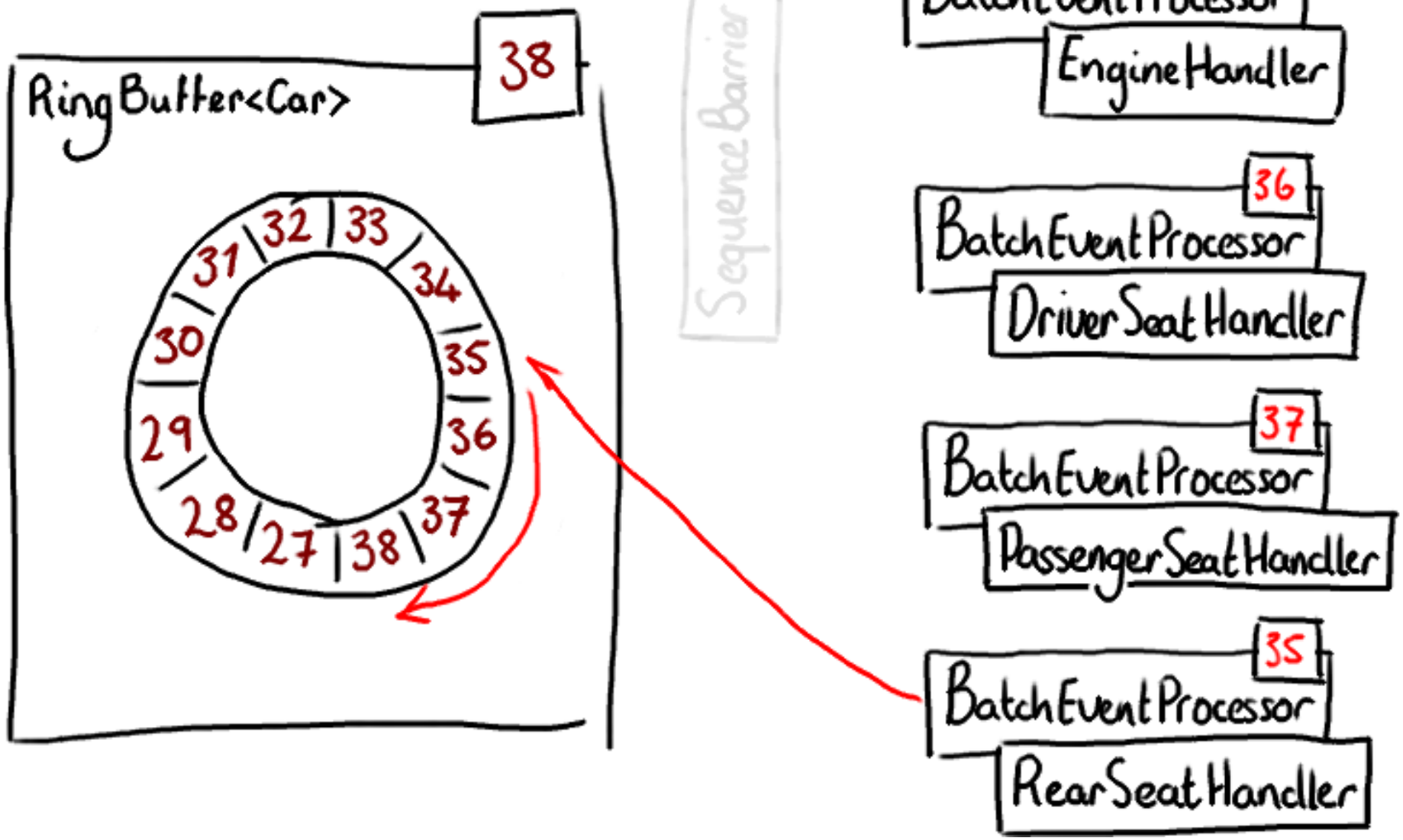

RingButter<Car>
38
31
32
33
34
30
35
29
36
28
27
38
37
Sequence Barrier
BatchEventProcessor
33
EngineHandler
BatchEventProcessor
36
DriverSeatHandler
BatchEventProcessor
37
PassengerSeatHandler
BatchEventProcessor
35
RearSeatHandler

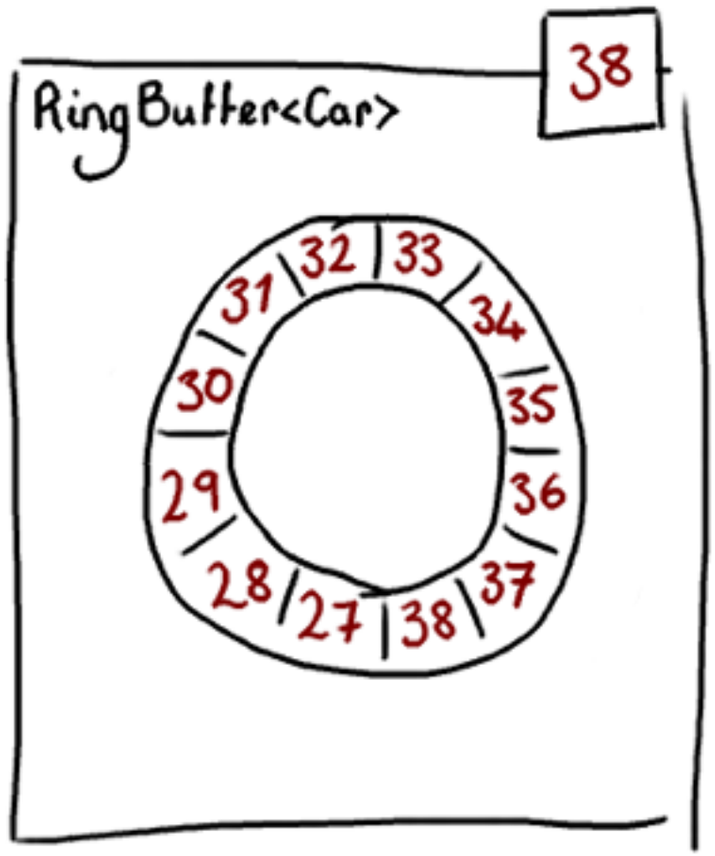

RingButter<Car>
38
30
31
32
33
34
35
36
29
28
27
38
37
SequenceBarrier
BatchEventProcessor
33
EngineHandler
waitFor(37)
BatchEventProcessor
36
DriverSeatHandler
waitFor(38)
BatchEventProcessor
37
PassengerSeatHandler
waitFor(36)
BatchEventProcessor
35
RearSeatHandler


RingButter<Car>
38
30
31
32
33
34
35
36
29
28
27
38
37
SequenceBarrier
38
BatchEventProcessor
33
EngineHandler
waitFor(37)
38
BatchEventProcessor
36
DriverSeatHandler
waitFor(38)
38
BatchEventProcessor
37
PassengerSeatHandler
waitFor(36)
38
BatchEventProcessor
35
RearSeatHandler


RingButter<Car>
38
30
31
32
33
34
35
36
29
28
27
38
37
Sequence Barrier
BatchEventProcessor
33
EngineHandler
BatchEventProcessor
36
DriverSeatHandler
BatchEventProcessor
37
PassengerSeatHandler
BatchEventProcessor
35
RearSeatHandler


RingButter<Car>
38
30
31
32
33
34
35
29
36
28
27
38
37
Sequence Barrier
33
BatchEventProcessor
EngineHandler
36
BatchEventProcessor
DriverSeatHandler
37
BatchEventProcessor
PassengerSeatHandler
35
BatchEventProcessor
RearSeatHandler


RingButter<Car>
38
30
31
32
33
34
35
29
36
28
27
38
37
Sequence Barrier
33
BatchEventProcessor
EngineHandler
36
BatchEventProcessor
DriverSeatHandler
37
BatchEventProcessor
PassengerSeatHandler
35
BatchEventProcessor
RearSeatHandler



Sequence Barrier

BatchEventProcessor 33
EngineHandler

BatchEventProcessor 36
DriverSeatHandler

BatchEventProcessor 37
PassengerSeatHandler

BatchEventProcessor 35
RearSeatHandler

Sequence Barrier

Sequence Barrier

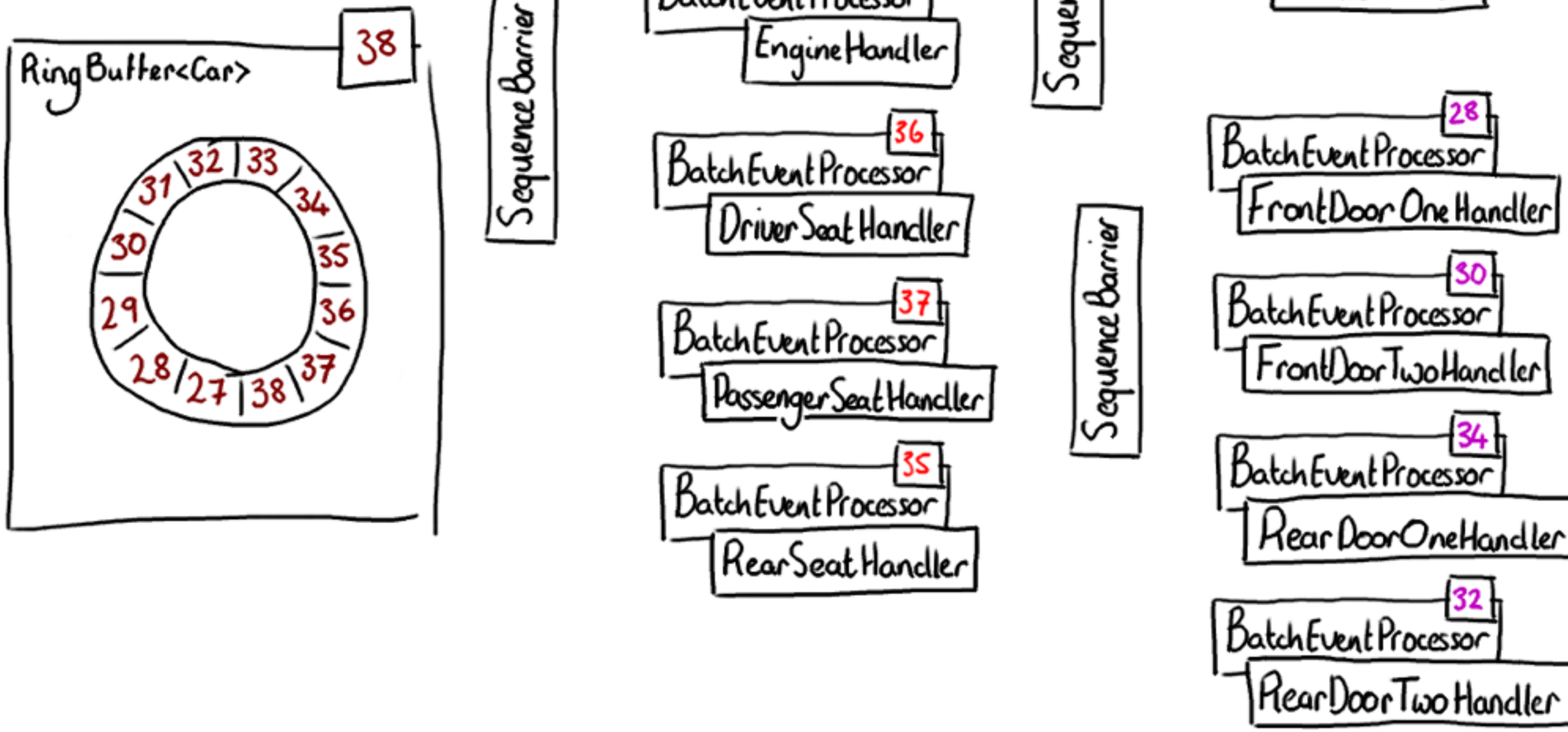
BatchEventProcessor 32
BonnetHandler

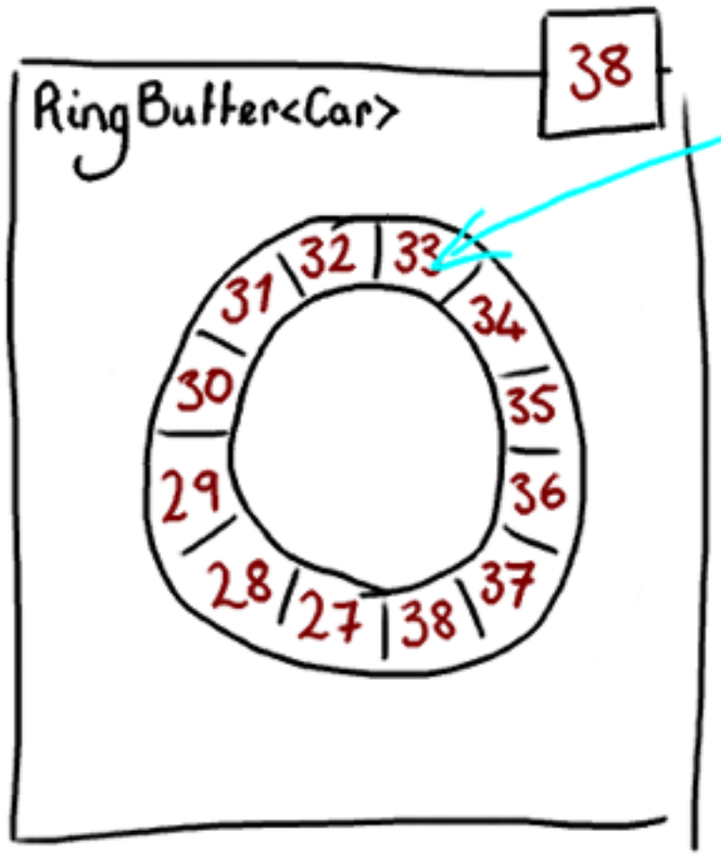
BatchEventProcessor 28
FrontDoorOneHandler

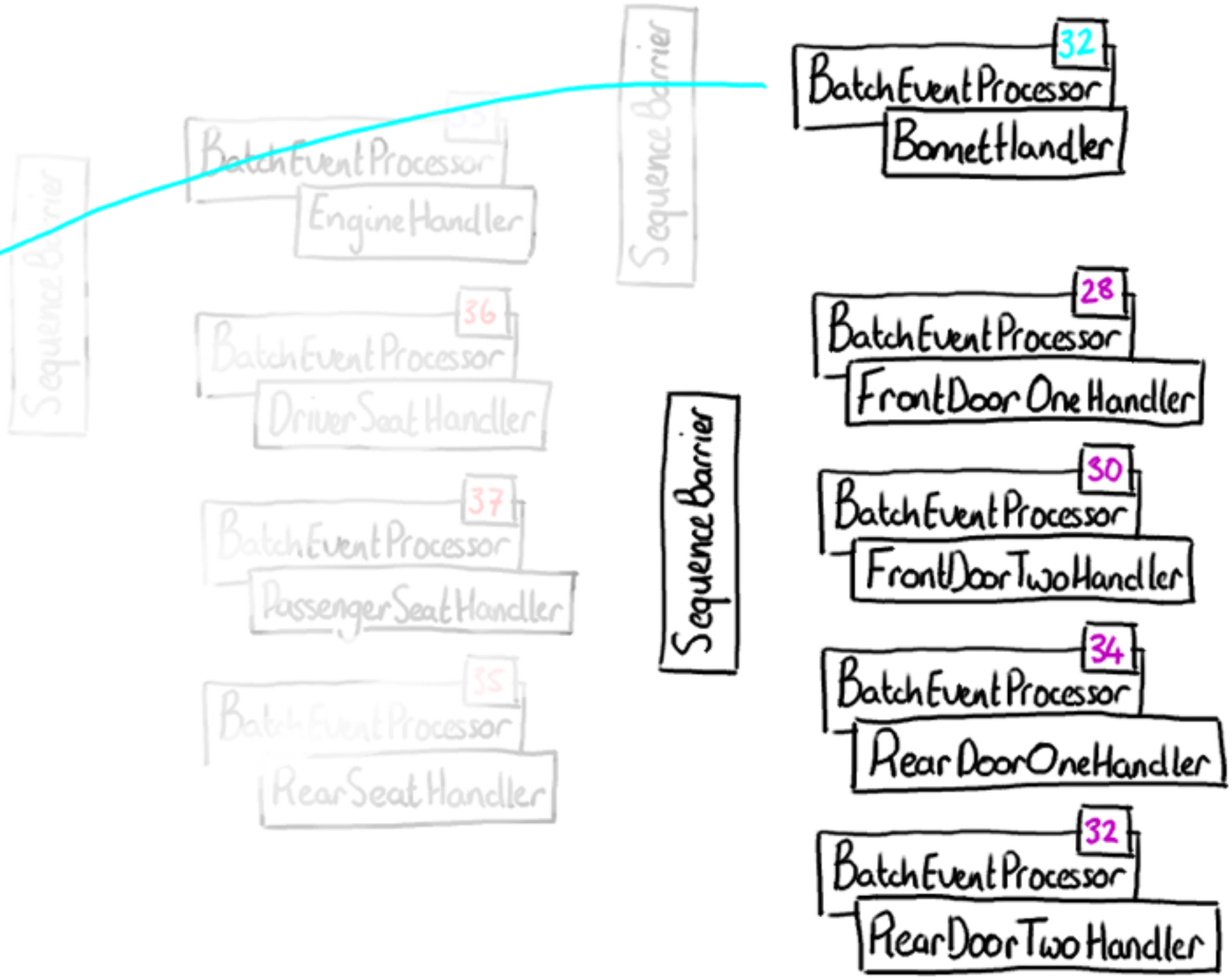
BatchEventProcessor 30
FrontDoorTwoHandler

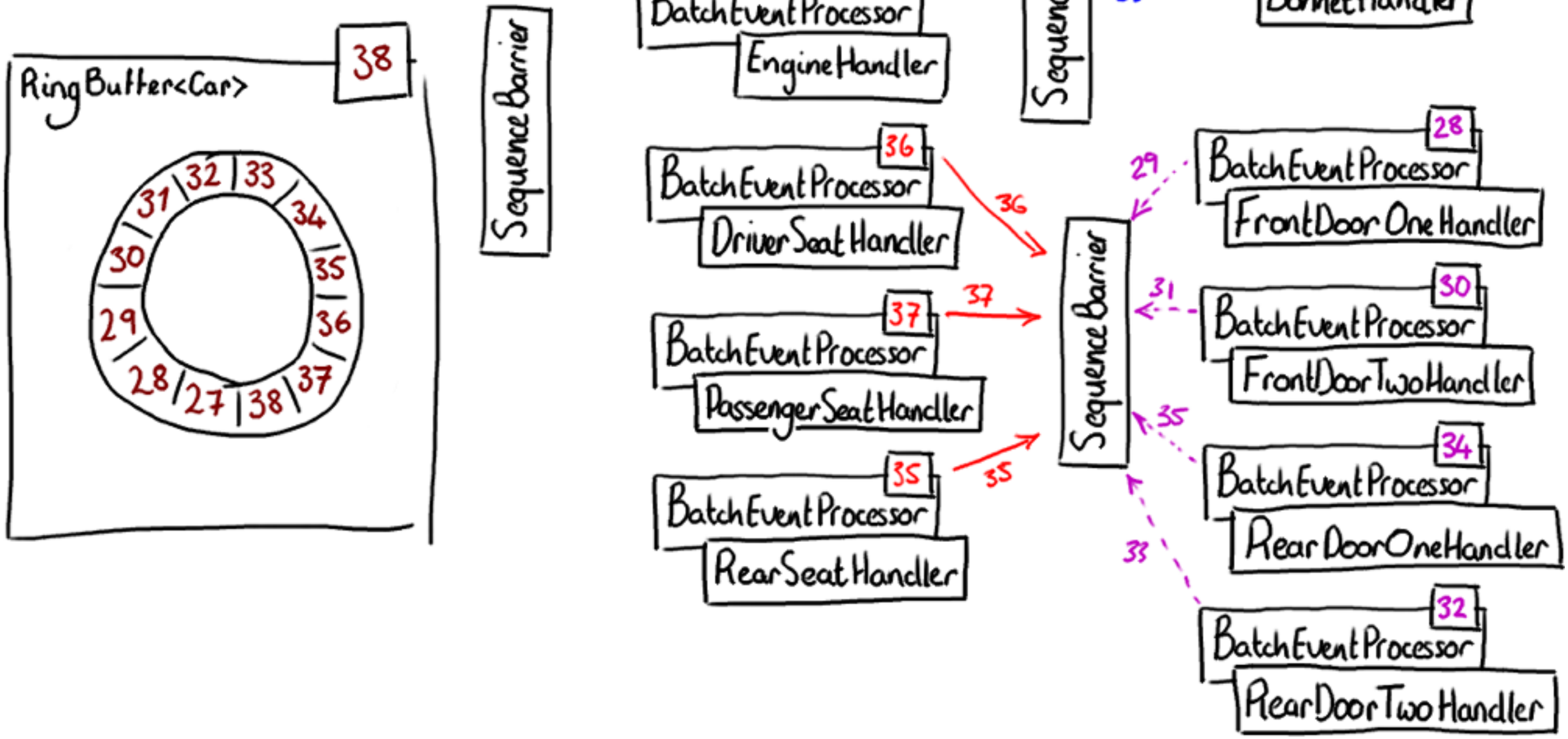
BatchEventProcessor 34
RearDoorOneHandler

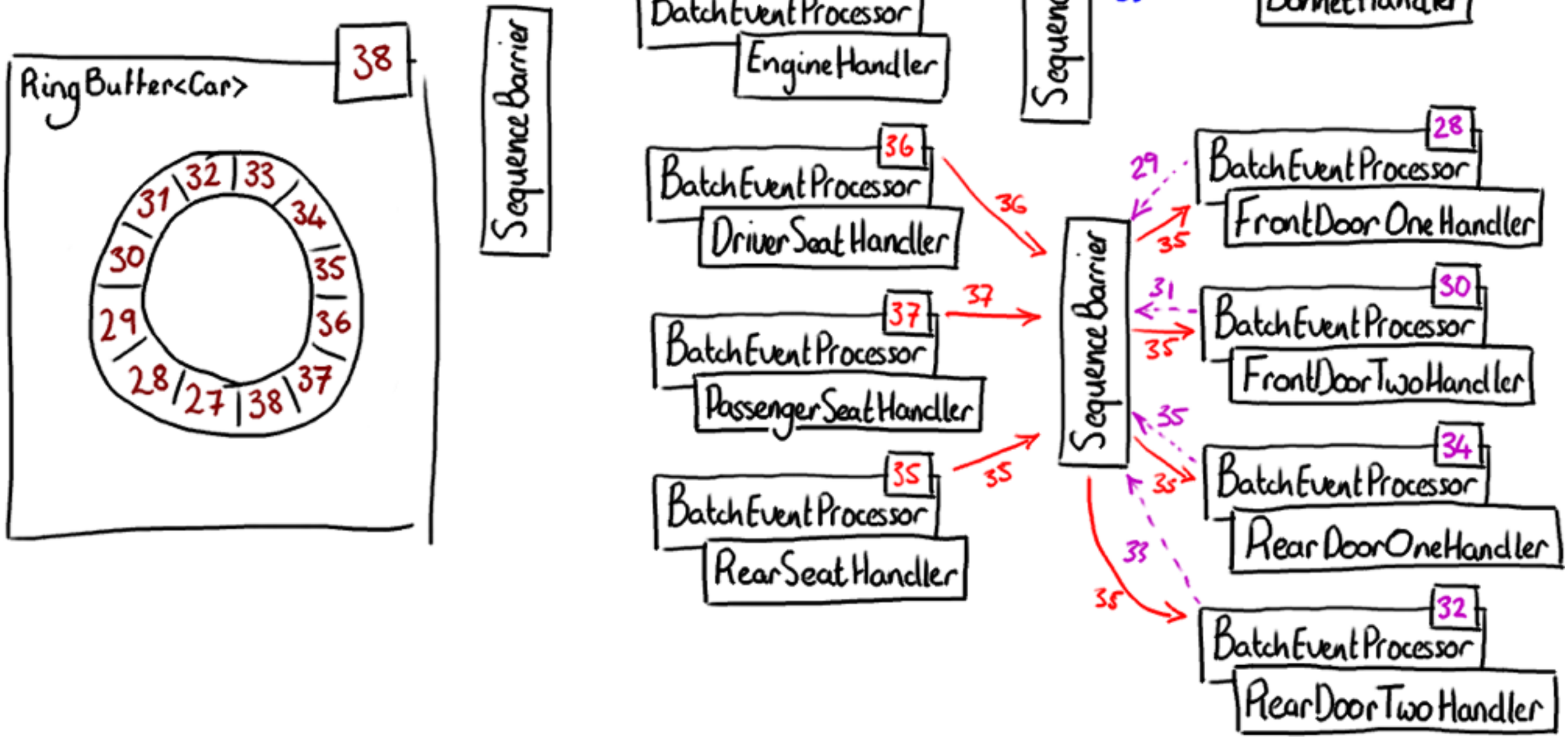
BatchEventProcessor 32
RearDoorTwoHandler

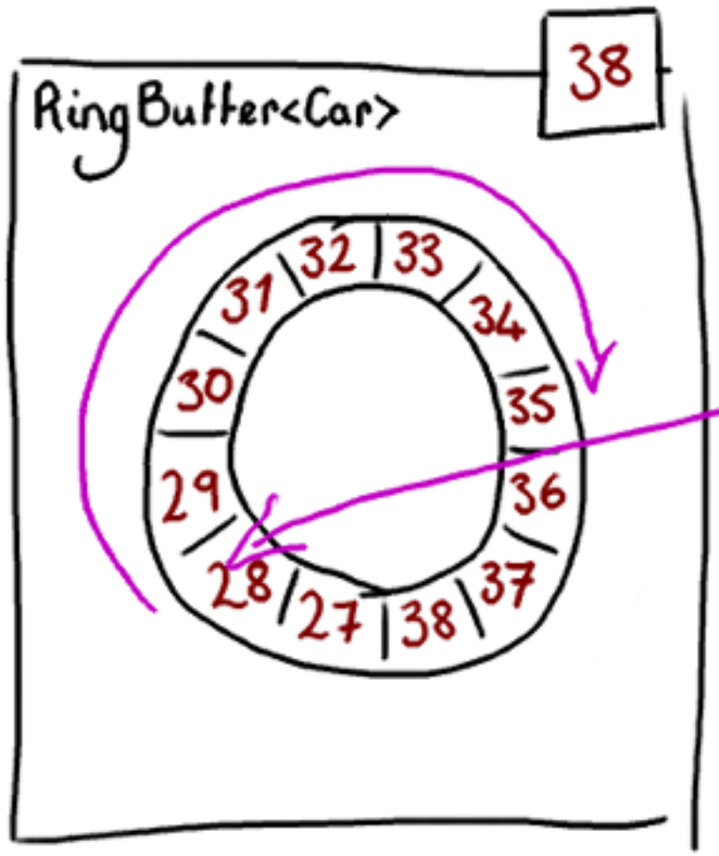

RingButter<Car>
38
30
31
32
33
34
35
29
36
28
27
38
37
Sequence Barrier
BatchEventProcessor
EngineHandler
33
Sequence Barrier
33
33
BatchEventProcessor
BonnetHandler
32
BatchEventProcessor
DriverSeatHandler
36
BatchEventProcessor
PassengerSeatHandler
37
BatchEventProcessor
RearSeatHandler
35
Sequence Barrier
BatchEventProcessor
FrontDoorOneHandler
28
BatchEventProcessor
FrontDoorTwoHandler
30
BatchEventProcessor
RearDoorOneHandler
34
BatchEventProcessor
RearDoorTwoHandler
32


RingButter<Car>
38
32
33
31
34
30
35
29
36
28
27
38
37


SequenceBarrier
BatchEventProcessor
EngineHandler
SequenceBarrier
BatchEventProcessor
DriverSeatHandler
BatchEventProcessor
PassengerSeatHandler
BatchEventProcessor
RearSeatHandler
SequenceBarrier
32
BatchEventProcessor
BonnetHandler
28
BatchEventProcessor
FrontDoorOneHandler
30
BatchEventProcessor
FrontDoorTwoHandler
34
BatchEventProcessor
RearDoorOneHandler
32
BatchEventProcessor
RearDoorTwoHandler


RingBuffer<Car>
38
30
31
32
33
34
35
36
37
38
27
28
29
SequenceBarrier
BatchEventProcessor
EngineHandler
33
SequenceBarrier
33
33
BatchEventProcessor
BonnetHandler
32
BatchEventProcessor
DriverSeatHandler
36
36
BatchEventProcessor
FrontDoorOneHandler
28
29
SequenceBarrier
37
37
BatchEventProcessor
PassengerSeatHandler
31
BatchEventProcessor
FrontDoorTwoHandler
30
35
35
BatchEventProcessor
RearSeatHandler
35
BatchEventProcessor
RearDoorOneHandler
34
33
BatchEventProcessor
RearDoorTwoHandler
32


RingButter<Car>
38
30
31
32
33
34
35
36
37
38
27
28
29
Sequence Barrier
BatchEventProcessor
EngineHandler
33
Sequence Barrier
33
33
BatchEventProcessor
BonnetHandler
32
BatchEventProcessor
DriverSeatHandler
36
36
BatchEventProcessor
FrontDoorOneHandler
28
29
35
Sequence Barrier
BatchEventProcessor
PassengerSeatHandler
37
37
31
35
BatchEventProcessor
FrontDoorTwoHandler
30
BatchEventProcessor
RearSeatHandler
35
35
35
35
BatchEventProcessor
RearDoorOneHandler
34
33
35
BatchEventProcessor
RearDoorTwoHandler
32


RingButter<Car>
38
32
33
31
34
30
35
29
36
28
37
27
38

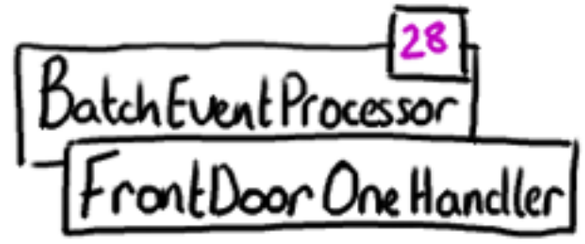

BatchEventProcessor
33
EngineHandler
BatchEventProcessor
36
DriverSeatHandler
BatchEventProcessor
37
PassengerSeatHandler
BatchEventProcessor
35
RearSeatHandler

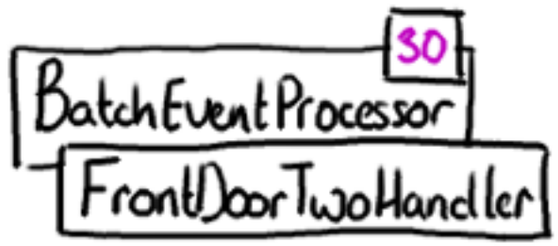

SequenceBarrier

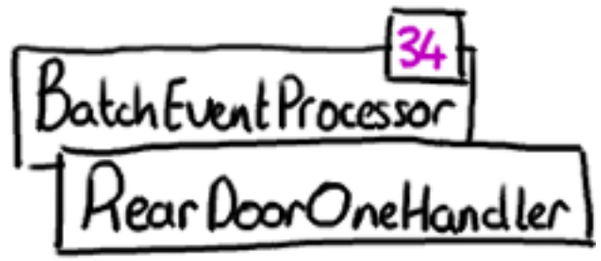

SequenceBarrier

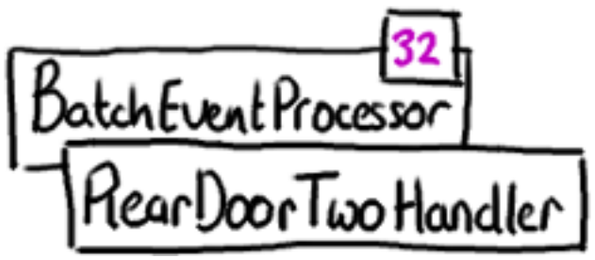
SequenceBarrier

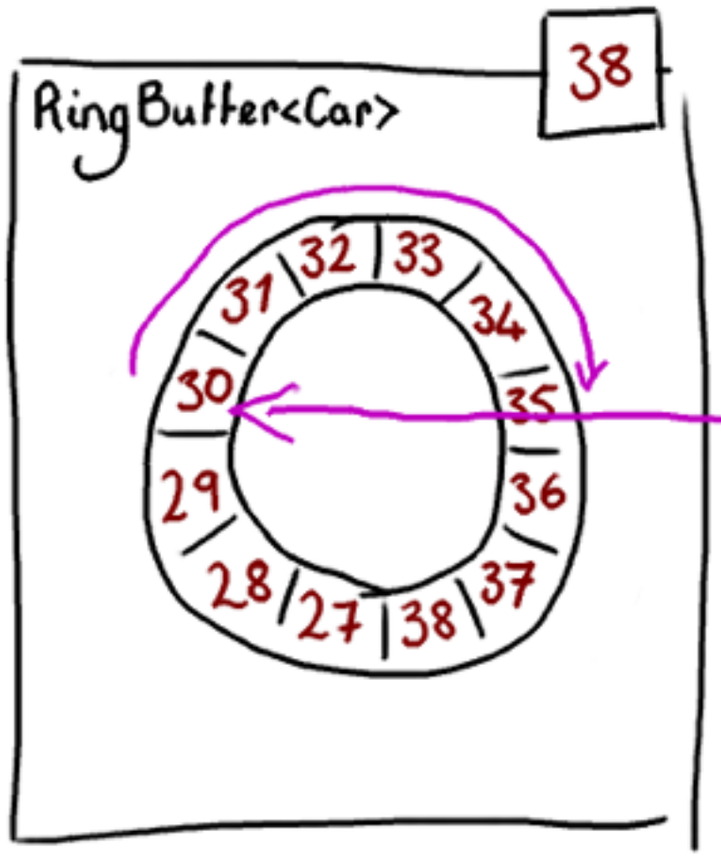

32
BatchEventProcessor
BonnetHandler

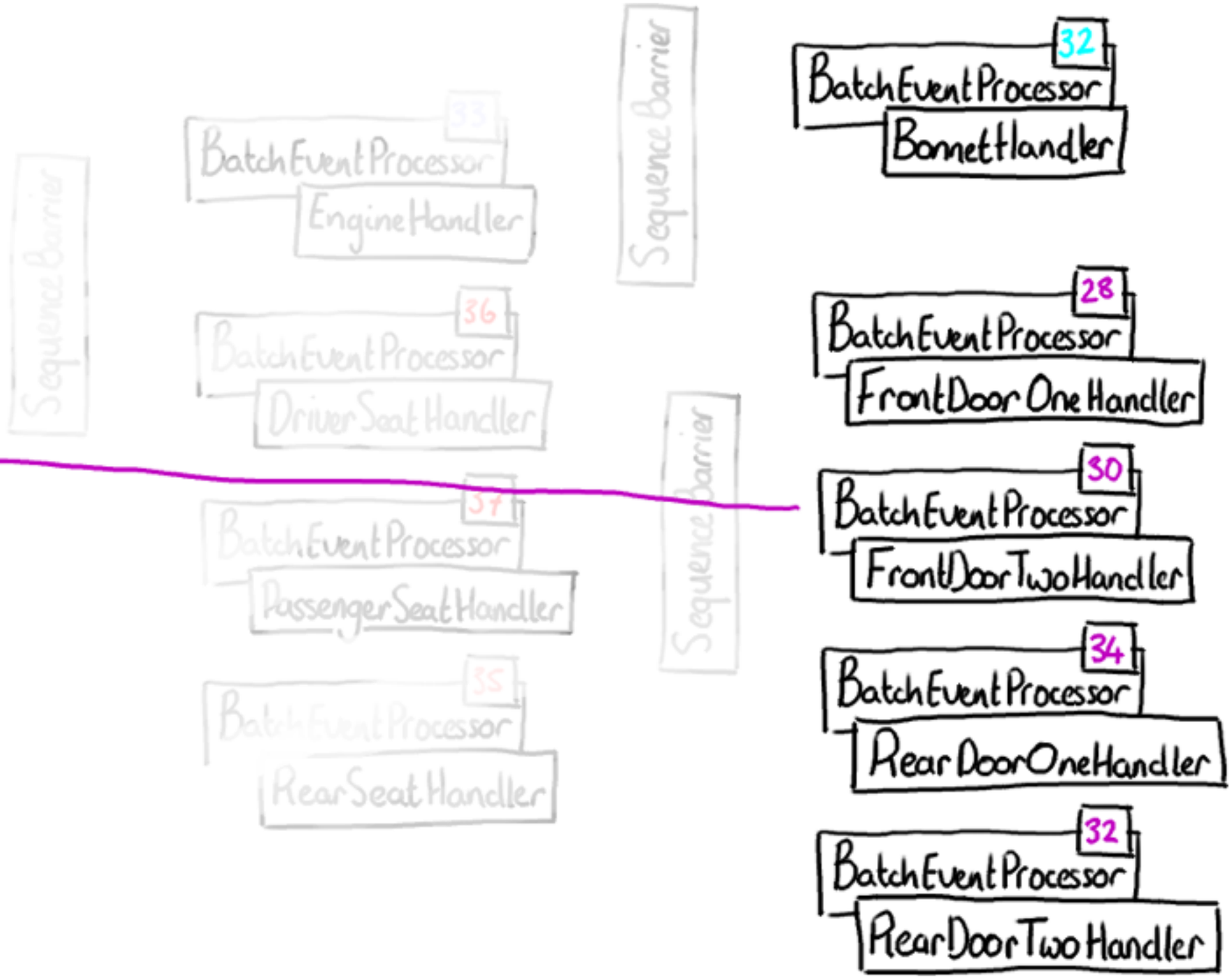

28
BatchEventProcessor
FrontDoorOneHandler


30
BatchEventProcessor
FrontDoorTwoHandler

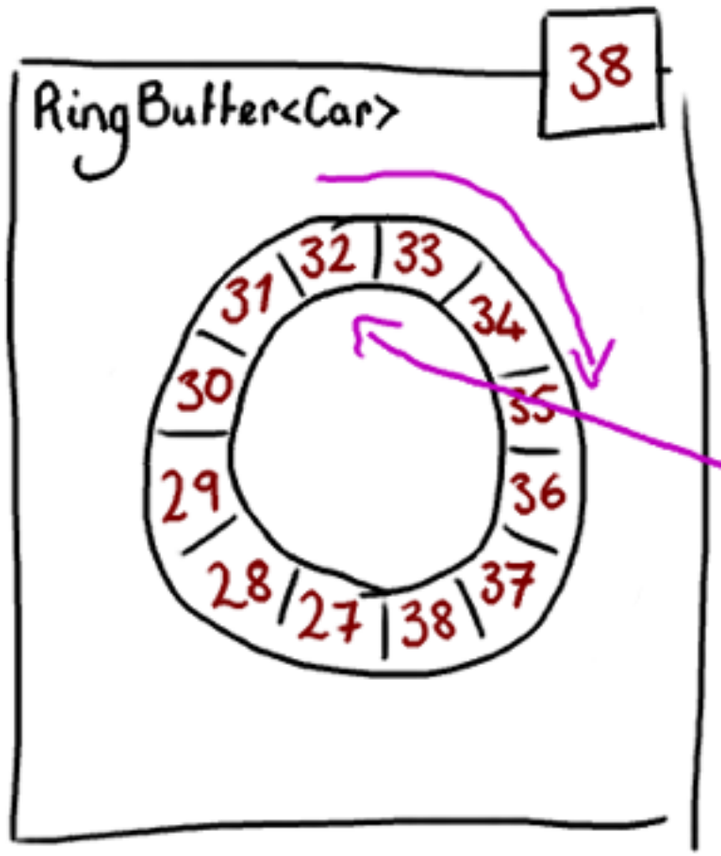

34
BatchEventProcessor
RearDoorOneHandler

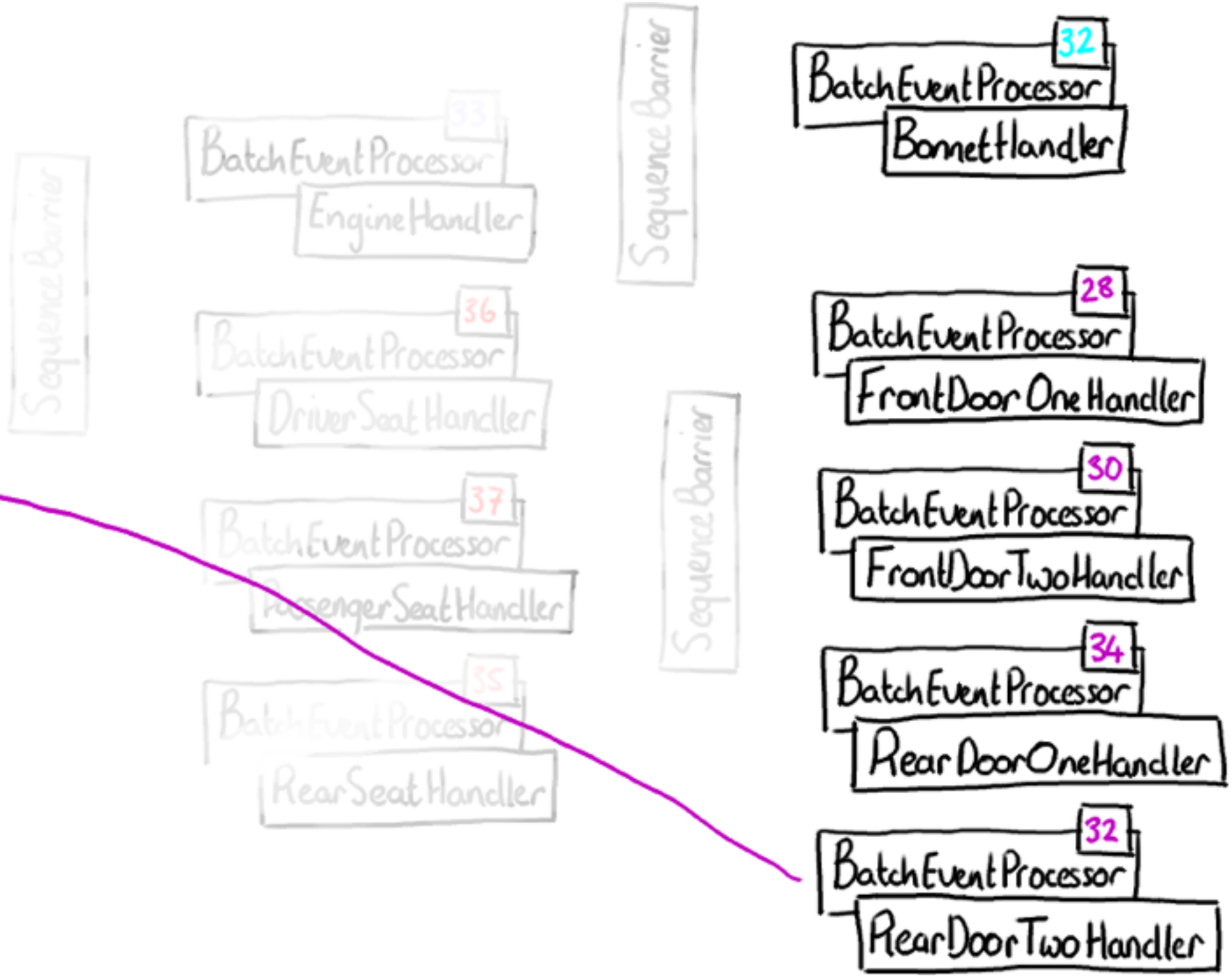

32
BatchEventProcessor
RearDoorTwoHandler

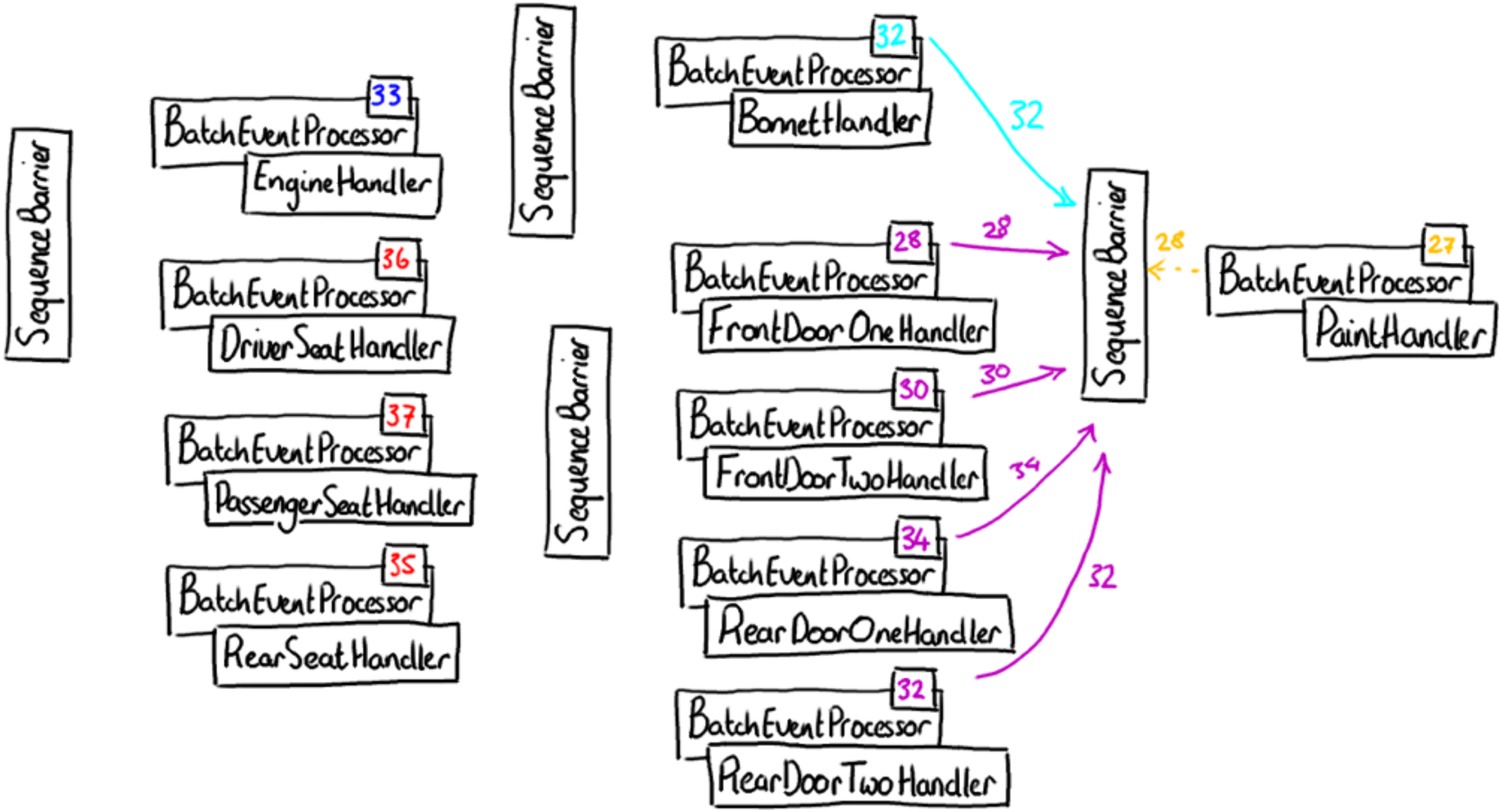

RingButter<Car>
38
32
33
31
34
30
35
29
36
28
27
38
37

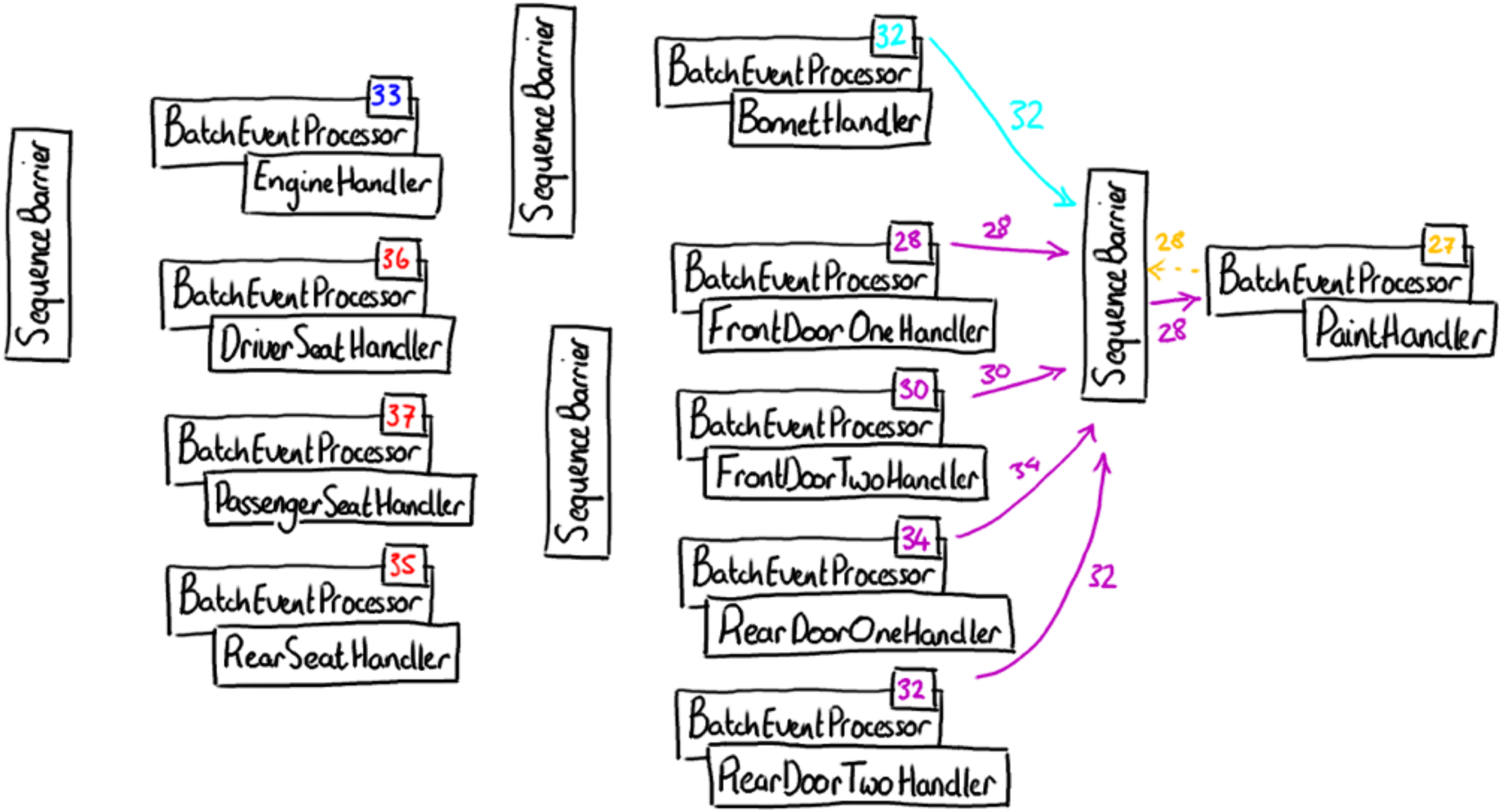

Sequence Barrier
BatchEventProcessor
33
EngineHandler
Sequence Barrier
BatchEventProcessor
32
BonnetHandler
BatchEventProcessor
36
DriverSeatHandler
BatchEventProcessor
28
FrontDoorOneHandler
BatchEventProcessor
37
PassengerSeatHandler
Sequence Barrier
BatchEventProcessor
30
FrontDoorTwoHandler
BatchEventProcessor
35
RearSeatHandler
BatchEventProcessor
34
RearDoorOneHandler
BatchEventProcessor
32
RearDoorTwoHandler

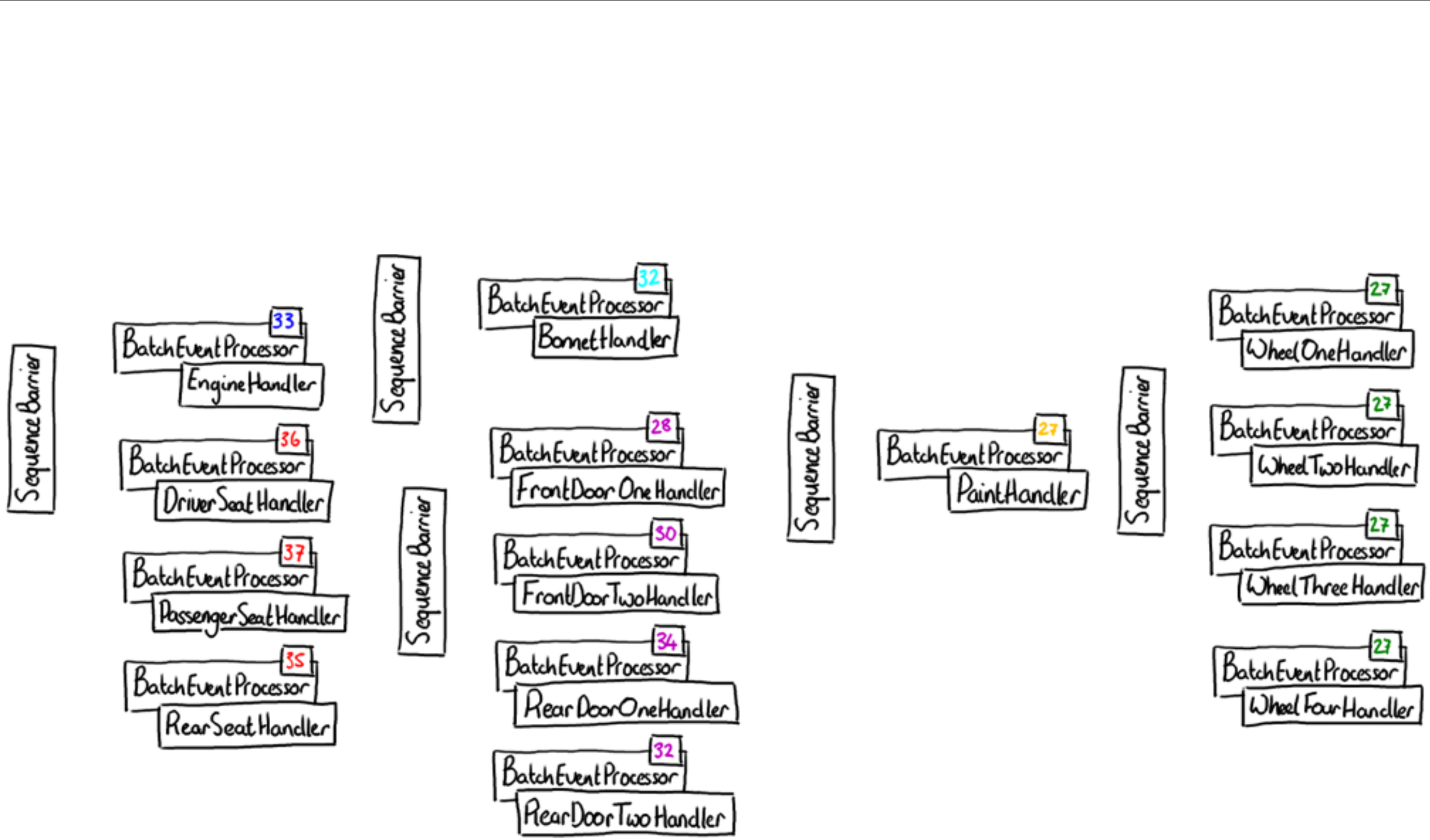

RingButter<Car>
38
32
33
34
31
30
35
29
36
28
27
38
37
Sequence Barrier
BatchEventProcessor
33
EngineHandler
BatchEventProcessor
36
DriverSeatHandler
BatchEventProcessor
37
PassengerSeatHandler
BatchEventProcessor
35
RearSeatHandler
Sequence Barrier
BatchEventProcessor
32
BonnetHandler
BatchEventProcessor
28
FrontDoorOneHandler
BatchEventProcessor
30
FrontDoorTwoHandler
BatchEventProcessor
34
RearDoorOneHandler
BatchEventProcessor
32
RearDoorTwoHandler

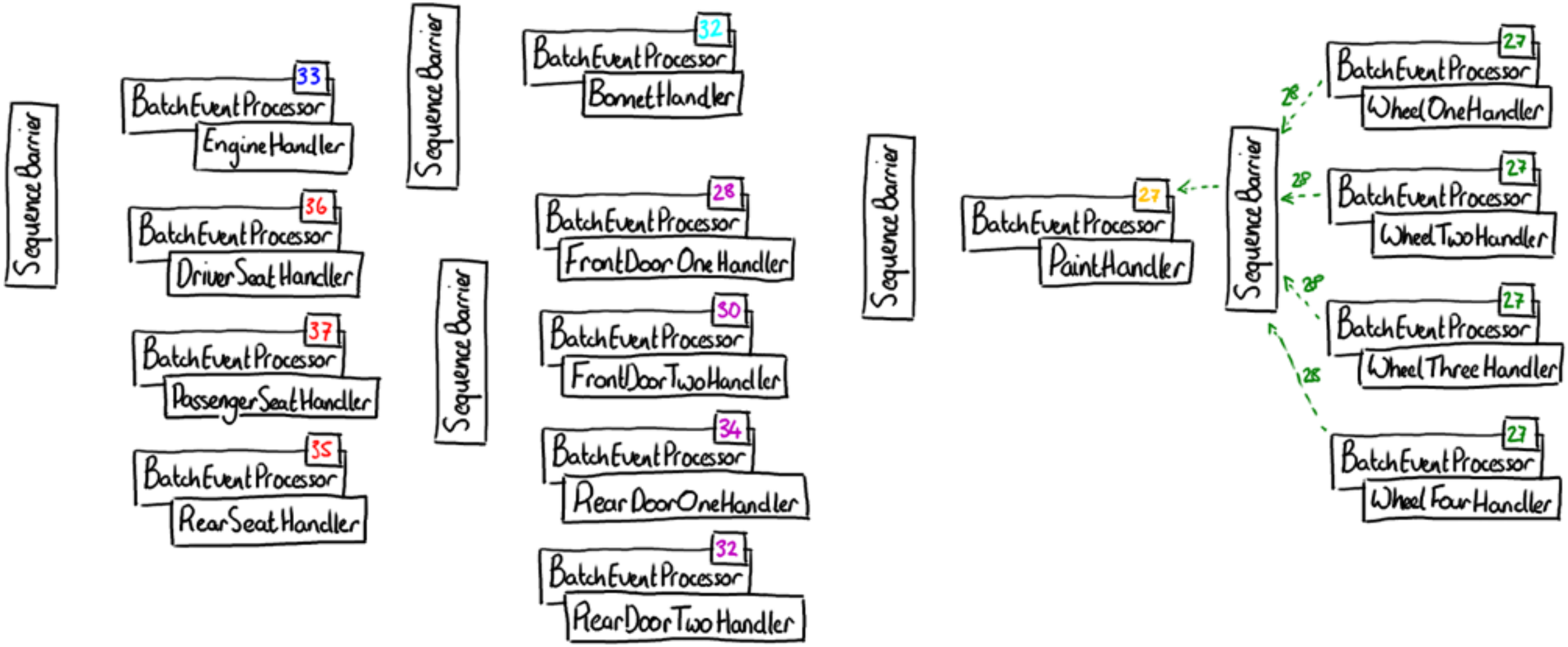

RingButter<Car>
38
32
33
31
34
30
35
29
36
28
27
38
37

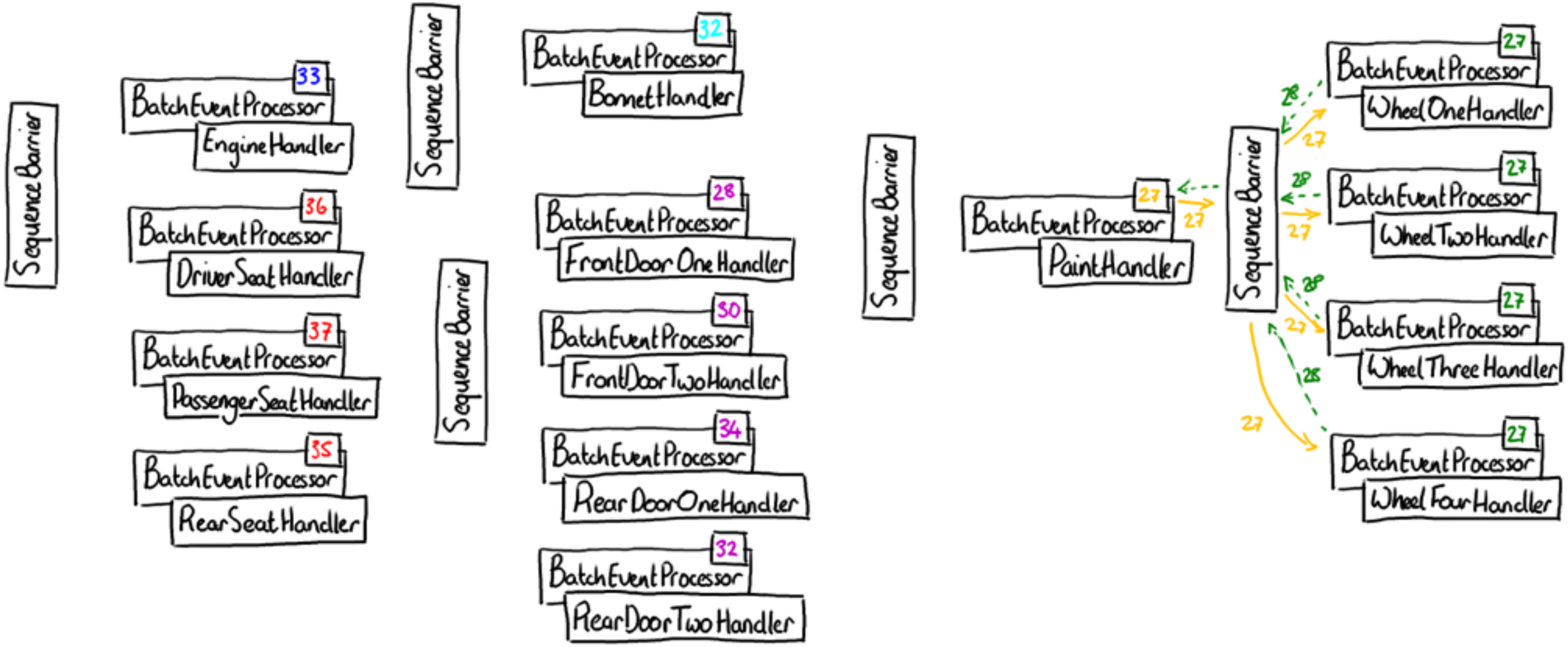

SequenceBarrier
BatchEventProcessor
33
EngineHandler
SequenceBarrier
BatchEventProcessor
32
BonnetHandler
BatchEventProcessor
36
DriverSeatHandler
BatchEventProcessor
28
FrontDoorOneHandler
SequenceBarrier
BatchEventProcessor
37
PassengerSeatHandler
BatchEventProcessor
30
FrontDoorTwoHandler
BatchEventProcessor
35
RearSeatHandler
BatchEventProcessor
34
RearDoorOneHandler
BatchEventProcessor
32
RearDoorTwoHandler

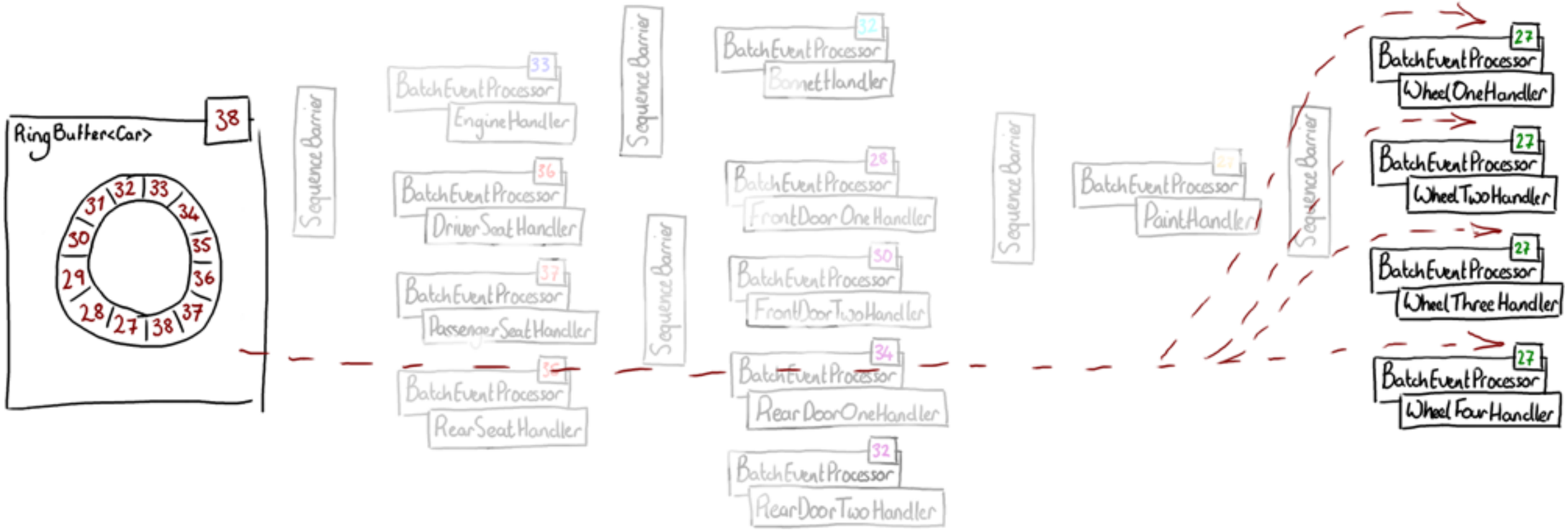

Sequence Barrier
33
BatchEventProcessor
EngineHandler
36
BatchEventProcessor
DriverSeatHandler
37
BatchEventProcessor
PassengerSeatHandler
35
BatchEventProcessor
RearSeatHandler
Sequence Barrier
Sequence Barrier
32
BatchEventProcessor
BonnetHandler
32
28
28
BatchEventProcessor
FrontDoorOneHandler
30
30
BatchEventProcessor
FrontDoorTwoHandler
34
34
BatchEventProcessor
RearDoorOneHandler
32
32
BatchEventProcessor
RearDoorTwoHandler
Sequence Barrier
28
27
BatchEventProcessor
PaintHandler


Sequence Barrier
33
BatchEventProcessor
EngineHandler
36
BatchEventProcessor
DriverSeatHandler
37
BatchEventProcessor
PassengerSeatHandler
35
BatchEventProcessor
RearSeatHandler
Sequence Barrier
Sequence Barrier
32
BatchEventProcessor
BonnetHandler
32
28
28
BatchEventProcessor
FrontDoorOneHandler
30
BatchEventProcessor
FrontDoorTwoHandler
34
34
BatchEventProcessor
RearDoorOneHandler
32
32
BatchEventProcessor
RearDoorTwoHandler
Sequence Barrier
28
28
27
BatchEventProcessor
PaintHandler


Sequence Barrier
BatchEventProcessor
EngineHandler
33
BatchEventProcessor
DriverSeatHandler
36
BatchEventProcessor
PassengerSeatHandler
37
BatchEventProcessor
RearSeatHandler
35
Sequence Barrier
Sequence Barrier
BatchEventProcessor
BonnetHandler
32
BatchEventProcessor
FrontDoorOneHandler
28
BatchEventProcessor
FrontDoorTwoHandler
30
BatchEventProcessor
RearDoorOneHandler
34
BatchEventProcessor
RearDoorTwoHandler
32
Sequence Barrier
BatchEventProcessor
PaintHandler
27
Sequence Barrier
BatchEventProcessor
WheelOneHandler
27
BatchEventProcessor
WheelTwoHandler
27
BatchEventProcessor
WheelThreeHandler
27
BatchEventProcessor
WheelFourHandler
27


Sequence Barrier
BatchEventProcessor
EngineHandler
33
BatchEventProcessor
DriverSeatHandler
36
BatchEventProcessor
PassengerSeatHandler
37
BatchEventProcessor
RearSeatHandler
35
Sequence Barrier
Sequence Barrier
BatchEventProcessor
BonnetHandler
32
BatchEventProcessor
FrontDoorOneHandler
28
BatchEventProcessor
FrontDoorTwoHandler
30
BatchEventProcessor
RearDoorOneHandler
34
BatchEventProcessor
RearDoorTwoHandler
32
Sequence Barrier
BatchEventProcessor
PaintHandler
27
Sequence Barrier
28
28
28
28
BatchEventProcessor
WheelOneHandler
27
BatchEventProcessor
WheelTwoHandler
27
BatchEventProcessor
WheelThreeHandler
27
BatchEventProcessor
WheelFourHandler
27


Sequence Barrier
BatchEventProcessor
EngineHandler
33
BatchEventProcessor
DriverSeatHandler
36
BatchEventProcessor
PassengerSeatHandler
37
BatchEventProcessor
RearSeatHandler
35
Sequence Barrier
Sequence Barrier
BatchEventProcessor
BonnetHandler
32
BatchEventProcessor
FrontDoorOneHandler
28
BatchEventProcessor
FrontDoorTwoHandler
30
BatchEventProcessor
RearDoorOneHandler
34
BatchEventProcessor
RearDoorTwoHandler
32
Sequence Barrier
BatchEventProcessor
PaintHandler
27
Sequence Barrier
BatchEventProcessor
WheelOneHandler
27
BatchEventProcessor
WheelTwoHandler
27
BatchEventProcessor
WheelThreeHandler
27
BatchEventProcessor
WheelFourHandler
27
28
27


RingBuffer<Car>
38
31
32
33
34
30
35
29
36
28
27
38
37
Sequence Barrier
BatchEventProcessor
33
EngineHandler
BatchEventProcessor
36
DriverSeatHandler
BatchEventProcessor
37
PassengerSeatHandler
BatchEventProcessor
35
RearSeatHandler
Sequence Barrier
BatchEventProcessor
32
BonnetHandler
BatchEventProcessor
28
FrontDoorOneHandler
BatchEventProcessor
30
FrontDoorTwoHandler
BatchEventProcessor
34
RearDoorOneHandler
BatchEventProcessor
32
RearDoorTwoHandler
Sequence Barrier
BatchEventProcessor
27
PaintHandler
Sequence Barrier
BatchEventProcessor
27
WheelOneHandler
BatchEventProcessor
27
WheelTwoHandler
BatchEventProcessor
27
WheelThreeHandler
BatchEventProcessor
27
WheelFourHandler

Don’t wrap the buffer!

```
ringBuffer.setGatingSequences(finalEventProcessor.getSequence());
```

Caveats

- Can't run this car example at high speed with only 2 cores
- Your ring buffer needs to be bigger than 12
- Event handlers need to be on separate threads

Is that it?

- Wait strategies
- Batch publishing
- Multiple publishers
- Different EventHandlers
- The Wizard
- You don’t even need a RingBuffer...

You get...

- A framework the encourages you to model your domain
- The ability to run in parallel but single-threaded
- Nice, simple Java
- Reliable ordering
- ...and it can be very fast

...and we got...

- Better design
- Great PR
- An improved product

More Information

- Google Code Site, including Wiki
  http://code.google.com/p/disruptor/
- Blogs, e.g. mine: mechanitis.blogspot.com
- Presentations
- Google Group

Q&A