DataCollider: Effective Data-Race Detection for the Kernel

刘伟森 PB15111595
王泽凡 PB15111593

- Challenges for old methods
- Insights
- Implement
- Advantage and Disadvantage
- Result

Challenges for old methods

- require a complete knowledge and logging of all locking semantics
- Locking semantics in kernel-mode can be complicated and convoluted
  - e.g. DPCs, interrupts

Insights

- Instead of inferring if a data race could have occurred, let’s cause it to actually happen!
- Use code and data breakpoints
- Randomly selection for uniform coverage

Implement

- Sampling
- Insert code breakpoint
- Detection

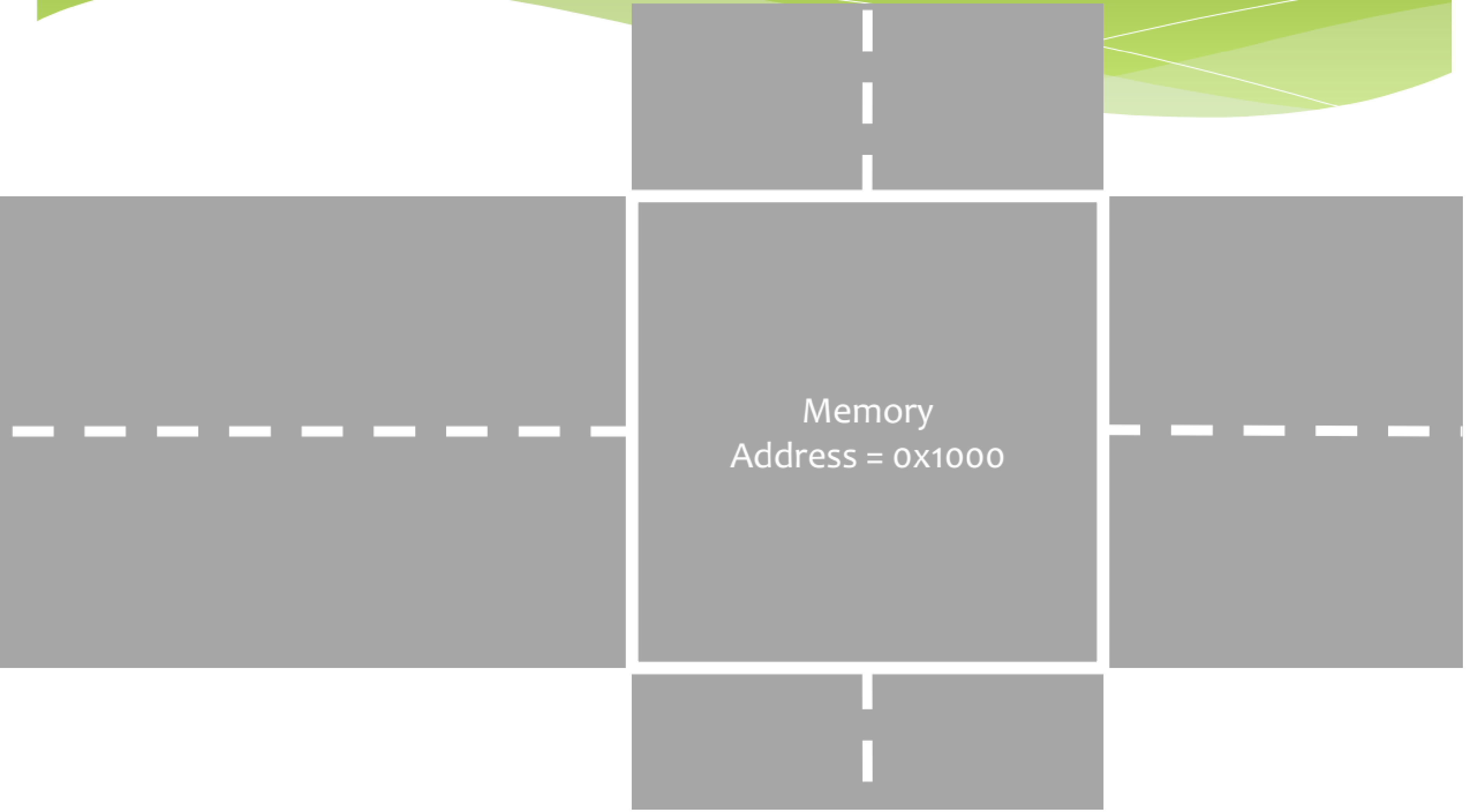
Detection

- Data breakpoint
- Repeated reads

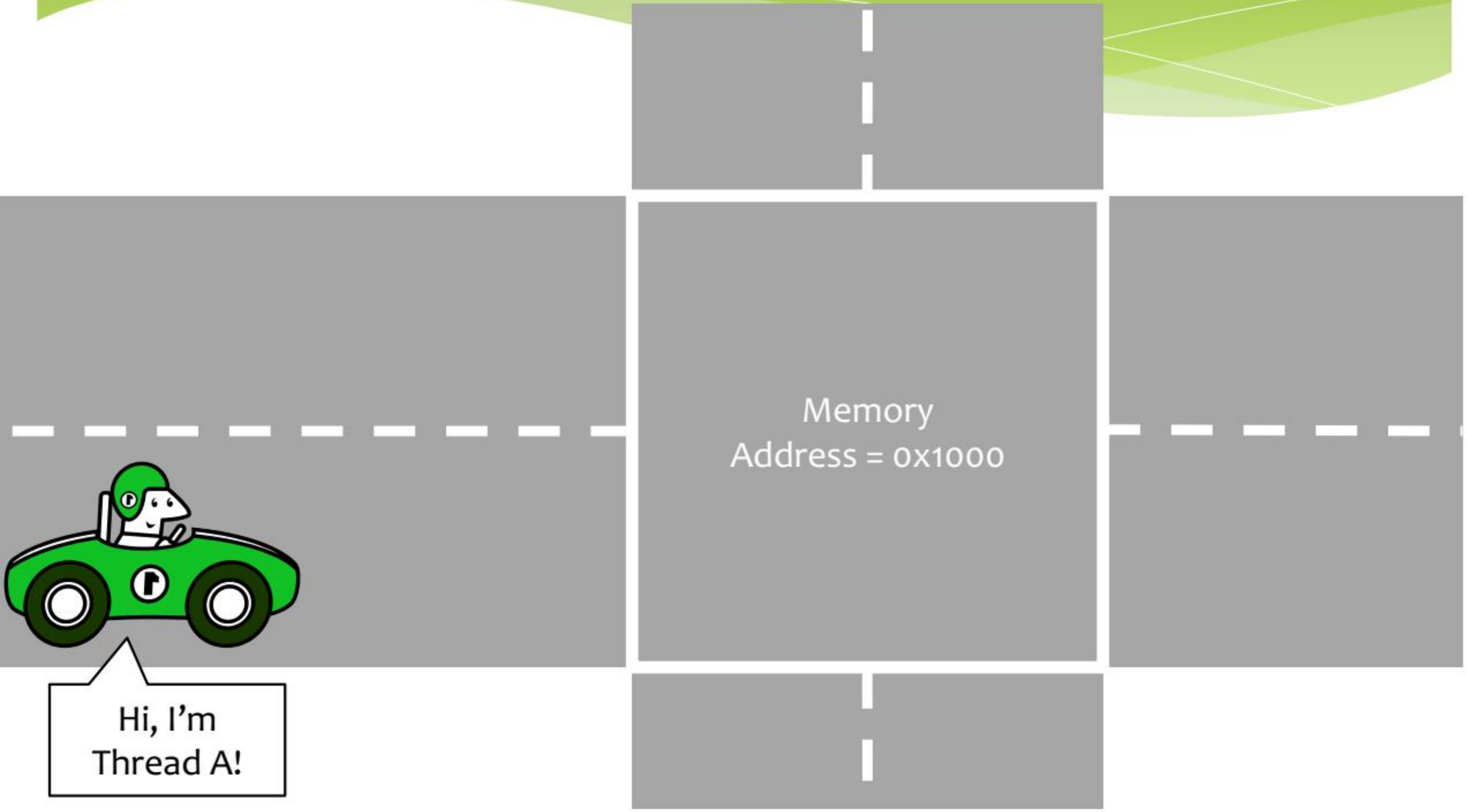
Detection

```
temp = read( loc, size );
if ( isWrite )
  SetDataBreakpointRW( loc, size );
else
  SetDataBreakpointW( loc, size );

delay();

ClearDataBreakpoint( loc, size );

temp’ = read( loc, size );
if(temp != temp’ || data breakpoint hit)
   ReportDataRace( );
```

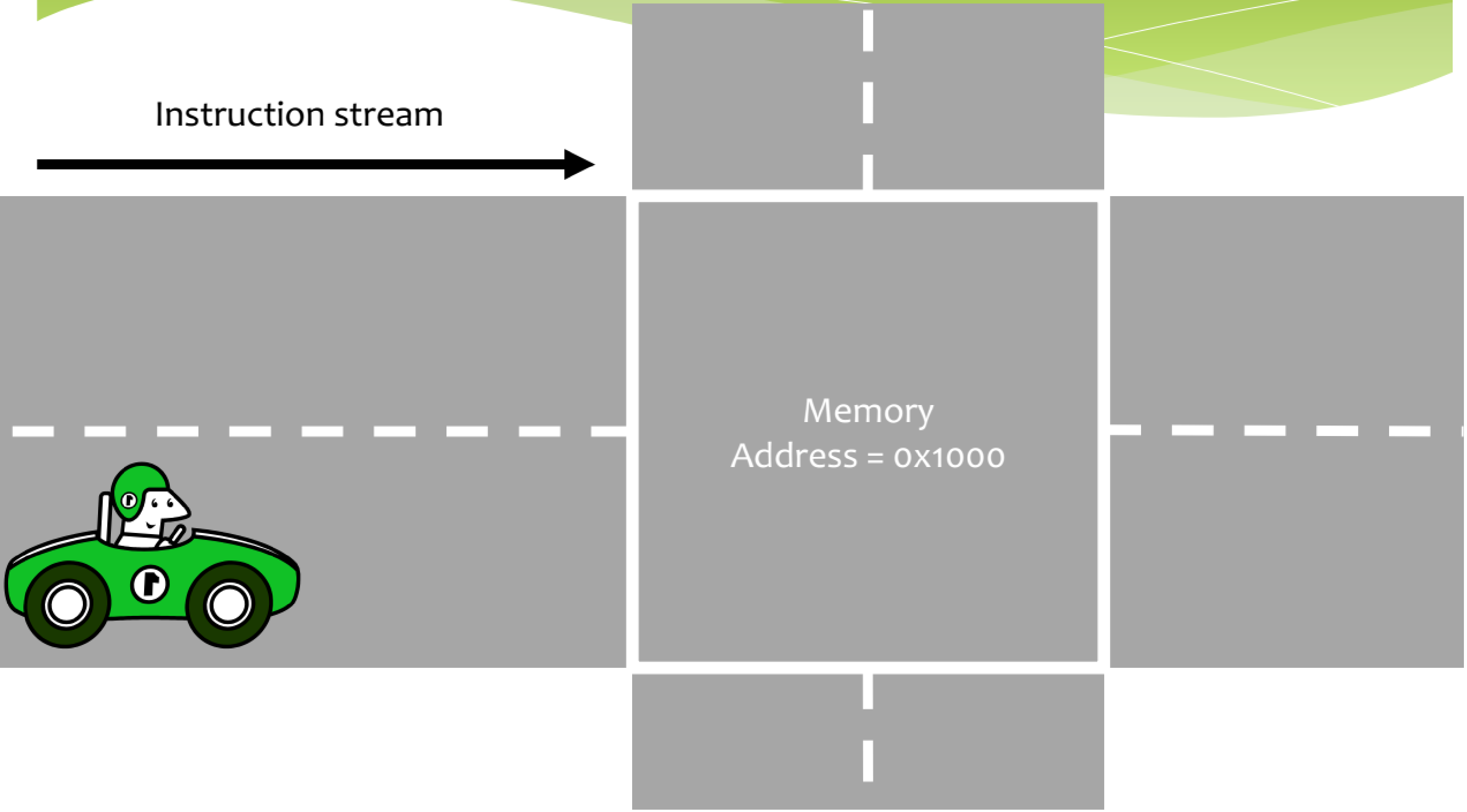
Intersection Metaphor



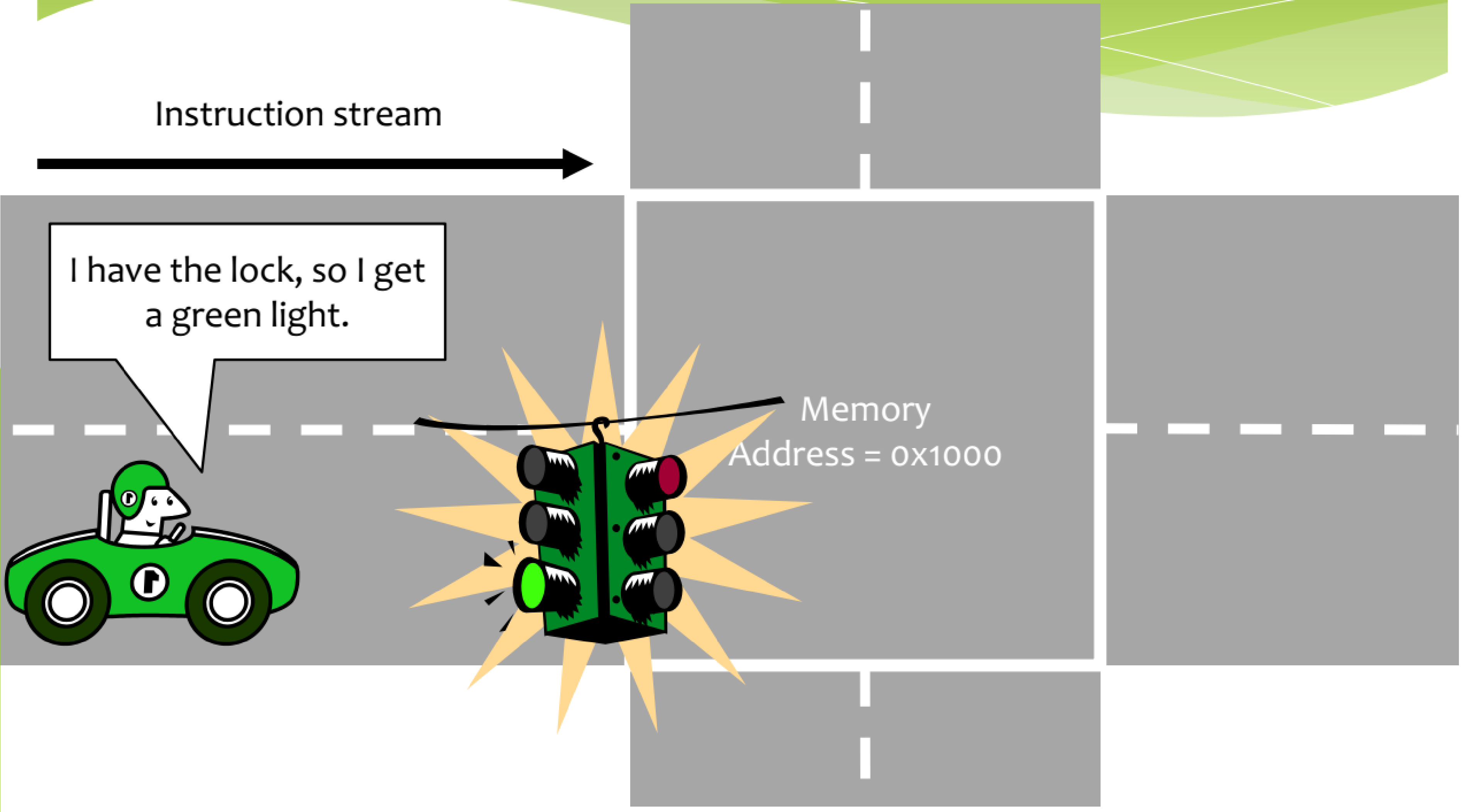
Intersection Metaphor



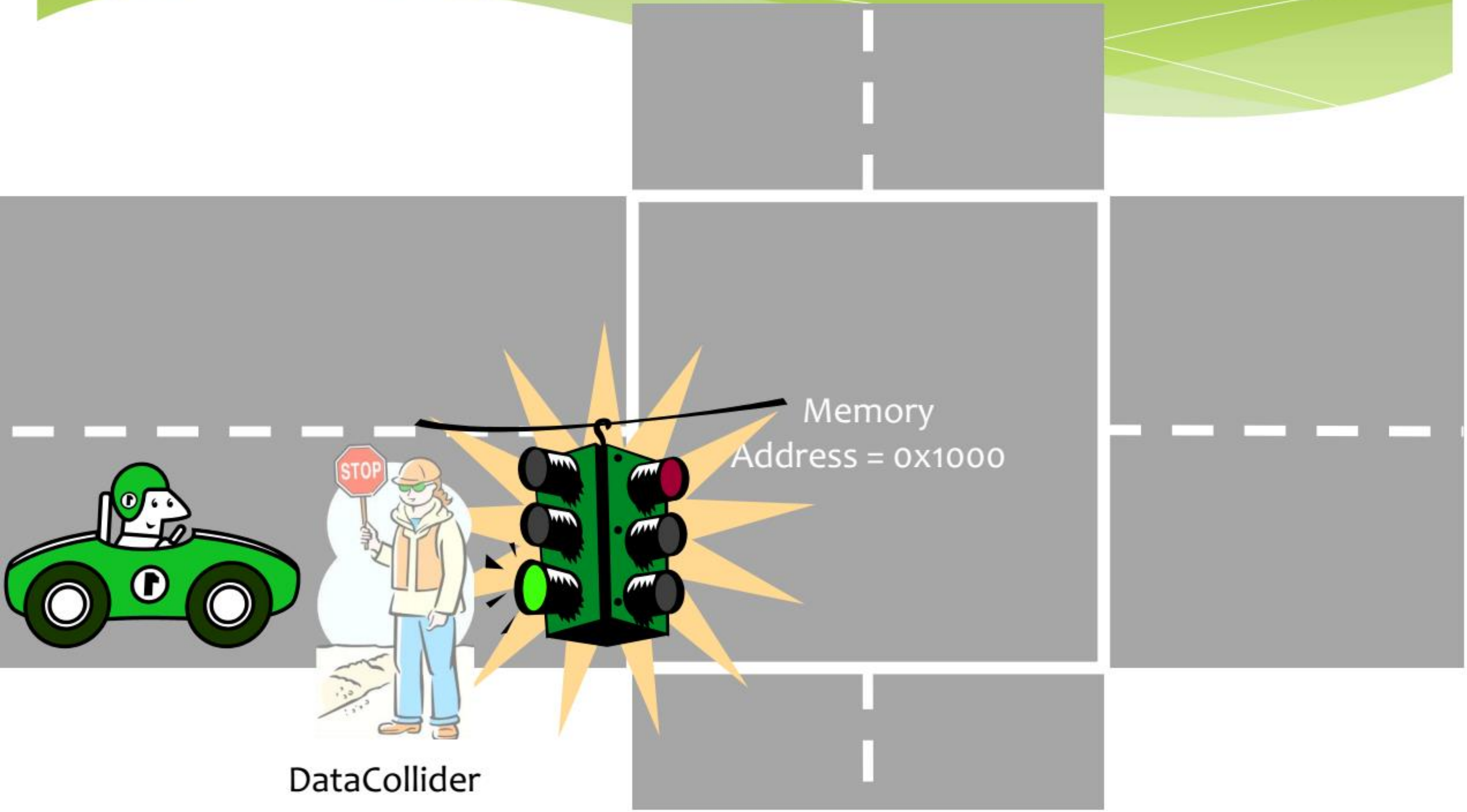
Intersection Metaphor



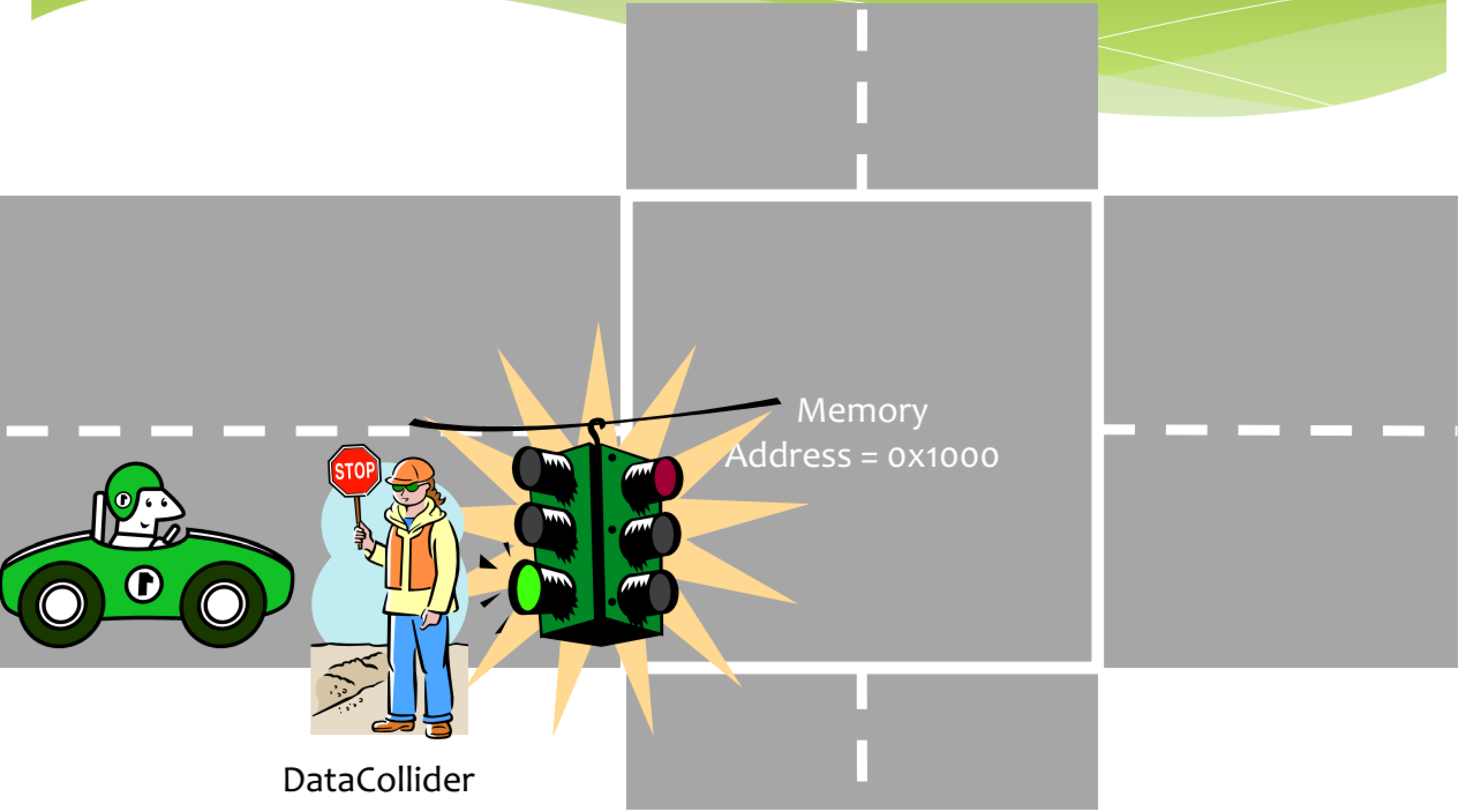
Intersection Metaphor



Intersection Metaphor



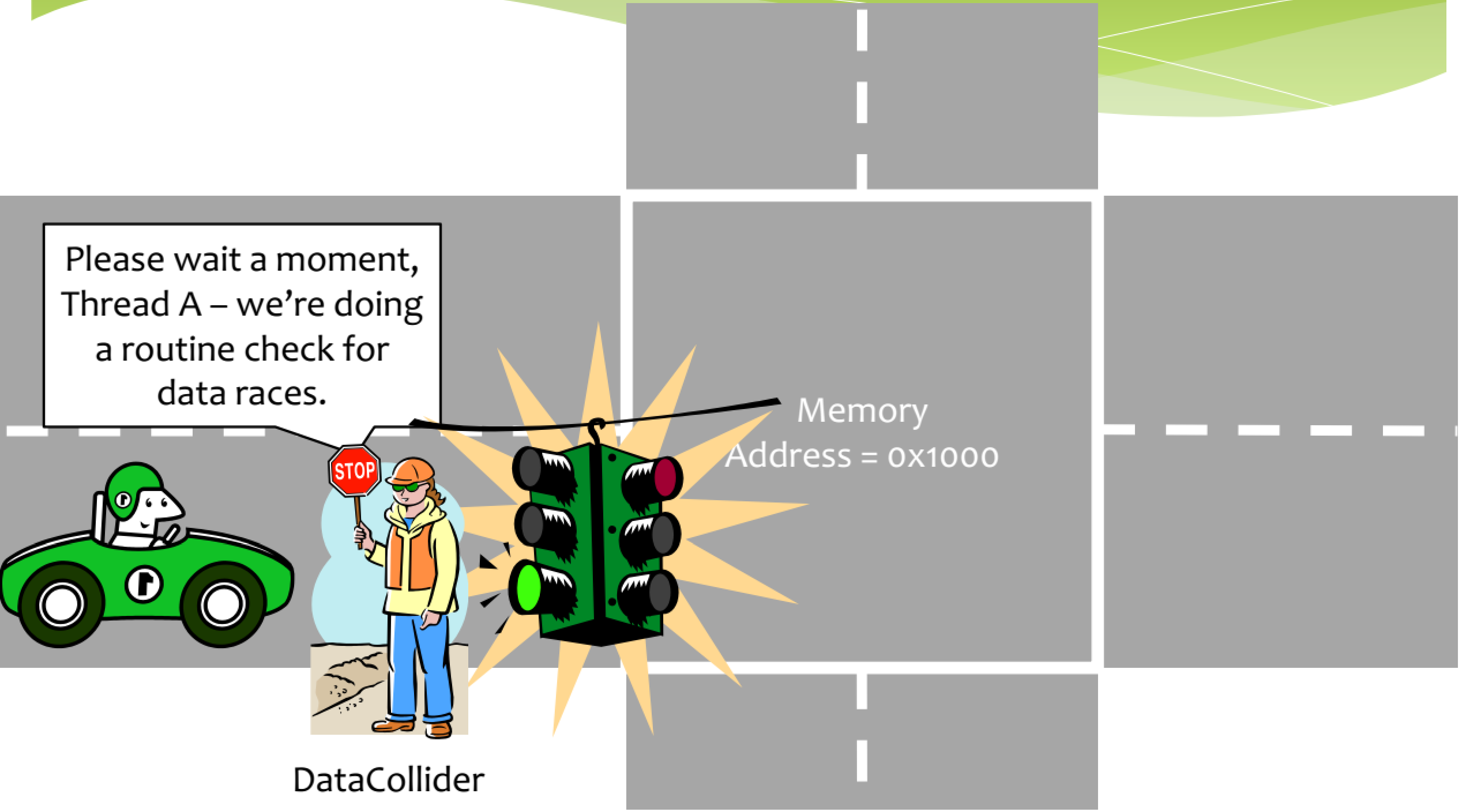
Intersection Metaphor



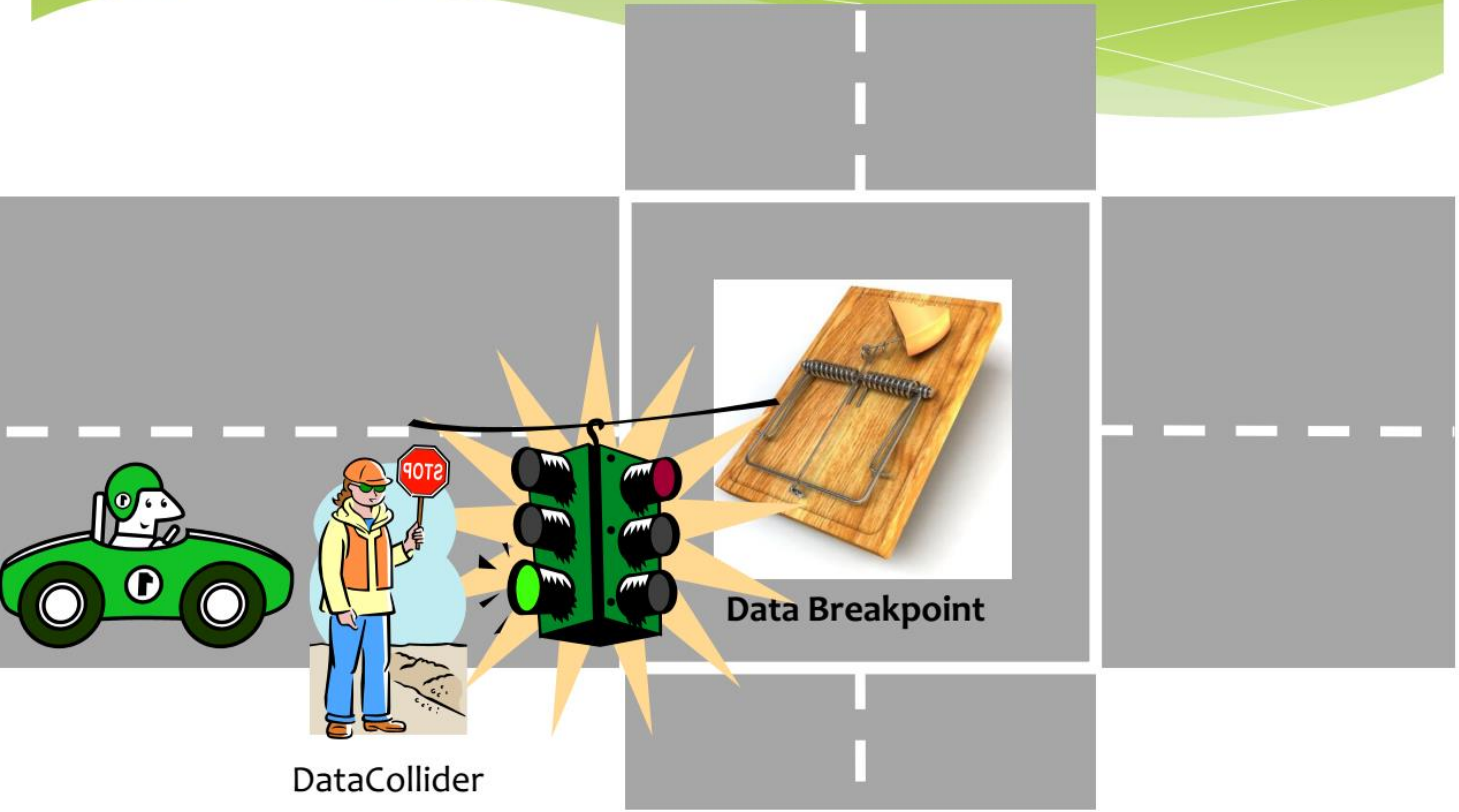
Memory
Address = 0x1000

DataCollider

Intersection Metaphor


Please wait a moment, Thread A – we're doing a routine check for data races.
STOP
Memory
Address = 0x1000
DataCollider

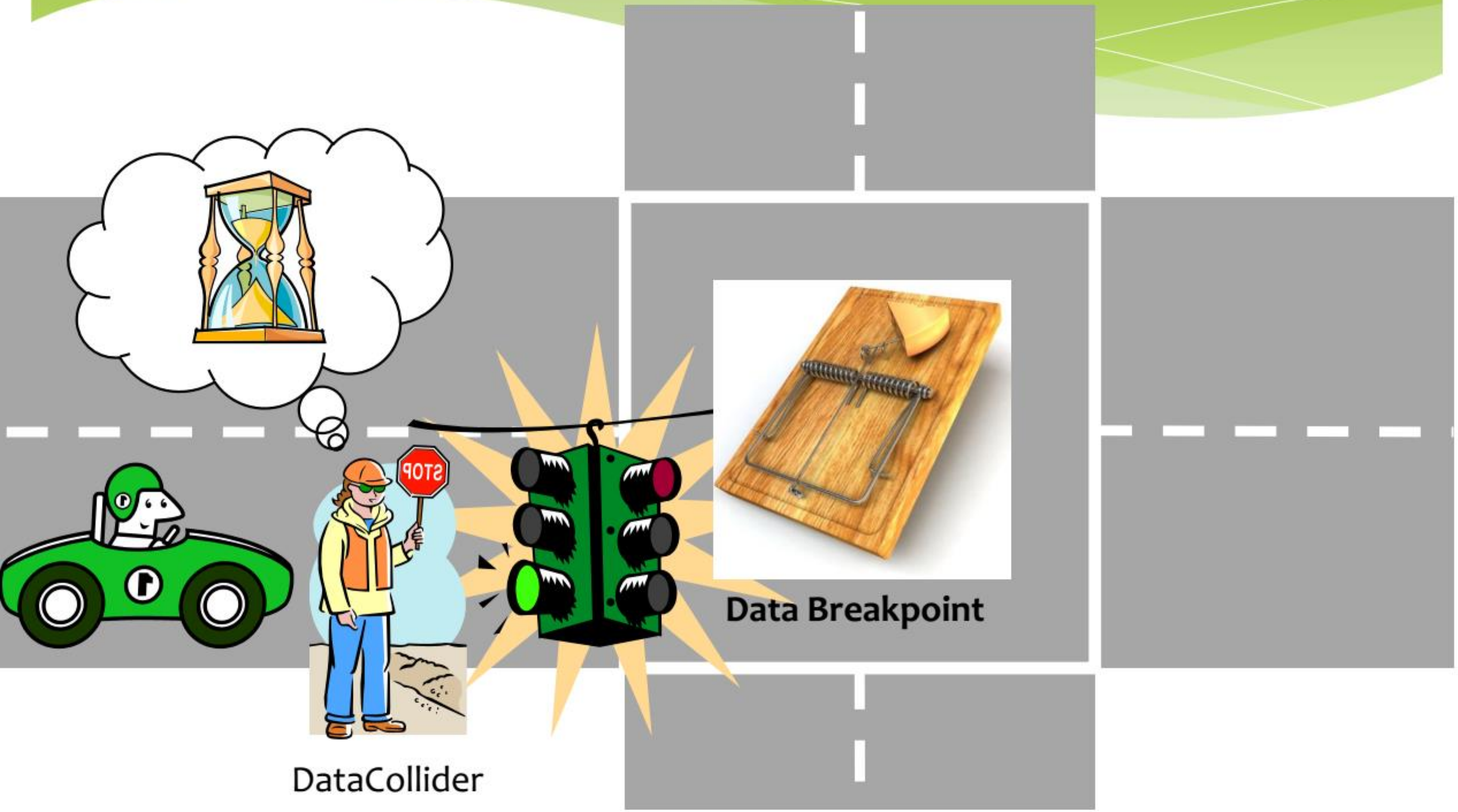
Intersection Metaphor



Data Breakpoint

DataCollider

Intersection Metaphor

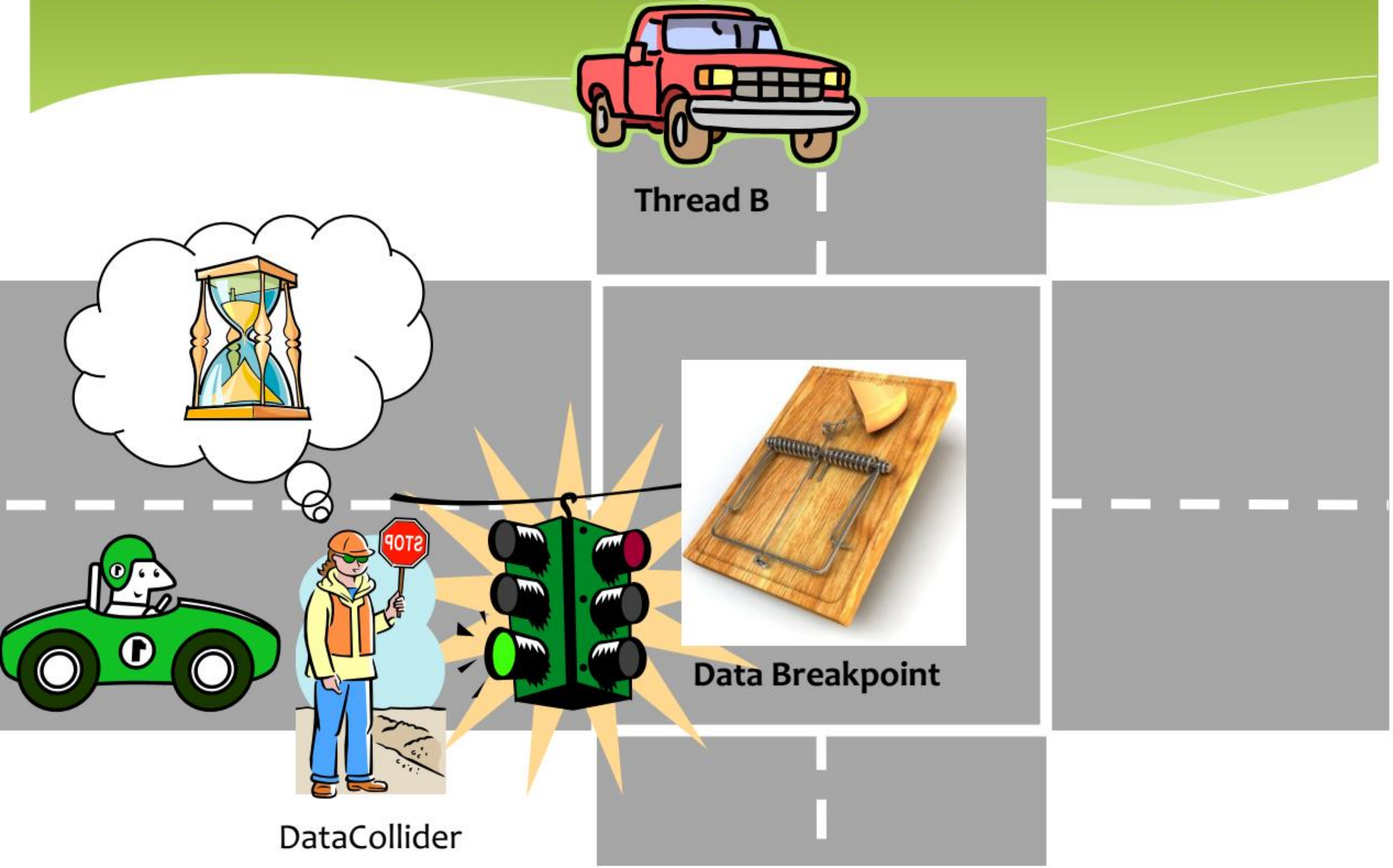


Data Breakpoint

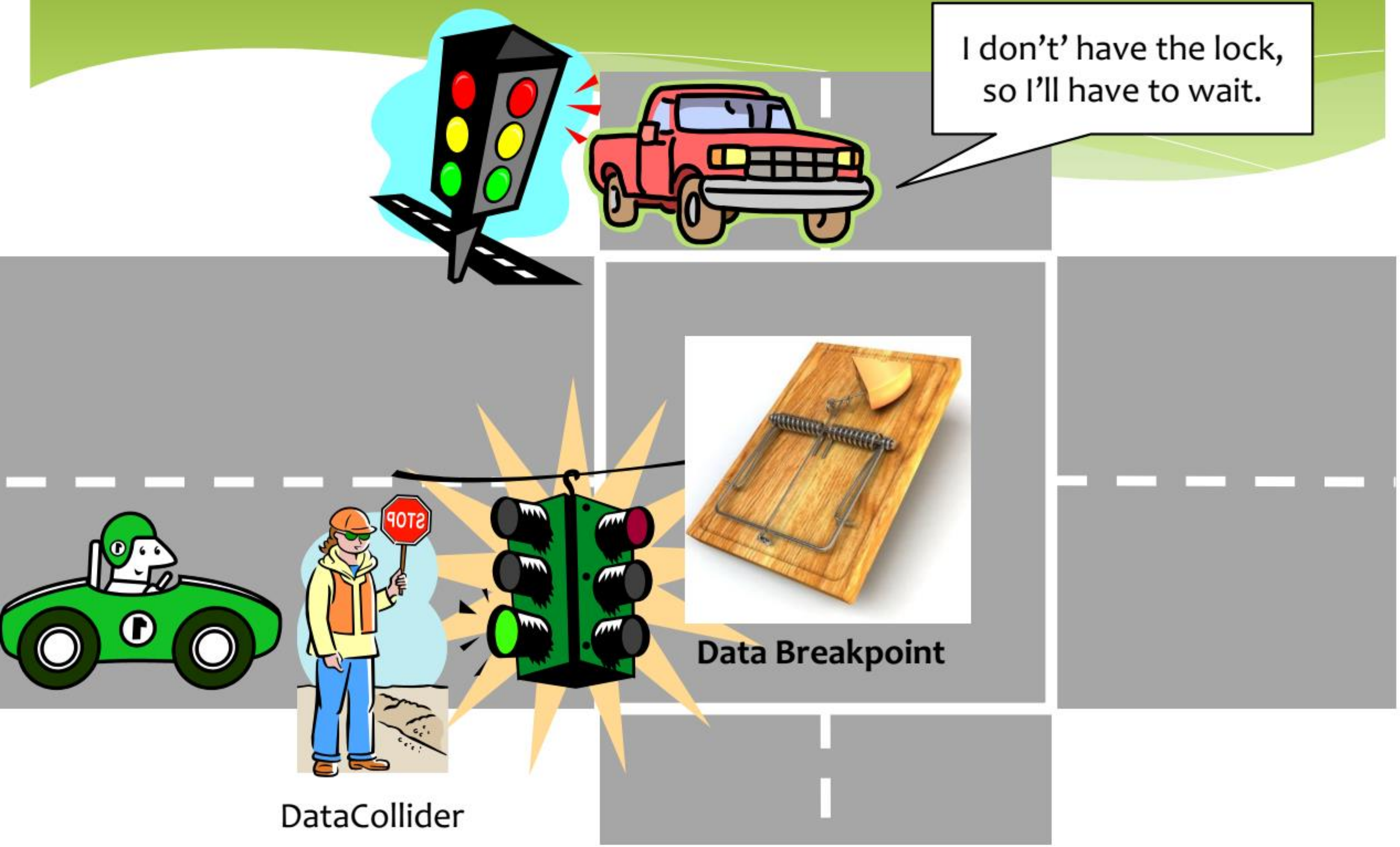
DataCollider

Intersection Metaphor: Normal Case

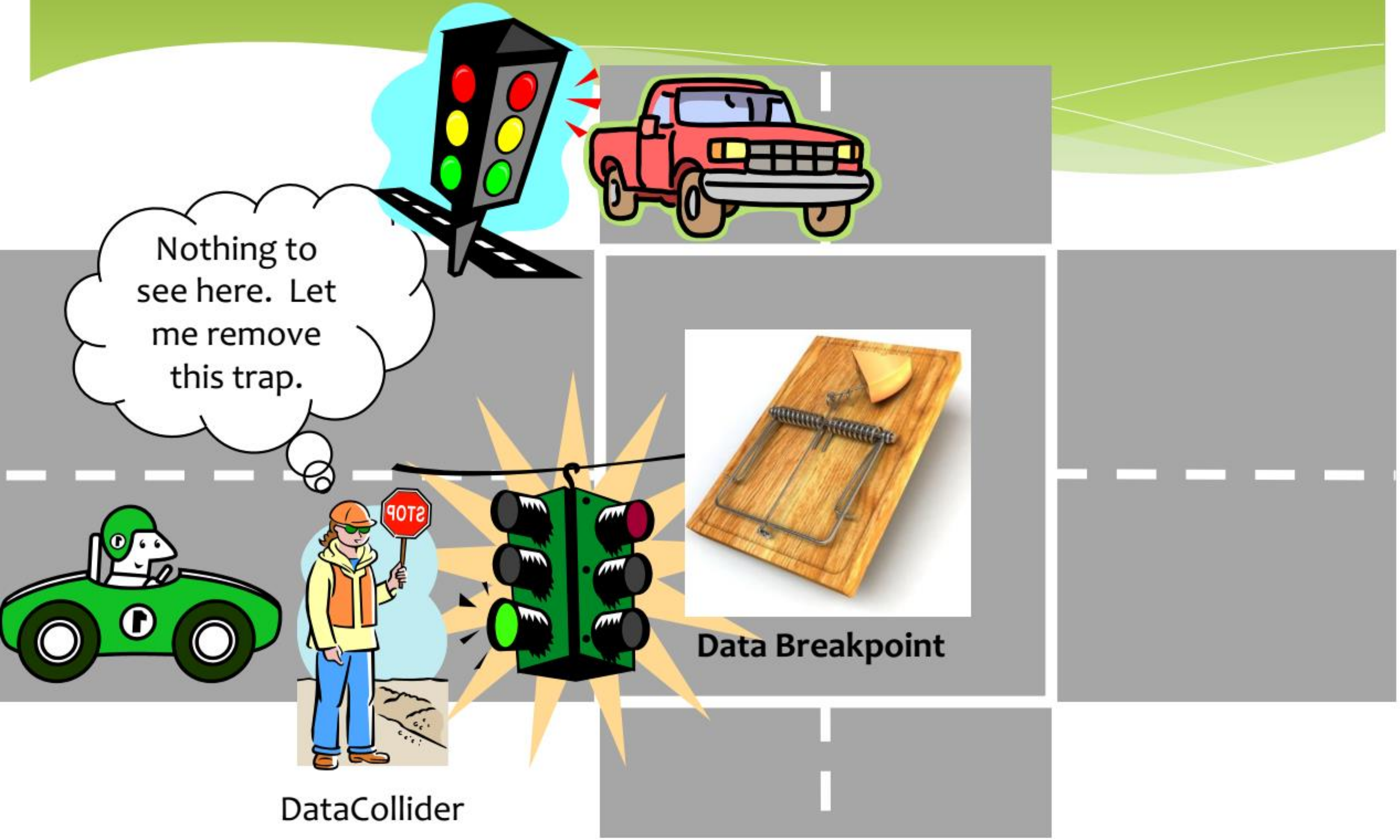
Intersection Metaphor: Normal Case



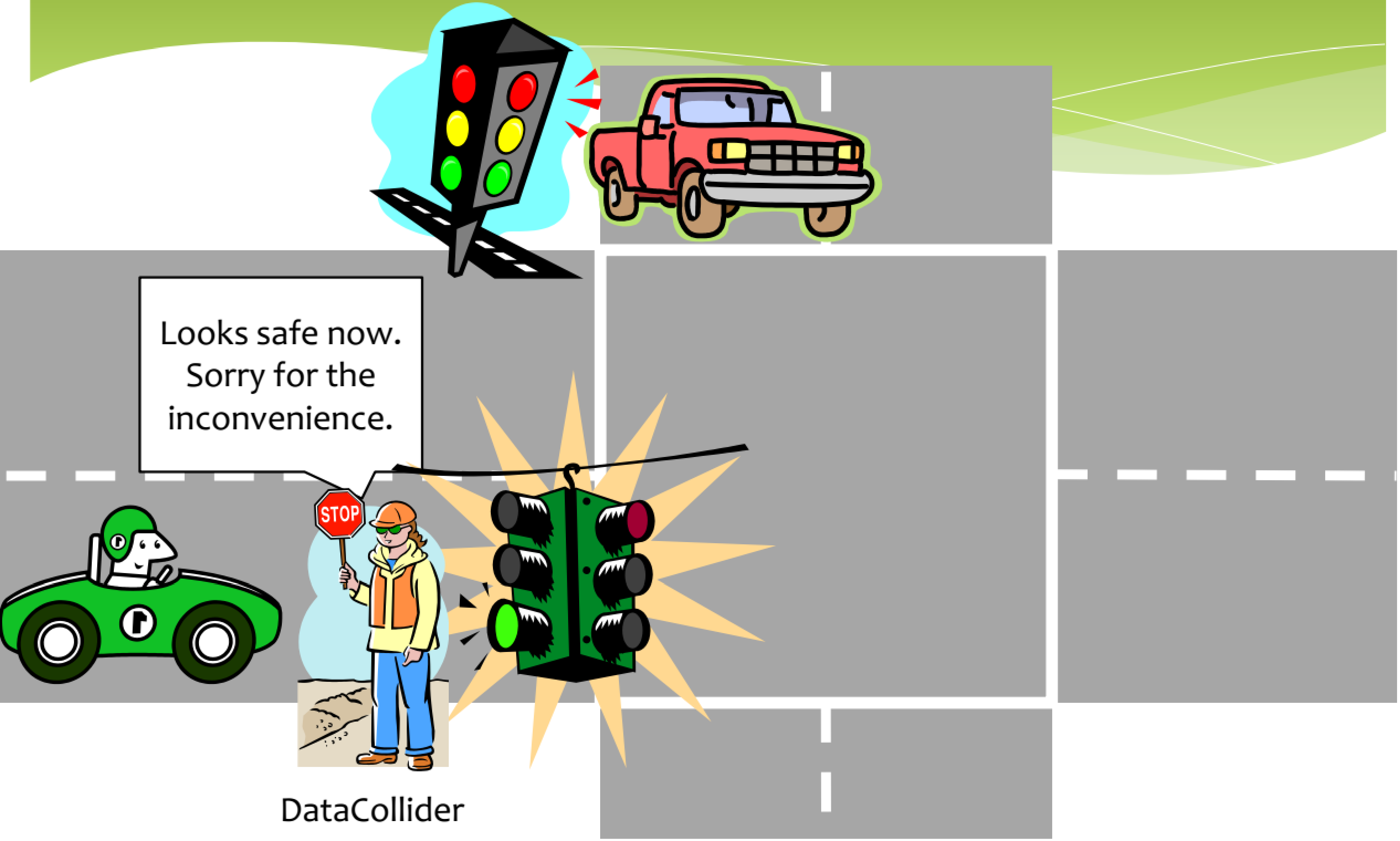
Intersection Metaphor: Normal Case



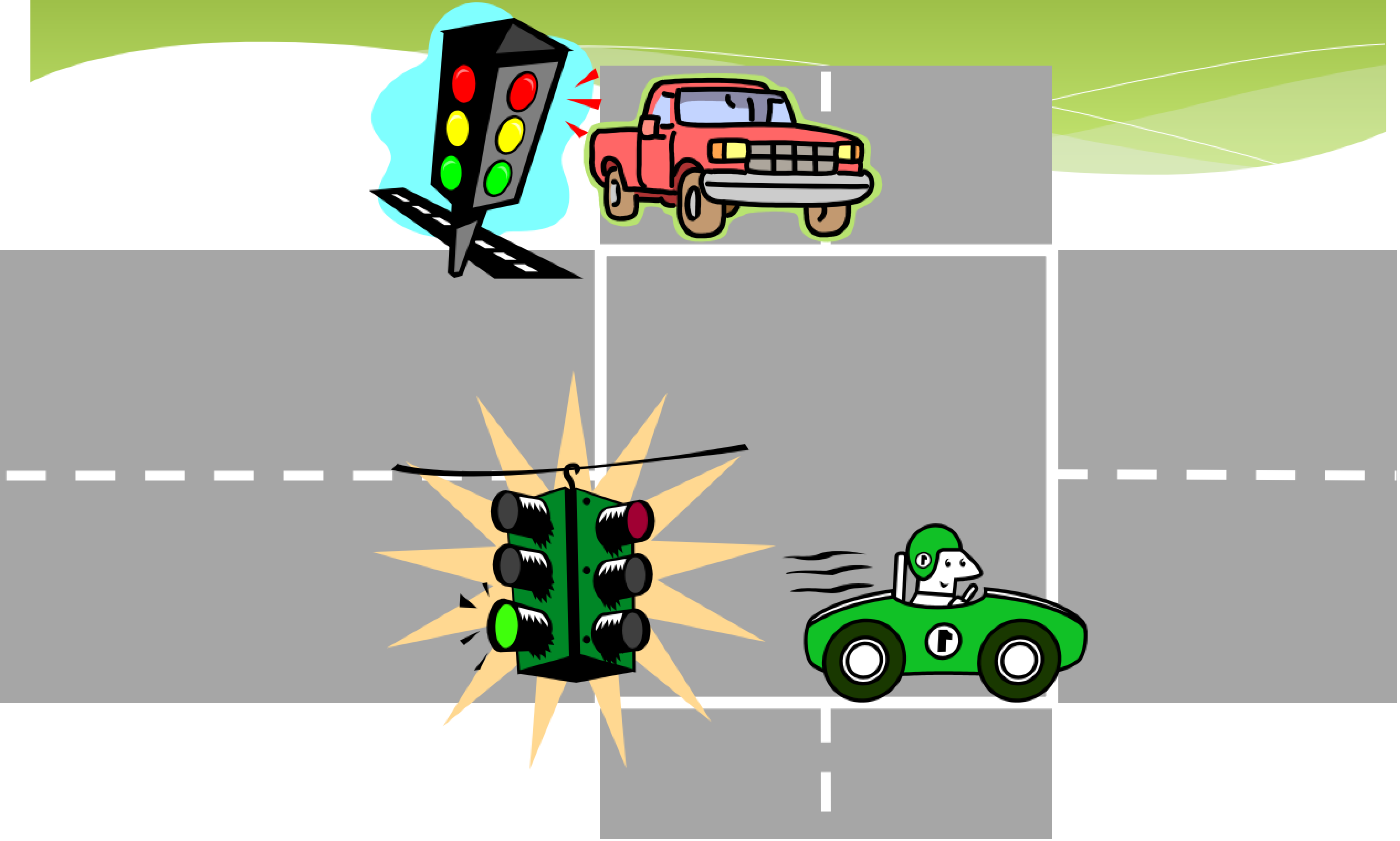
Intersection Metaphor: Normal Case



Intersection Metaphor: Normal Case

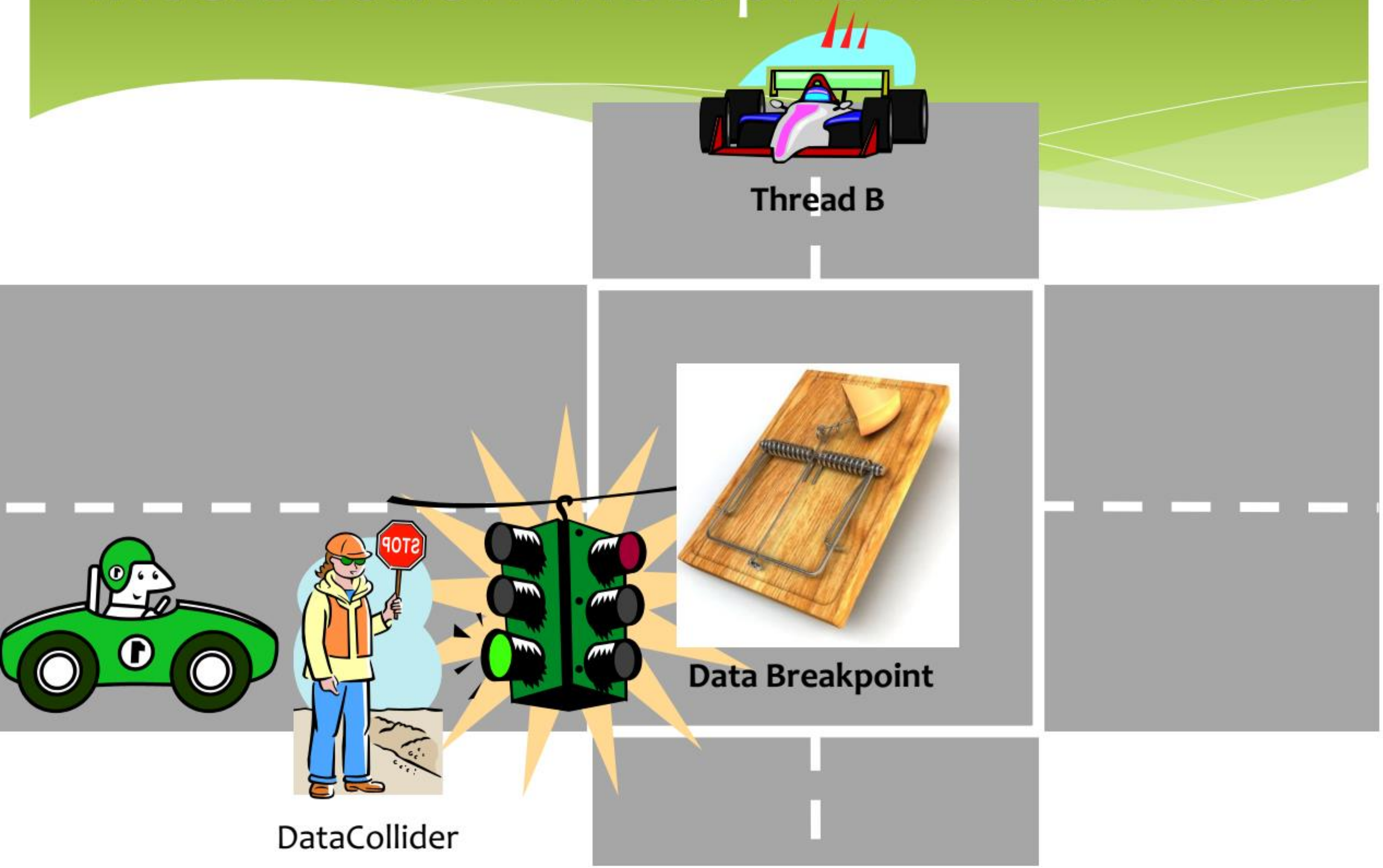


Intersection Metaphor: Normal Case



Intersection Metaphor: Data Race

Intersection Metaphor: Data Race

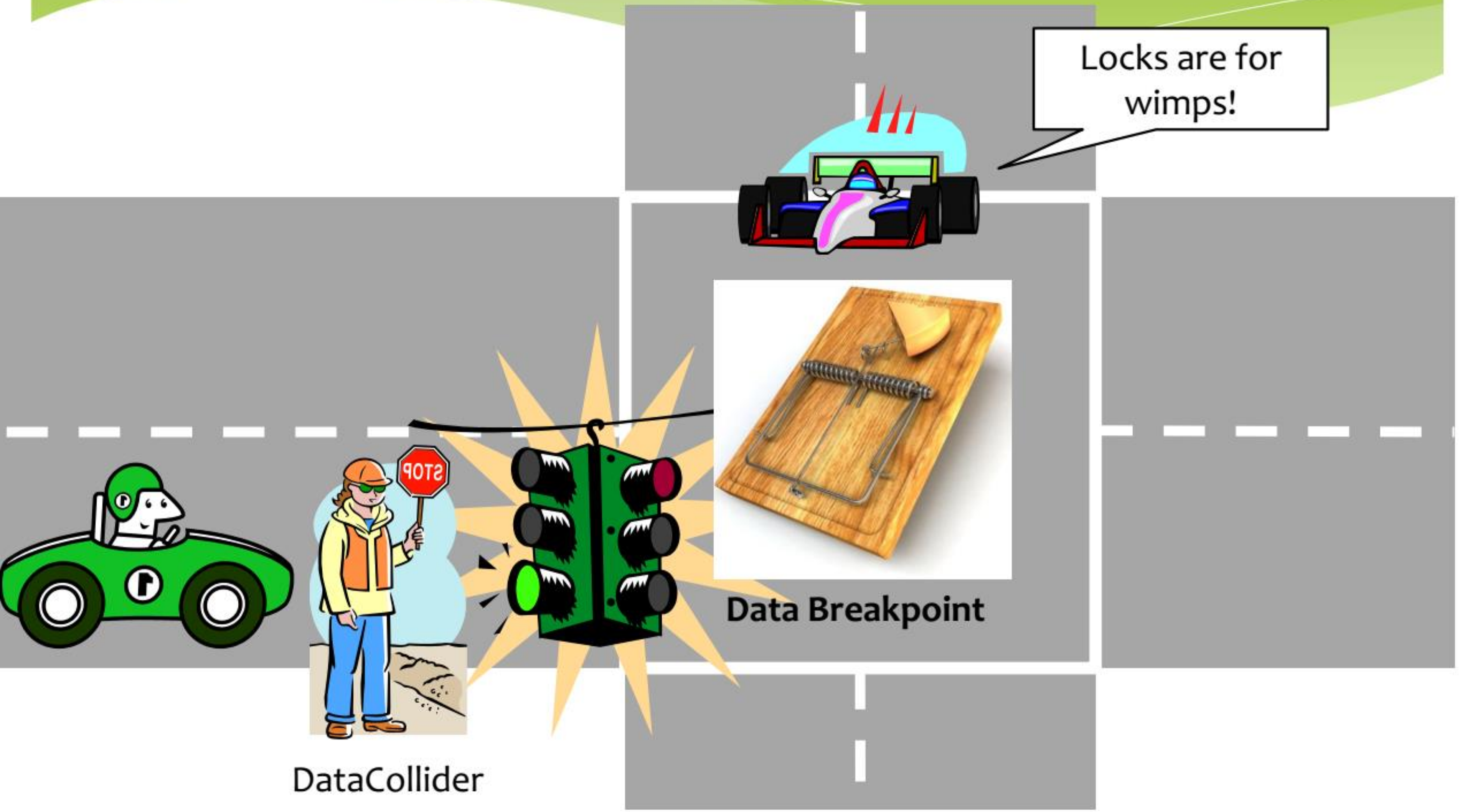


Thread B

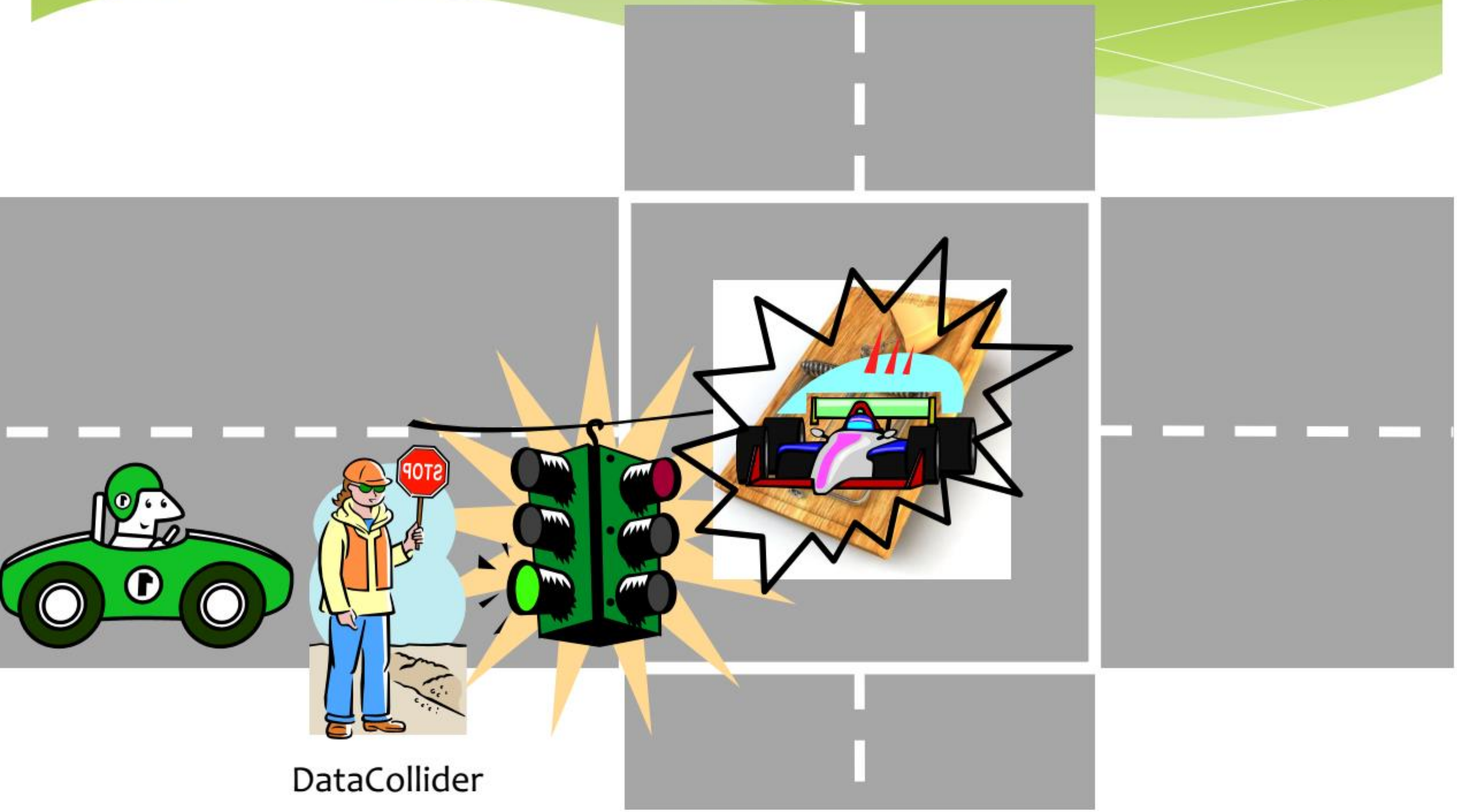
Data Breakpoint

DataCollider

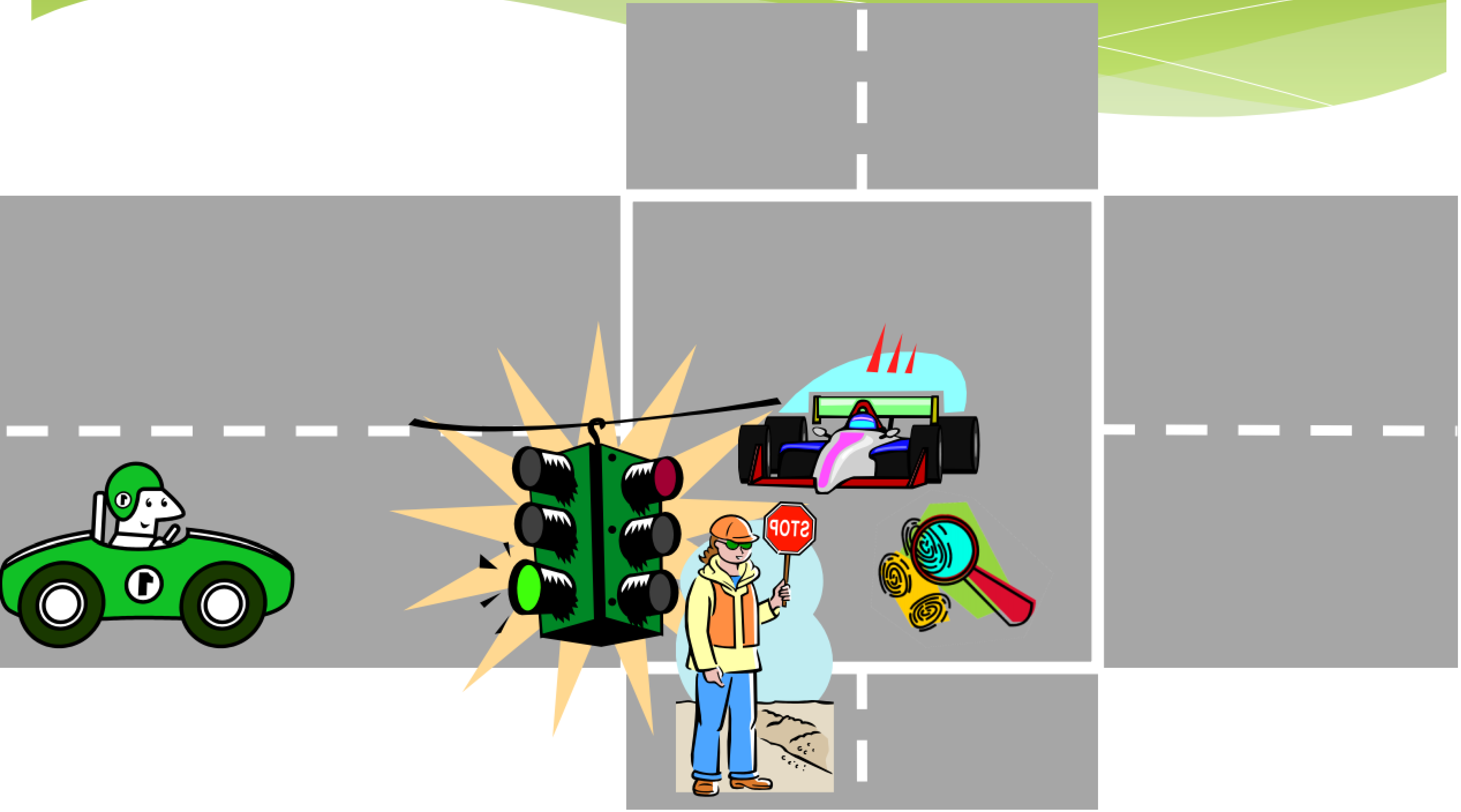
Intersection Metaphor: Data Race



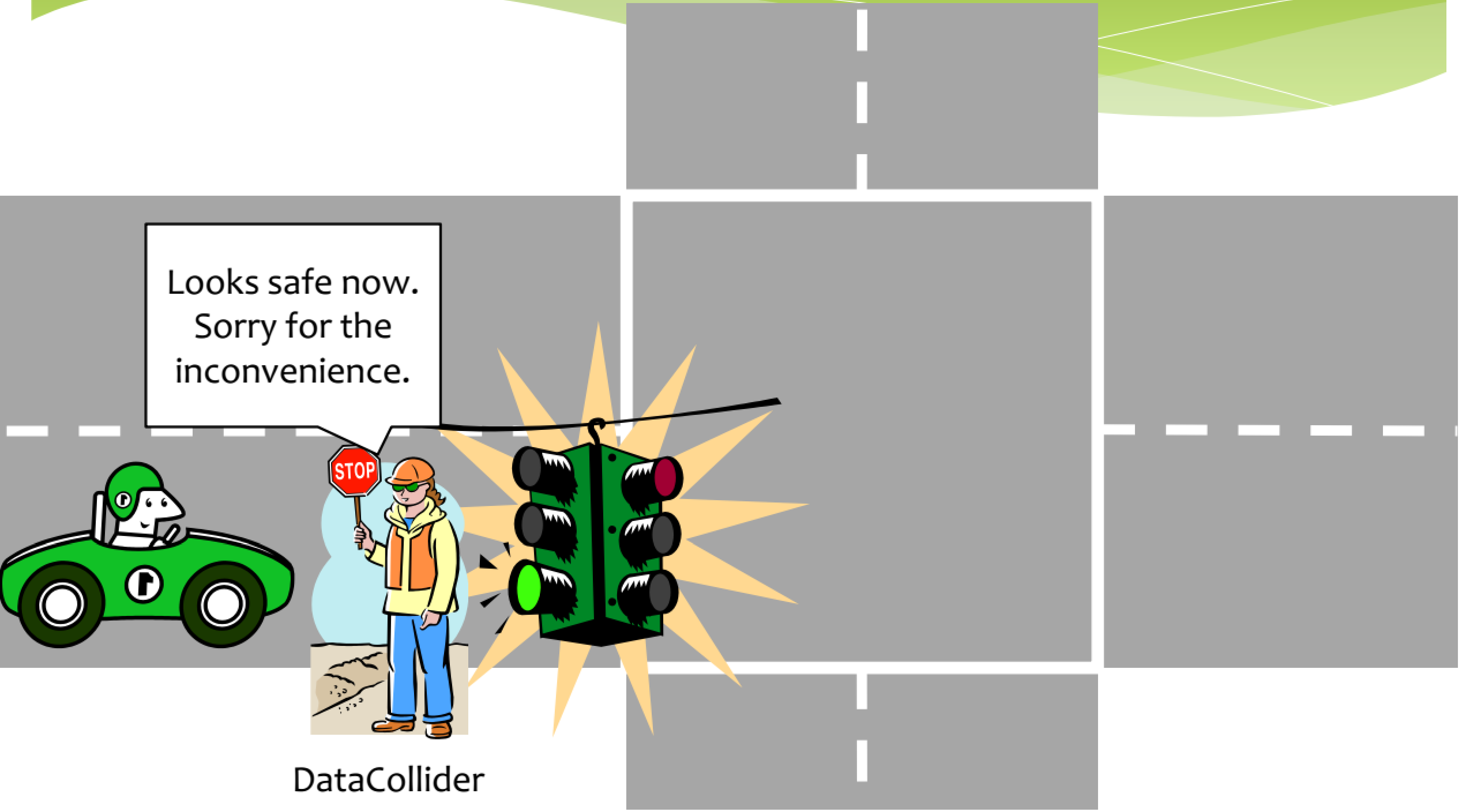
Intersection Metaphor: Data Race



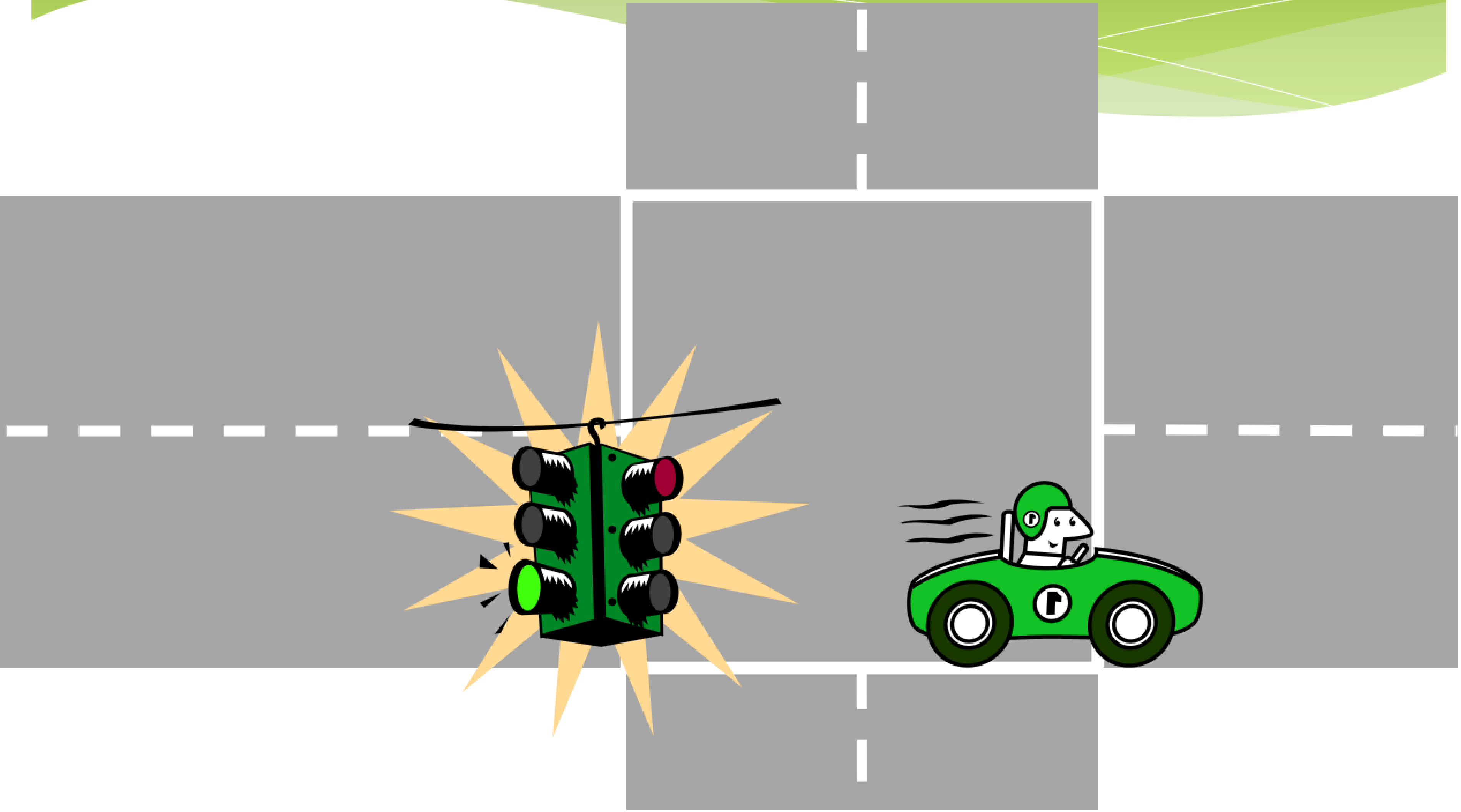
Intersection Metaphor: Data Race



Intersection Metaphor: Data Race



Intersection Metaphor: Data Race



Data breakpoint

- Advantage
  - Setting the data breakpoint will catch the colliding thread in the act .
  - This provides much more actionable debugging information.
- Disadvantage
  - Works on virtual address

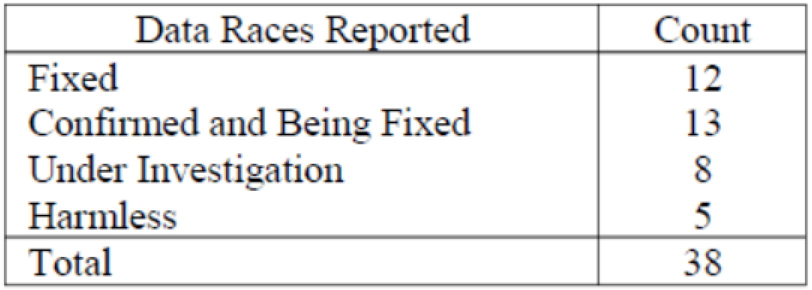
Repeated reads

- Advantage
  - The additional approach helps detect races caused by:
    - Hardware interaction via DMA
    - Physical memory that has multiple virtual mappings
- Disadvantage
  - Cannot detect:
    - read conflicts at a breakpoint of write operation
    - Multi-writes but the value doesn’t change

Results: bugs found

- 25 confirmed bugs in the Windows OS have been found

- 8 more are still pending investigation



Data Races Reported	Count
Fixed	12
Confirmed and Being Fixed	13
Under Investigation	8
Harmless	5
Total	38

Some Problems

- Causing kernel crash?
- Sampling in other methods?
- Using DataCollider in user programs?

Thanks!