DWARVES FOUNDATION

Singleton

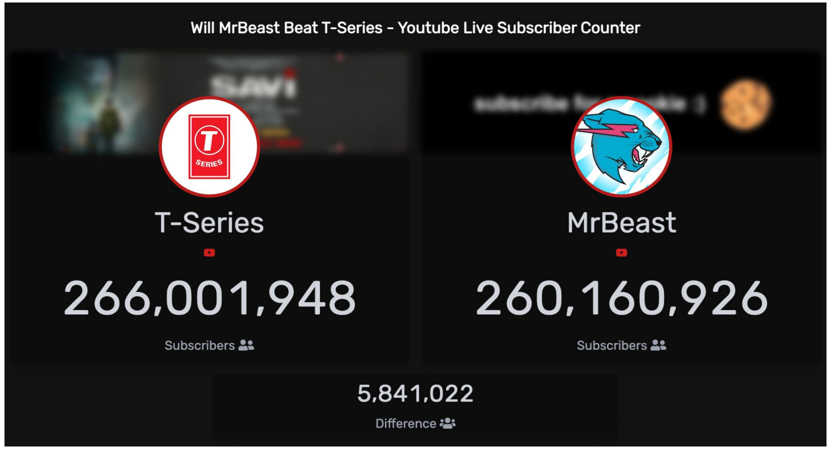
Creational patterns

Agenda

1. Problem / Solution
2. Characteristics
3. Applicability
4. Pros & Cons
5. Q&A

DWARVES FOUNDATION

Problem


Will MrBeast Beat T-Series - Youtube Live Subscriber Counter

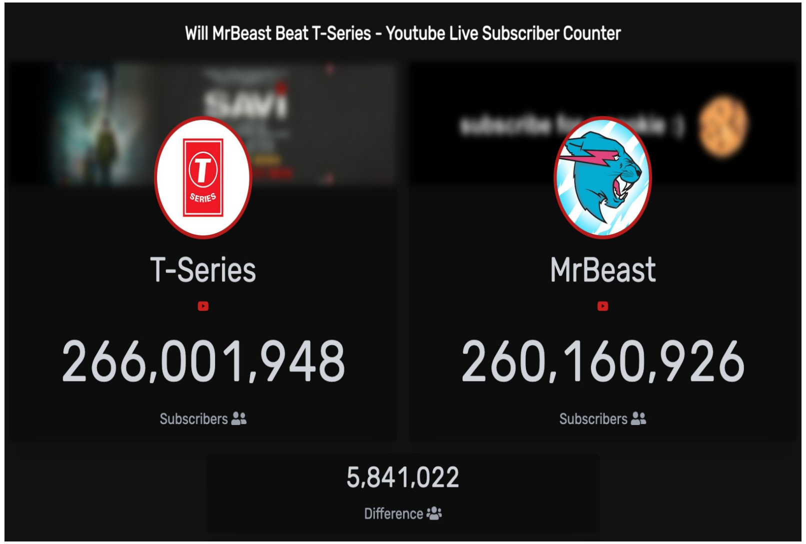
T-Series	MrBeast
266,001,948	260,160,926
Subscribers	Subscribers

5,841,022
Difference

DWARVES FOUNDATION

Problem

- Different counters are being used for different viewers
- Existing counter can be accidentally resetted when there is a new viewer

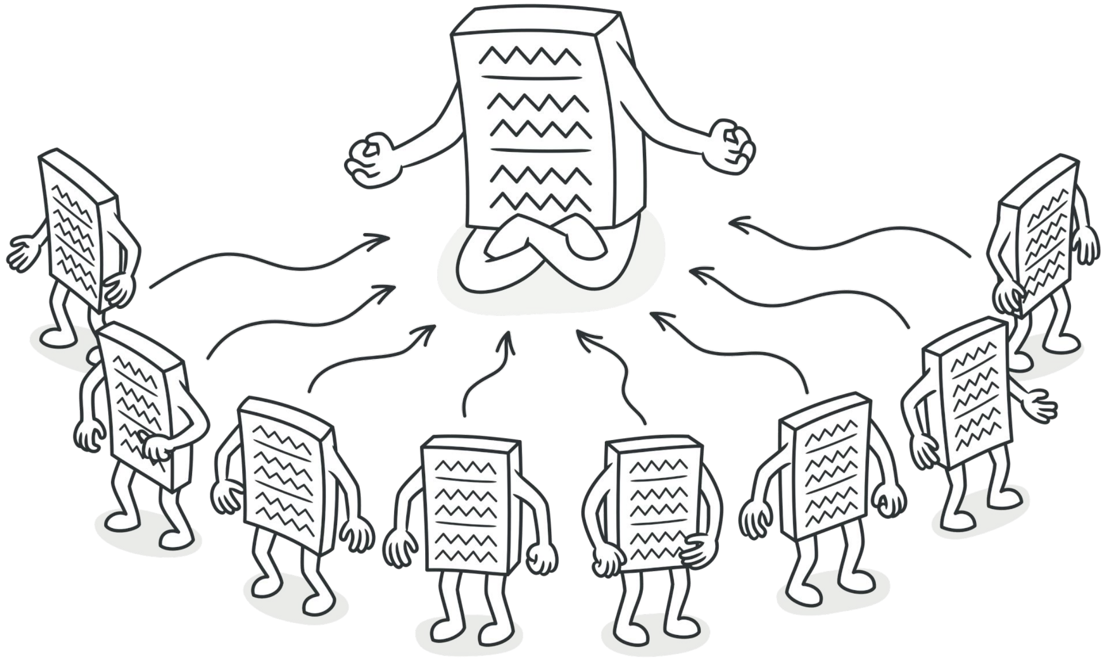


DWARVES FOUNDATION

Characteristics

- 1 class/type - 1 instance
- Provide global access to that single instance
- Thread-safe instantiation

Applicability



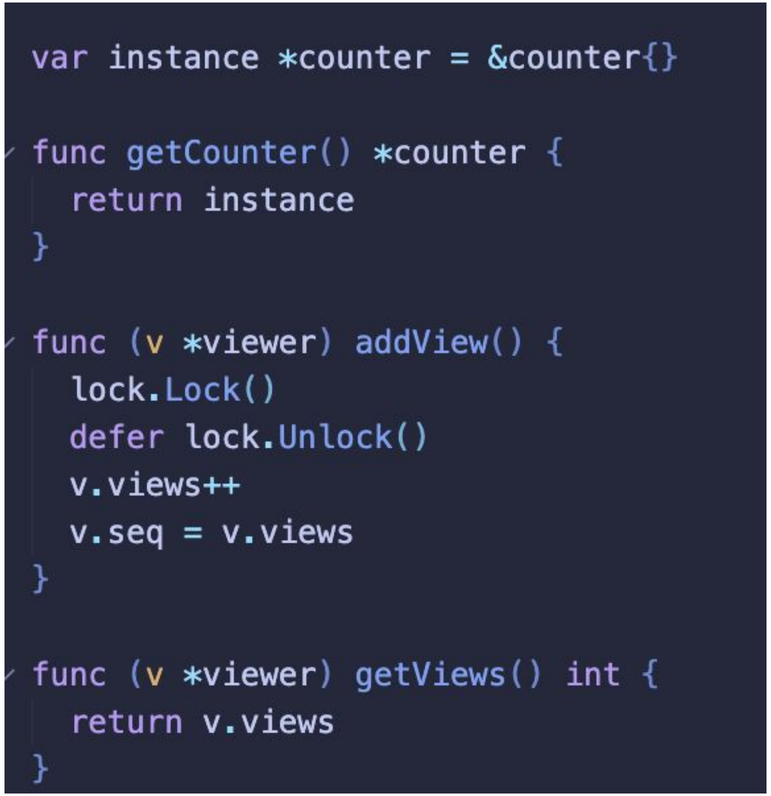
DWARVES FOUNDATION

Pros & Cons

- Manage one & only shared resource

- Violates Single Responsibility Principle
- Tightly coupled codebase
- Complex to debug and test

Pseudocode - Eager



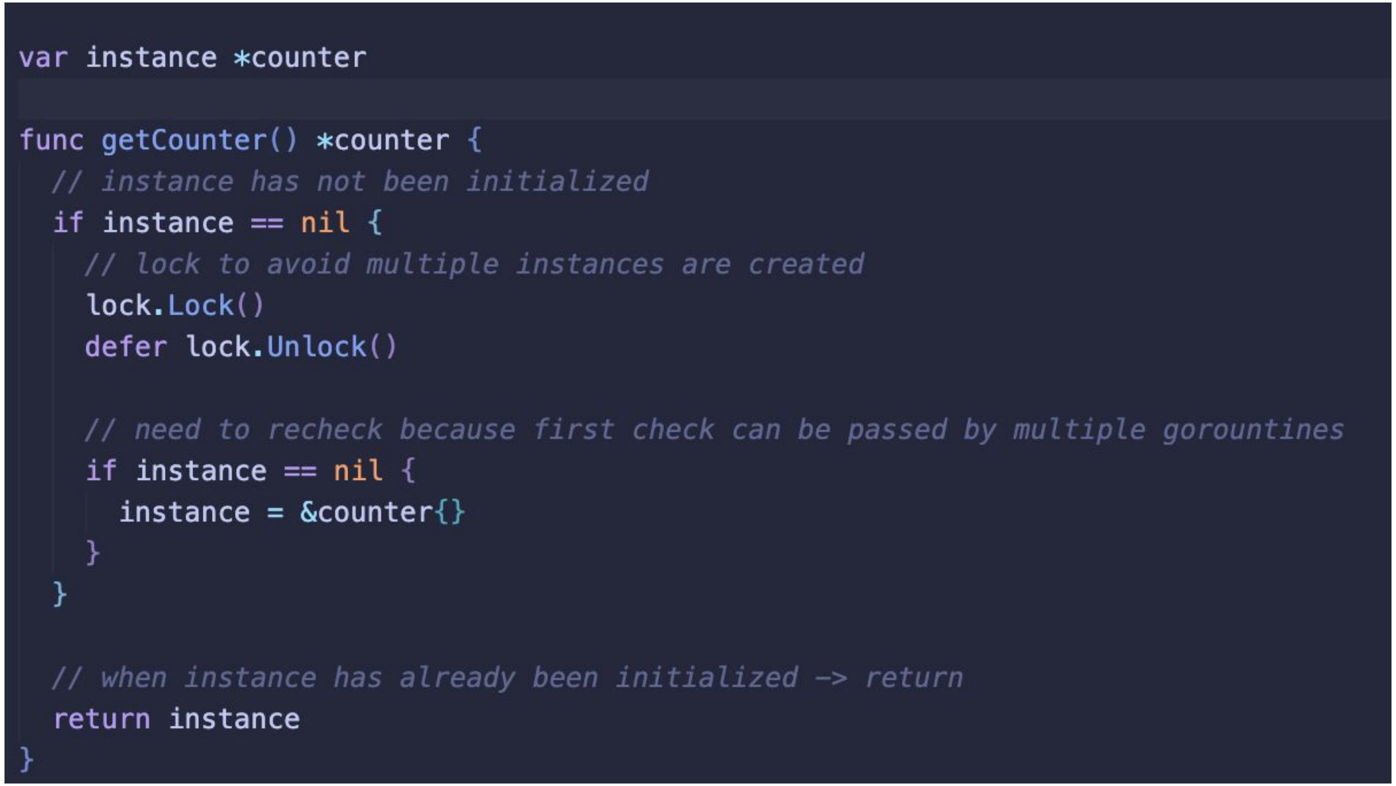
```
var instance *counter = &counter{}

func getCounter() *counter {
  return instance
}

func (v *viewer) addView() {
  lock.Lock()
  defer lock.Unlock()
  v.views++
  v.seq = v.views
}

func (v *viewer) getViews() int {
  return v.views
}
```

Pseudocode - Lazy



```
var instance *counter

func getCounter() *counter {
  // instance has not been initialized
  if instance == nil {
    // lock to avoid multiple instances are created
    lock.Lock()
    defer lock.Unlock()

    // need to recheck because first check can be passed by multiple gorountines
    if instance == nil {
      instance = &counter{}
    }
  }

  // when instance has already been initialized -> return
  return instance
}
```

Thanks for listening!