Life After Locks

Alternative concurrency models

James Geall

AARDWARE

Who am I

- James Geall
- Developing software for 16 years, 10 years with my own company
- jageall@gmail.com
- @jageall

What to expect

- A war story
- Code – lots of this
- If time allows – a look at some other patterns that are useful in this model

The war story

- Locks… are a lot like tribbles

Locks are hard

- Difficult to reason about
- Difficult to debug
- Pollute the domain model

Demo

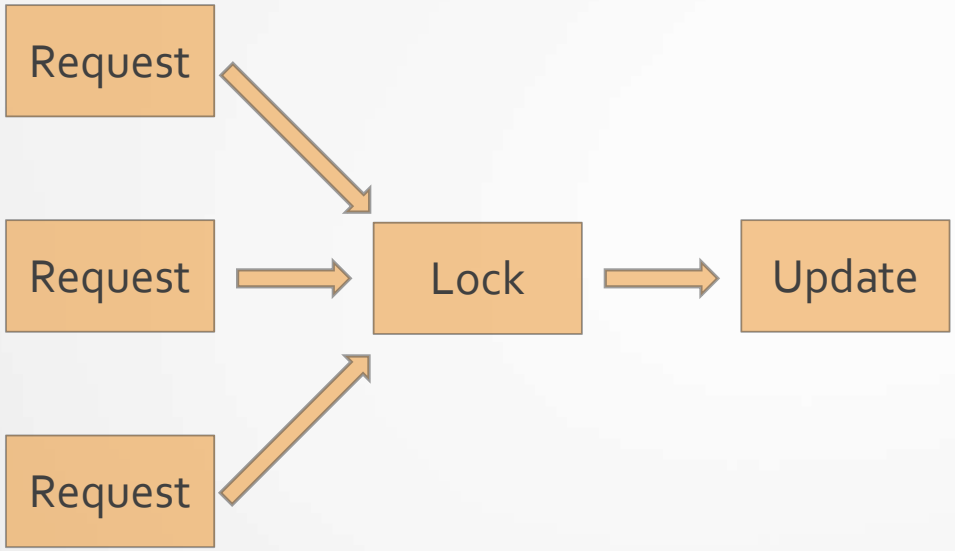
- Some issues with locks….

A review of the issues

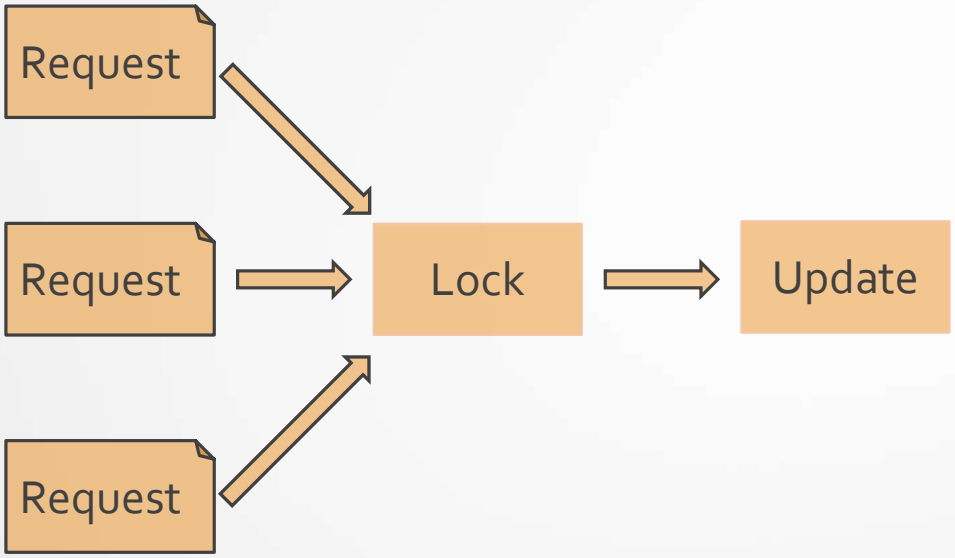
- Order matters
  - This gets significantly more difficult as the system grows
- Scope matters
  - This gets significantly more difficult as the system grows
- Proving correctness is very hard
  - Tests/bugs are difficult to reproduce

What happens if we build a model of our concurrency explicitly?

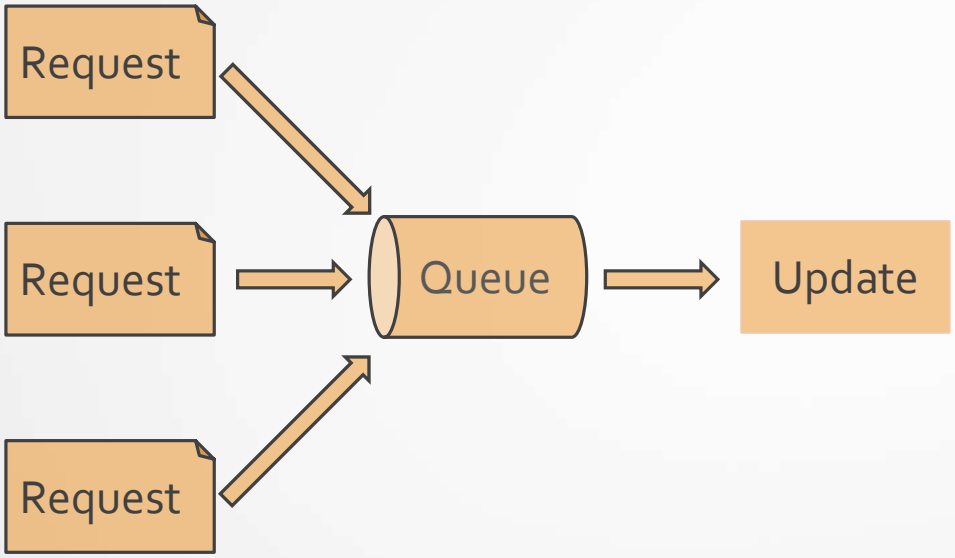
- The Actor model (Carl Hewitt, 1973)
- Staged Event Driven Architecture (Matt Welsh, 2000)



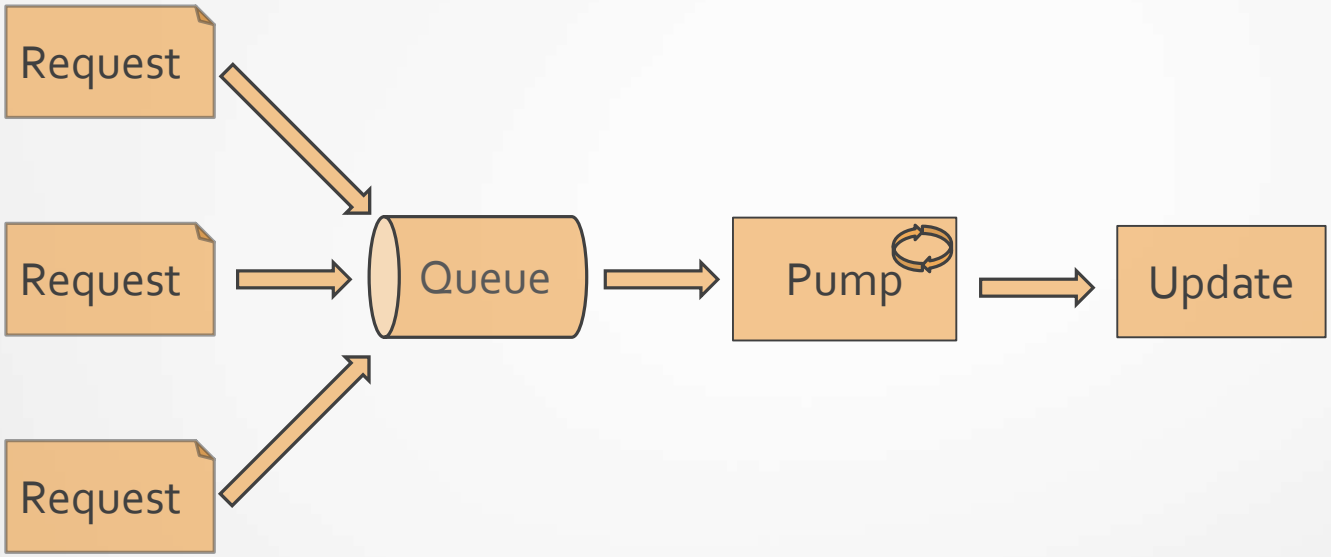
If requests were…



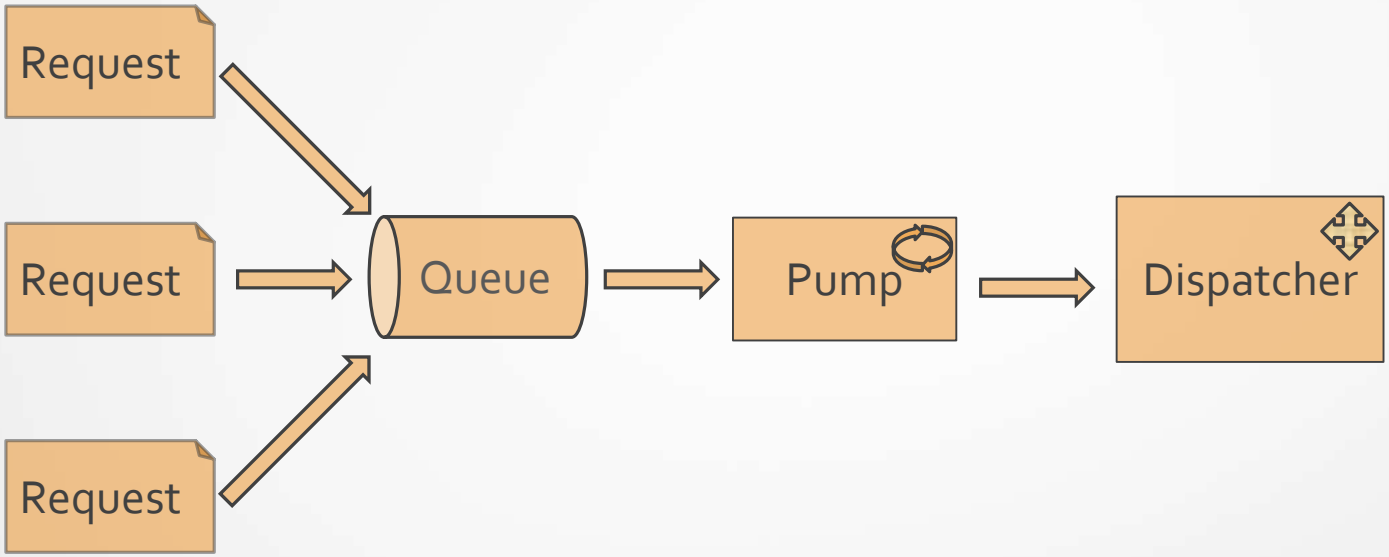
messages… then this looks like a….



Queue of messages…



We need a thread for processing the queued messages…



And a way to get those messages to the right place

Lets build that…

Important rules

- No IO on the bus thread – much like on a UI message pump
- Don't be too chatty – like a network there is a limit to bandwidth
- Messages must be immutable once sent

Other benefits

- Monitoring/Instrumentation
- Adding functionality rarely changes existing code
- Tests tend to be behavioural
- State tends to be private

Messaging Resources

- Enterprise Integration Patterns – Gregor Hohpe
- 8 Fallacies of distributed computing
- Circuit Breaker – Michael Nygard

Questions