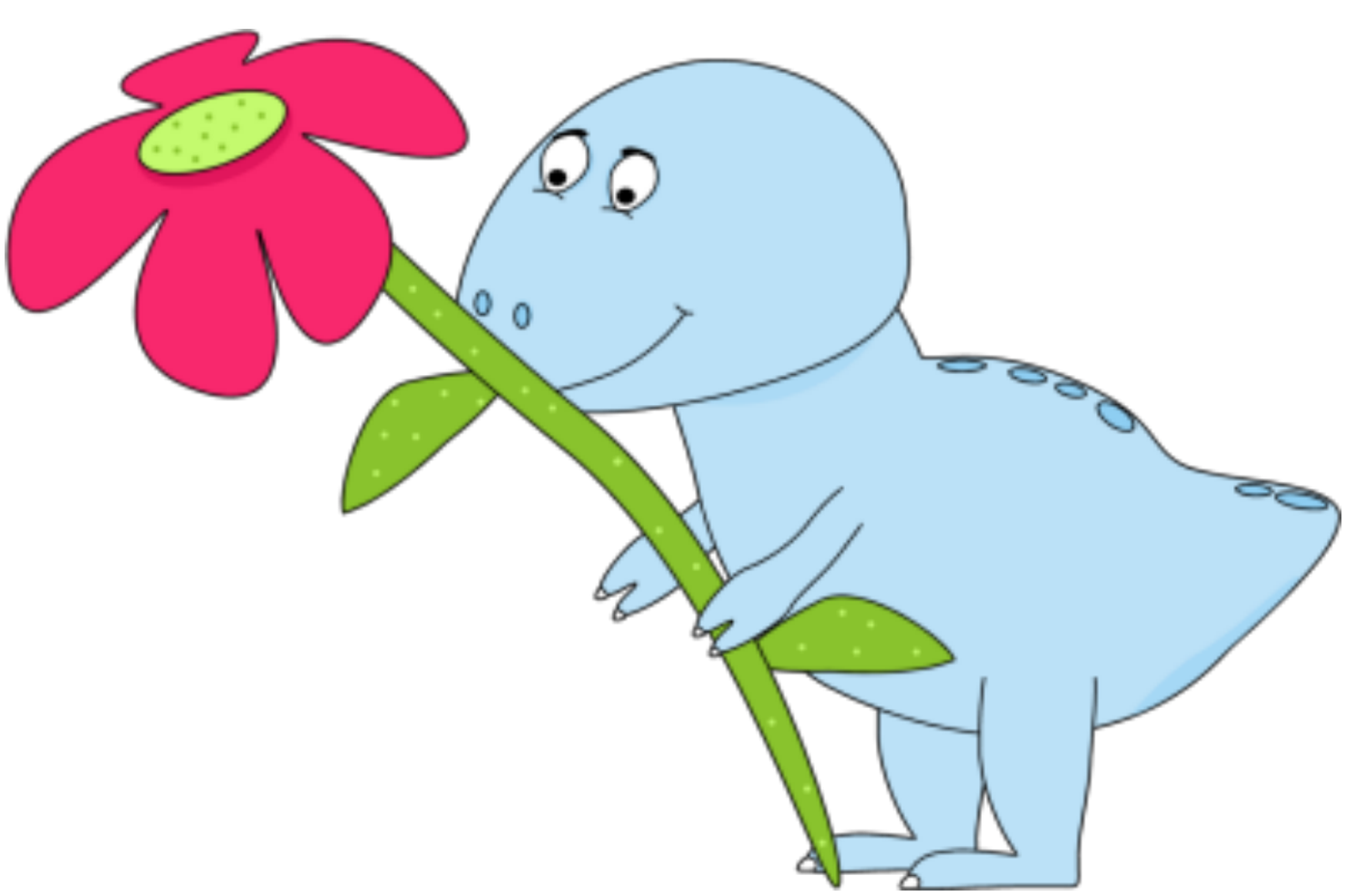
Modern Abstractions

and the web… and avoiding pitfalls because life is short

Irina Guberman

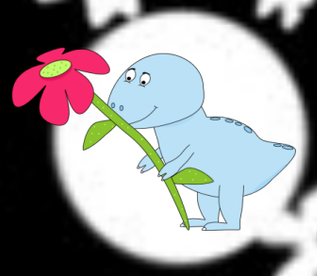
twitter: @irina_guberman

Long long ago in a galaxy far far away… a lowly software engineer named Irina came up with a social networking website idea she thought was really cute and sweet and cool: Compliments as a service… And she called her imaginary social networking website…



Kudos

She imagined millions and millions of users flocking to her amazing website. Who wouldn't want to give compliments that everyone can see online and better yet… receive(!!!) compliments that EVERYONE can see online! Collect compliments, send compliments as birthday gifts… yes millions and millions of users…



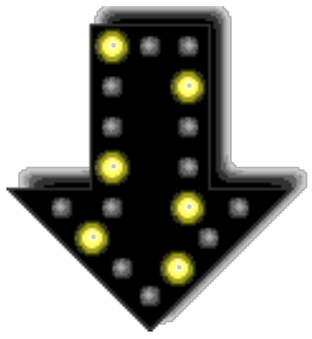
But back to reality… Kudos the website, was yet to be created. Irina, a backend Java engineer at the time, was already working at a company serving hundreds of millions of online users. To her, a backend Java engineer, that’s what Kudos really looked like:

Her
very
own
Java
webserver...

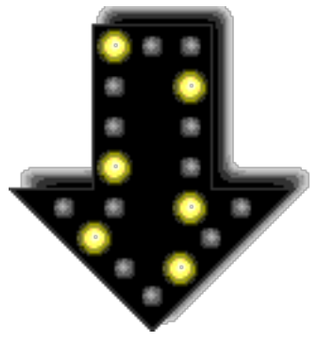
s00
scary...

Irina who was trying to start a startup while working for a startup faced two serious limitations as old as money:

Time



Money



Software

Hardware

PODPIS POSIADACZA KARTY /
AUTHORISED SIGNATURE

So, she knew that this Kudos business will be hard. But seriously, everything is hard! Making a lasagna is hard!

And lasagna becomes a reality even for a stay-at-work mom!

sometimes, evolution sucks

…They are searching for Engineers who are passionate about Java development AND concurrent programming…

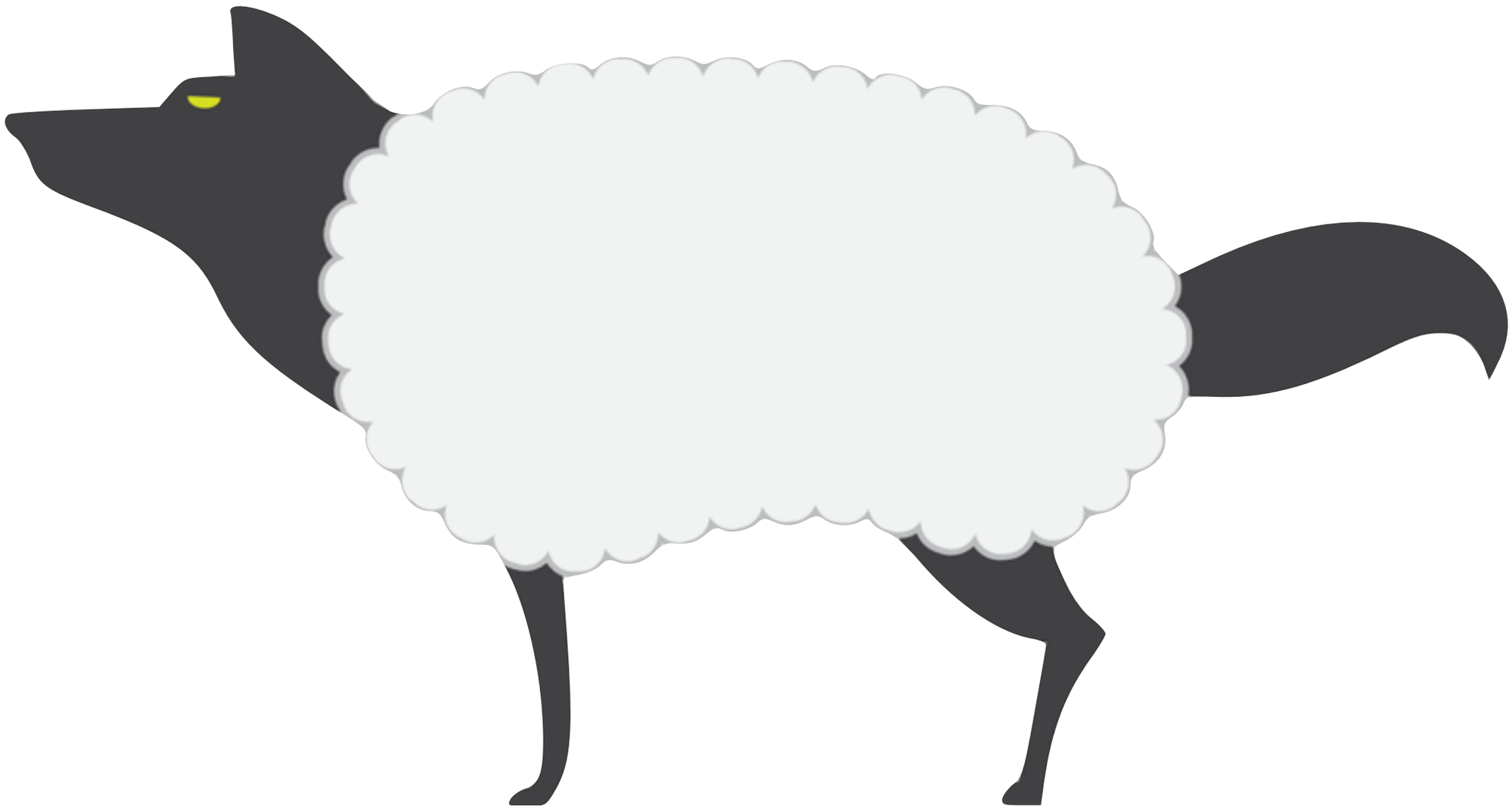
Peter Alvaro

“I see what you Mean”

https://www.youtube.com/watch?v=R2Aa4PivG0g&list=PLcGKfGEEONaCIl5eU53uPBnRJ9rbIH32R

Java

Java Concurrency Package



CONCURRENCY
Y U SO HARD?
DIYLOL.COM

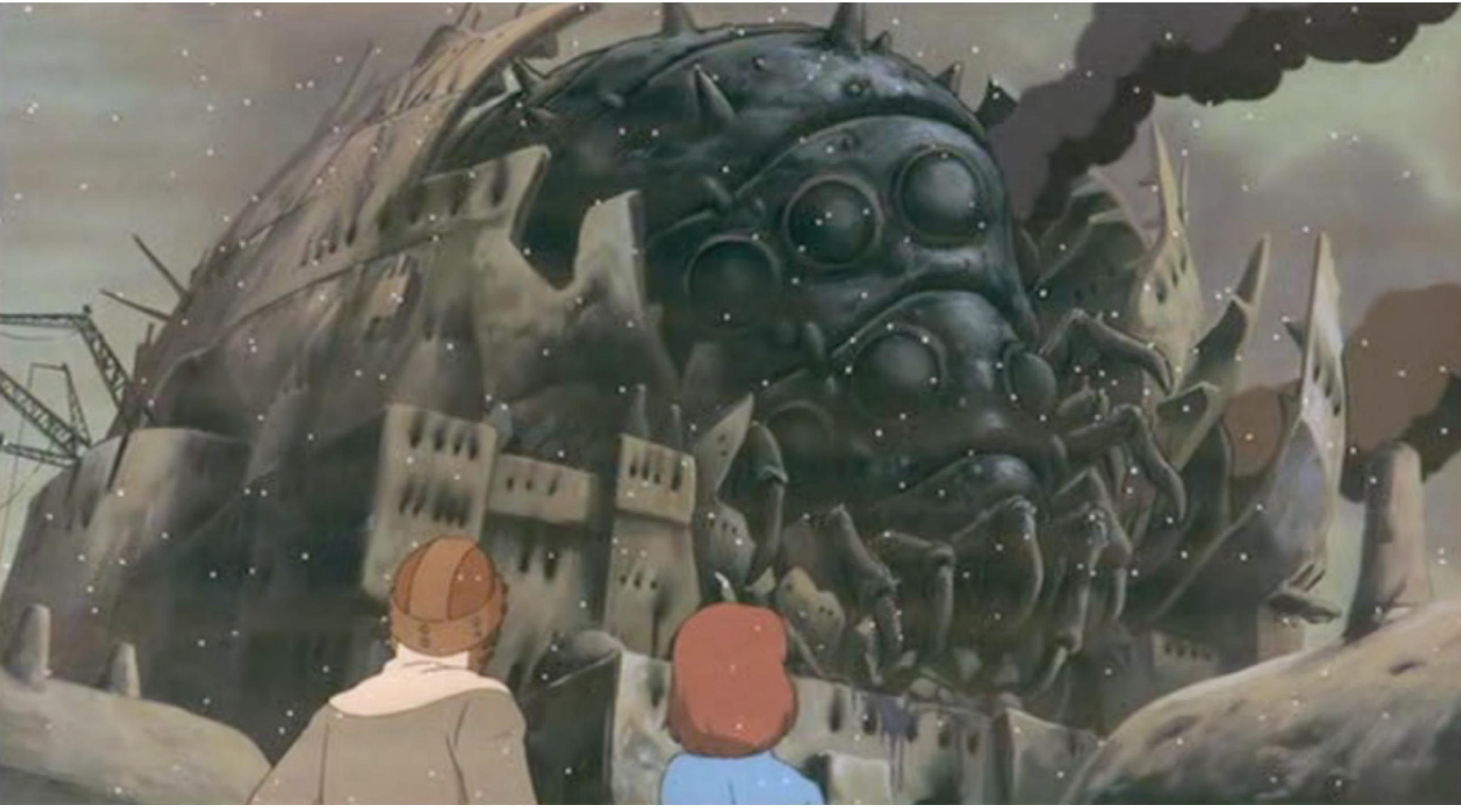
Mutability and Shared State

vs.

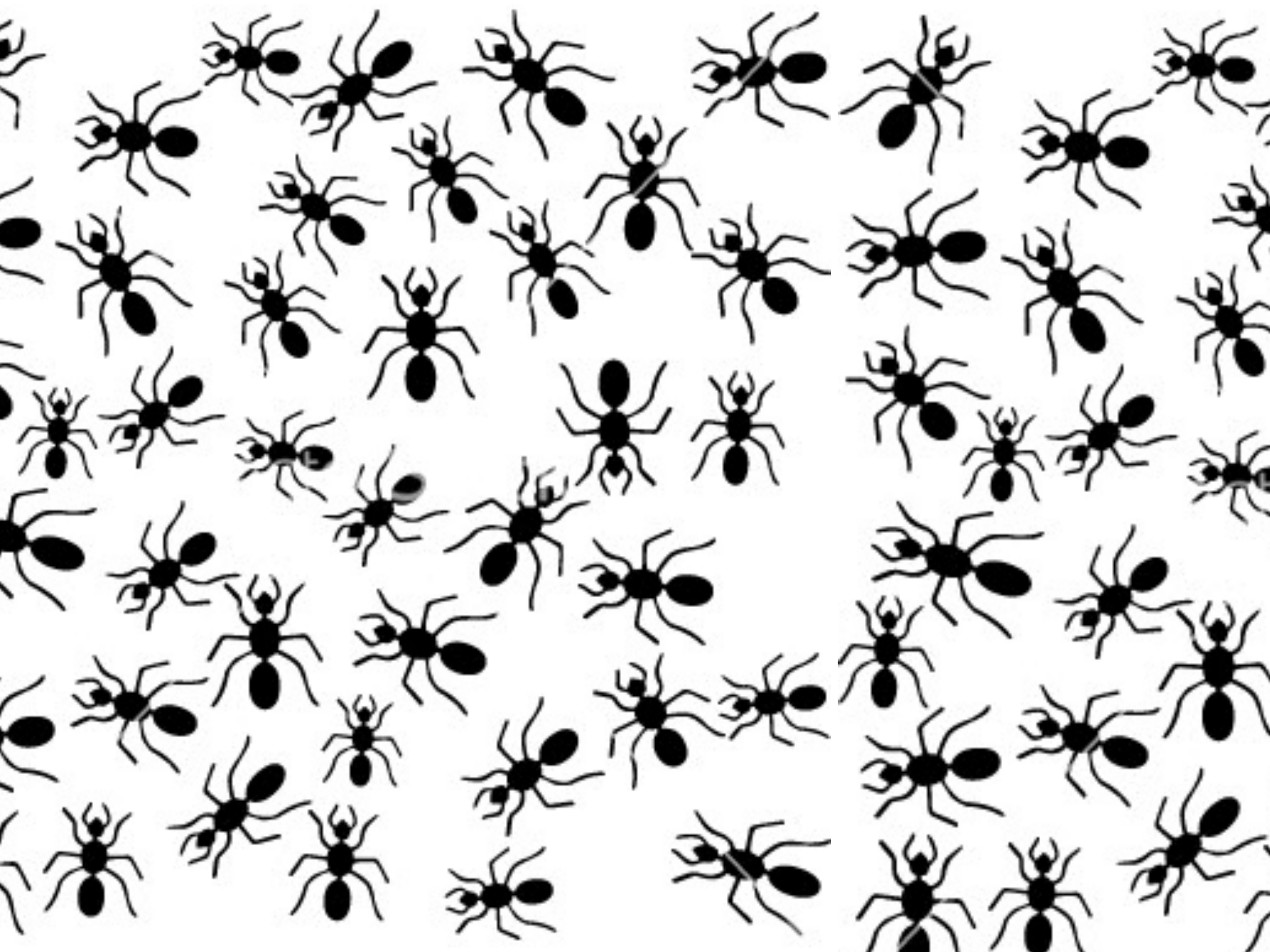
Concurrency

Fault Tolerance

Monolith vs.



Fault Tolerance



Concurrency Abstractions

Actor Model
Carl Hewitt, 1973

Communicating Sequential Processes
Tony Hoare, 1978

Akka/Scala

Erlang

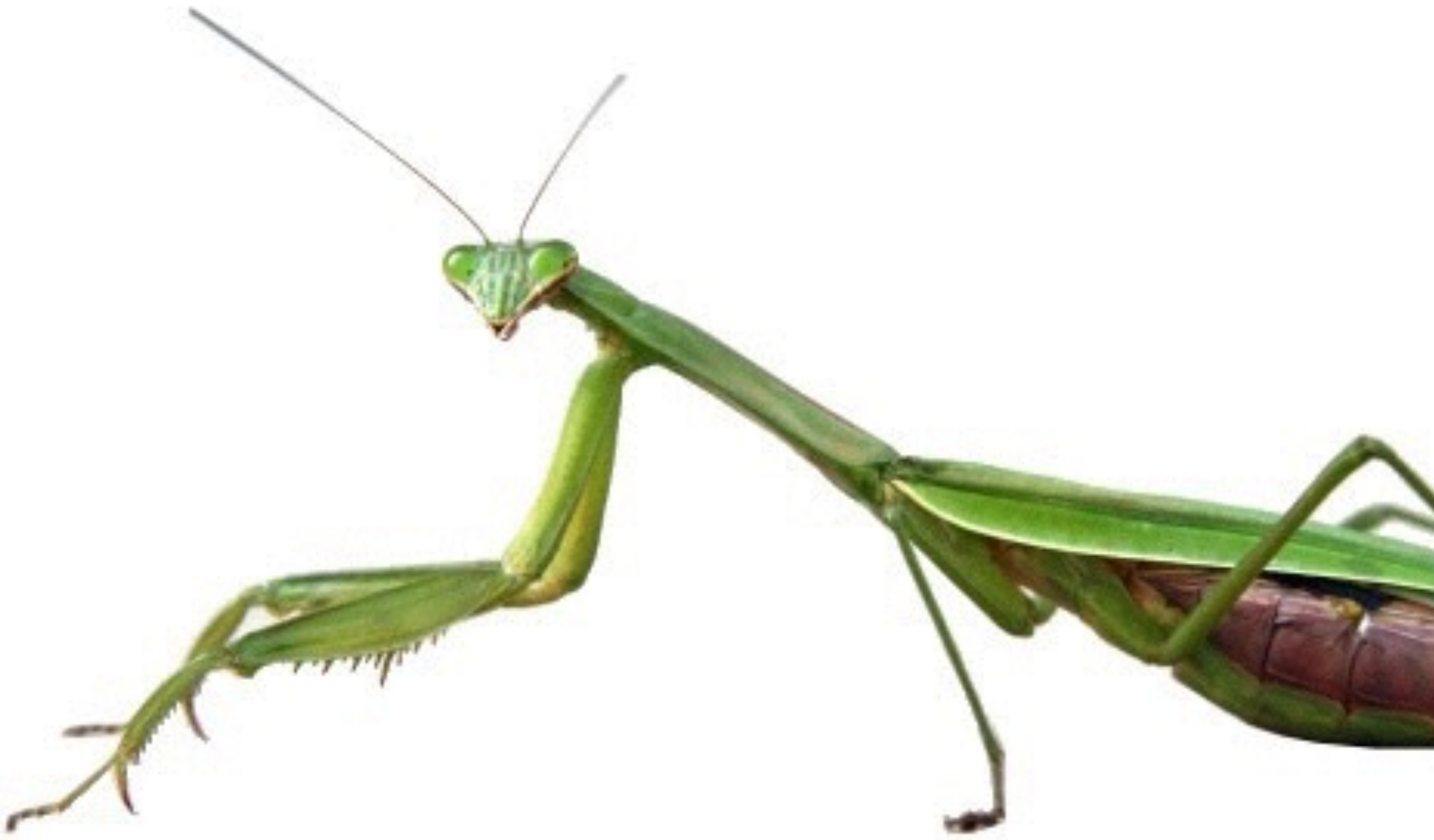
Golang

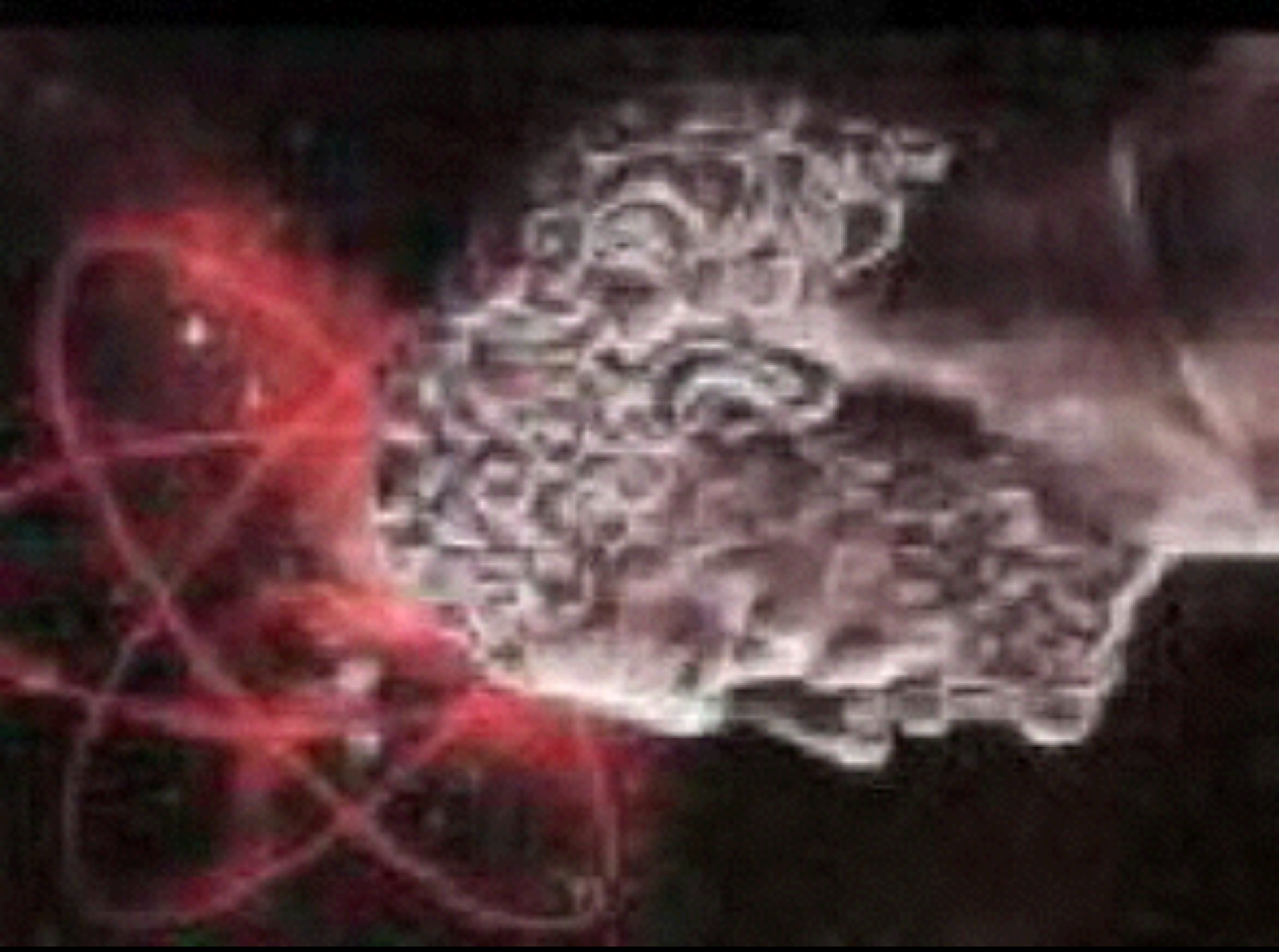
Clojure core.async

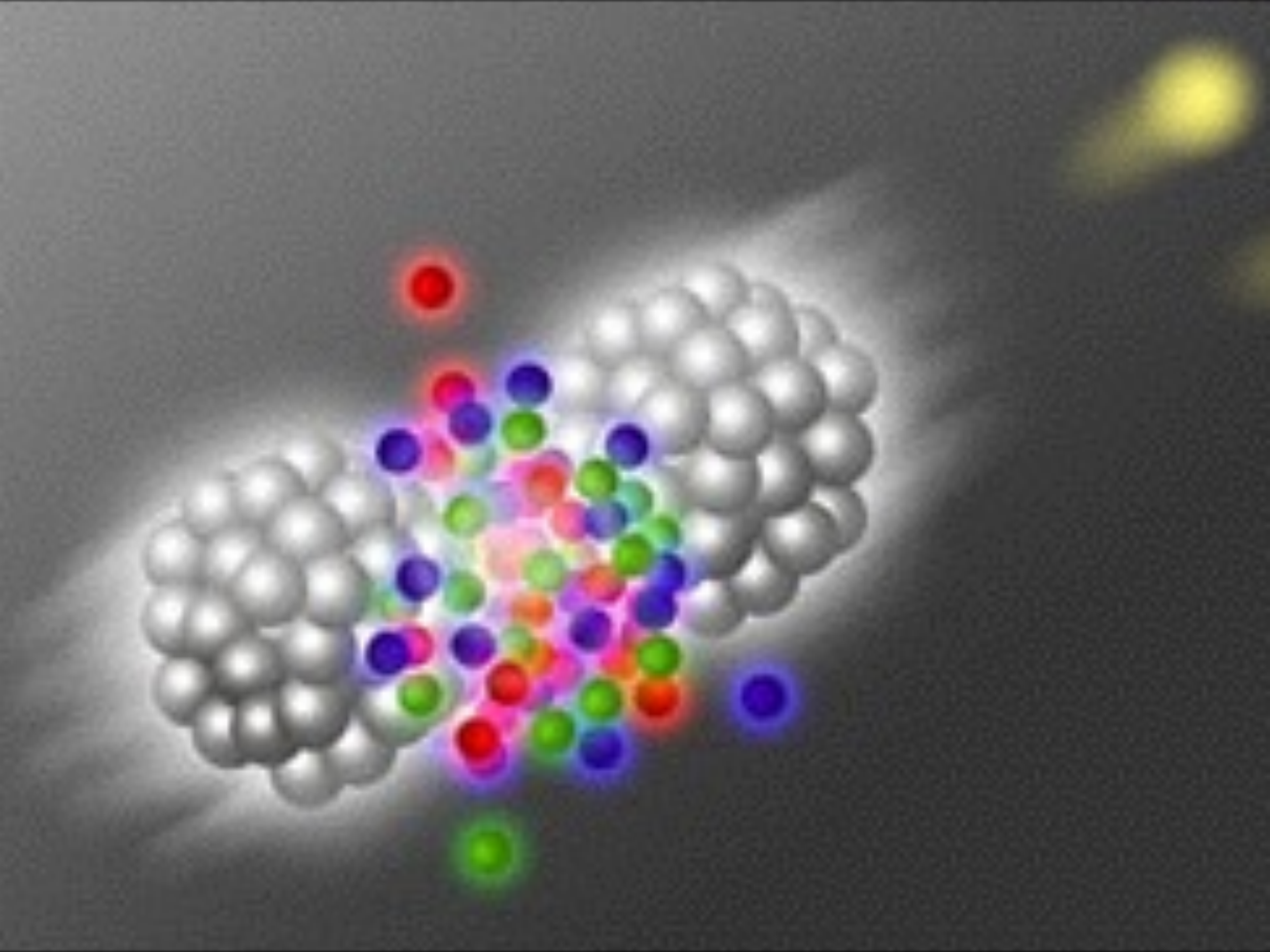
Java CSP

Erlang

binary space **atom table**



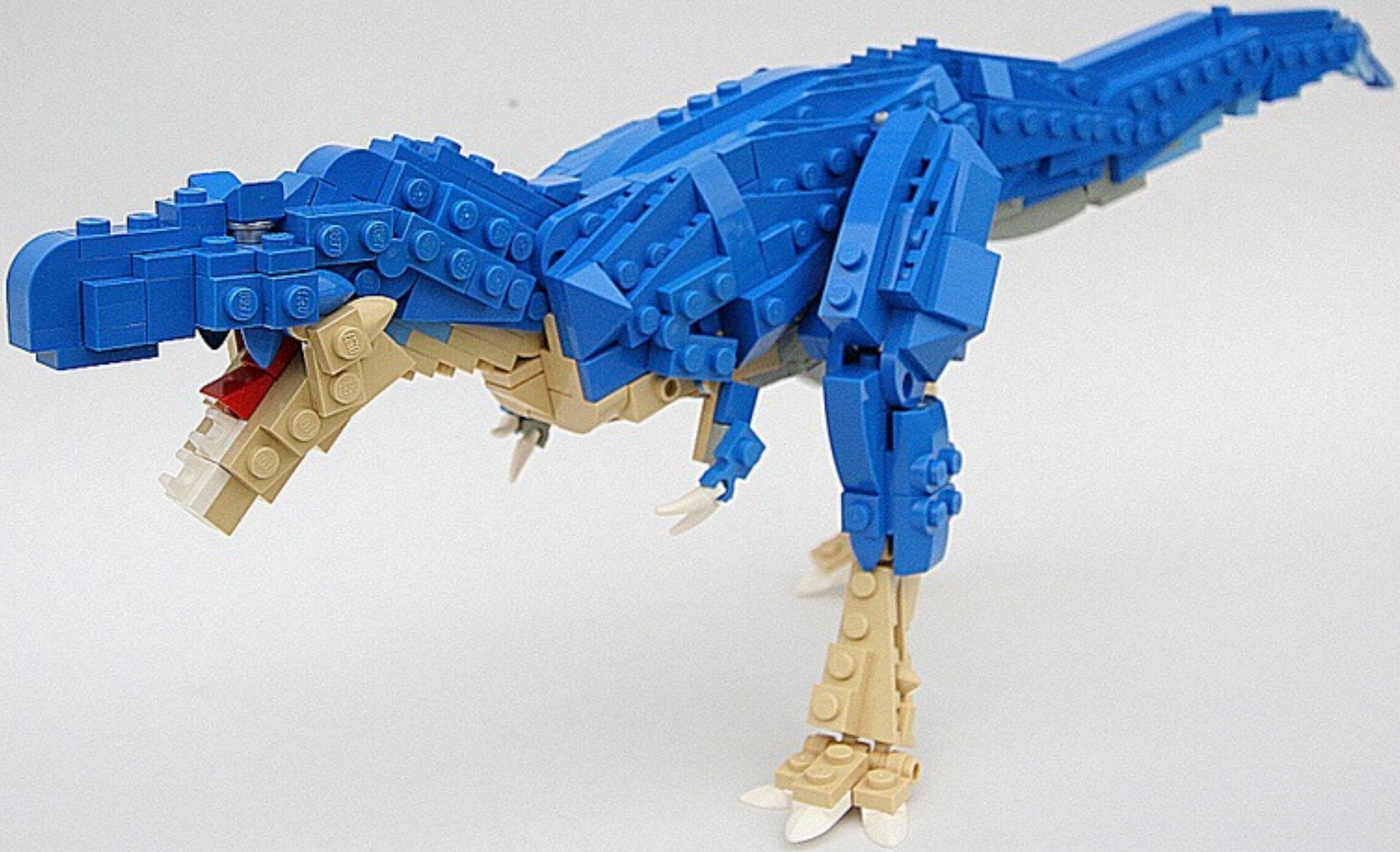


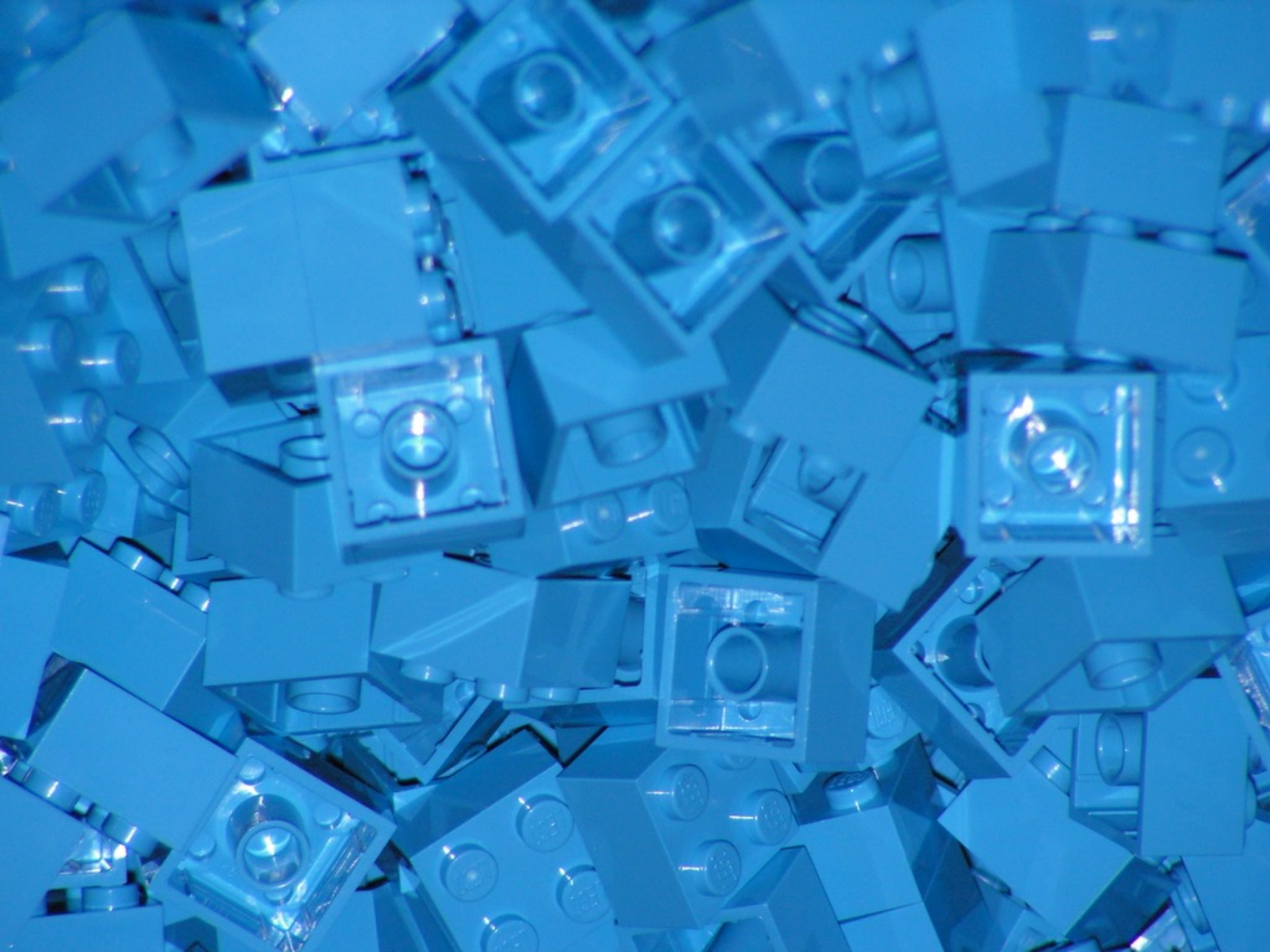


OO pattern/principle	FP pattern/principle
• Single Responsibility Principle	• Functions
• Open/Closed principle	• Functions
• Interface Segregation Principle	• Functions
• Factory pattern	• Yes, functions
• Dependency Inversion Principle	• Functions, also
• Strategy pattern	• Oh my, functions again!

http://www.slideshare.net/ScottWlaschin/fp-patterns-ndc-london2014

twitter: @ScottWlaschin





Web Frameworks.

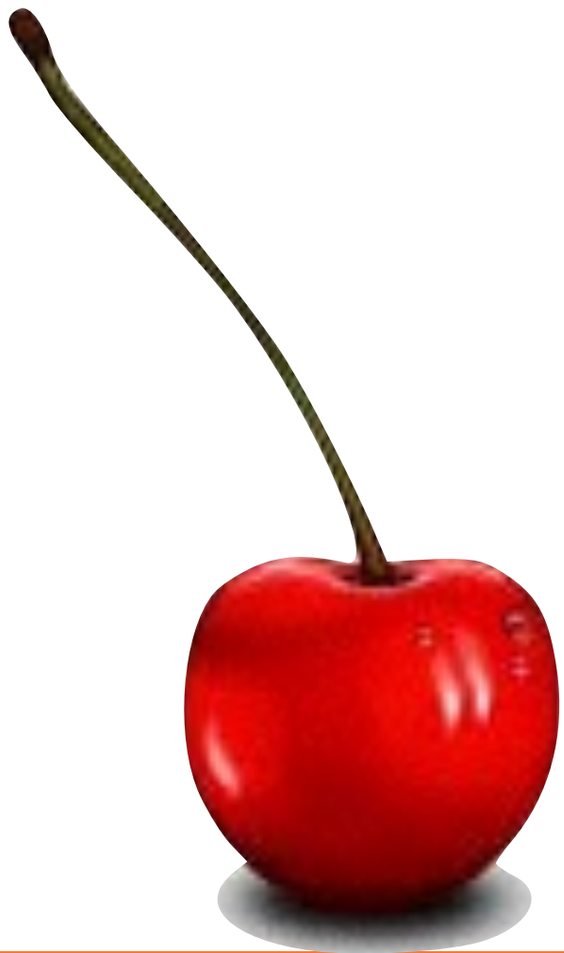
- **Ruby on Rails?** Beautiful but slow.

- **PHP?** Fast and ugly.

“…like sleeping with a short blanket.
When you cover your head,
your feet get cold.
And when you cover your feet... sigh…”

- Jose Valim

Phoenix Framework

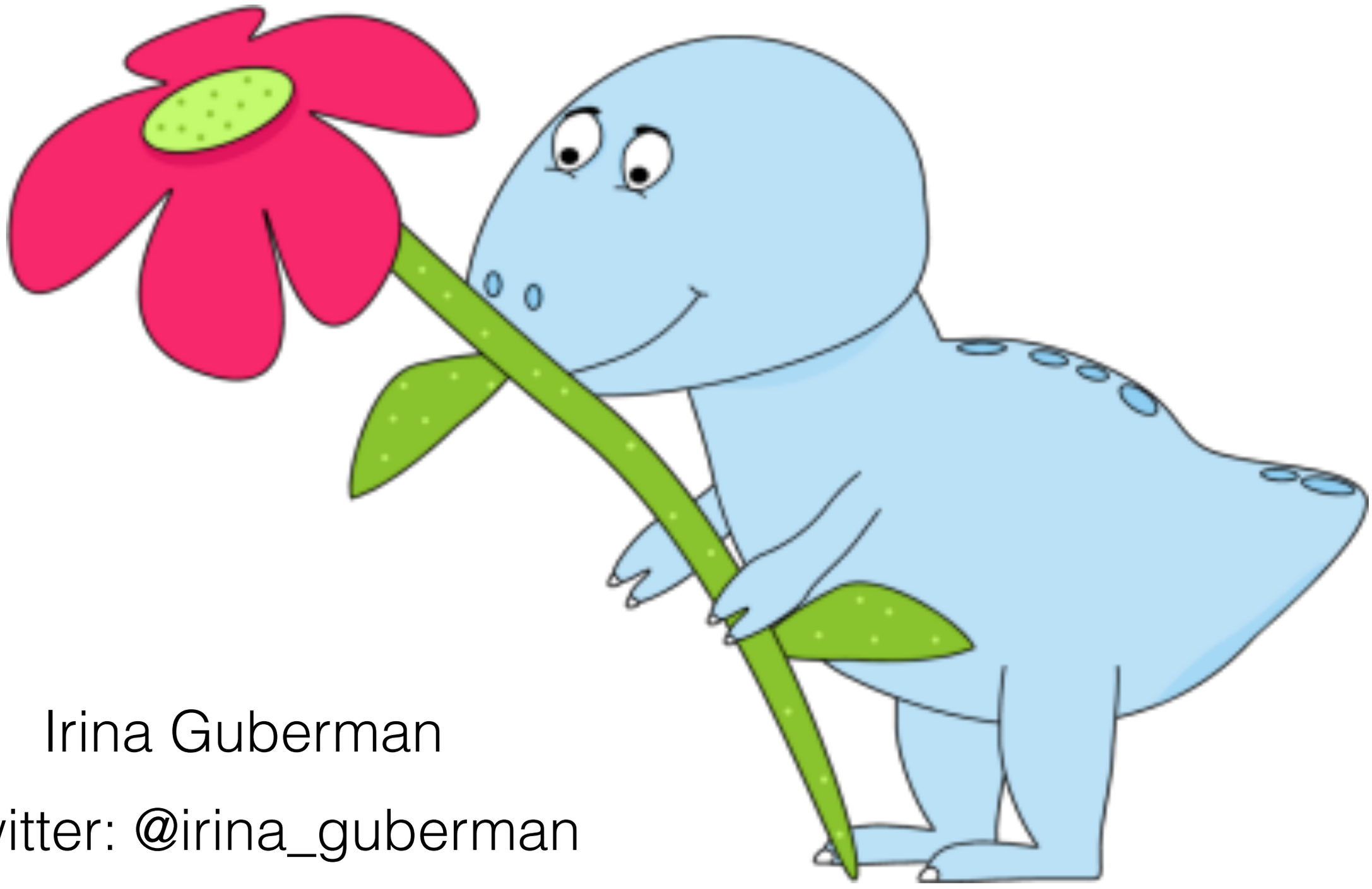


Chris McCord

Productive. Reliable. Fast.

A productive web framework that
does not compromise speed and maintainability

Compliments?



Irina Guberman

twitter: @irina_guberman