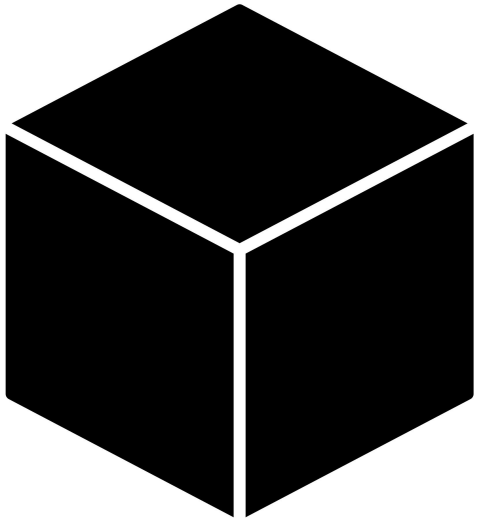
Decision making in black box scenarios

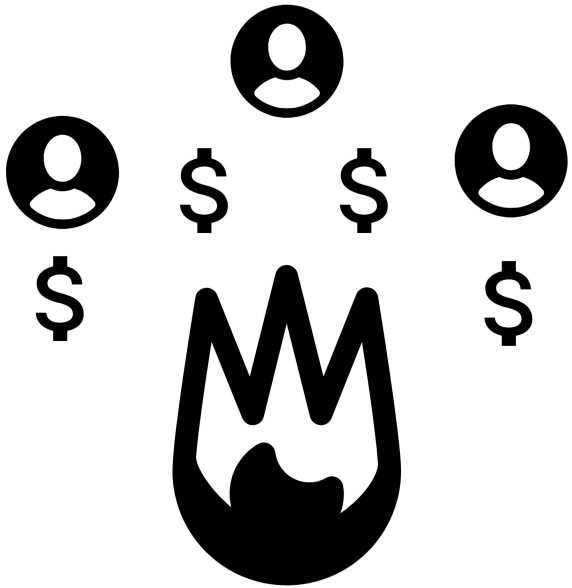
Boyan Angelov

Lead Dev Berlin



A story

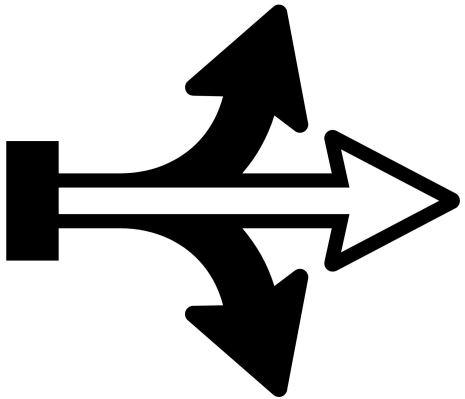
- Legacy black box
  - Complex
  - Arcane
  - No support
- Critical
  - Operational bottleneck
  - Financial repercussions
  - No backups

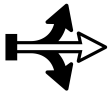


Stay tuned for the ending…

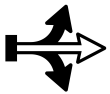
The importance of decision making

- Thousands per day
- Relative importance
- Stacking up
- No decision is a decision
- Consequences

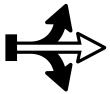




Deciding to select a tech stack

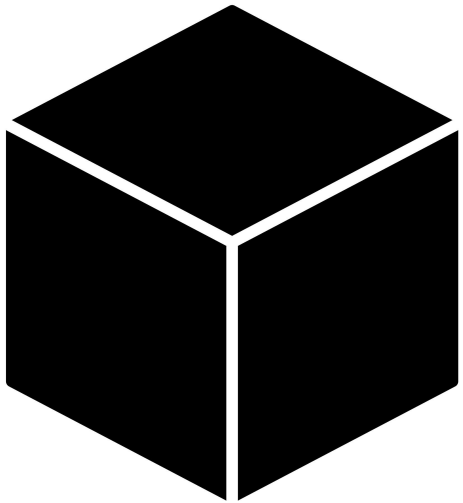


Prioritizing a task or a project



Deciding to hire

Defining black boxes

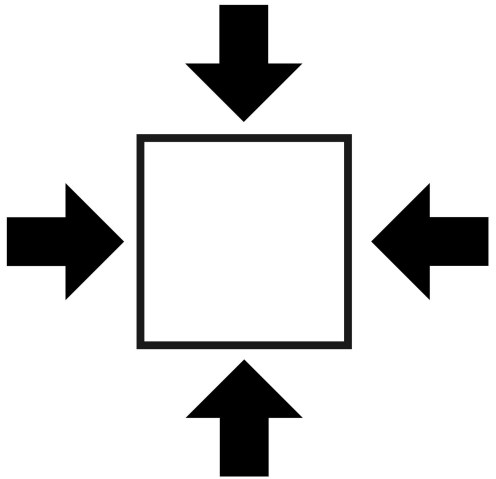


Any system that we can’t understand fully under constraints.

- Legacy systems
- Complex systems
- Large teams

The problem: constraints

With unlimited time, we can understand *any* black box and make a perfect decision.

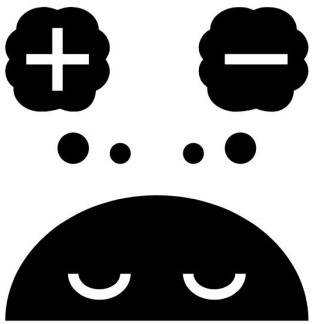


- Knowledge
- Budgets
- Time

Solution: two new types of thinking

Engineer

Our normal mode of operation.



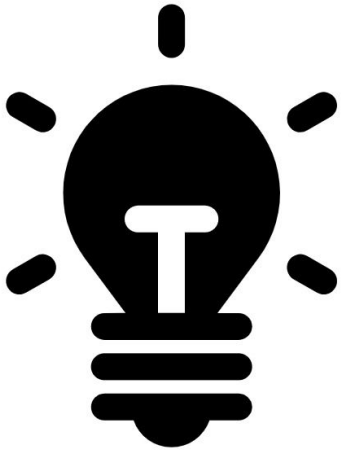
Zen monk

- Leave it be
- Wait
- Don’t rush
- Respect the system

Scientist

- Have a scientific approach: conduct controlled experiments (i.e. switching on and off of components)
- Recognize boundaries
- Recognize relationships and feedback loops
- Measure and change inputs and outputs
- Replicate the system

How the story ended



- With all those methods we saved the day
- It remained a black box, but under control
- Moved to a new system

Parting thoughts

Become a **systems thinker** (zen monk + scientist)!

It’s a skill that can be learned.

Happy unboxing!