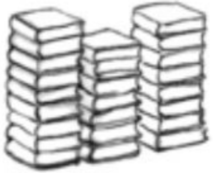
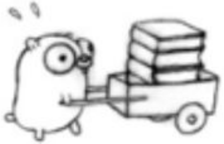
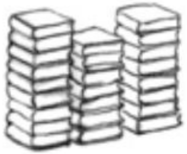
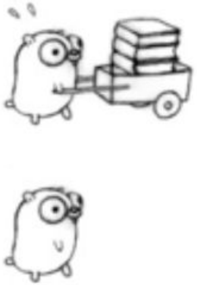
Microservices with Go

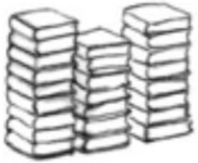
Concurrency & Parallelism

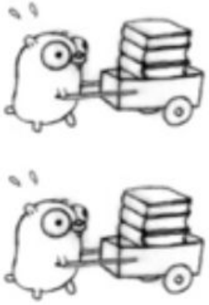




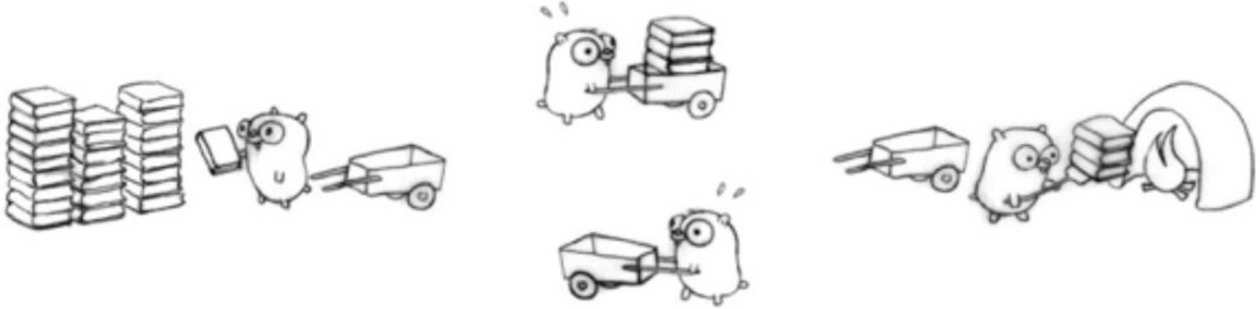


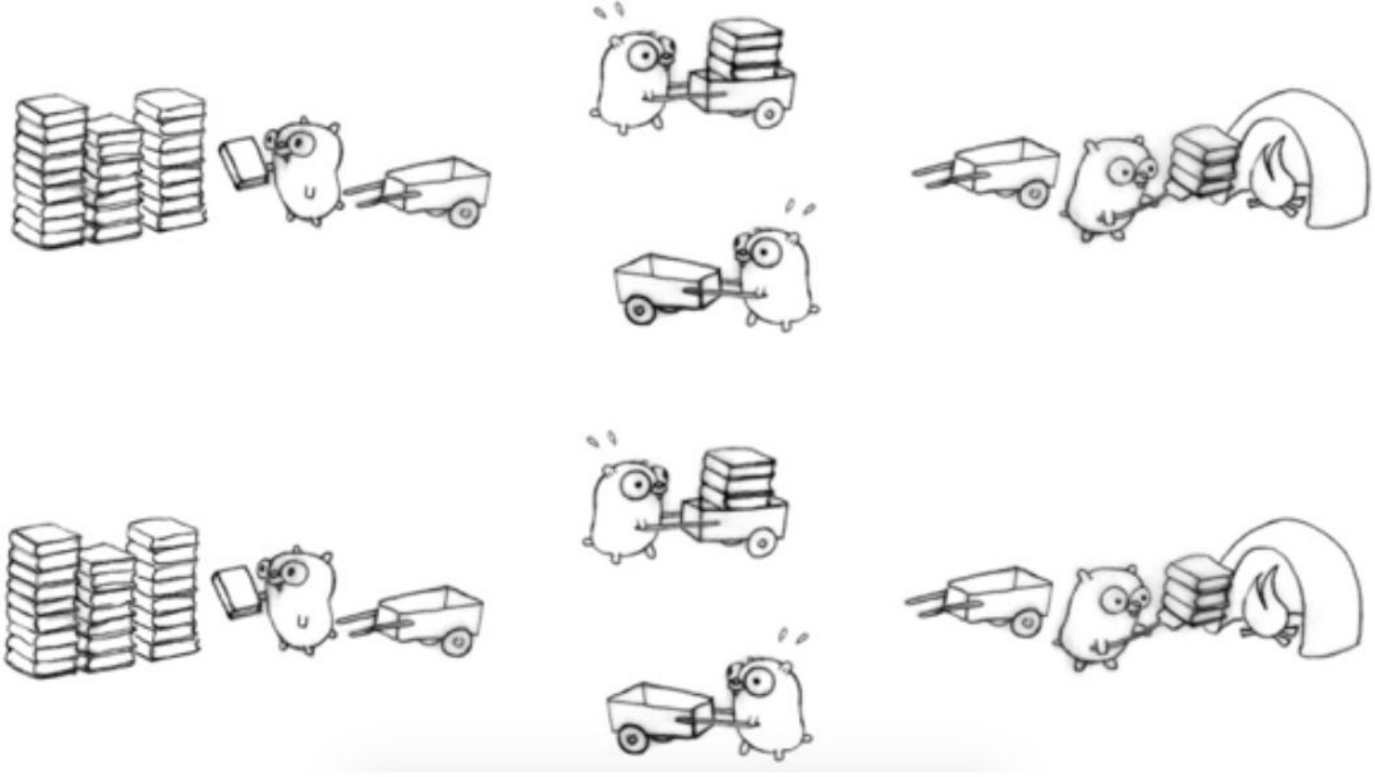




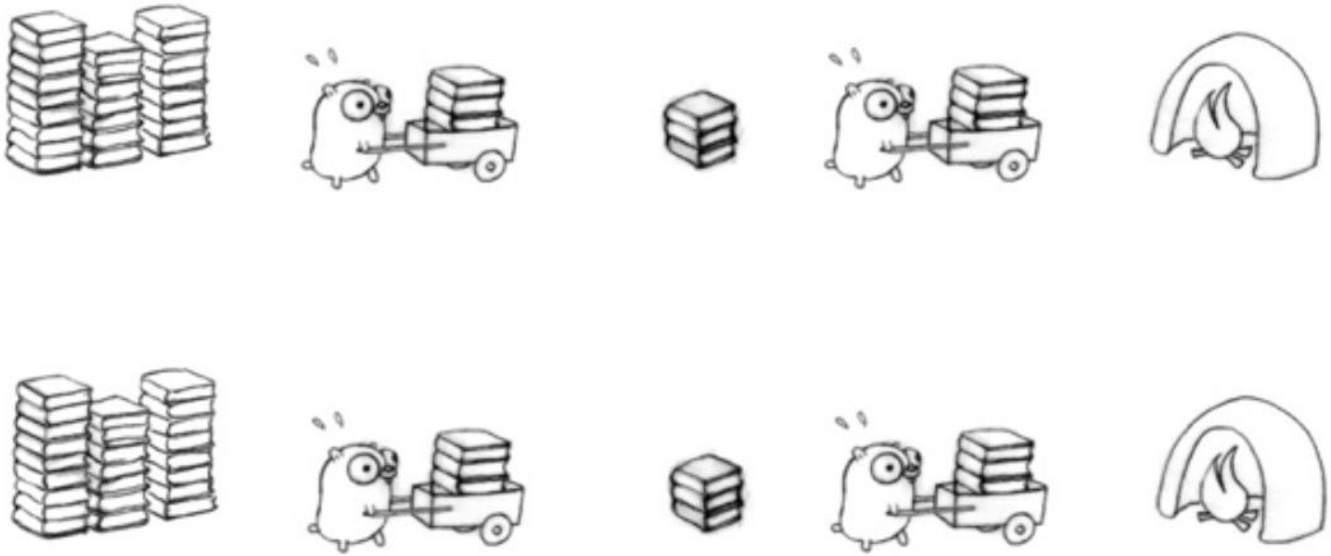


Another design

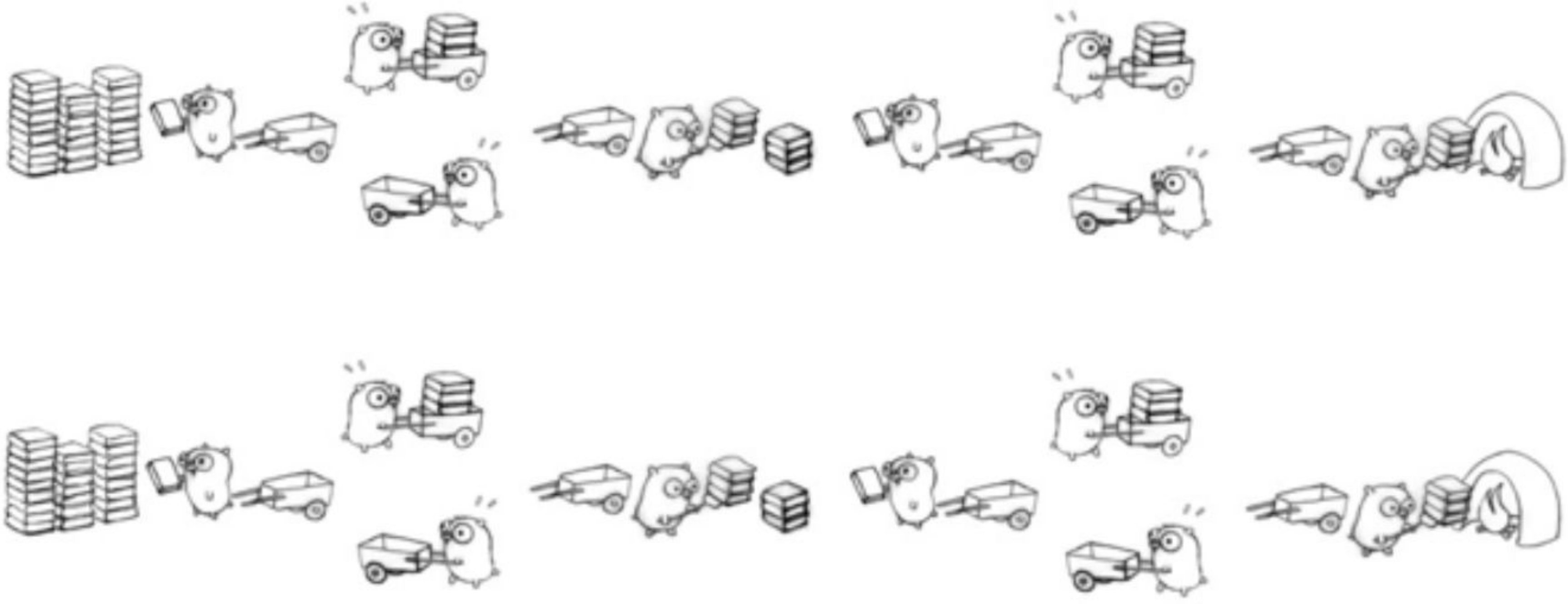




Another design



¿What if we mix them all?



Final concepts

- Books: Content (json, xml, files, signals, etc)...
- Gopher: CPUs.
- Wheelbarrow: Network traffic, IO signals, etc.
- Oven: Frontends, other APIs, native mobile apps, etc.

Final concepts

- **Concurrency**
  - Composition of processes running independently.
  - It is based on the **structure** of the solution.

- **Parallelism**
  - **Simultaneous execution** of processes, (may or may not be related).
  - It is about the **execution** of the solution.

Concurrence is a way of structuring a solution, breaking it down into parts that can be executed independently and that converge into a final result.

A concurrent solution can execute its parts in parallel <u>only if there is more than one processor</u>.