it's time to letgo

Avoiding the Tower of Babel

Effectively evolving events in micro-service ecosystems

Sebastián Ortega

Carol H.
Hey! I love the bike and I'd like to buy it. How does $25 sound?

Me
Tell you what... Let's leave it at $30 and I'll throw in a bike lock

Founded in 2015

Phenomenal growth

80 MM+ downloads

Focused in USA and Turkey

Offices in Barcelona and NY

SPLIT THE MONOLITH!

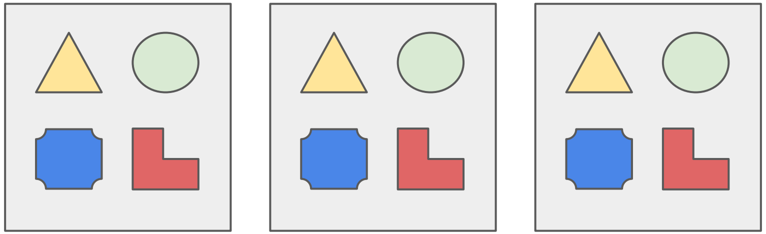
events at letgo by
Rafa Gómez

A µ-service is...

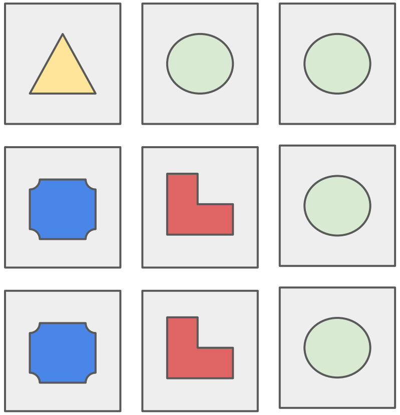
- it’s own process
- small
- independently deployed
- networked
  - API calls
  - pub/sub of events

Are service architectures scalable?

- No intrinsic advantage: you can distribute your monolith

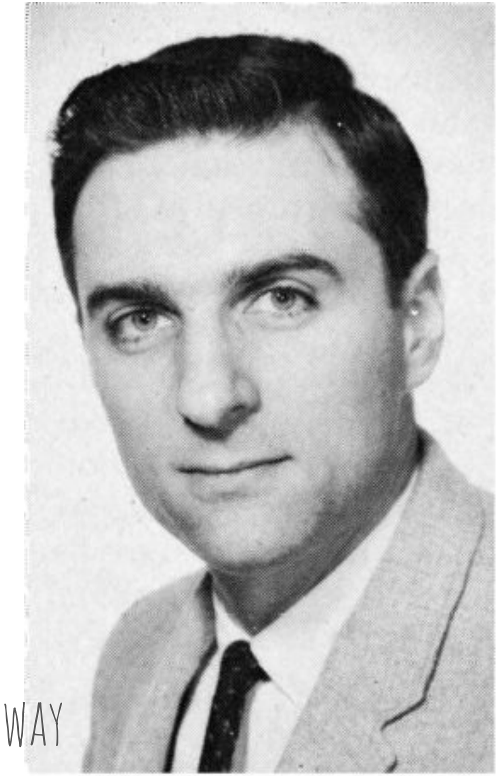


- Except for having more **freedom** to scale independently each service



Are service architectures scalable?

- **Social scalability**: the true scalability property of services
- Conway's law in reverse
- Squads: team with **all the roles** and empowered through **independence**
    - Prefer service **choreography** to service **orchestration**



Melvin Conway

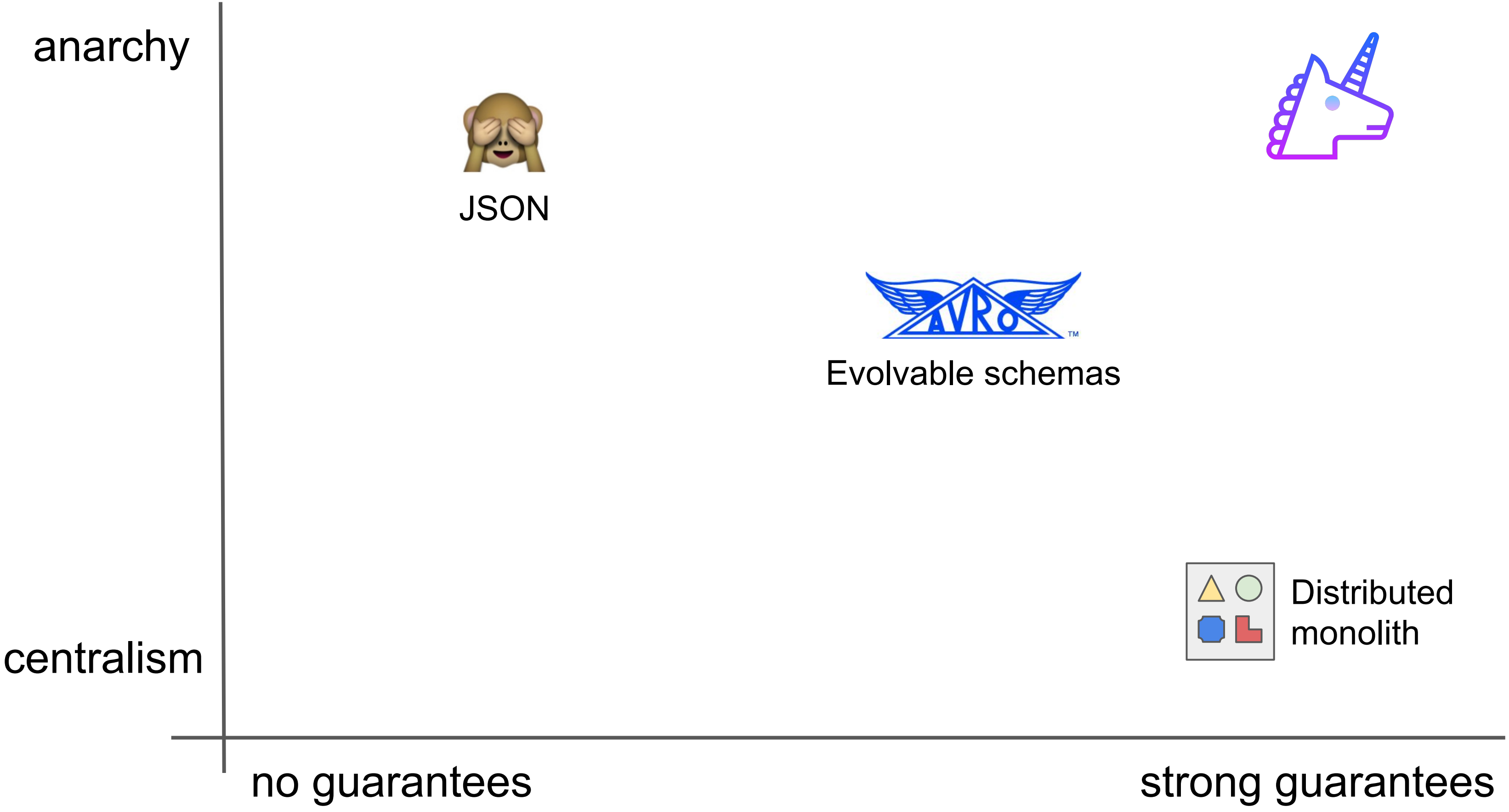
Is this for free?

- NOPE
- more complex to operate
- bigger footprint
- prone to (excessive) heterogeneity
- challenging to manage change at the boundaries

Don't even consider microservices unless you have a system that's too complex to manage as a monolith.

—Martin Fowler

How to avoid Babel without curtailing independence?



anarchy

centralism

no guarantees

strong guarantees

JSON

Evolvable schemas

Distributed monolith

Apache AVRO

- Serialization protocol
- Schema-based
- Binary
  - Fast parsing!
  - Compact
  - Tools to inspect :(
- Polyglot: java, scala, js, python, C#...
- Evolvable schemas!

```
{
 "type": "record",
 "namespace": "com.tidx",
 "name": "ProductVisit",
 "fields": [
   {
     "name": "product_id",
     "type": "string",
     "doc": "Identifier of
the visited product"
   },
   ...
```

Evolvable schemas

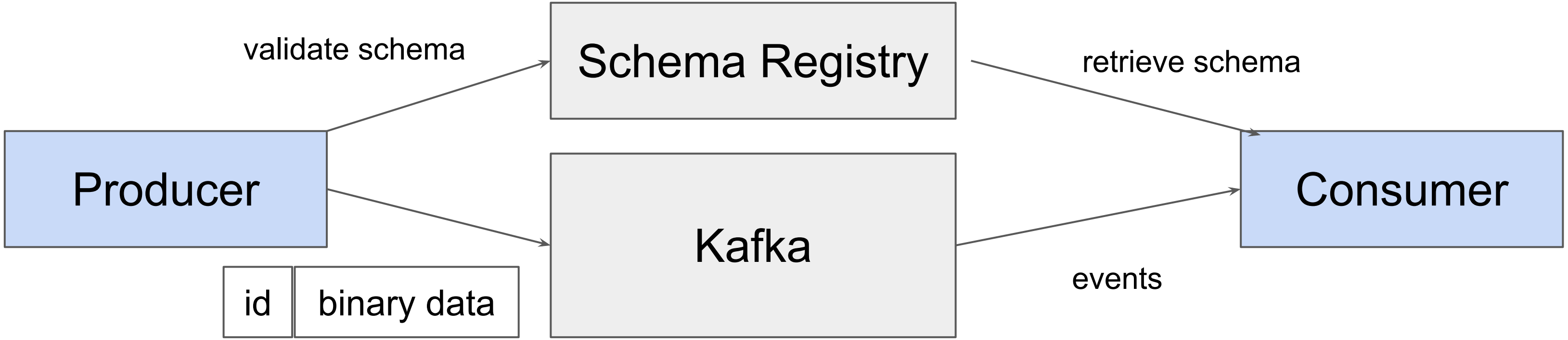
- Not every schema will be easy to evolve
- Keep required fields to the minimum
- Provide default values when possible
- Introduce only optional/with defaults fields
- Let’s see an example

Apache Kafka

- Distributed commit log open sourced by LinkedIn
- Its creators founded Confluentic to provide commercial support
- Kafka offers
  - exactly-once delivery of events
  - high-throughput (millions of events per second)
  - high-reliability
- How?
  - for kafka, payloads are just byte arrays
  - no parsing, zero-copy optimizations
- So, Kafka is blind regarding schemas and schema validation…

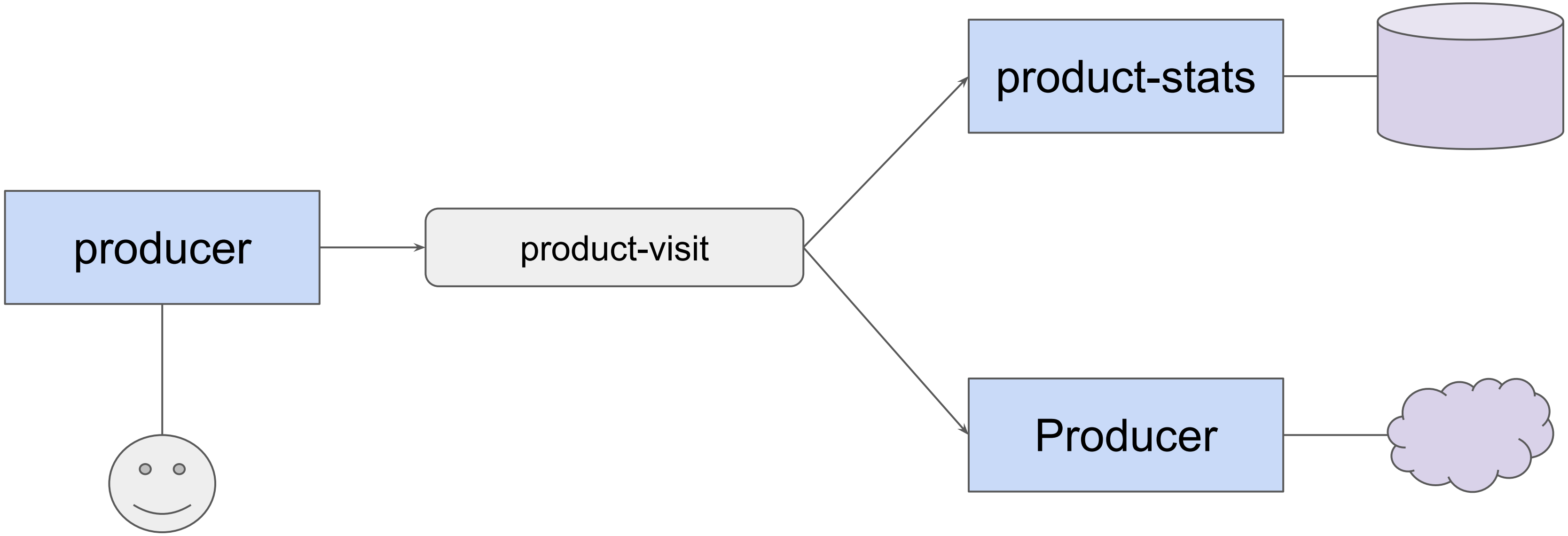
Kafka - Schema Registry

- Component developed by Confluentic (also OSS)
- Used at the edges of Kafka
- Enforces schema compatibility: **no need to cut red tape**

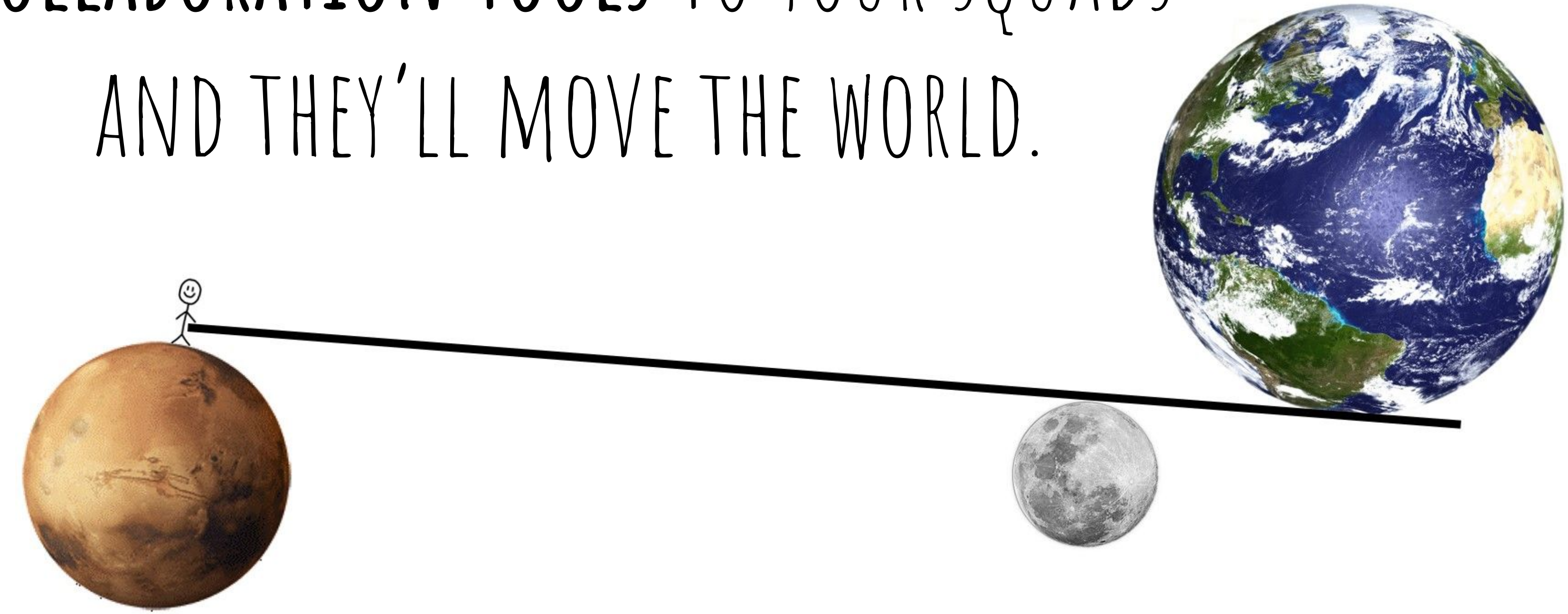


Demo time

code at github



Give **autonomy** and the **best collaboration tools** to your squads and they'll move the world.



Thank you

@_sortega

(letgo is hiring)
we.letgo.com/contact