Building the Stonehenge

Using Gall’s law

by Fabricio Buzeto

Fabricio Buzeto

a.k.a Fabs

In love with code since 2002

Entrepreneur since 2005

Researcher since 2008

Startuping since 2011

CTO @ **bxblue**

Gall’s Law

Gall's Law

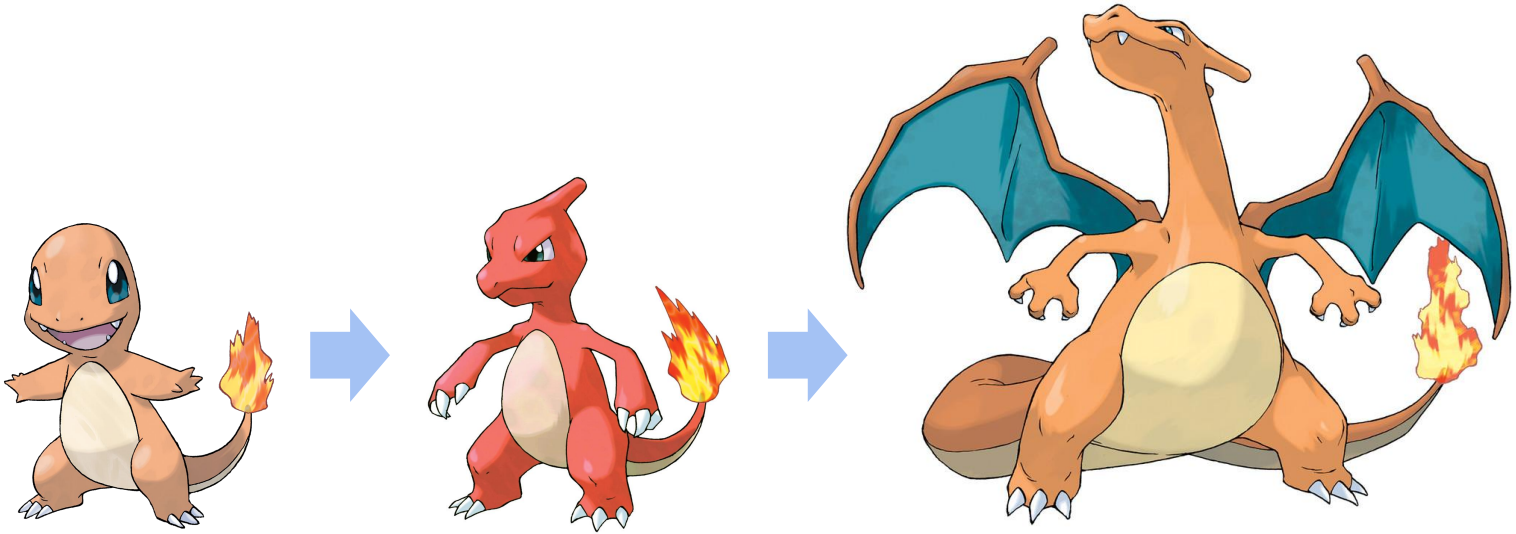
"

A complex system that works is invariably found to have **evolved** from a simple system that worked.

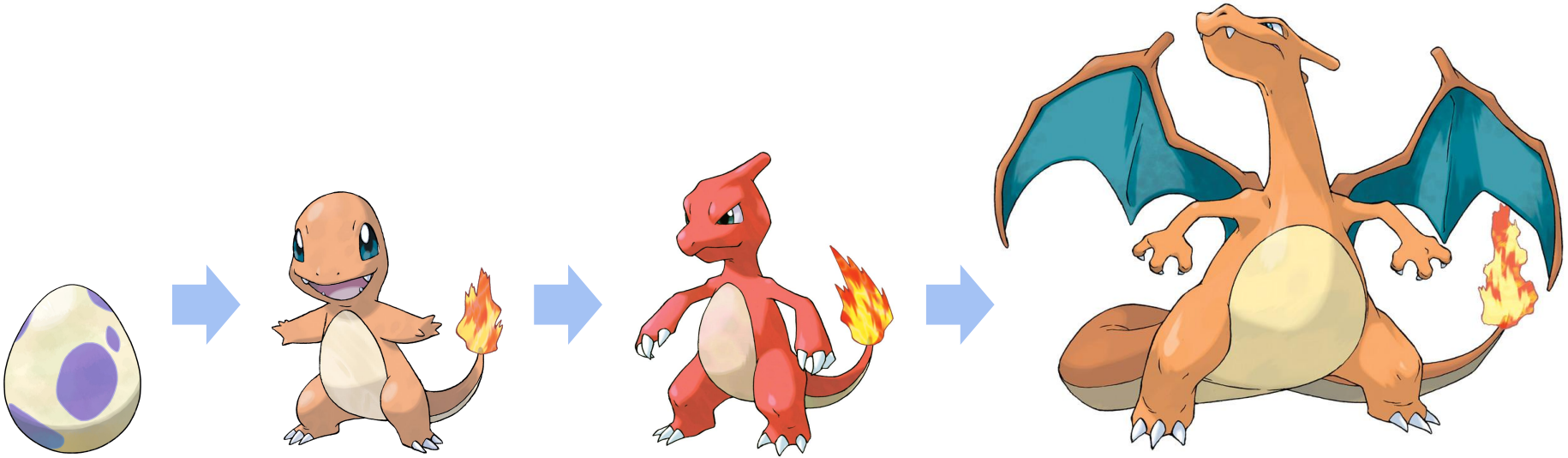
"

John Gall - 1975
"Systemantics: How Systems Really Work and How They Fail."

How complex systems evolve?



How complex systems evolve?



Gall's Law

"

A complex system that works is invariably found to have evolved from **a simple system** that worked.

"

John Gall - 1975
"Systemantics: How Systems Really Work and How They Fail."

Game of Life

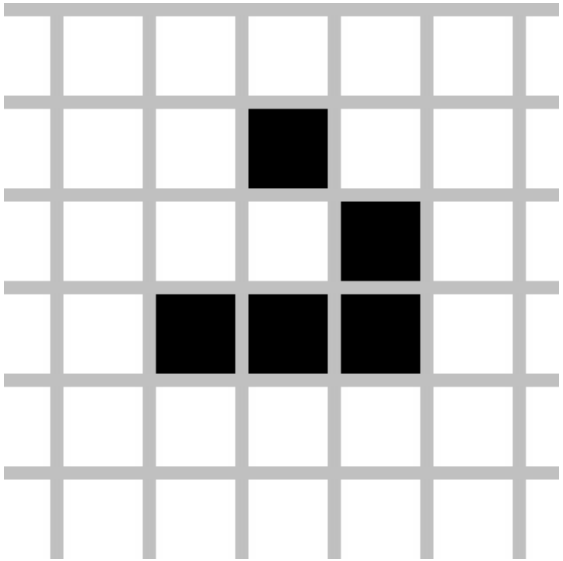
John Conway - 1970

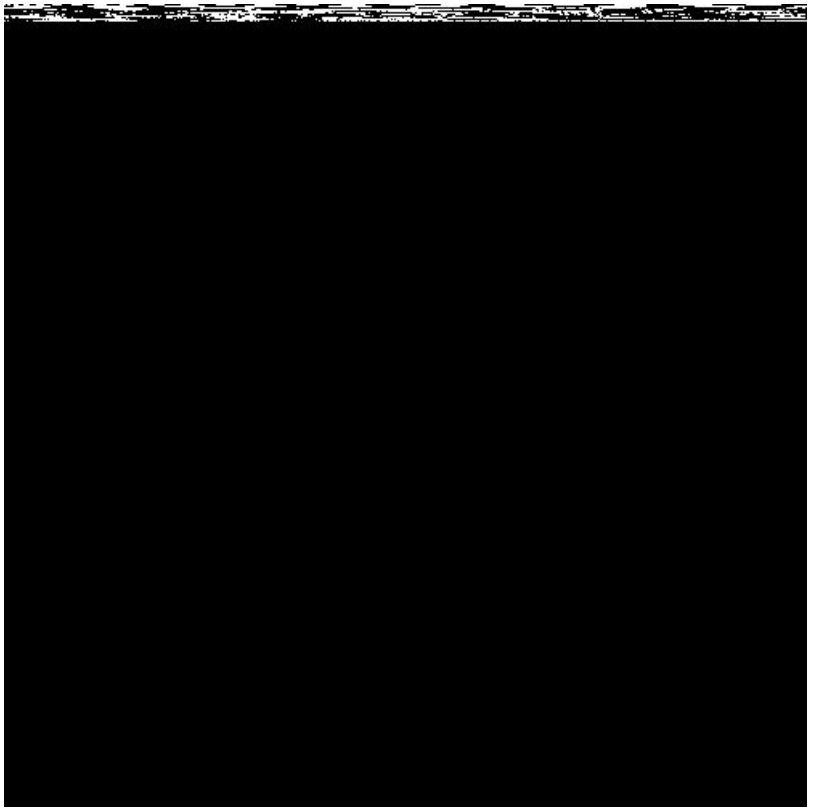
1. Any living cell with 2 or 3 neighbors survives

2. Any dead cell with 3 living neighbors comes to life

3.Any remaining living cell dies

Simple rules can lead to complex behaviors





Gall's Law

Continuing

John Gall - 1975
"Systemantics: How Systems Really Work and How They Fail."

"

A simple system **may or may not** work.

"

What “works” means?

What's "working"

What a working software does?

A software that **works**, is a software that fulfil its **purpose**

What’s Purpose

For what?

For what purpose?

- Business ←
- Politics
- Learning

Whos is the Client?

Who we need this?

Who needs it?

- The User ←
- The Sponsor
- The Company
- The Team

Gall's Law

Continuing

John Gall - 1975
"Systemantics: How Systems Really Work and How They Fail."

"

A complex system designed from scratch never works and **cannot be patched up to make it work.** You have to start over with a working simple system.

"

Starting simple


Qual Canal
?

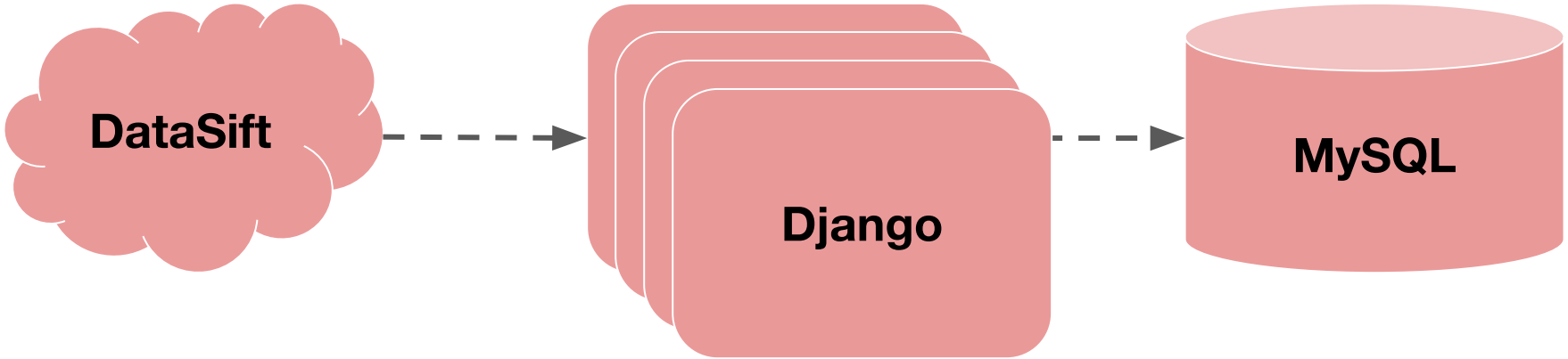
Qualcanal the first years

AVG 1k Twets/day
Peak 1k Tweets/hour

Qualcanal the first years

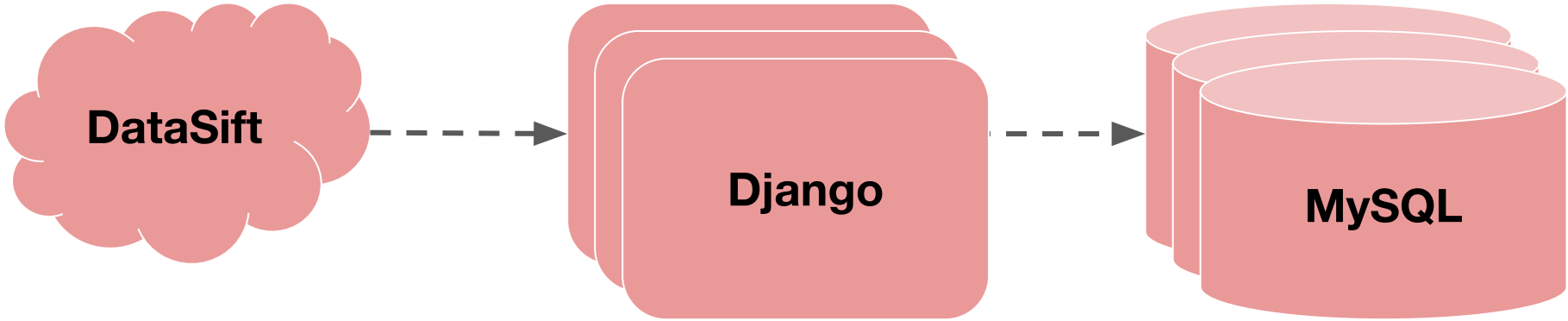
AVG 10k Twets/day
Peak 5k Tweets/hour

Qualcanal the first years



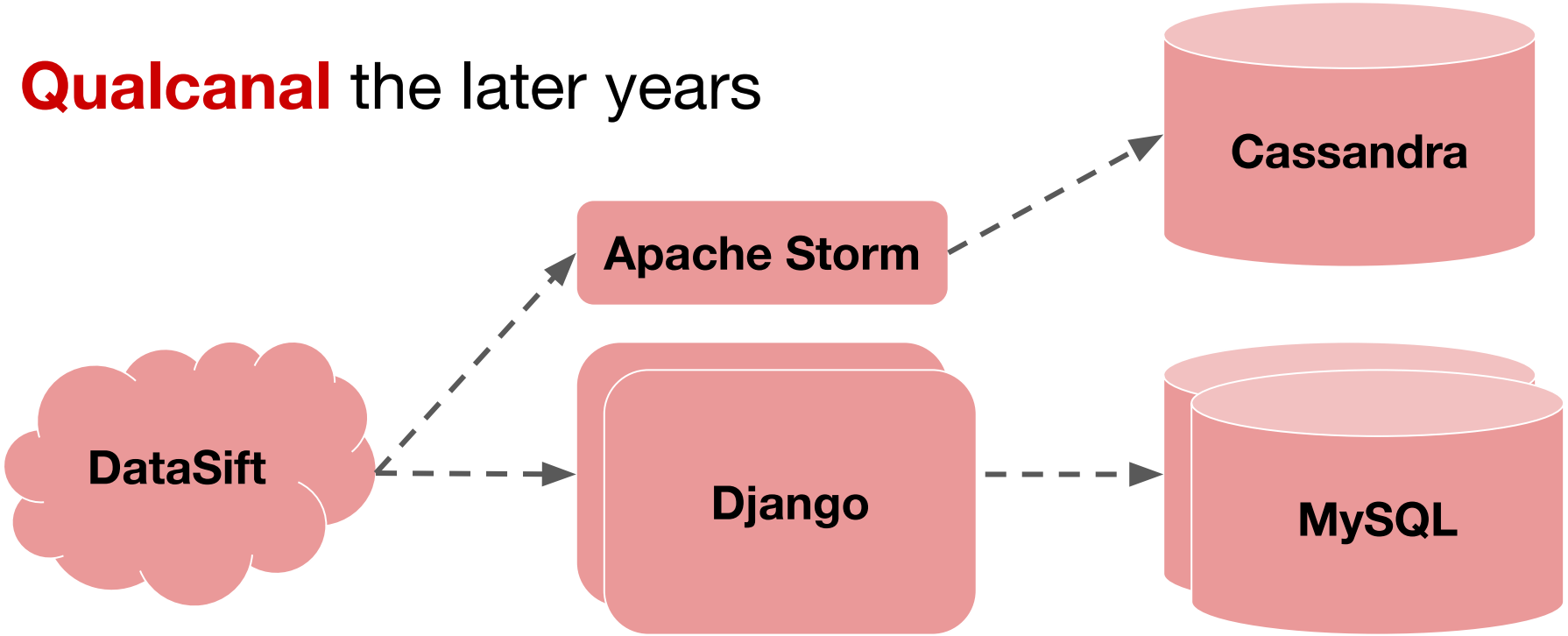
AVG 50k Twets/day
Peak 5k Tweets/minute

Qualcanal the first years

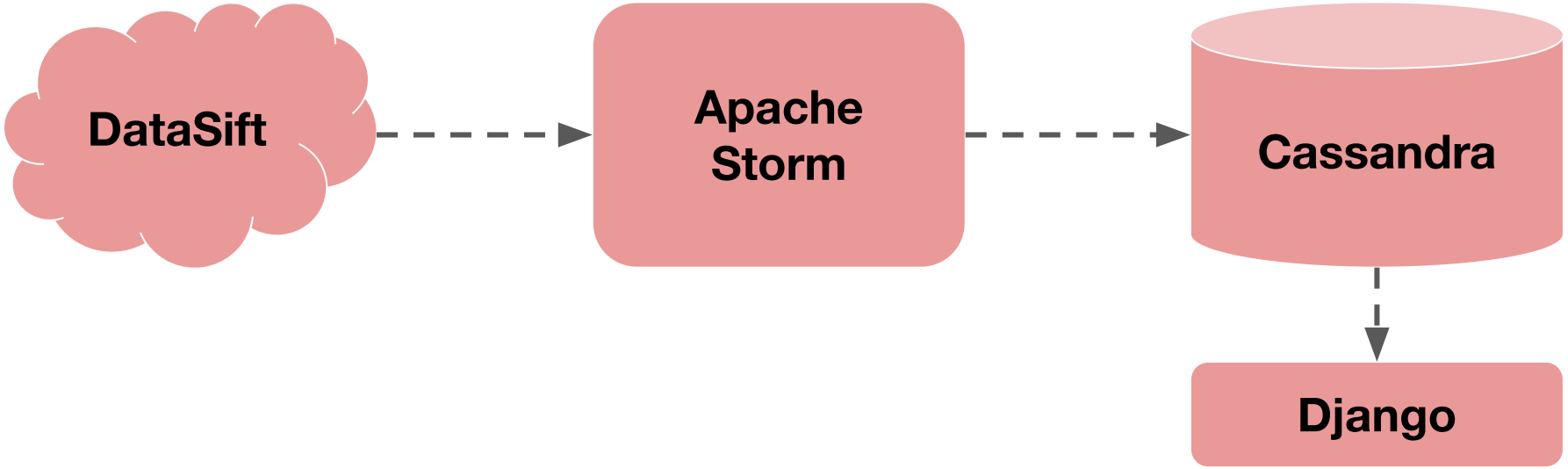


AVG 100k Twets/day
Peak 15k Tweets/minute

Qualcanal the later years



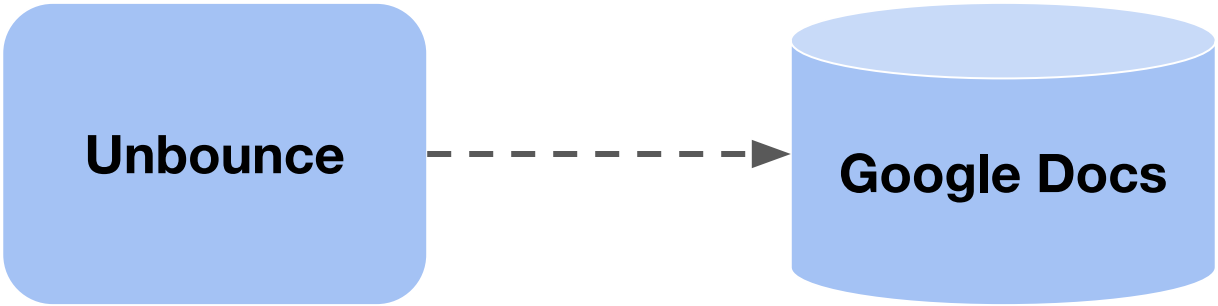
Qualcanal the later years



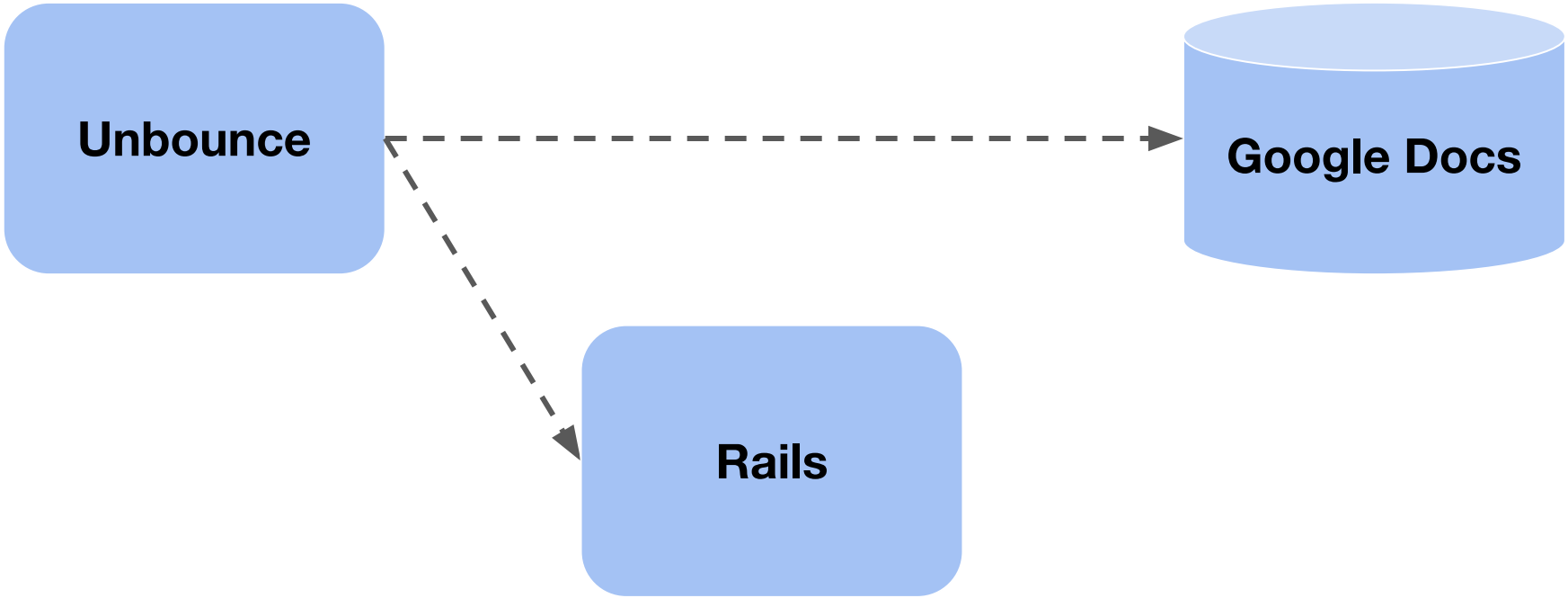
AVG 1M Twets/day
Peak 50k Tweets/minute

bxblue

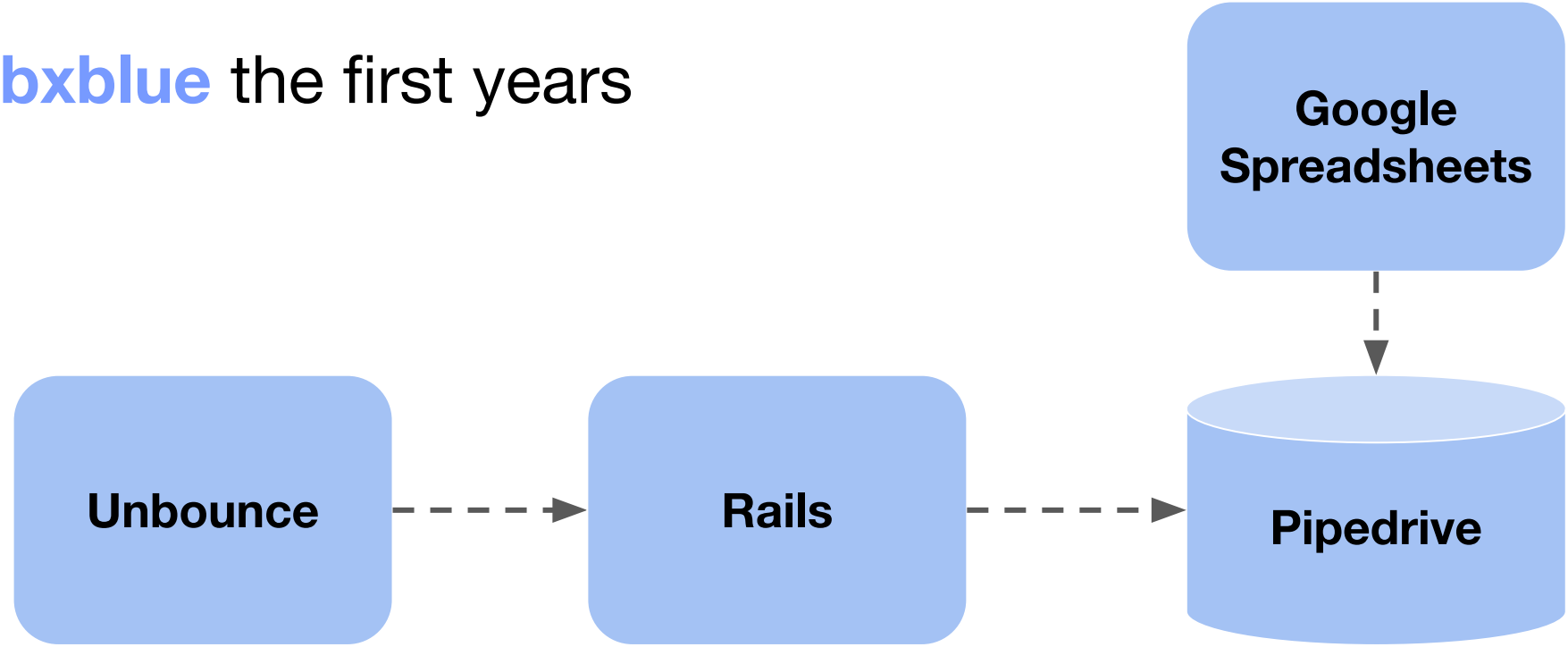
bxblue the first years



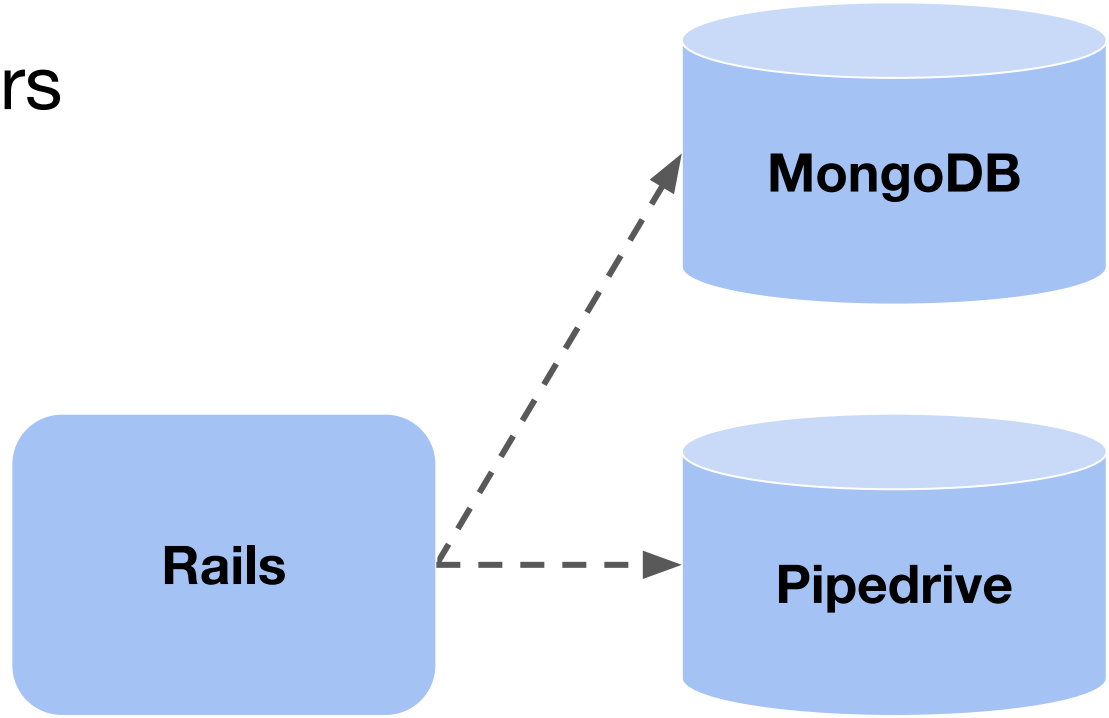
bxblue the first years



bxblue the first years

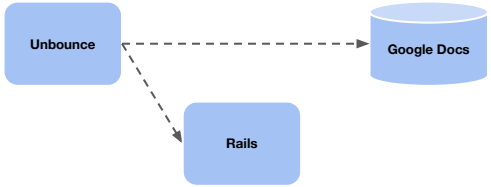


bxblue the first years

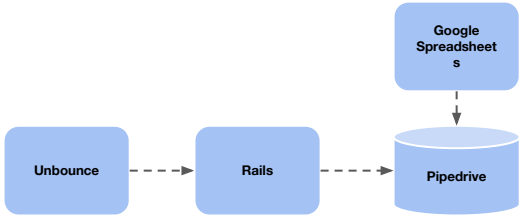


bxblue the first years

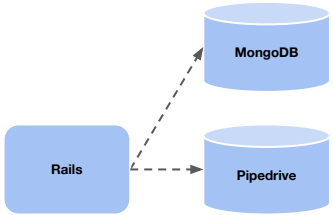
Will users want migrate their loans?



How do loans are sold?



How can I do it faster?



How can I do it properly?

The monolith

Why?

The Utah monolith - November 2020

The monolith

Simple like that

Simple to Develop

Simple to Test

Simple to Deploy

Simple to Reuse

Simple to Scale*

Chris Richardson
https://microservices.io/patterns/monolithic.html

The monolith

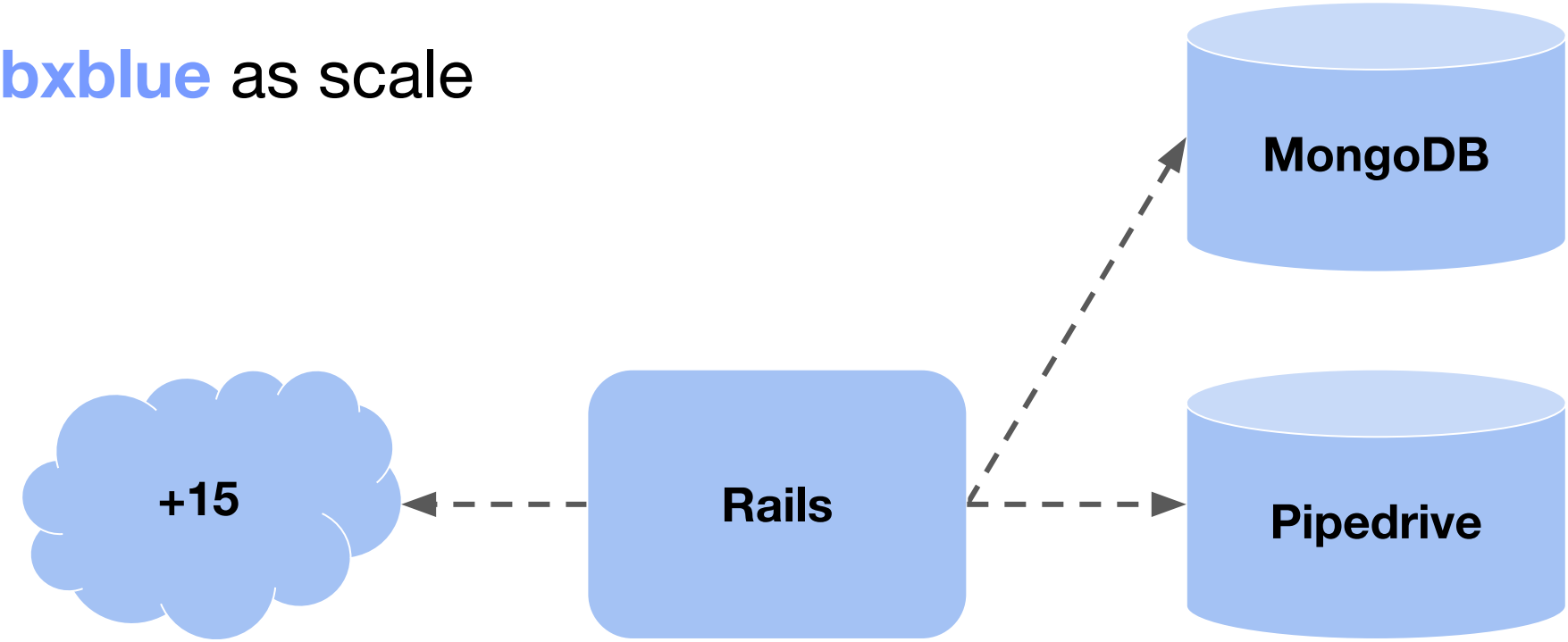
Drawbacks

Hard to Scale:

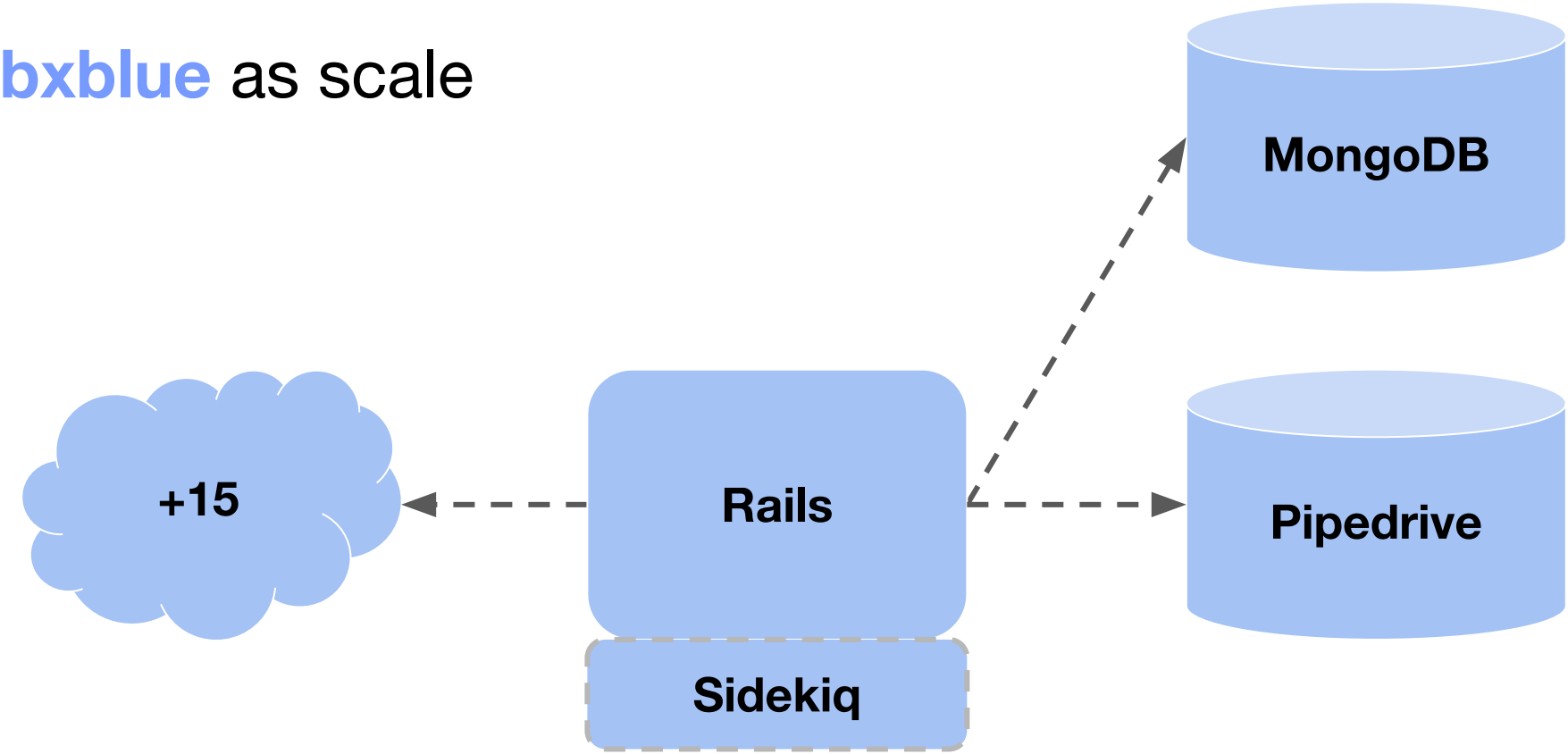
- Tests
- Team
- Deploy
- Stack
- Changes

Chris Richardson
https://microservices.io/patterns/monolithic.html

bxblue as scale



bxblue as scale



Don’t trust yourself

Trust the machine

Automate your tools

- Tests
- Code Quality
- Deploy
- Monitoring

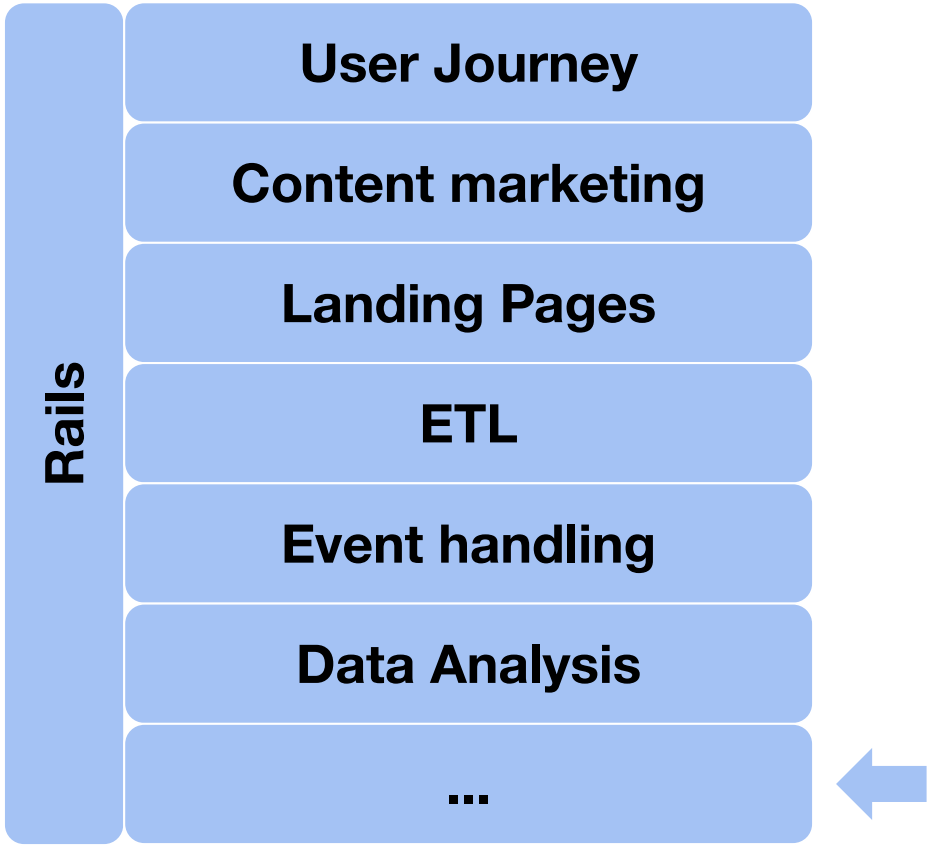
Automate your tasks

Tests	Code Quality	Deploy	Monitoring
Unit Tests	Coverage	CI/CD	Errors
Commons	Linter	Source Control	Servers
Integration	Code Quality	Cloud Pipeline	Logs
Speed	Security		Journey

Automate your tasks - bxblue

Tests	Code Quality	Deploy	Monitoring
rspec	simplecov	LayerCI	Sentry
factorybot	rubocop	Github	New Relic
VCR	reek	Heroku	Logentries
Knapsack	breakman		Metabase

bxblue as scale



Let’s complicate things

Distributed Systems

What does it mean for a system to be distributed?

Unmesh Joshi - 2020
https://martinfowler.com/articles/patterns-of-distributed-systems/

They run on multiple servers.

They manage data.

Microservices

Miniservices as well

Chris Richardson
https://microservices.io/patterns/monolithic.html

Simple,
self-contained,
loosely coupled,
single focused,
services

The Citadel

I love my monolith

DHH - 2020
https://m.signalvnoise.com/the-majestic-monolith-can-become-the-citadel/

Large,
self-contained,
Monolith

Supported by, small
single focused,
problem specific,
services

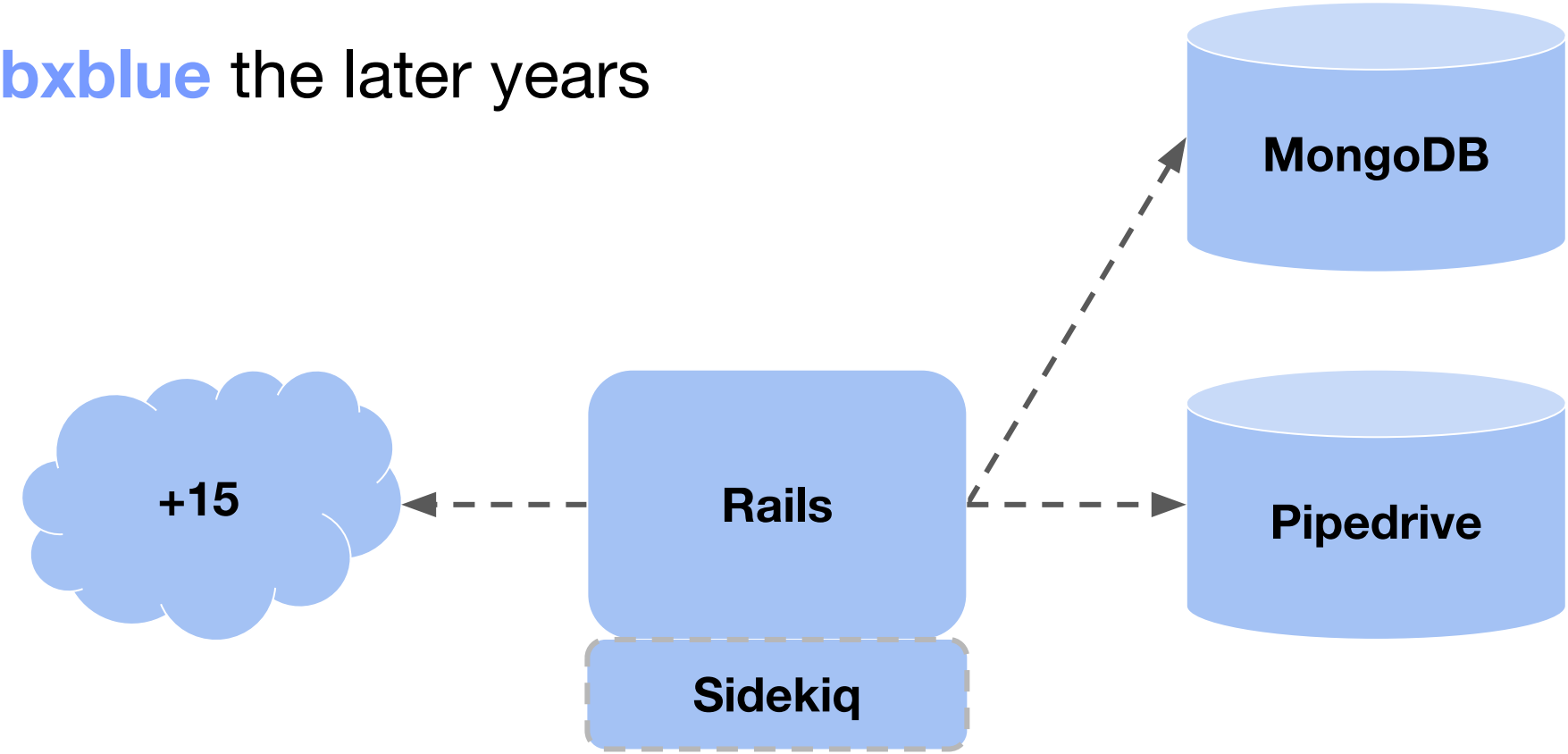
Macroservices

More than Hungry Microservices

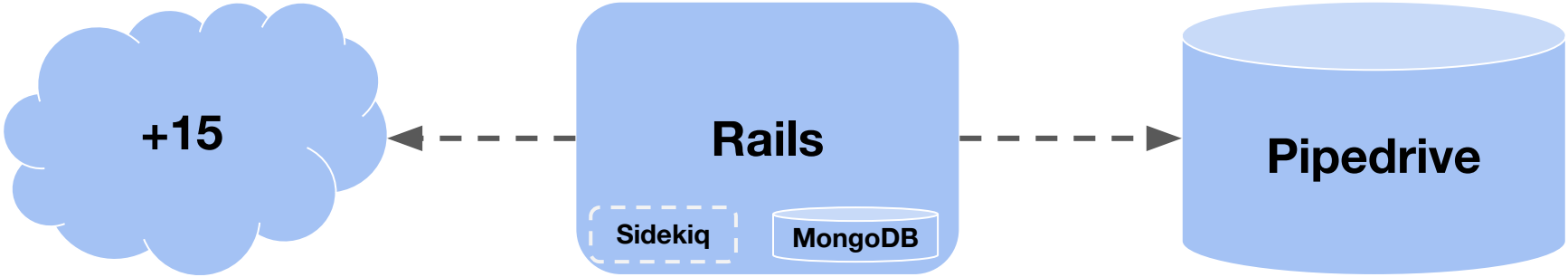
Simple,
self-contained,
context-focused,
multi-purpose
services

Nitesh Gupta - 2020
https://dzone.com/articles/micro-service-mini-service-and-macro-service

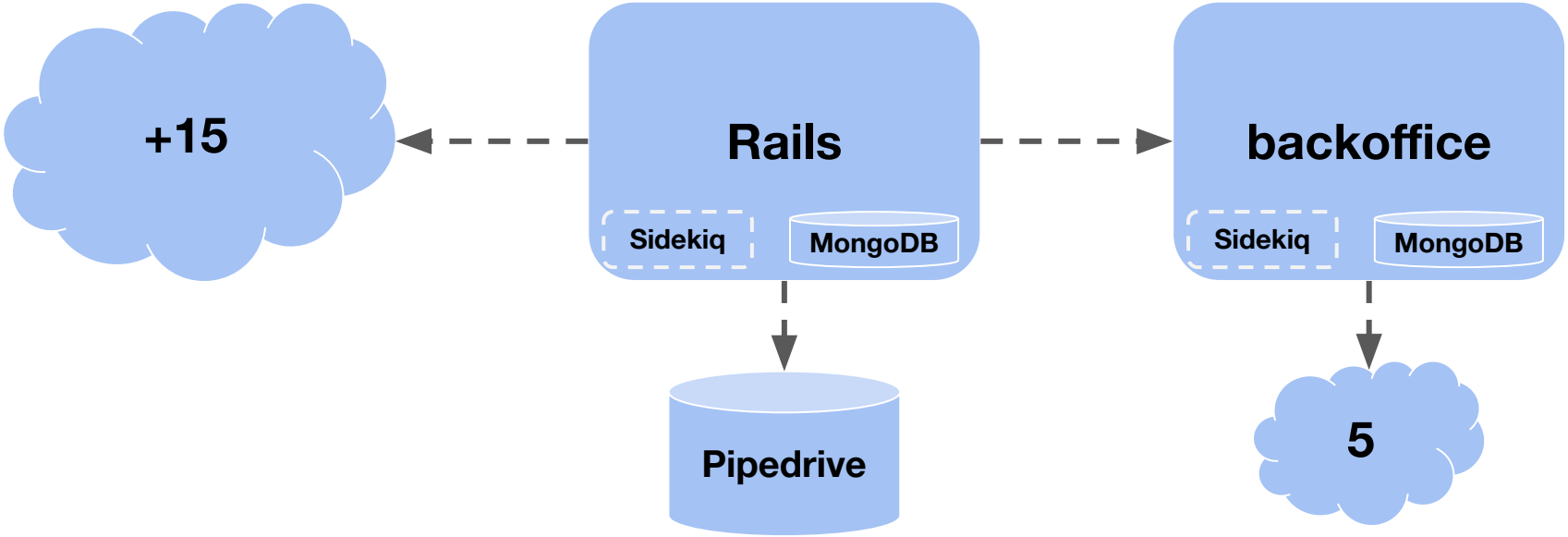
bxblue the later years



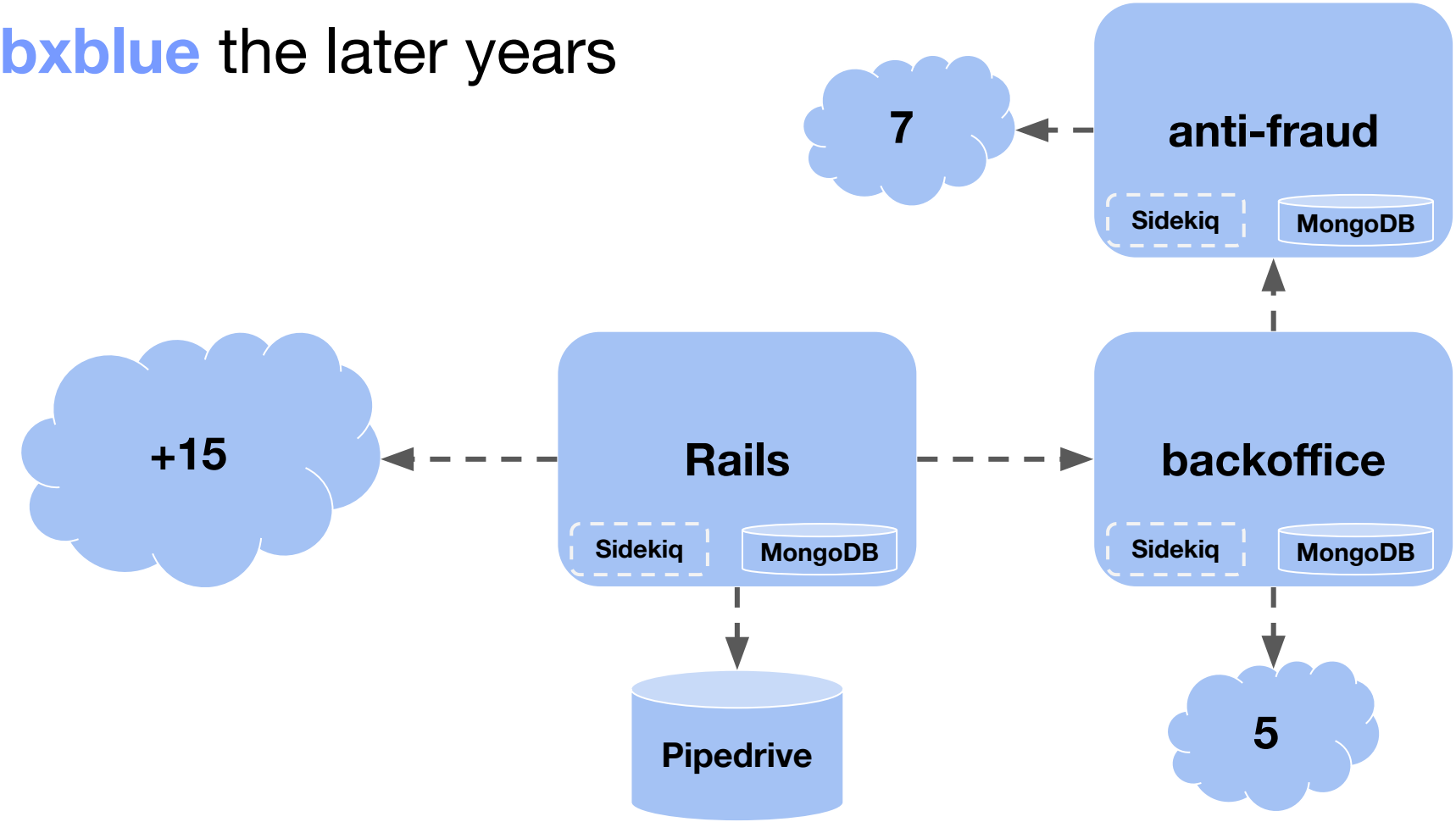
bxblue the later years



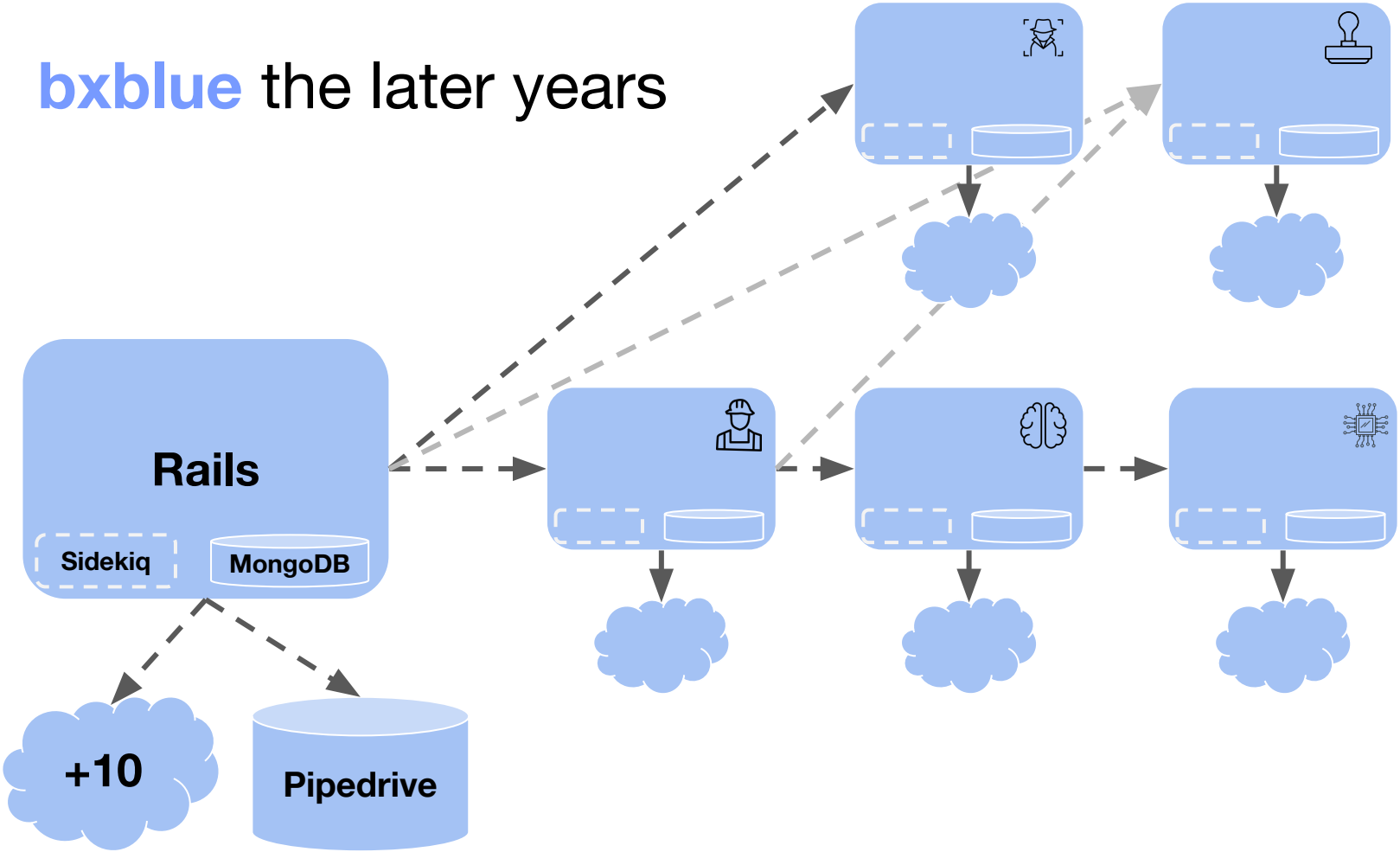
bxblue the later years



bxblue the later years



bxblue the later years



The Stonehenge

Stonehenge

One step further onto
Macro Services

Simple,
self-sufficient,
context-focused,
service-enabled,
applications

Stonehenge

One step further onto Macro Services

Self-sufficient applications

Means they work by themselves.

Stonehenge

One step further onto Macro Services

Context-focused applications

Means they handle a single context very well

Stonehenge

One step further onto Macro Services

Service-enabled applications

Means they are available to integrate and scale with others

Law of conservation of complexity

Complexity has to go somewhere

Larry Tesler ~ 1980
http://www.designingforinteraction.com/tesler.html

Every application has an **inherent amount of complexity** that cannot be removed or hidden

Dead Code

Don’t let the zombies bite you

30% of files

25% of classes

5%~10% of methods

Simone Romano and Giuseppe Scanniello - 2018
http://www.cs.wm.edu/~denys/pubs/TSE'18-DeadCode.pdf

Dead Products

* for seed round or crowdfunded companies death rate is 97%

70% of Companies **will fail** until 20 months after last fundraising*

CBInsights - 2020
https://www.cbinsights.com/research/startup-failure-post-mortem/

Summing up

Gall's Law

A complex system that works is invariably found to have evolved from a simple system that worked.

John Gall - 1975
"Systemantics: How Systems Really Work and How They Fail."

A simple system **may or may not** work.

A **complex system** designed from scratch **never works** and cannot be patched up to make it work.

Don’t trust yourself

Trust the machine

Automate your tools

- Tests
- Code Quality
- Deploy
- Monitoring

Why Stonegenge?

"The simplest solution is usually the best"

friar William of Ockham 1287–1347
Occam's razor

It's all **Distributed Systems**

Decision making is **hard**

Things will **change**

Thanks

Building the Stonehenge
Using Gall’s law

by Fabricio Buzeto

about.buzeto.com

@nukdf

bxblue.com.br


Fabricio Buzeto
Bernardo Lima
Clarissa Lima Borges
Bruno Pedroso
Marcus Vinícius Da Silva...
Arthur Nobre
danny
Mateus Luiz Freitas Barros
João Vitor Tadra
Brendo Soares
Nídia Couto
Matheus Albuquerque
Juliana
Tayanna Cruz
Pedro Mello
Mariáh Novaresi
Elisa Schuster
Rebecca Dutra
Luan Pallin
Gustavo Gorenstein
Mariana Macedo
Gabriel Teles
Thalysse Viana
Venha trabalhar na
bxblue
bxblue.com.br/vagas
We're hiring