

The Architecture of an API Platform

Johannes Lundberg
@johannesl

”Self-taught developer. **46elks** co-founder.
Strongly believe it's possible to combine
powerful features with ease of use. Enjoys
vegetarian food and world travel.”

@johannesl

The essence of an API

Turning traditional complexity
into programmatic simplicity.

Two kinds of APIs

DATA ACCESS

- Open data
- Twitter & Facebook
- Public transportation
- Partner APIs

SIDE EFFECTS

- Cloud storage
- Online payment
- Infrastructure as a service
- Cloud communications

Self contained

- Open data
- Twitter & FB
- TV channels

**DATA
ACCESS**

**SIDE
EFFECTS**

- Cloud storage
- Paypal
- Printing

- Transportation
- Social media
- Open data

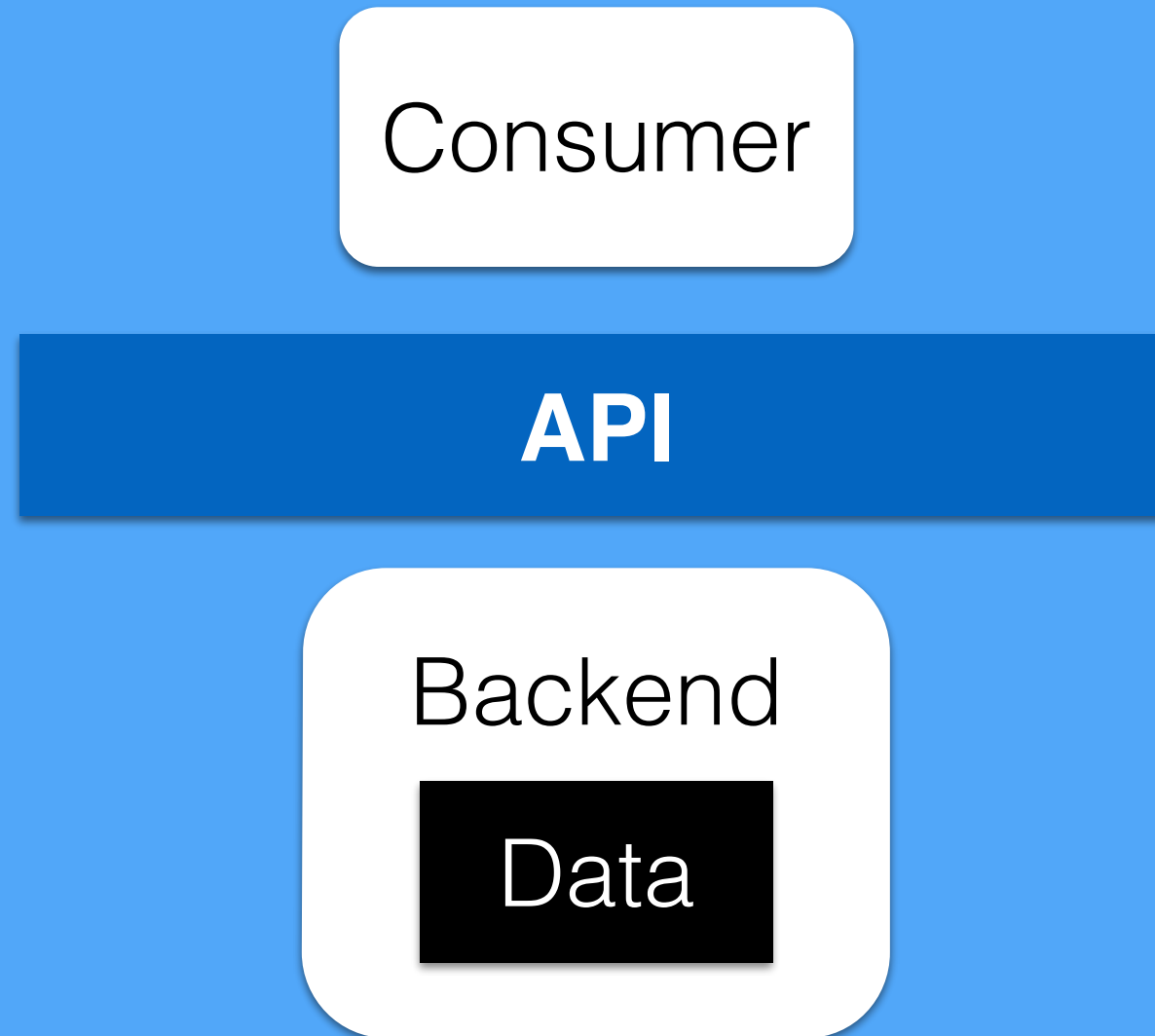
**DATA
ACCESS**

**SIDE
EFFECTS**

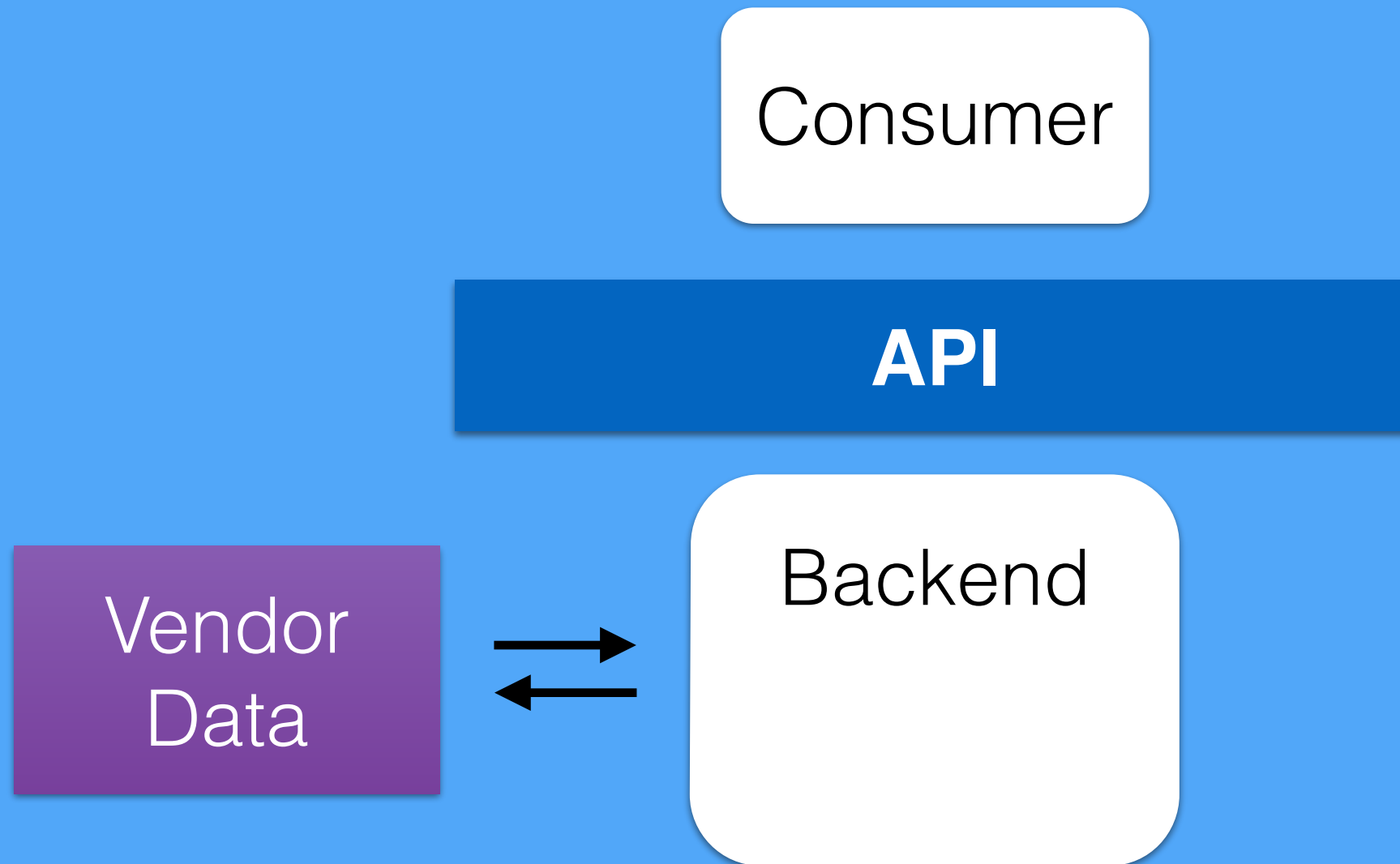
- Payments
- Travel tickets
- Telephony & SMS

Vendor dependent

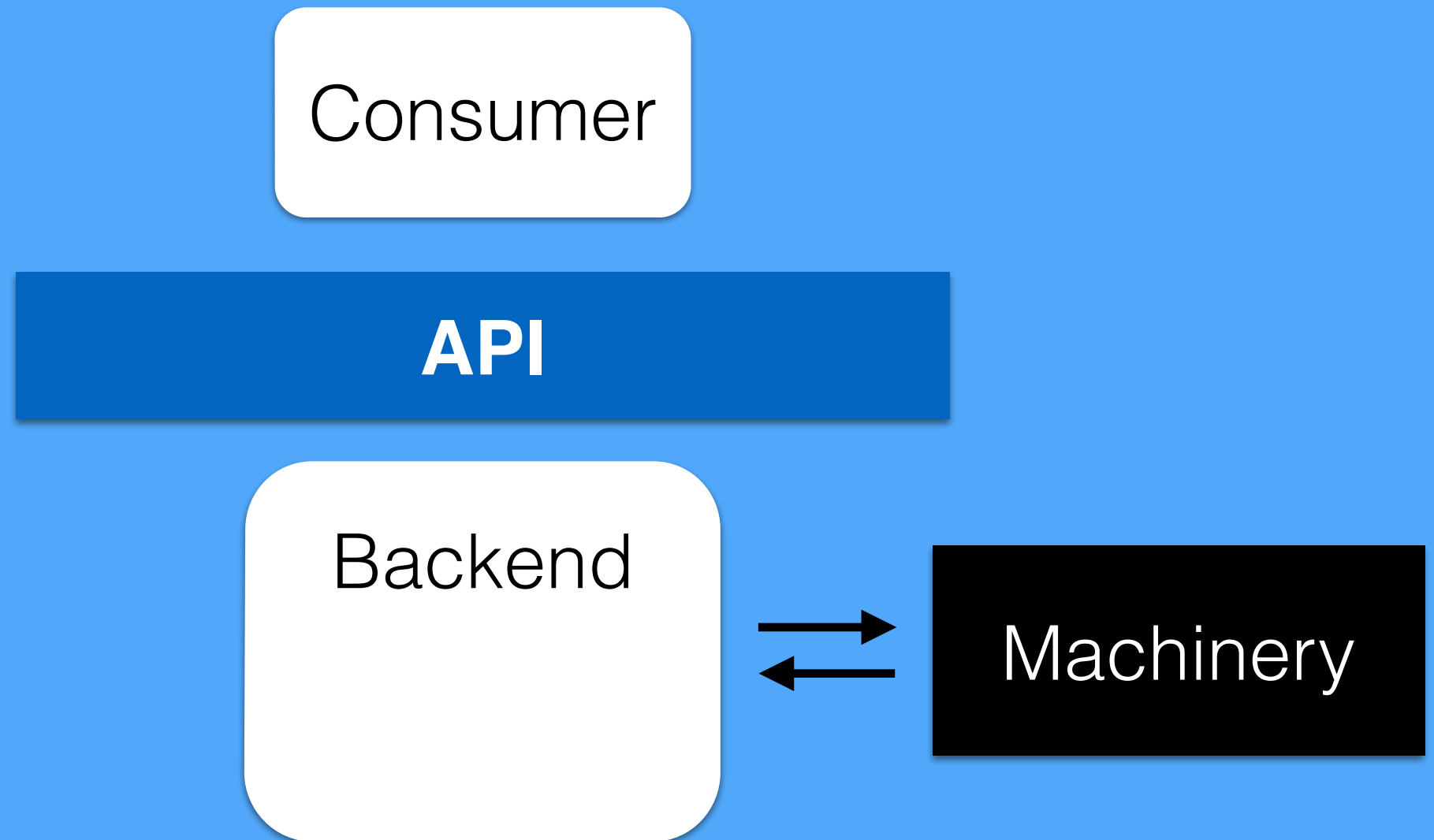
DATA ACCESS - Self contained



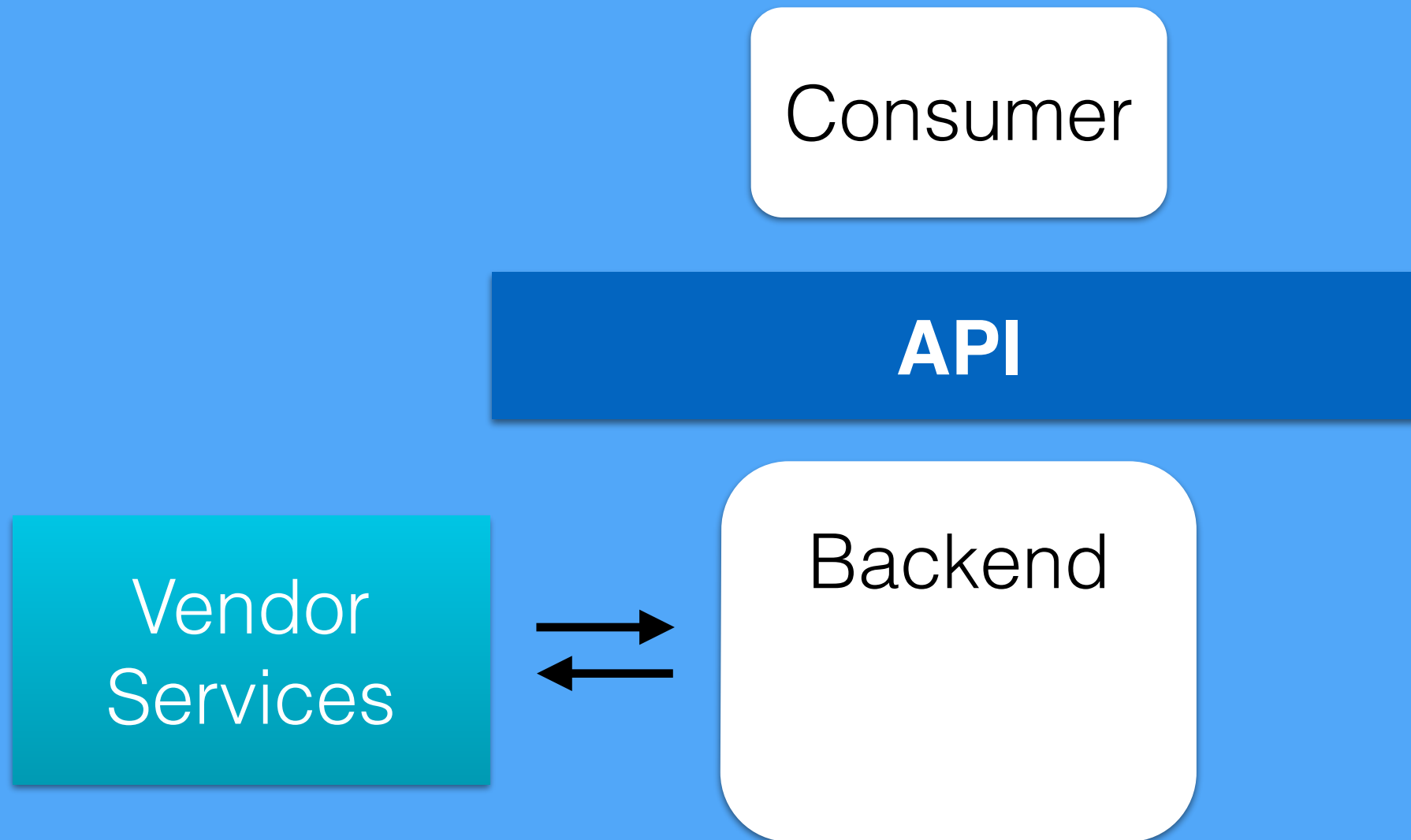
DATA ACCESS - Self contained



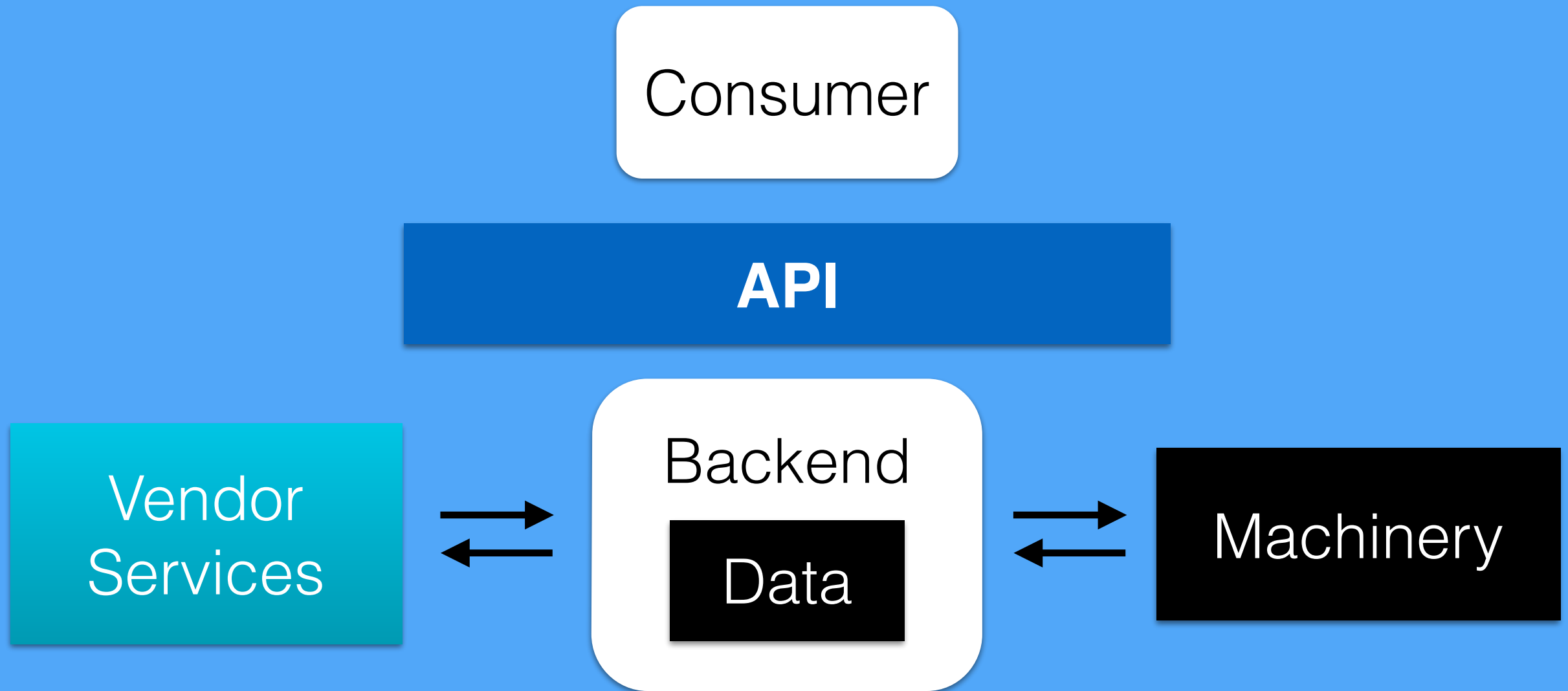
SIDE EFFECTS - Self contained



SIDE EFFECTS - Vendor dependent



REALITY?



REALITY??

Consumer

API

Vendor
Data

Vendor
Services

Vendor
Services

Backend

Data

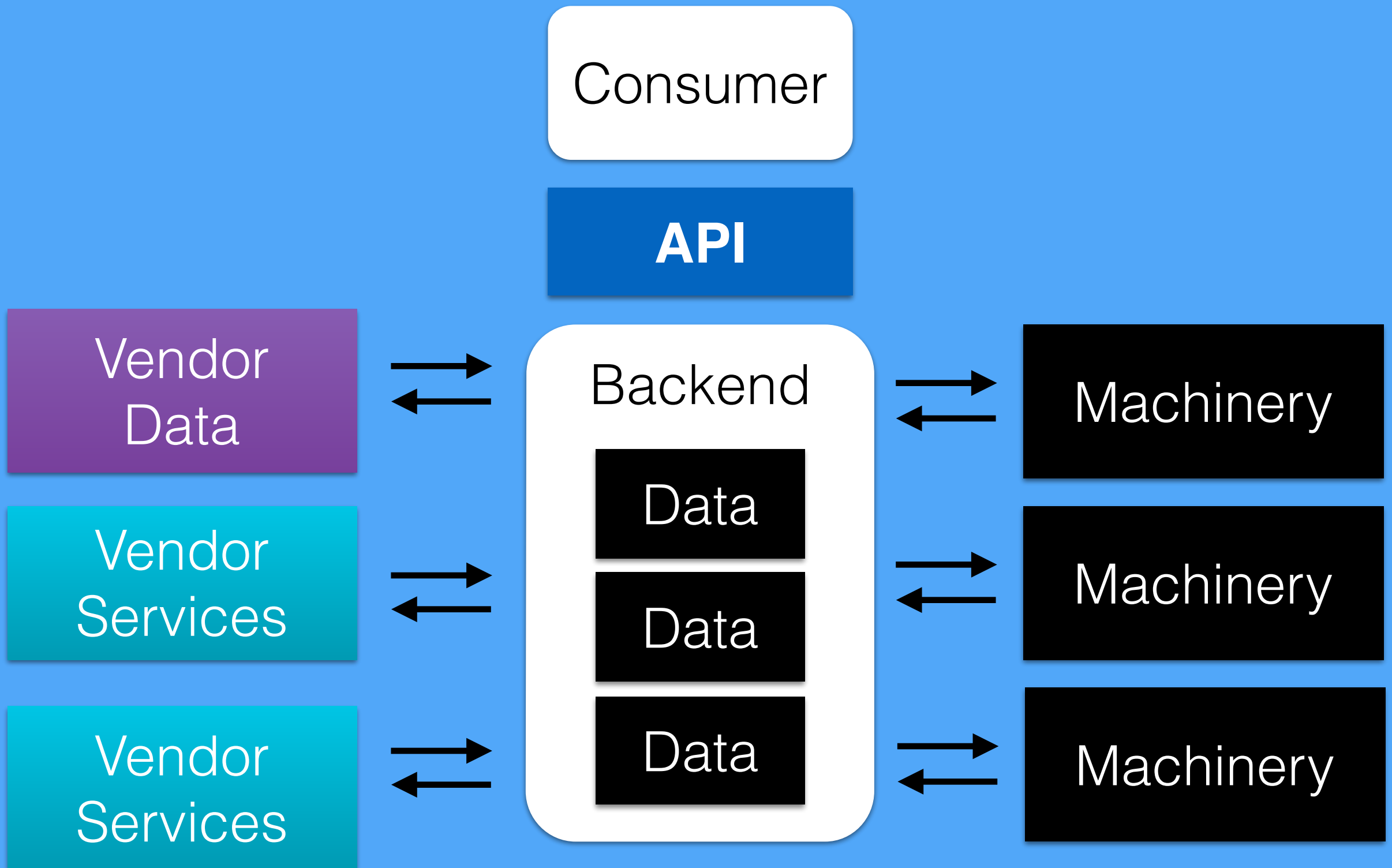
Data

Data

Machinery

Machinery

Machinery



What's the main
features of your API?

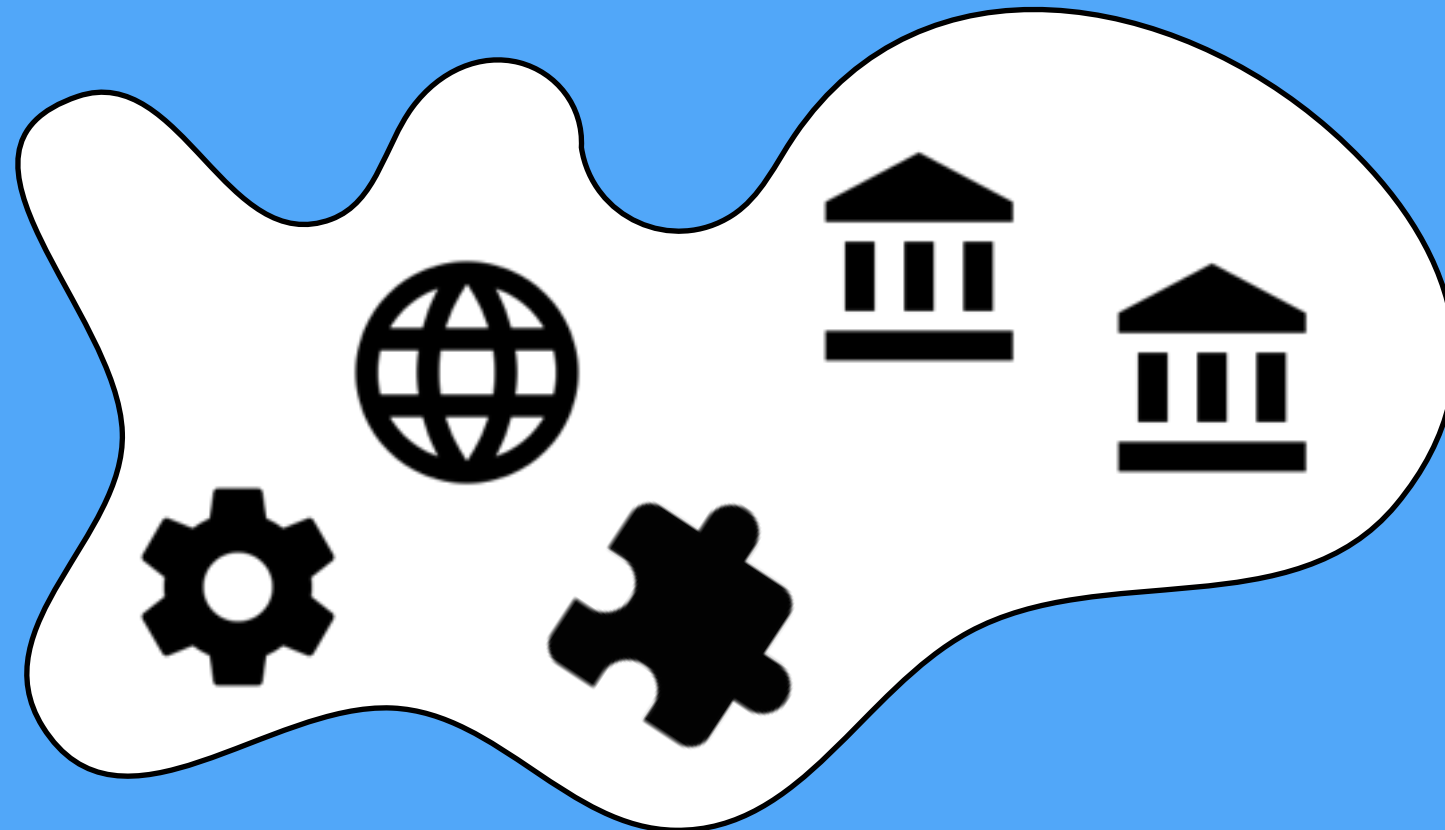
We all know
development costs..

Prototype first!

Prototype first! Yes, **with customers.**

Consumer

API



1. Your API should work.
2. According to specs.
3. Maintainability..
4. Scalability..
5. Feature creep..

The golden rule of API design

**When in doubt
Leave it out!**

`XML` vs `JSON`

`Servers` vs `IaaS`

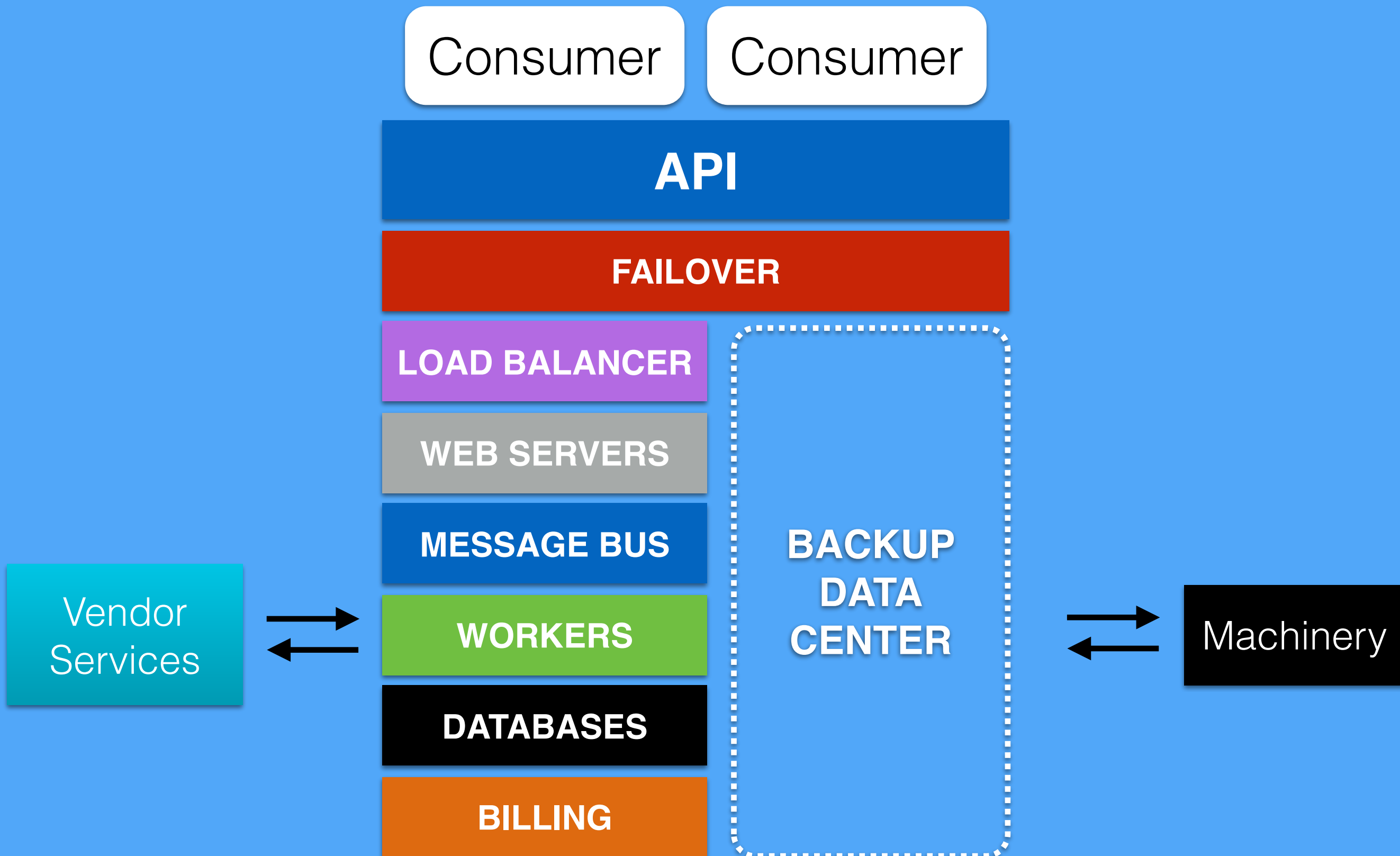
`PUSH` vs `POLL`

`OAuth` vs `Basic`

`Consistency` vs `Reliability`

`RabbitMQ` vs `ZeroMQ`

Something like this..



Do versioning.

Use **JSON**.

Talk to your API consumers.

Select vendors **carefully**.

**Building APIs
should be complex**



**Voice, SMS and MMS
in one easy API**

Customers, Employees,
Event audience, Machines

Customer
applications



SMS, MMS
Phone calls

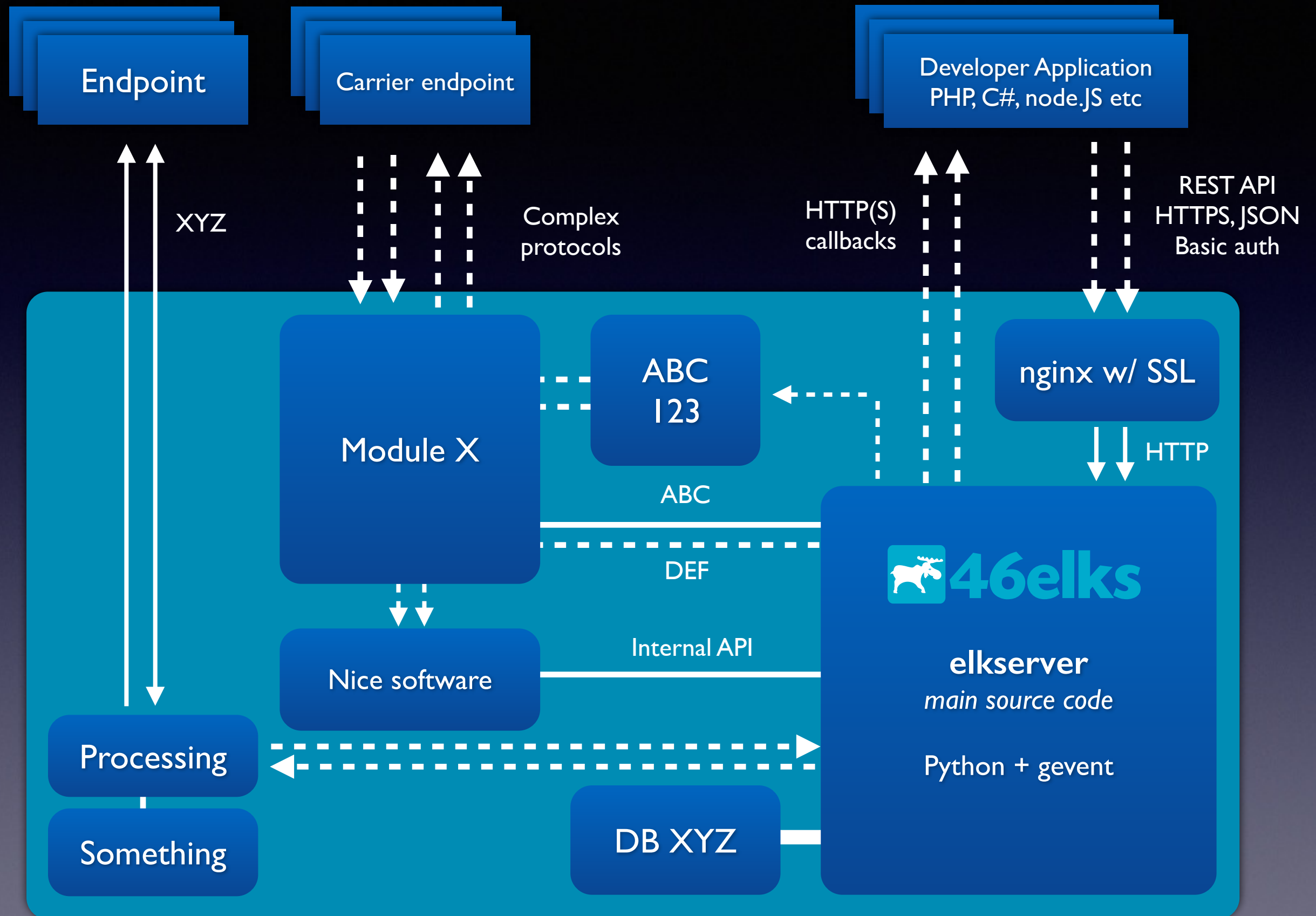


Simple API

platform engine

TELCO NETWORKS

THE INTERNET



Conceptual module interaction

The Architecture of an API Platform

Johannes Lundberg
@johannesl