

Engine Yard Cloud Defining PaaS

Ezra Zygmuntowicz
twitter: @ezmobius



Engine Yard Architecture



Repeatable/Disposable Infrastructure

- Gather application requirements
- Define your cluster size/shape
- Instantiate your deployment on the cloud
- Monitor and maintain your running systems
- Deploy new code with confidence

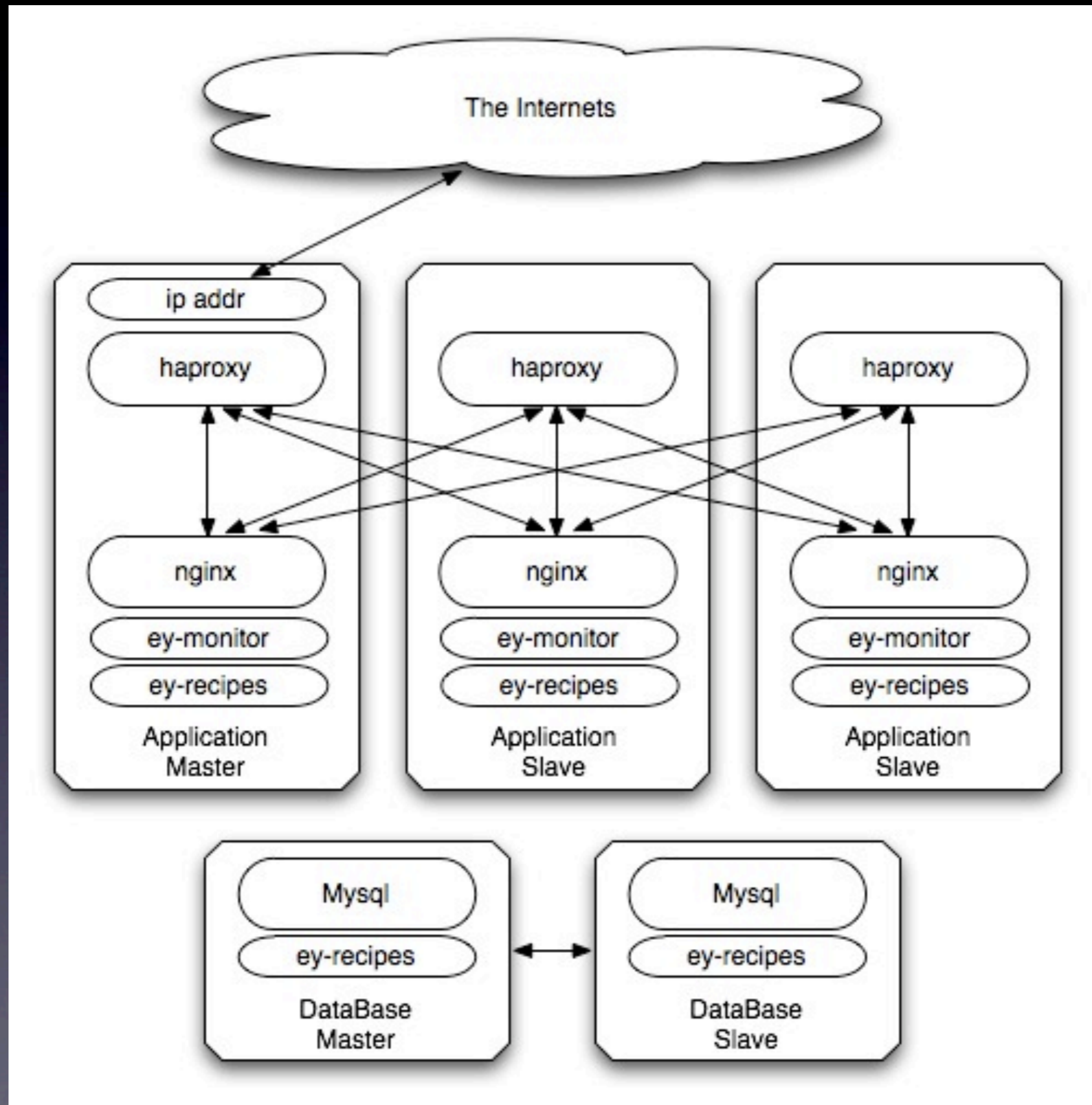
Command & Control from a Distance



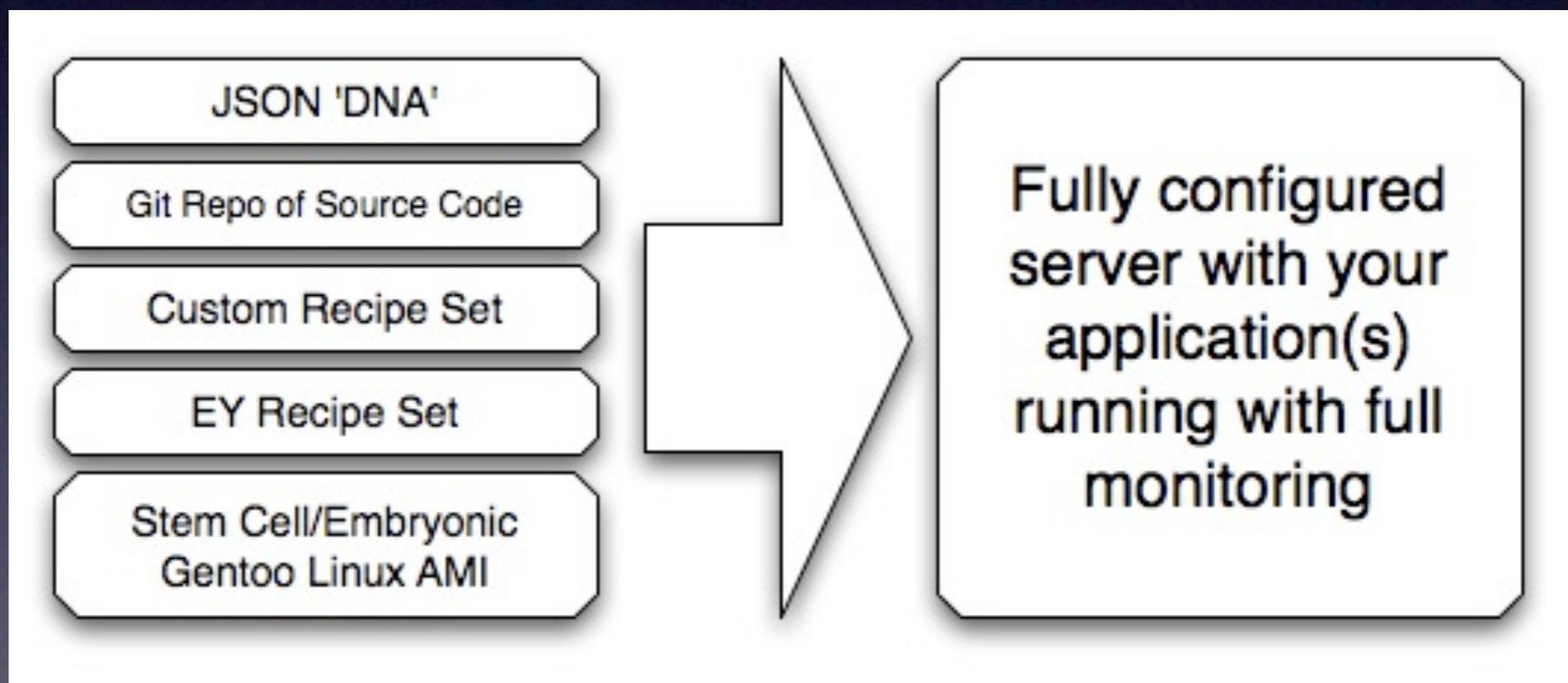
Your clusters keep running even if our platform app fails

Define your Application, Dependencies and Cluster Shape

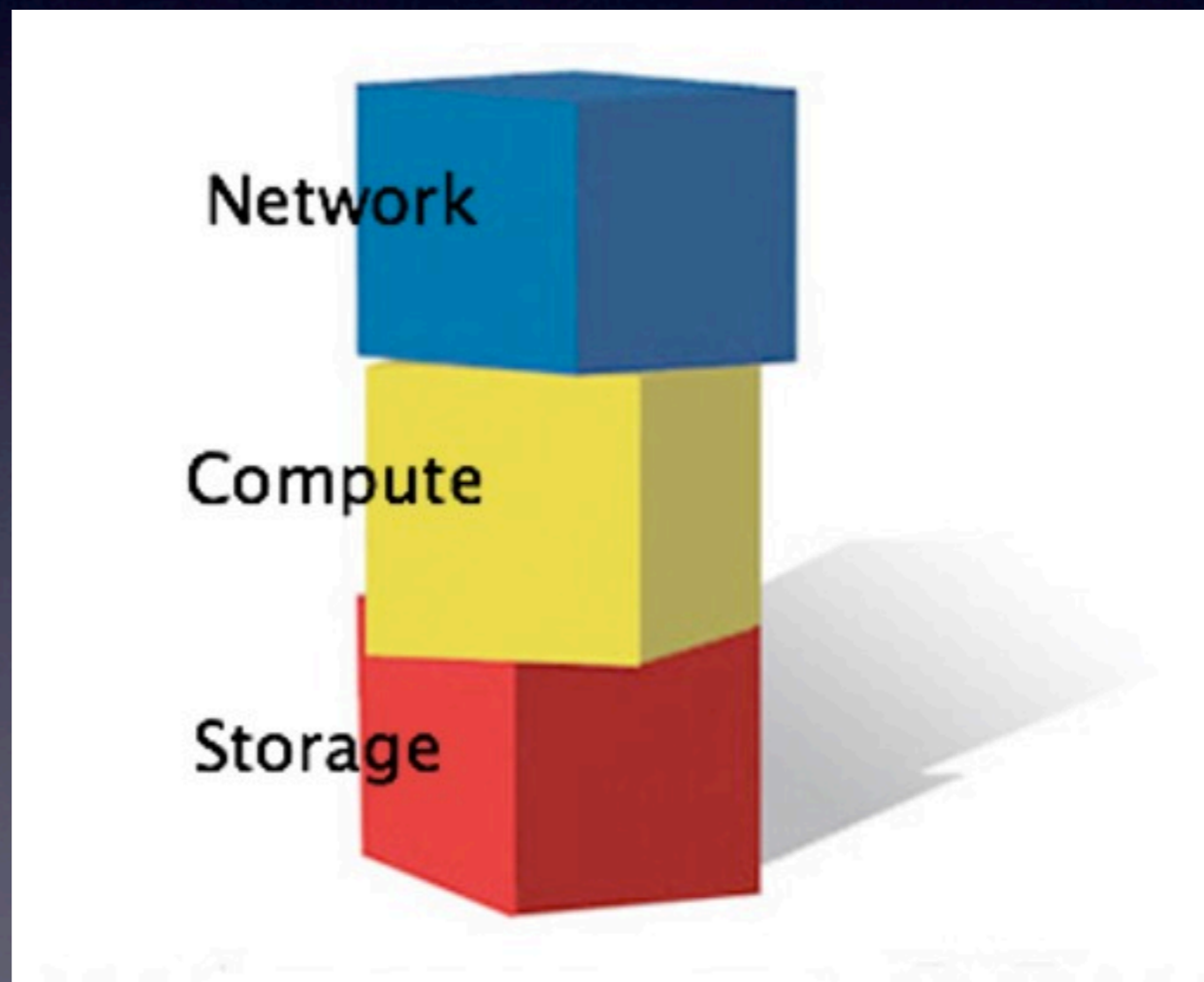
Self-Healing Clusters



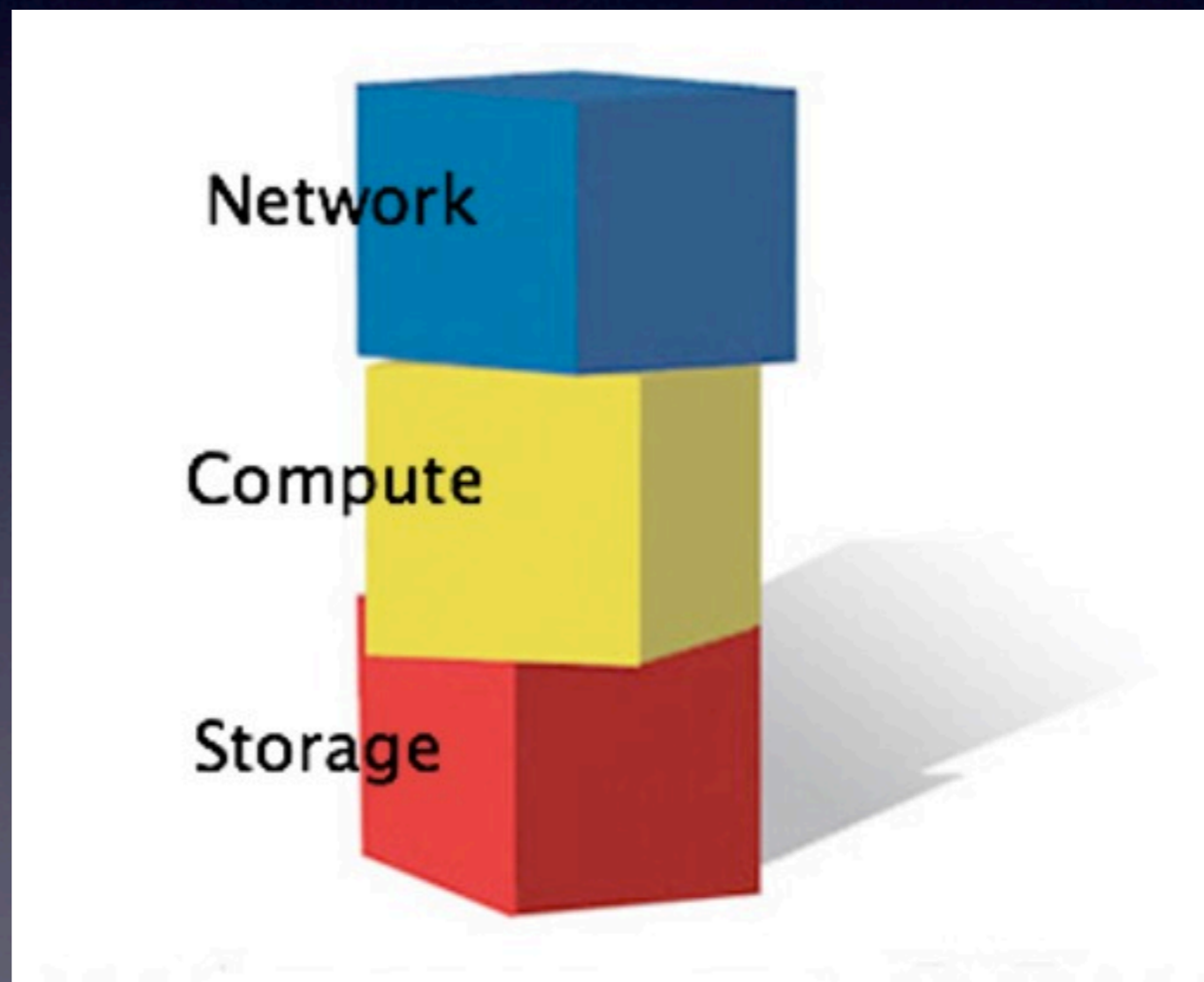
Cooking Servers



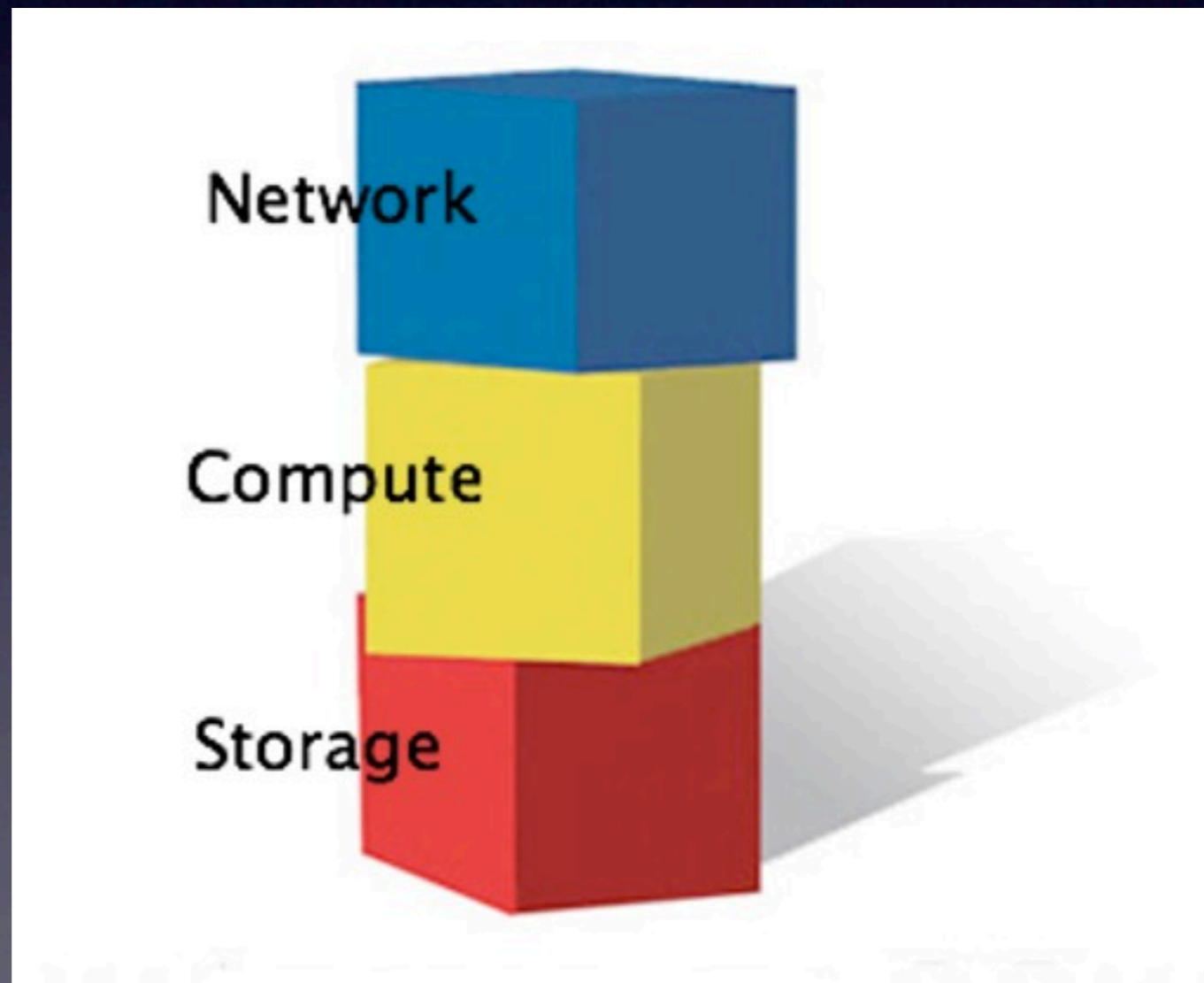
IaaS gives you basic building blocks



IaaS gives you API's for said blocks

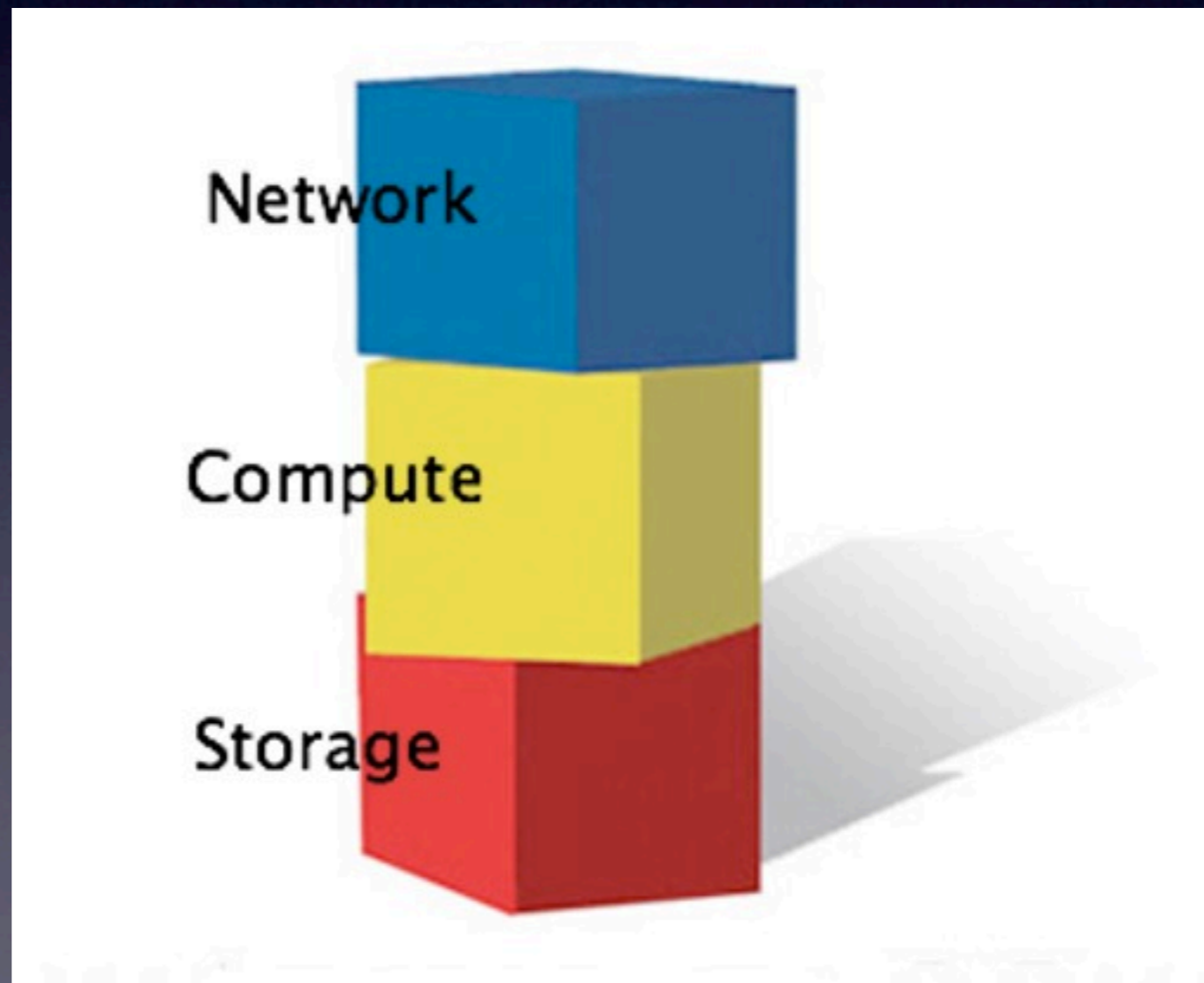


IaaS is for geeks like me



IaaS

is too complex or irrelevant
for most app developers



PaaS:
assembles basic blocks
into higher level
abstractions for
Applications



PaaS:

Applications are what
we all want to run
So focus needs to shift
to Applications needs

PaaS Implementation Decisions

PaaS Implementation Decisions

Multitenancy with IaaS
account

VS

Managing IaaS account

PaaS Implementation Decisions

What kind of scale is
your average customer
going to be?

PaaS Implementation Decisions

Build your own security
abstractions[hard] or
rely on battle tested
security(linux,
hypervisors)[easier]?

PaaS Implementation Decisions

How much of the
underlying systems and
API's are exposed to
end users?

PaaS Implementation Decisions

Customers bring their
own IaaS account or
not?

PaaS Lessons Learned

Distribute work/
responsibility to the
edges as much as
possible

PaaS Lessons Learned

Don't have central
point of failure that can
“bring down the
house”

PaaS Lessons Learned

Less sharing is better
for the customer
more sharing is better
for the PaaS provider

Balance is King here

Questions?