



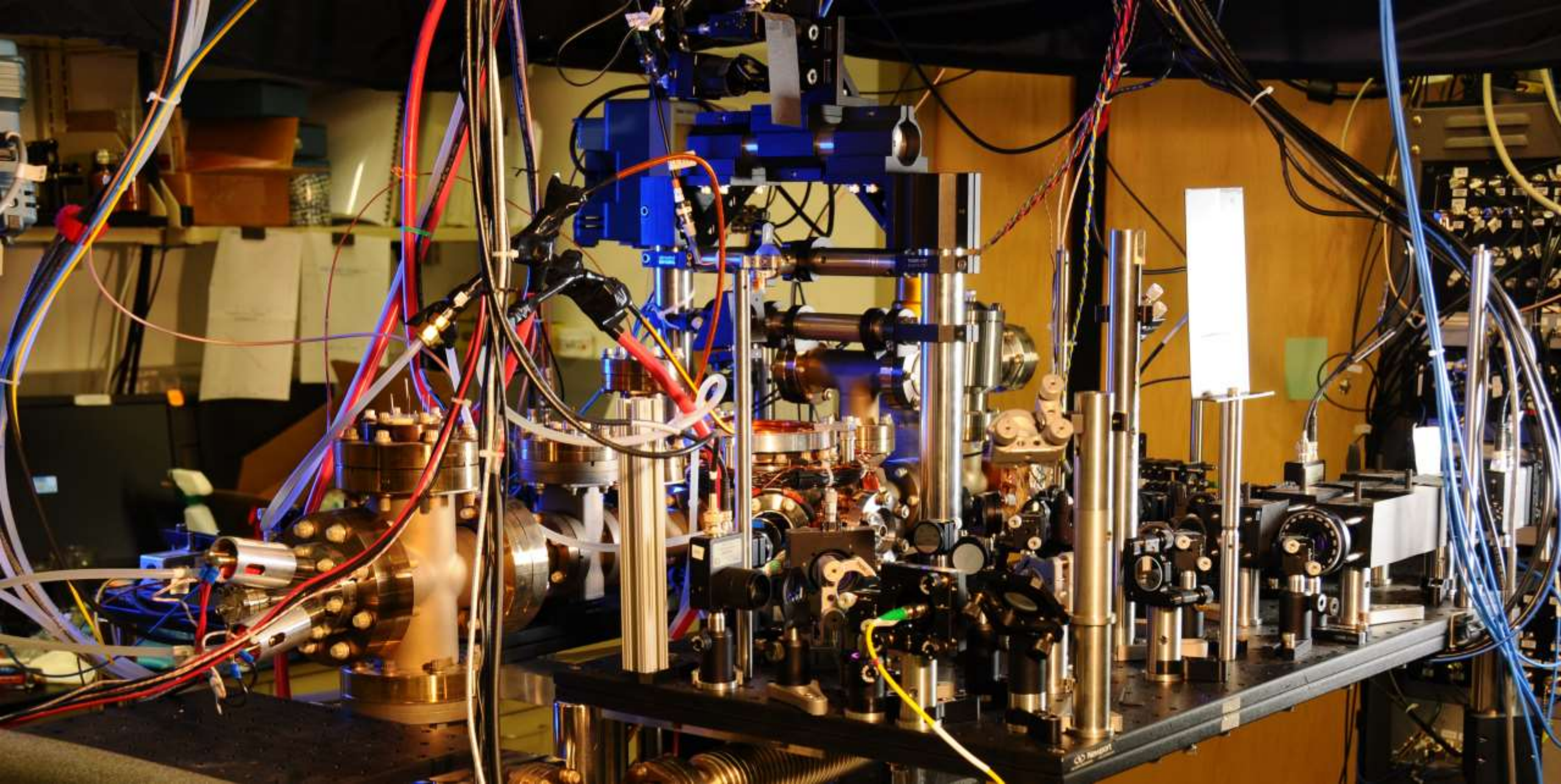
Your business  
doesn't have  
technology problems











# You still have problems















*Solving  
technology  
problems isn't  
your business*

You don't have  
technology problems

How much time does IT  
spend on solving  
business problems?





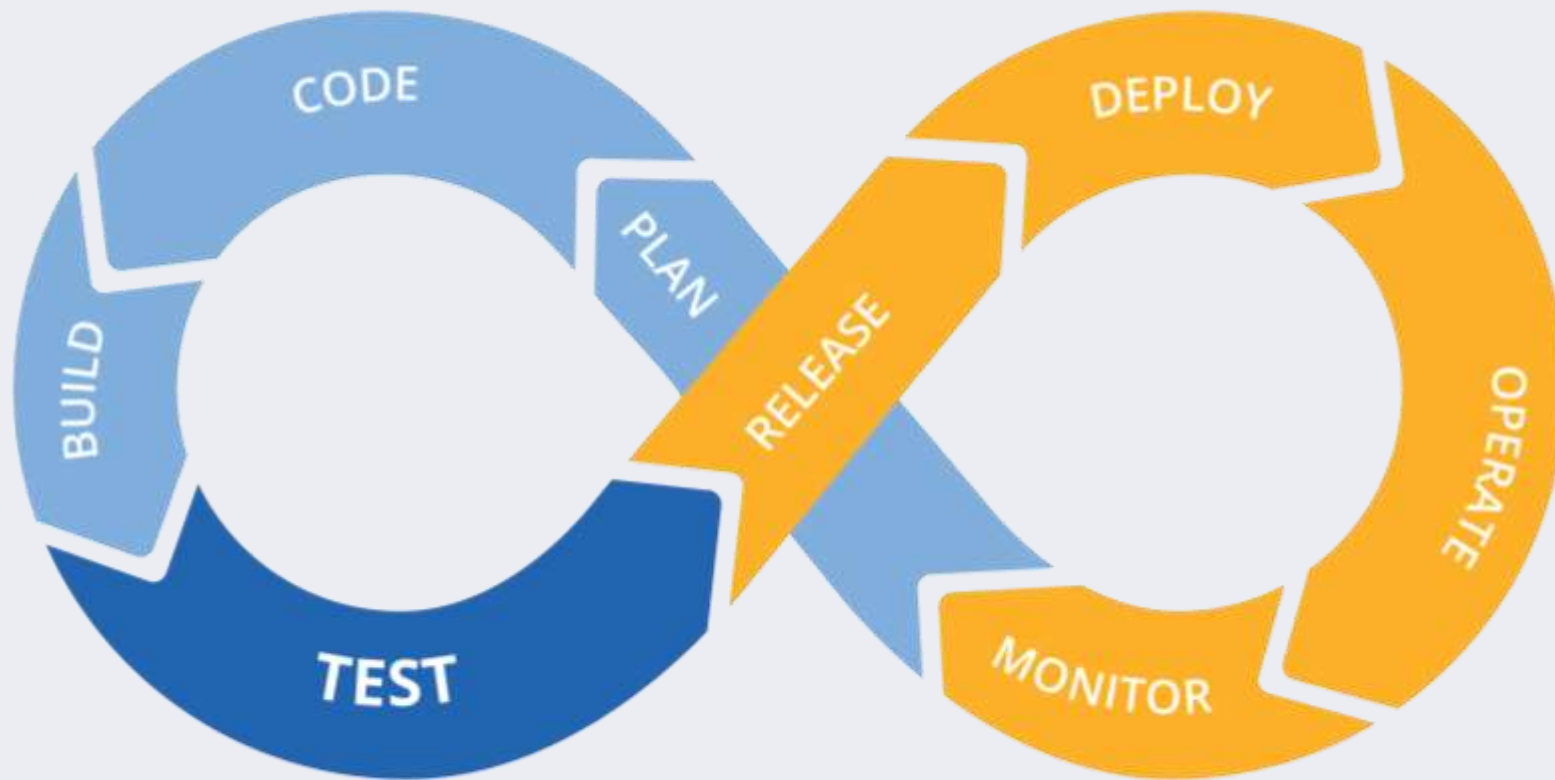
0 - 30%

30 - 60%

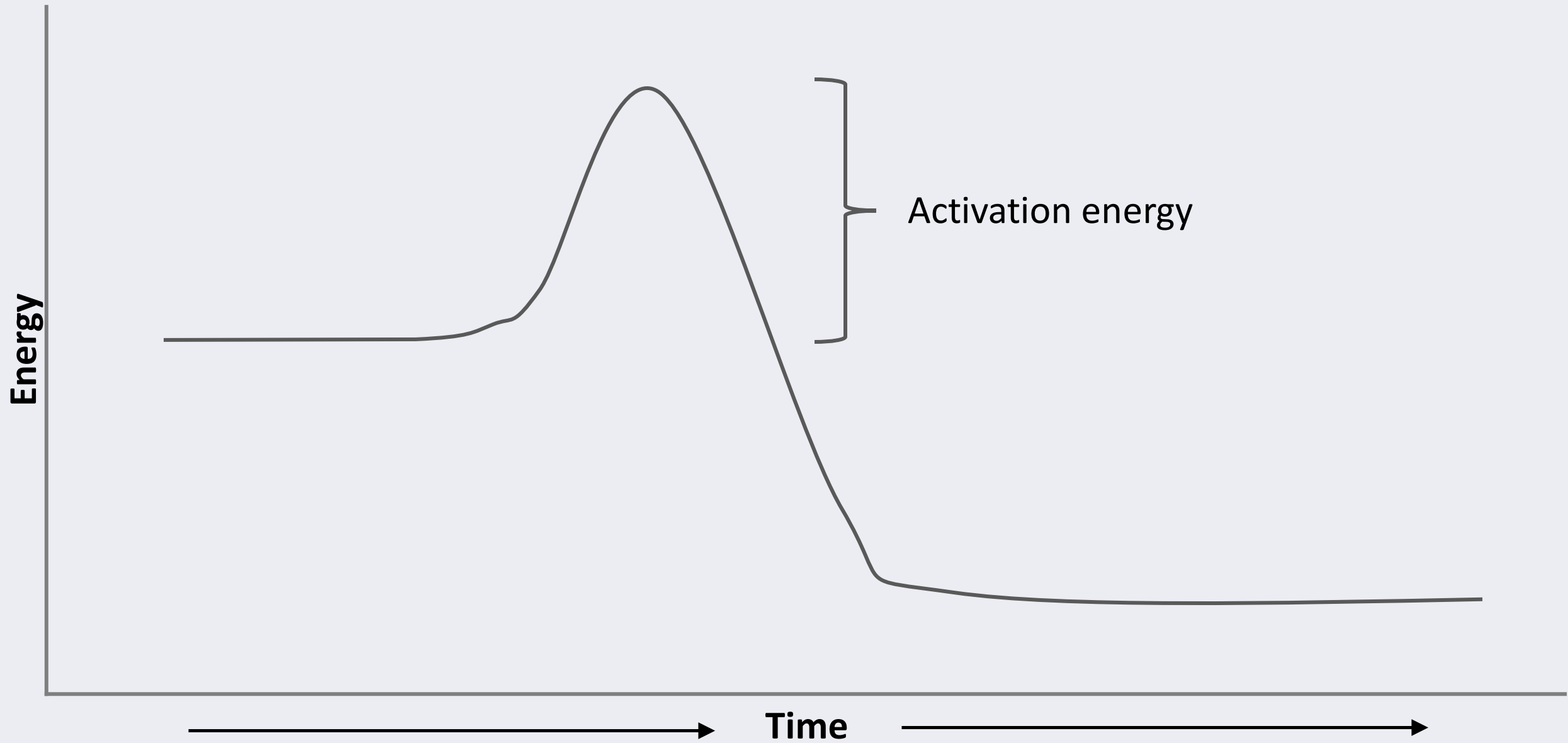
60 - 90%

We have nothing new  
to offer here

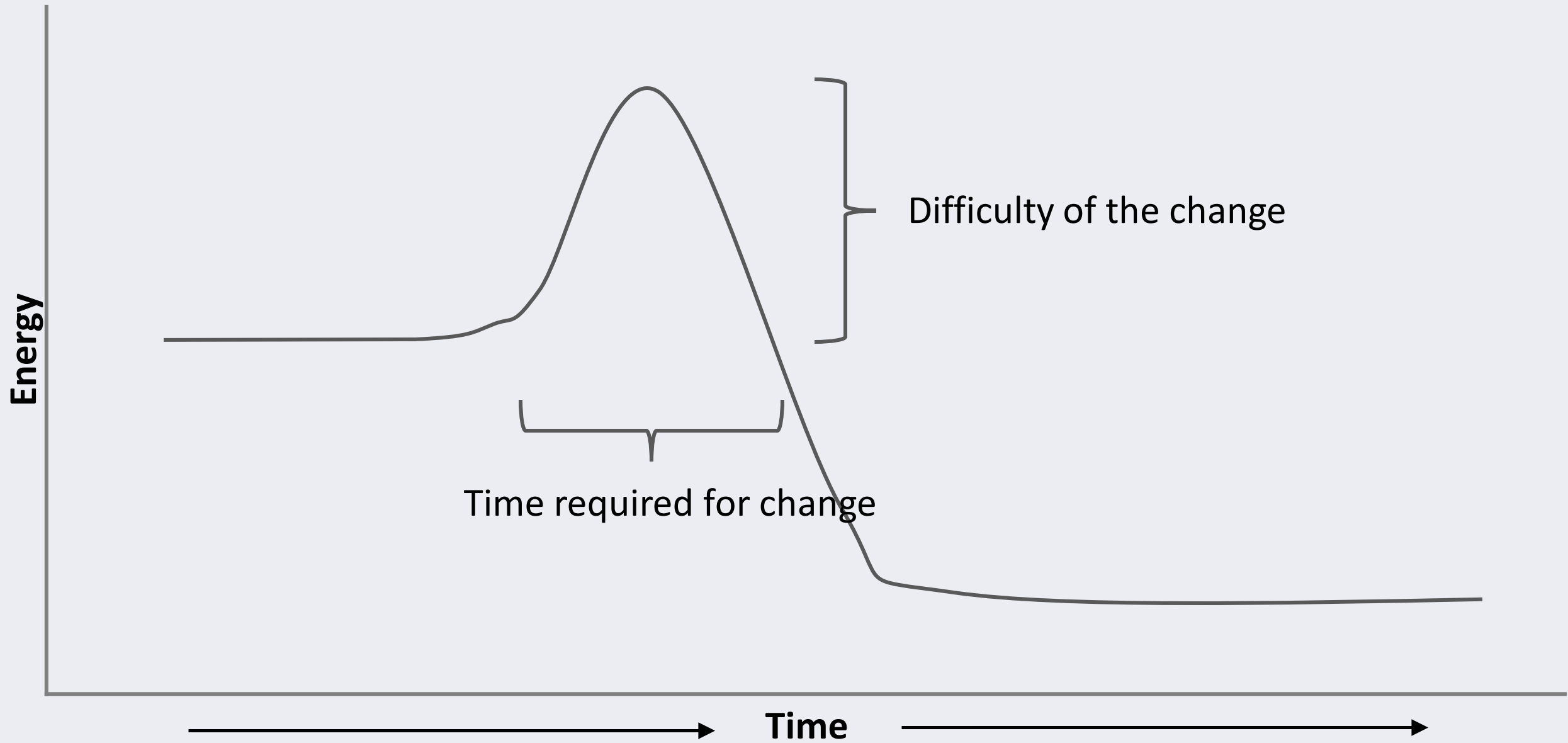


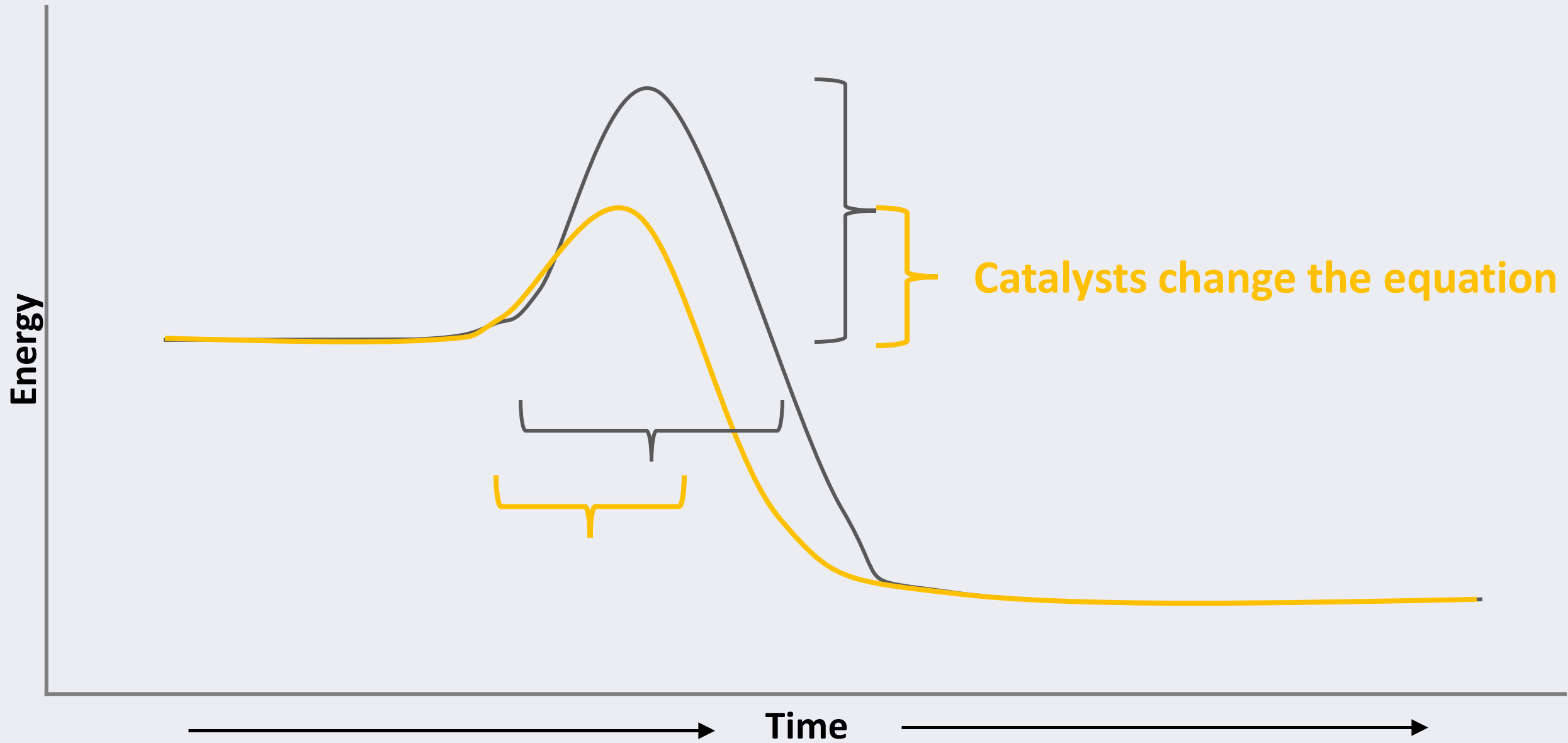


If it's understood, why  
are so few doing it?



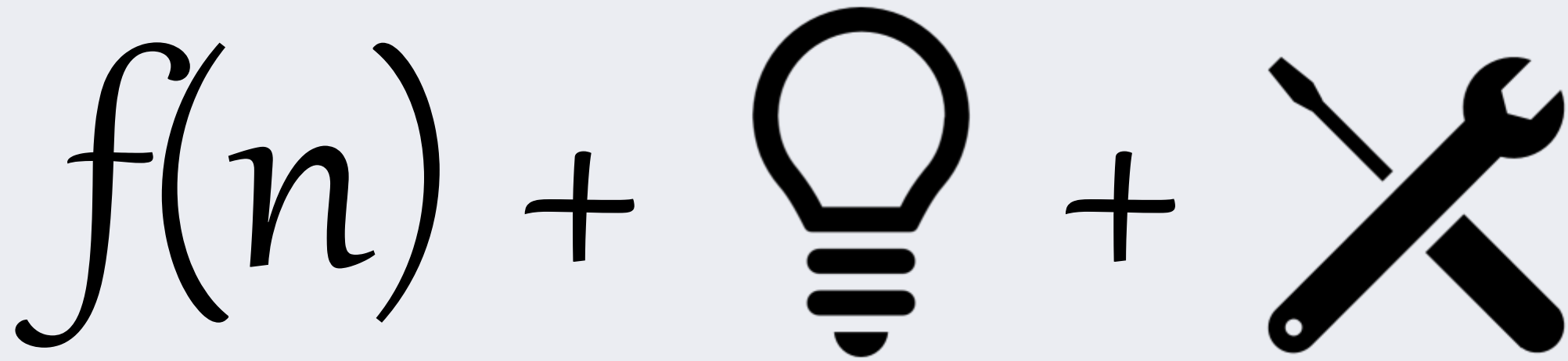


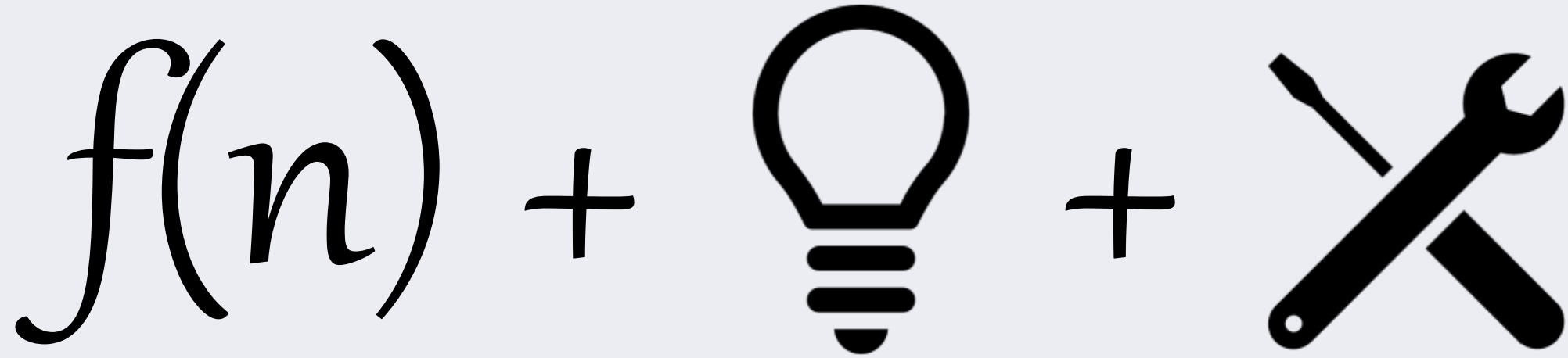




# What is serverless?









# Serverless as catalyst

# Unique value proposition to DevOps laggards





# Data driven decision making

# Small, product focused teams

# Cloud adoption & automation



# Discovery through experimentation



# How to try it

# Challenges



Where it's already started



***StanleyBlack&Decker***

NORDSTROM



The Seattle Times

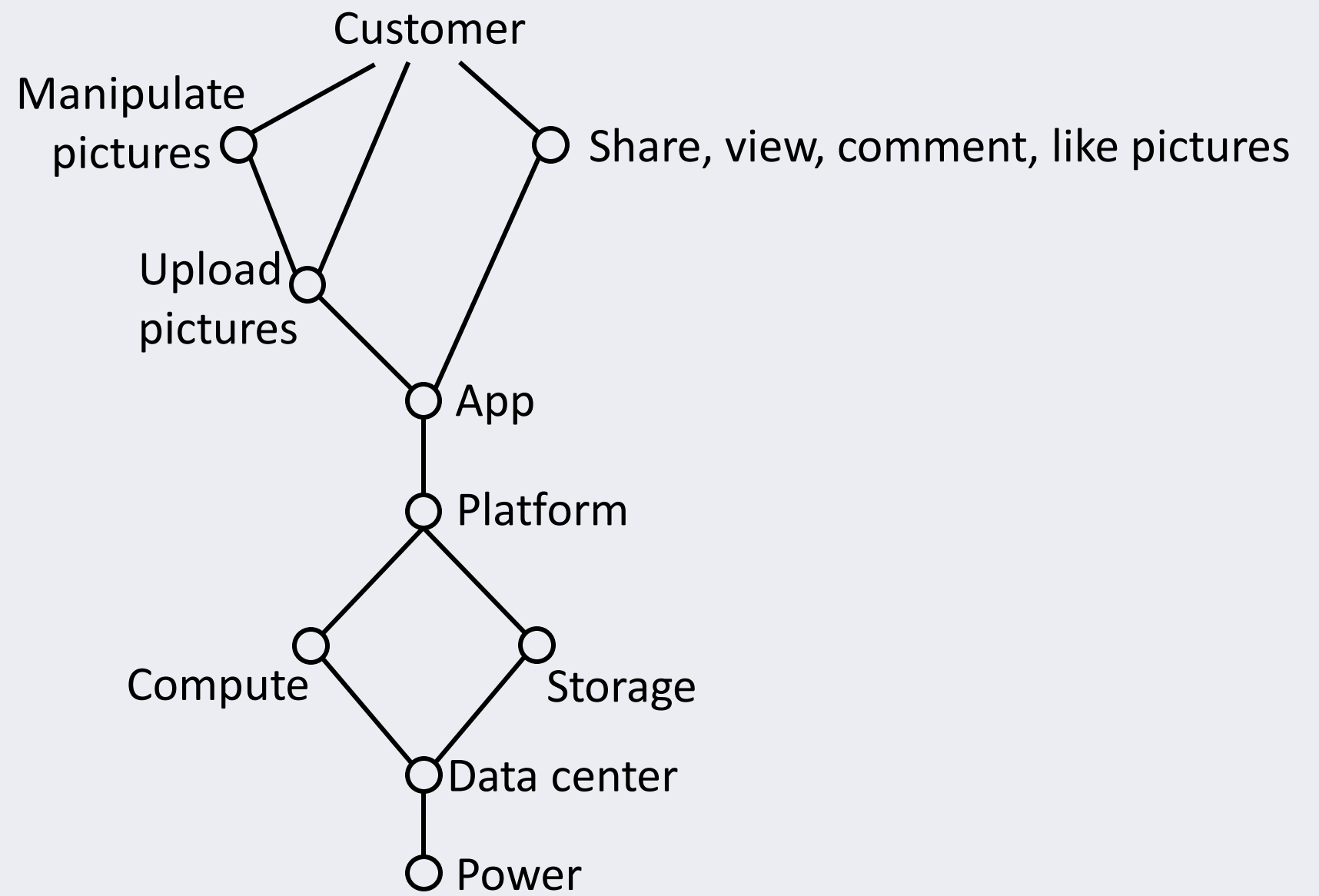
# GameStop®

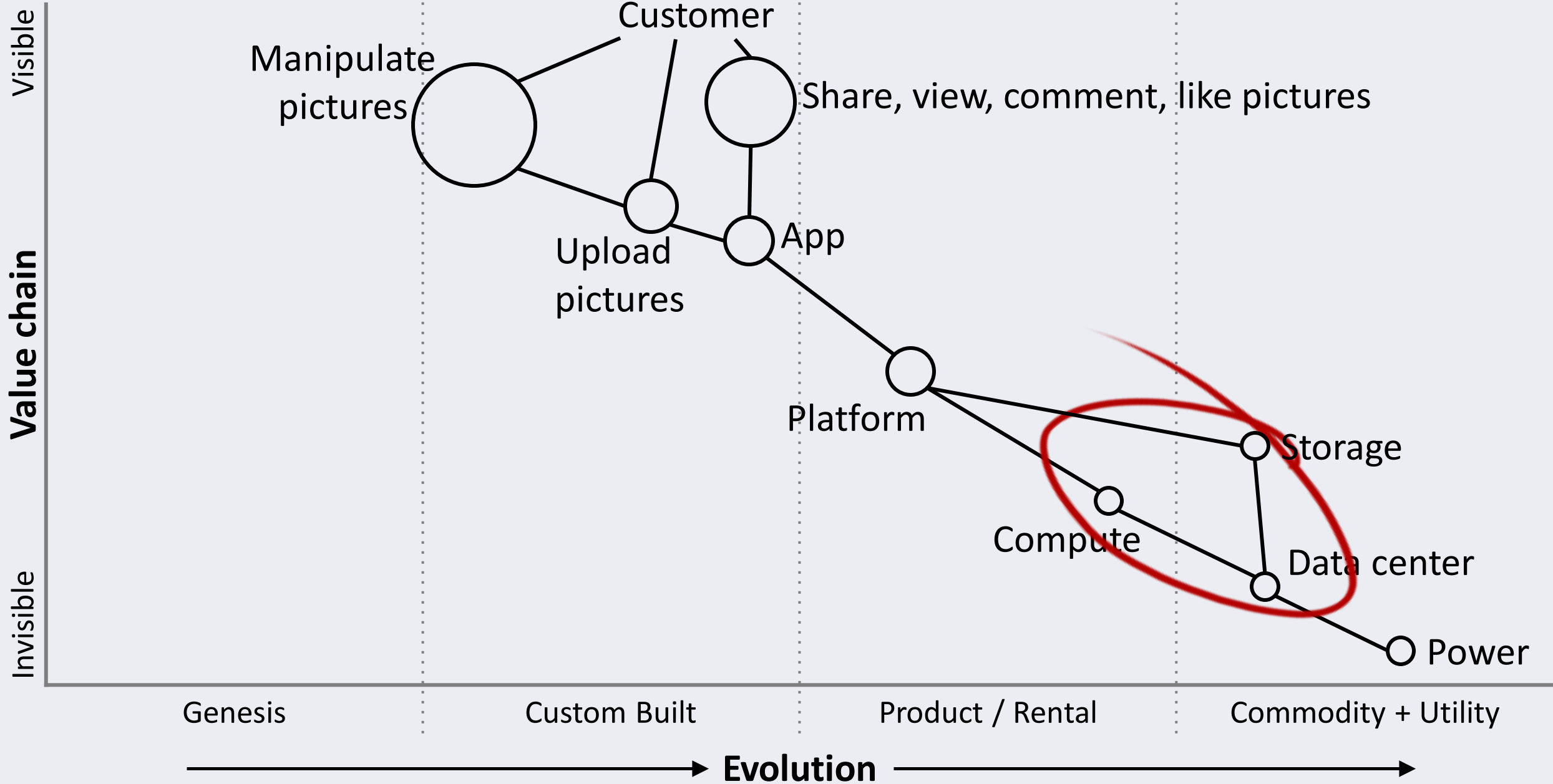
POWER TO THE PLAYERS®

Visible

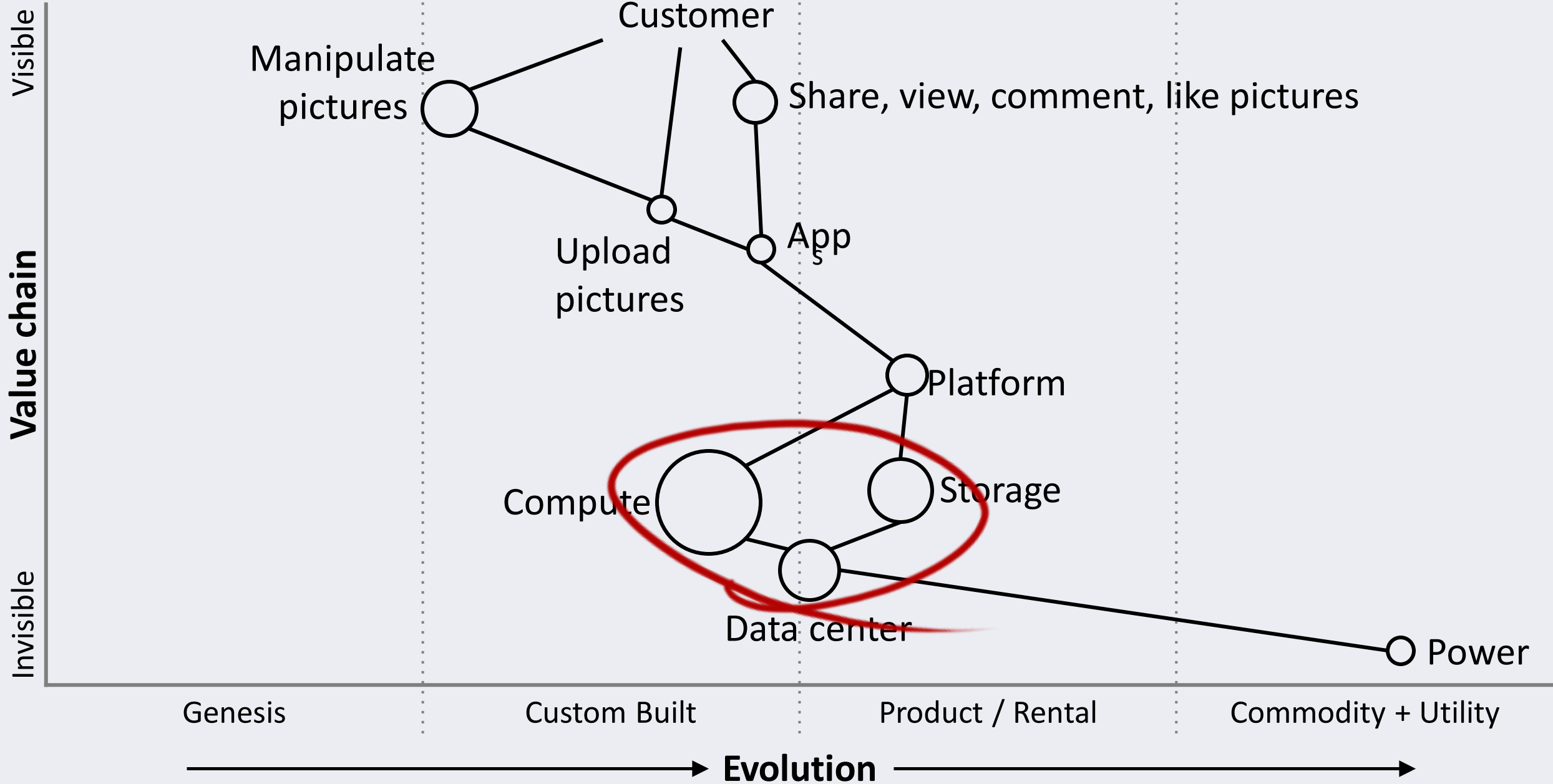
**Value chain**

Invisible









# What does it mean for my people?



Solutions Architect  
(Bit blueprint creator)



Data Architect  
(Stores bits, manages access)



Developer  
(Describes how to move bits)



Team Lead  
(Keeps bit machine well-oiled)



Product owner  
(Talks to customer, attempts to talk to bit-team)



Release Engineer  
(Moves bits)



Ops Engineer  
(Watches bits)

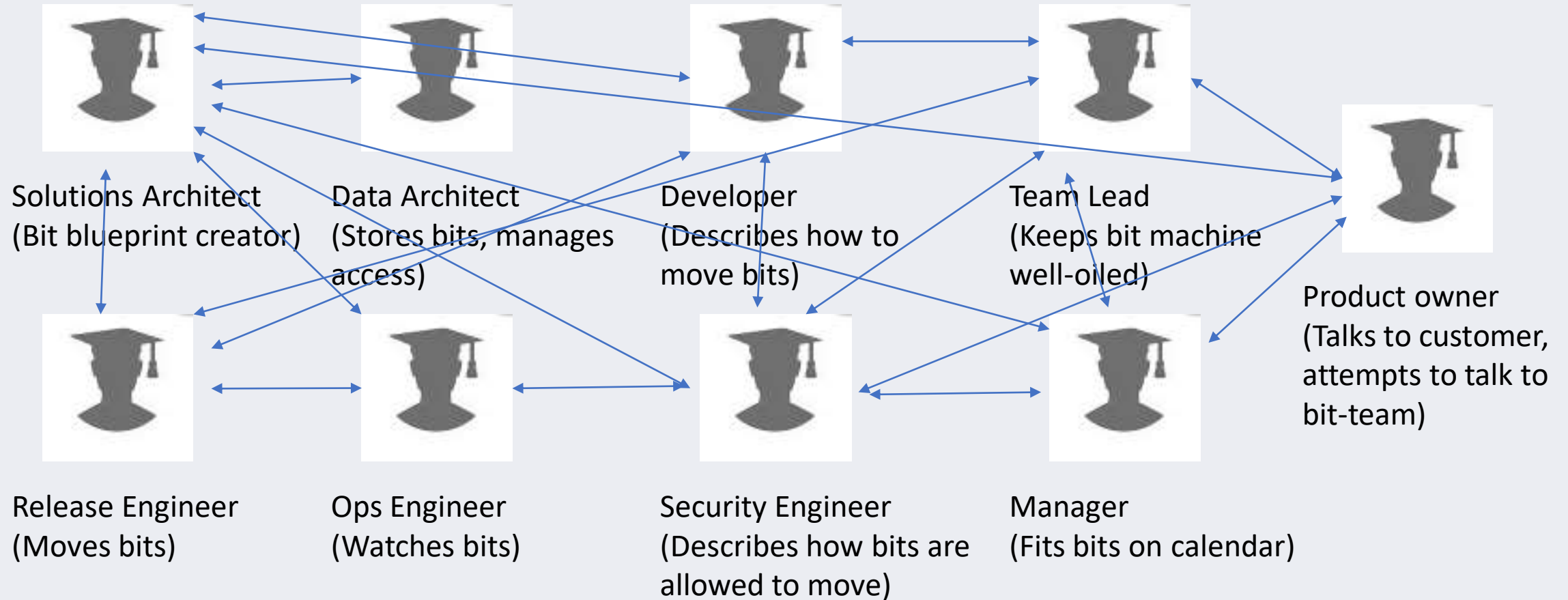


Security Engineer  
(Describes how bits are allowed to move)

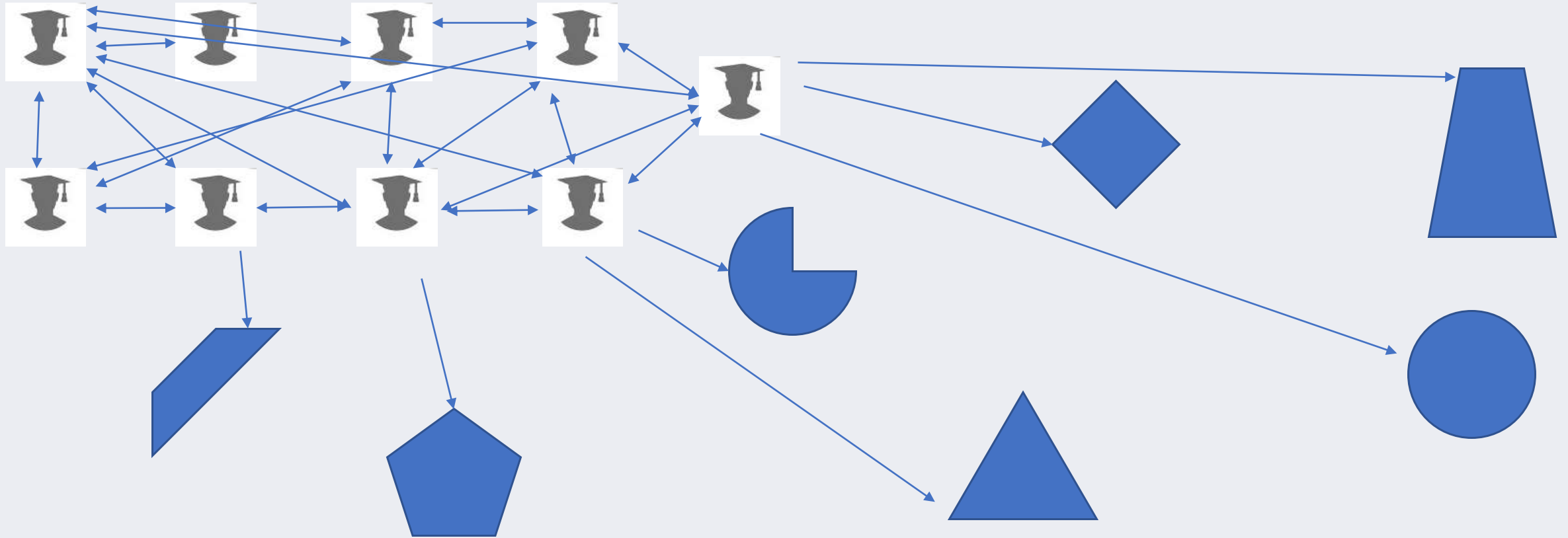


Manager  
(Fits bits on calendar)

# And how the communicate?

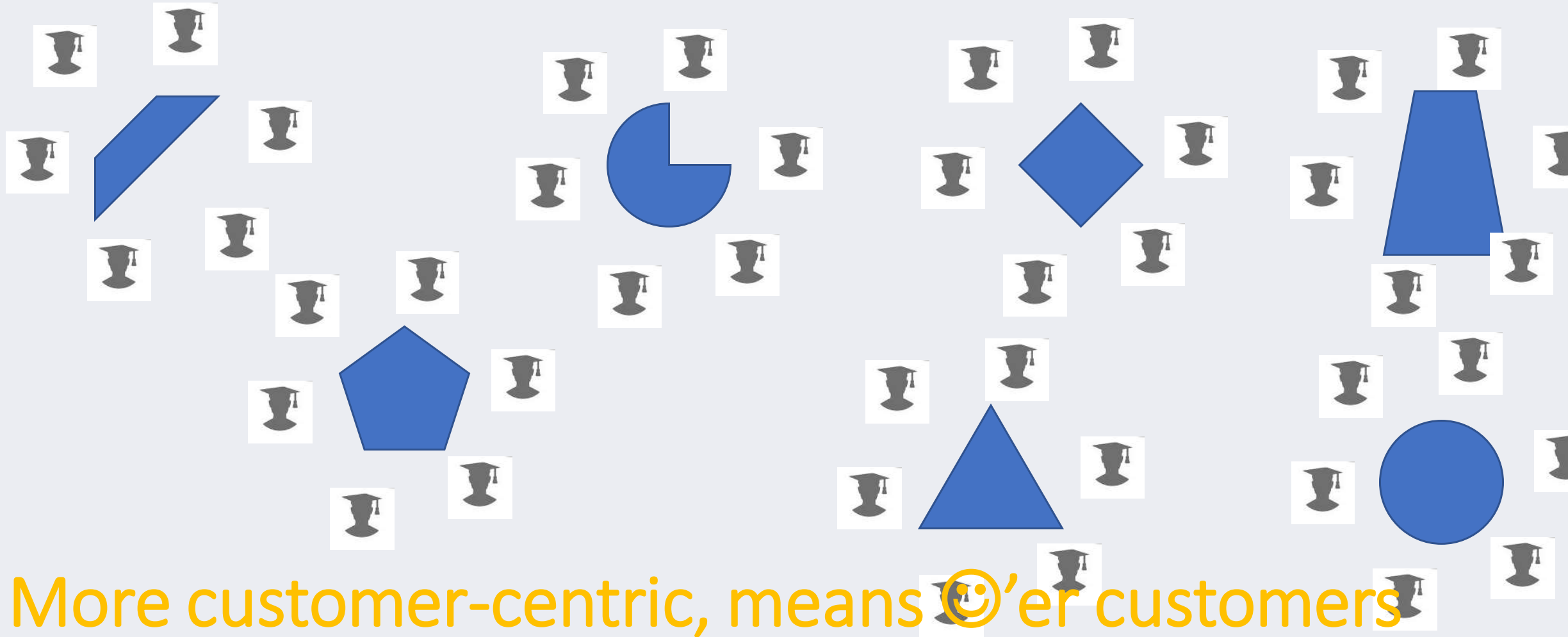


# And what products do they support?



## Is this customer-centric?

# Product Teams

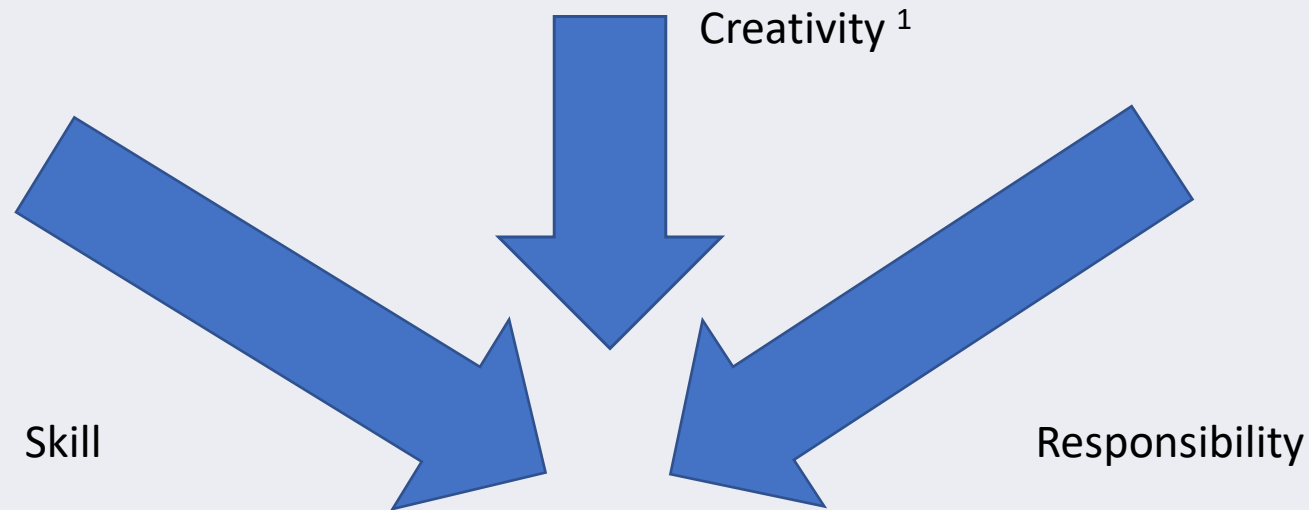


More customer-centric, means ☺'er customers



# Don't fire your way to innovation

- Instead, seek to merge skill with responsibility
- Your people, instead of being glorified bit managers, can do what they are good at – injecting creativity



<sup>1</sup> what people are still better at than computers, for the time being

# But what about governance?

- “We standardized on Java!”
- “We only run [insert OS here]!”
- “We only use [insert framework here]!”
- The above are the result of trying to solve the Principal-agent problem: “the agent knows some things that the principal does not, and it is too costly for the principal to monitor every action that the agent take” (“Misbehaving: The Making of Behavioral Economics.”)
- So instead of monitoring every action, the principal (IT Governance) reduces the jurisdiction(?), and thus cost, to govern.

# The dark underbelly of limiting choice

- By being prescriptive about standards, you've succeeded in reducing the governance cost
- (Applause!)
- But you've prohibitively raised the cost of experimentation.
- Your talented high-dollar developers can only use the hammer you've provided.
- (Insert Hammer/Nail)
- Don't be surprised if your talent leaves
- Or worse, you talent only starts to see nails.

# So you are proposing the wild west, except with modern weapons?

- No.
- With evolutionary architectures, strict data access policies, serverless technologies, and infrastructure-as-code, you can simultaneously allow variation and have cheaper, more reliable IT governance that you ever imagined possible.
- The tools aren't here yet, but they're coming quick.



# Adoption Paths

- Build vs Buy equation begins to change
  - If you have to customize \*anything\*, you may want to build
- Come back to activation energy
  - Experiment with internal tools
  - Greenfield “Shining City on the Hill” model
  - Skunkworks
  - Others?