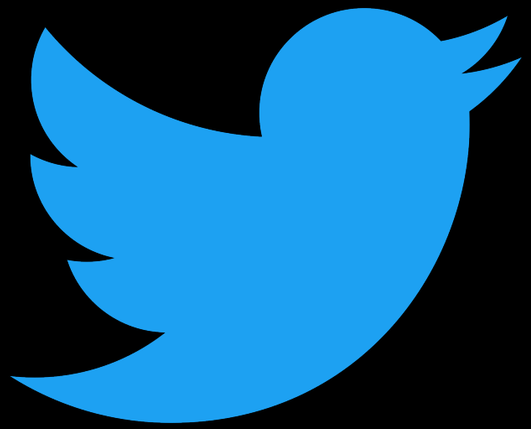
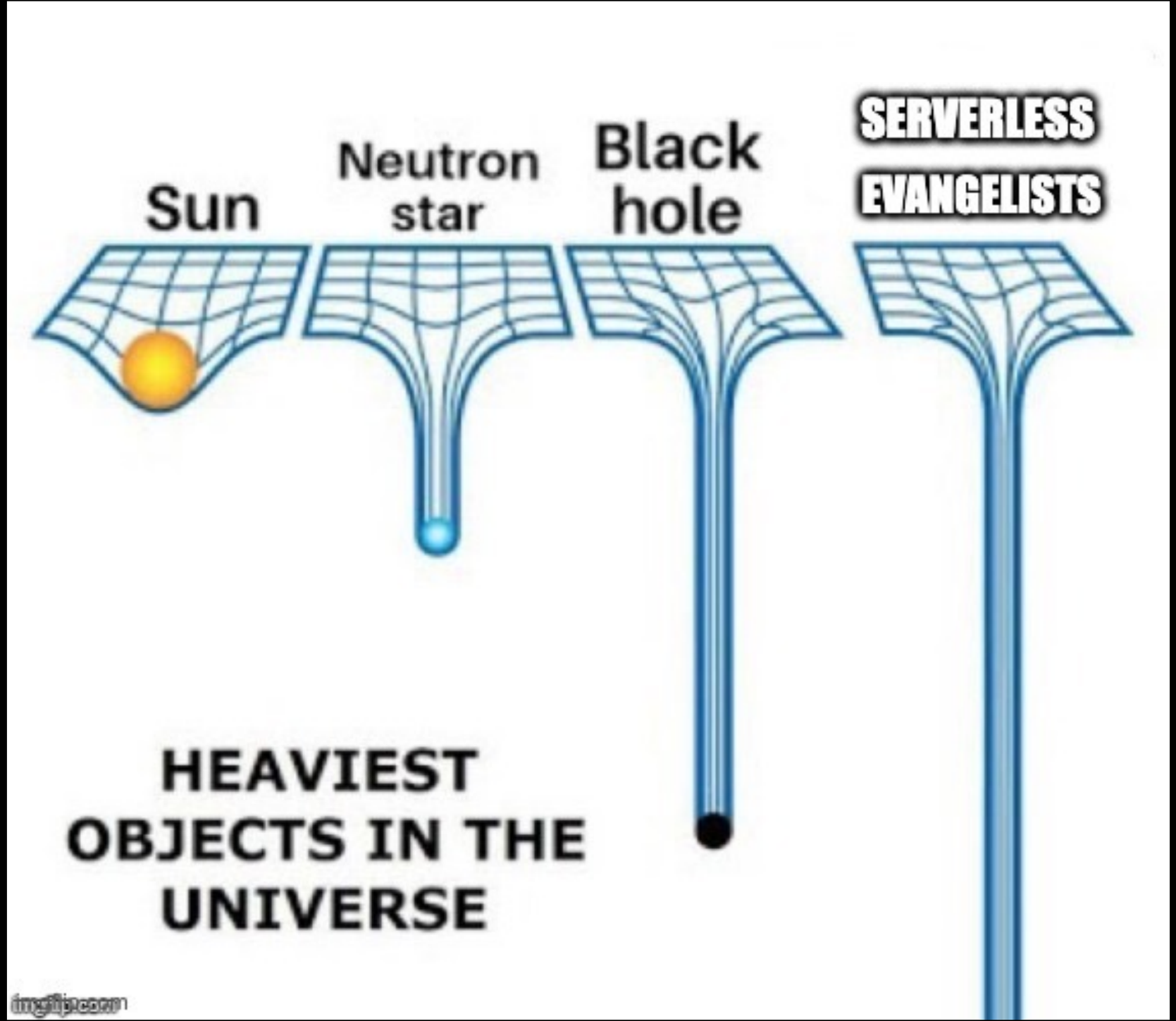
Evitando el *cosplay* en el diseño de software

Pablo Bermejo 

Distinguished Software Architect
@DXCSpain

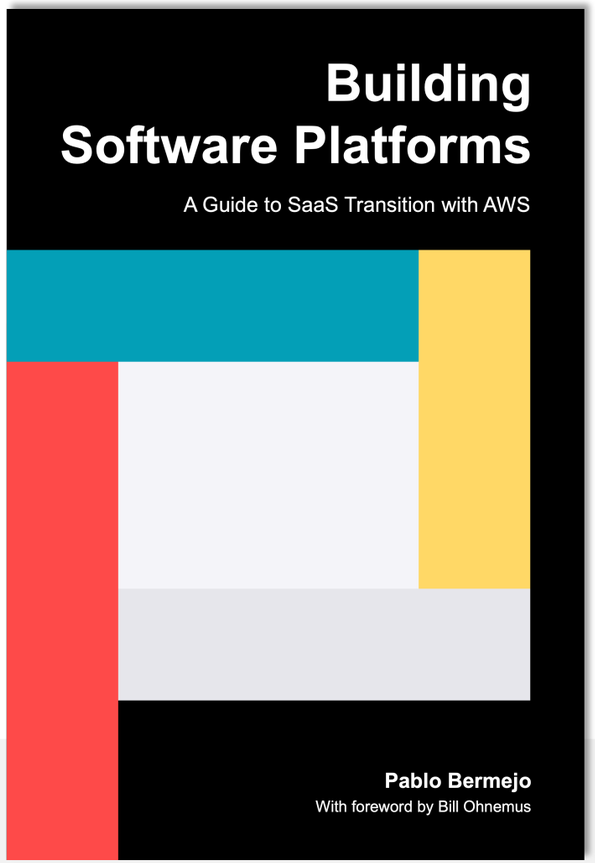


@peibolsang


Sun
Neutron star
Black hole
SERVERLESS EVANGELISTS
HEAVIEST OBJECTS IN THE UNIVERSE
imgflip.com

available at amazon

Leanpub



leanpub.com/software-platforms


Building
Software Platforms
ME
A TECHNICAL
CHALLENGE AT WORK
imgflip.com


MY POINT
YOU

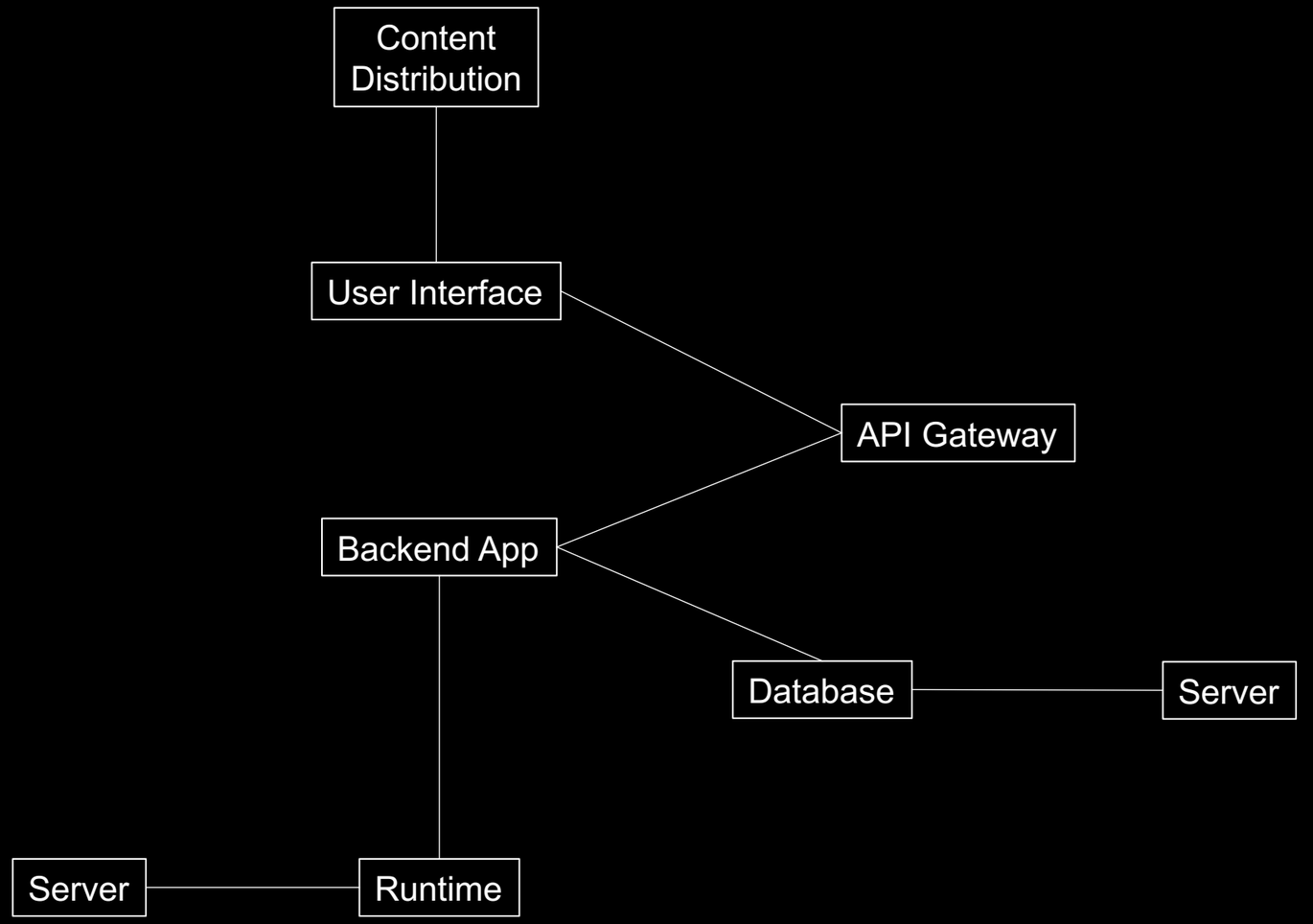
Parte 1 

Estrategia y dirección tecnológica

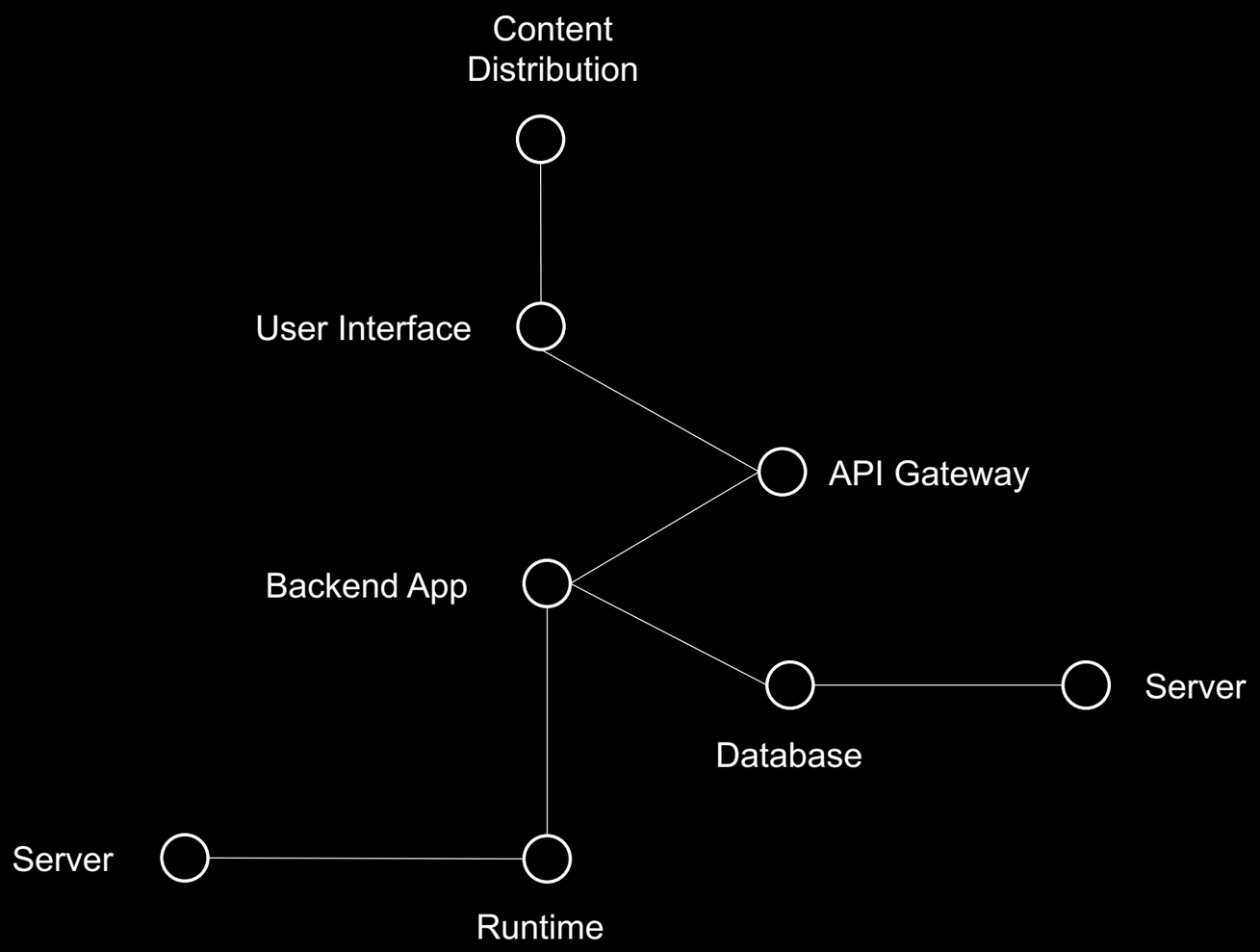
¿Qué usamos para saber dónde tenemos que ir?

Mapas de Wardley

DIAGRAM

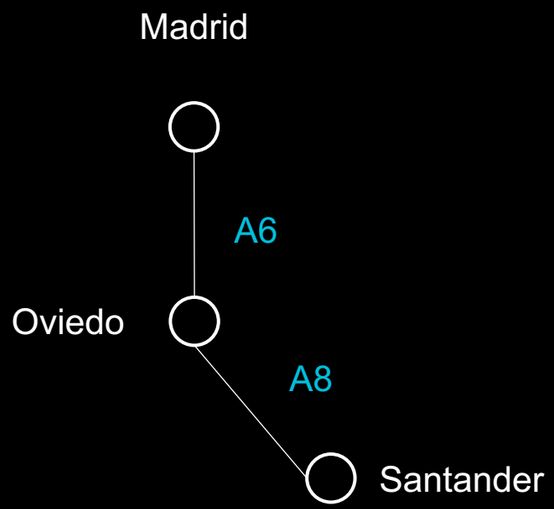

Content Distribution
User Interface
API Gateway
Backend App
Database
Server
Server
Runtime

DIAGRAM


Content
Distribution
User Interface
API Gateway
Backend App
Server
Database
Server
Runtime

¿Es un mapa?

DIAGRAM

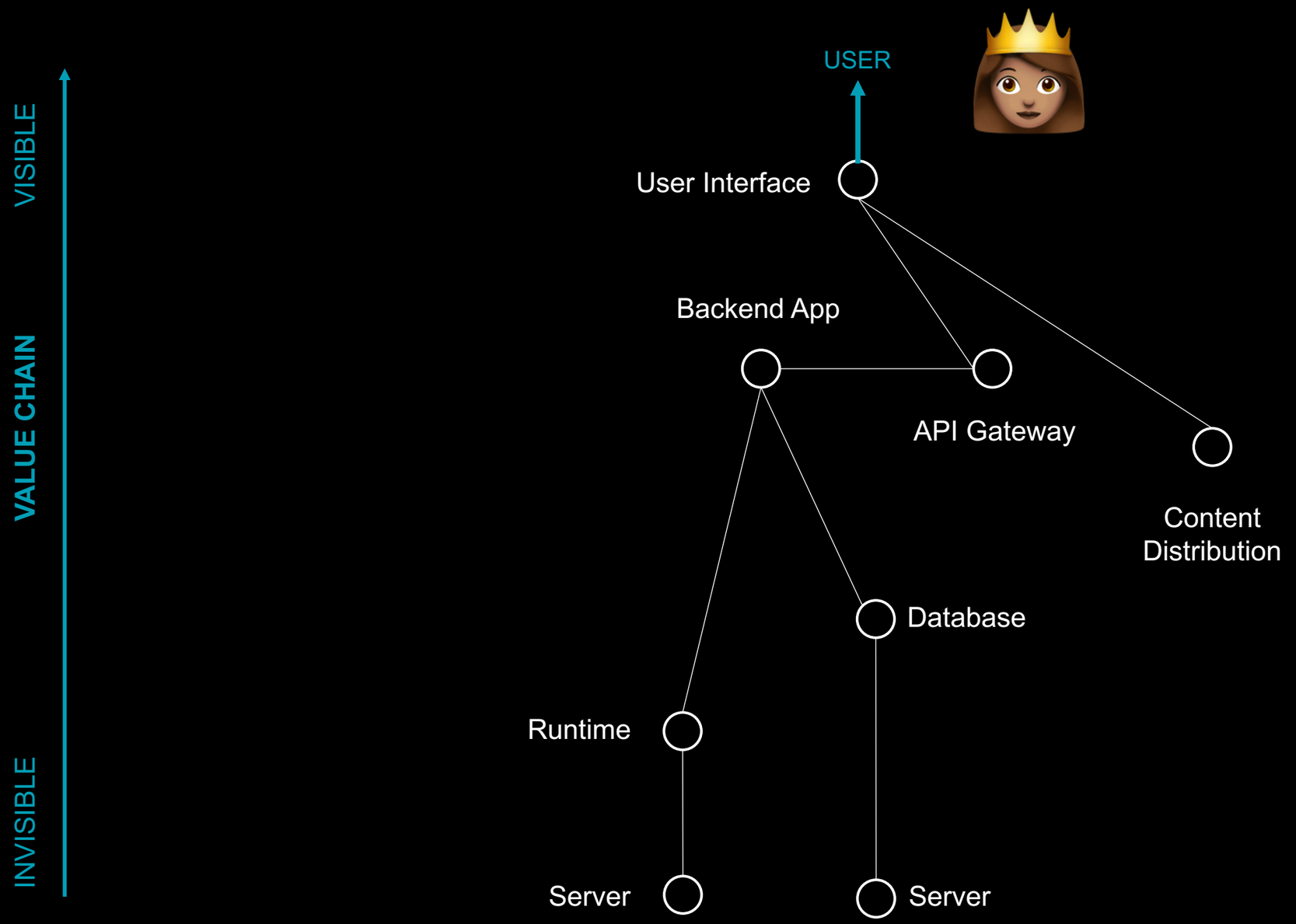


No! ❌

Nos falta saber nuestra posición con respecto al norte

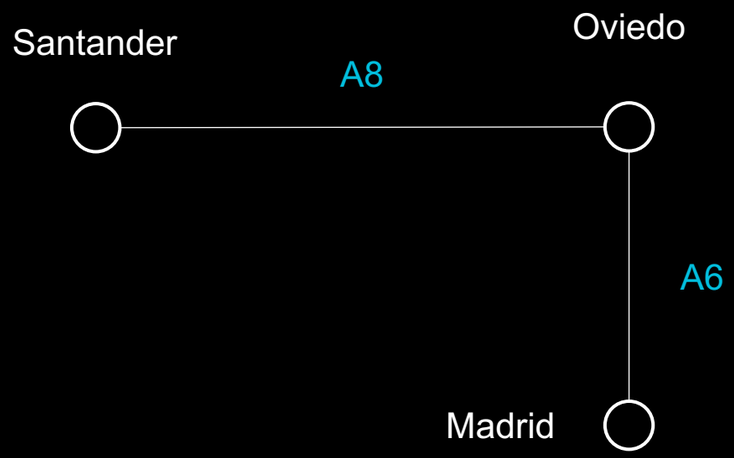
¿Cuál
es el norte?

DIAGRAM
+ position



Y ahora,
¿es un mapa?

DIAGRAM



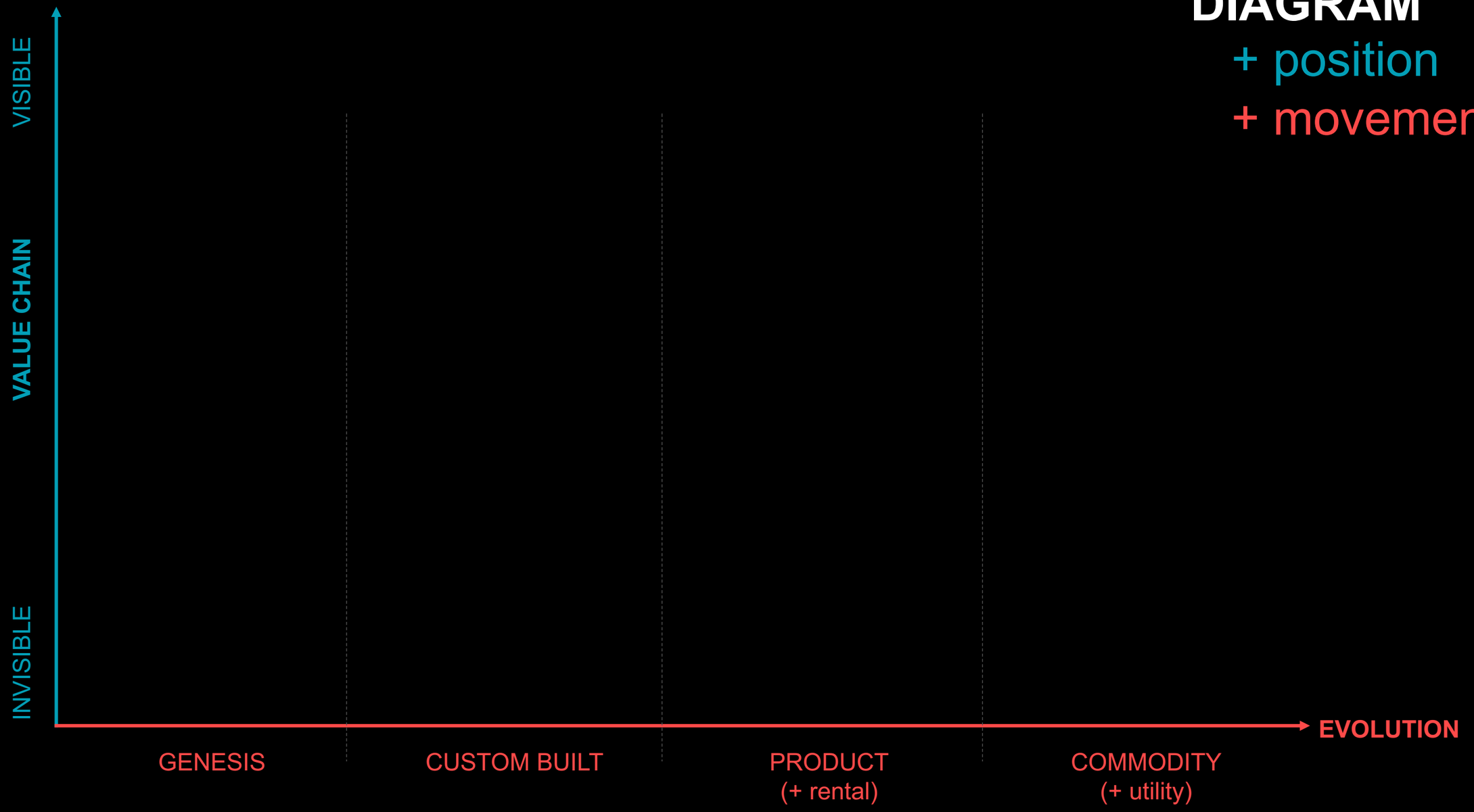
No! 

Nos falta determinar el movimiento para saber la dirección

DIAGRAM

+ position

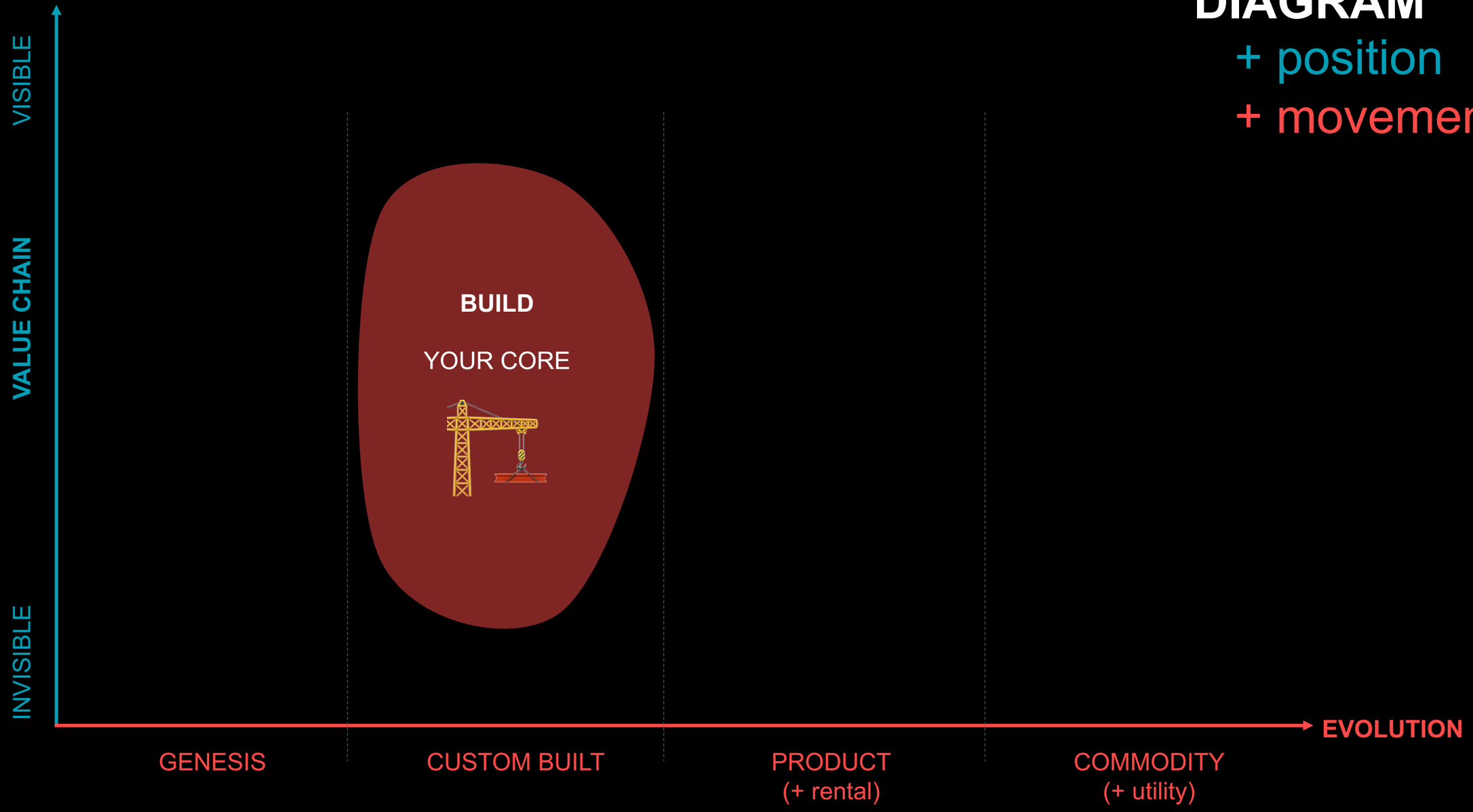
+ movement



DIAGRAM

+ position

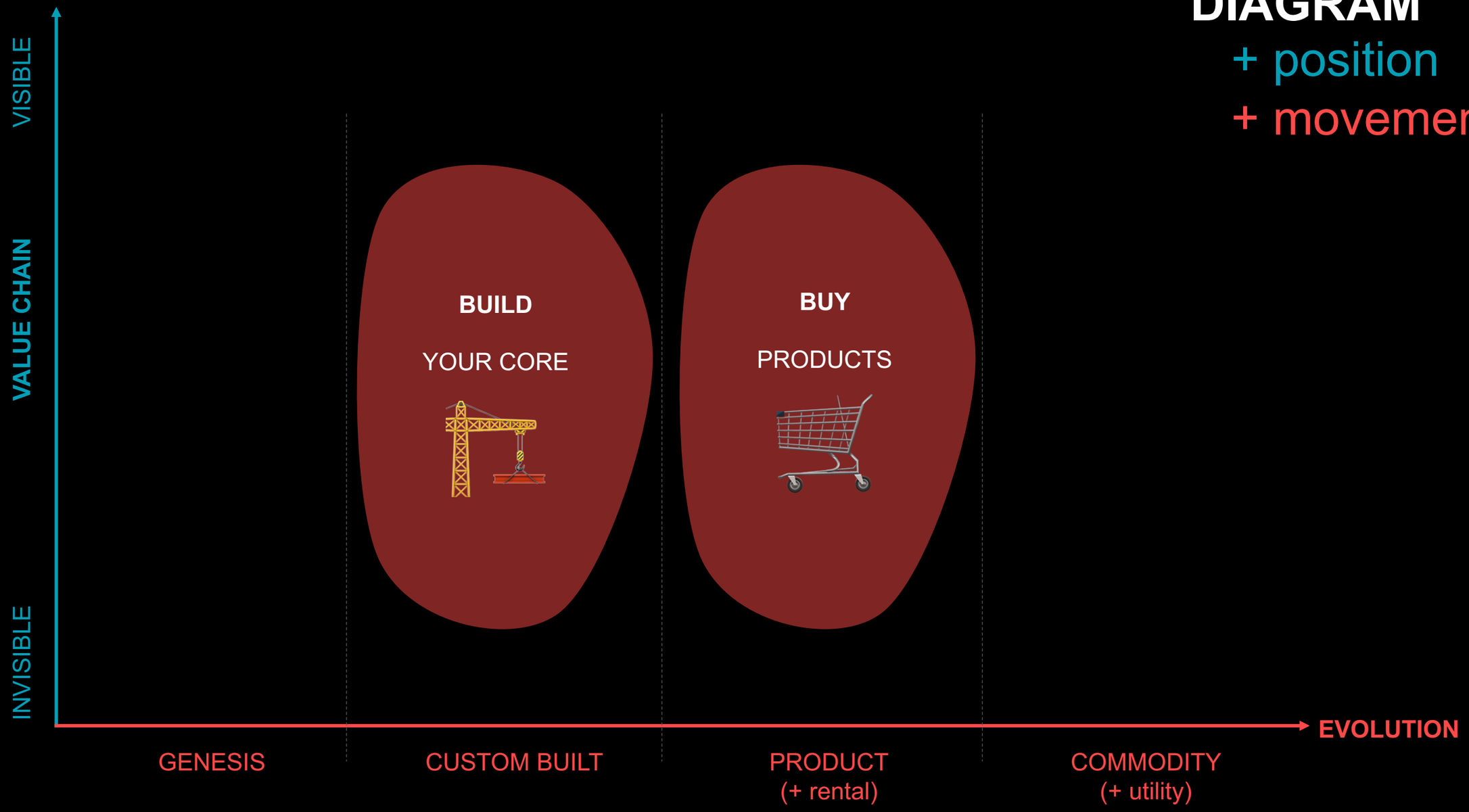
+ movement



DIAGRAM

+ position

+ movement



DIAGRAM

+ position

+ movement

VISIBLE

VALUE CHAIN

INVISIBLE

BUILD

YOUR CORE

BUY

PRODUCTS

SUBSCRIBE

TO MANAGED SERVICES

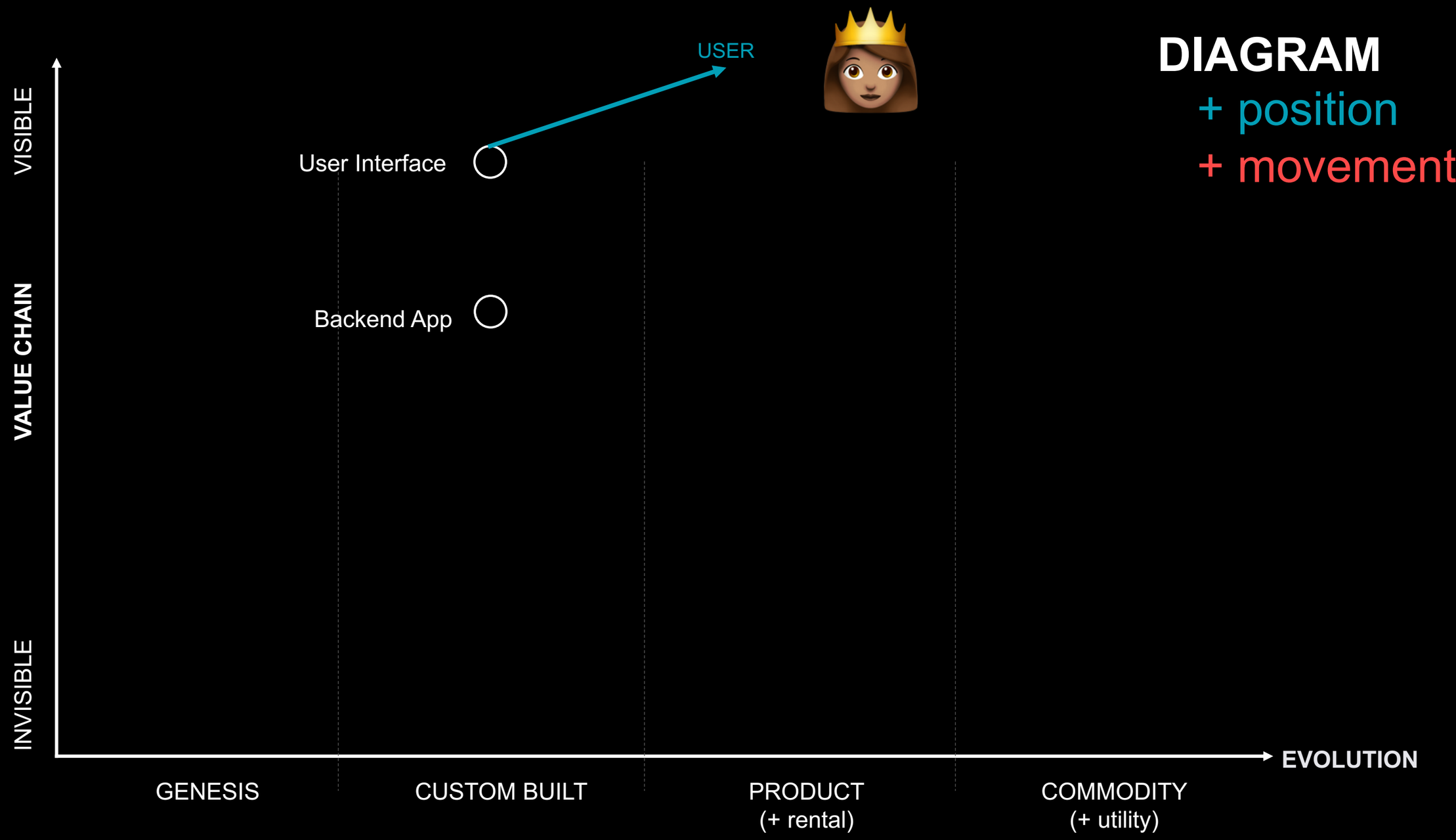
EVOLUTION

GENESIS

CUSTOM BUILT

PRODUCT
(+ rental)

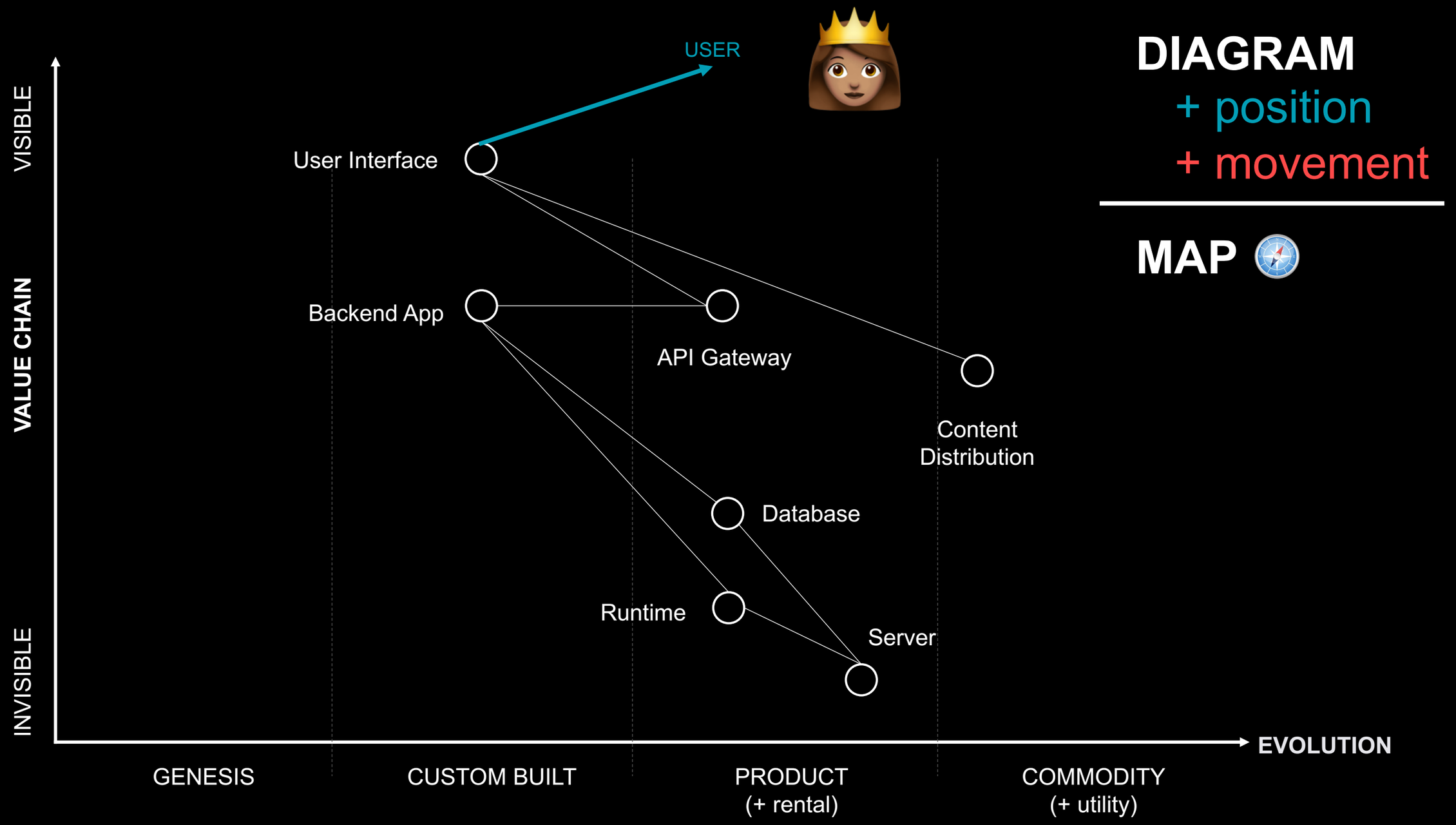
COMMODITY
(+ utility)



DIAGRAM

+ position

+ movement

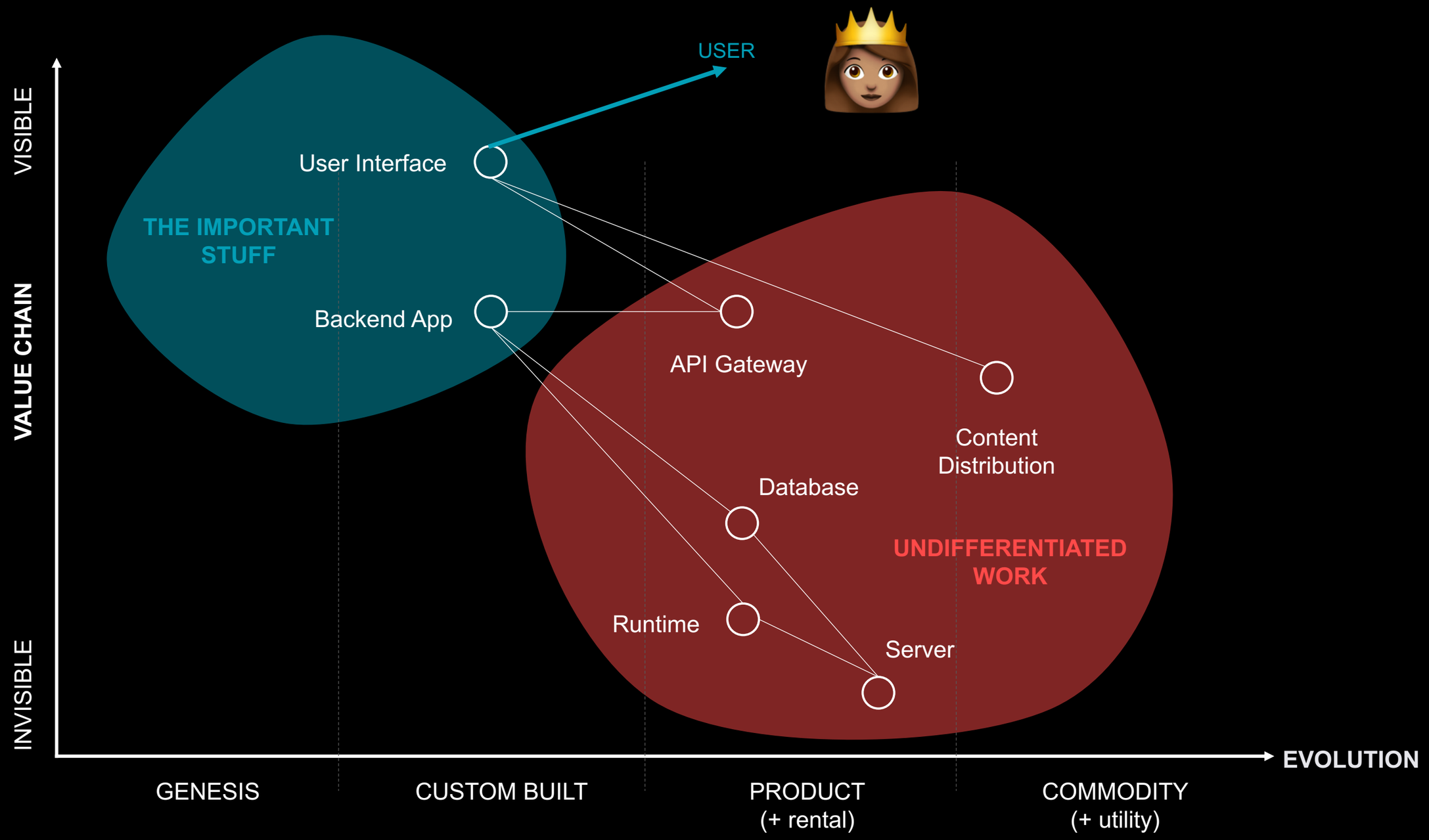


DIAGRAM

+ position

+ movement

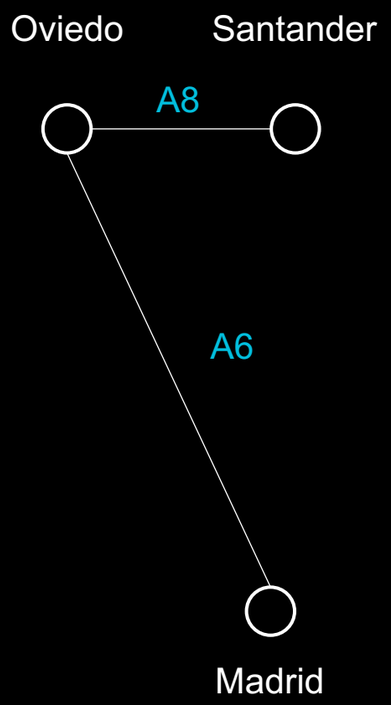
MAP

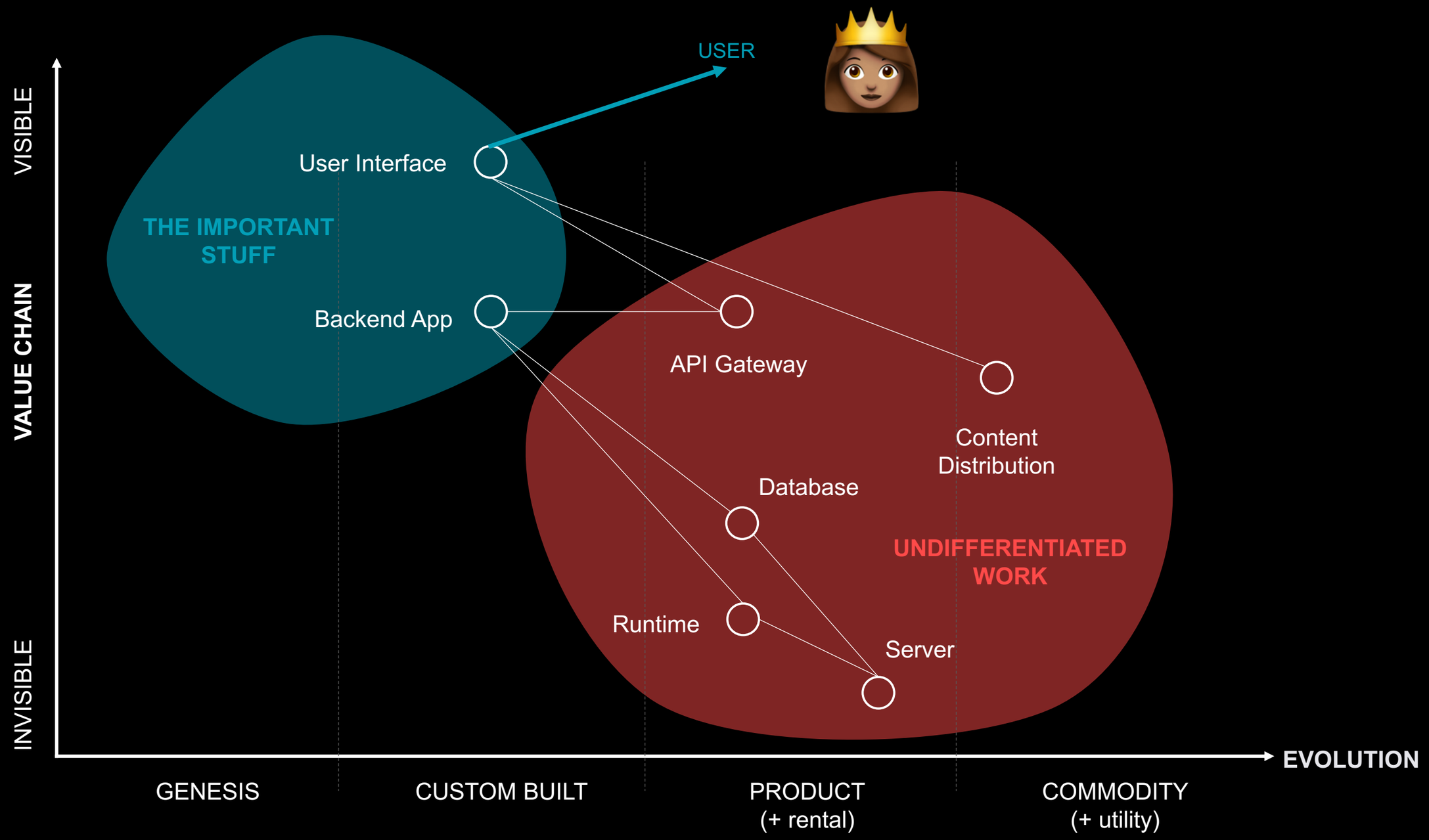

USER
VISIBLE
VALUE CHAIN
INVISIBLE
User Interface
THE IMPORTANT STUFF
Backend App
API Gateway
Content Distribution
Database
UNDIFFERENTIATED WORK
Runtime
Server
EVOLUTION
GENESIS
CUSTOM BUILT
PRODUCT (+ rental)
COMMODITY (+ utility)

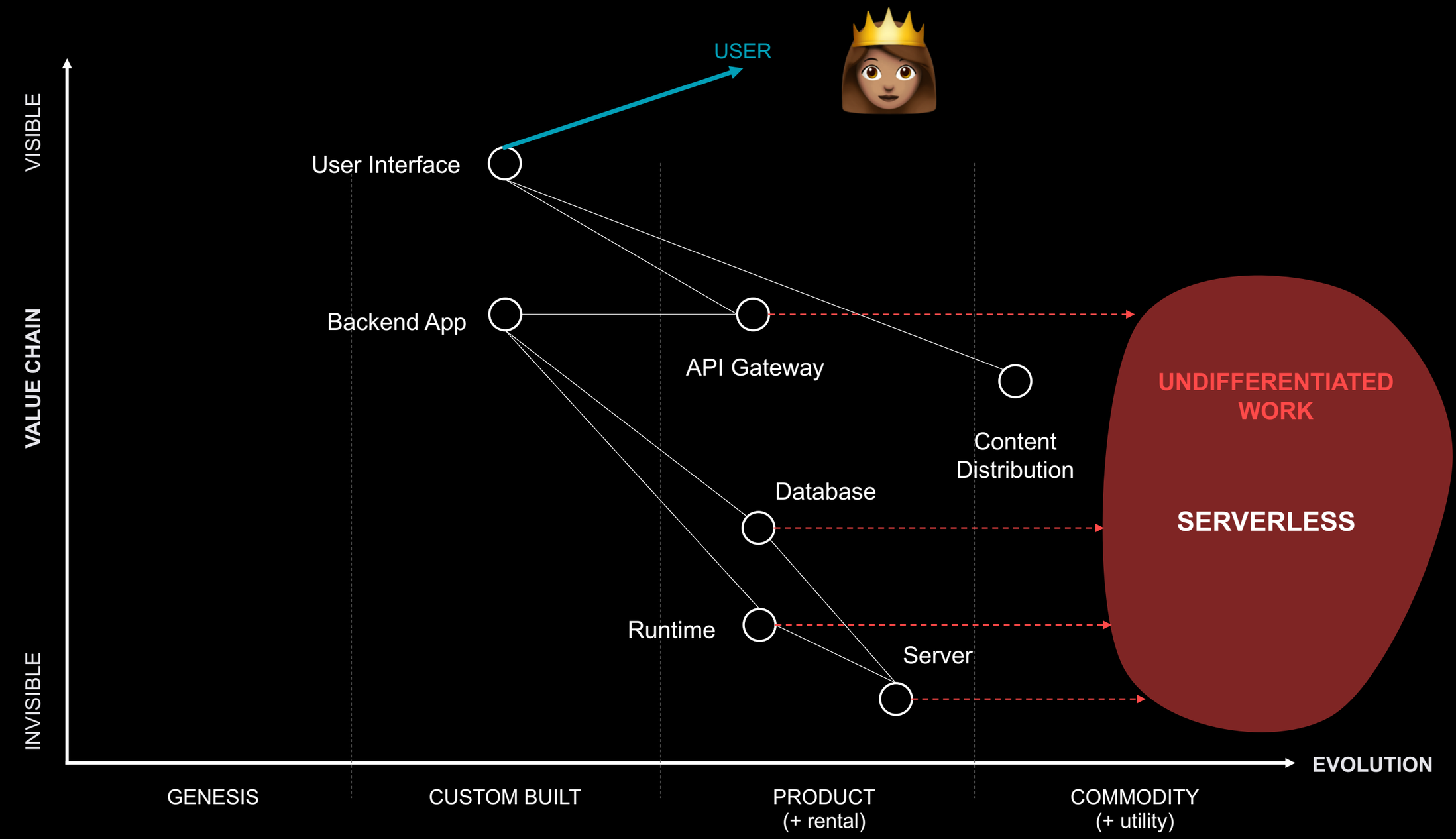
Hemos pasado de un diagrama de arquitectura básico a un artefacto de estrategia tecnológica

Y el movimiento, ¿dónde está?

MAP

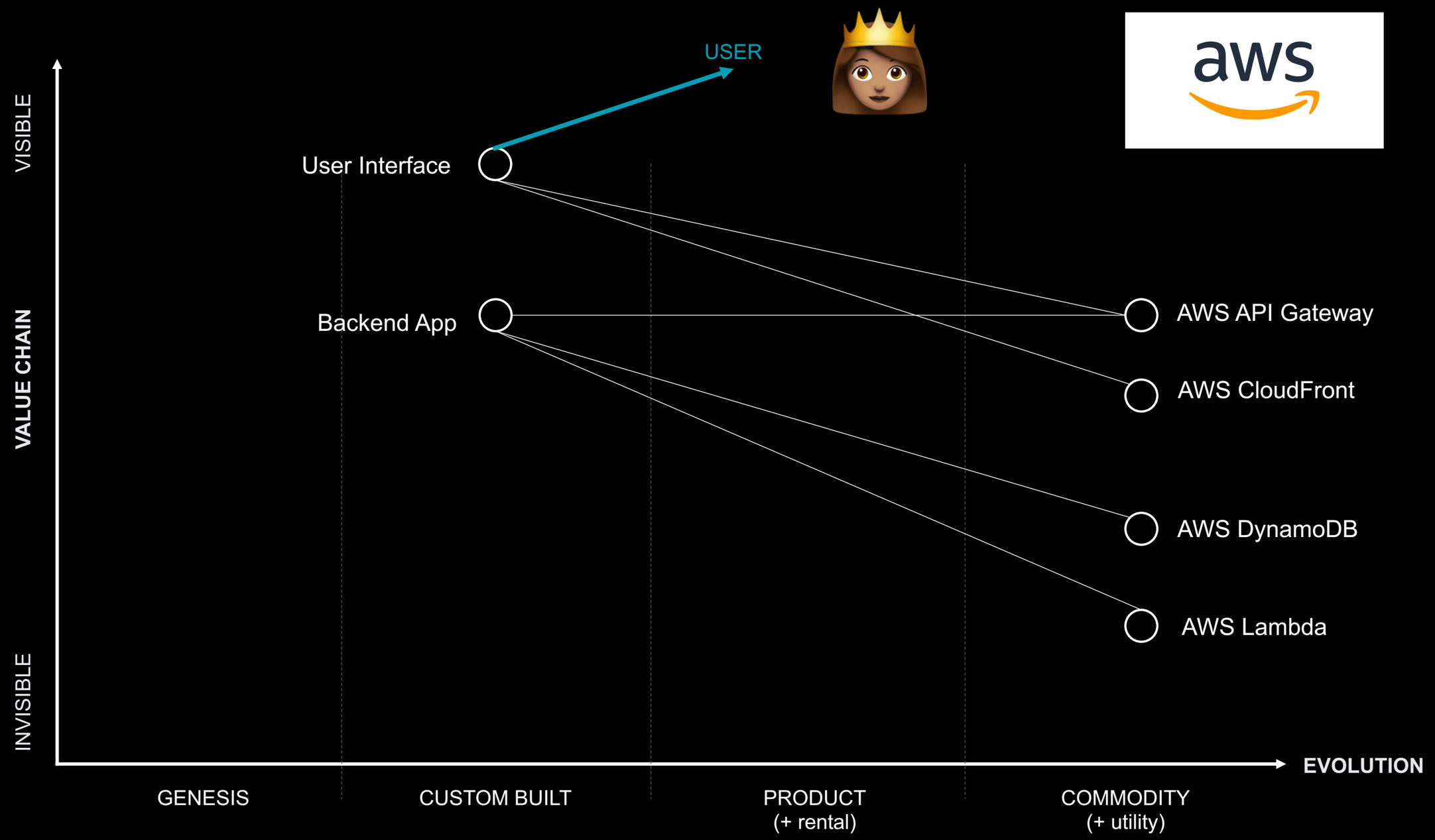



USER
VISIBLE
VALUE CHAIN
INVISIBLE
User Interface
THE IMPORTANT STUFF
Backend App
API Gateway
Content Distribution
Database
UNDIFFERENTIATED WORK
Runtime
Server
EVOLUTION
GENESIS
CUSTOM BUILT
PRODUCT (+ rental)
COMMODITY (+ utility)


USER
VISIBLE
VALUE CHAIN
INVISIBLE
User Interface
Backend App
API Gateway
Content Distribution
Database
Runtime
Server
UNDIFFERENTIATED WORK
SERVERLESS
EVOLUTION
GENESIS
CUSTOM BUILT
PRODUCT (+ rental)
COMMODITY (+ utility)

Esto es serverless 

No queremos que nuestros equipos de desarrollo hagan cosplay  de proveedores cloud.

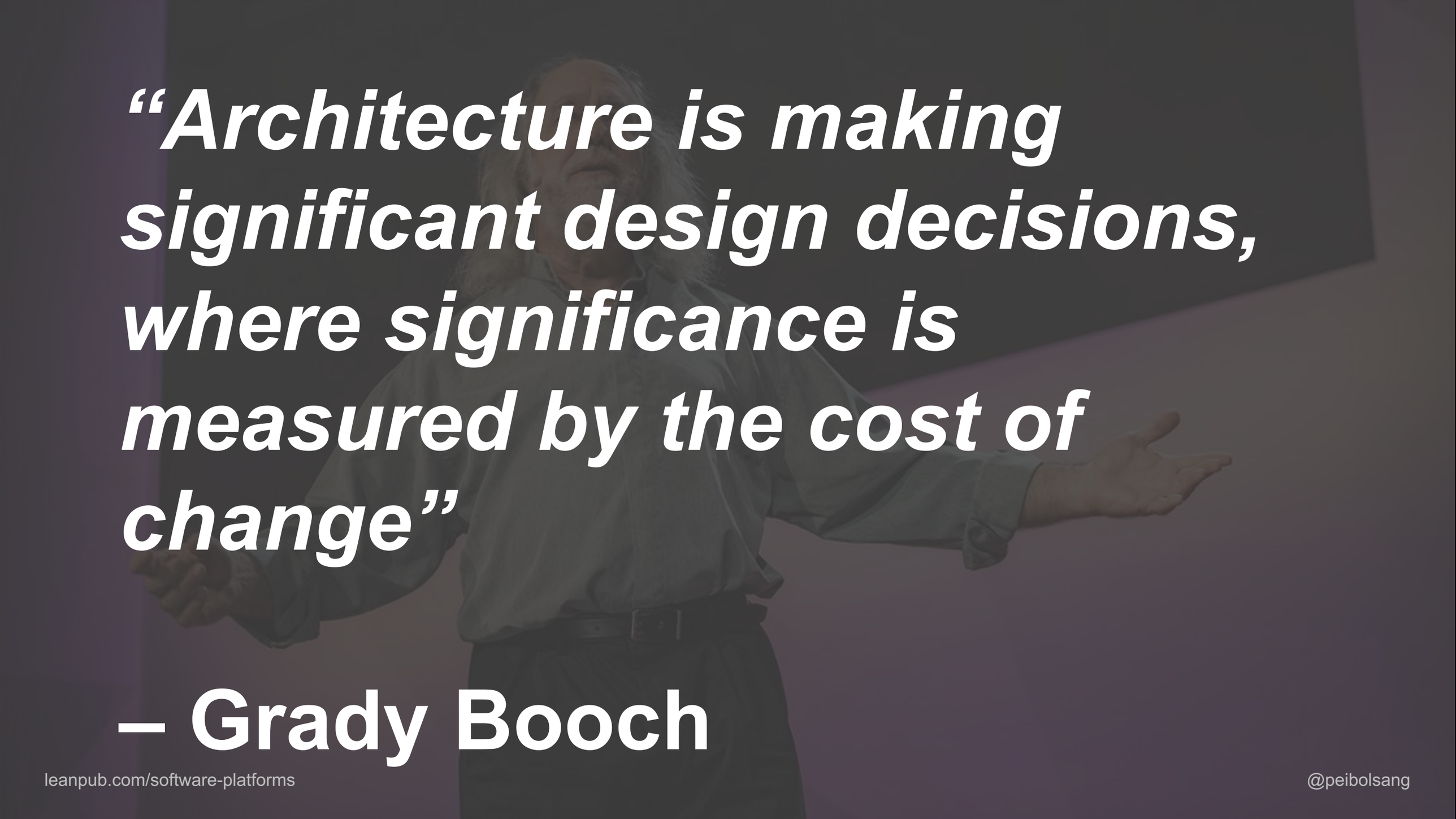

USER
VISIBLE
VALUE CHAIN
INVISIBLE
User Interface
Backend App
AWS API Gateway
AWS CloudFront
AWS DynamoDB
AWS Lambda
EVOLUTION
GENESIS
CUSTOM BUILT
PRODUCT
(+ rental)
COMMODITY
(+ utility)


IS THIS A NEW CLOUD PROVIDER?
INTERNAL PRODUCT TEAMS MANAGING INFRASTRUCTURE AND CLUSTERS
COMPANIES
imgflip.com

¿Qué impacto tiene serverless en la arquitectura del software 📐?

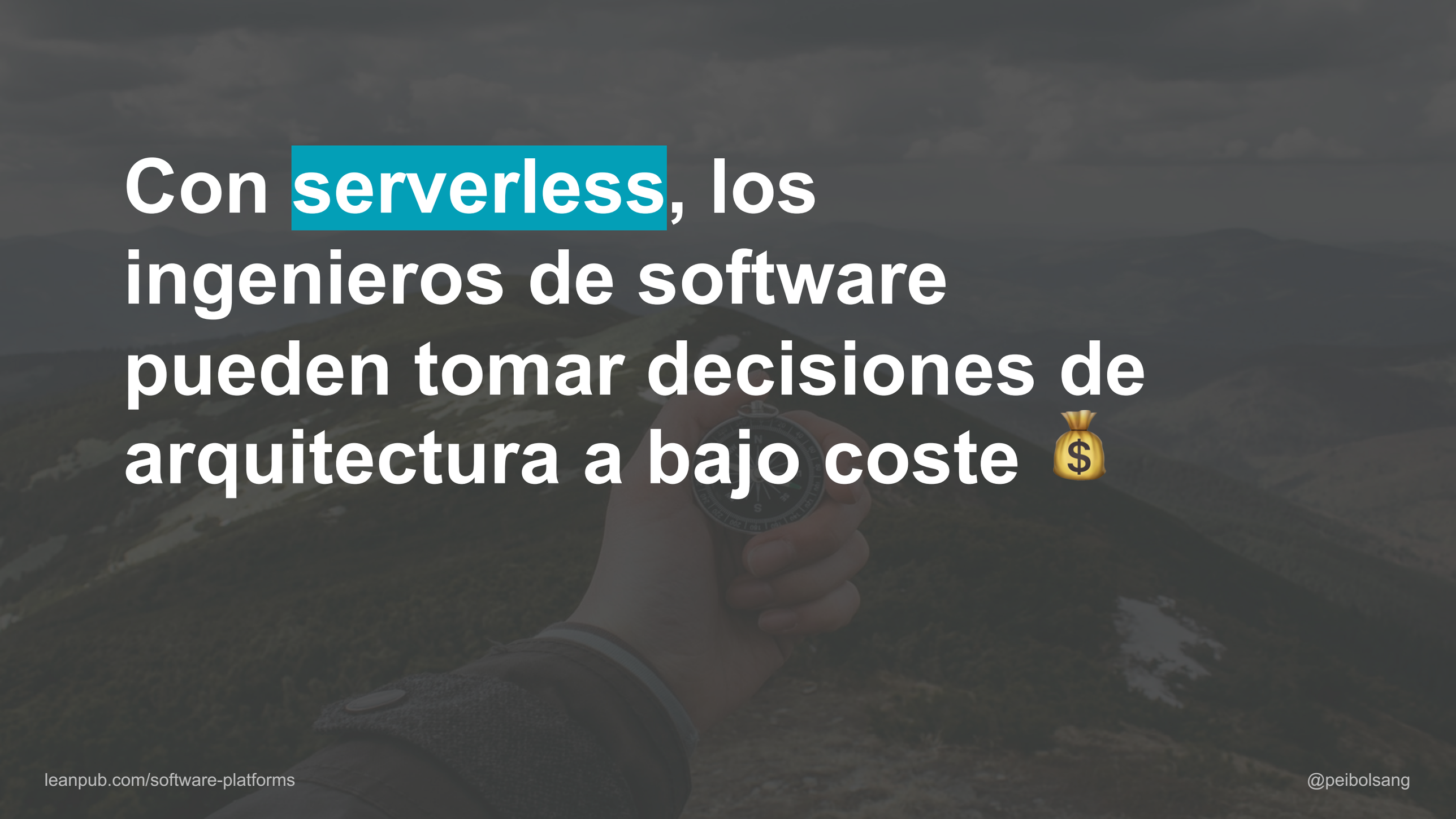
“Serverless does not mean architecture-less”

– Gregor Hohpe



“Architecture is making significant design decisions, where significance is measured by the cost of change”

– Grady Booch

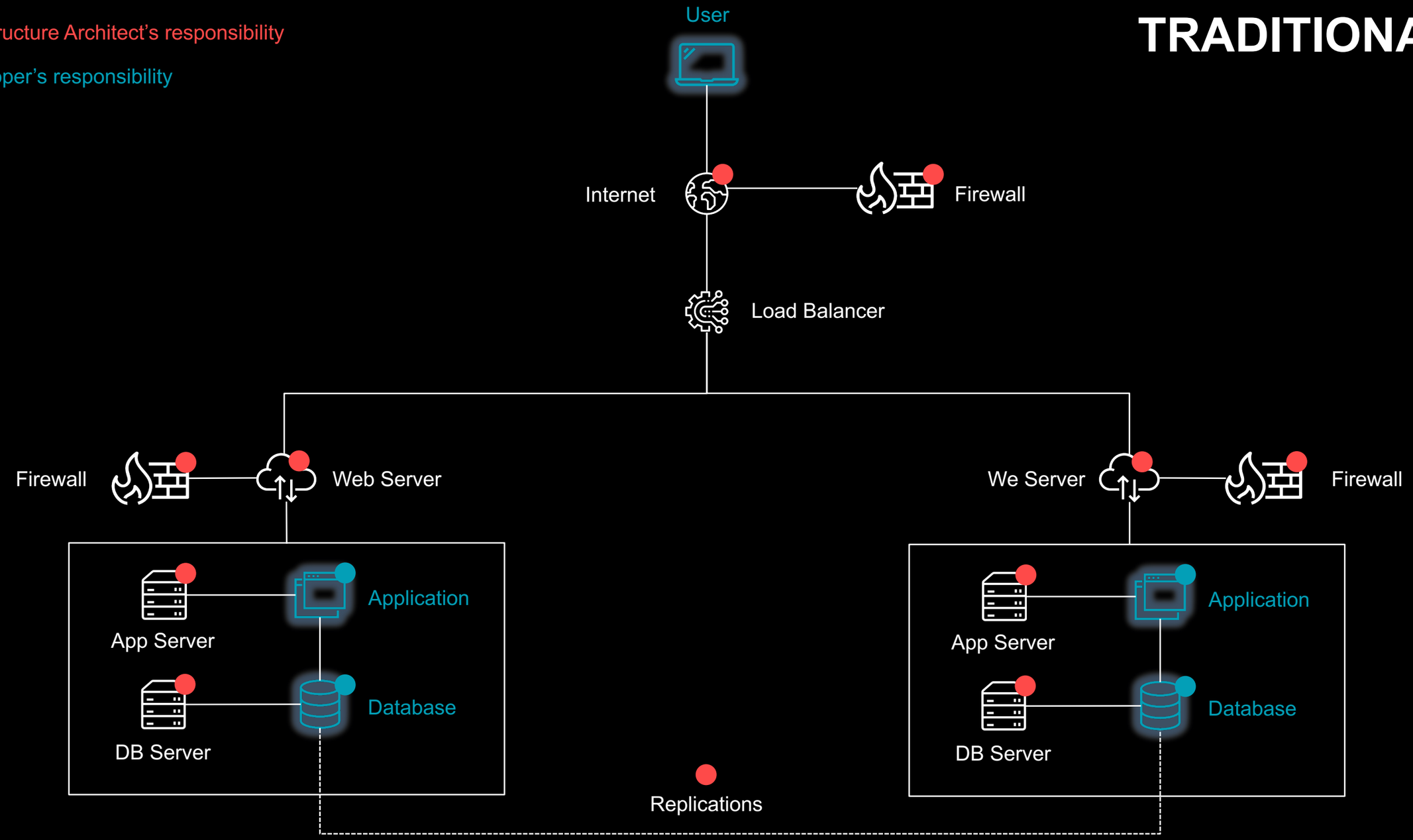


Con serverless, los ingenieros de software pueden tomar decisiones de arquitectura a bajo coste 💰

TRADITIONAL

Infrastructure Architect's responsibility

Developer's responsibility

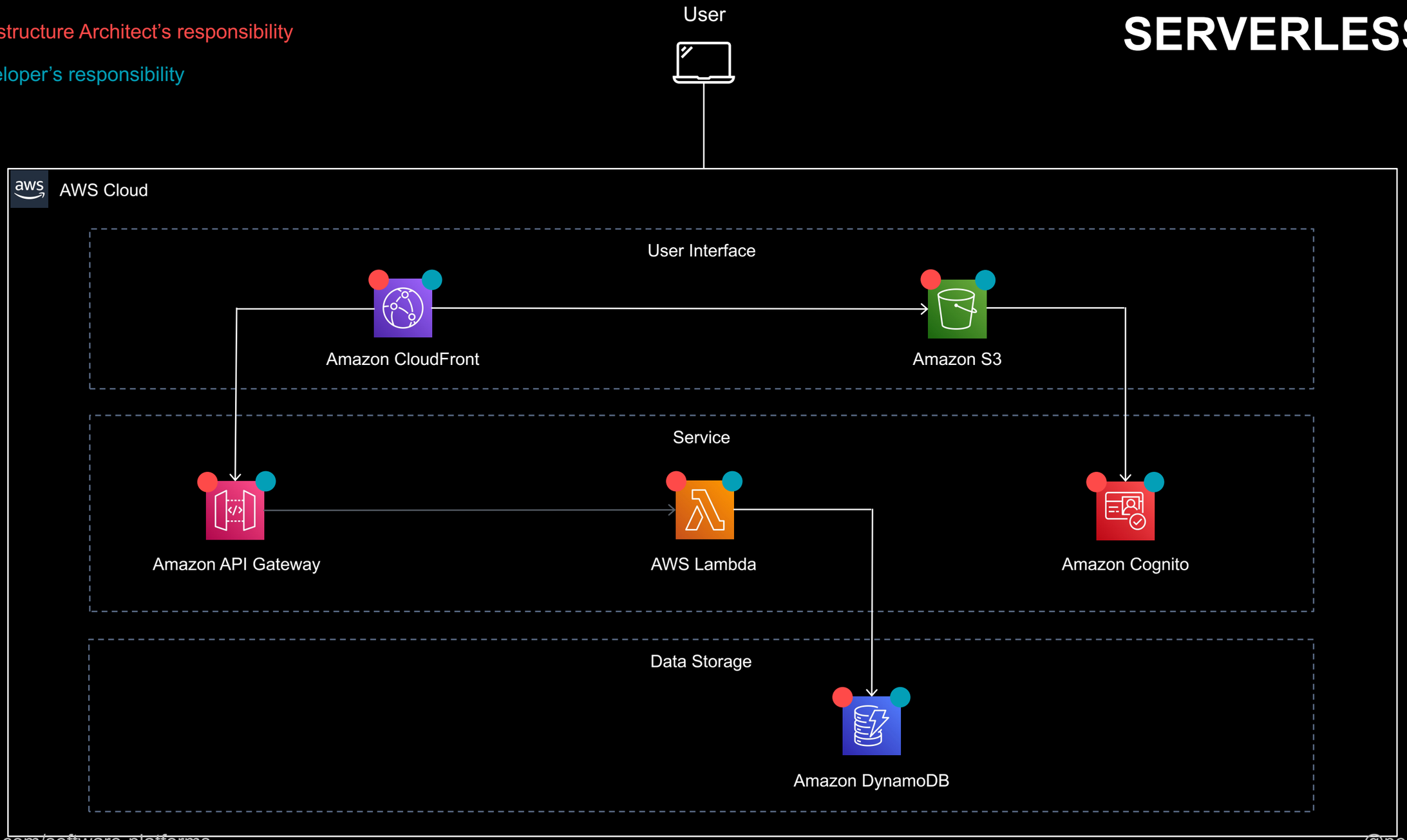


leanpub.com/software-platforms

@peibolsang

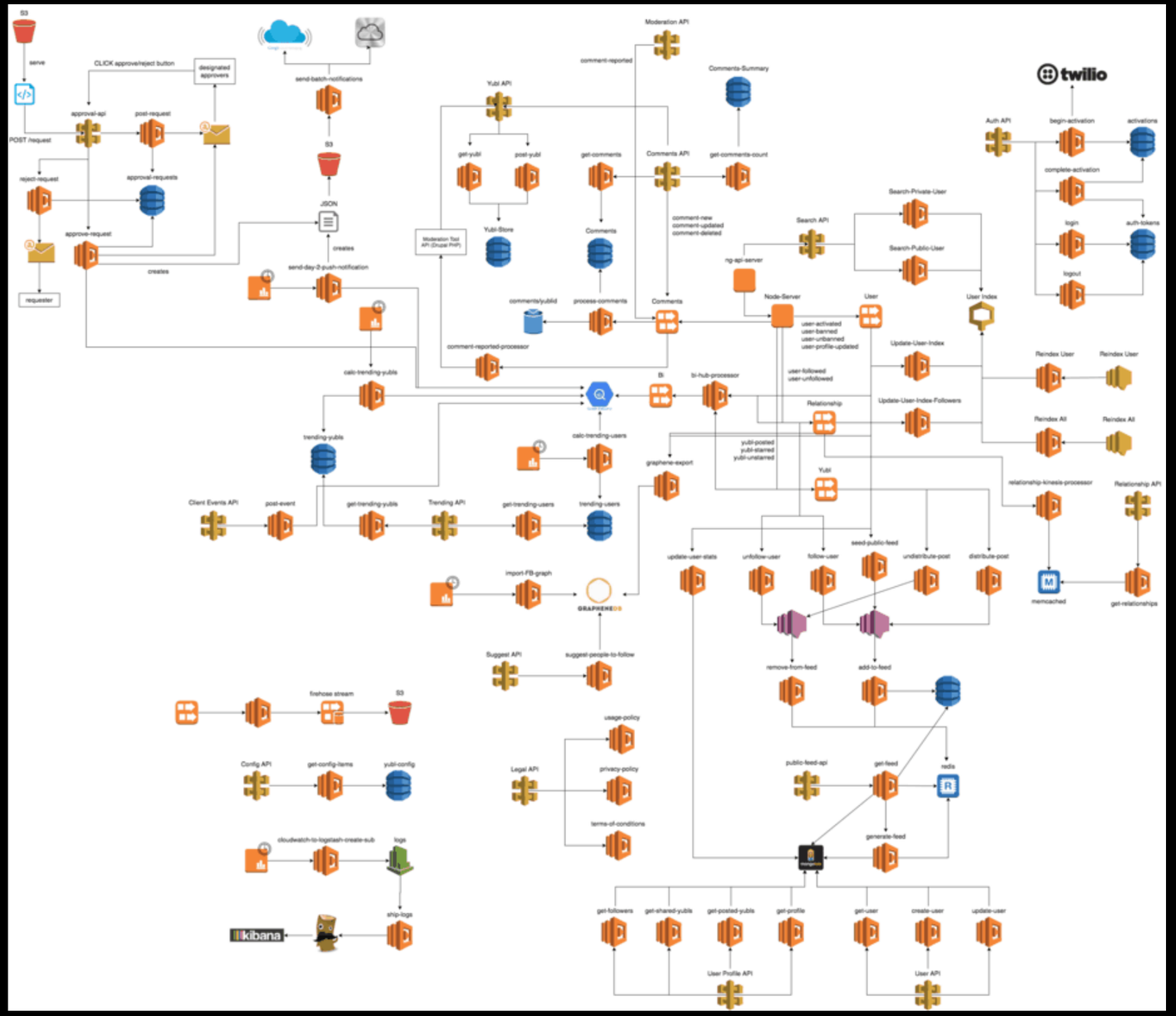
SERVERLESS

- Infrastructure Architect's responsibility
- Developer's responsibility

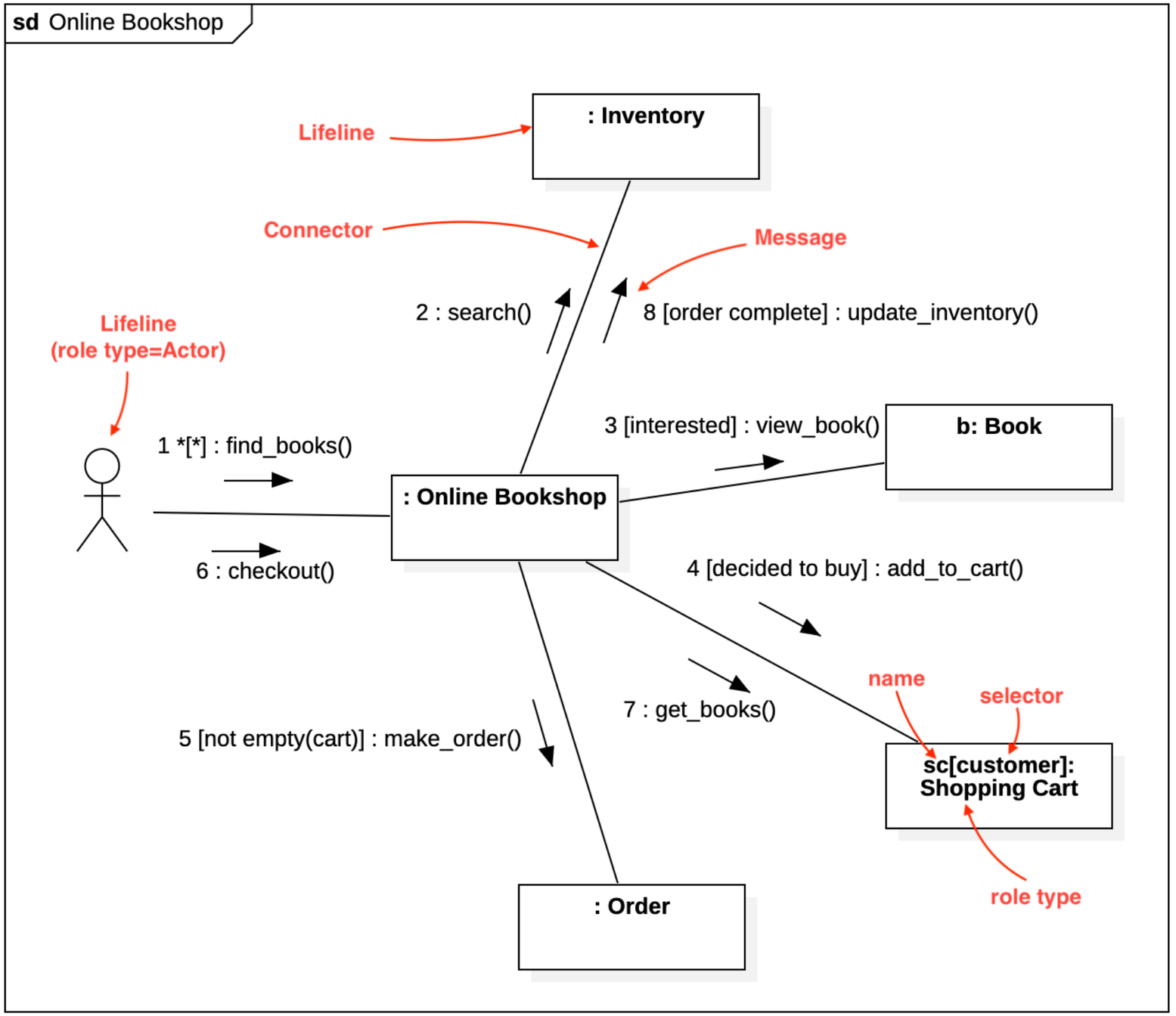


No parece complejo, ¿verdad?

HOLD MY


S3
serve
CLICK approve/reject button
designated approvers
POST /request
approval-api
post-request
reject-request
approval-requests
approve-request
requester
creates
send-batch-notifications
JSON
send-day-2-push-notification
Moderation API
comment-reported
Comments-Summary
Yubl API
get-yubl
post-yubl
Yubl-Store
Moderation Tool API (Drupal PHP)
get-comments
Comments API
get-comments-count
comment-new
comment-updated
comment-deleted
Comments
commentsYublId
process-comments
comment-reported-processor
Search API
Search-Private-User
Search-Public-User
ng-api-server
Node-Server
User
User Index
user-activated
user-banned
user-unbanned
user-profile-updated
twilio
Auth API
begin-activation
activations
complete-activation
login
auth-tokens
logout
Update-User-Index
Reindex User
Reindex All
calc-trending-yubls
trending-yubls
bi-hub-processor
user-followed
user-unfollowed
Relationship
Update-User-Index-Followers
calc-trending-users
trending-users
graphene-export
yubl-posted
yubl-starred
yubl-unstarred
Yubl
Client Events API
post-event
get-trending-yubls
Trending API
get-trending-users
relationship-kinesis-processor
Relationship API
memcached
get-relationships
update-user-stats
unfollow-user
follow-user
seed-public-feed
undistribute-post
distribute-post
import-FB-graph
GRAPHENEDB
Suggest API
suggest-people-to-follow
remove-from-feed
add-to-feed
firehose stream
usage-policy
Config API
get-config-items
yubl-config
Legal API
privacy-policy
terms-of-conditions
public-feed-api
get-feed
redis
generate-feed
cloudwatch-to-logstash-create-sub
logs
ship-logs
kibana
get-followers
get-shared-yubls
get-posted-yubls
get-profile
get-user
create-user
update-user
User Profile API
User API

¿Qué es ese diagrama en realidad?


sd Online Bookshop
Lifeline
: Inventory
Connector
Message
2 : search()
8 [order complete] : update_inventory()
Lifeline
(role type=Actor)
1 *[*] : find_books()
3 [interested] : view_book()
b: Book
: Online Bookshop
6 : checkout()
4 [decided to buy] : add_to_cart()
7 : get_books()
name
selector
5 [not empty(cart)] : make_order()
sc[customer]:
Shopping Cart
: Order
role type

La realidad es que con serverless básicamente estamos desplegando nuestros diagramas de colaboración UML 

La arquitectura de software significa estar en el negocio de los trade-offs 

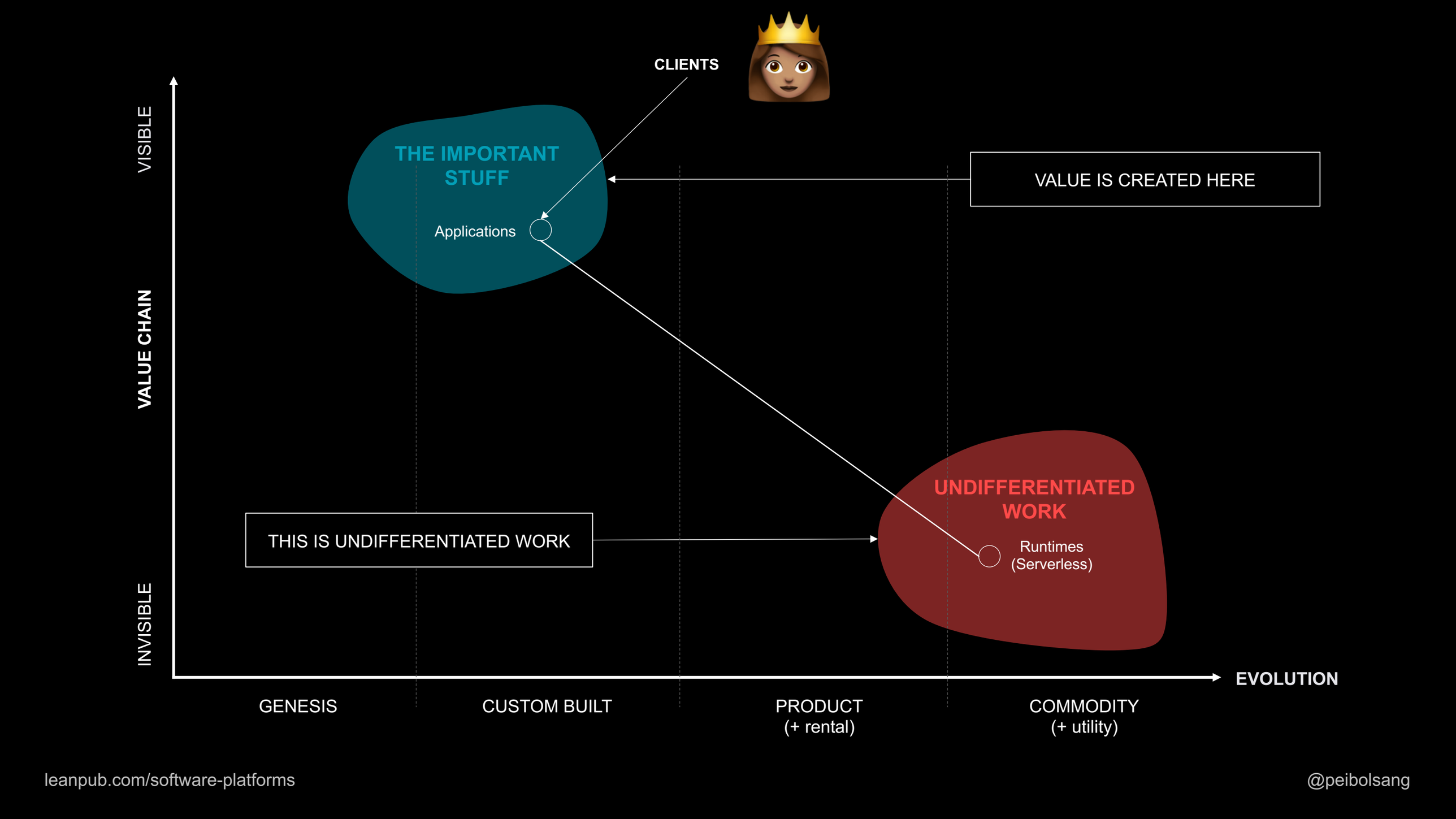
“Engineering trade-offs and stupidity are undistinguishable if you are unaware of the trade-offs”

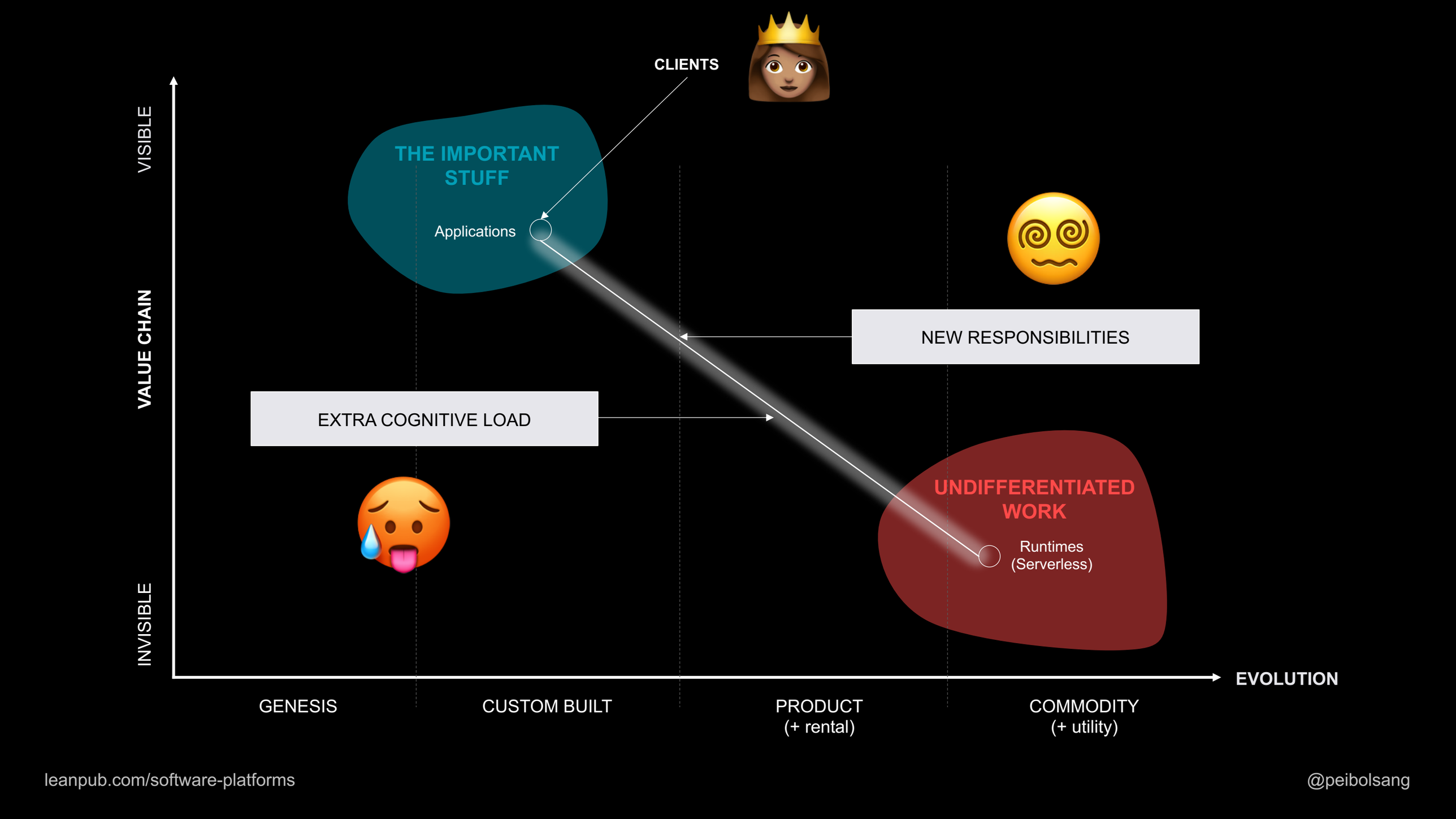
– Eric Elliot

No es oro todo lo que reluce

Los proveedores cloud nos gestionan la complejidad 

Pero la complejidad es como la energía: no se destruye, solo se transforma 🔮


CLIENTS
VISIBLE
VALUE CHAIN
INVISIBLE
THE IMPORTANT STUFF
Applications
VALUE IS CREATED HERE
THIS IS UNDIFFERENTIATED WORK
UNDIFFERENTIATED WORK
Runtimes (Serverless)
EVOLUTION
GENESIS
CUSTOM BUILT
PRODUCT (+ rental)
COMMODITY (+ utility)


CLIENTS
VISIBLE
VALUE CHAIN
INVISIBLE
THE IMPORTANT STUFF
Applications
NEW RESPONSIBILITIES
EXTRA COGNITIVE LOAD
UNDIFFERENTIATED WORK
Runtimes (Serverless)
EVOLUTION
GENESIS
CUSTOM BUILT
PRODUCT (+ rental)
COMMODITY (+ utility)

EXTRA COGNITIVE LOAD

How to develop in local

Select right cloud service

How to access logs

Learn new APIs and SDKs

How to debug after deployment

How to access logs

New integrations

NEW RESPONSIBILITIES

Write infrastructure-as-code scripts

Set-up CI/CD pipelines

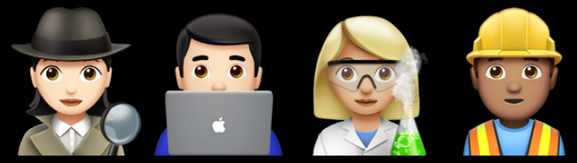
Design my own architecture

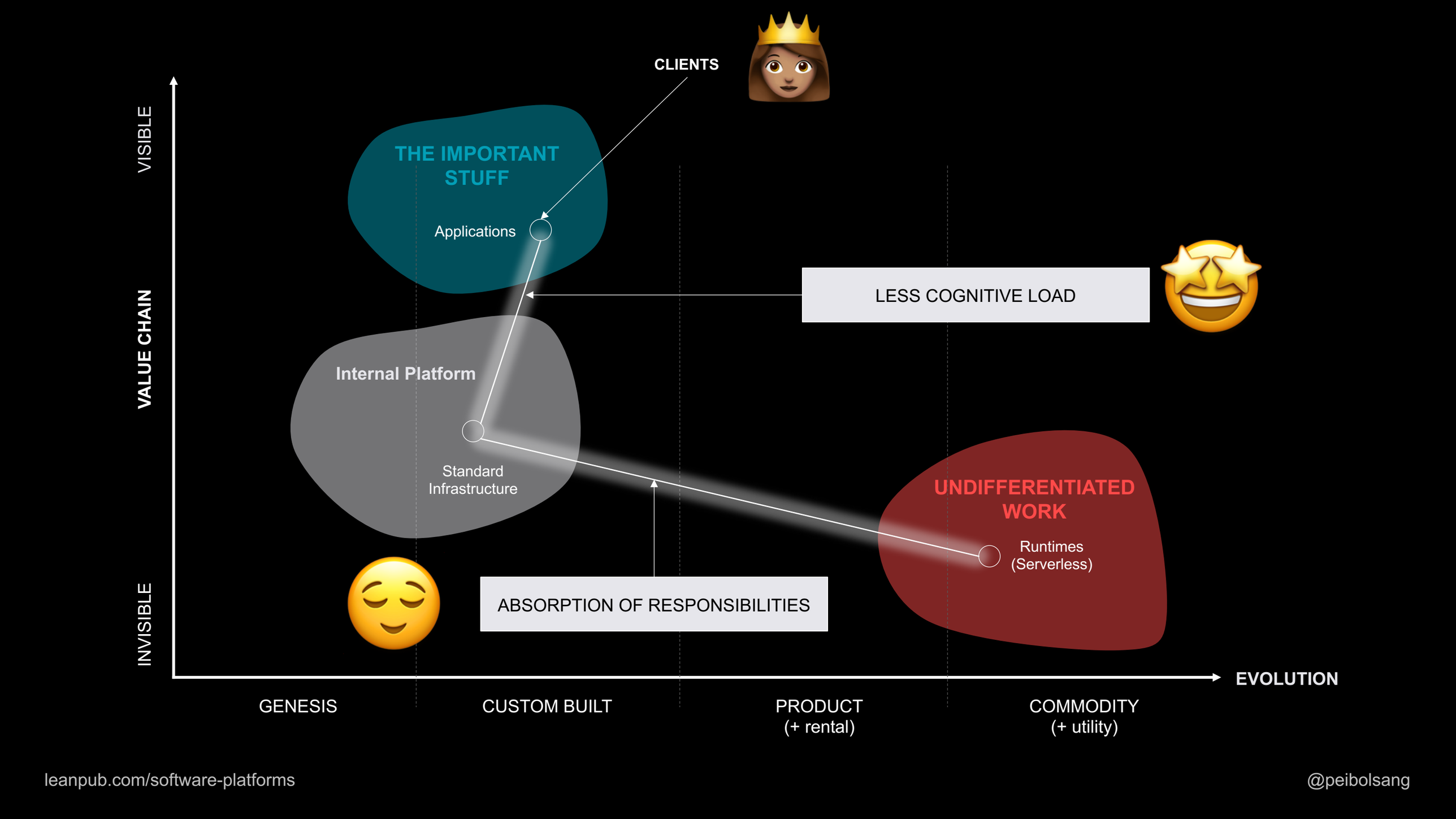
NoOps

New NFR: Costs (FinOps)

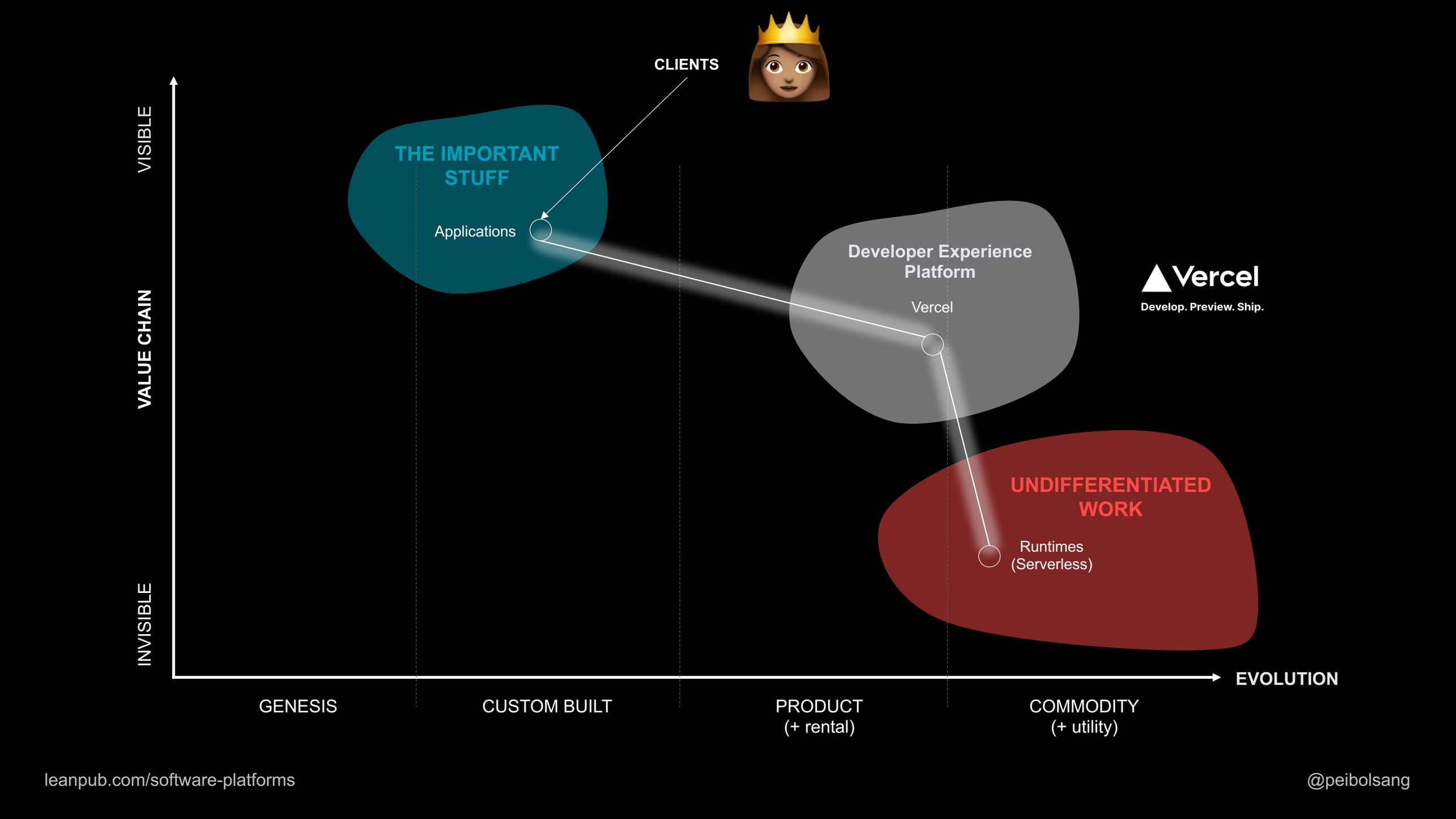
New NFR: Sustainability (GreenOps)

¿Cómo se puede solucionar esto?

Con plataformas internas de software que ayuden a los equipos de desarrollo a centrarse en resolver problemas de negocio 

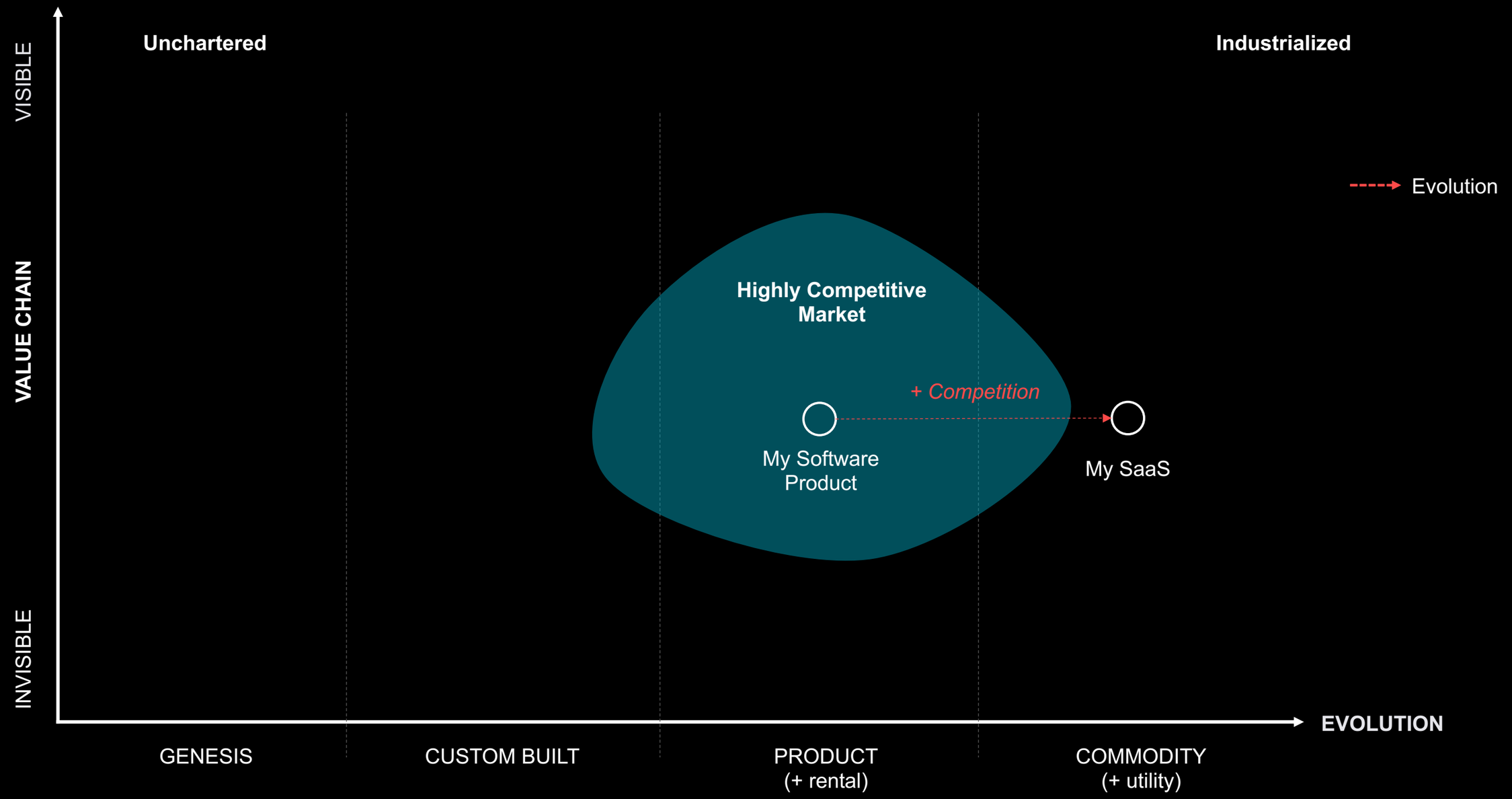

CLIENTS
VISIBLE
VALUE CHAIN
INVISIBLE
THE IMPORTANT STUFF
Applications
LESS COGNITIVE LOAD
Internal Platform
Standard Infrastructure
UNDIFFERENTIATED WORK
Runtimes (Serverless)
ABSORPTION OF RESPONSIBILITIES
EVOLUTION
GENESIS
CUSTOM BUILT
PRODUCT (+ rental)
COMMODITY (+ utility)


AWS CREATING SERVICES
PLATFORM TEAM
DEVELOPERS
imgflip.com


CLIENTS
VISIBLE
VALUE CHAIN
INVISIBLE
THE IMPORTANT STUFF
Applications
Developer Experience Platform
Vercel
Vercel
Develop. Preview. Ship.
UNDIFFERENTIATED WORK
Runtimes (Serverless)
EVOLUTION
GENESIS
CUSTOM BUILT
PRODUCT (+ rental)
COMMODITY (+ utility)

Parte 2 

Arquitectura Técnica de Productos Software

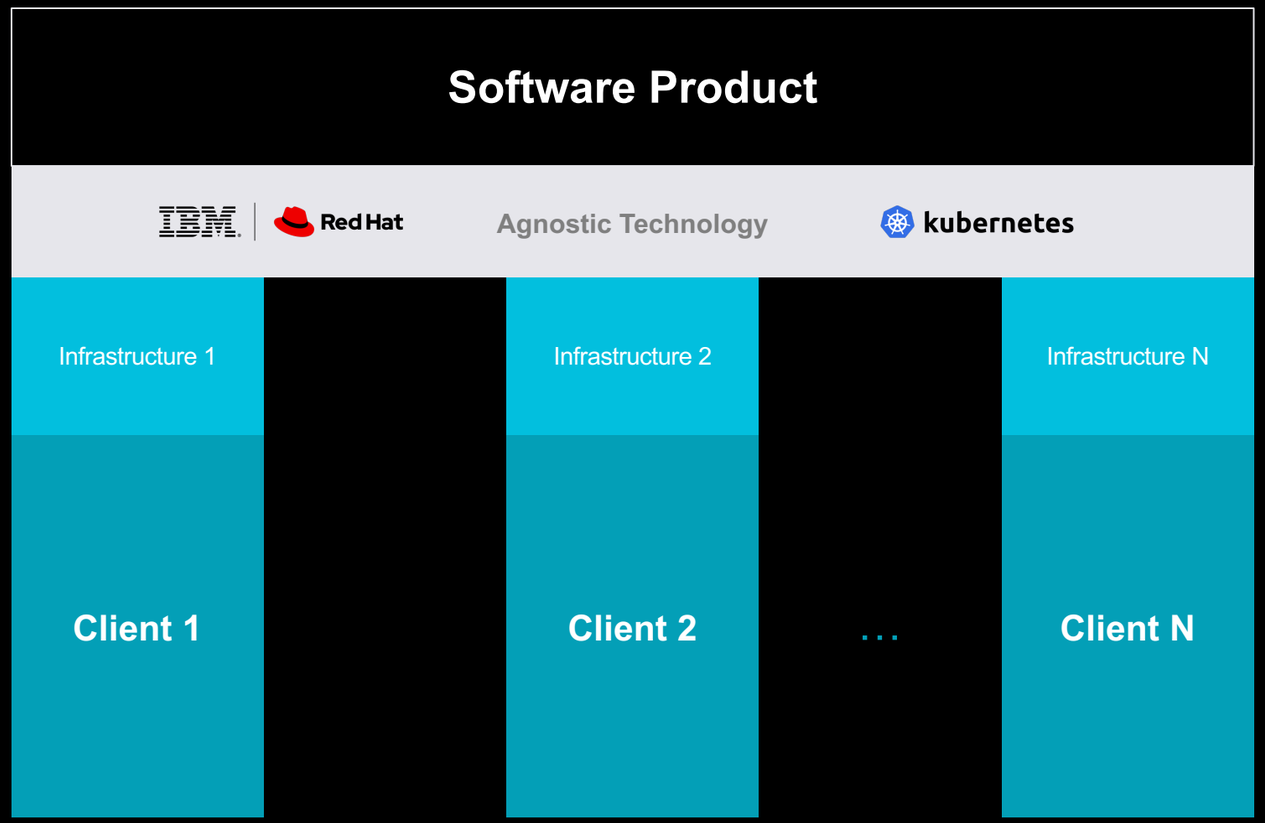

Unchartered
Industrialized
VISIBLE
VALUE CHAIN
INVISIBLE
Evolution
Highly Competitive Market
+ Competition
My Software Product
My SaaS
EVOLUTION
GENESIS
CUSTOM BUILT
PRODUCT (+ rental)
COMMODITY (+ utility)

Pero,
¿por qué es importante movernos hacia servicio 🌍 ?

No hay nada mágico en el código fuente de tu producto que lo haga irresistible a tus clientes 

El servicio es lo que importa 

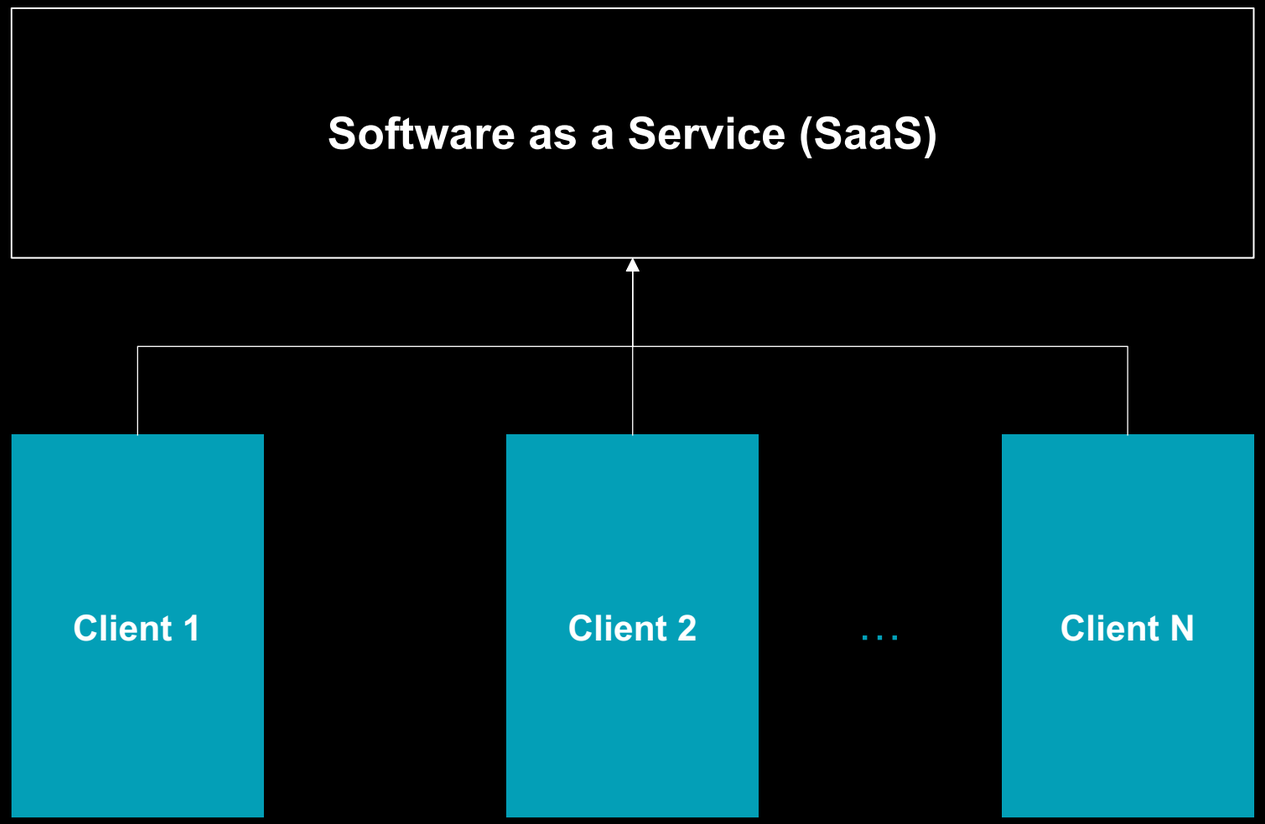
Product-based Economy



The principal need is deploying the software product on multiple client environments simultaneously

The challenge is doing it at scale without building different versions of the product for different clients

Service-based Economy

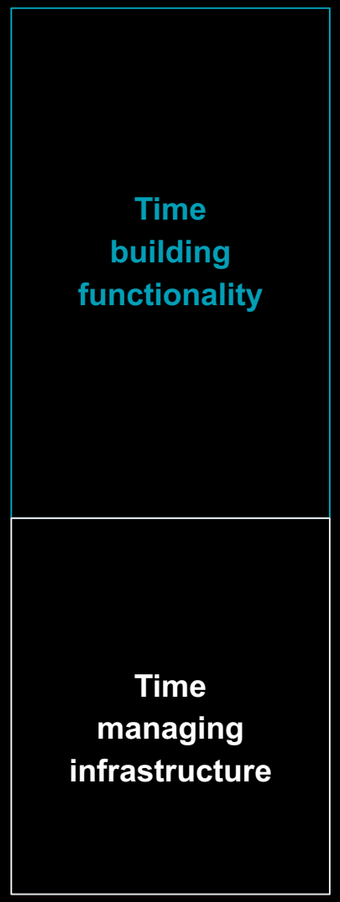


The principal need is delivering software as a service over the Internet. There are no portability requirements

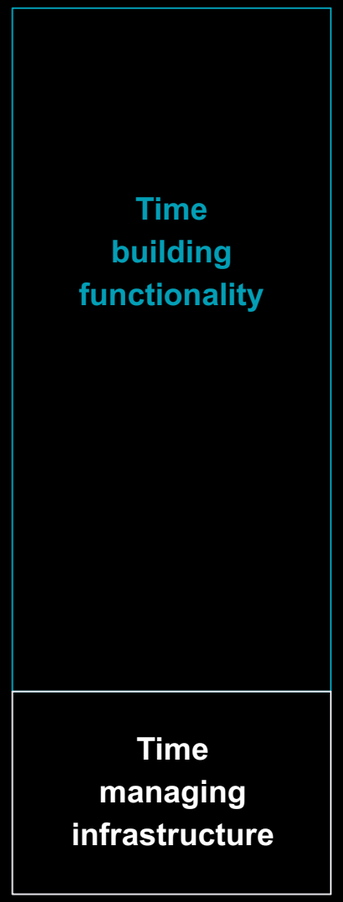
The challenge is doing it at the **minimum cost** possible, since the software provider owns the costs in their totality

What technology can I use to build my SaaS?

Using agnostic technology



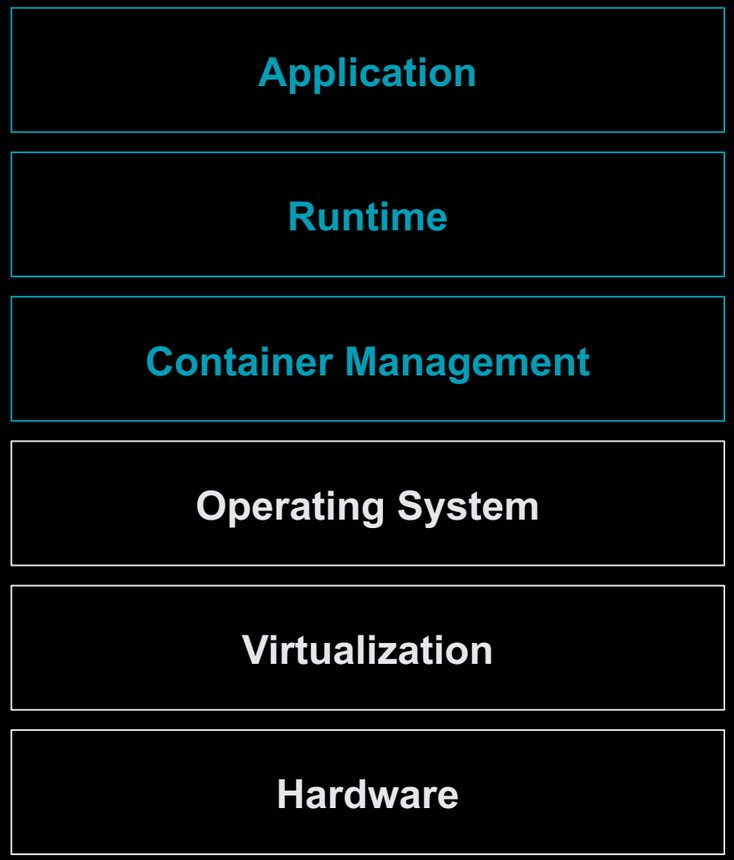
Using managed cloud services



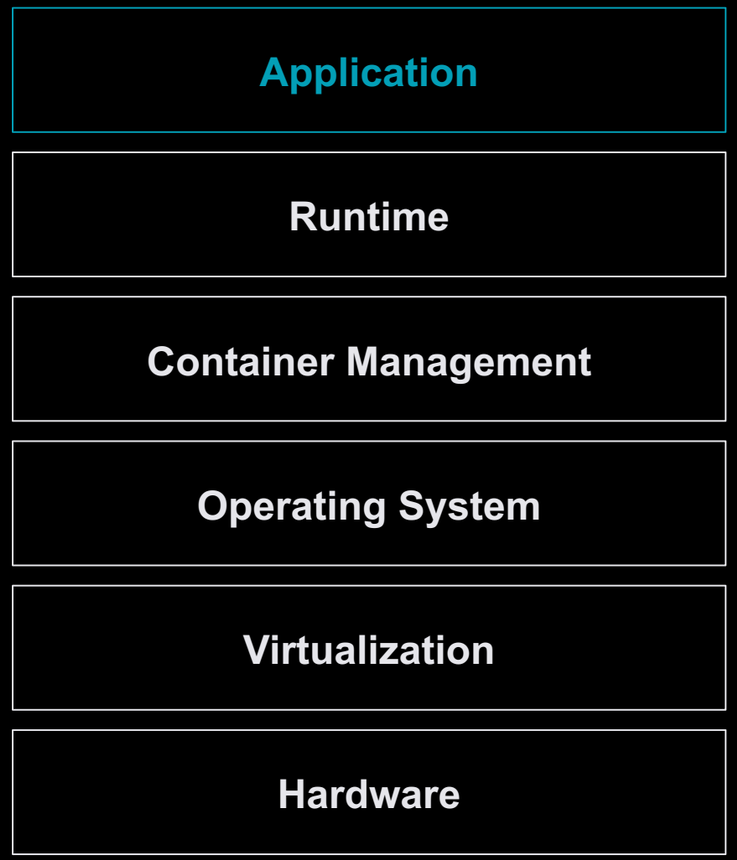
We must minimize undifferentiated work (cost of doing business) and maximize functionality (delivered value)

"In the future, all the code you ever write is business logic" – Werner Vogels, CTO Amazon.com

Using agnostic technology

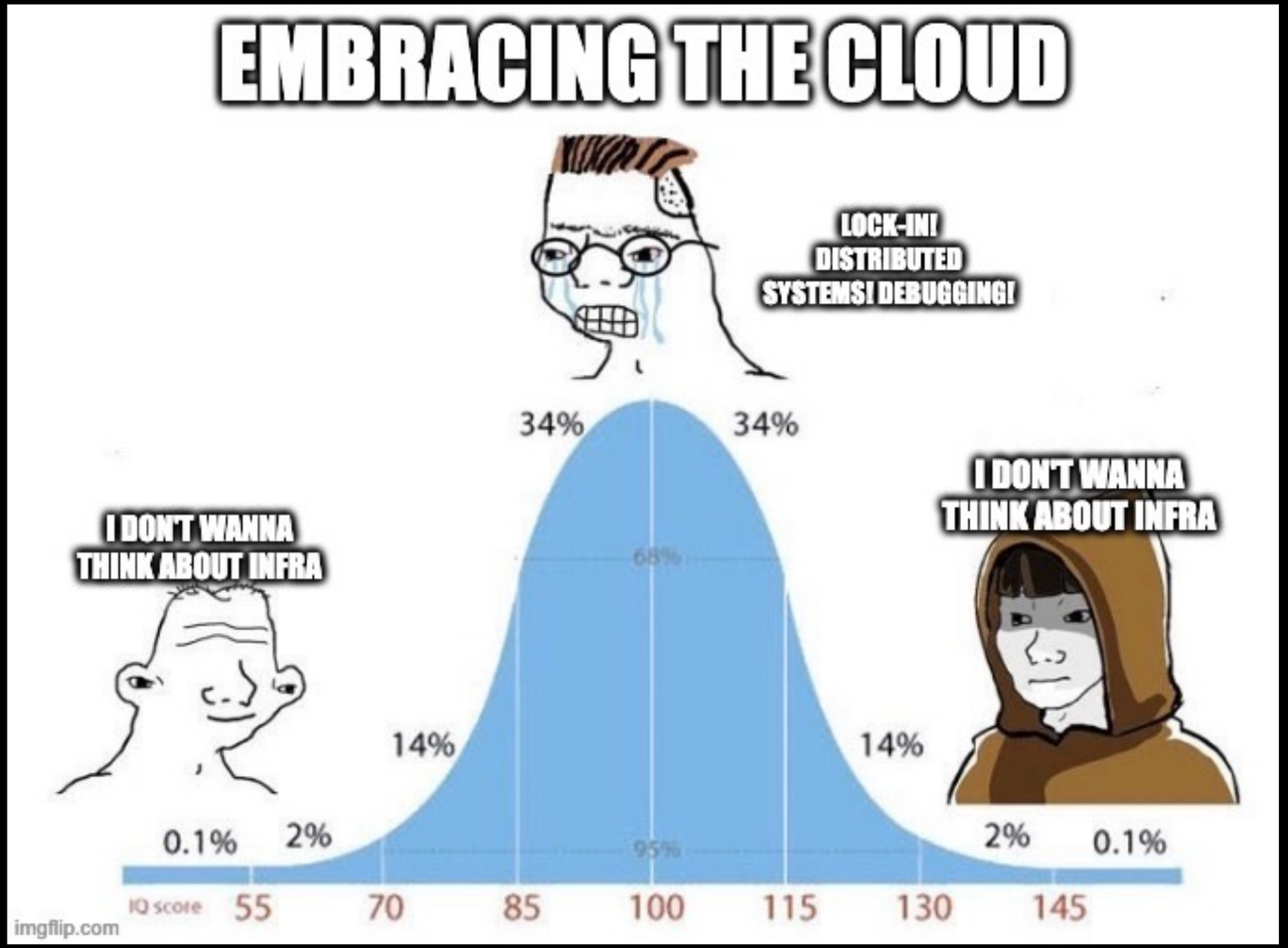


Using managed services



Managed by user (you)

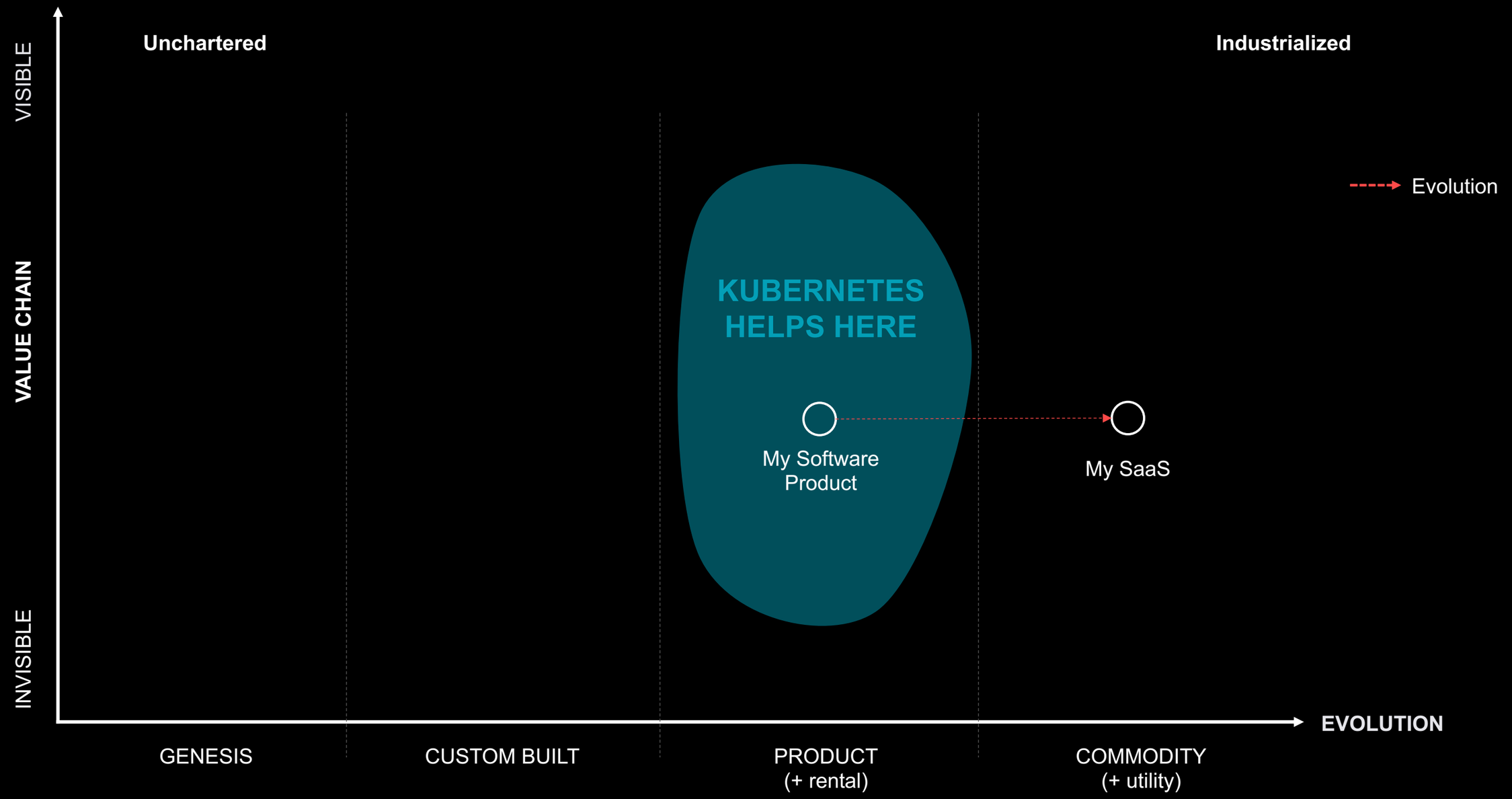
Managed by cloud provider

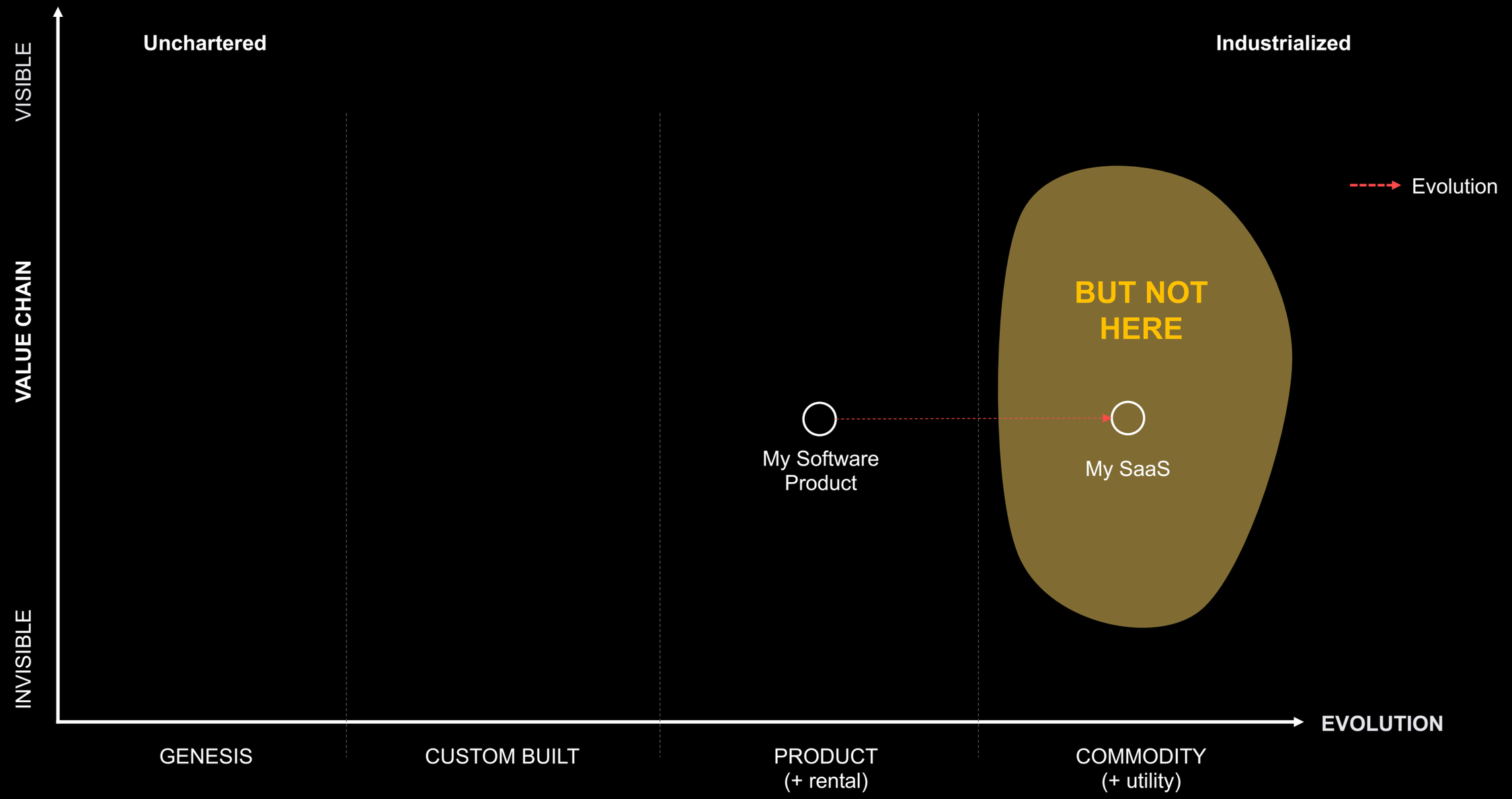

EMBRACING THE CLOUD
LOCK-IN! DISTRIBUTED SYSTEMS! DEBUGGING!
I DON'T WANNA THINK ABOUT INFRA
I DON'T WANNA THINK ABOUT INFRA
34%
34%
14%
14%
0.1%
2%
2%
0.1%
68%
95%
IQ score
55
70
85
100
115
130
145
imgflip.com

Y ahora, me voy a meter en un jardín.

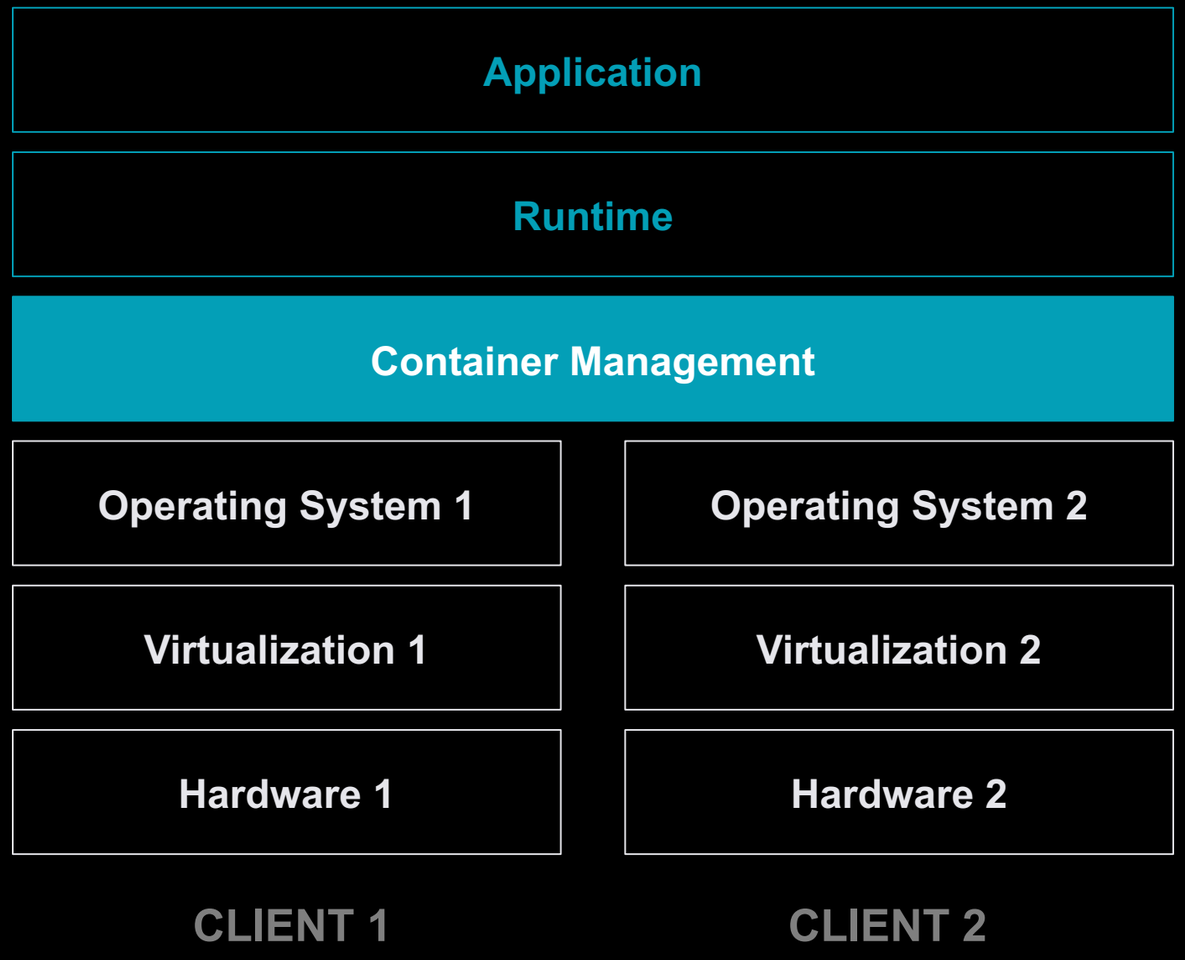
El problema de Kubernetes es que está sobre-utilizado 🔥

Soluciona un problema importante, pero no es el futuro 🚀

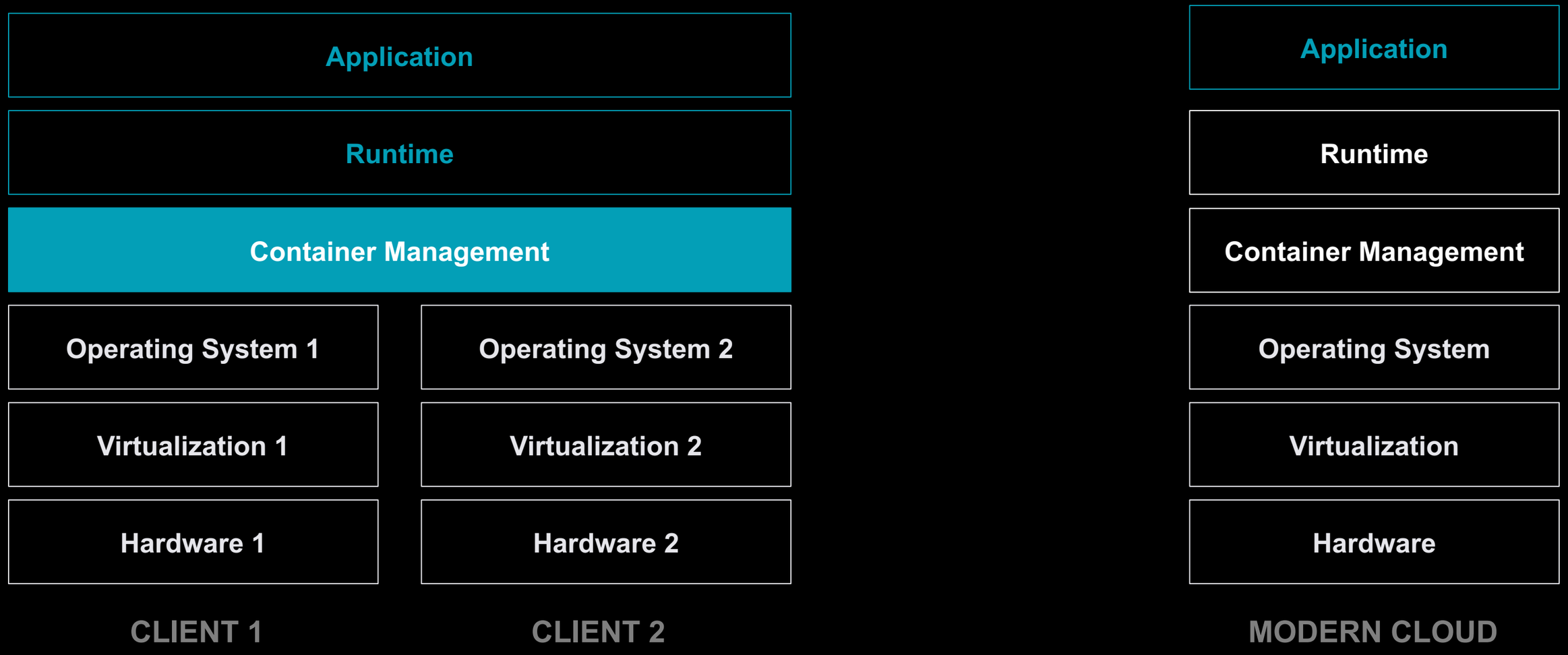

Unchartered
Industrialized
VISIBLE
VALUE CHAIN
INVISIBLE
Evolution
KUBERNETES
HELPS HERE
My Software
Product
My SaaS
EVOLUTION
GENESIS
CUSTOM BUILT
PRODUCT
(+ rental)
COMMODITY
(+ utility)


Unchartered
Industrialized
VISIBLE
VALUE CHAIN
INVISIBLE
Evolution
BUT NOT HERE
My Software Product
My SaaS
EVOLUTION
GENESIS
CUSTOM BUILT
PRODUCT (+ rental)
COMMODITY (+ utility)

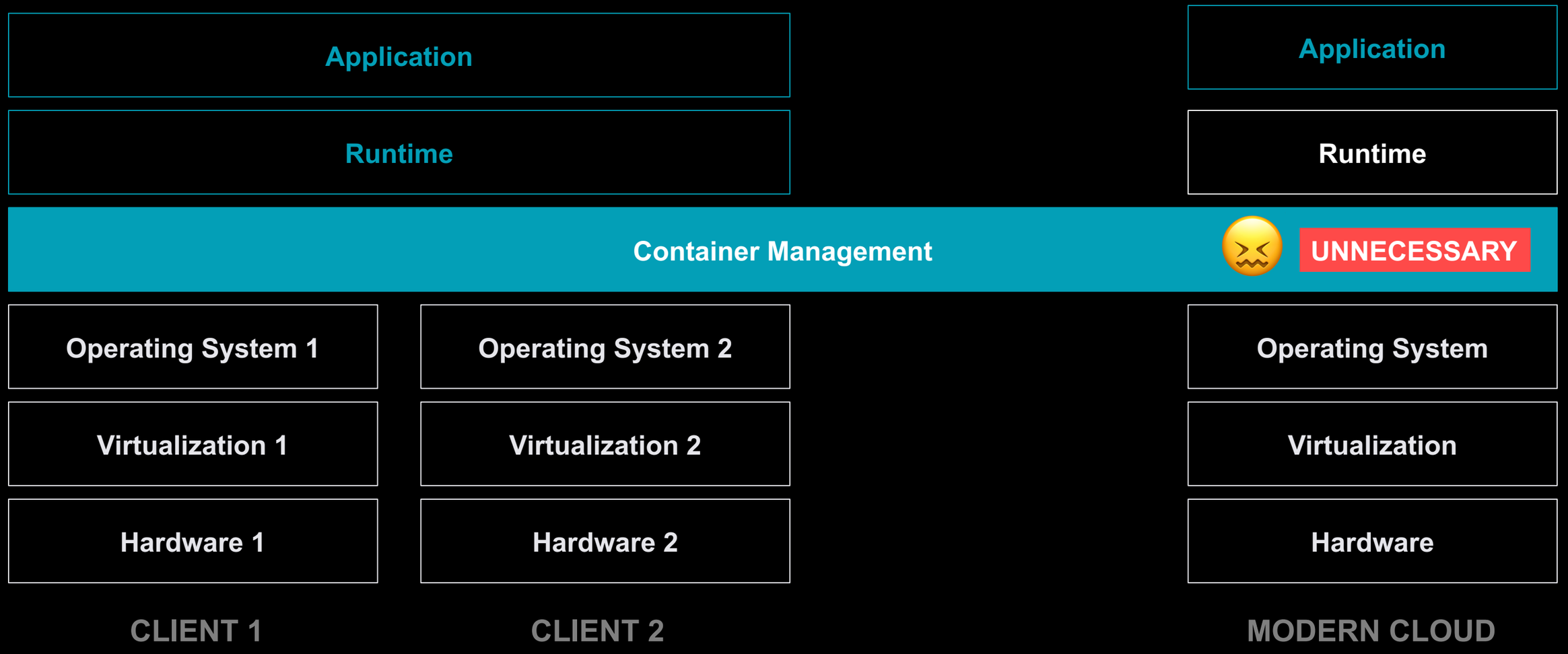
How Kubernetes is helping us



How Kubernetes is helping us



How Kubernetes is NOT helping us



Corey Quinn
@QuinnyPig

Me: Kubernetes is the solution to a problem that isn't particularly sympathetic. Specifically "I want to work at @awscloud but your hiring bar is too high so I'm going to run k8s so I can cosplay as a cloud provider myself at home."

Thank you!