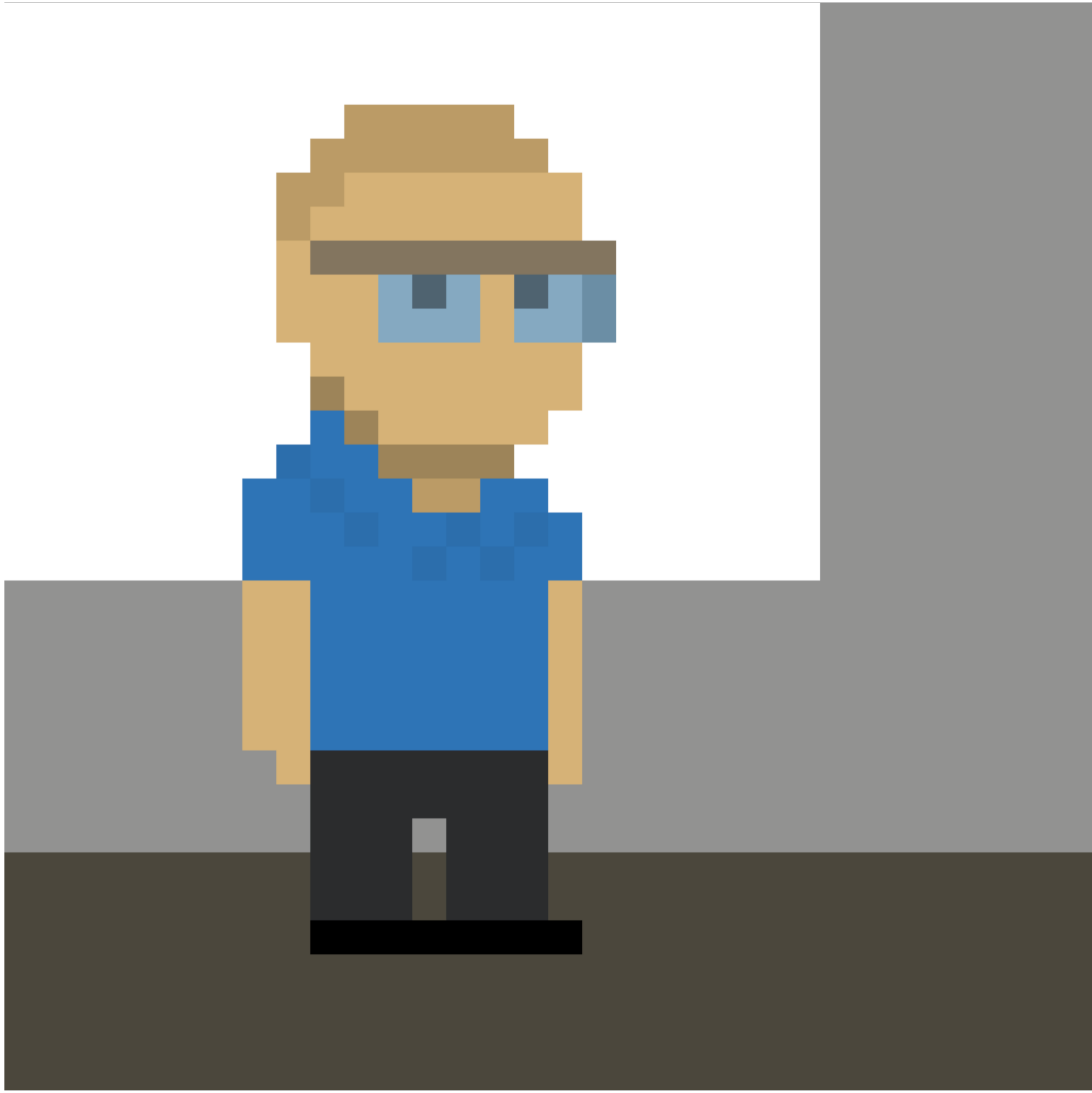
How serverless impacts design

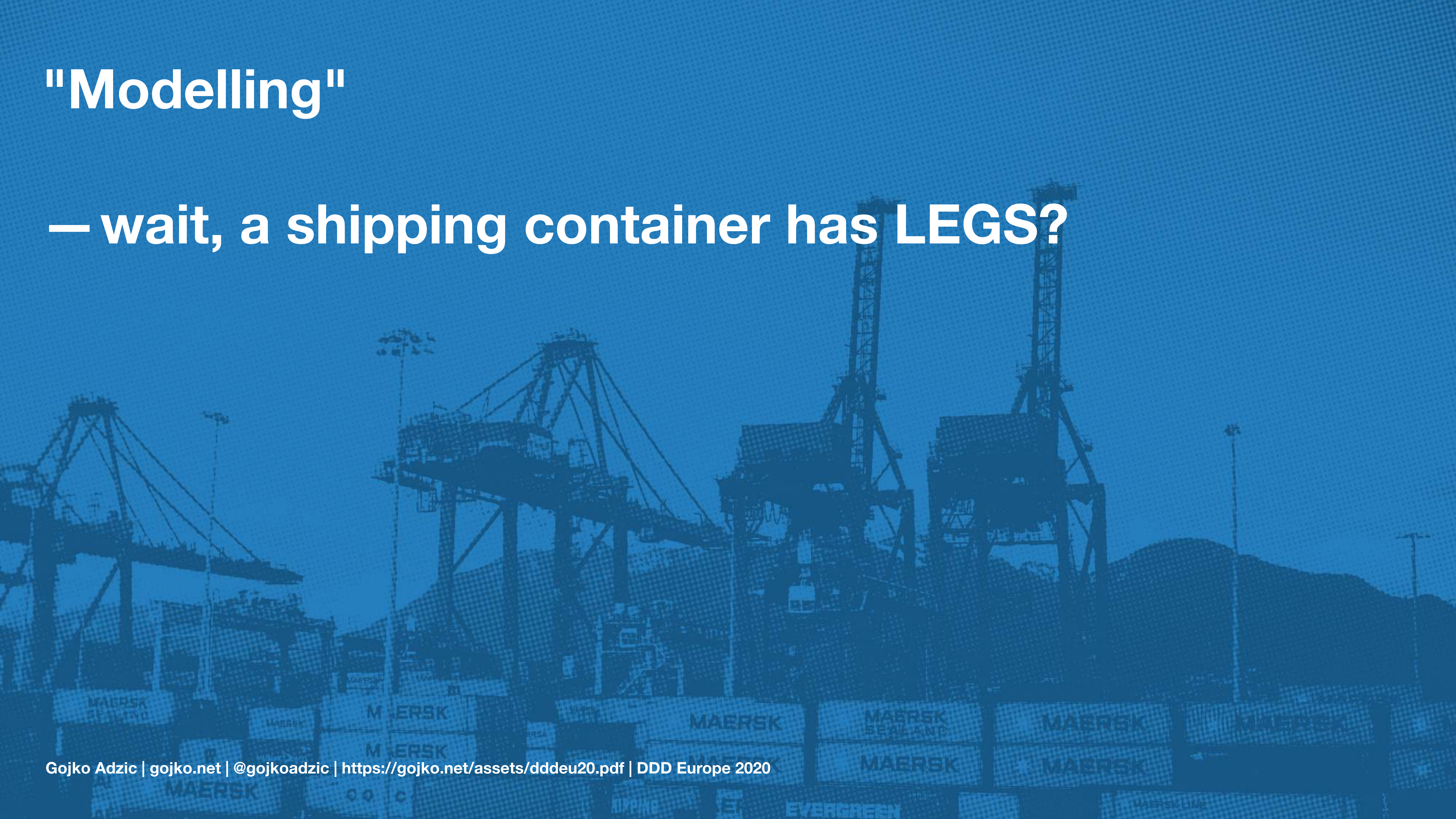
https://gojko.net/assets/dddeu20.pdf

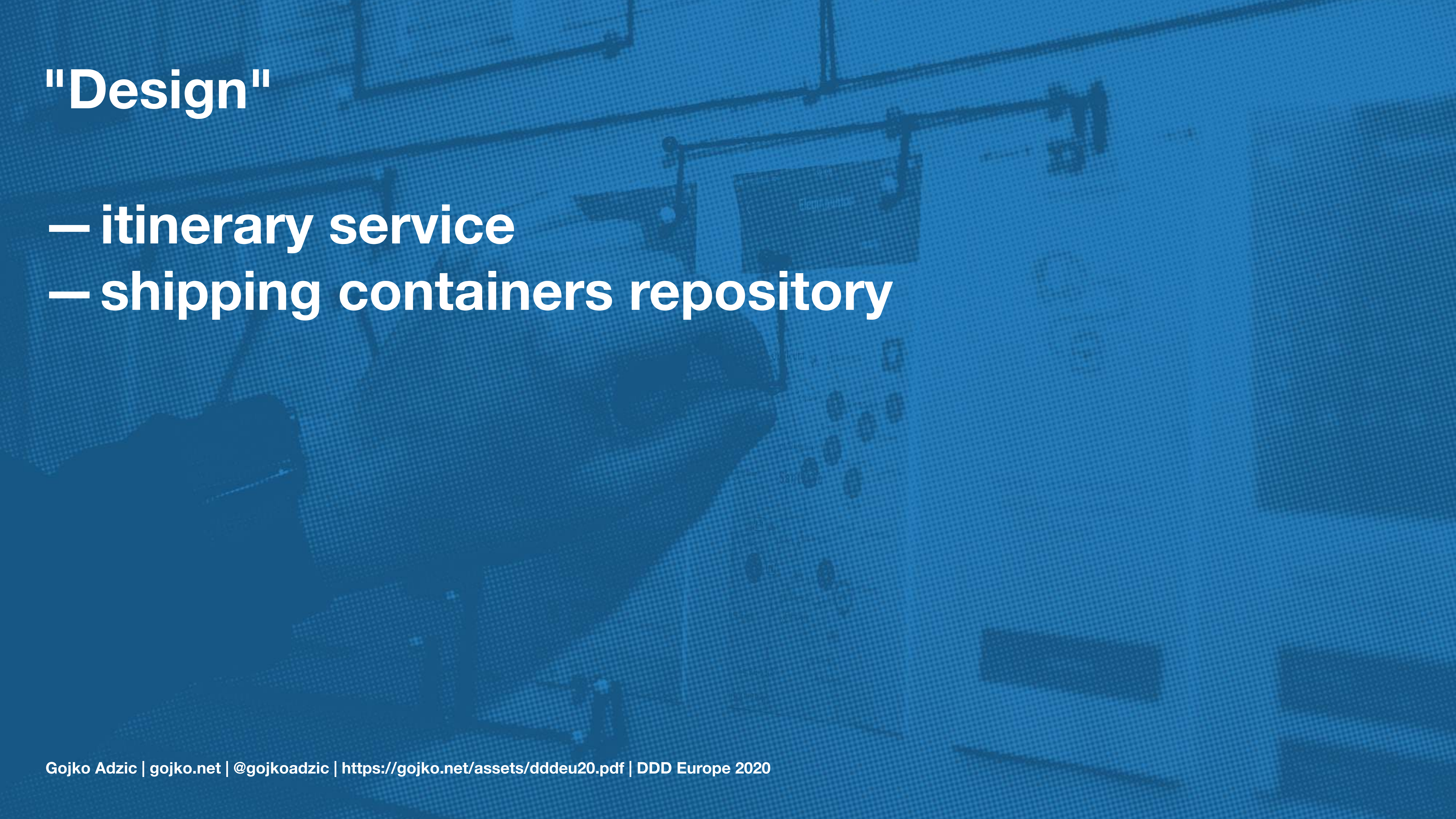


Gojko Adzic | gojko.net | @gojkoadzic | gojko@gojko.com

"Modelling"

— wait, a shipping container has LEGS?



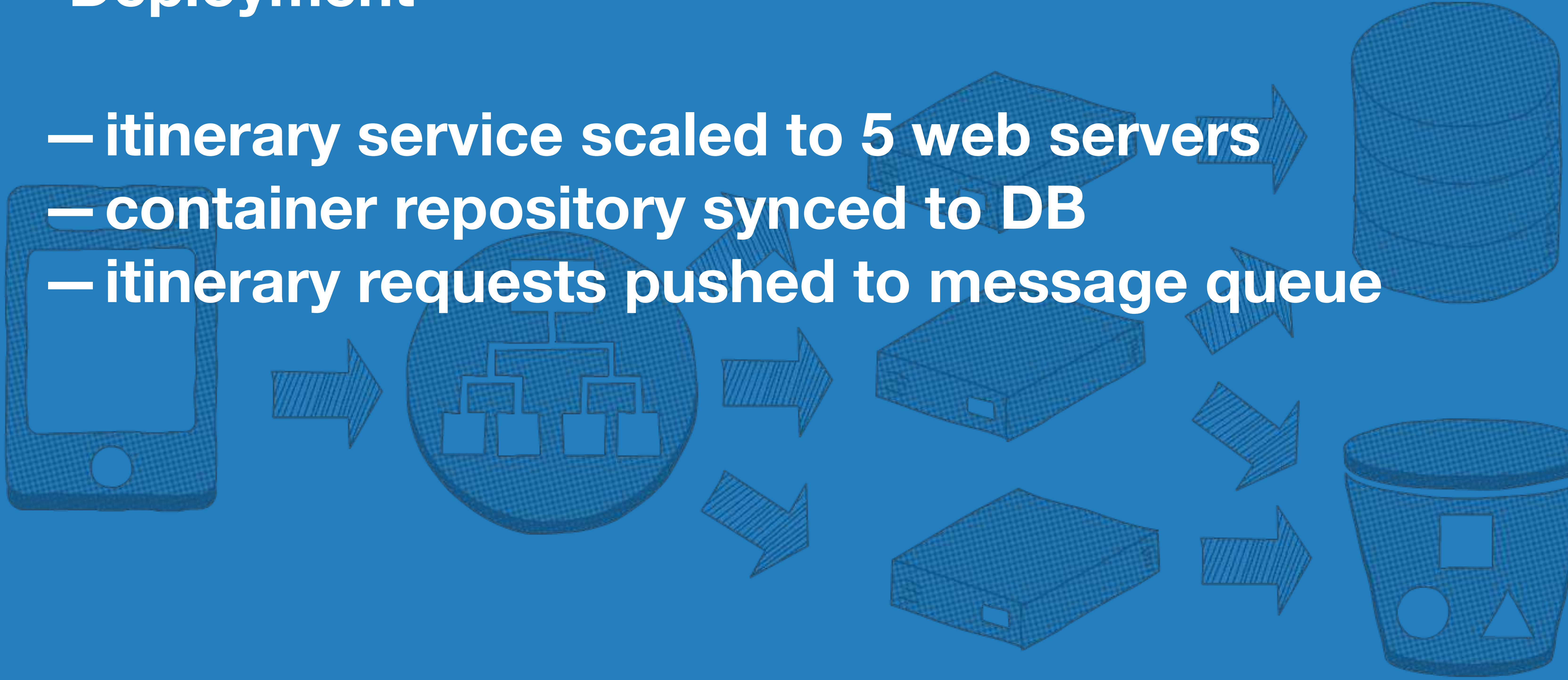


"Design"

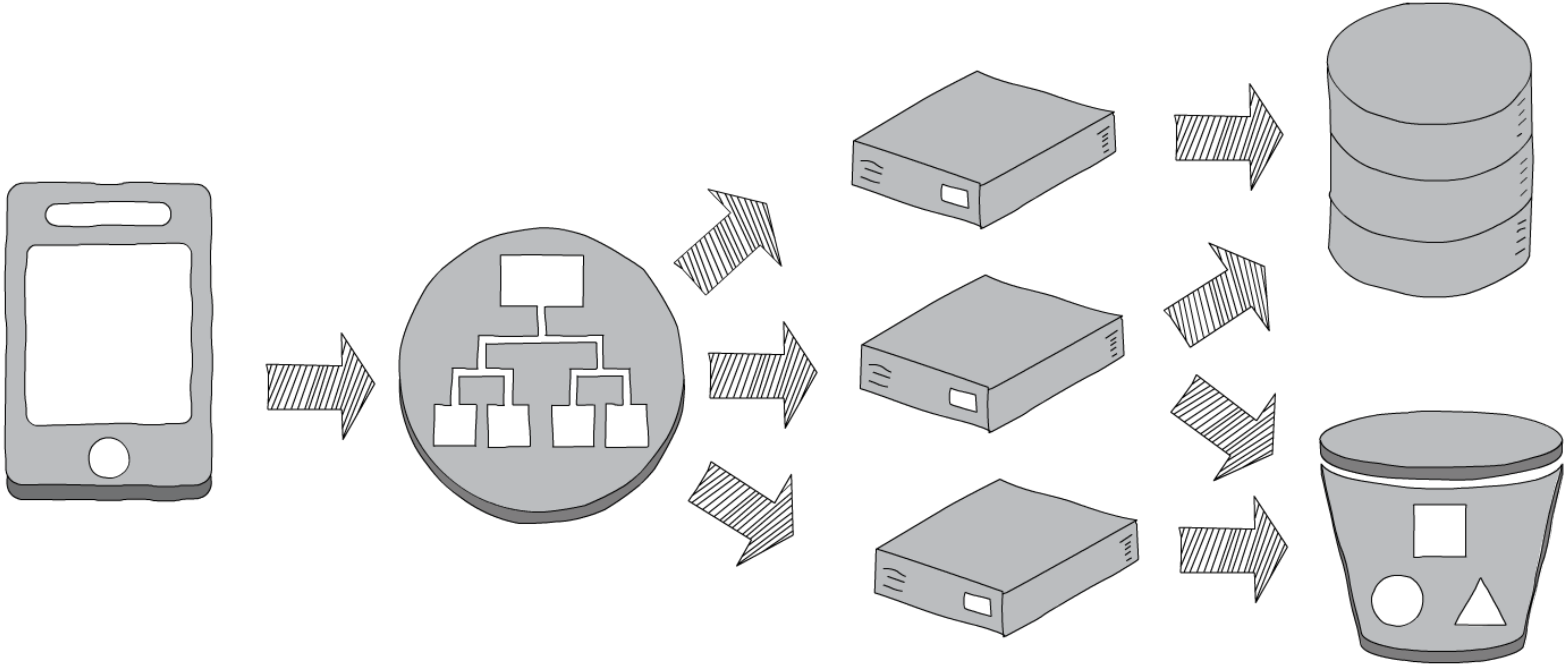
— itinerary service
— shipping containers repository

"Deployment"

- itinerary service scaled to 5 web servers
- container repository synced to DB
- itinerary requests pushed to message queue



"Server"



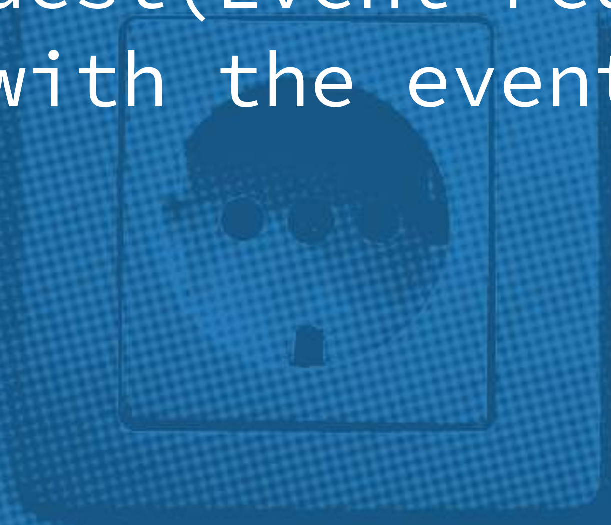
NGINX

APACHE

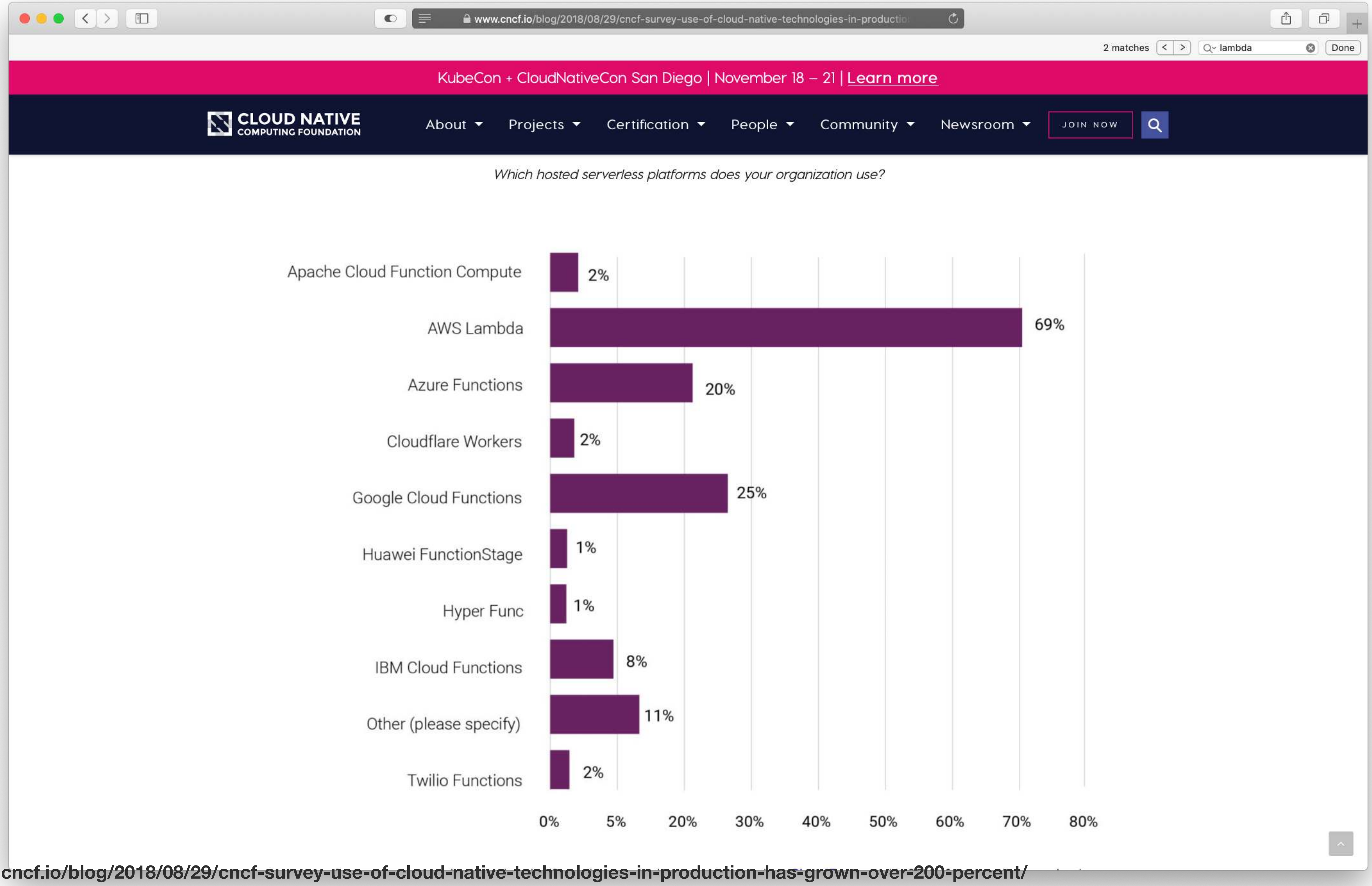
ACTIVE MQ

JBoss®
by Red Hat

~~Serverless~~ Socketless

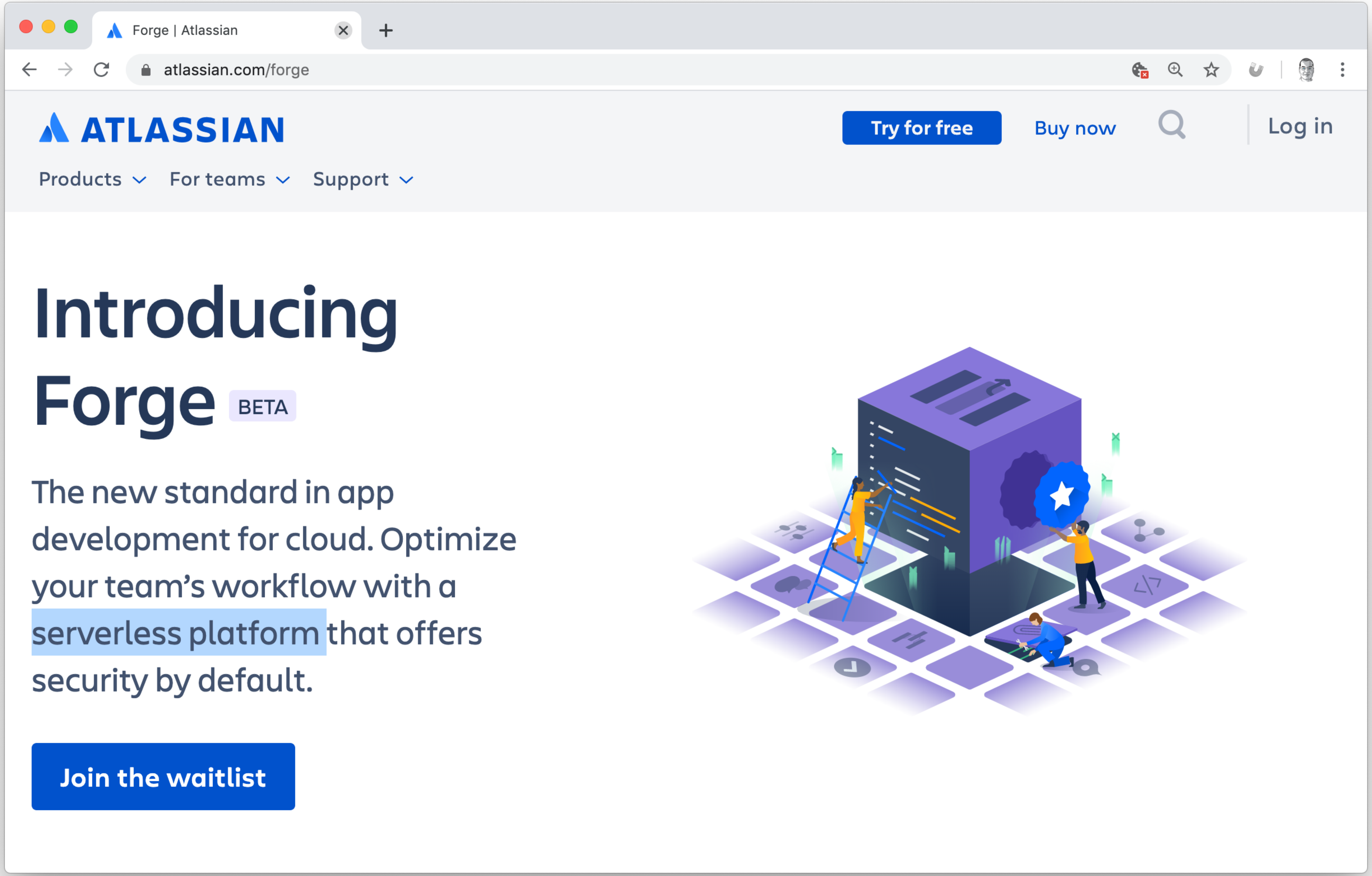


```
public class LambdaMain implements RequestHandler<Event, Response> {
  public Response handleRequest(Event request, Context context){
    // do something useful with the event
  };
};
```


www.cncf.io/blog/2018/08/29/cncf-survey-use-of-cloud-native-technologies-in-productio
2 matches
lambda
Done
KubeCon + CloudNativeCon San Diego | November 18 – 21 | Learn more
CLOUD NATIVE
COMPUTING FOUNDATION
About
Projects
Certification
People
Community
Newsroom
JOIN NOW

Which hosted serverless platforms does your organization use?

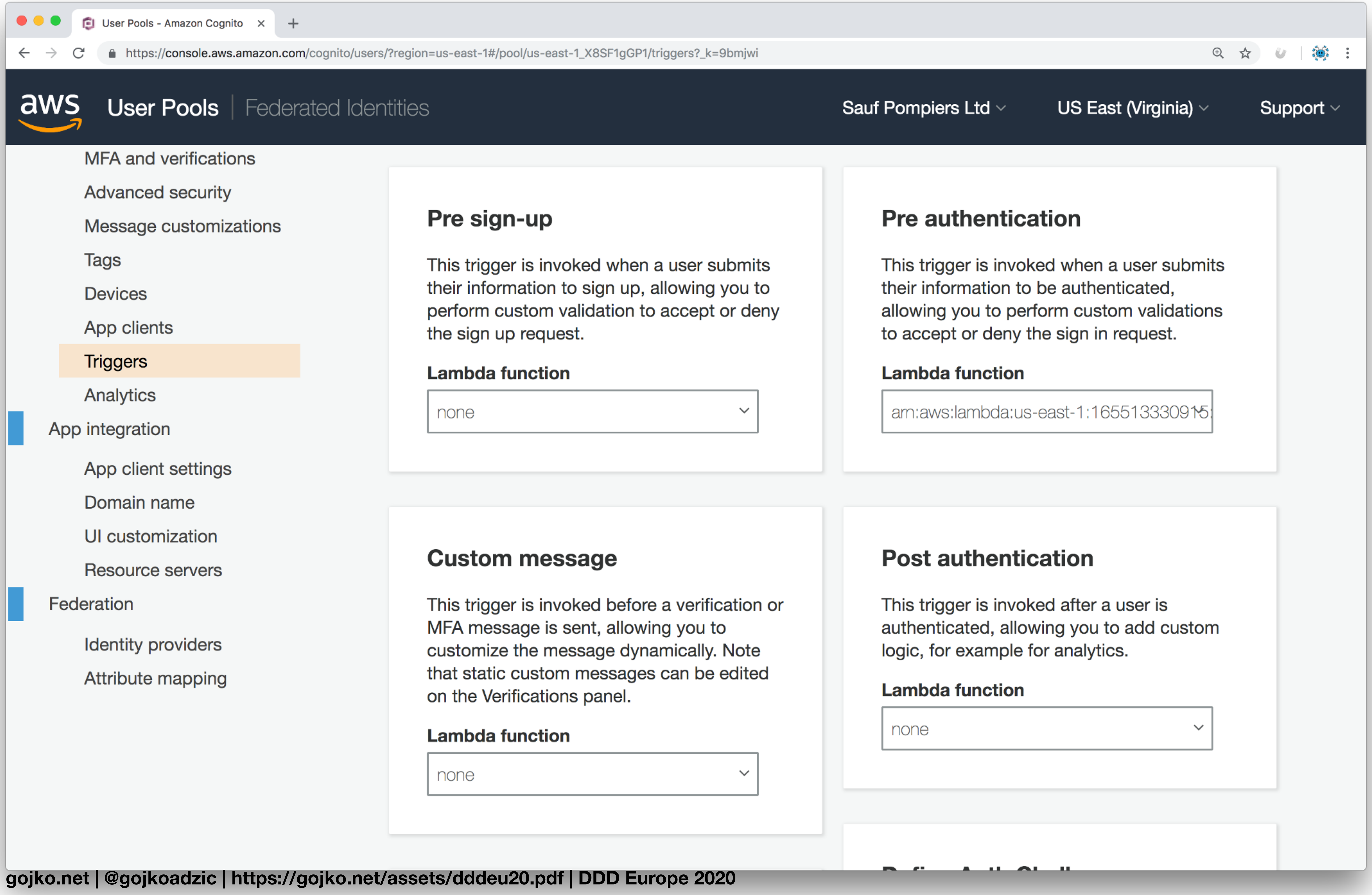
https://www.cncf.io/blog/2018/08/29/cncf-survey-use-of-cloud-native-technologies-in-production-has-grown-over-200-percent/



Introducing Forge BETA

The new standard in app development for cloud. Optimize your team’s workflow with a serverless platform that offers security by default.

https://atlassian.com/forge



Pre sign-up

This trigger is invoked when a user submits their information to sign up, allowing you to perform custom validation to accept or deny the sign up request.

Lambda function: none

Pre authentication

This trigger is invoked when a user submits their information to be authenticated, allowing you to perform custom validations to accept or deny the sign in request.

Lambda function: arn:aws:lambda:us-east-1:16551333091 5:

Custom message

This trigger is invoked before a verification or MFA message is sent, allowing you to customize the message dynamically. Note that static custom messages can be edited on the Verifications panel.

Lambda function: none

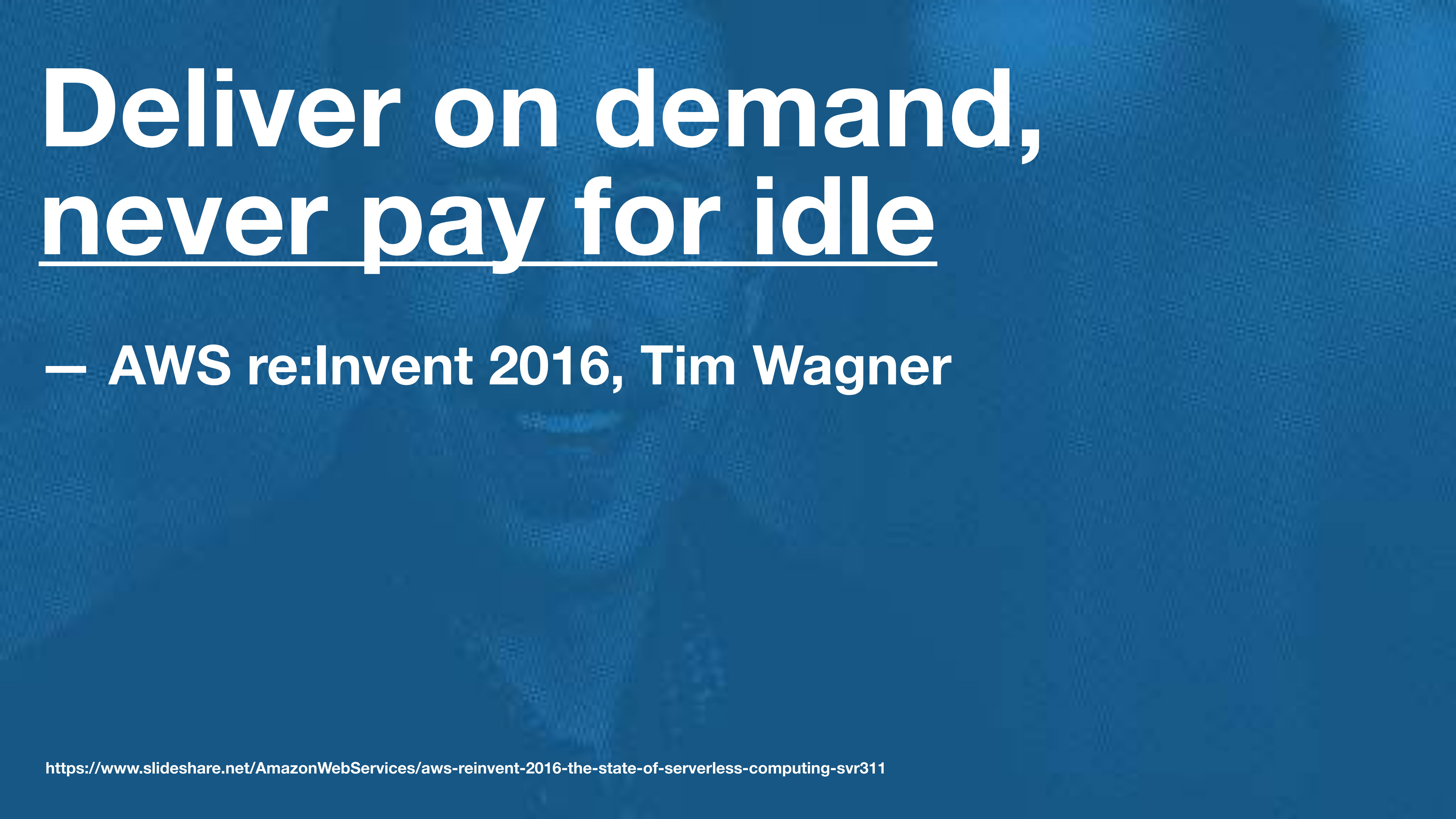
Post authentication

This trigger is invoked after a user is authenticated, allowing you to add custom logic, for example for analytics.

Lambda function: none

~~Serverless~~ Distractionless

- Generic: hire from the cloud provider
- Supporting: customise provider services
- Core: more time left for this

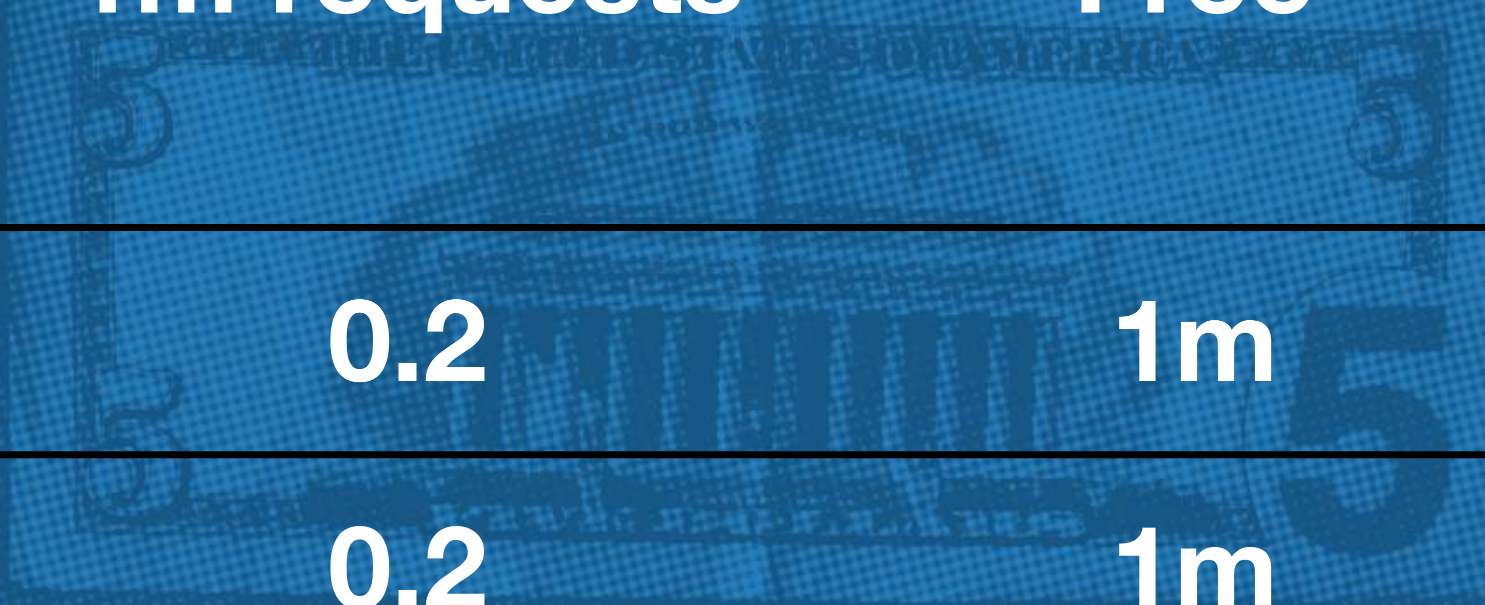


Deliver on demand, never pay for idle

— AWS re:Invent 2016, Tim Wagner

https://www.slideshare.net/AmazonWebServices/aws-reinvent-2016-the-state-of-serverless-computing-svr311

~~Serverless~~ Reservationless



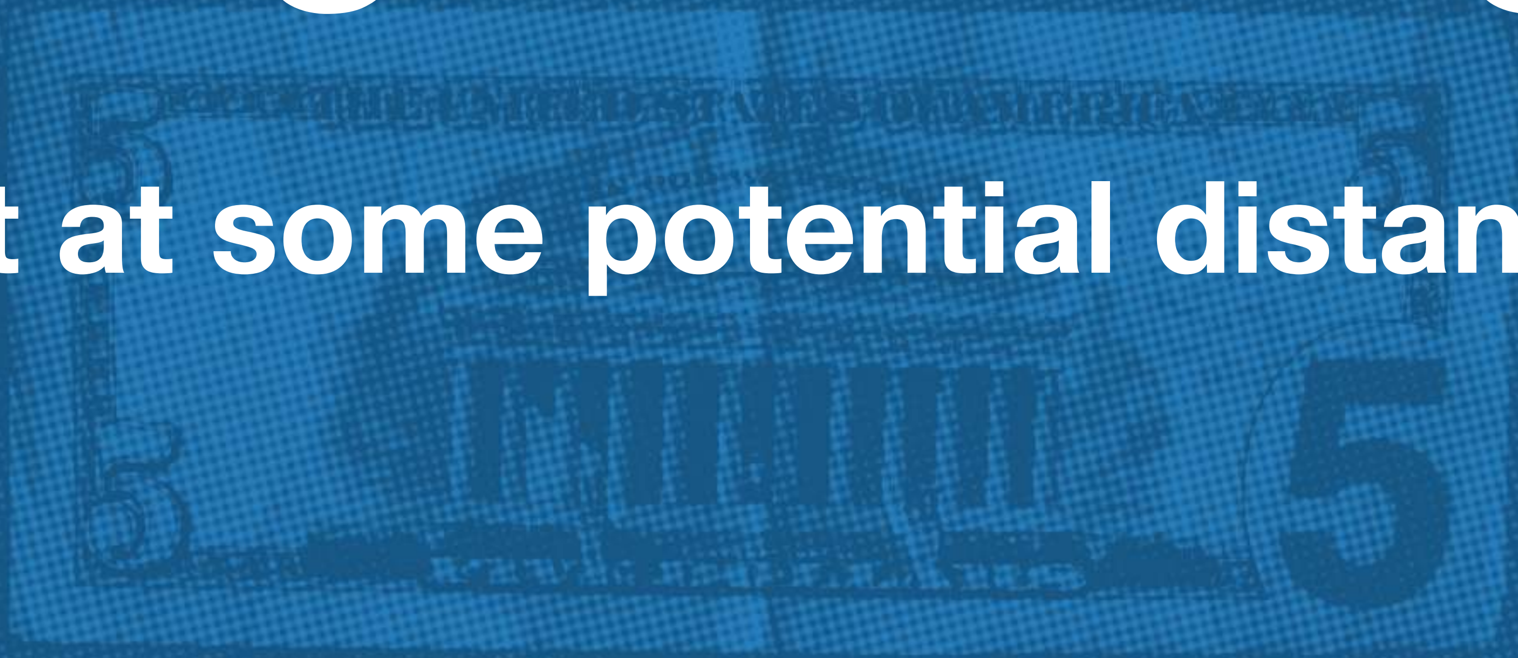
provider	1m requests	Free	CPU Time 512MB,100ms
AWS	0.2	1m	.000000834
Azure	0.2	1m	.0000008
GCP	0.4	2m	.000000925

Lambda US-east-1; Azure, central US; `https://aws.amazon.com/lambda/pricing/`; `https://cloud.google.com/functions/pricing`; `https://azure.microsoft.com/en-us/pricing/details/functions/`

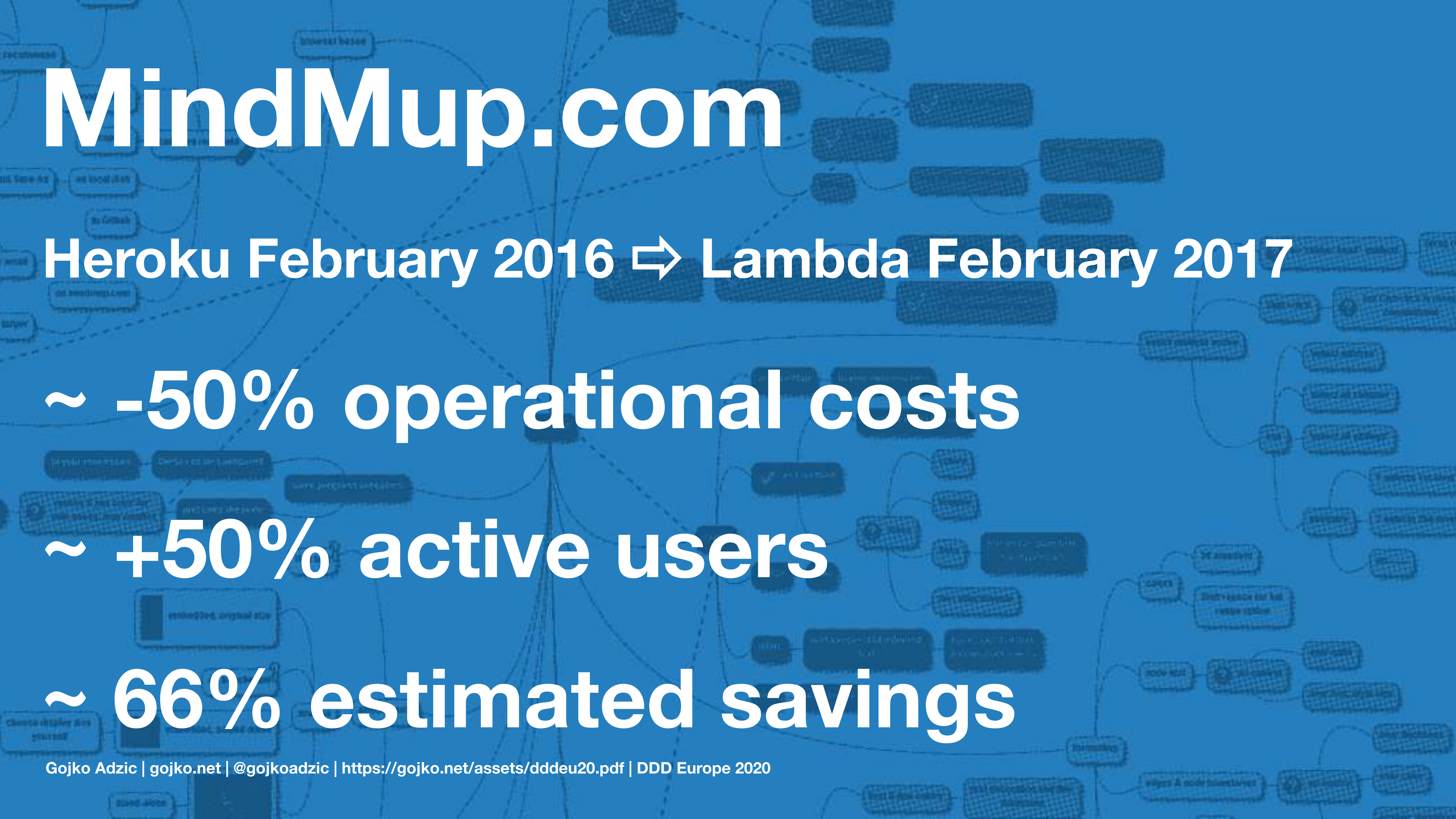
Paying for utilisation

- — not capacity
- — not environments
- — not instances

Serverless financially rewards good design



(instantly, not at some potential distant future)



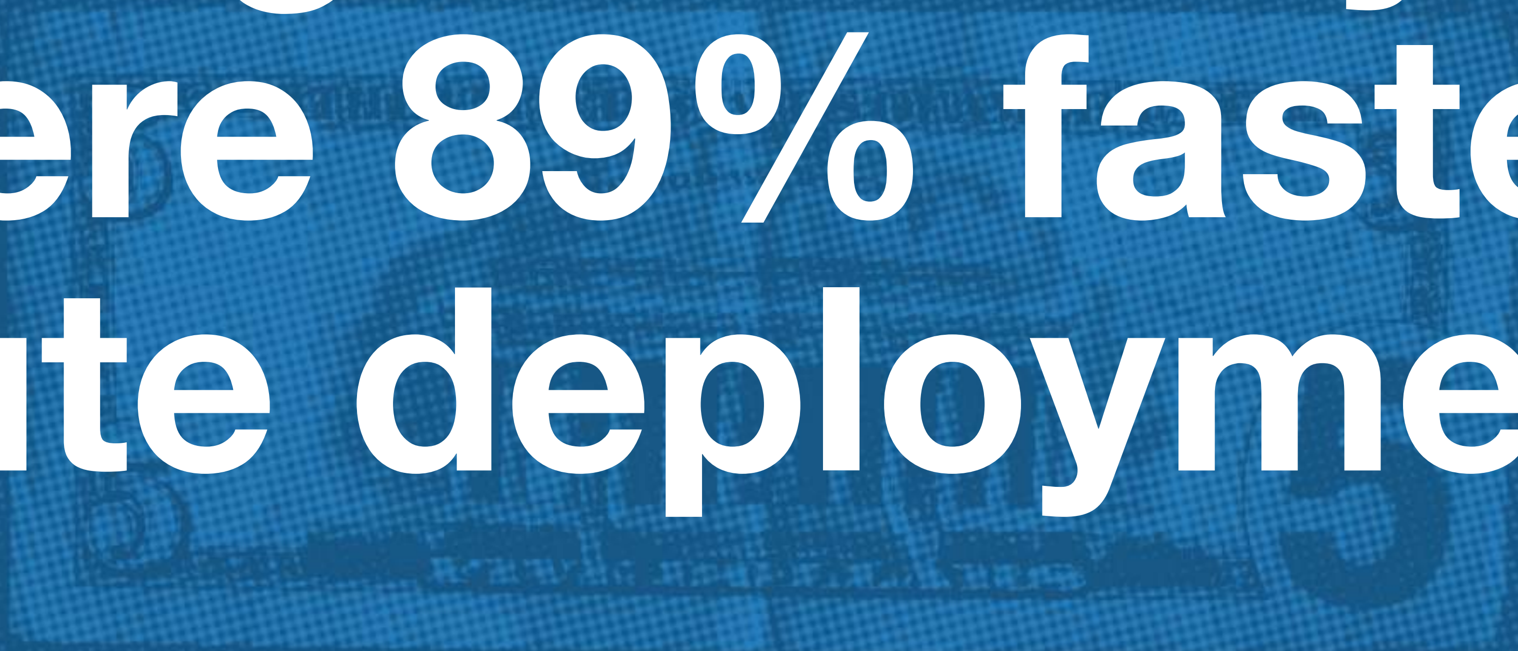
MindMup.com

Heroku February 2016 ⇨ Lambda February 2017

~ -50% operational costs

~ +50% active users

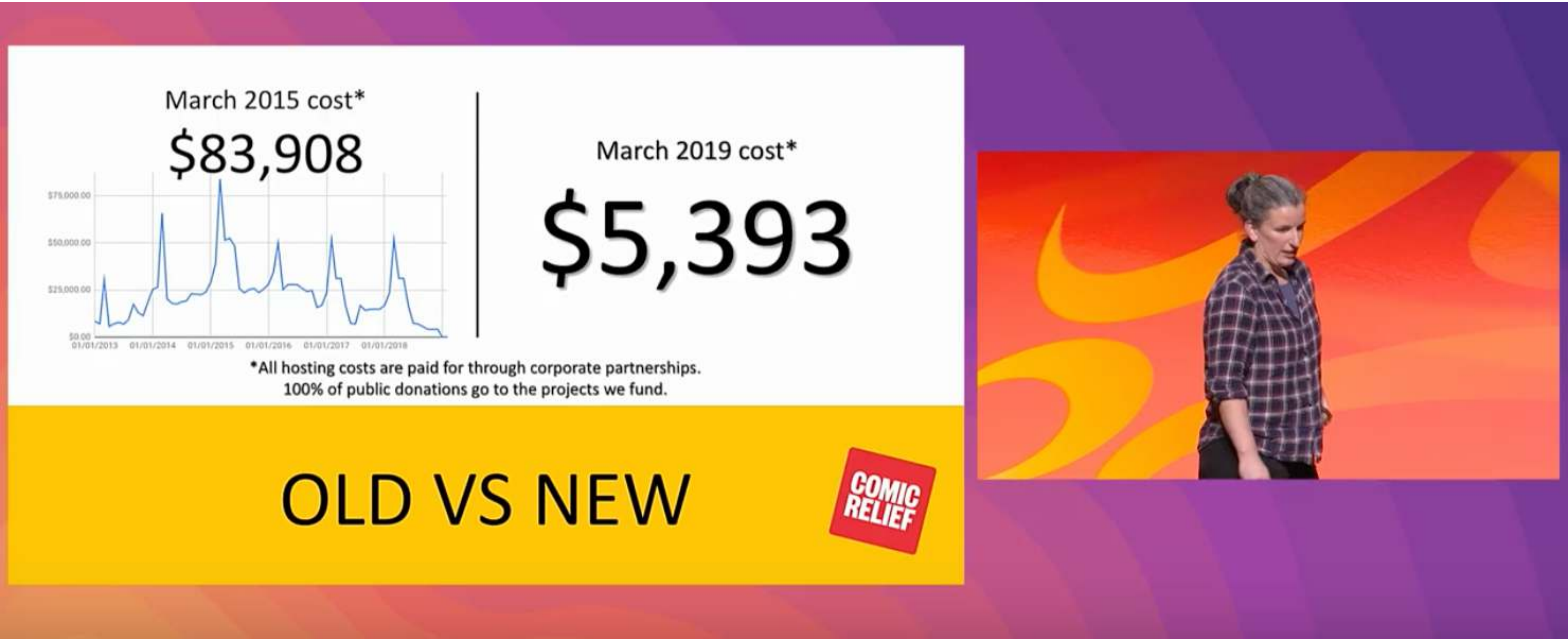
~ 66% estimated savings



"lowered five-year operating costs by 60% and were 89% faster at compute deployment"

— IDC white paper on AWS Serverless

https://pages.awscloud.com/Gated_IDC_Generating_Value_Through_IT_Agility.html



March 2015 cost*

$83,908

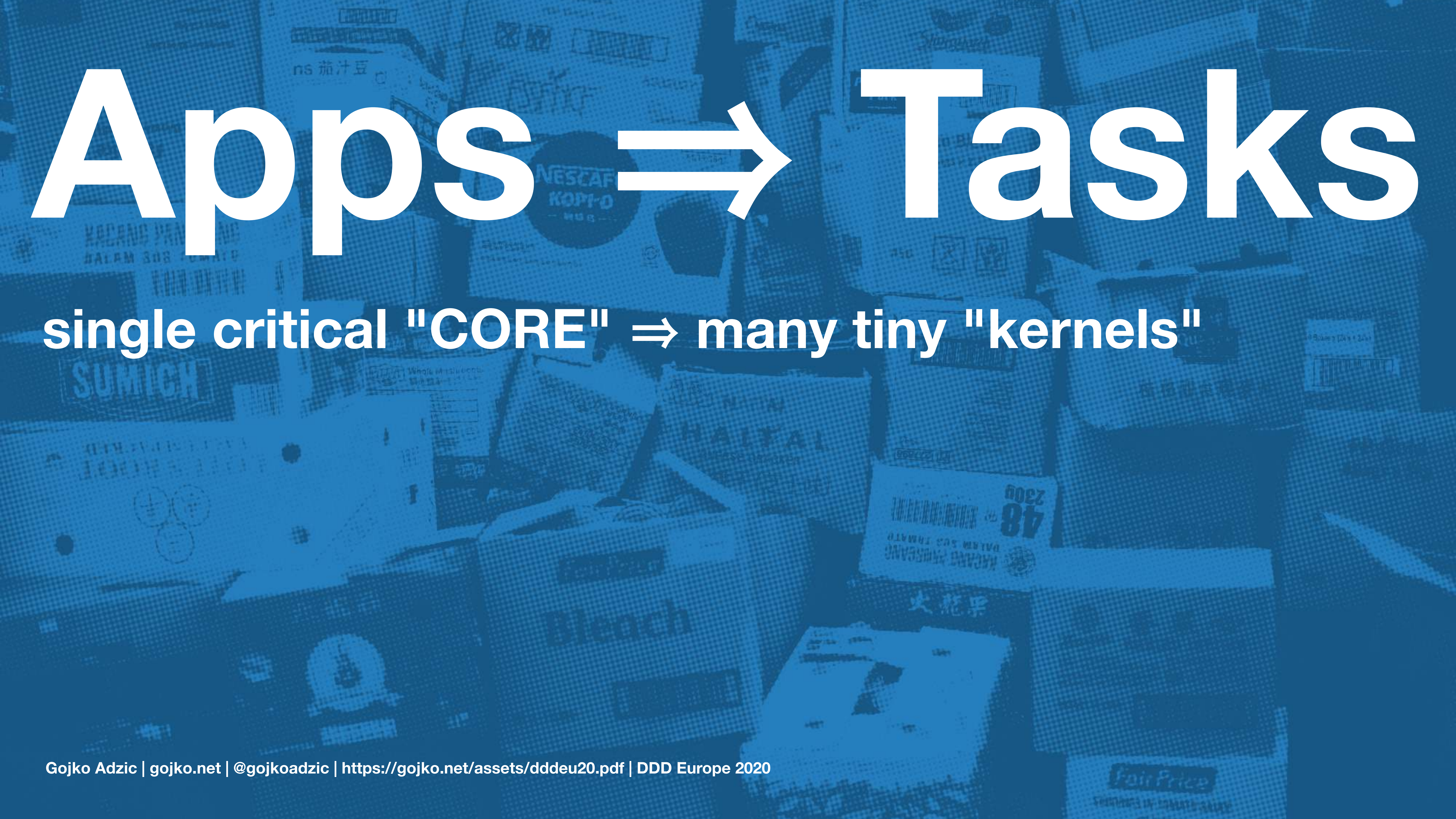
March 2019 cost*

$5,393

*All hosting costs are paid for through corporate partnerships.
100% of public donations go to the projects we fund.

OLD VS NEW

Caroline Rennie, Product Lead, Comic Relief: https://www.youtube.com/watch?v=Kb2Qk3vAkJI



Apps ⇒ Tasks

single critical "CORE" ⇒ many tiny "kernels"

Good news: very forgiving regarding design mistakes

Apps	*Tasks*
bounded contexts around teams, products	each "task" a potential context?
conceptual consistency	security/access
anti-corruption layers carefully planned	change blast radius inherently small

Bad news: "hello world" is highly distributed

https://martinfowler.com/bliki/FirstLaw.html

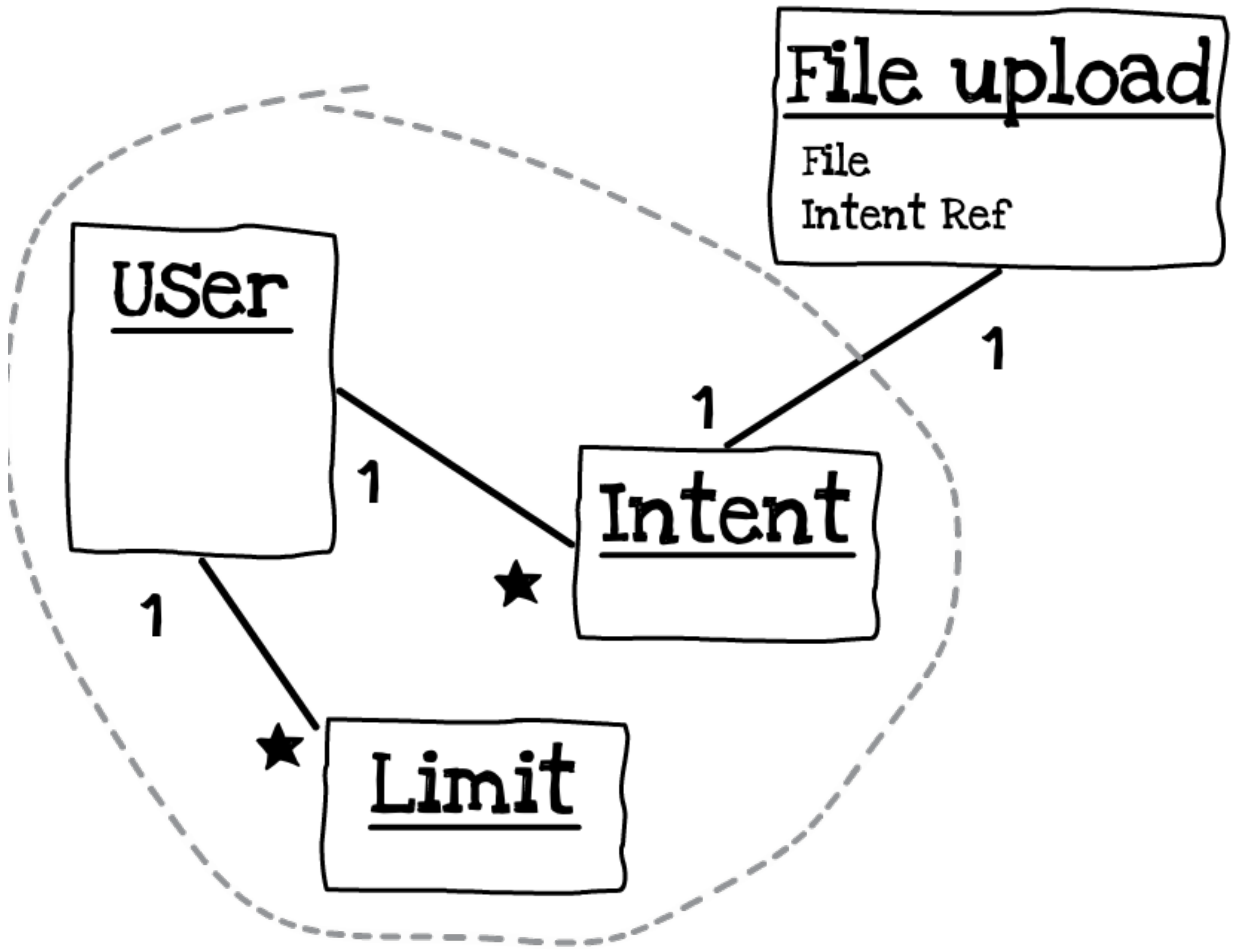
Don't

https://martinfowler.com/bliki/FirstLaw.html

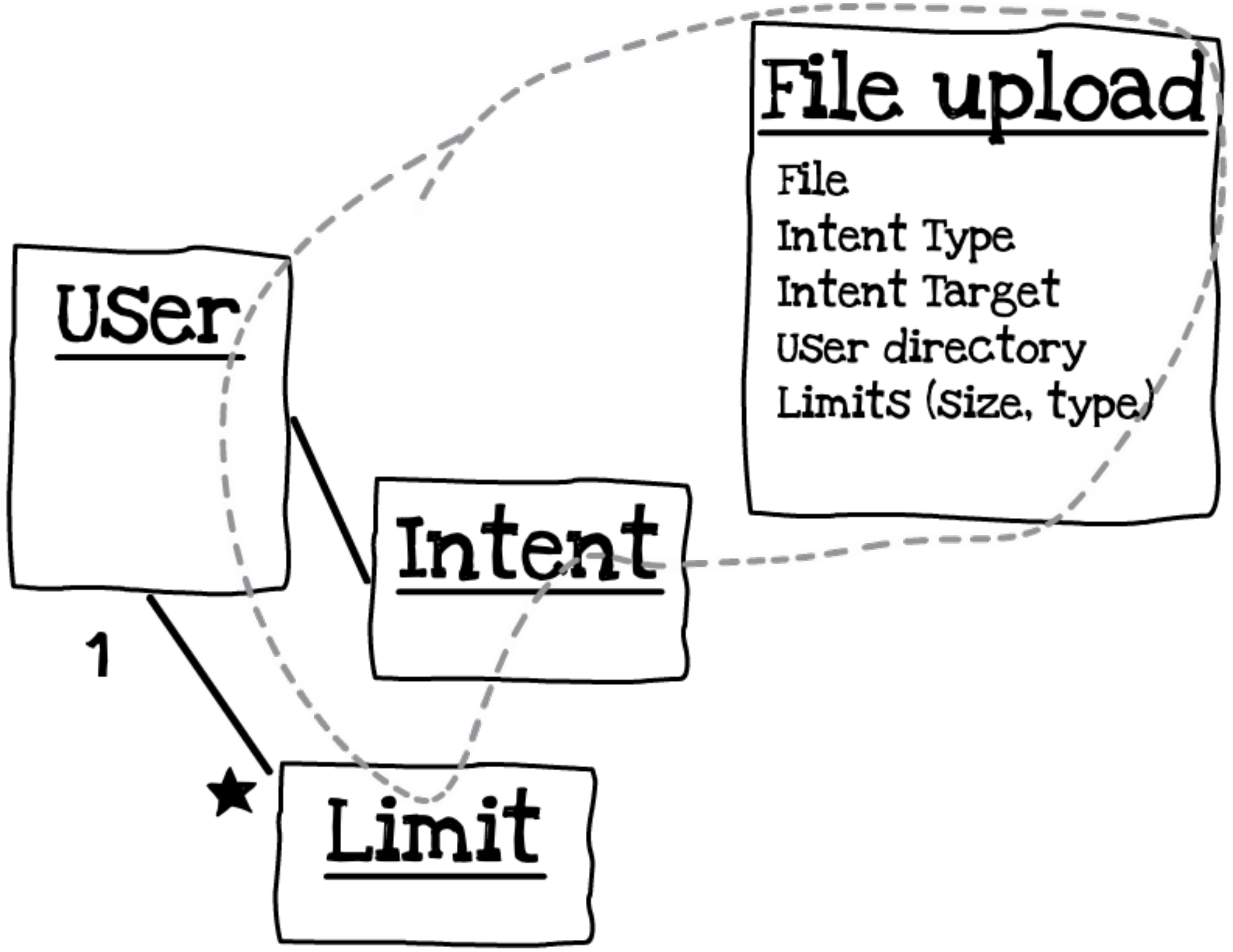
Don't distribute your objects

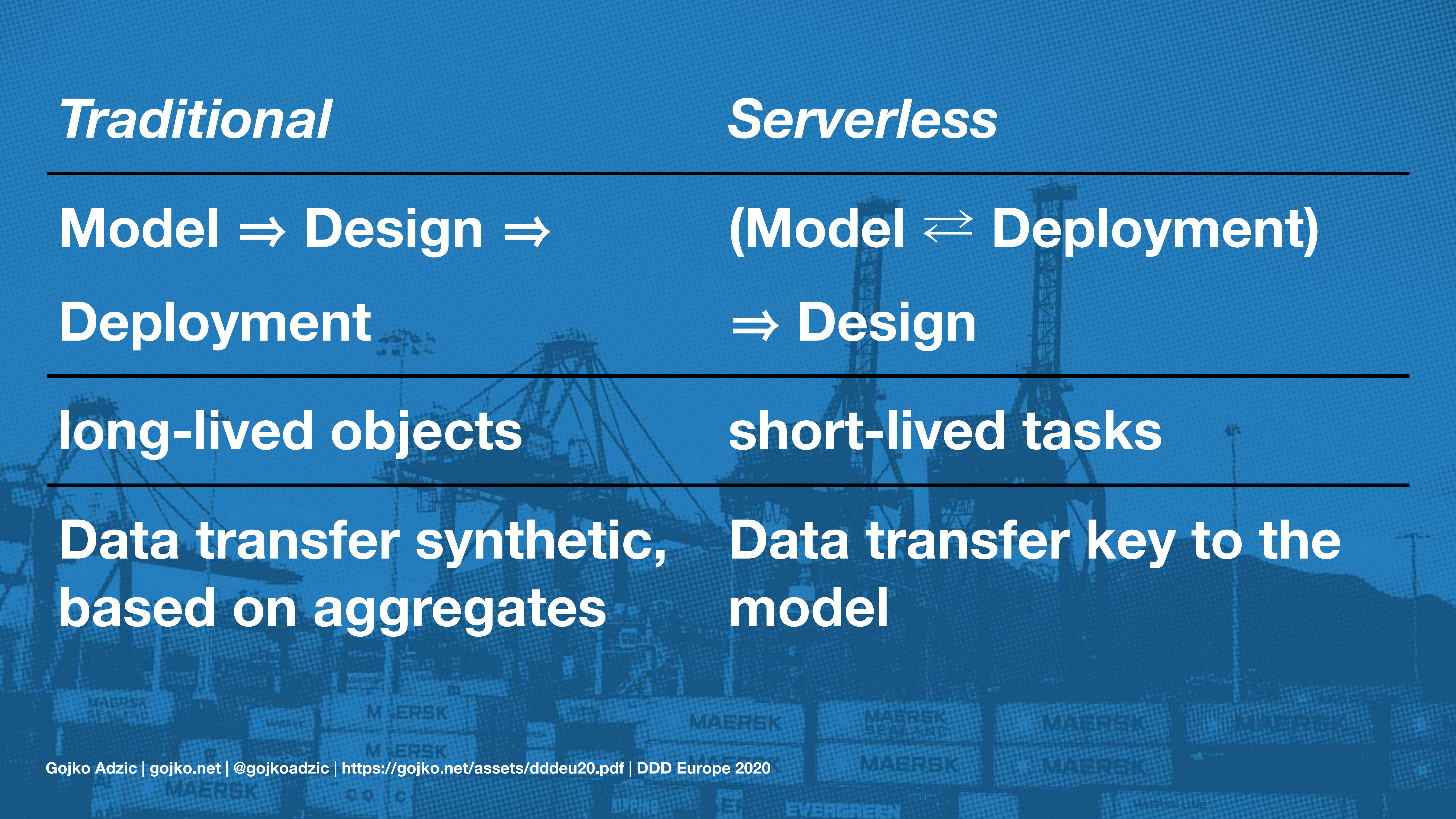
https://martinfowler.com/bliki/FirstLaw.html

Time=money, very literally



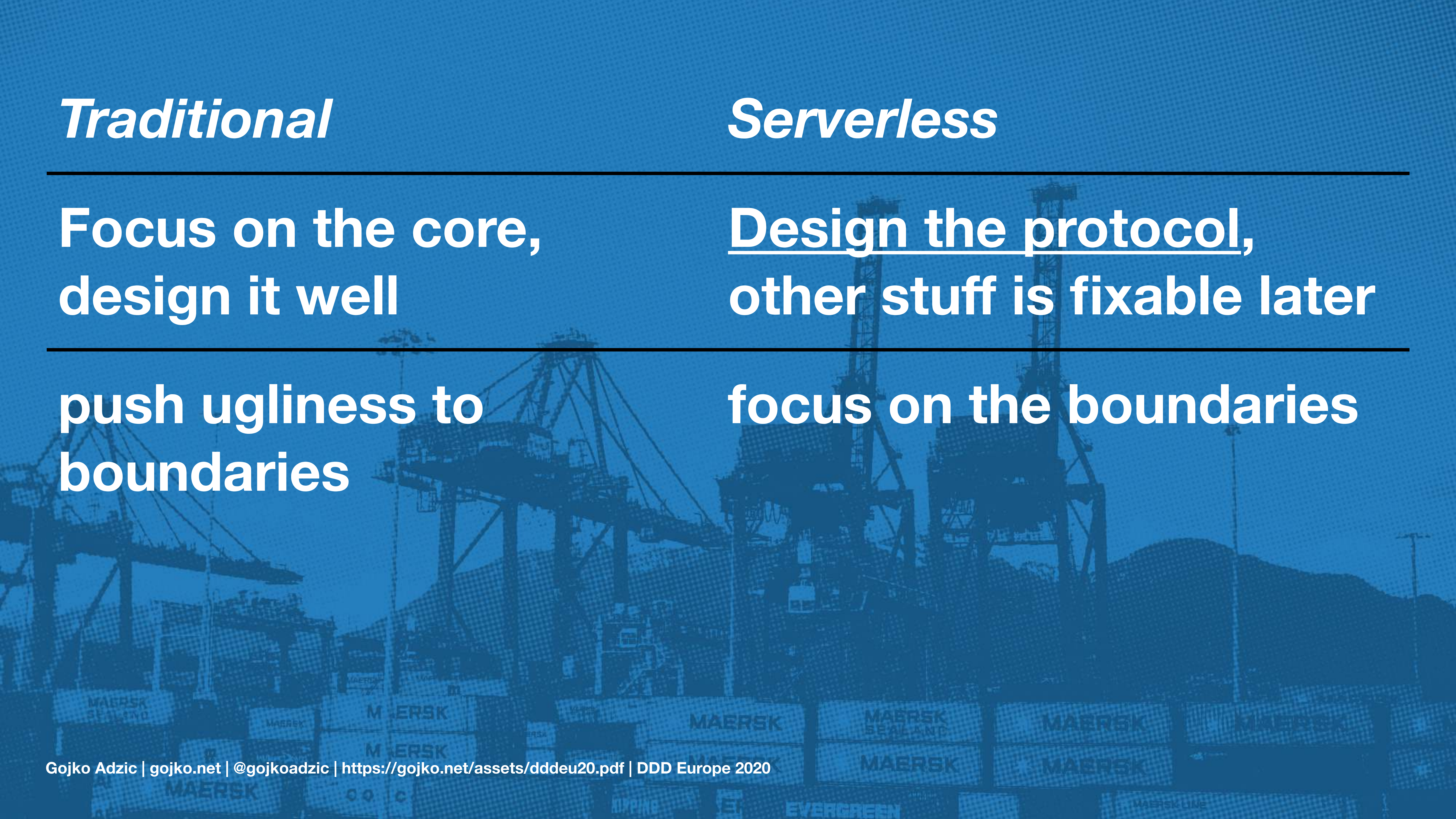
Time=money, very literally



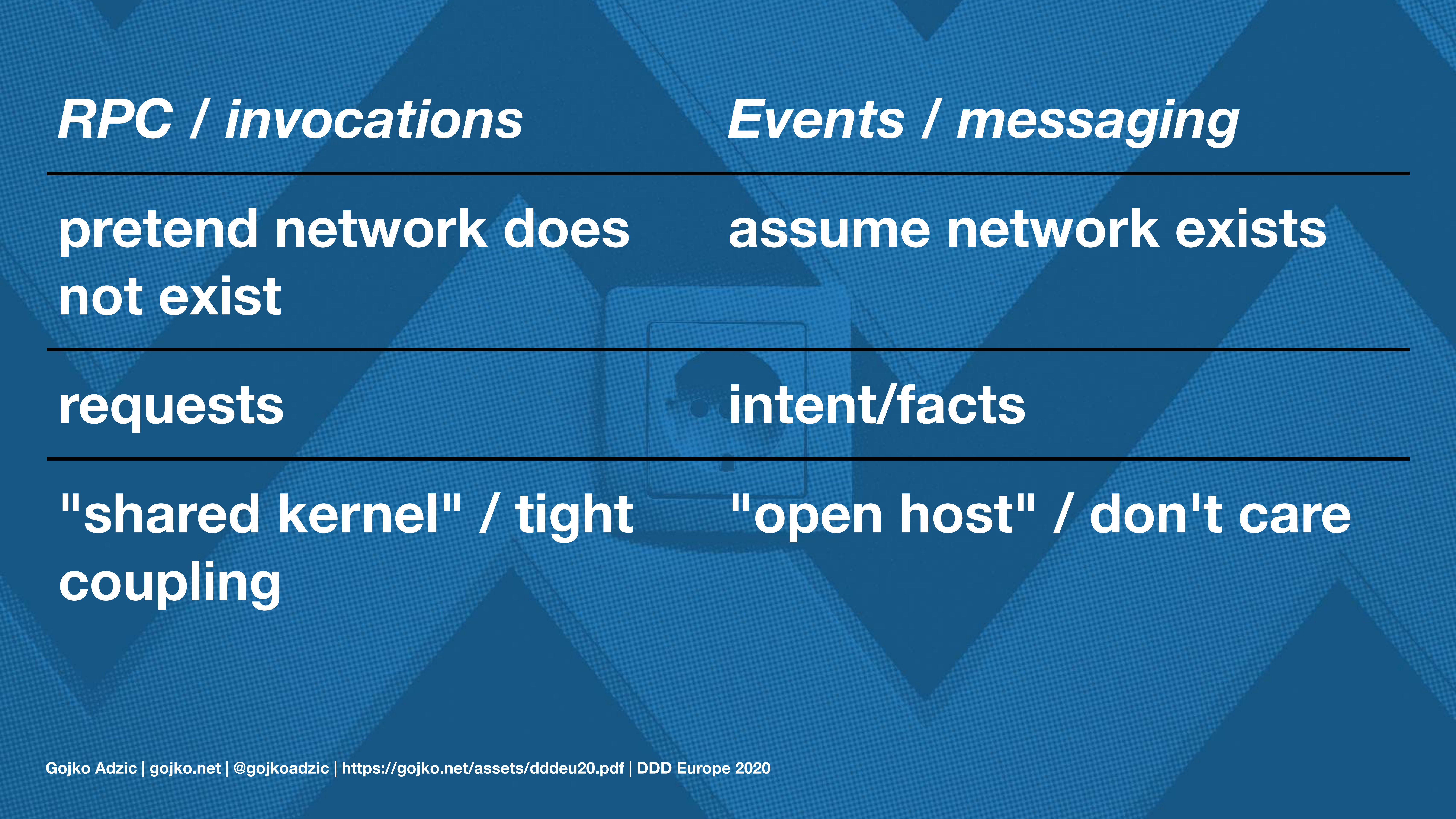


Traditional	*Serverless*
Model ⇒ Design ⇒ Deployment	(Model ⇄ Deployment) ⇒ Design
long-lived objects	short-lived tasks
Data transfer synthetic, based on aggregates	Data transfer key to the model

Events become "mini-aggregates"



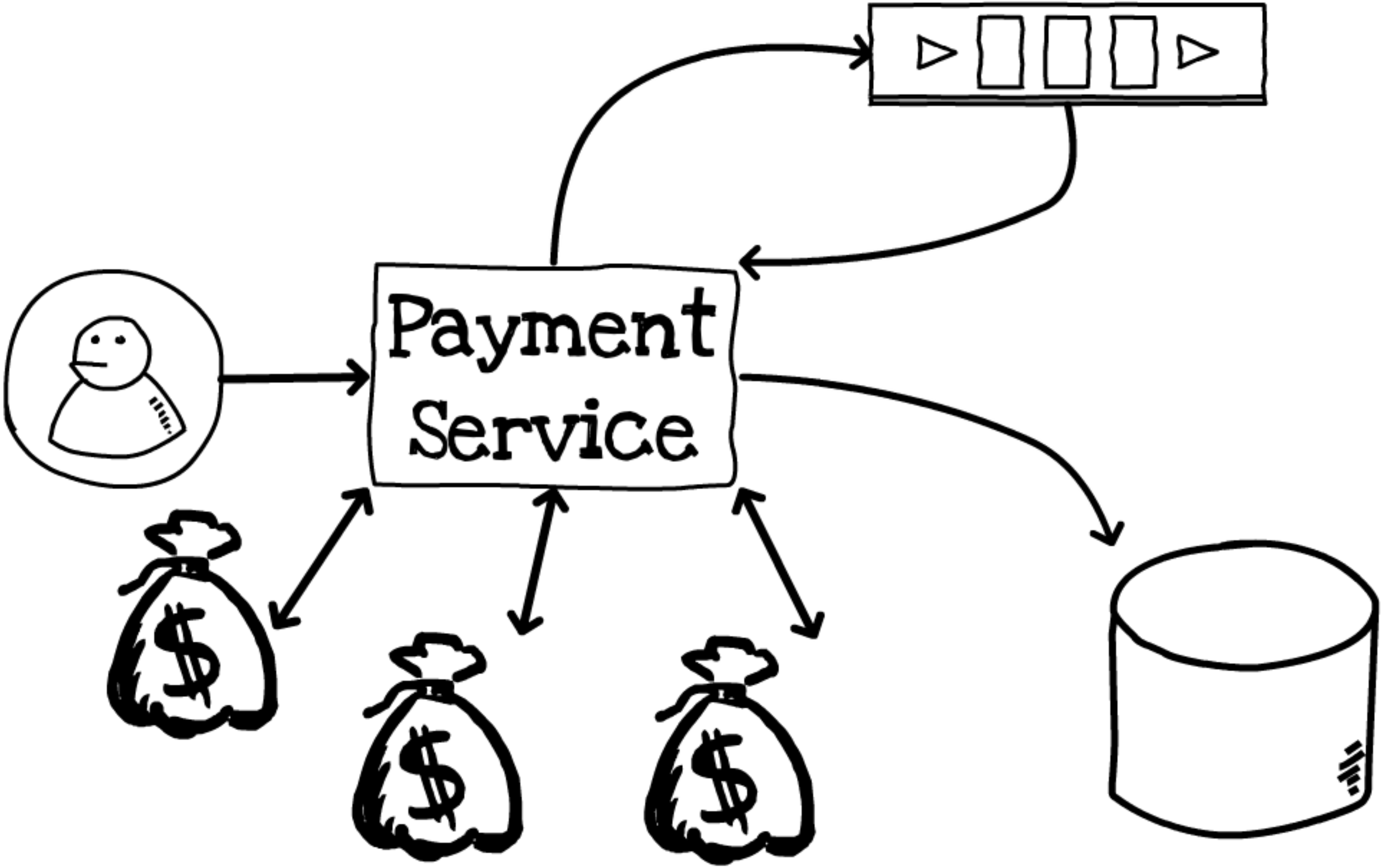
Traditional	*Serverless*
Focus on the core, design it well	Design the protocol, other stuff is fixable later
push ugliness to boundaries	focus on the boundaries

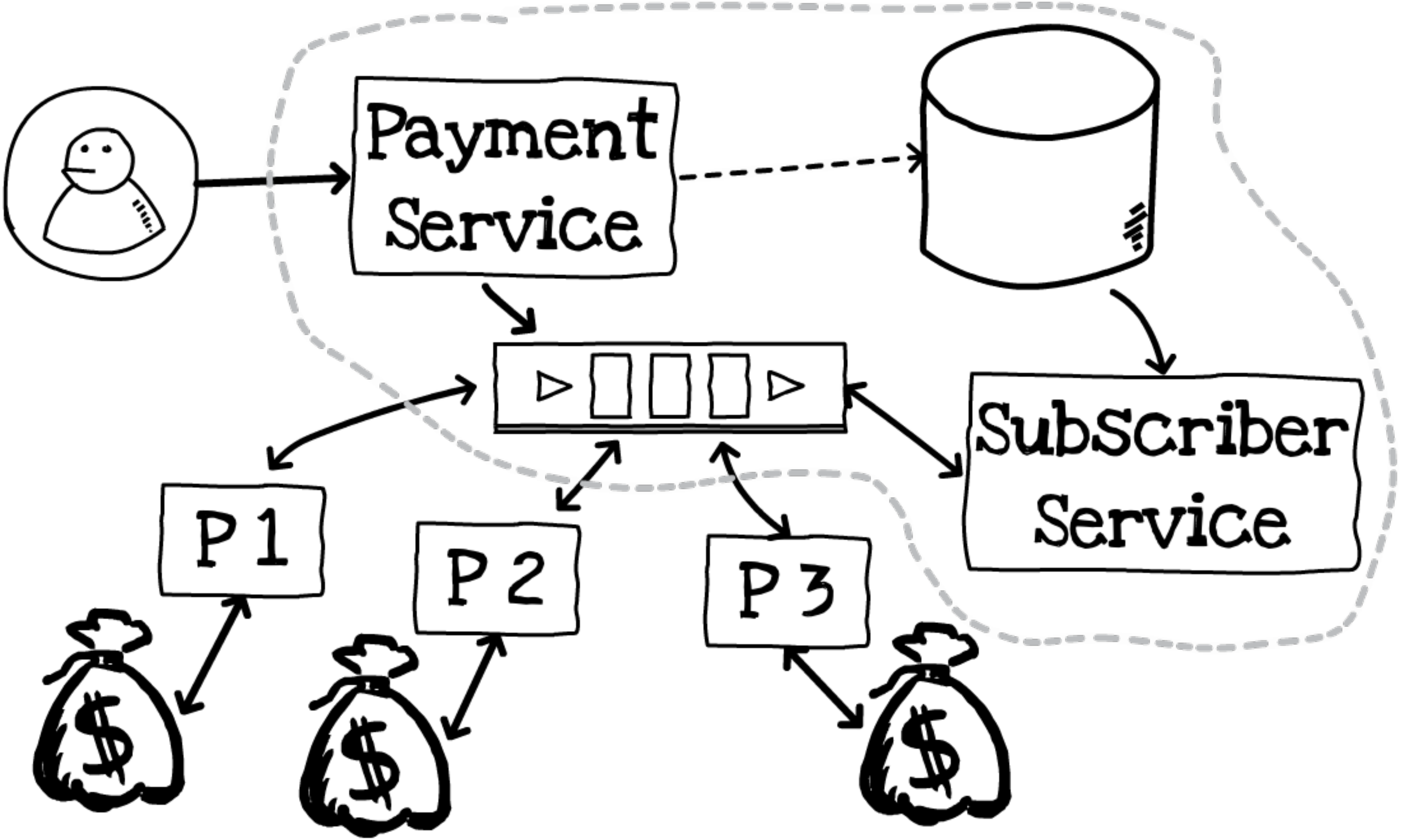


RPC / invocations	*Events / messaging*
pretend network does not exist	assume network exists
requests	intent/facts
"shared kernel" / tight coupling	"open host" / don't care

Key challenge for "protocol" design

Design events complete enough to avoid chattiness, but still generic enough to allow decoupling and reuse

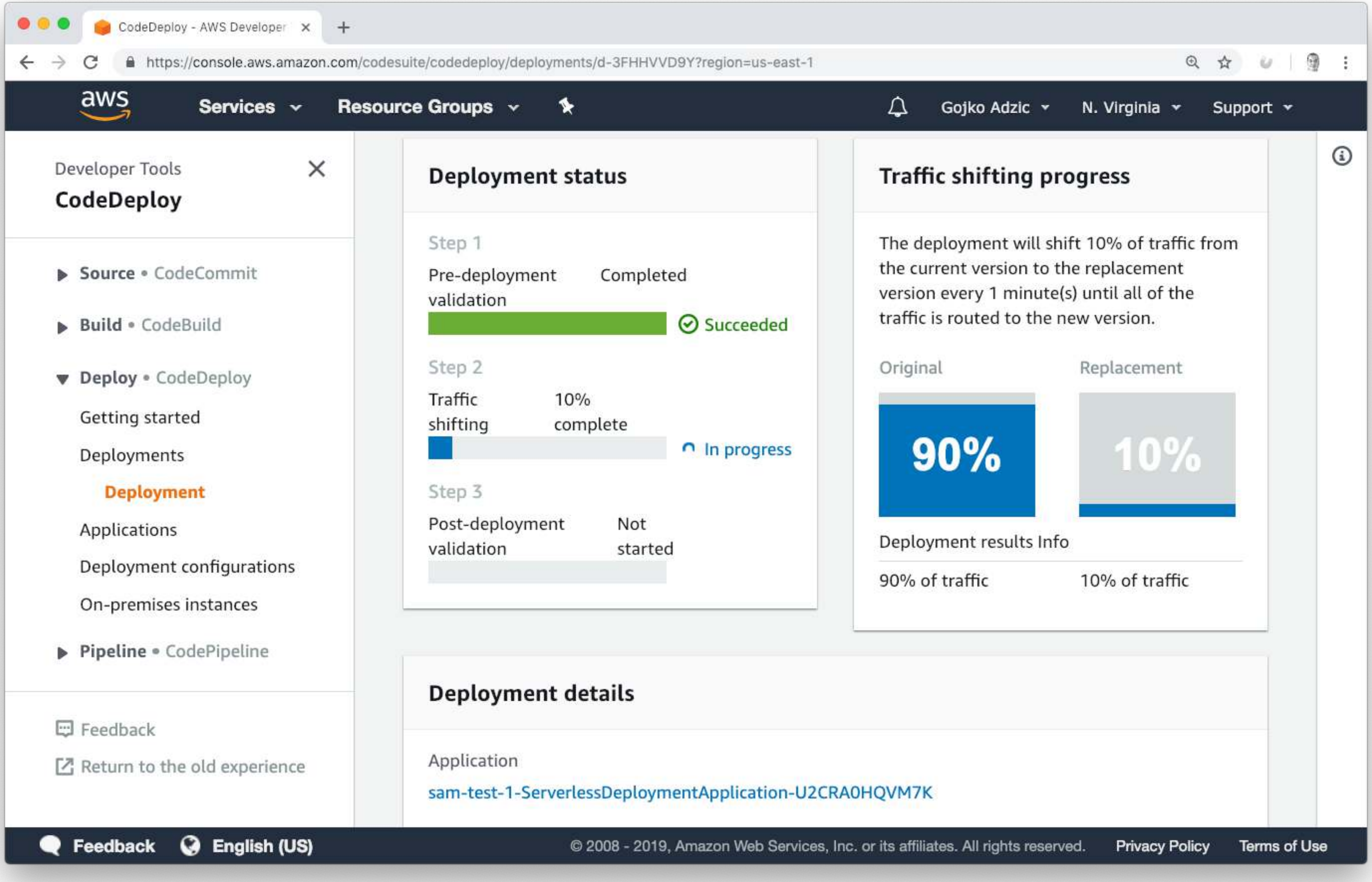

Payment
Service


Payment Service
Subscriber Service
P1
P2
P3
$
$
$

Groups of tasks end up as bounded contexts...

use runtime security needs as a hint about context boundaries!

Traditional	*Serverless*
Infrastructure is stateful or stateless	Infrastructure is transient
Reserved capacity	Utilised capacity



Deployment status

Step 1

Pre-deployment validation — Completed

✓ Succeeded

Step 2

Traffic shifting — 10% complete

In progress

Step 3

Post-deployment validation — Not started

Traffic shifting progress

The deployment will shift 10% of traffic from the current version to the replacement version every 1 minute(s) until all of the traffic is routed to the new version.

Original	Replacement
90%	10%

Deployment results Info

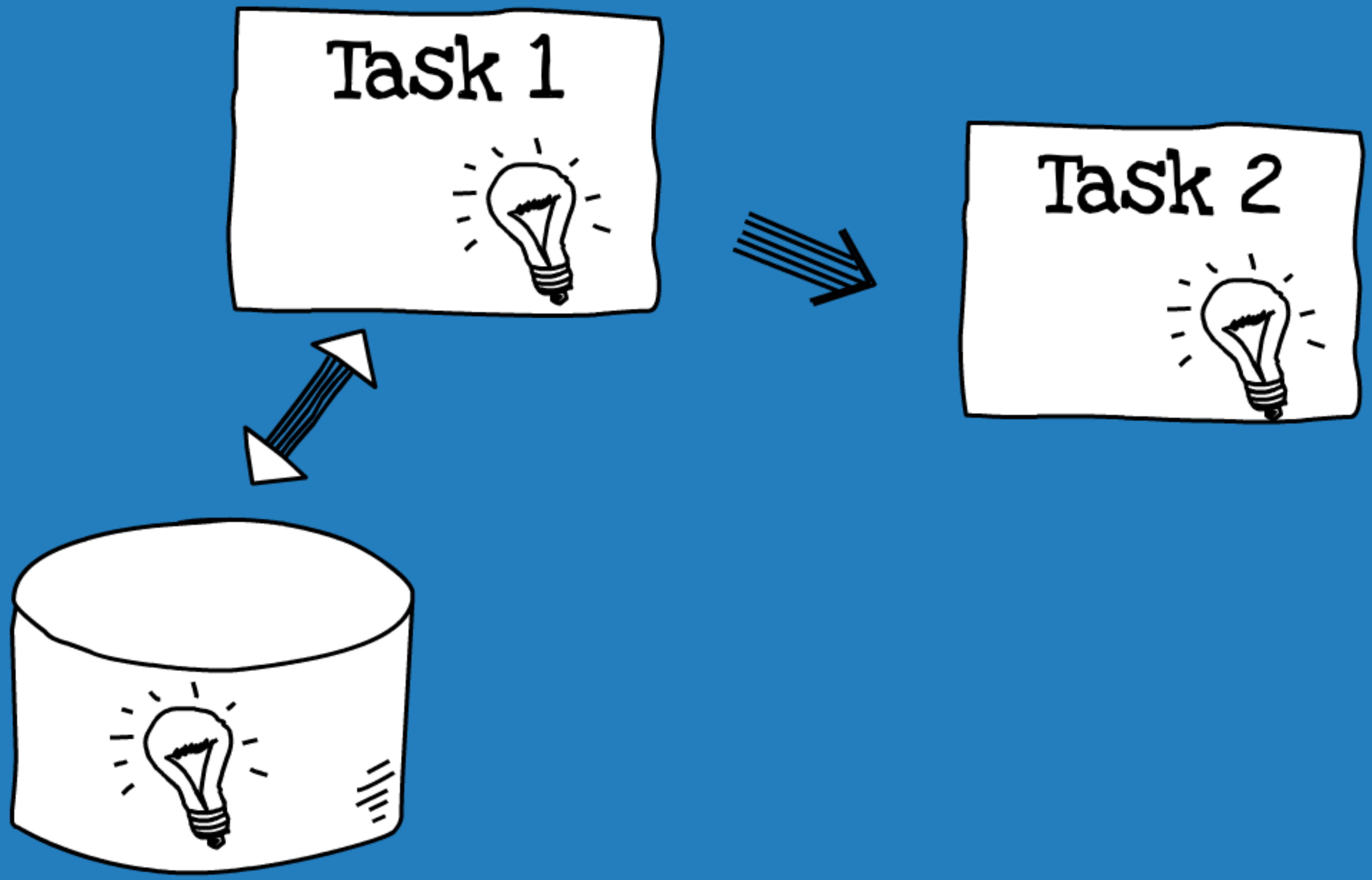
90% of traffic	10% of traffic

Deployment details

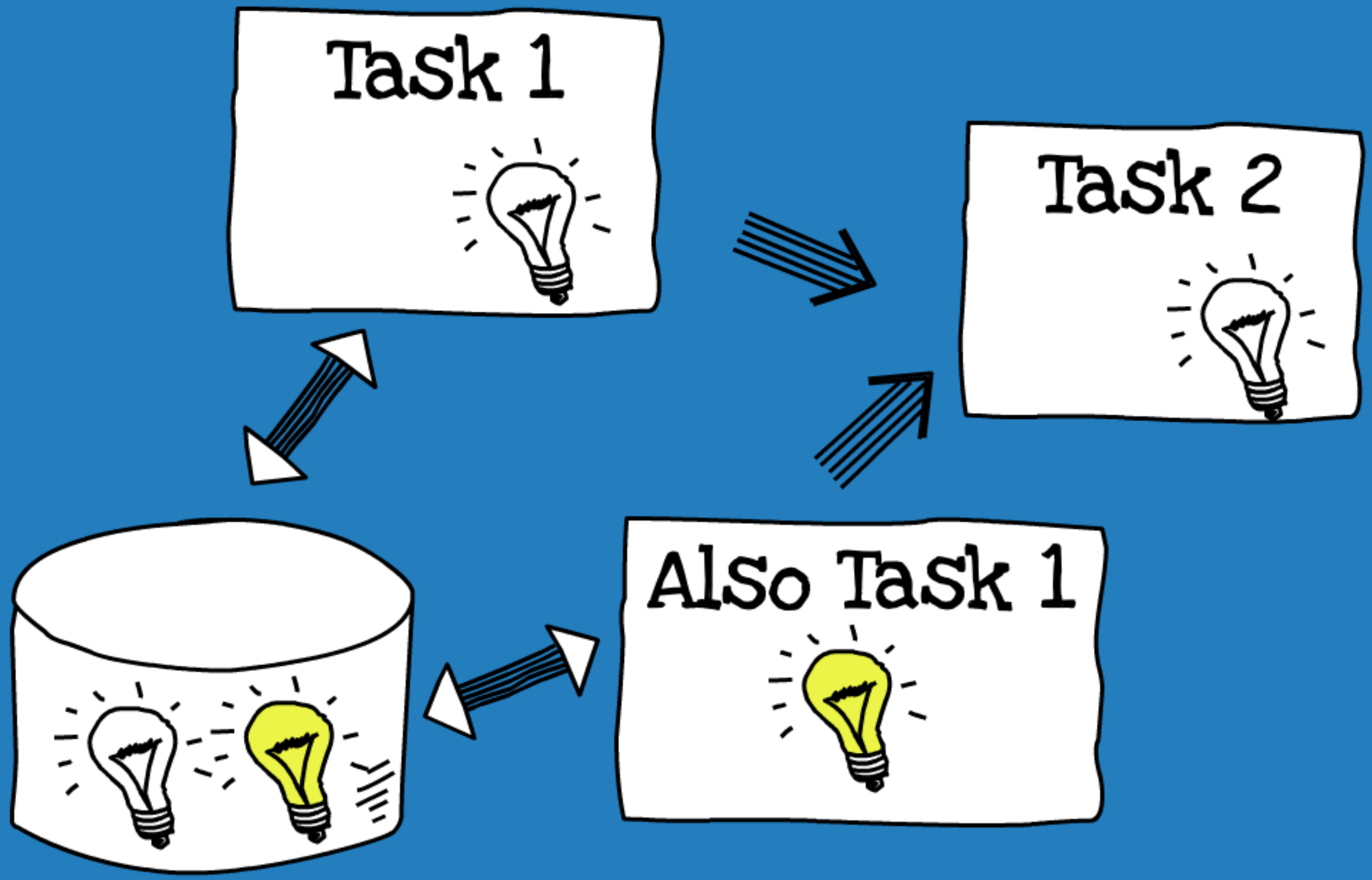
Application

sam-test-1-ServerlessDeploymentApplication-U2CRA0HQVM7K

Model changes over time, but consistent at any point in time



Model changes over time, may be inconsistent at single point in time



Traditional	*Serverless*
Infrastructure is stateful or stateless	Infrastructure is transient
Reserved capacity	Utilised capacity
Model Universe	Model Multiverse

Version-tolerant design

Gojko Adzic | gojko.net | @gojkoadzic | https://gojko.net/assets/dddeu20.pdf | DDD Europe 2020

RUNNING
SERVERLESS
Gojko Adzic
introduction to aws lambda and the serverless application model

RUNNING SERVERLESS

Gojko Adzic

http://leanpub.com/running-serverless/c/dddeu

50% off this week

RUNNING
SERVERLESS
Gojko Adzic
Introduction to aws lambda and the serverless application model