How to not be a headless chicken

Tips for embracing decoupled content architectures

Phil Hawksworth, Netlify

These slides and links

findthat.at/chicken

Why this talk?

9 years as Technical Director at agencies

Clients

Mo clients, mo problems

(Mo clients, mo architectures)

unsplash.com/photos/Bli_WDC2oU0

Old challenges

Timeless challenges

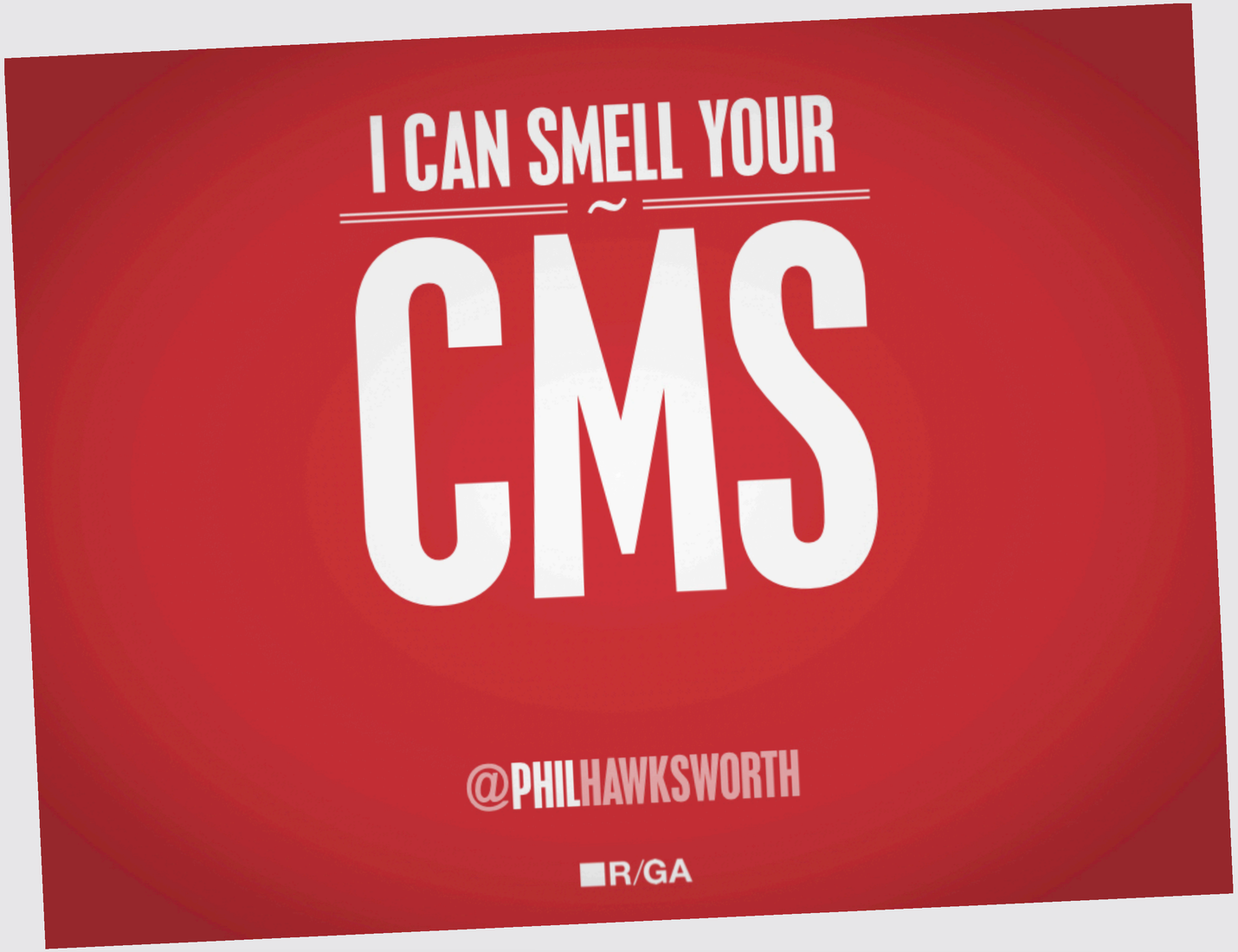
Large investments

(Time, money, emotion)

The result

Code smells

CMS smells


I CAN SMELL YOUR
~
CMS
@PHILHAWKSWORTH
R/GA

```
<p class="footnote"> </p>  
<p class="footnote"><strong></strong><strong></strong> </p>
<p class="footnote"> </p>
<p class="footnote"> </p>  
<p class="footnote"> </p>  
<p class="footnote"> </p>  
<p class="footnote"> </p>  
<p class="footnote"><strong></strong> </p>
<p class="footnote"> </p>
```

@PHILHAWKSWORTH #SMELLCMS

Who is this talk for?

Not to persuade you that headless CMS are a great choice

Help you recommend a headless CMS to your clients / stakeholders

How to not be a headless chicken

Tips for embracing decoupled content architectures

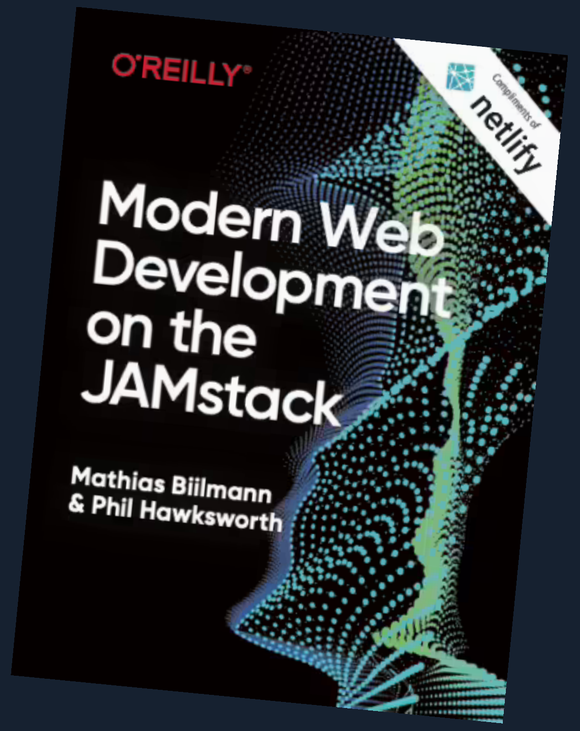
Phil Hawksworth, Netlify

Phil Hawksworth

Director of Developer Experience, Netlify

@PhilHawksworth

Director of Developer Experience, Netlify



findthat.at/**jamstack/book**

REMOTELY INTERESTING

findthat.at/**interesting**

Let’s talk

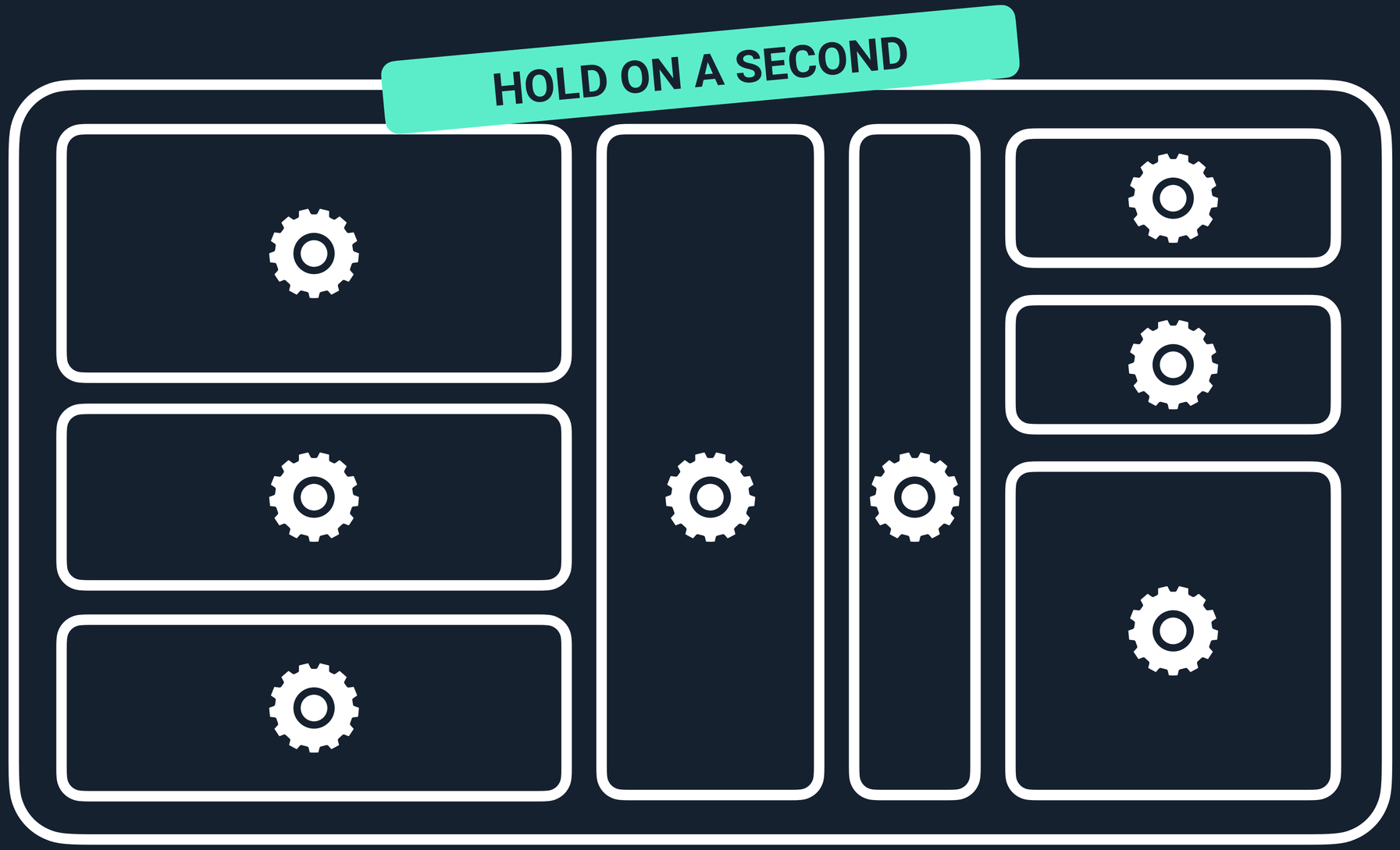
1. The old skool and the shift
2. Objections to overcome
3. A few of the benefits

1

The old skool and the shift

unsplash.com/photos/K6kZi3AmGR8

HOW BEAUTIFULLY SIMPLE


HOLD ON A SECOND

EXPENSIVE MAGIC

THINGS

THINGS

Tight coupling

Often not well understood

A product of evolution

An evolving product

Experts required

And available for those with the budget

One-stop-shop

Looooooooong feature lists

Industry standard, best in class, enterprise grade, globally recognised, reassuringly expensive

Nobody ever got fired for choosing [_____]

Except they did

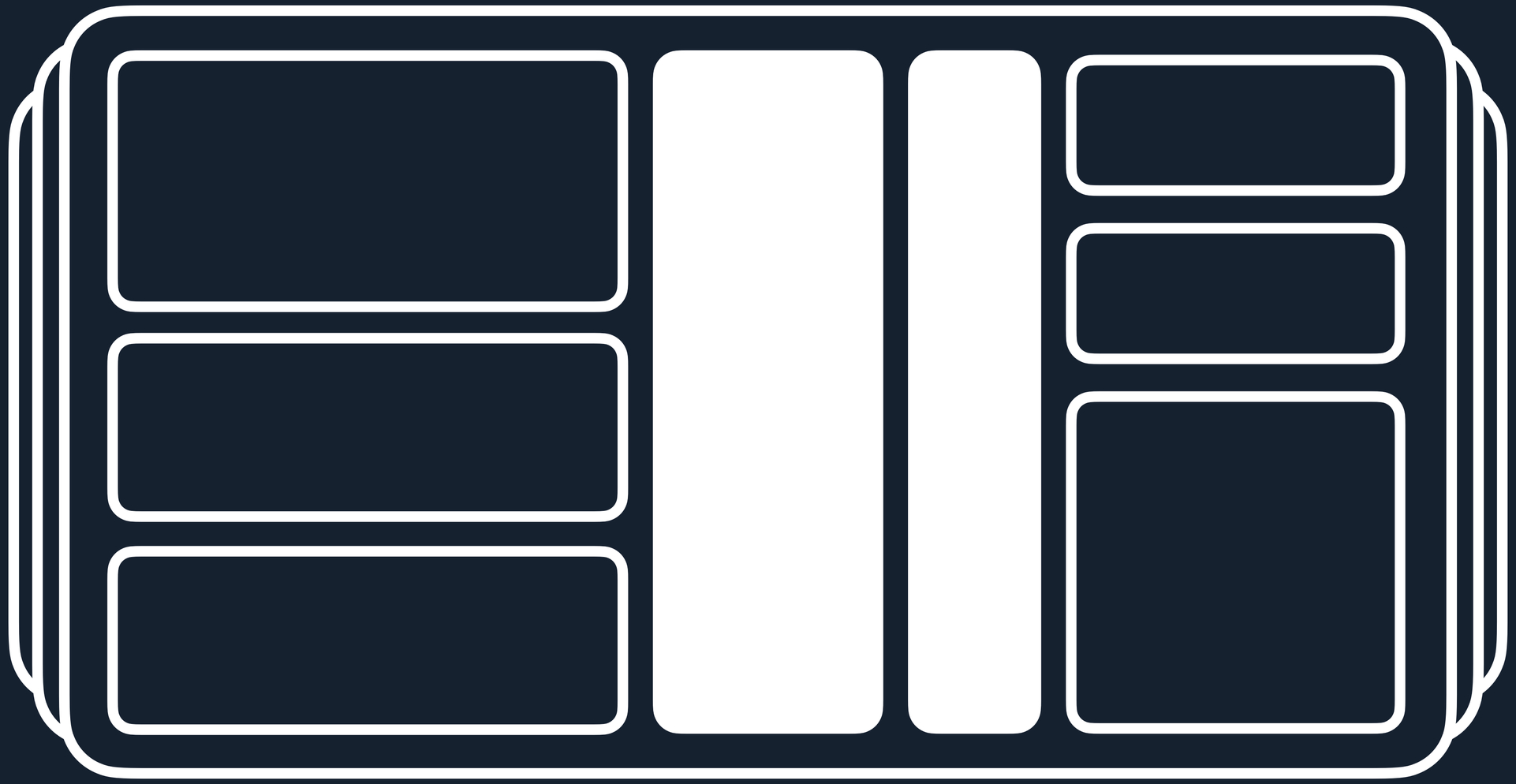
unsplash.com/photos/uIPAVuxITEw

Ability to execute

The CMS became the biggest obstacle to successful delivery

I found myself seeking, work arounds and shortcuts

As did customers



Workarounds became common and formalised

Results of workarounds outperformed core platform

If your CMS does not have an API in 2021, perhaps you shouldn't be using it

— Knut Melvær
July 26th 2021 (at about 6:51pm)

2

Objections to overcome

Moving from a monolithic to a decoupled architecture can be daunting

All these dependencies add too much risk

What if they go away?
What if they break?
Can they be trusted?

All these dependencies add too much risk

Their entire business depends on their continued good standing of this specific service.

Their success demands dedicated domain experts.

What team would you allocate to each of these capabilities?

Using lots of services will impact performance

This is going to run like a dog

Using lots of services will impact performance

There is likely to be far more dedicated engineering and infrastructure for each dedicated service than you'd be able to dedicate.

A service which is "nearby" isn't necessarily faster.

We can’t let this content our of our control

You’e suggesting we give away the family jewels

We can’t let this content our of our control

Each vendor is likely to have more strict and established compliance and regulations than you'd be ready to commit to yourself.

Look for certification and compliance when selecting vendors.

Decoupling can also mean leveraging first party APIs.

Using these services will lock us in

What if we want to make some changes?

Using these services will lock us in

Using APIs demands that you have clear delineation and build to defined integrations.

You'll probably have looser coupling and greater flexibility than building everything together in-house.

The headless CMS category is exploding

3

A few of the benefits

Decoupling

Time to market can drastically improve

Let alone time to dev environment!

Your feature roadmap is not in lockstep with a that of the platform

Your services operated by experts

so you can build and differentiate

The skills you need are widely available

Front-end code is no longer limited to being a product of a back-end system

Your results
Are back under
your control

To finish

DO YOU?

Have difficulty providing environments in time for project deadlines?

Have frustrations about time to market or the freedom to build what you envisage?

CAN YOU?

Try this architecture on an internal or ancillary project?

Resource a project which would need devs versed in front-end and APIs?

Courage & confidence

findthat.at/chicken

Thanks

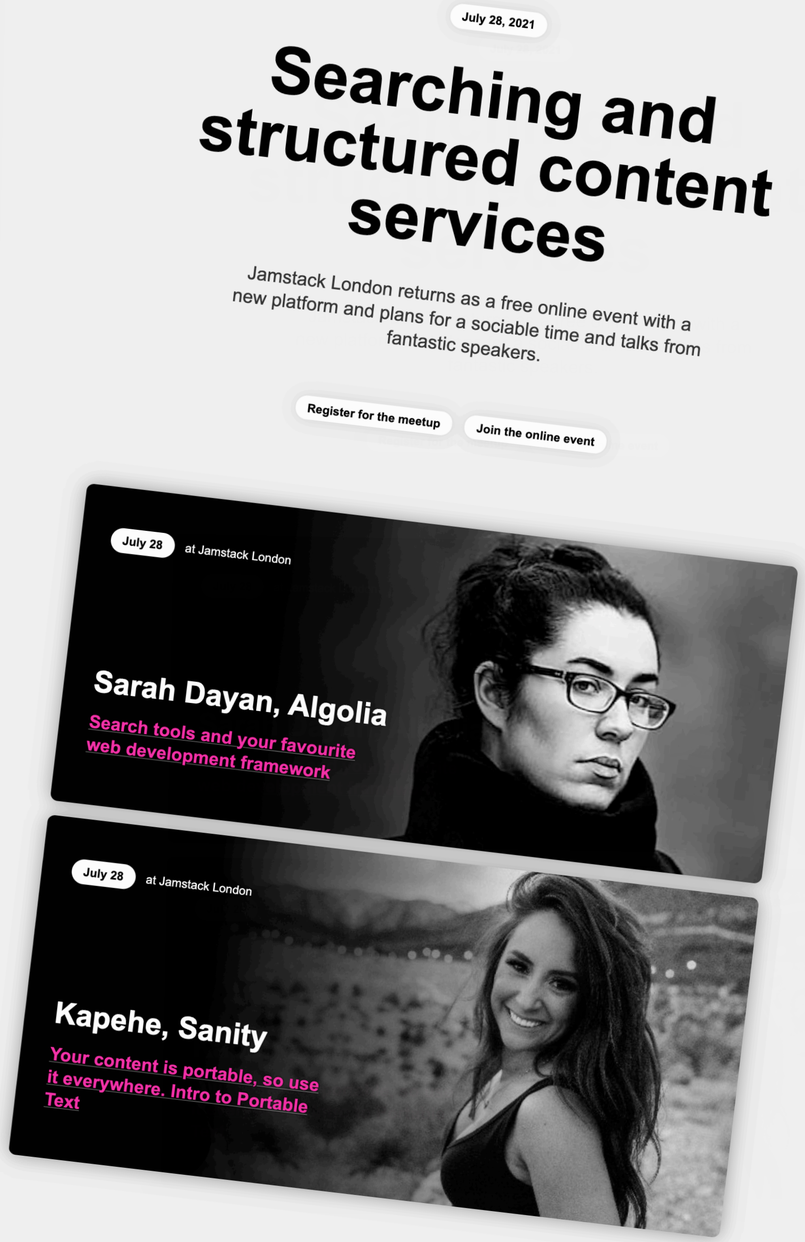
@philhawksworth

Jamstack Community Survey 2021

findthat.at/jamstack/survey

This Wednesday

jamstacklondon.com



findthat.at/chicken

Thanks

@philhawksworth