Delivering JS at scale

LeicesterJS | 21st March 2019

comparethemarket™

part of the

Who the hell are we?

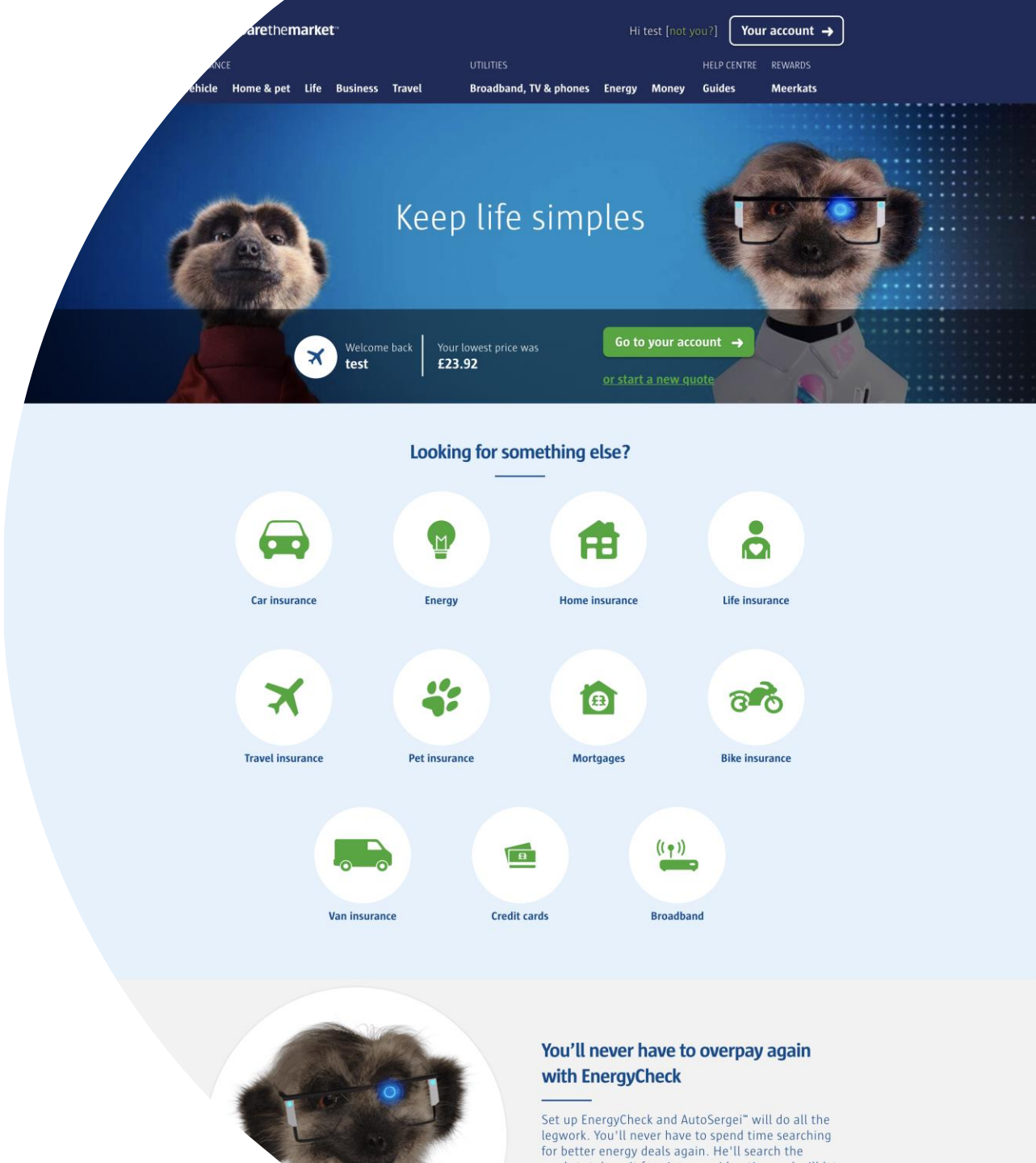
James Ford – Tech Lead | Meerstrap

Tom Foyster – Tech Lead | Motor

comparethemarket™ part of the bgl group

What do we do?

- Collect customer information and send it to multiple providers.
- Process the data that providers send back and display it.



“At Scale?”

What we mean when we say scale.

comparethemarket™

part of the bgl group

Scale: Products

Scale: Motor (March 5th – 11th)

900K
users

1.2M
sessions

16M
pageviews

Scale: People

18
Teams

300
People in 'Tech'

3
Locations

Scale: Code

90
Active Repos / Day

148
Microservices

300
Releases / Month

Risks

The risks we face when operating at scale.

comparethemarket™ part of the bgl group

“We're Number One!”

- Which makes us a great big target.
  - GDPR
  - FCA requirements
  - Bad people

- If we screw up, we have a big impact.
  - On customers
  - On revenue
  - In penalties / reparations

“Delivery?”

How are we able to deliver at scale?

part of the 

Measuring delivery

(For the purposes of this talk)

- Number of successful releases
  - Positive impact on Customers

Delivery is measured by the release of new code.

How to deliver at scale

Lessons we've learnt (the hard way), solutions we've made

part of the 

Our tech history

In the beginning…

- On-premise Monorepo(s)
- Multiple products, individual teams

Five years ago

- Cloud-based Microservices
- Architecture & Microservice teams

Two years ago

- Shared code libraries
- Code library teams

How not to deliver at scale

Monorepos: Dependency hell.

- Complex codebase
- Too many interdependencies
- Side effects and breaking code

Low confidence in releases

Lots of cross-product communication needed

Solving the complexity challenge

- Agile development, smaller stories
- More independence, safer releases
- Microservices and focused, dedicated codebases
- Broader technology base (best for specific use cases)

But…

- Still have repeated / duplicate effort

Solving the consistency challenge

- Shared libraries, components
- Dedicated teams for shared code

- More efficient
- More consistent
- Best in class solutions

Inner Sourcing

Where we are today

part of the 

Inner sourcing

- Meerstrap
  - Work with product teams to find commonalities
  - Improve and standardise the common stuff
  - Establish default structures and architecture patterns
  - Point of collaboration between development ‘Rock Stars’
  - Let product worry about product
  - Create an adoption ‘path of least resistance’

Meerstrap

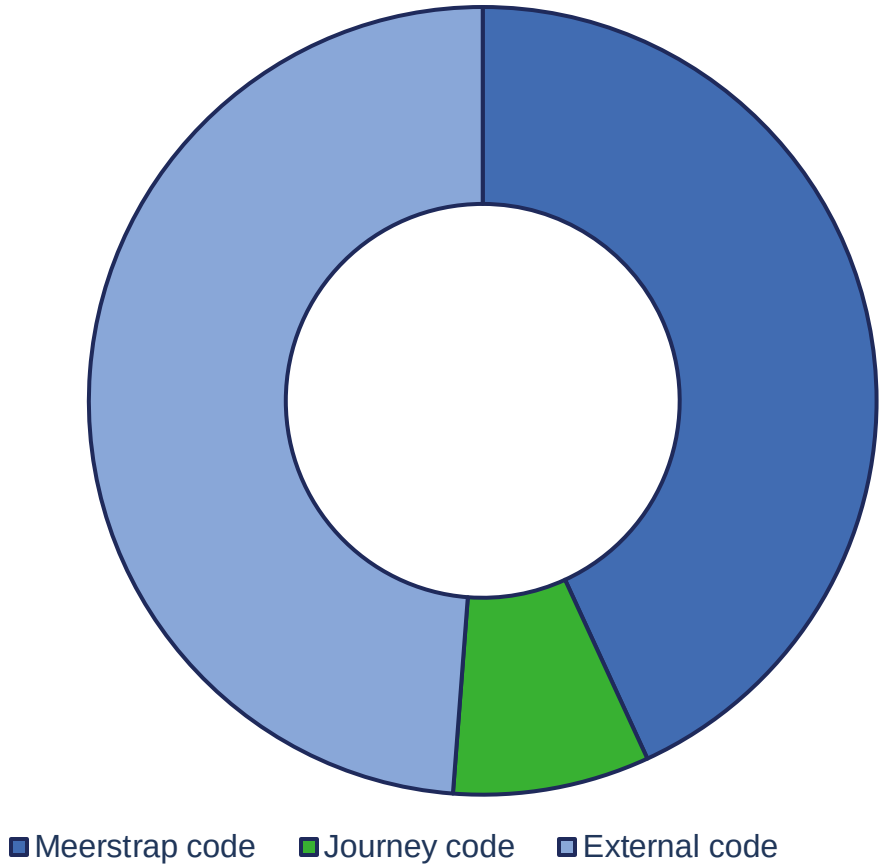
Scale: Meerstrap

13
Teams sharing

7
developers

140
shared atoms

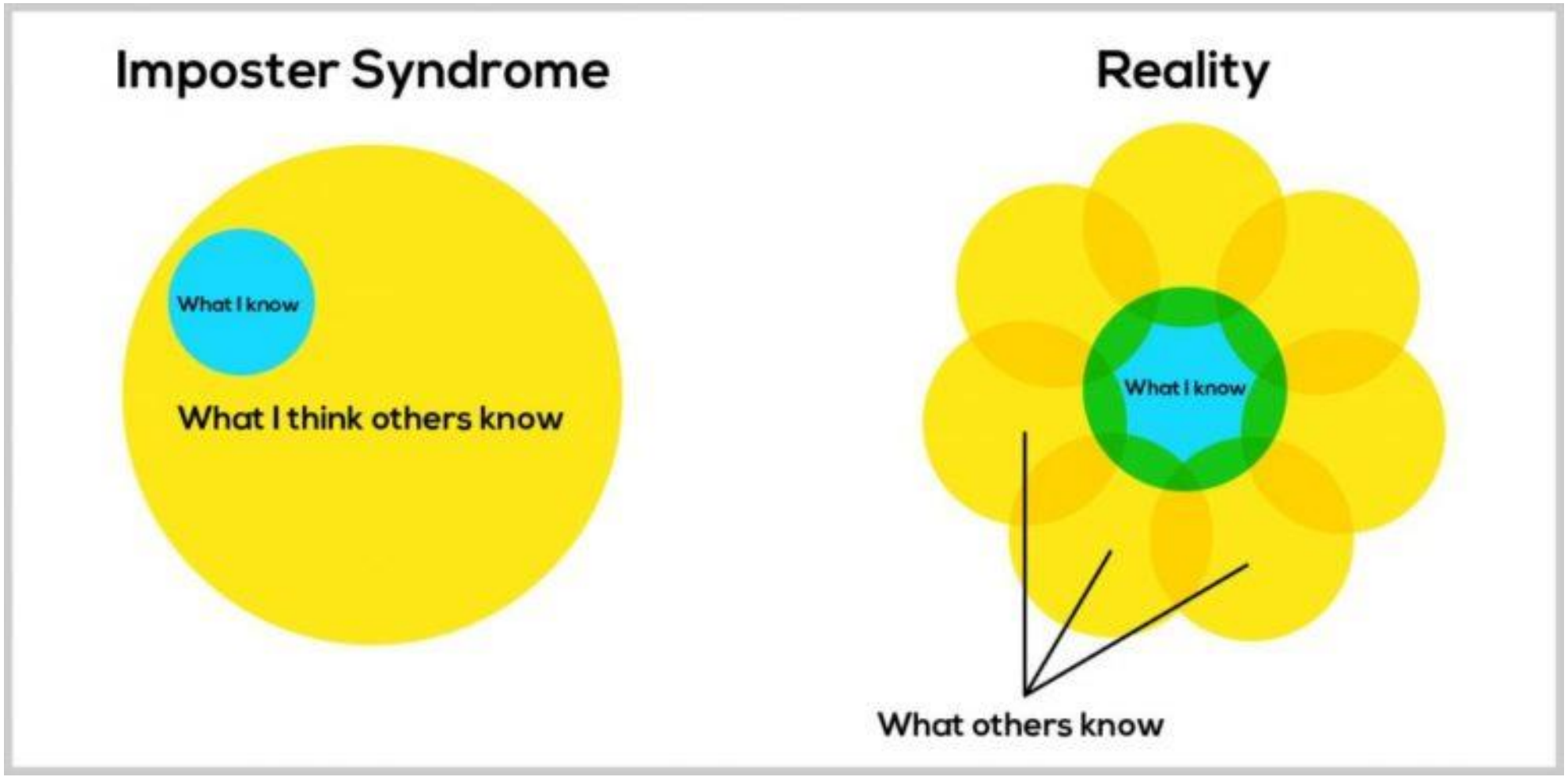
Source Maps!



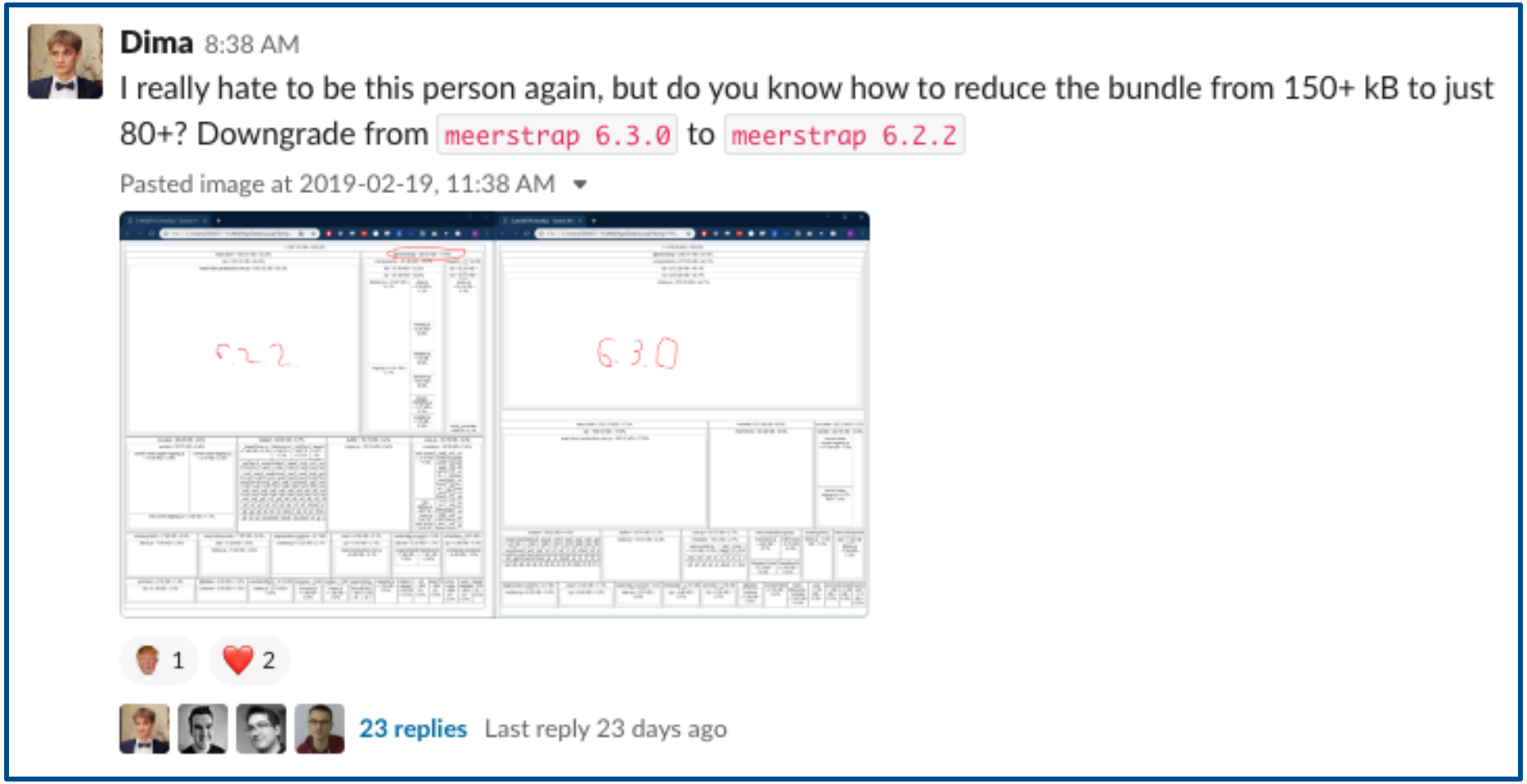
Unique code:
7%

Shared code:
37%

The domain knowledge challenge



Tough crowd.


Dima 8:38 AM
I really hate to be this person again, but do you know how to reduce the bundle from 150+ kB to just 80+? Downgrade from meerstrap 6.3.0 to meerstrap 6.2.2
Pasted image at 2019-02-19, 11:38 AM
6.2.2
6.3.0
1
2
23 replies
Last reply 23 days ago

Today's challenges: Solved?

- Encouraging the right behaviours
- Encouraging developer participation
- Accepting that v1 is not necessarily the best

- Dedicated team (because it's a lot of work)
  - Have a roadmap
  - Act as the quality gate for solutions
  - Take up the negotiation, resolution process for changes
  - Treat developers as customers:
    - Maintain compatibility
    - Work out migration paths
  - Communication, training, developer support

Delivering JS at scale

Our Main Theme

part of the

Confidence.

Without it, you will not make release.

comparethemarket™ part of the bgl group

Ensuring confidence

- Guard against known risks
- Have an undo process
- Improve ways of working

“If we screw up, it has a big impact.”

Mitigating risk: Technical solutions

- Testing
  - Unit tests
  - End to End tests
  - Cross-browser, cross-device tests
- Monitoring
  - Uptime
  - Errors and logs
- Deployments (and rollbacks)

Mitigating risk: How we work

- How we work
  - Pair and Mob programming
  - Multi-role sign-off processes

How to gain confidence

- Reduce your scope
- Abstract your complexity
- Avoid side-effects
- Test
- Monitor
- Build a safety net

Be confident, and you will release.

Final thoughts

Time for some words of wisdom.

comparethemarket™ part of the bgl group

Tomorrow's challenges

We still have challenges.

- We don't want to create bored, unengaged developers
- We want to encourage the right mindset
- We must avoid creating knowledge silos
- We should avoid wasted effort

Takeaways

- Scale
  - has its own risks
  - makes certain ways of working possible
- Delivery
  - relies on confidence
- Confidence
  - comes from testing, monitoring, and ways of working
- Adoption
  - works when it's the path of least resistance
  - works when you're open to collaboration
  - works when it's not restrictive and doesn't disenfranchise people

That’s all folks.

@comparemktech

@TomFoyster

@psyked

comparethemarket™

part of the bgl group