Watch out!
The nanoservices are coming

Matthew Clark
@matthew1000
Head of Architecture, BBC D&E North & Nations
Velocity London, 19 October 2017
This is the story of the \textit{nanoservices pattern}.

Small units of logic that are easily understood, deployed, operated and shared.
From own data centres

To the cloud!

From fewer, bigger releases

To Continuous Delivery!

From separate development, operations, UX and test teams

To devops!
Cloud, CD and devops give you:

Technology freedom!
Operational freedom!
Capacity on demand!
Faster, more frequent releases!
E2E Ownership!
Make the common things easy and the special things possible.
<table>
<thead>
<tr>
<th>Pros</th>
<th>Microservices</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple teams</td>
<td></td>
<td>Maintenance overhead</td>
</tr>
<tr>
<td>Multiple technologies</td>
<td></td>
<td>Comms overhead</td>
</tr>
<tr>
<td>Smaller and safer deploys</td>
<td></td>
<td>Multi-version challenges</td>
</tr>
<tr>
<td>Shareable &amp; flexible</td>
<td></td>
<td>Harder to refactor</td>
</tr>
<tr>
<td>Easily scalable</td>
<td></td>
<td>Cost</td>
</tr>
</tbody>
</table>
[Microservices] bring an enormous amount of operational complexity to the table. Suddenly, you need to continuously deploy many different (possibly containerized) services. New concerns arise: service discovery, distributed logging, tracing and so on. You are now even more prone to the fallacies of distributed computing. Versioning of interfaces and configuration management become a major concern. The list goes on and on.

Sander Mak
https://www.oreilly.com/ideas/modules-vs-microservices
Microservices

Microservices is a variant of the service-oriented architecture (SOA) architectural style that structures an application as a collection of loosely coupled services. In a microservices architecture, services should be fine-grained and the protocols should be lightweight. The benefit of decomposing an application into different smaller services is that it improves modularity and makes the application easier to understand, develop and test. It also parallelizes development by enabling small autonomous teams to develop, deploy and scale their respective services independently.\[1\] It also allows the architecture of an individual service to emerge through continuous refactoring.\[2\] Microservices-based architectures enable continuous delivery and deployment.\[3\]
Nanoservice

Responsible for a single, non-trivial, business-understandable task

Smaller than a microservice, larger than a function

 Likely to call other nanoservices, to offer rich functionality

The ideal sized thing to release (or MVT) in a CD environment

Example:
  – Get customer order history
  – Place order
Merkel 'wins fourth term' in Germany

The chancellor is re-elected but right-wing nationalists make historic breakthrough, exit polls say.

24m Europe

- Why this vote matters
- Germany's next generation
- Election marks large shift

Defiance after Trump urges NFL boycott

Players kneel in protest during the US anthem as the president's remarks are strongly condemned.

2h US & Canada

Corbyn pushes back on EU single market

The Labour leader says staying a member could limit the government's powers to invest in industry.

1h UK Politics

Moeen hits thrilling ton in England win


5m BBC Sport

Man clings to train's windscreen wiper

A man's been arrested after apparently riding a 110mph train in Perth by clinging on at the back.

3h Australia

Oscar winner 'cut herself daily'

The Academy Awards have been dominated by Weinstein, and the #MeToo movements.

4h France

Uber seeks talks to renew London licence

5h Business

Johnson 'did not change Brexit policy'

Missing serviceman's family retrace steps

1h Suffolk

'Flammable material' sparked M3 closure

1h Hampshire & Isle of Wight

Man falls 50ft down disused mineshaft

1h Cornwall

Olympic champion 'cut herself daily'

http://www.bbc.co.uk/news
Architecture

Developer ➔ Nanoservice store ➔ Nanoservice platform

User ➔ Nanoservice platform

Nanoservice platform ➔ Content cache
Nanoservice platform ➔ Metadata store
Nanoservice platform ➔ Queue

Data (APIs etc.)
Each nanoservice is developed & managed independently
Nanoservices: advantages

Great ‘Lego bricks’ for sharing

Easy to understand concepts, reduces barrier-to-entry

*Should* be good value (sharing infrastructure, highly cacheable)

Looks after much of the operational side (scaling, resilience, monitoring, etc.)

Fast deployment

→ Greater development speed
Challenges

Single point of failure

Must be conservative, isolate and limit

Can be restrictive

Not super-fast

Needs great tooling to manage and understand usage
Pros:
- Multiple teams ✓
- Multiple technologies ?
- Smaller and safer deploys ✓
- Shareable & flexible ✓
- More scalable ✓

Microservices

Cons:
- Maintenance overhead
- Comms overhead
- Multi-version challenges
- Harder to refactor
- Cost
Serverless

AWS Lambda functions

Azure functions

Google Cloud functions

Why should they have all the fun?

The world needs powerful serverless platforms.
Serverless will fundamentally change how we build business around technology and how you code. …

You thought devops was big but it’s chicken feed compared to this…

…expect to see the whole world being overtaken by serverless by 2025

Simon Wardley, in *Why the fuss about serverless?*
https://www.linkedin.com/pulse/why-fuss-serverless-simon-wardley
Is nanoservices a paradigm for future serverless architecture?
It’s not just us

https://medium.com/netflix-techblog/developer-experience-lessons-operating-a-serverless-like-platform-at-netflix-a8bbd5b899a0
Nanoservices have brought the BBC:

Sharing

Flexibility

Efficiency

Velocity
Summary

The nanoservice pattern can get the best from CD, reusability, and flexibility.

As we free ourselves from the practical constraints of running systems, why not change our development approach to be more about smaller, shareable, understandable components?

Keep in touch...
Matthew Clark - @matthew1000 - matthew.clark@bbc.co.uk