

# Microservices

Oxford University  
Software Engineering Programme  
Sep 2015



© Paul Fremantle 2014. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).  
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.  
See <http://creativecommons.org/licenses/by-sa/3.0/>

# Agenda

- Why?
- History and evolution
- Architecture
- Pros and Cons
- More resources



# Microservices

- Building a single app from multiple services
  - Each service in its own process
  - Lightweight communications between each other
  - Usually HTTP but not necessarily



# Microservices Characteristics

## (Martin Fowler)

- **Componentization**
  - Replacability
- **Organisation**
  - around business capabilities instead of around technology.
- **Smart endpoints and dumb pipes**
  - explicitly avoiding the use of an Enterprise Service Bus (ESB)
- **Decentralised data management**
  - with one database for each service instead of one database for a whole company.
- **Infrastructure automation**
  - with continuous delivery being mandatory.

<http://martinfowler.com/articles/microservices.html>



© Paul Fremantle 2014. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).  
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.  
See <http://creativecommons.org/licenses/by-sa/3.0/>

You build it you run it  
Amazon story  
2001

Exactly equal to Microservices!

(this isn't new!)



© Paul Fremantle 2014. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).  
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.  
See <http://creativecommons.org/licenses/by-sa/3.0/>

# Benefits of Microservices

- Independent organization makes it easier for developers
  - Even if you are the only developer!
  - Simple code
  - Simple test cases
  - Simple scaling
  - Faster to build, deploy and test



# Microservices deployment model

- Increasingly fitting with “containerisation”
  - Docker
  - CoreOS
  - Kubernetes
  - Etc
- Container model is lightweight virtualization with each “VM” running a single process



# Real world examples

- The previous case studies are in many cases microservices
  - eBay, Netflix, Amazon
  - Many more out there and growing rapidly



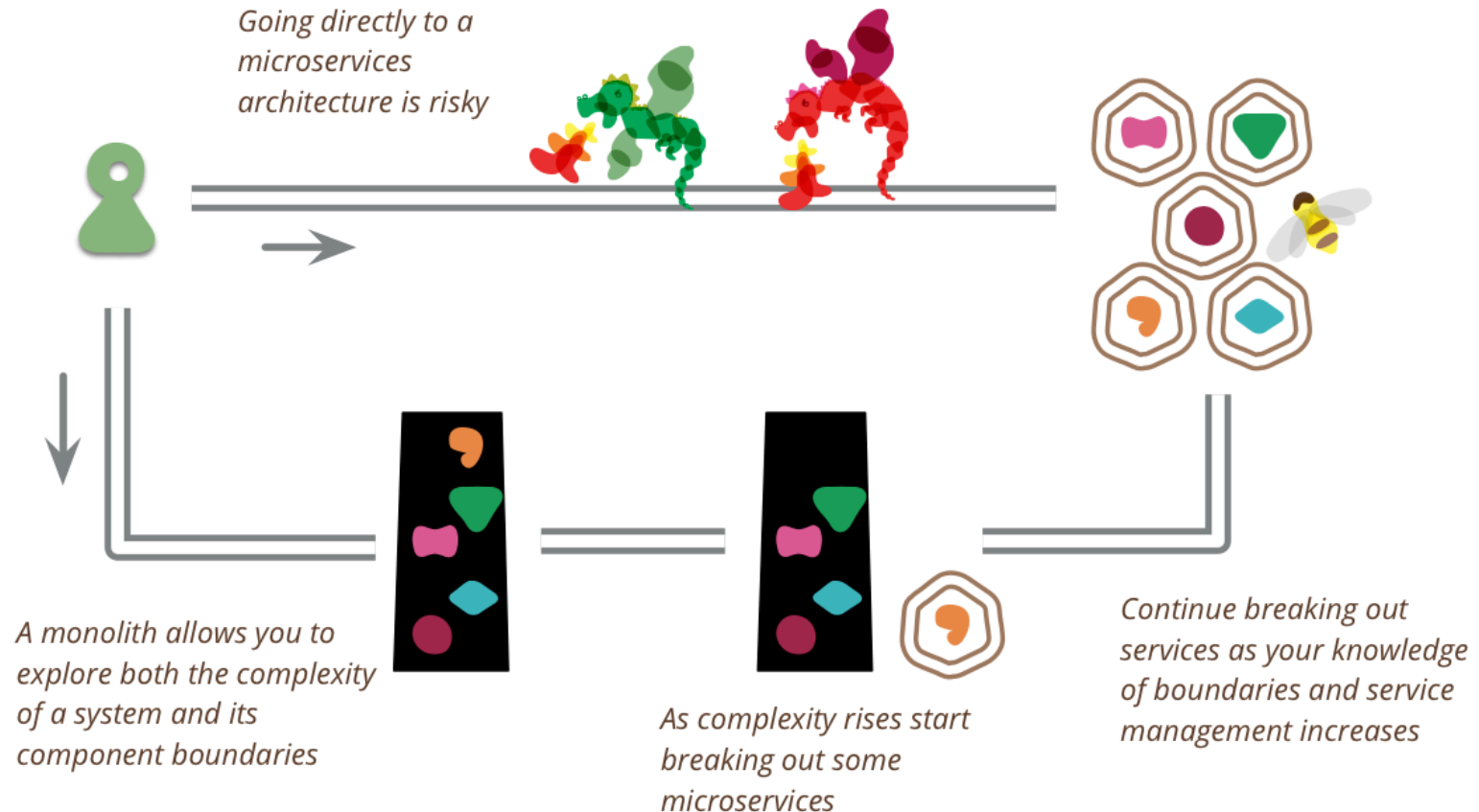


# Polyglot

- Microservices can be built in multiple languages
  - Hackathon last year I built a simple app
  - Node, Python and MQTT
  - One day's effort
  - Microservice architecture
  - <http://pzf.fremantle.org/2013/12/commshack.html>



# Start with a Monolith?

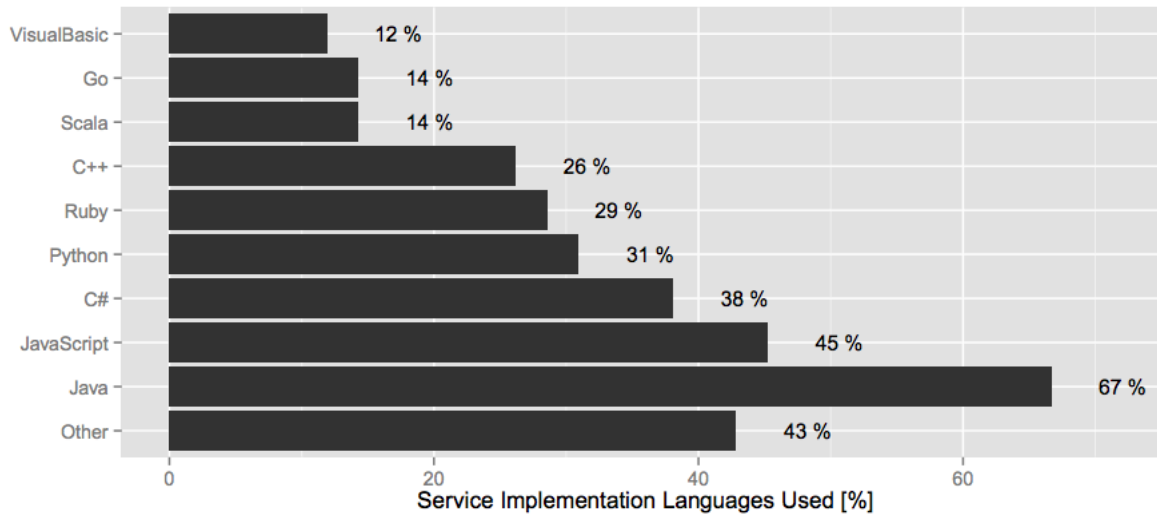


<http://martinfowler.com/bliki/MonolithFirst.html>

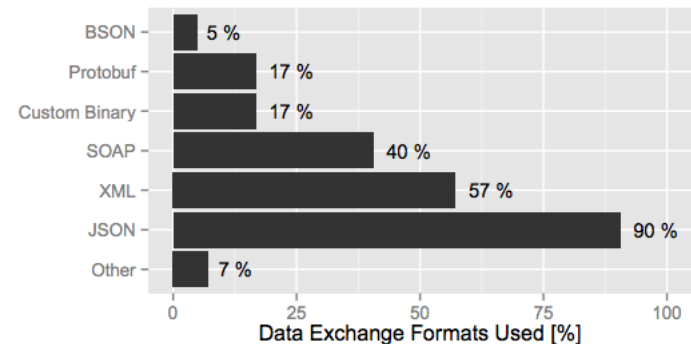


© Paul Fremantle 2014. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).  
Licensed under the Creative Commons 3.0 BY-SA (Attribution-ShareAlike) license.  
See <http://creativecommons.org/licenses/by-sa/3.0/>

# What are services like in reality?



Most services (51%) were 1,000-10,000 LoC  
Only 3% of services in the survey were <100 LoC  
43% 100-1,000 LoC



All the Services Large and Micro: Revisiting Industrial Practice in Services Computing

<https://peerj.com/preprints/1291.pdf>



© Paul Fremantle 2014. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).

Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.

See <http://creativecommons.org/licenses/by-sa/3.0/>

# Cons!

- Debugging
- Deployment and devops
- Operations overhead
- Implicit interfaces and contracts
- Latency
- Transactions
- Etc.!



© Paul Fremantle 2014. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).

Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.

See <http://creativecommons.org/licenses/by-sa/3.0/>

# Smart endpoints and dumb pipes

- Microservices are based on the idea of simple RESTful APIs directly implemented
- Need to manage contracts cleanly and carefully
- ESB is not part of this architecture
  - But an API Gateway might be?



# API Gateway and Microservices

- Versioning
- Single URI structure out of many independent backends
- Contracts and documentation
- More discussion later



# Resources

- <http://www.slideshare.net/chris.e.richardson>
- <http://martinfowler.com/articles/microservices.html>
- <http://www.thoughtworks.com/insights/blog/microservices-nutshell>

