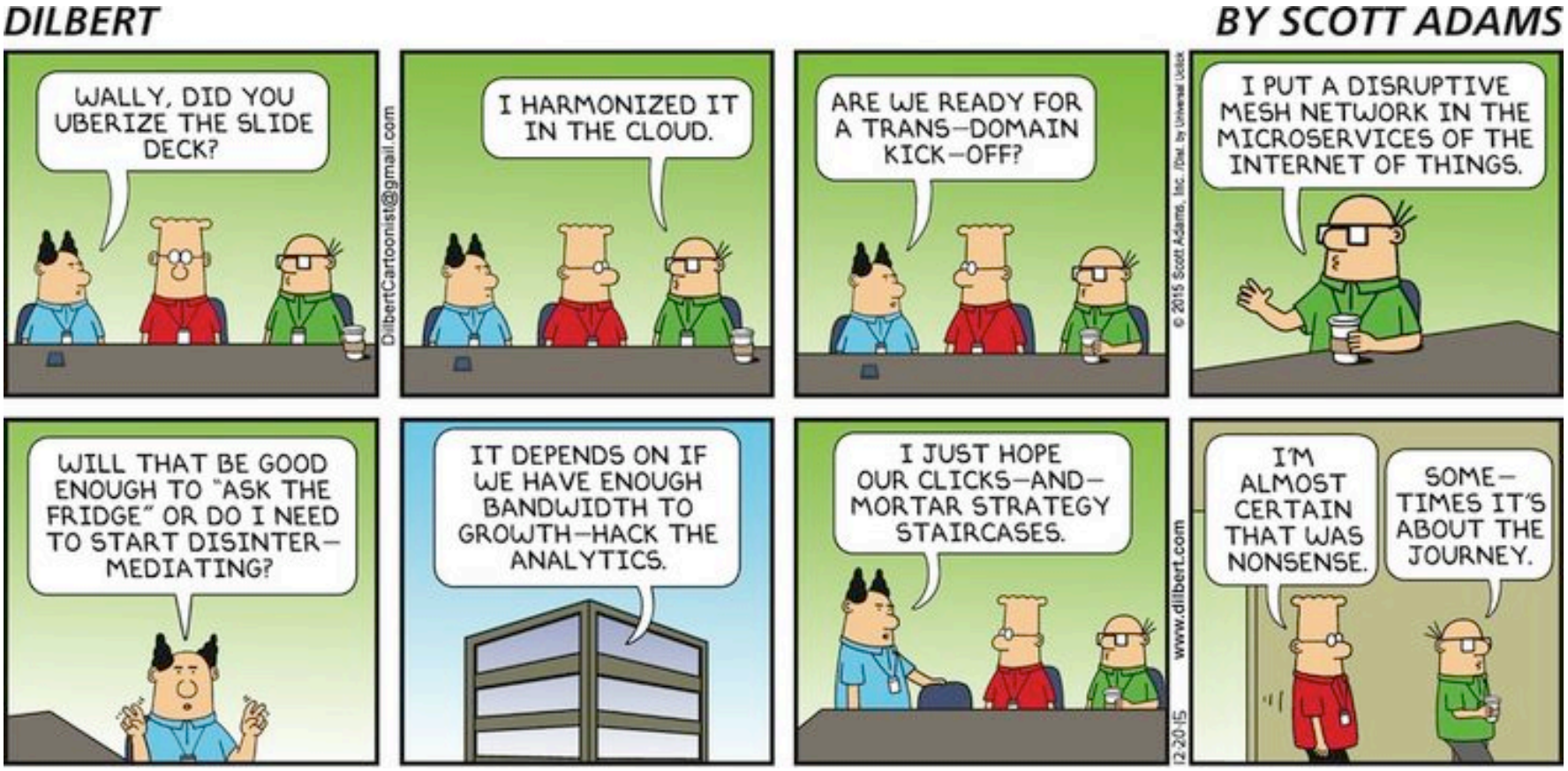
Introduction
Service Oriented Architecture

Oxford University
Software Engineering
Programme
June 2016

Introduction

- Aims
- Pre-requisites
- Contents
- Connections
- Resources
- Rules of Engagement
- Introductions



Apologies for the Jargon

- There is a lot!
  - Microservices, SOA, DevOps, REST, SOAP, WSDL, Swagger, JSON, XML, OAuth2, TLS, etc
  - Please ask if I fail to explain an acronym

Aims

- To understand:
  - Benefits and challenges of SOA
  - Services, Microservices and APIs
  - Security models
  - Mediation, Composition, Governance
- Implementation of
  - SOAP and REST based services
  - Microservices
  - ESB and BPMN flows
  - OAuth2 and SSL secured services
  - API Gateways and clients

Pre-requisites
(Some familiarity required)

- Languages: Java, Node, Python
- Data formats: JSON and XML
- Tools: Unix shell, Eclipse, Text editors

Contents

- Overview and course outline
- Case studies and motivations
- SOAP and WSDL
- SOAP Implementation technologies
- REST introduction
- REST example flows
- Advanced REST
- Microservices architecture
- Deployment, DevOps, containers and cloud-native applications
- Integration and ESBs
- Security
- API and API Management
- Orchestration and Choreography
- Governance
- Overview, futures, recap

Practicals

A. My aim is to have more practicals than everyone can manage:
   - Some people finish early, so there are extensions and bonus practicals for them.
   - You might wish to do more at home?!?

B. The practicals are quite directive to start with:
   - This is a complex area with a lot to cover.
   - Extensions are more freeform.
   - You need to think and not just do as I say to get the most out of them.

C. The practicals are mainly re-written from the previous run:
   - I don't know how long they will take you
   - There may be bugs
   - I may be wrong about #A!

"This really is an innovative approach, but I'm afraid we can't consider it. It's never been done before."

Practicals

- Basic HTTP server and client
- SOAP server and client
- HTTP service in Java
- Evolving the Richardson Maturity Model towards a RESTful service
- Microservice and Docker deployment
- ESB flows
- SSL and OAuth2 security
- API Management and Analytics
- BPMN workflows
  - Plus some bonus exercises

Resources

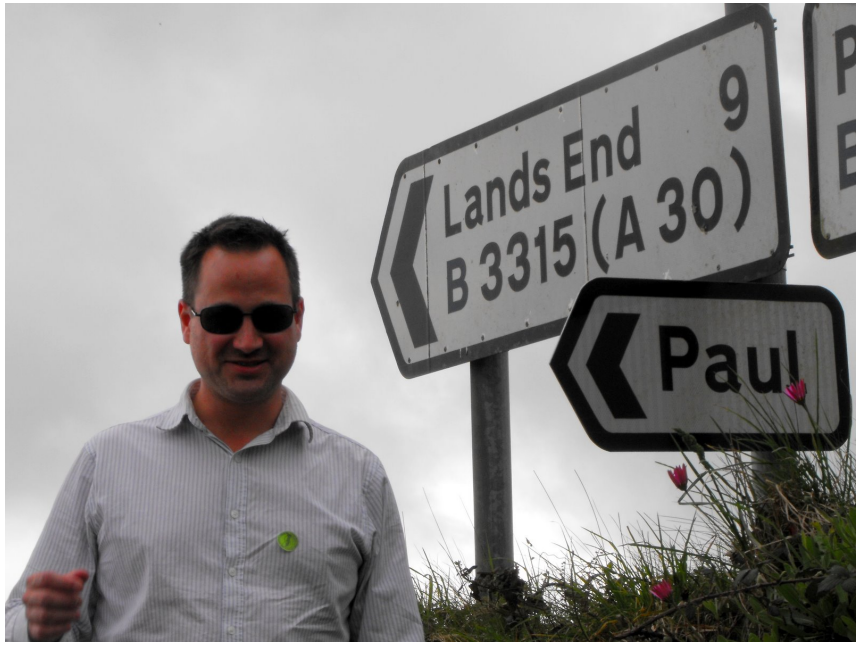
- Weerawarana et al, *Web Services Platform Architecture,* (Pearson, 2005)
- Erl, *SOA* (Prentice-Hall, 2005)
- Richardson and Ruby, *RESTful Web Services* (O'Reilly, 2007)
- Webber et al, *REST in Practice* (O'Reilly, 2010)
- Fielding, *Architectural Styles and the Design of Network-based Software Architectures,* (University of California, 2000)
- Various W3C, OASIS, IETF, OMG standards

Rules of Engagement

- Ask questions as we go along
  - We will “park” any that are better answered later
  - **Don’t wait till the end to ask or raise concerns**
- Timings are flexible
- Please keep mobile phones silent or better still turned off
- If you have improvements or bug reports, please submit issues or pull requests:
  - https://github.com/pzfreo/ox-soa2/issues/new

Paul Fremantle

- CTO and Co-Founder of WSO2
- Previously Senior Technical Staff Member, IBM WebSphere architecture
- Co-Chair Web Services Reliable eXchange at OASIS (WSRM)
- VP, Apache Synapse and Member of ASF
- MA in Maths and Philosophy
- MSc in Computation
- Studying for a PhD:
  - Security and privacy in IoT middleware



You?

Approximate Schedule

Monday	Tuesday	Wednesday	Thursday	Friday
Introduction	REST introduction	Microservices	Security	Design Exercise
Case Studies and motivation	REST example flow	Docker Introduction	Governance	History and futures Conclusions
SOAP and WSDL	Evolving REST practicals	DevOps, deployment, management	API Management	
SOAP Implementation	Advanced REST	Integration and ESBs	Composition	

Let's get started