



# Microservices Architecture

Copyright © The Open Group 2016

THE *Open* GROUP

# Microservices Architecture

- Are microservices an evolution of SOA?
- How should they be used in enterprise architecture?

# Panelists



**Peter Maloney**  
Senior  
Engineering  
Fellow,  
Raytheon



**Som  
Balakrushnan**  
Consultant,  
Salesforce.com



**John Bell**  
Principal,  
Ajontech



**Ovace  
Mamnoon**  
Practice  
Principal, HPE

# History

- SOA
  - Control distributed computing costs by leveraging infrastructure built for the web
  - Increase agility by allowing more responsive change
  - Built on XML and HTTP
  - Prior alternatives (CORBA, DCOM) were complex and expensive
  - SOA is now perceived as complex and expensive

# Microservices

- Microservices Architectures were developed as a push back against the complexity of SOA. They have emerged from the lessons learned in real-world use. The idea is to focus on the single business function and create services that implement the operations required by that function

# Definition

- Microservices Architecture is a style of architecture that defines and creates systems through the use of small independent and self contained services aligned closely with business activities.
- Microservices Architecture is a subset of a full SOA architecture with the added constraints of service independence.

# Characteristics

- A Microservice is independent of other external services
- Supports elastic deployment
  - Resilient against failure
  - Dynamically scalable
- Supports parallel development and operations
  - Independence of development teams
  - Independence of deployment and governance
- Tends toward small modules, supported by small teams, and short, fast, less costly development times.

## An MSA...

- Service is independent of other services
- Uses this independence and the parallelism it permits to achieve architectural resilience and scalability
- Is constrained focusing on single responsibility per service
- Service is not comprised of other services due to the independence requirement



# Principles

- **Independence**
  - A Microservice is independent of other services
- **Single Responsibility**
  - A Microservice is focused on doing one thing
- **Self Contained**
  - Everything the Microservice requires is packaged with the service deployment unit

## Best to use when

- Rapid development is required
- New development
- No dependence on existing infrastructure
- Can support multiple parallel teams

## Also

- Supports technologies like node.js
- Used by companies like Netflix and Twitter
- May orchestrate at application layers or a higher services layer

# Questions

# Microservices Architecture

**Thank you!**