



Microservices and SOA

CON7364

Luis Weir
Oracle Ace Director
Principal, Capgemini

Robert Wunderlich
Sr. Principal Product Manager

September 22, 2016

September 18–22, 2016
San Francisco



Table of Contents

1 What is a Microservice

2 Microservices vs. SOA

3 Reference Architecture

4 Use Cases

A collage of three images. On the left, a close-up of a red umbrella with a textured surface. In the center, a person stands on a train platform at sunset, taking a photo with a camera. On the right, a group of people are holding up their smartphones to take pictures of something off-camera.

Introduction

Resume

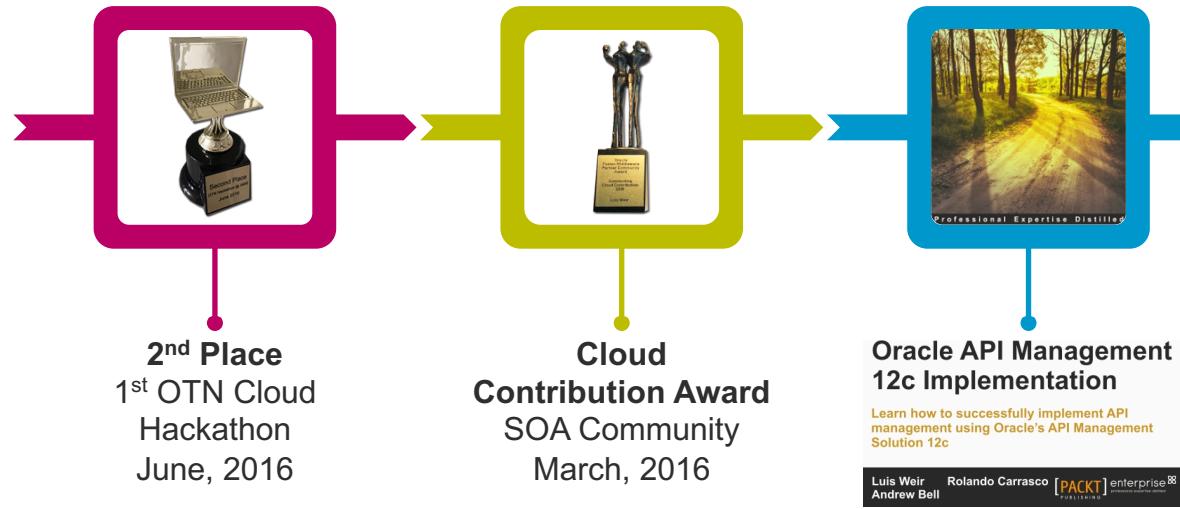


Luis Weir

Oracle Ace Director – Cloud Principal at Capgemini UK

I am an Oracle Ace Director, Cloud Principal and a Thought Leader specialised in Oracle Fusion Middleware & Oracle PaaS. With more than 15 years of experience implementing IT solutions across the globe, I have been exposed to a wide variety of business problems many of which I've helped solve by adopting SOA architectural styles such as traditional SOA, API management and now Microservices. My current focus is in assisting organisations define and implement solutions and strategies that can help them realise the benefits that such technologies have to offer.

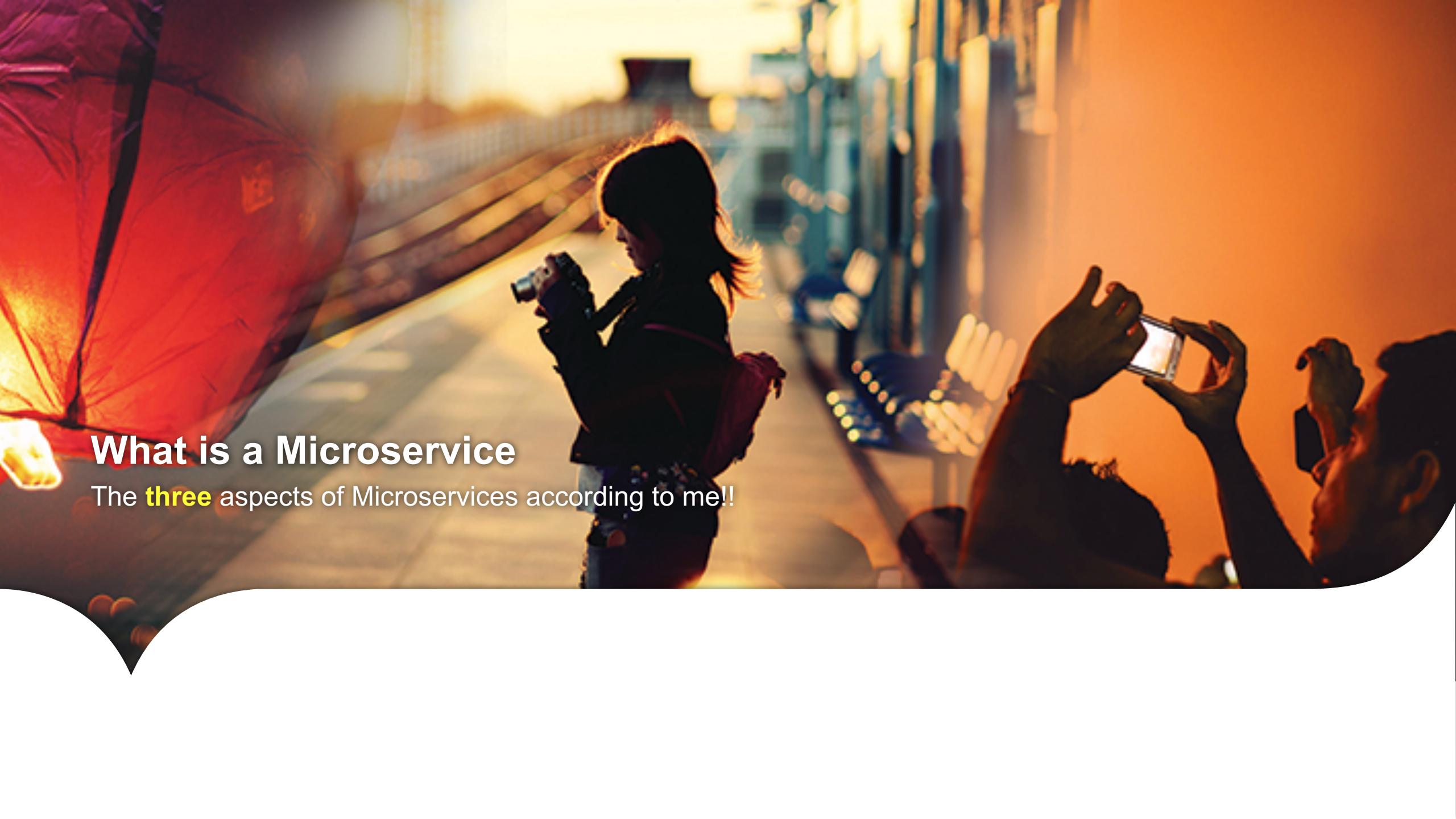
I am very passionate about technology. I have been the lead author of two books (Oracle SOA Governance 11g Implementation and Oracle API Management 12c Implementation), I am a regular blogger and speaker in major conferences and events. A well-known industry expert especially when it comes to Oracle middleware technologies I am also an OTN certified SOA black belt.



Latest Media:

- Oracle Magazine May/June 2016 (<http://bit.ly/1RTCAU3>)
- Systematic Approach for Migrating to Oracle Cloud SaaS (<http://bit.ly/1Xr6acs>)
- Oracle Magazine Jan/Feb 2016 (<http://ora.cl/Vhh>)
- API Management Implementation (<http://ora.cl/Gcw>)
- A Word About Microservices and SOA (<http://bit.ly/25Dk5go>)



A photograph of a woman with long hair, seen from behind, holding a camera and taking a picture of a large, glowing orange hot air balloon in the foreground. In the background, a train is visible on tracks under a warm sunset sky.

What is a Microservice

The **three** aspects of Microservices according to me!!

What is a Microservice?

“Loosely coupled service oriented architecture with bounded context”,

– Adrian Cockcroft, April 2015



adrian cockcroft
@adrianco



Following

@kellabyte @mamund I used to call what we did "fine grain SOA". So microservices is SOA with emphasis on small ephemeral components

RETWEETS

3

LIKES

4



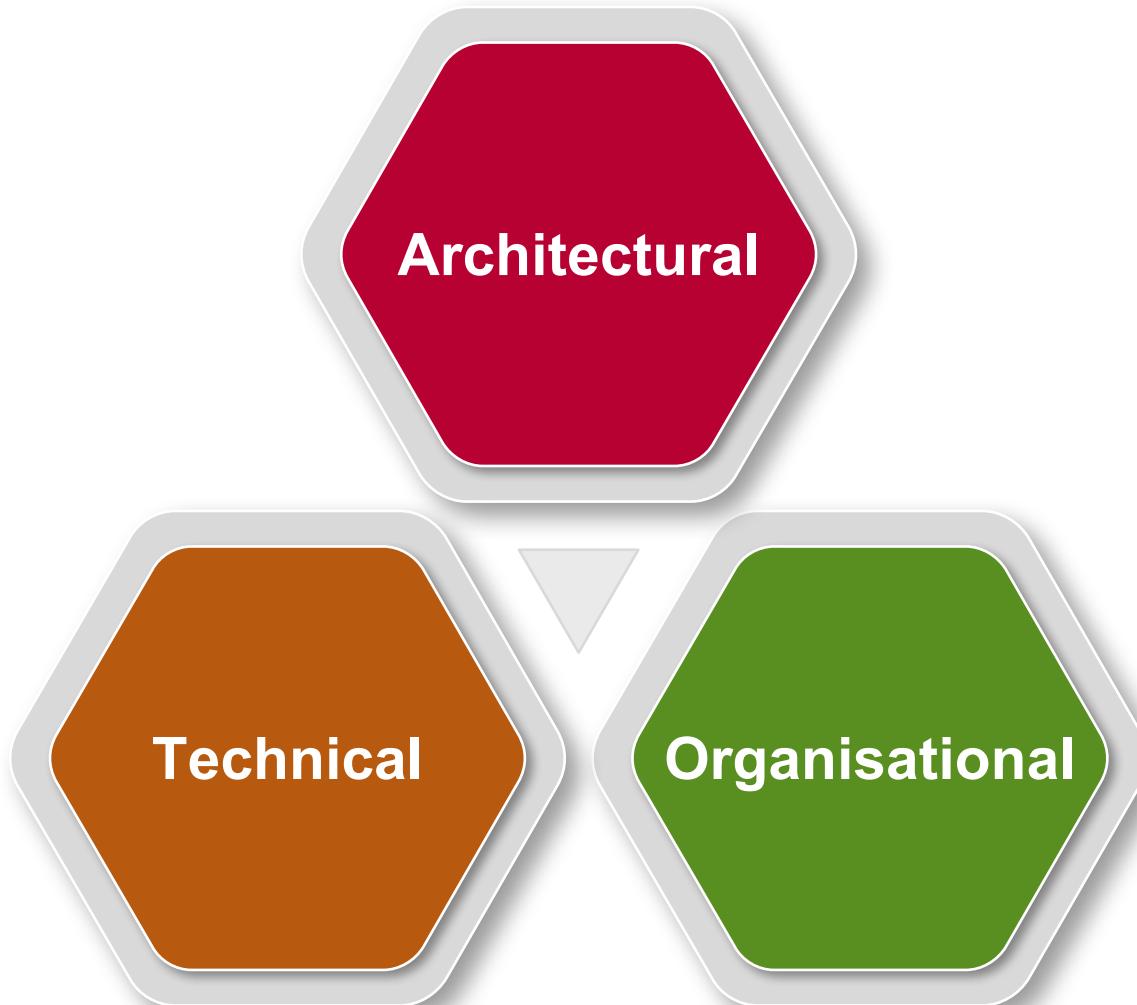
5:16 PM - 10 Dec 2014

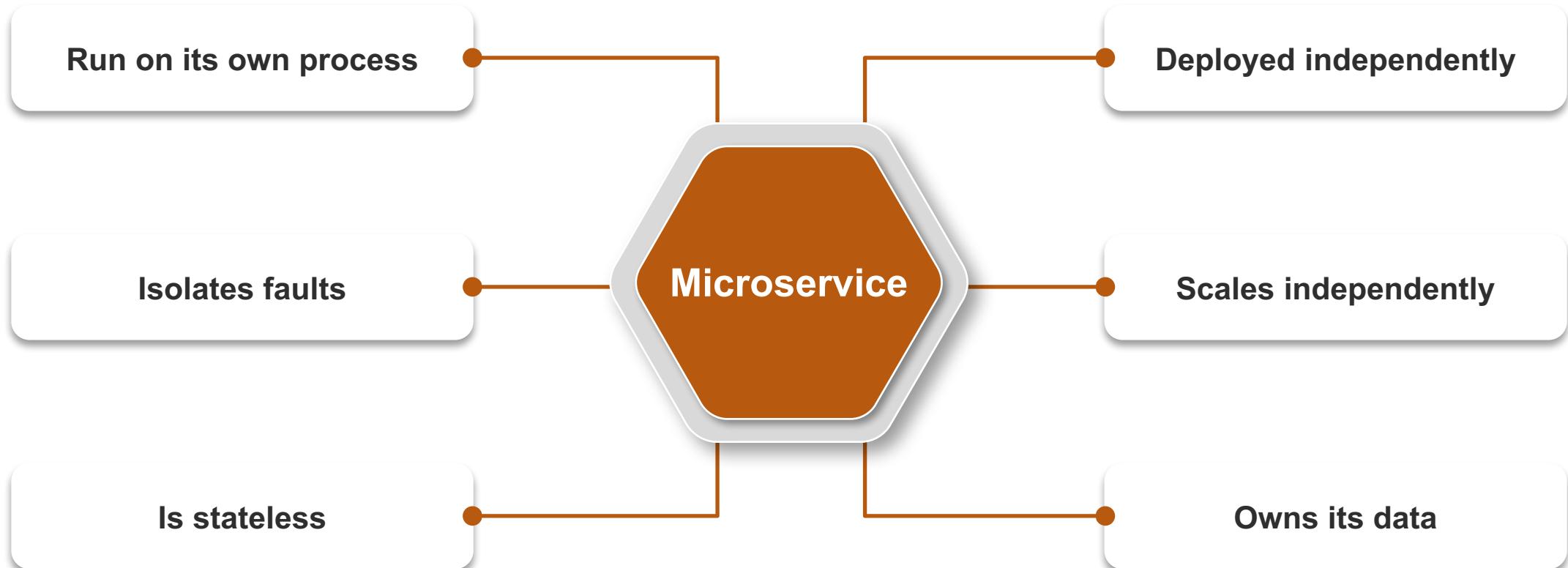


Functional decomposition of systems into manageable and independently deployable components,

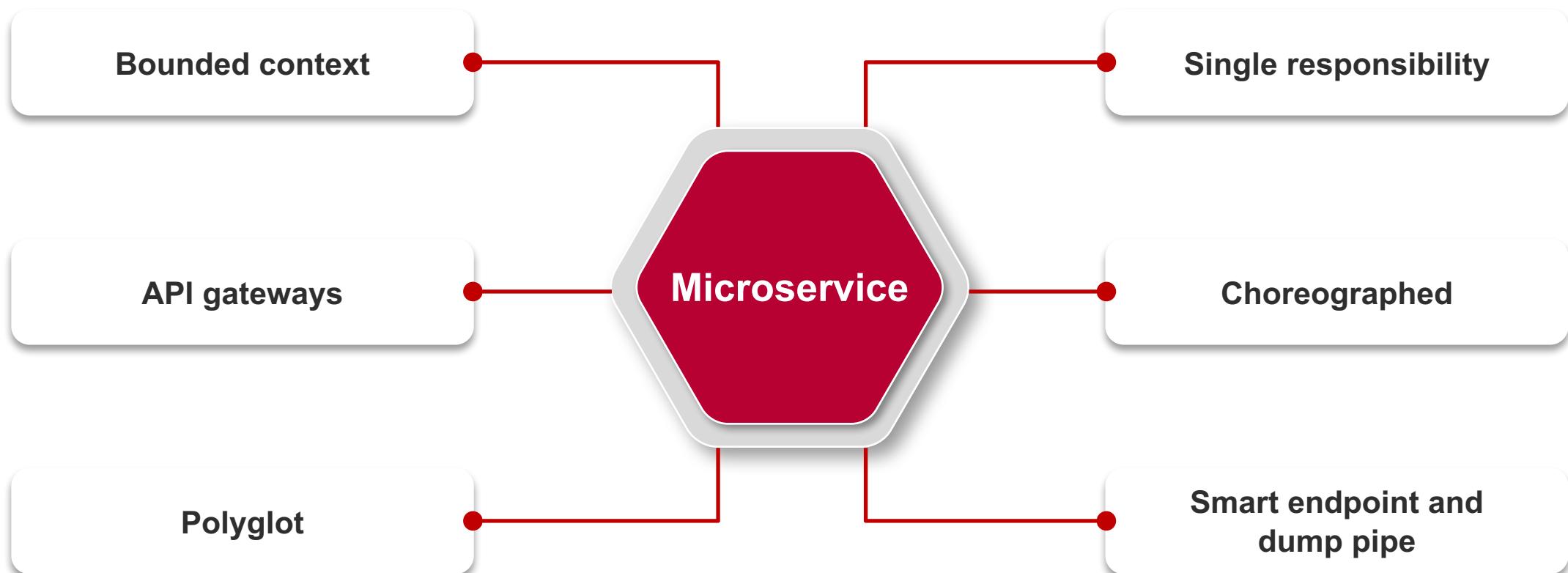
Microservice Architectures by Dr. Andreas Schroeder
(<http://bit.ly/1TOGZK8>)

The **three** aspects of Microservices Architecture (according to me!)

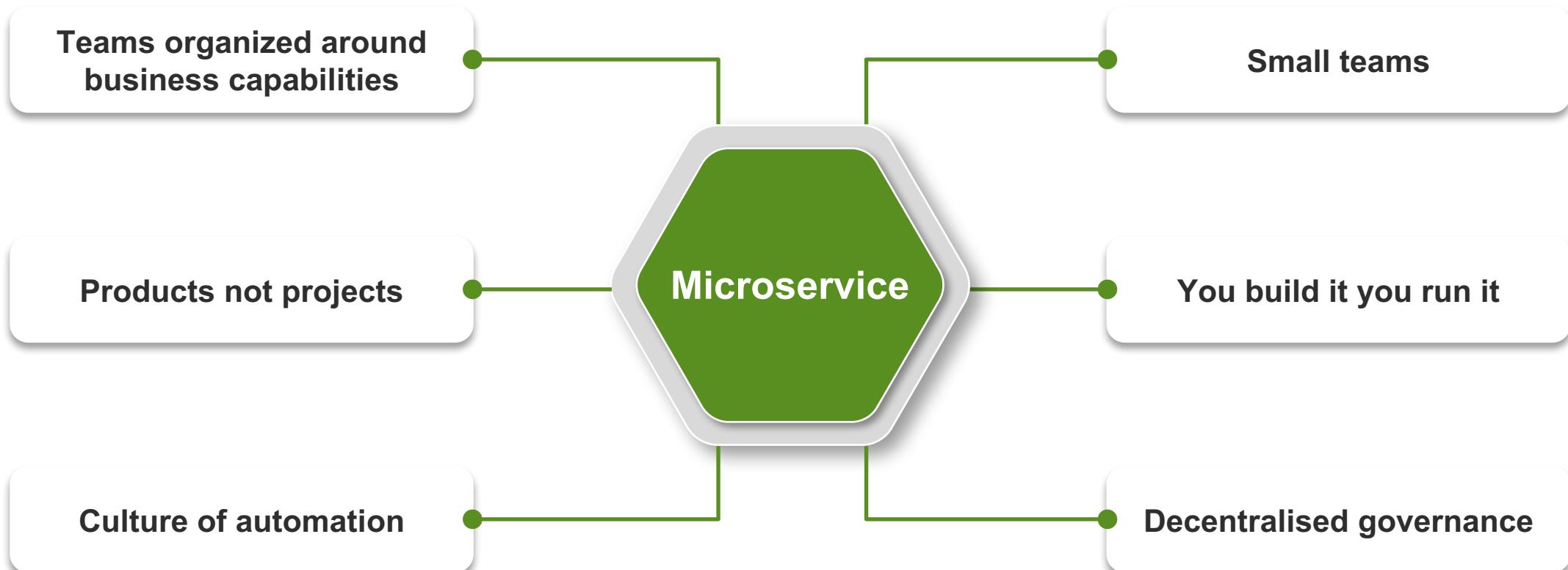




Architectural



Organisational





Microservice vs. SOA

Key differences and communalities

~~Microservices vs SOA~~ Wrong comparison... The difference lies in the realisation style to implement SOA

“The value of the term microservices is that it allows to put a label on a useful subset of the SOA terminology”,

Martin Fowler (minute 14), GOTO conference,
Berlin November 2014



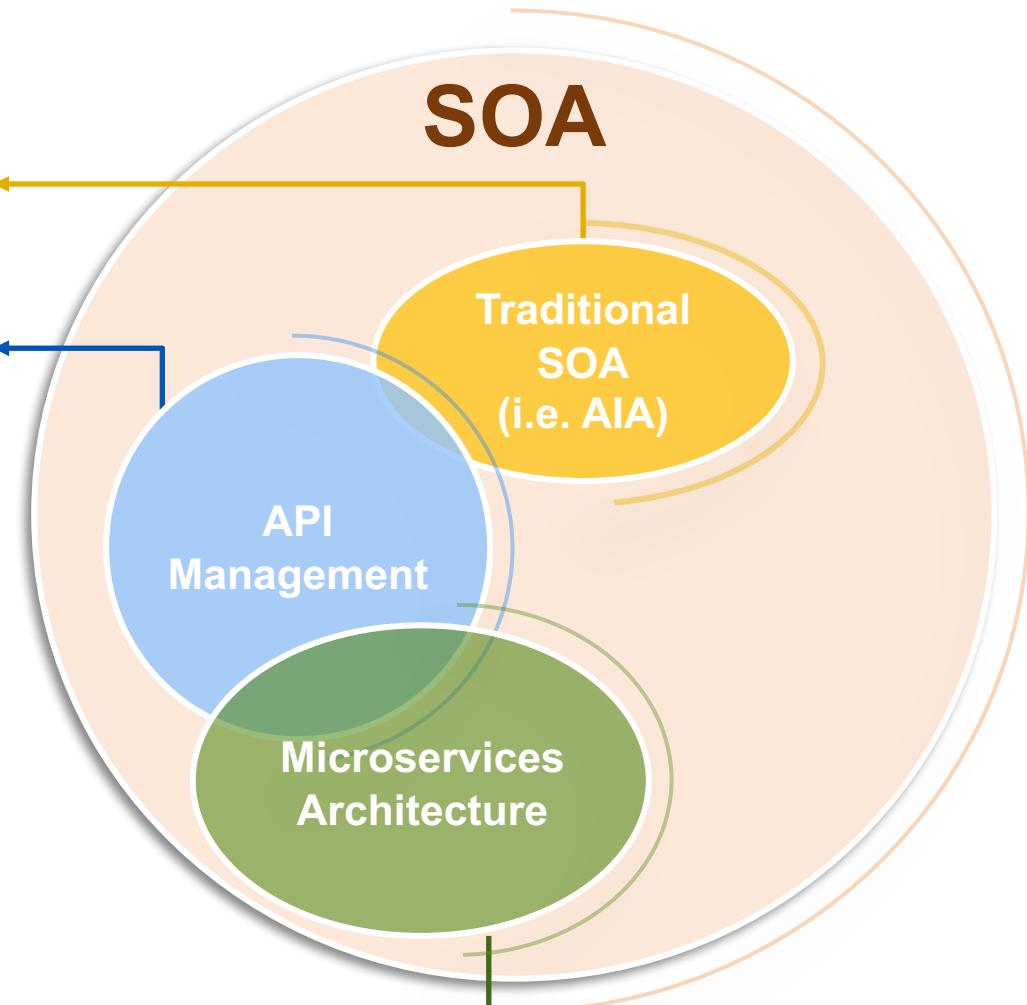
Inspiration from Martin Fowler's Microservices presentation at GOTO conference, Berlin November 2014 (minute 14)

<https://www.youtube.com/watch?v=wgdBVIX9ifA>

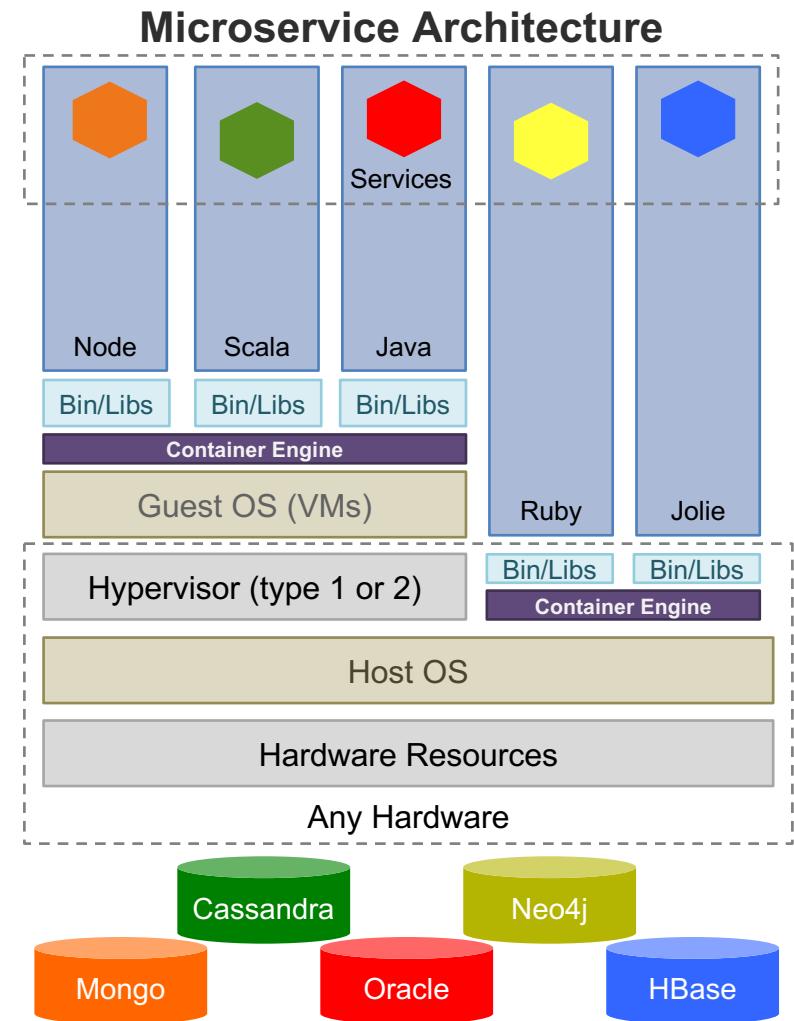
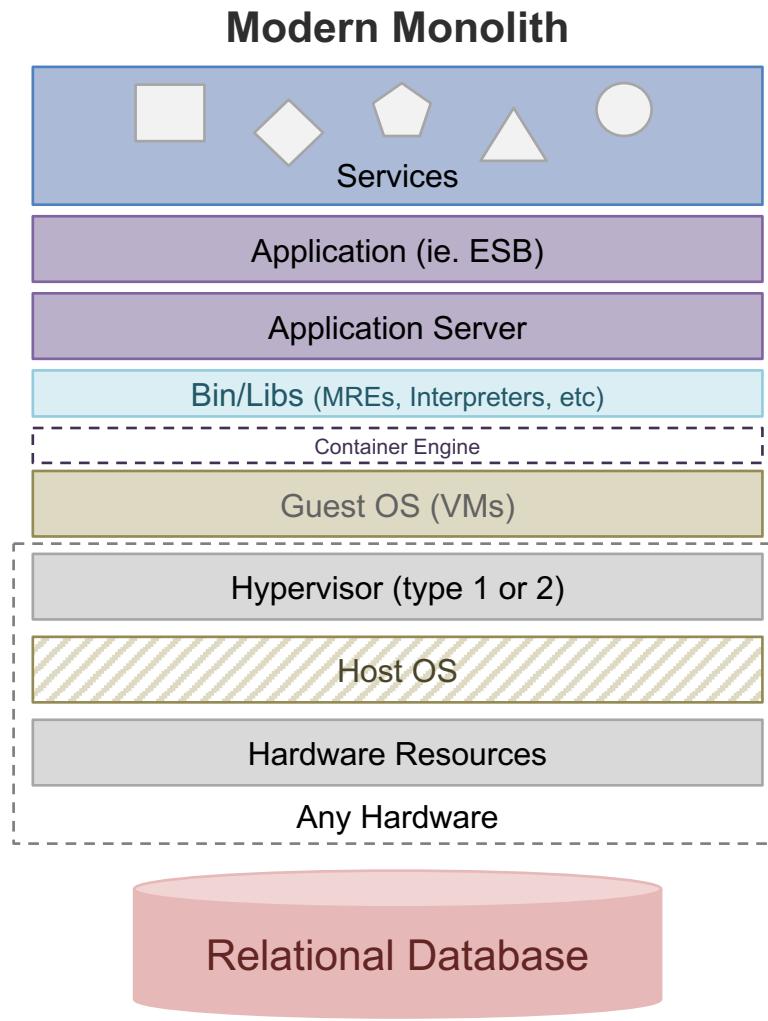
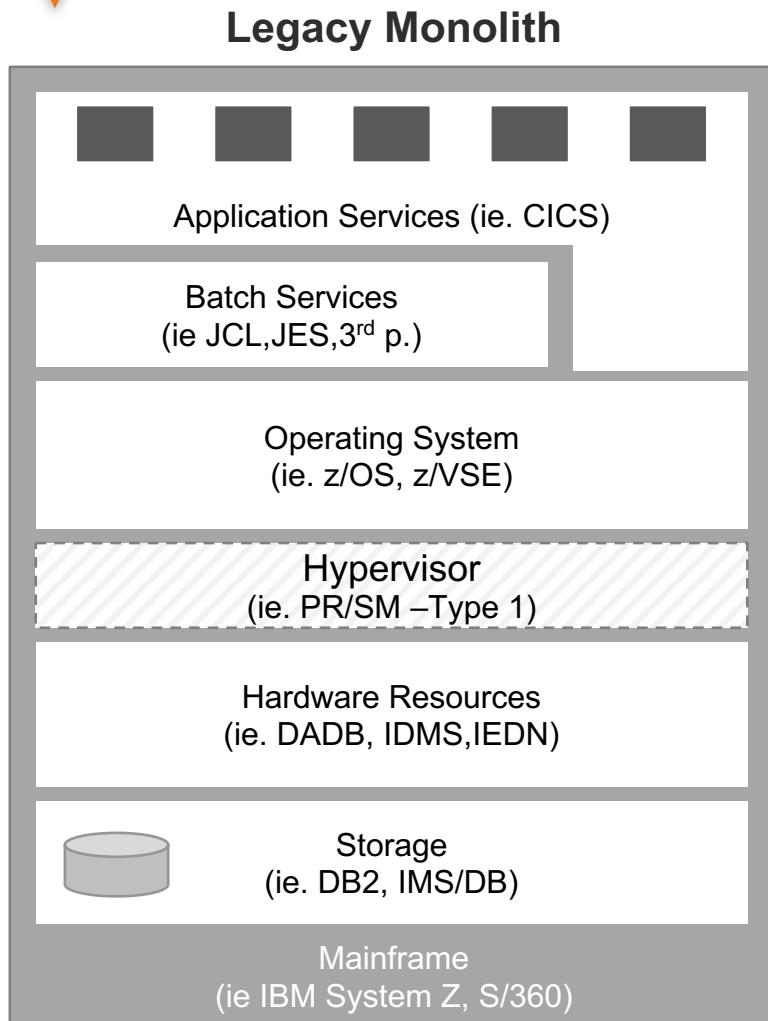
Typically adopted to deliver horizontal integrations

Best for vertical integrations

Not for integration.
Best for building modern systems



Microservices vs SOA – Technical Stack



Architectural

Pattern	Traditional SOA	MSA
Monolith pattern (http://bit.ly/1Gjr2Y0)	Yes	No
Polyglot Programming & Persistence (http://bit.ly/18BvDlj & http://bit.ly/1XYiak2)	Not traditionally (use of Suites)	Yes
API gateway pattern (http://bit.ly/1WTyNLJ)	Yes	Yes
Orchestration (http://bit.ly/1U0SWil)	Yes	No
Choreography (http://bit.ly/1ssALZQ)	No	Yes
Event Collaboration (http://bit.ly/25Dk7oE)	Yes	Yes
Canonical Schema (http://bit.ly/1r6KkfK)	Very common	No
Schema centralization (http://bit.ly/1sVlqkc)	Very common	No
Decouple Contract (http://bit.ly/1O8mVpm)	Yes	Could be....
Bounded Context (http://bit.ly/1o7AK8B)	Some times	Yes
Ubiquitous Language (http://bit.ly/1c8nXQe)	Some times	Yes
Bulkhead (http://bit.ly/1c8nXQe)	Not really...	Yes
Tolerant Reader (http://bit.ly/1aa4mr9)	Some times	Yes
Client-side Service Discovery (http://bit.ly/1OunUyq)	Initially only (service registry)	Recommended
Server-side Service Discovery (http://bit.ly/1X3RmzA)	Yes	Yes
ESB Pattern (http://bit.ly/1ZISKeT)	Yes	Across bounded contexts (dump pipe)

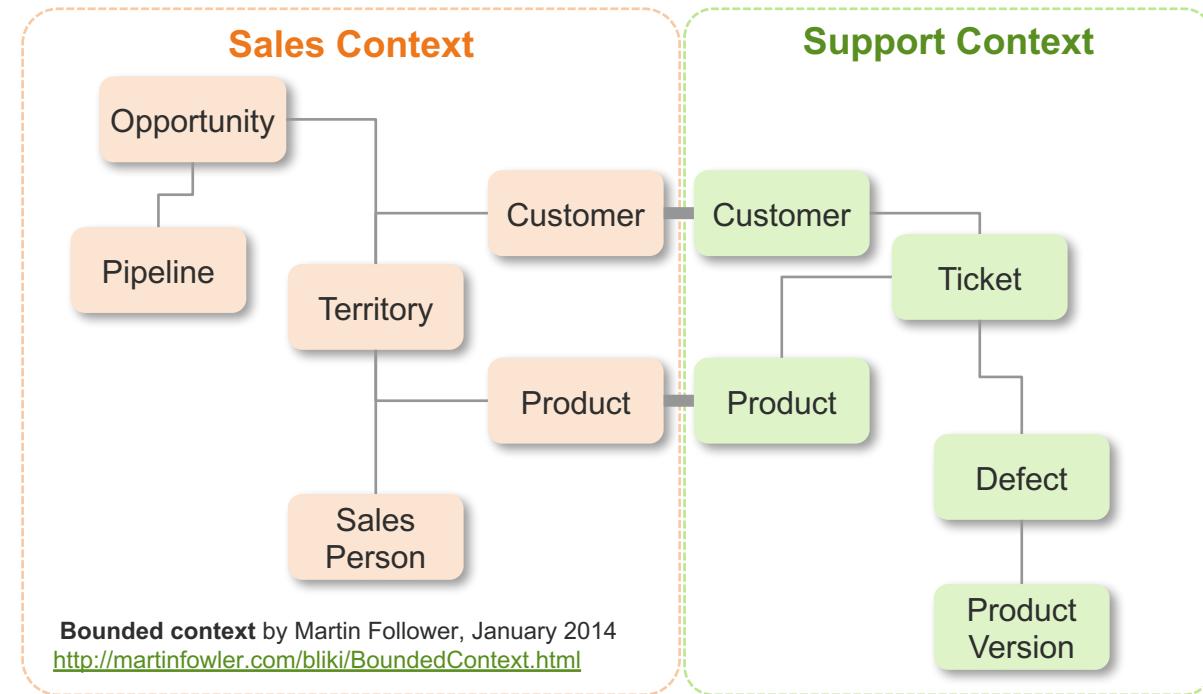
Choreography vs Orchestration – Which one is which?



Bounded Context

“Gather together those things that change for the same reason, and separate those things that change for different reasons” – The single responsibility principle by Robert C. Martin, November 2009, <http://bit.ly/1VDgw79>

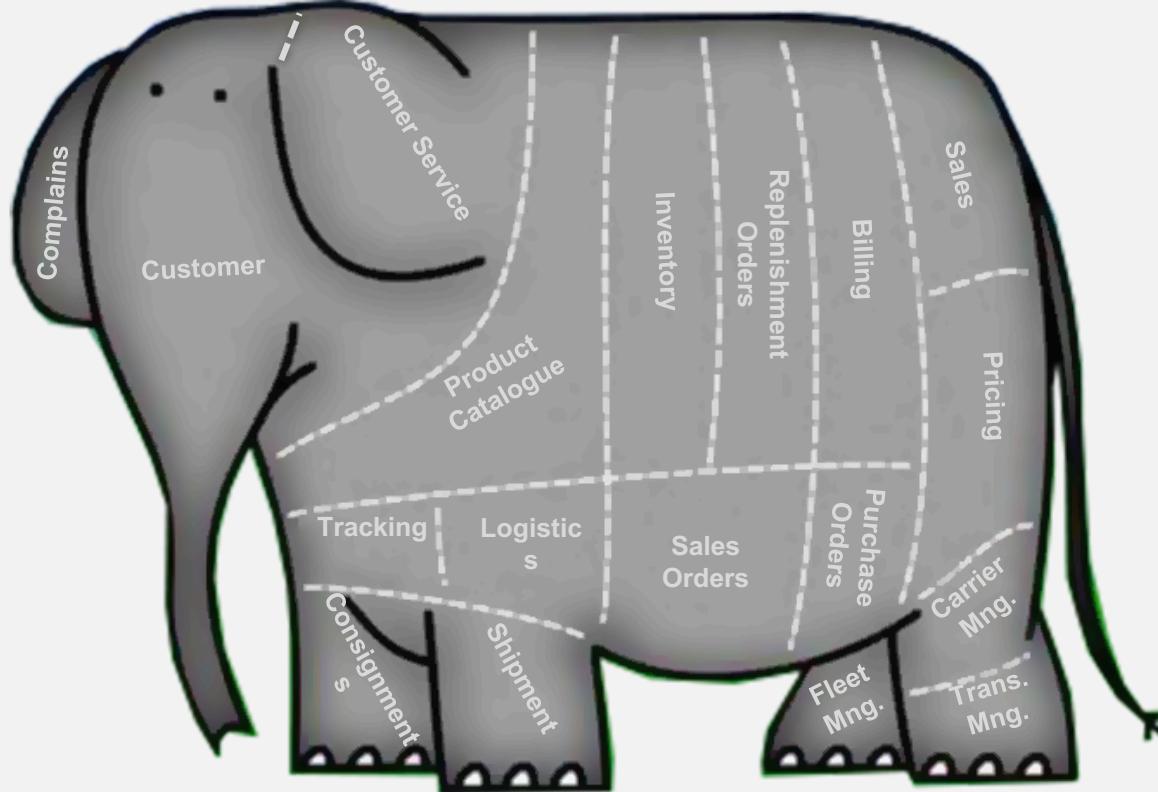
Use Bounded Context
to Separate Concerns



“Domain driven design (DDD) divides up a large system into Bounded Contexts, each of which can have a unified model – essentially a way of structuring Multiple Canonical Models.”

How to eat the Elephant?

One piece at the time!



Understand the problem. Slice and dice your elephant by defining boundaries in the business capabilities. Modernise one piece at the time. Starting small

Recommend

Domain Driven Design & Microservices by Eric Evans



Goto Berlin, Nov 2014

<https://www.youtube.com/watch?v=yPvef9R3k-M>

Principles of Microservices By Sam Newman



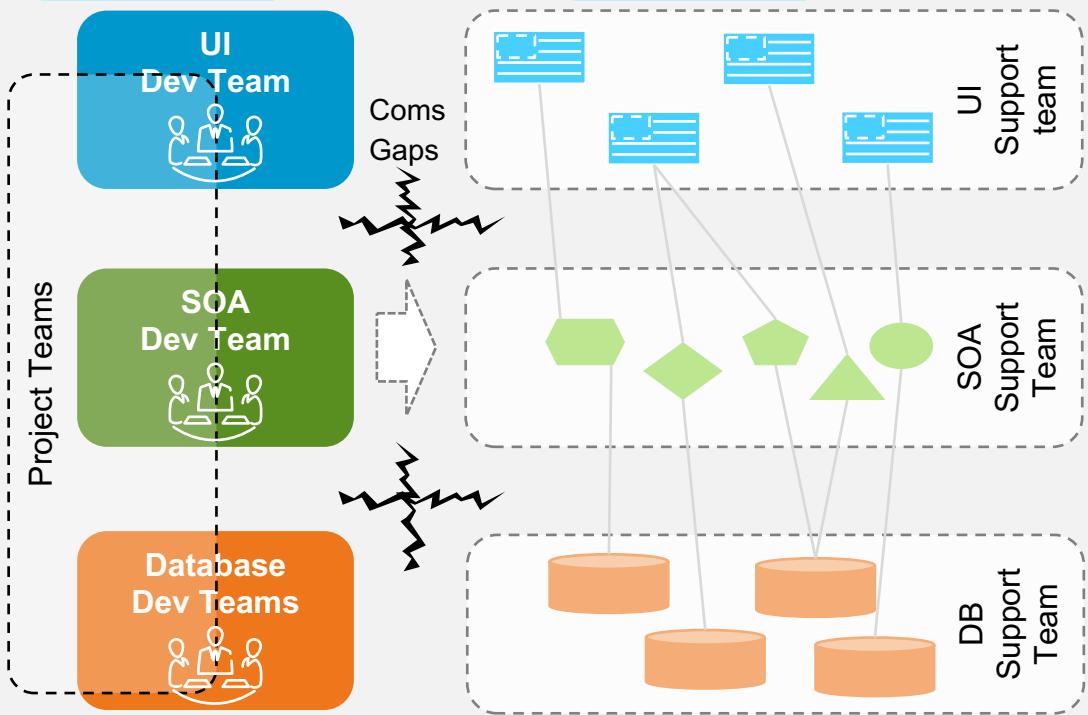
Devoxx Belgium, Nov 2015

<https://www.youtube.com/watch?v=PFQnNFe27kU>

Organisational

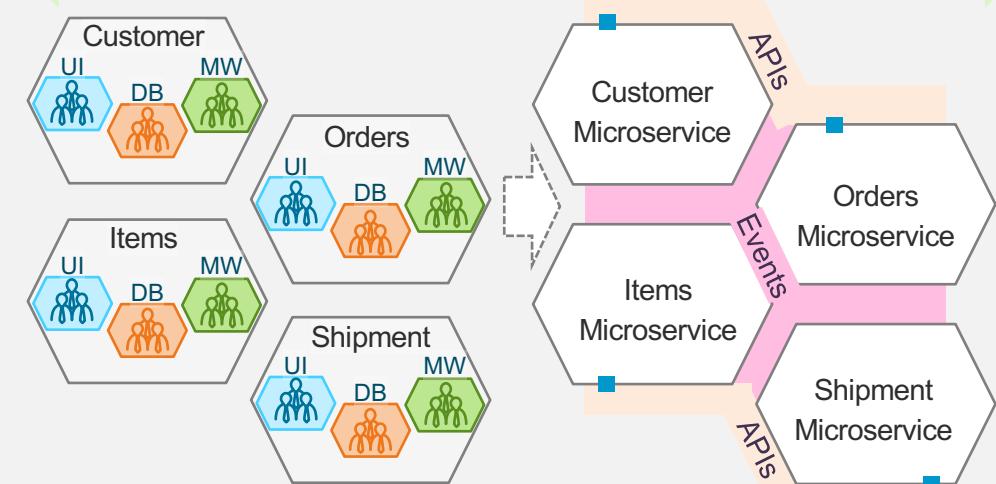
Traditional Operations Model

Development and support teams organized by technologies resulting in siloes(Conway's law in action)



MSA Operations Model

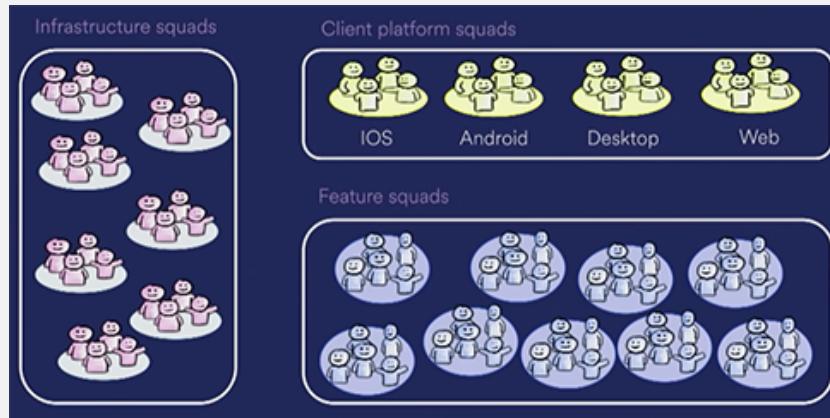
Multi-disciplinary [small] teams organized by business capability resulting in modular systems



DevOps / Continuous Delivery



Modeling Microservices at Spotify with Petter Mahlen



At a microservices talk in March in Sweden, Petter Mahlen, Backend Infrastructure Engineer at Spotify, spoke to a packed house at Jfokus about microservices.

<http://tinyurl.com/msasspotify>

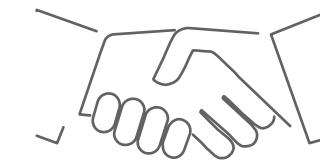


Reference Architecture

A reference architecture suitable for SOA 2.0:
Microservices, API Management and Integration

SOA 2.0 Reference Architecture

SYSTEMS OF ENGAGEMENT



Single Purpose APIs

Special Purpose APIs

Presentation APIs

Public [Consumer] APIs

Partner [B2B] APIs

SYSTEMS OF DIFFERENTIATION

Business APIs



API Applications



CX

HCM

SaaS

ERP

EPM

Legacy, etc

SYSTEMS OF RECORDS

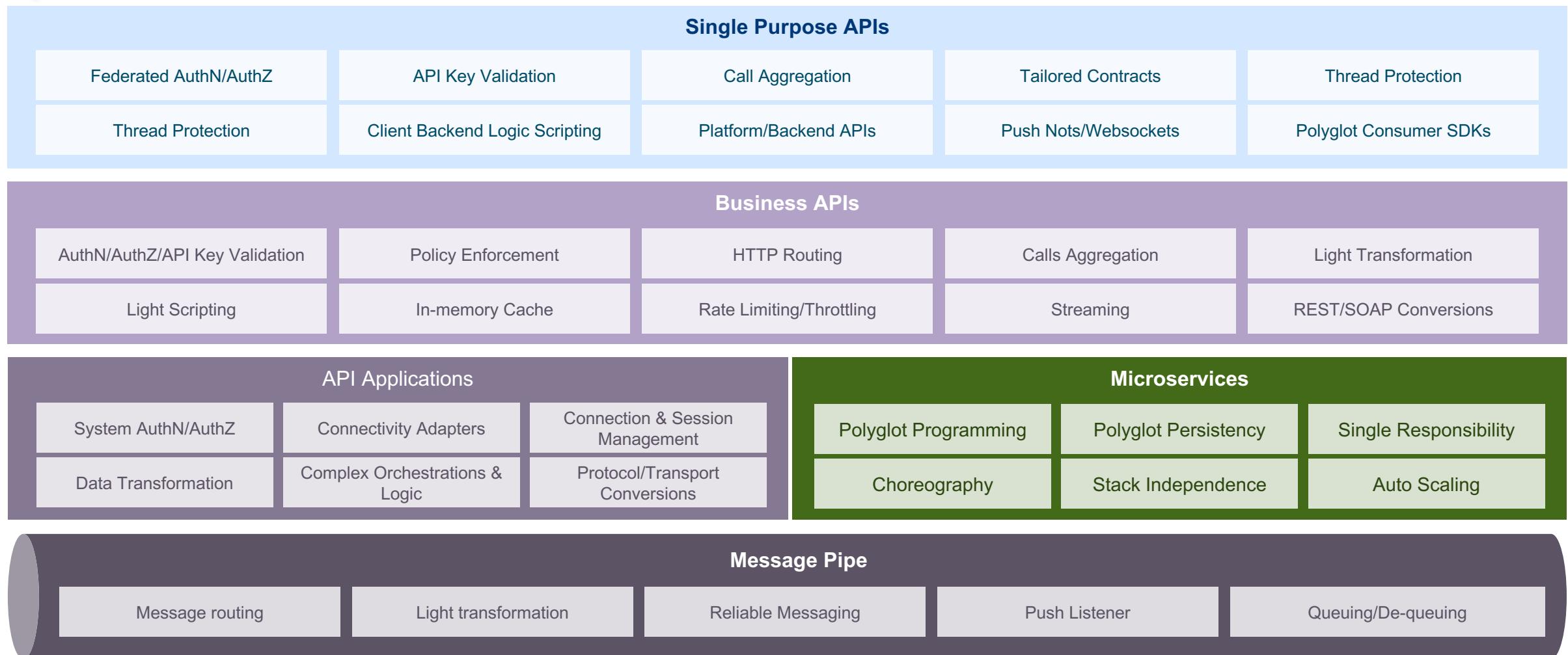
Microservices



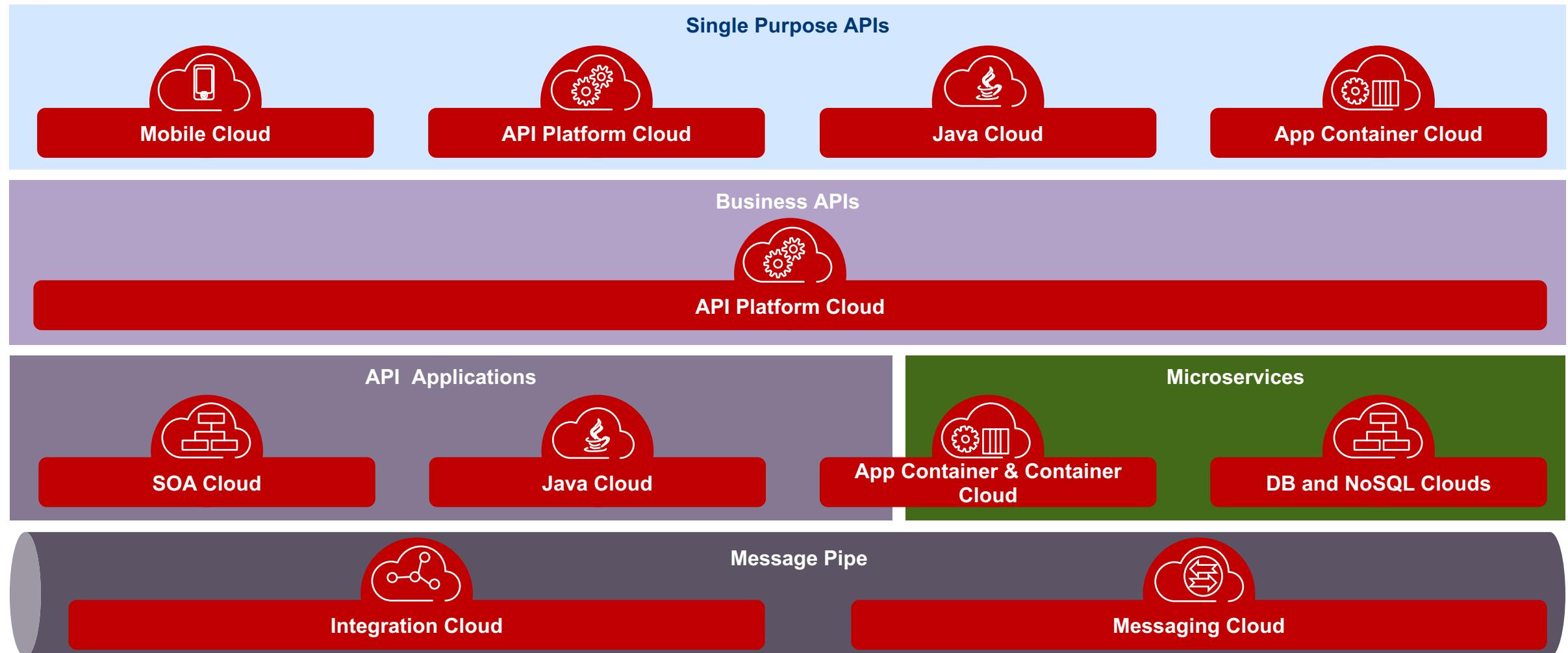
SYSTEMS OF INNOVATION

Message Pipe

SOA 2.0 Capability Model



SOA 2.0 In the Oracle Cloud PaaS

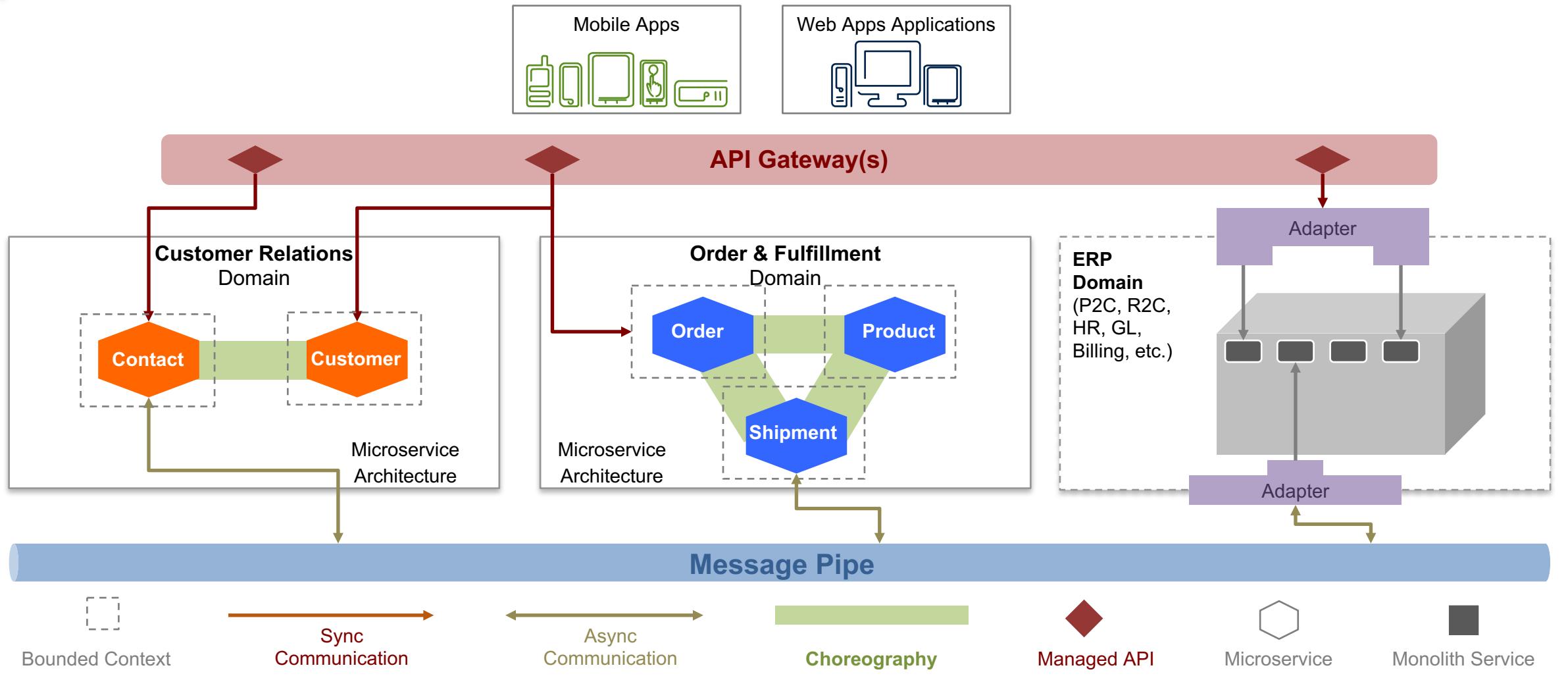




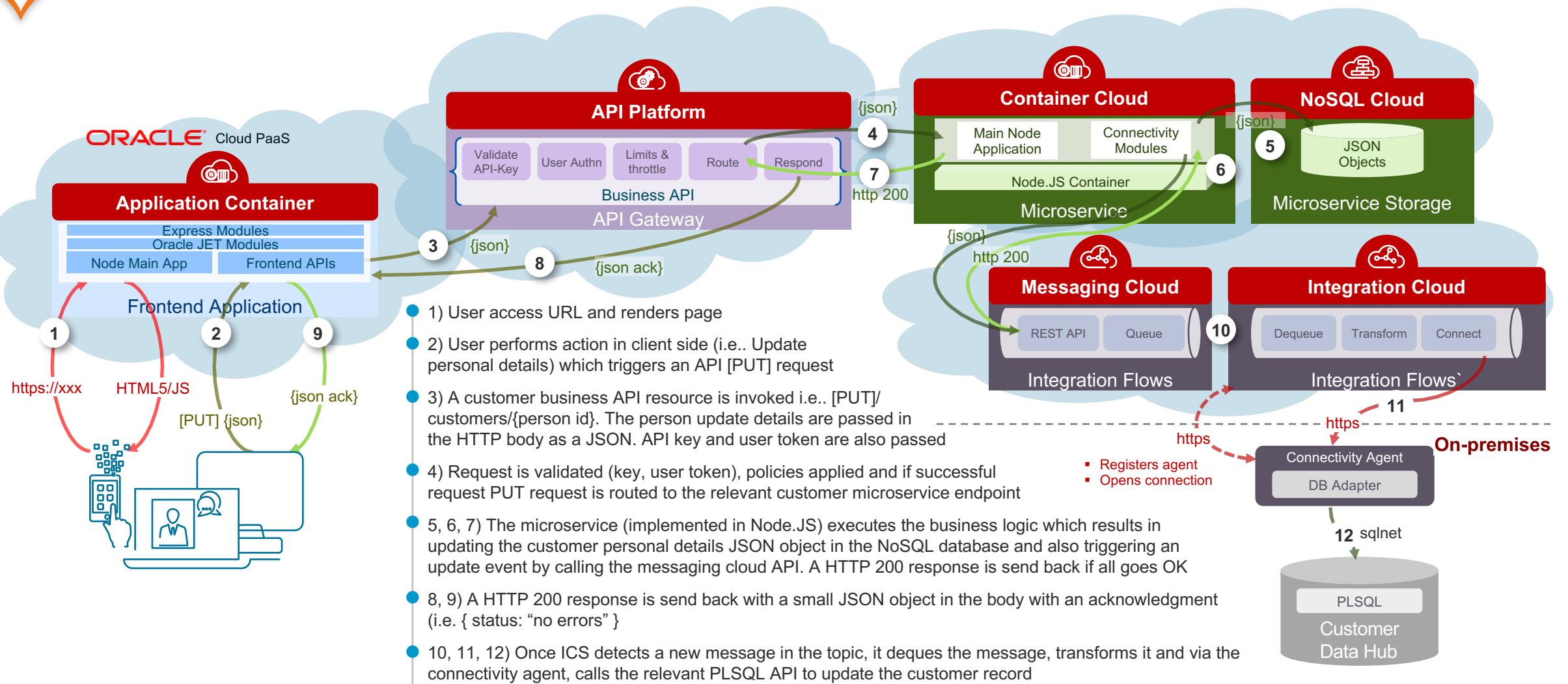
Use Case

Modern application based no Microservice Architecture with integration to on-premises legacy systems for synchronization

Conceptual Solution Architecture



Modern Application in Oracle PaaS





Conclusion

Why adopt Microservices?

Conclusion - Why adopt Microservices Architecture (I)



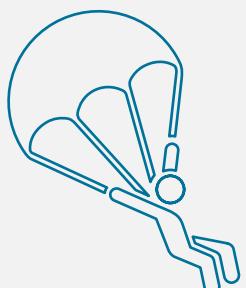
Modularity

Eat the elephant one piece at the time. Phased implementation approach. Starting small. Small teams owning full lifecycle of their piece (a business capability)



Segmented complexity

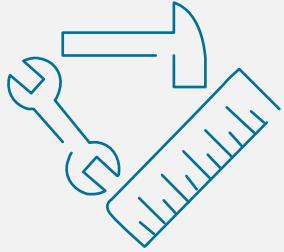
Separate a big problem into smaller problems handled by small teams ensures mental models are retained avoiding a “legacy in the making”



Ease of deployment/speed

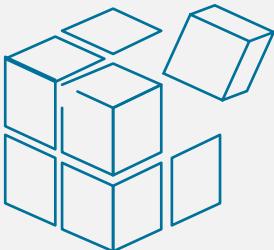
Moving away from entire system deployment (i.e. “one line change to a million-line-long monolithic”). Deploy services independently and fast (i.e. with containers) and introduce automation (continuous delivery)

Conclusion - Why adopt Microservices Architecture (II)



Scalability & resilience

Scale independently and possibly on-demand. Bulkheads to isolate problems and avoid whole system failures (avoiding the cascade effect), Then purposely test resilience



Breaking organizational silos

Organize small teams based on business capabilities in order to avoid organizational silos being reflected in the way systems are built (Conway's law)



Enabling cloud transition

Building container-based modular applications whilst adhering to basic principles (like 12 factor, Lehman's law, and the reactive manifesto) cloud adoption is a real option



ORACLE®

Diamond
Partner
Cloud Premier



About Capgemini and Sogeti

With more than 180,000 people in over 40 countries, Capgemini is a global leader in consulting, technology and outsourcing services. The Group reported 2015 global revenues of EUR 11.9 billion. Together with its clients, Capgemini creates and delivers business, technology and digital solutions that fit their needs, enabling them to achieve innovation and competitiveness. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Learn more about us at www.capgemini.com.

Sogeti is a leading provider of technology and software testing, specializing in Application, Infrastructure and Engineering Services. Sogeti offers cutting-edge solutions around Testing, Business Intelligence & Analytics, Mobile, Cloud and Cyber Security. Sogeti brings together more than 23,000 professionals in 15 countries and has a strong local presence in over 100 locations in Europe, USA and India. Sogeti is a wholly-owned subsidiary of Cap Gemini S.A., listed on the Paris Stock Exchange.

www.capgemini.com

www.sogeti.com

The information contained in this presentation is proprietary.
Copyright © 2016 Capgemini and Sogeti. All rights reserved.
Rightshore® is a trademark belonging to Capgemini.

Integrated Cloud Applications & Platform Services

ORACLE®