Microservices

An implementation overview

Tim James

ASCENDLE

Agenda

- Introductions
- Microservices
- Demo
- Recap
- Q&A

ASCENDLE

Introduction

ASCENDLE

Timothy James

linkedin.com/in/tdjdev

Technical Lead

Ascendle

Scrum as a Service

ASCENDLE

Our clients are market leaders and innovative pioneers.

Honeywell

MONSTER

amadeus Hospitality

tmpworldwide
THE DIGITAL BRAND AUTHORITY

PerkinElmer

wex

FLORATINE

Connecting Point
MARKETING GROUP

LIMRA

TRADEPORT
RETURNS, MANAGED.

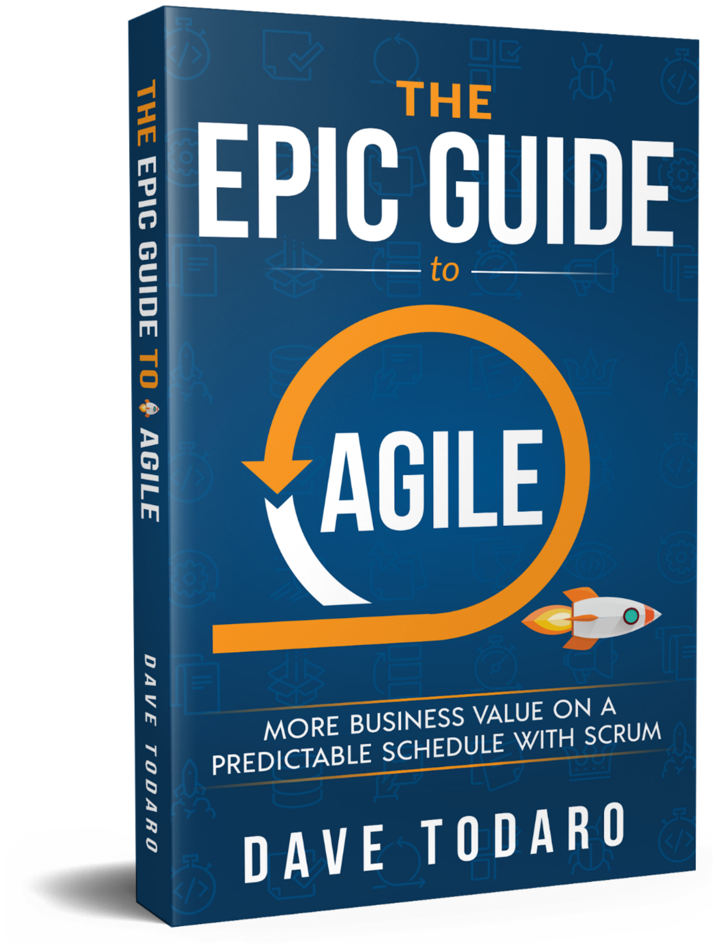
SENSITECH
United Technologies
SUPPLY CHAIN VISIBILITY

EPILEPSY FOUNDATION
New England

CREATIVE INFO
SYSTEMS

handplan

Pxl



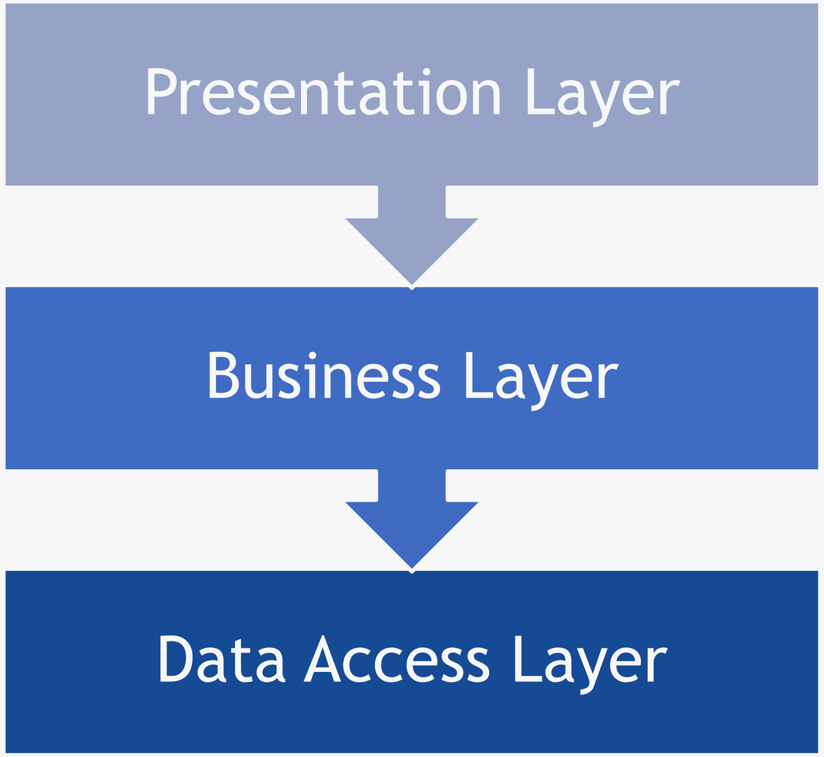
And we wrote the book.

Microservices

ASCENDLE

Traditional Development

- Uses a tiered approach for encapsulation
  - Commonly called the N-Tier model
  - Individual layers for individual processes
  - All within a single runtime
- Challenges include:
  - Tightly coupled layers
  - Maintainability
  - Scaling up



ASCENDLE

Welcome, Microservices

- Encapsulates code features into individual hostable services
- By creating individual services you can:
  - Write each service in any programming language
  - Guarantee actors only leverage publicly accessible endpoints
  - Allow code and state to be independently versioned, deployed, and scaled
  - Grant each service a unique URL
  - Encourage code reuse

Microsoft's Definition

Applications composed of small, independently versioned, and scalable customer-focused services that communicate over standard protocols with well-defined interface.

ASCENDLE

Microservice Benefits

Agility

Easier to update a service without redeploying entire application and rollback in case of errors

Parallel Development

Small teams can be focused on delivering individual services

Maintainability

Minimizing scope and in-code dependencies creates smaller more maintainable code bases

Fault Isolation

Individual service failures may not disrupt the entire application

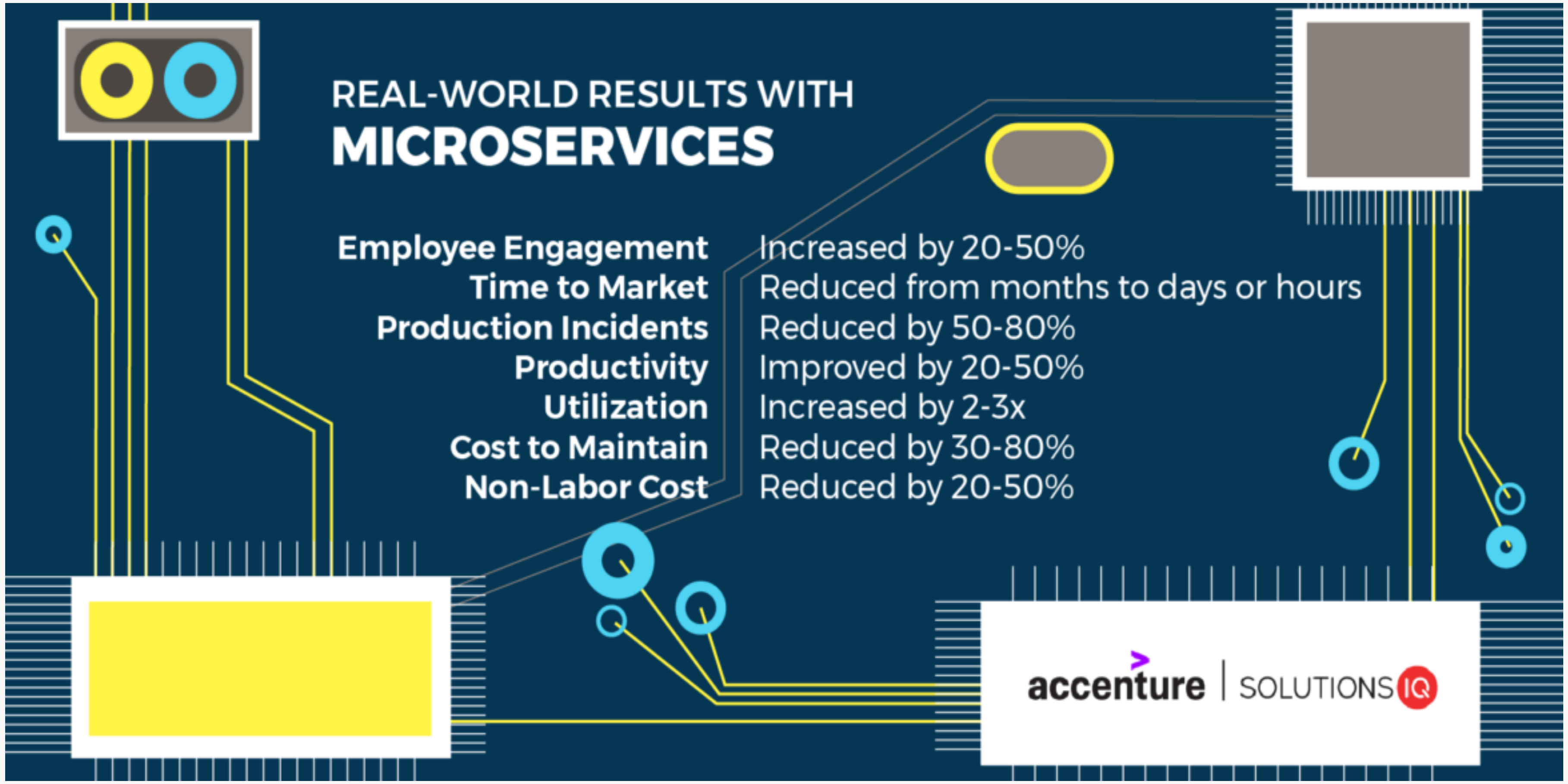
Scalability

With independent versions and deployments, services can be scaled independently

Pitfalls of Service Based Architectures

- Cross-talk between services exacerbates network latency and message processing time
- End-to-end testing becomes more complex
- Build, release, and deployment cycles become more complex
- Load balancing and fault tolerance mechanisms need to be considered before implementation begins
- Development teams need to be aware of the nanoservice anti-pattern
- Services act as information boundaries

Quantifying the Benefits



REAL-WORLD RESULTS WITH **MICROSERVICES**

Employee Engagement	Increased by 20-50%
Time to Market	Reduced from months to days or hours
Production Incidents	Reduced by 50-80%
Productivity	Improved by 20-50%
Utilization	Increased by 2-3x
Cost to Maintain	Reduced by 30-80%
Non-Labor Cost	Reduced by 20-50%

accenture | SOLUTIONS IQ

ASCENDLE

Common Patterns

"Pure"

- No side effects, no dependencies

Envelope

- Encapsulates the functionality of other services, usually 3rd party ones

Orchestration

- Invokes other services in order, and aggregates their results

Engine

- Typical domain-driven service that represents a business process

ASCENDLE

Primary Use Cases

New Builds

- New cloud applications that can be broken into domain-driven chunks

Lift & Shift

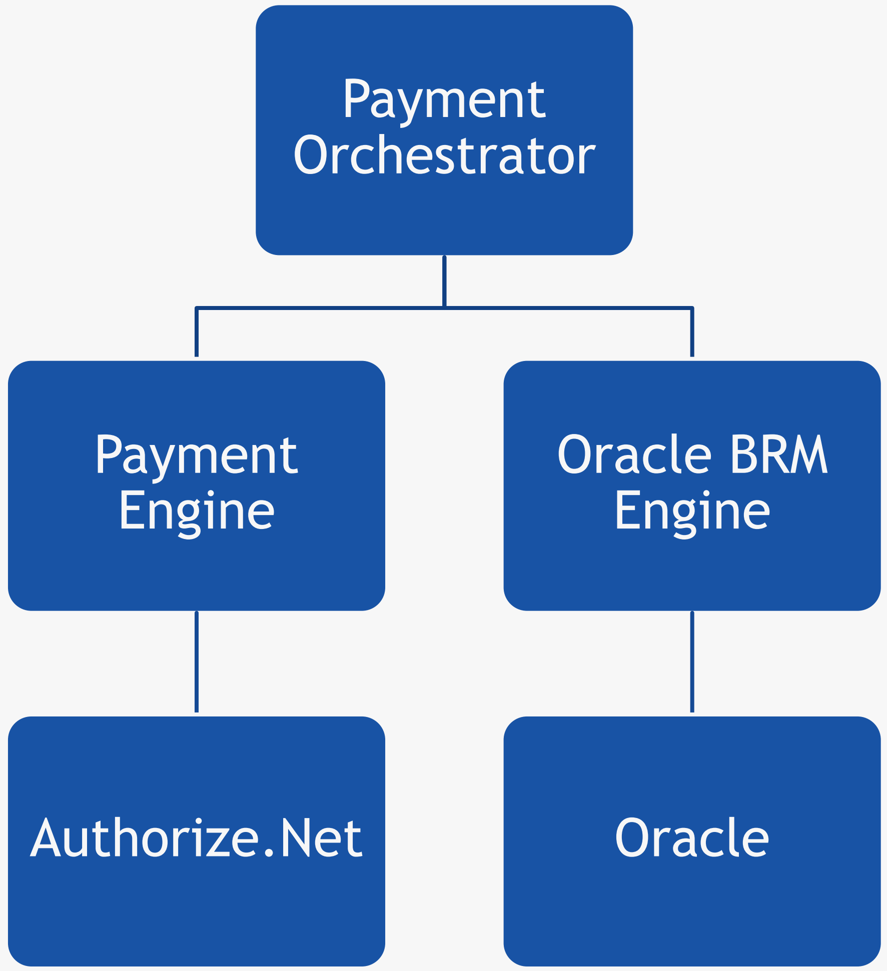
- Moving existing applications onto cloud infrastructure

Extend

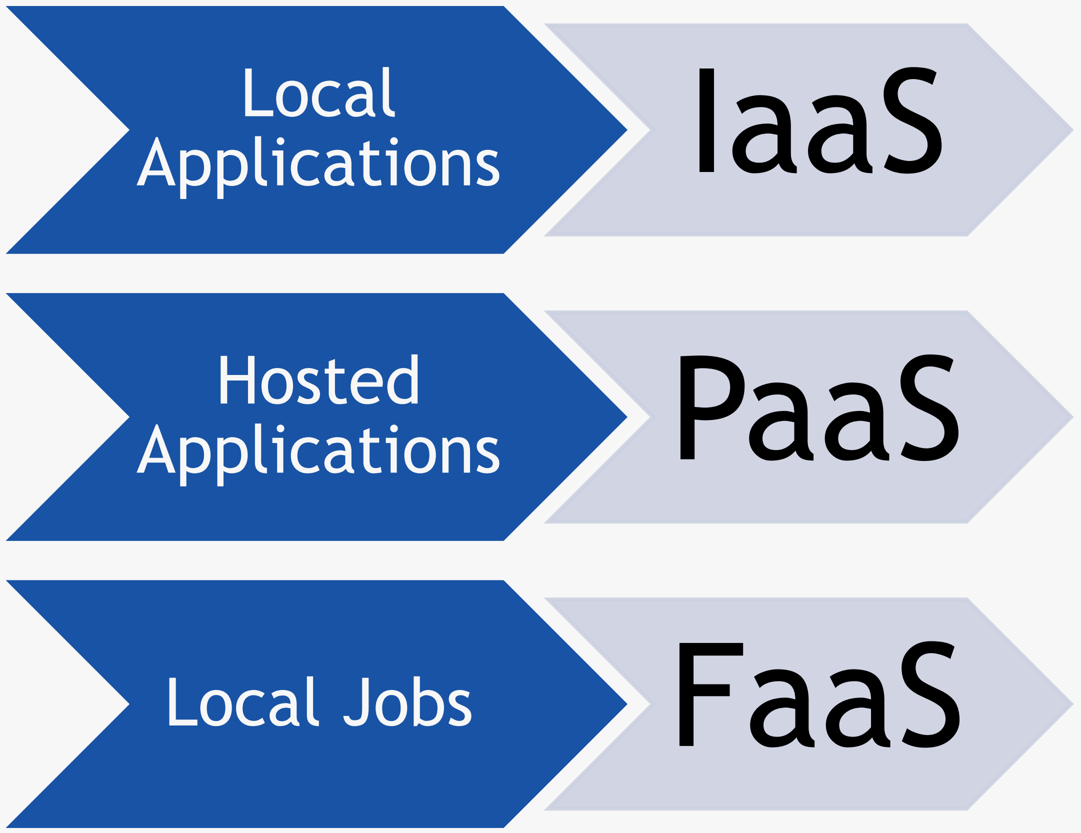
- Extending applications, especially legacy ones, with additional features

ASCENDLE

Example:
A New Build

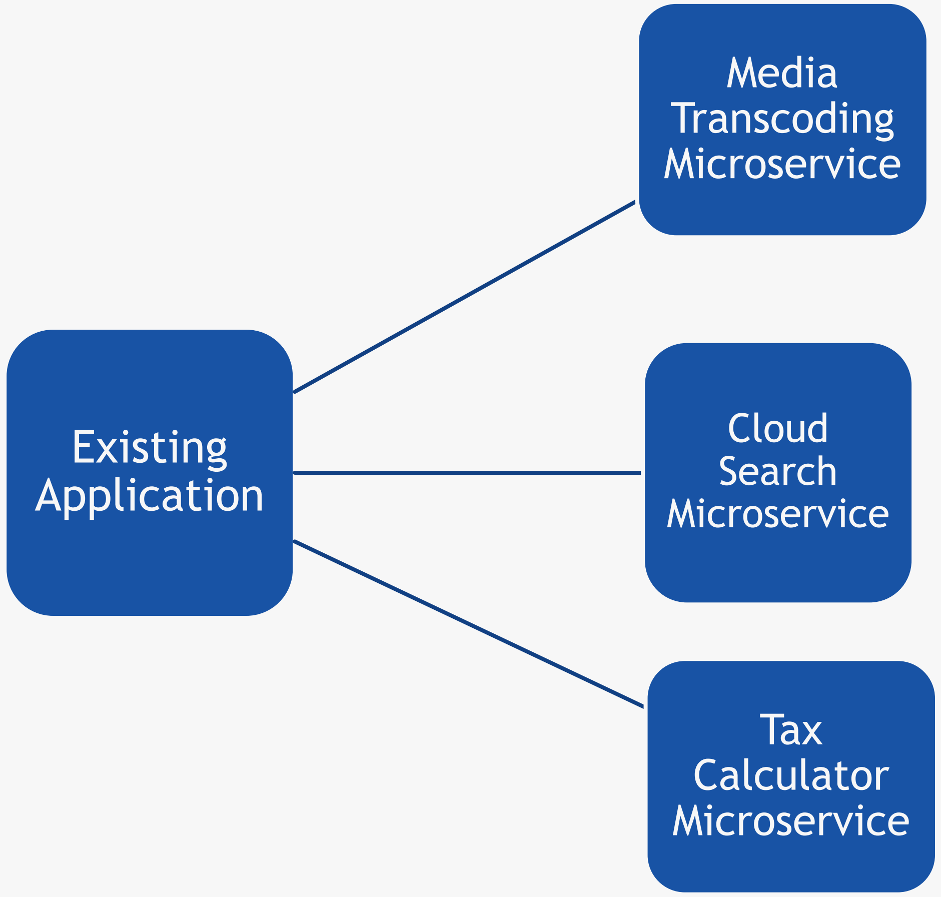


Example: Lift and Shift



ASCENDLE

Example: Extend


Existing Application
Media Transcoding Microservice
Cloud Search Microservice
Tax Calculator Microservice

What Tools Can I Use?

- Java and Spring
- Ruby and Grape
- Python and Flask
- .NET and ASP.NET Core
- NodeJS and Sails.js
- Function as a Service with
- Citizen Integrator tools

ASCENDLE

Where can I deploy?

- AWS
- Azure
- Google
- On-Prem
- Other Providers

ASCENDLE

What About Containers?

- Yes!
- Explicit
  - Define and manage yourself
- Implicit
  - Allow a hosting provider to do it for you
- Container-less
  - Do it the “old-fashioned” way

ASCENDLE

Demo Use Case

ASCENDLE

U.S. Export Compliance Use Case

As a business I need to be able to:

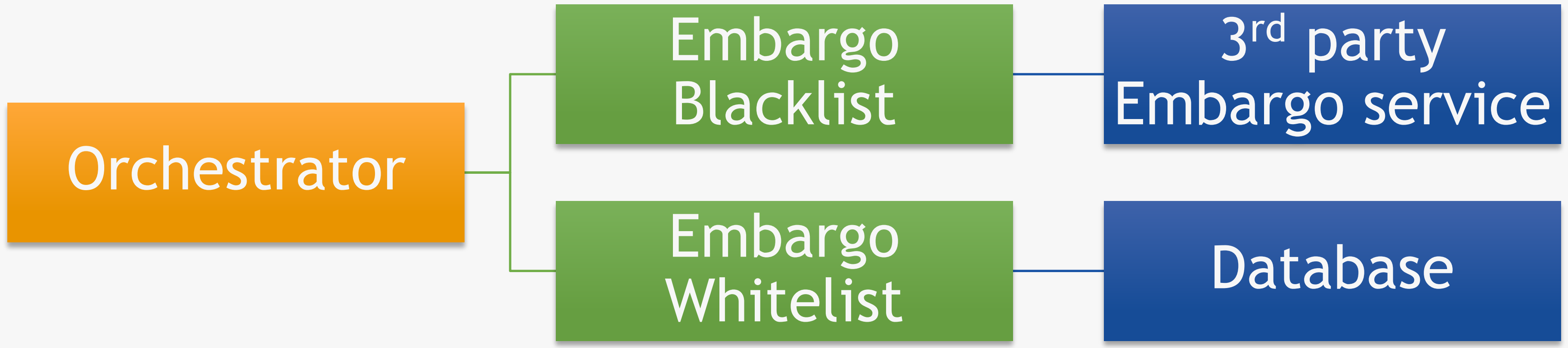
- Verify an individual's identity and whether I can do business with them
- Whitelist false positives by email address for a specific amount of time

This service will be called from:

- Our support site when a user registers their account
- Our public site when a user fills out a demo form or attempts to purchase a product

ASCENDLE

Architecture Diagram



ASCENDLE

Demo

ASCENDLE

Recap

ASCENDLE

Top takeaways

- Microservices are powerful architectural pattern
- Not a silver bullet, evaluate your needs before use
- Lots of options for implementation
- Enables teams to work in parallel
- Encourages code-level maintainability

Thank You

ASCENDLE

Q&A

ASCENDLE

Tim James

Technical Lead

linkedin.com/in/timjdev

tim@ascendle.com

ascendle.com

Text MICROSERVICE to 33777

For bonus content and a summary of this presentation

ASCENDLE

References

- https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-overview-microservices
- https://en.wikipedia.org/wiki/Microservices
- https://www.infoq.com/news/2014/08/failing-microservices/
- https://docs.microsoft.com/en-us/azure/architecture/microservices/
- https://www.solutionsiq.com/resource/blog-post/microservices-done-right-part-3-quantifying-the-benefits-of-microservices/
- https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ee658109(v=pandp.10)?redirectedfrom=MSDN
- https://www.guru99.com/microservices-tutorial.html