

OUTLINE

- Introduction to Monolith architecture and the problems it imposed on software development
- Evolution of microservice architecture to cope up with the problems of monolith architecture
- Evolution of distributed architectures
 Monolith -> Object Oriented -> Component based (SOA) -> Microservices
- Characteristics of microservices & its impacts on software development
- Interesting future directions of microservices

INTRODUCTION – MONOLITH

- Problems associated with large-scale software development were first experienced in 1960s
- Last decade has seen a shift towards concept of service-orientation and led to natural evolution of microservices
- Monolith architecture
- A software application whose modules cannot be executed independently
- Very difficult to use in distributed systems without specific frameworks
 Network Objects, RMI or CORBA might help but to a very little extent

Issues

- Difficult to maintain and evolve
- Dependency hell Adding or updating results in inconsistent system
- Small changes require rebooting the whole application
 Limited scalability
- Technology lock in for developers

NEW TECHNOLOGY – MICROSERVICES

- Microservice a cohesive, independent process interacting via messages
- Microservice architecture a distributed application where all its modules are microservices
- Guideline to design and implement distributed applications
- Partition the components of a distributed application into independent entities
- Programming of simple services to implement a single functionality
- Design and develop a highly maintainable & scalable software

MICROSERVICE ARCHITECTURE AS A SOLUTION TO MONOLITH

- Implement limited and specific set of functionalities
- Fosters continuous integration
- Changing the module doesn't require rebooting the whole application
- Microservices extend containerisation
- Scaling
- No lock-in to specific technology





EVOLUTION OF MICROSERVICES (1/5) YESTERDAY 1980's – integration of design & development made it hard to find a clear distinction between them 1990's – Object-oriented software – "patterns"

Example : Model View Controller(MVC) Service-Oriented Computing (SOC) – Component-based software engineering (CBSE) Services offers functionalities to other components, accessible via message passing 1990s and collier 20104 Traditional SOA

• • 9

0,00



EVOLUTION OF MICROSERVICES (3/5) TODAY

- Microservices, a new trend in software architecture
- Composition of small services,
- Introduced in 2011
- Fine grained SOA (Netflix)
- Characteristics
 - Size Service maintainability and extendability
 - Bounded context Related functionalities are combined into services
 - Independency Services are operationally independent
 - Modularity System provides Isolation of different functionalities
 - Flexibility System supports growing business needs
 - Evolution System stays maintainable with changes

EVOLUTION OF MICROSERVICES (4/5) TODAY Team 1968 - Melvin Conway – design is a copy of the organization's communication patterns Organize cross-functional teams around services

- Amazon CTO Werner Vogels "you build, you run it" principle
- Total automation Continuous integration, independent deployment, beneficial in rapidly changing business environments
- Choreography over orchestration
- Orchestration requires a conductor
- Choreography decentralized Impact on quality and management

9

11

Availability, Reliability, Maintainability, Performance, Security, Testability/

EVOLUTION OF MICROSERVICES (5/5) TOMORROW New issues posed by microservices

- Dependability building dependable systems with microservices Interfaces - need to specify formal me sage specifications between se
- Behavioural Specifications To check that the services have compatible actions
 Choreographies a possible future, still unclear in so many aspects
- Moving Fast with Solid Foundations start from scratch or reuse existing results?
- Trust and Security
- Greater Surface Attack Area
 Network Complexity
- Trust
- Heterogeneity

AUTHOR'S CONCLUSION

- Microservices architecture gained popularity recently both in academia and in the industrial world
- Still in its infancy and still there is a lack of agreement on what microservices actually are
- Few authors presented a revolutionary perspective to microservices, the author here has given an evolutionary perspective
- > No comprehensive collection of documentation in this field

12



CRITIQUE: WEAKNESSES

Security challenges

- Authentication is necessary in 3rd party services
- > The architecture is exposed to attacks
- Involves complex network activity
- > Attack on one microservice can bring down the application
- No common security infrastructure for heterogeneous systems

15

CRITIQUE: EVALUATION

About the paper

- Survey paper submitted by Cornell University in June 2016 Cited more than 30 times
- Explains the practical difficulties faced by developers in a Monolith architecture & introduces microservices and explains how it resolves the issues
- Provides an evolutionary perspective to this technology
- > Provides a clear distinction between different technologies that led to microservices from the scratch
- Paper acknowledges some of the challenges of microservices but there are not a lot of solutions to these pitfalls
- The architectural details and cost details of the microservices are outside the scope of this paper

16





