Fiche module				
Unité d'Enseignement	_			
Éléments Constitutifs	**SERVICE ORIENTED COMPUTING(SOC)**			
Niveau	**3ème année GIA**			
Semestre	**5**			
Volume horaire	**CI: 45H**	**-**	**-**	**-**
Enseignant Responsable	**Neila BEN LAKHAL**			

Prerequisites

- Web Programming
- Database fundamentals
- Object oriented modeling
- Students must familiarize themselves with the following technologies:
  - GIT platform
  - XML standards

Learning objectives (EN):

- Types of software Architecture.
- The evolution of the architecture of information systems: from monolithic to MSA.
- We will go over the basic Web service technology available today and learn:
  - WS-* standards (XML, SOAP, WSDL, BPEL).
- We will learn creating and calling SOAP-based Web services:
  - Bottom Up: Service class first
  - Top Down: Service description (WSDL) first
  - Using XML Schema to represent complex datatypes
- We will learn SOA principles and automating Business Processes as Web services composition
- We will compare types of services composition:
  - Dynamic | static composition
  - Orchestration |choreography
- We will discuss advanced topics on SOAP-based Web services e.g. Reliability of message exchange, security (privacy and authenticity), transaction, ...
- Implementing and calling RESTful Web services
  - What is the difference between SOAP-based and RESTful Web services?
- What is a Microservice?
- What is the difference between a (big) web service and a micro-service?
- Service oriented architecture (SOA) vs. Microservice architecture (MSA)?
- MSA principals, key concepts.
- Docker platform, containerizing a service-based application.

PLAN

Chapter 1.

Demystifying Software architecture

1. Software architecture definitions.
2. Software architectural styles taxonomy, from monolithic to MSA.
3. Fundamental architectural styles.
4. Evolutionary architectural styles.

Chapter 2.

Service oriented architecture (SOA)

1. Defining service concept. Service prominent characteristics.
2. Key principles for service orientation: Standardized service, contract Service, loose coupling, Service abstraction, Service reusability, Service autonomy, Service statelessness, Service discoverability, Service composability.
3. SOA key principles and fundamental actors.
4. Service lifecycle.
5. SOA delivery strategies top-down, bottom-up, and meet-in-the middle.
   Service composition, orchestration and choregraphy.
   Implementing SOA: XML, XSD and WS-* standards.
6. What are Web services?
7. A more complete definition of Web services
8. Characteristics of Web services
    a. Types of Web services: Simple Web services and Complex Web services or business processes.
    b. Functional and non-functional properties
    c. State properties
    d. Loose coupling
    e. Service granularity
    f. Synchronicity
9. Service interface and implementation
10. Roles of interaction in the SOA
    g. Web services provider
    h. Web services requestor
    i. Web services registry
11. Operations in the SOA
    j. The publish operation
    k. The find operation
    l. The bind operation
12. Web service lifecycle
13. The Web services technology stack

Chapter 3.

Web Services-Core functionality and standards

1. SOAP Protocol
    a. SOAP Protocol for Inter-application communication
    b. Structure of a SOAP message, SOAP header, SOAP body.
    c. Communication modes and messaging exchange patterns
    d. Error handling in SOAP
    e. SOAP over HTTP: message serialization steps
2. Describing Web services with WSDL
    a. WSDL: Web Services Description Language
    b. WSDL interface definition
    c. WSDL implementation
    d. Using WSDL to generate client stubs/Proxies
    e. Non- functional descriptions in WSDL
3. Platforms and APIs for web services implementation: Apache CXF, JAX-WS, AXIS, PHP.
4. Implementing Web services with APIs. in PHP and JAVA
    a. Bottom-up Web services implementation
    b. Top-down Web Services implementation
5. Implementing simple web service.
6. Implementing complex web service.
7. Web service discovery and invocation: APIs examples.
8. Web service testing with SOAPUI and Eclipse.
9. Composing web services
10. Orchestrating web services with BPEL

Chapter 4.

REST Architecture

1. Restful web services, introduction and motivation
2. The REST architectural style
3. HTTP protocol, http verbs, URL structure and the programmable Web.
4. Stateless, URI, resource, and methods. Manipulating resources through representation.
5. RESTful services vs hybrid REST services.
6. Testing real world services with SOAPUI and POSTman clients.
7. Description and Discovery of RESTful Web Services :The need for an interface contract
    a. Web Application Description Language
    b. RESTful API Modeling Language
    c. Swagger
8. Restful service WADL vs SOAP WSDL.
9. Implementing REST services client http library.(JSON/XML response).
10. Comparing SOAP to REST web services: data exchange format, security, composition.

Chapter 5.

Microservice Architecture (MSA) : key architectural elements

1. Monolithic application
2. Need for microservice architecture, problems with monolithic applications.
3. Microservice (MS) definitions, characteristics.
4. MS vs. Big services.
5. Migrating from monolithic architecture to MSA: approaches and recommendations.
6. Key architectural elements of MSA
    a. Communication models for microservices: synchronous and asynchronous.
    b. Service registration
    c. Service discovery
7. Essential patterns of microservice-centric applications
    a. Circuit breaker pattern
    b. API gateway pattern.
    c. Service registration pattern.
8. MSA vs SOA.
9. Key enabling technologies for Microservices Architecture.
    a. Microservices implementation Microframeworks
    b. Microservices isolation and scalability tools: Containerization platform
    c. Microservices composition, orchestration and choregraphy:
        i. Containerized Apps composition: Docker compose, docker swarm, Kubernetes
        ii. Service composition engines: BPEL, Conductor (Netflix), JOLIE.

Chapter 6.

Docker: a Defacto Platform for Microservices Architecture

1. What is Docker?
2. Docker Container history
    a. Why do we need containers in IT
    b. Containerization vs virtualization
3. Docker containers properties
4. Docker ecosystem
5. Images, building and publishing images in Dockerhub
6. Docker architecture
7. Our first container: Demo
8. Installing docker
9. Running a container
    a. Background containers
    b. Foreground containers
    c. Interactive containers
10. Attaching /detaching form containers
11. Stopping containers
12. Removing containers
13. Sum-up Docker commands: ps, run, start, restart, attach, stop, rm, create
14. Lifecycle of a docker container
15. Docker containers networking basics
    a. Why do we need networking?
    b. Accessing a container

c. Port mapping : explicit vs implicit port mapping

16. Network models
    a. Single host networks: host network, none network, default and used defined bridge networks
    b. Multi-host networks: overlay and macvlan network
17. Volumes and data persistence
    a. Bind mount
    b. Volumes : anonymous volumes
    c. Named volumes , Dockerfile volume instruction.
    d. Docker volumes managements
    e. Data-only containers
18. Images, pulling images,image registries, naming, tagging, searching dockerhub.
    a. Images and layers
    b. Image hash digest
    c. Multi-architecture image
    d. Removing images
19. Dockerizing an application
    a. Method 1: Creating Dockerfile
        i. Publishing an image, tag, push image, build process, publish image
        ii. Dockerfile syntax, instructions
        iii. ENV, WORKDIR, EXPOSE, ENTRYPOINT, CMD (Shell and EXEC form)
        iv. Image layers reuse, caching.
        v. Container layer vs image layer.
        vi. Copy on write mechanism
        vii. Storage drivers.
    b. Method 2: DOCKER command “by Hand”
        i. Docker commit command
    c. Method 3: DOCKERFILE and configuration management tool
    d. Method 4: scratch image and import files
        i. Docker export, docker import, docker save and docker load commands
20. Docker compose: multi-container docker apps
    a. Services, volumes, networks and docker stacks
    b. Docker compose commands.
    c. YAML syntax.
    d. Orchestrating services with docker-compose.
    e. Scalability in docker compose
    f. Adding load balancing.
    g. Docker-compose limitations.
21. Orchestrators
    a. Docker swarm
        i. Swarm managers,
        ii. Swarm workers
        iii. Services, tasks
        iv. Building a swarm cluster
        v. Docker machine, VMs and nodes.
    b. Docker stack
        i. Docker stack management subcommands
        ii. Services

iii. Local muti-node swarm
iv. Deploying multi-service app stack
v. Managing apps stack
vi. Networking in a docker swarm

22. Running a Containerized App in the cloud

BIBLIOGRAPHIE & NETOGRAPHIE

[1] Web Services Concepts, Architectures and Applications. Gustavo Alonso, Fabio Casati, Harumi Kuno, et Vijay Machiraju. Springer Verlag (2010).

[2] PHP 5 avancé. Eric Daspet et Cyril Pierre de Geyer. Eyrolles (2007).

[3] Services Web avec J2EE et .NET : Conception et implémentations. Libero Maesano, Christian Bernard, et Xavier Le Galles, Eyrolles (2011).

[4] SOA in Practice. Nicloai M. Josuttis. O'Reilly (2007).

[5] Pro PHP XML and Web Services, Robert Richards. Apress (2006).

[6] RESTful Web Services, Leonard Richardson et Sam Ruby. O'Reilly Media (2008).

[7] Web services composition: A decade's overview. Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo, S. Bourne, and X. Xu. Information Sciences, vol. 280 (2014).

[8] Microservices: yesterday, today, and tomorrow Nicola Dragoni, Saverio Giallorenzo, Alberto Lluch-Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, Larisa Safina (2016).

[9] Reflections on the REST architectural style and principled design of the modern web architecture Roy T. Fielding, Richard N. Taylor, Justin R. Erenkrantz, Michael M. Gorlick, Jim Whitehead, Rohit Khare, and Peyman Oreizy. 2017. (impact paper award). In Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, ACM, New York, NY, USA, 4-14, (2017).

[10] Restful web services vs. big web services: making the right architectural decision. Cesare Pautasso, Olaf Zimmermann, and Frank Leymann.In Proceedings of the 17th international conference on World Wide Web, ACM, New York, NY, USA, 805-814, (2008).

[11] Towards an Understanding of Microservices, D. Shadija, M. Rezai, R. Hill, In Proceedings of the 23rd International Conference of Automation and Computing (ICAC), University of Huddersfield, IEEE Computer Society, (2017).

[12] Web services: Principles and Technology, Michael P. Papazoglou, Pearson Education Limited, second edition, (2012).

[13] The teaching materials of the book "Web Services: Concepts, Architectures and Applications" ETH Zurich, Springer-Verlag Berlin Heidelberg (2004).

[14] SOA and Web Services Interface Design: Principles, Techniques, and Standards James Bean, ISBN: 9780123748911, Morgan Kaufmann Publishers (chap1,2,3,10) (2010).

[15] PHP Web Services, APIs for the Modern Web by Lorna Mitchell, O'Reilly Media, (2013).

[16] RESTful Web Services by Leonard Richardson, Sam Ruby, Published by O'Reilly Media, (2008).

[17] Microservices vs. Service-Oriented Architecture, by Mark Richards Copyright © 2016 O'Reilly Media.

[18] Microservices, M Fowler and J Lewis. ThoughtWorks, 2014

[19] So You're Thinking of Decomposing Your Monolith into Microservices. Jones, T. (2017)

[20] Microservices: Decomposing Applications for Deployability and Scalability Richardson, C (2014).

[21] Microservices: yesterday, today, and tomorrow. Nicola Dragoni, Saverio Giallorenzo, Alberto Lluch-Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, and Larisa Safina. CoRR, abs/1606.06, (2016)

[22] RESTful Java Web Services: A pragmatic guide to designing and building RESTful APIs using Java. Bogunuva Mohanram Balachandar. Packt Publishing, third edition. (2017)

[23] Building Web Services with Java: MAKING SENSE OF XML, SOAP, WSDL, AND UDDI. Steve Graham and Al. Sams Publishing; 2nd Edition (2004)

[24] Web Services and SOA: Principles and Technology, Papazoglou, M. 2nd Edition. Prentice Hall, Pearson Education. (2011).

[25] Essentials of Microservices Architecture Paradigms, Applications, and Techniques, Chellammal Surianarayanan, Gopinath Ganapathy, Raj Pethuru, Taylor & Francis; 1st Edition (2019).

[26] Software Architect's Handbook, Joseph Ingeno Packt Publishing (2018).

[27] Architectural Patterns, Pethuru Raj, Anupama Raman, Harihara Sub., Packt Publishing (2017).

[28] Fundamentals of Software Architecture, Mark Richards and Neal Ford, O'Reilly Media (2020).

[29] Excerpt From: Neal Ford. "Fundamentals of Software Architecture." Apple Books.

[30] Building Web Services with Java, Steve Graham et al, Second Edition, Sams publishing, (2005).

[31] Docker deep dive, Nigel Poulton (2018).

[32] Docker: Up & Running: Shipping Reliable Containers in Production, by Sean P. Kane, Karl Matthias, O'Reilly Media; 2nd Edition (2018).

[33] Using Docker: Developing and Deploying Software with Containers, Adrian Mouat 1st Edition, (2016).

[34] Docker in Practice 2nd Edition, Ian Miell, Aidan Hobson Sayers, Manning Publications (2019).

[35] Docker in Practice 1st Edition, Ian Miell, Aidan Hobson Sayers, Manning Publications (2016).

[36] Docker in Action, Jeff Nickoloff, Aiden Humphreys, Manning Publications (2018).

[37] Mastering Docker: Unlock new opportunities using Docker's most advanced features, 3rd Edition, Russ McKendrick, Scott Gallagher, Packt publishing, (2018).