

Vue d'implémentation Spring Boot avec Enterprise Architect

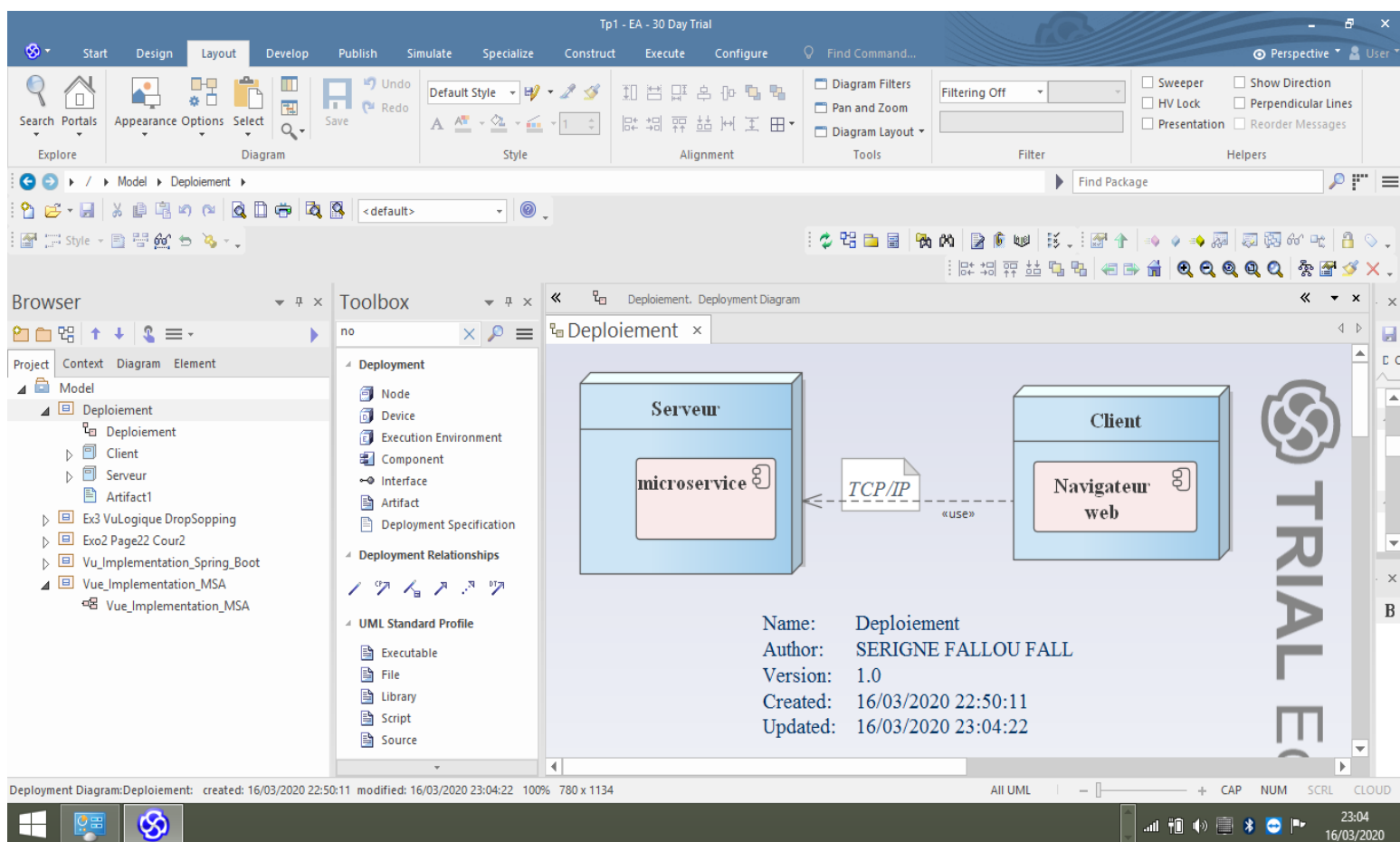
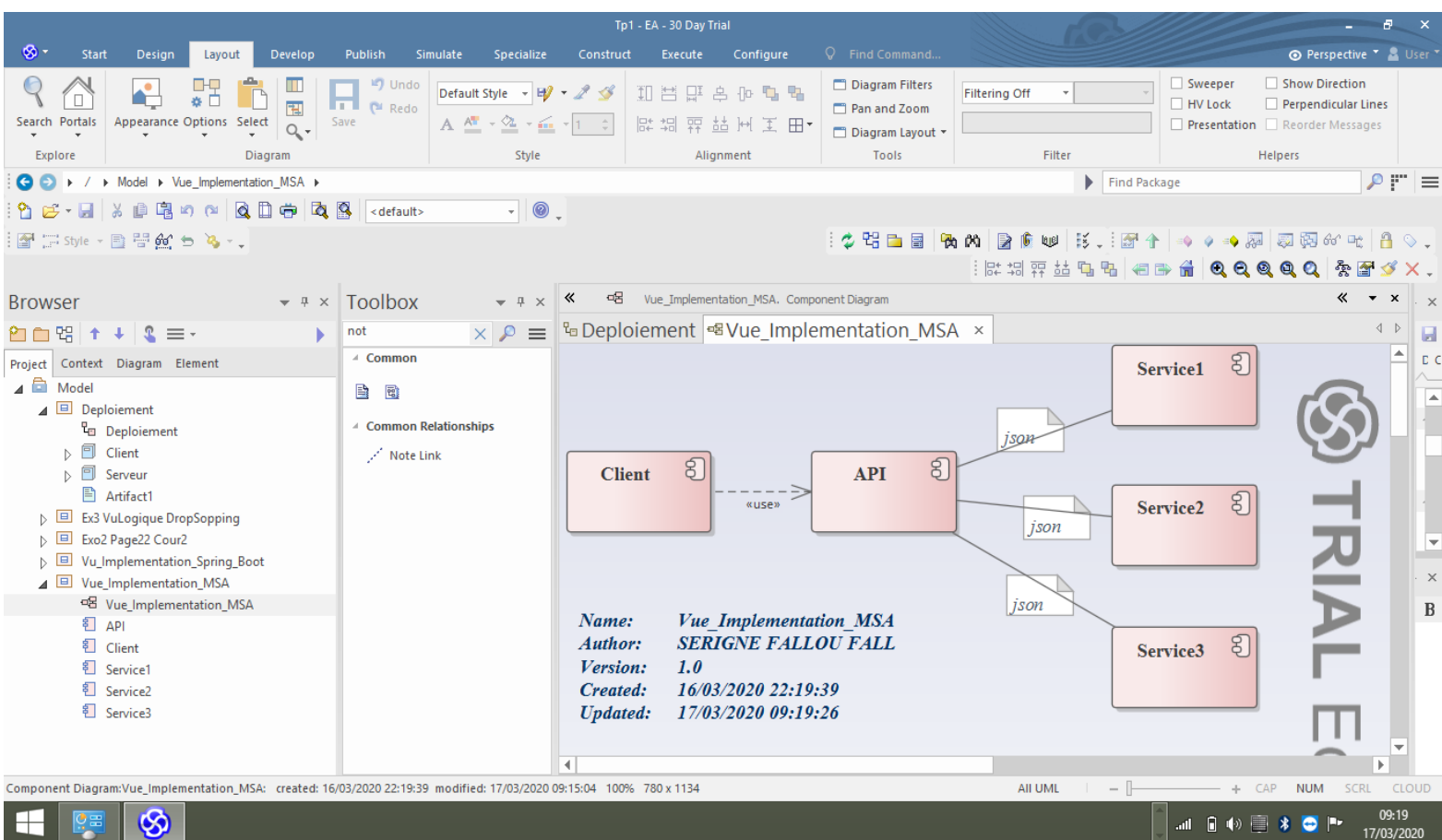


Diagramme de déploiement de notre solution



Vue d'implémentation de MSA

Etude comparatif entre MSA et SOA

Les microservices

Les microservices sont autant une architecture d'infrastructure qu'une architecture logicielle. En fait, on pourrait supposer que le code présent dans le service est élaboré selon des principes de type SOA. Toutefois, dans le cas de microservices, l'implication tient à ce que l'infrastructure sur laquelle chaque service s'exécute est indépendante des autres. Aussi son autonomie est-elle préservée sur la couche serveur autant qu'au niveau du code. Ce procédé implique également que la répartition des charges peut intervenir service par service, ce qui offre aux équipes davantage de possibilités pour optimiser les performances, comparé à la SOA. En d'autres termes, accroître la capacité pour un service rattaché revient à lancer de manière indépendante de nouvelles instances de ce service, et ainsi à offrir une évolutivité accélérée. Cela entraîne en outre une résolution plus rapide des problèmes, car dans un tel modèle, il est très probable que des services puissent s'arrêter sans entraîner l'immobilisation de toute l'application.

Les microservices profitent à l'infrastructure mais également aux déploiements. Ceux-ci sont plus rapides, car vous déployez individuellement différents services. Il suffit en effet de déployer uniquement les services mis à jour et non plus l'intégralité de l'application. Cette caractéristique apporte une forme de continuité dans la mise à disposition et au sein des déploiements, car les plus petites unités peuvent toutes être déployées indépendamment à moindre risque.

L'approche SOA

L'approche SOA est similaire aux **microservices** ; tout comme ces derniers sont semblables à une architecture logicielle MVC (Model-View Controller). Les idées sont les mêmes : fragmenter les applications en composants plus petits, avec une empreinte plus réduite et plus faciles à gérer. Toutefois, la plus grande différence entre ces technologies tient à l'objet et au timing de leurs introductions respectives. L'architecture SOA est arrivée la première, puis vint l'architecture MVC et enfin, les microservices.

L'approche SOA (Service Oriented Architecture) est un terme purement logiciel. Son objectif consiste à faire appliquer un schéma de conception plus efficace en matière de développement d'applications. Mais la couverture s'arrête là. En effet, cette approche n'affecte pas l'infrastructure. Ce point est important, car les applications sont devenues si complexes qu'elles occupent l'intégralité de la pile, voire des datacenters entiers. L'idée des microservices consiste à étendre la mise en oeuvre de ces plus petites unités et techniques d'isolement à l'infrastructure, ainsi qu'à la conception des applications.

La technologie SOA concerne généralement une application distincte comprenant plusieurs services internes. C'est en cela qu'elle est, selon moi, similaire à une architecture MVC. Mais il existe une grande différence : l'architecture MVC met en place une barrière entre des composants applicatifs de trois types. Elle ne leur donne pas une pleine autonomie ; en fait, les relations sont dictées. L'approche SOA leur accorde une autonomie et ne requiert des applications que le seul partage d'un schéma. Les services sont ainsi plus portables et l'élaboration de nouvelles applications encore plus simple. Mais l'API moderne fait la même chose dans une large mesure en installant une couche de services dédiée, elle aussi, au déchargement d'une plus grande part de logique. L'approche SOA a été élaborée à une époque où les API ne se trouvaient pas là où commençait le processus de développement d'applications.

L'approche SOA est une pratique que le développeur exécute au sein d'un ou plusieurs codes applicatifs, tandis que les microservices dictent jusqu'à la structure de l'équipe et aux rôles informatiques : les équipes de développement sont cloisonnées en fonction des services qu'elles élaborent. Ce constat se vérifie également pour les applications SOA. Toutefois, dans le cas SOA, le développeur est tenu de disposer de connaissances approfondies sur chaque service. Etant donné que les microservices peuvent également servir de mécanismes de mise à disposition d'infrastructure, ils ont une incidence plus importante sur le déploiement et sur ladite infrastructure.