International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Mastering Service Virtualization: A Guide for SDLC Professionals

Dr. Ranjith Gopalan

PhD, Sr Architect
Email: *ranjith.gopalan[at]gmail.com*

Abstract: ***Service virtualization is a concept that has gained prominence in recent years as organizations strive to improve the efficiency and speed of software development and delivery. This paper aims to explore the latest techniques and tools in API service virtualization and Data base service virtualization, their impact on the software development life cycle. This paper explains the importance of API service virtualization and its important in unit testing API source code and application testing were service response late or third-party service not available. Paper talks Service virtualization plays a crucial role in the software testing life cycle (STLC) by allowing testers to simulate the behavior of dependent systems that are not yet available for testing. This is especially important in today's fast-paced development environments where teams are often working on interconnected systems that may not be ready for testing at the same time. By using service virtualization, testers can create virtualized versions of these dependencies, enabling them to test their own systems in isolation and without delays. By allowing teams to simulate dependencies, accelerate testing cycles, improve test coverage, and automate testing processes, service virtualization enables organizations to deliver high-quality software more quickly and reliably. This paper explains the important of database virtualization and its impact in SDLC and will provide great direction and insight to student and Delivery manager to understand the best strategy and process to follow. The paper talks that database service virtualization is a critical component of the SDLC, especially for professionals working in the niches of database service virtualization and machine learning-enhanced database service virtualization. By simulating databases in a virtual environment, developers can test their applications more efficiently, identify and fix issues early in the development process, and leverage machine learning algorithms to create more realistic testing scenarios. Ultimately, database service virtualization helps to improve the quality and reliability of software products, leading to greater success in the competitive world of software development.***

Keywords: Service Virtualization, CA DevTest, Parasoft, API Mocking, IBCO Data Virtualization, IBM Cloud Pak

1. Introduction

In today's fast-paced and competitive business environment, organizations are constantly looking for ways to improve the efficiency and speed of software development and delivery. Service virtualization has emerged as a critical concept in this pursuit, allowing teams to simulate the behavior of certain components or services in a virtual environment. This enables developers to test and validate their code without relying on the availability of the actual components or services, thereby reducing dependencies and accelerating the development process.

This paper explores the most recent advancements in API and database service virtualization, examining their effects on the software development life cycle. Understanding these advancements and their implications for the SDLC enables organizations to streamline development processes and deliver superior software products with greater efficiency.

The following sections will examine the various aspects of service virtualization and its application in SDLC, shedding light on the benefits and challenges associated with this approach. Additionally, we will discuss real-world examples of organizations leveraging service virtualization to achieve notable improvements in their software development practices.

2. Methodology

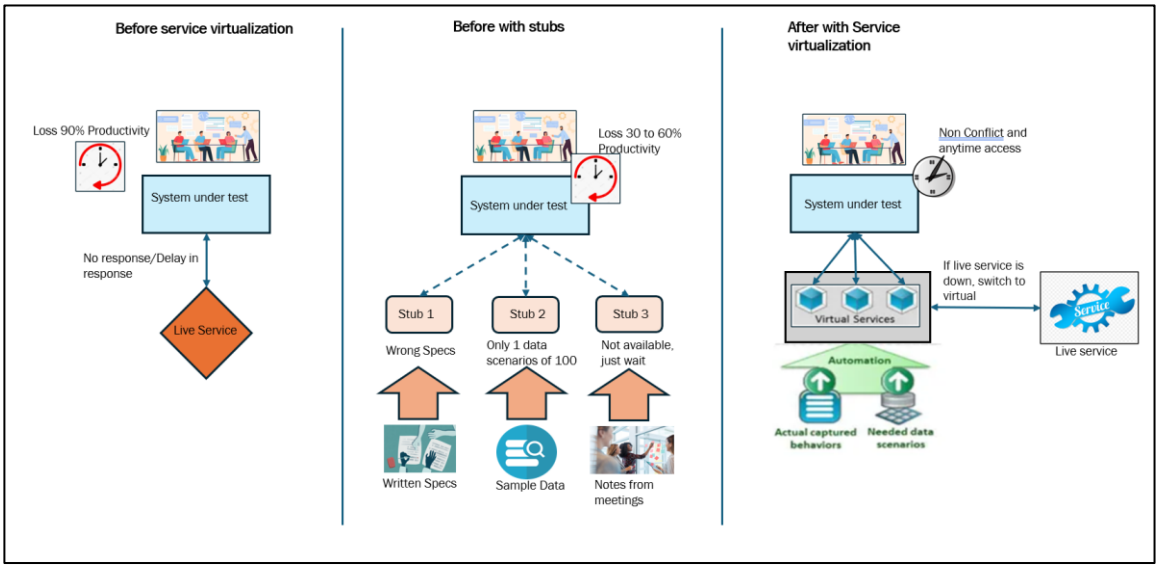
Service virtualization leverages various methodologies to create and simulate virtual environments for testing and development. One of the latest techniques in service virtualization is the use of AI and machine learning algorithms to create intelligent virtual services that mimic the behavior of real services with higher accuracy and precision.

Additionally, advanced techniques such as containerization and microservices architecture are being integrated into service virtualization to enable more granular and efficient simulation of complex systems and components. These modern approaches facilitate the creation of lightweight and scalable virtual environments, allowing for rapid testing and validation of software in a wide range of scenarios.

Furthermore, the adoption of cloud-based service virtualization platforms has revolutionized the way organizations approach testing and development. Cloud-based solutions offer on-demand access to virtualized services, enabling teams to perform comprehensive testing across diverse environments and configurations without the need for complex infrastructure setup.

Incorporating these latest techniques in service virtualization has significantly impacted the software development life cycle. It has led to reduced time-to-market, improved software quality, and enhanced collaboration between development and testing teams. By leveraging intelligent virtual services and modern architectural approaches, organizations can effectively address the complexities of

today's software development landscape, ultimately delivering robust and reliable software products.



The above figure explains the importance of service virtualization.

API service virtualization

API virtualization is the process of creating virtual versions of systems upon which new applications depend. These virtual models enable the assessment of how well software applications communicate with these systems. API virtualization serves a crucial role in the integration of software applications with cloud services, service-oriented architectures (SOA), and external data and APIs. The advantages of API virtualization include enhanced testing efficiency, as it allows teams to evaluate application interactions with dependent systems outside of the live production environment. It also reduces the risk of production system disruptions during testing phases and contributes to a shorter time-to-market by facilitating earlier testing, thus speeding up development processes.

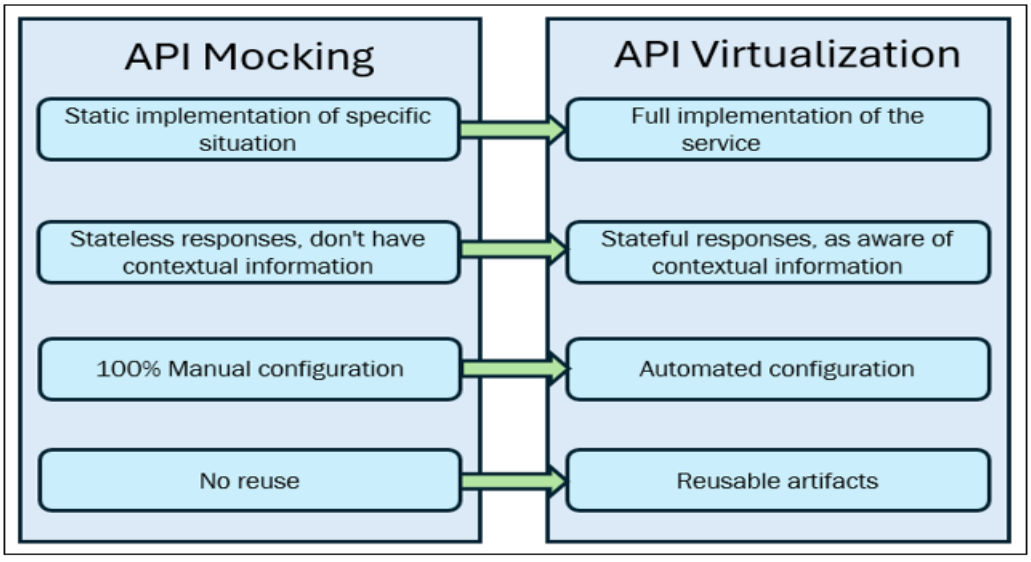
API Virtualization vs. API Mocking

API virtualization and API mocking are often mentioned in the same breath, yet they fulfill distinct roles.

API Virtualization is akin to having a mirror image of your production API, a virtual replica that behaves and responds just like the real deal. This powerful feature enables developers to conduct thorough testing at any stage of development by simulating the API's behaviors and responses. It's comparable to sampling a cake mid-bake, giving you a taste of the final product before it's fully done.

On the other hand, API Mocking is the art of creating a decoy of real software components solely for testing purposes. Developers craft these mocks to stand in for actual components during the testing phase. Think of it as a cardboard cutout of a cake; it may look like a cake, but it's not meant for eating. It serves as a visual stand-in, nothing more.

In essence, while both virtualization and mocking are integral to the API lifecycle, they cater to different needs and stages of the development process. Virtualization offers a dynamic, interactive testing environment, whereas mocking provides a static, surface-level imitation for preliminary testing.



Above figure explains difference between API Mocking and API Virtualization

Database service virtualization

Database Service Virtualization is a crucial concept in the realm of Software Development Life Cycle (SDLC) that aims to replicate the behavior of a database service in a simulated environment. This allows developers and testers to efficiently conduct testing and development activities without relying on the actual database service. In essence, Database Service Virtualization involves creating a virtualized representation of the database service, including its schemas, data, and interactions. This virtualized environment closely mimics the behavior of the actual database service, enabling developers and testers to perform various tasks such as data manipulation, querying, and testing without impacting the production environment.

Machine learning-enhanced database service virtualization takes this concept a step further by leveraging artificial intelligence and predictive analytics to enhance the accuracy and efficiency of the virtualized environment. Machine learning algorithms can analyze historical data patterns, predict future behaviors, and optimize the performance of the virtualized database service. This advanced approach not only ensures a more realistic simulation but also helps in uncovering hidden insights and trends that can further improve the software development process.

Methodologies related to database service virtualization are given below and combine aspects of both database management and service virtualization to ensure efficient testing and development.

Data Subset Virtualization: Data subset virtualization involves creating a subset of the original database with relevant and representative data for testing purposes. This allows developers to work with a smaller dataset that still accurately represents the production environment, saving time and resources in the testing process.

The goal is to create a virtual subset of the production database for testing. The process involves selecting a representative data subset from the production environment, virtualizing it to reflect the original structure and relationships, and utilizing it for testing to minimize the need for complete database replicas. This results in quicker provisioning of test data, less storage use, and enhanced test speeds.

Data Masking and Obfuscation:

This aims to safeguard sensitive data while enabling realistic testing. The method includes identifying confidential information and applying masking techniques to replace real values with fictitious or altered data, preserving data integrity. The advantages are compliance with privacy laws like GDPR and the use of realistic test data without revealing sensitive details.

Data Generation Algorithms:

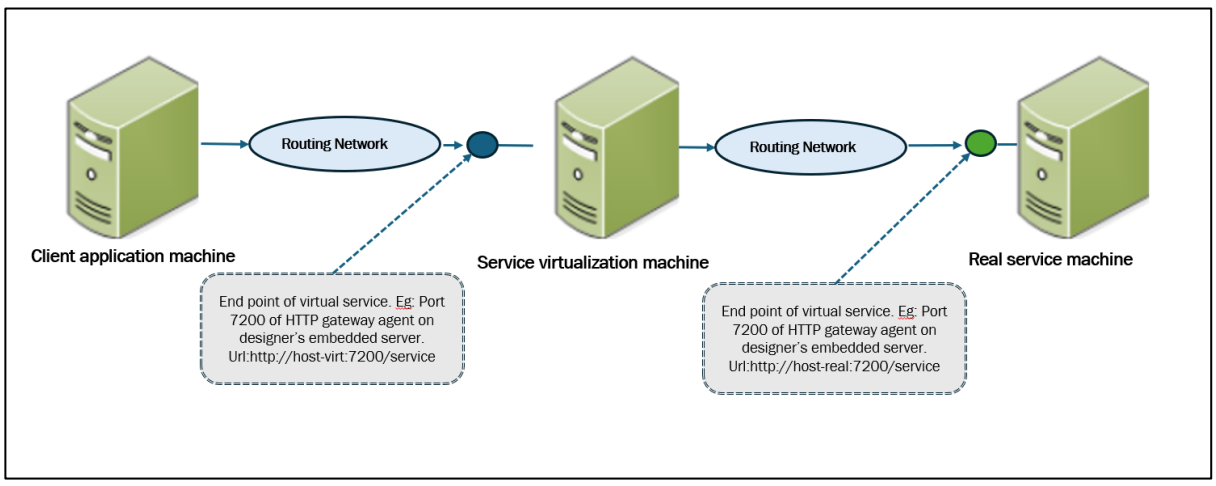
The objective here is to produce synthetic data that mirrors actual scenarios. Algorithms are developed to generate realistic data patterns, such as customer demographics or transaction histories. These algorithms dynamically create test data that complies with business rules and constraints, offering scalable data generation, customizable scenarios, and less dependence on actual production data.

Database Schema Virtualization:

The aim is to separate the database schema from the data itself. A virtual schema that replicates the production schema is created and used for testing, allowing for schema modifications without affecting the actual data.

3. Model and Analysis

Service Virtualization uses several TCP ports for communication. To configure Service Virtualization to work correctly in a protected network environment, you must verify that all required network ports are open.in order to record and simulate the communication between a client application and a real service endpoint, you must place Service Virtualization between them. In this scenario, communication from the client application to the virtual service, and from the virtual service to the real service is as follows:



The above diagram illustrates that the client application has been reconfigured to interact with a virtual service rather than the actual service. This virtual service may be deployed on either the Service Virtualization Designer's embedded server or the Service Virtualization Server.

Implementing API Virtualization

To effectively implement API virtualization, consider the following steps:

- **Choose the Appropriate Tool**: Select an API virtualization tool that aligns with project requirements. Look for features such as straightforward setup, comprehensive simulation options, and broad API compatibility.
- **Develop Virtual APIs:** Create virtual versions of APIs that replicate the expected behavior and specifications, allowing for extensive and early testing without relying on production APIs.
- **Incorporate into Development Workflow**: Integrate virtual APIs smoothly into existing development workflow to facilitate their use throughout the development and testing phases.
- **Advocate for API Virtualization:** Champion the adoption of API virtualization across organization by emphasizing its advantages, such as increased efficiency, cost savings in testing, and overall improvement in API quality. Discuss the latest techniques in service virtualization and their transformative impact on the software development life cycle (SDLC).

Tools for API virtualization

This paper explains the different tools used for API virtualization.

1) Custom Code

With Java or Python, it's possible to write custom code that records live service transactions over the HTTP transport protocol. This code can also construct a data protocol to parse transactions in JSON, XML, or SOAP formats, and save them as either stateless or stateful transactions. These transactions can be set up as stateless or stateful before the recording begins.

After the recording process is complete, the custom code allows the recorded transactions to be replayed and simulated to emulate the actual service endpoint's behavior. By using an EC2 or VMC machine, the recorded transactions for each API can be deployed into a virtual environment, creating a virtual URL. This URL can then be utilized by client applications for testing and development. The virtual image can be set up in such a way that the application will first attempt to connect to the live URL; if there is no response, it will then retrieve a dynamic response from the virtual URL. The key to a successful API virtualization implementation lies in the creation of dynamic responses, which involves optimizing custom code with the appropriate data combinations.

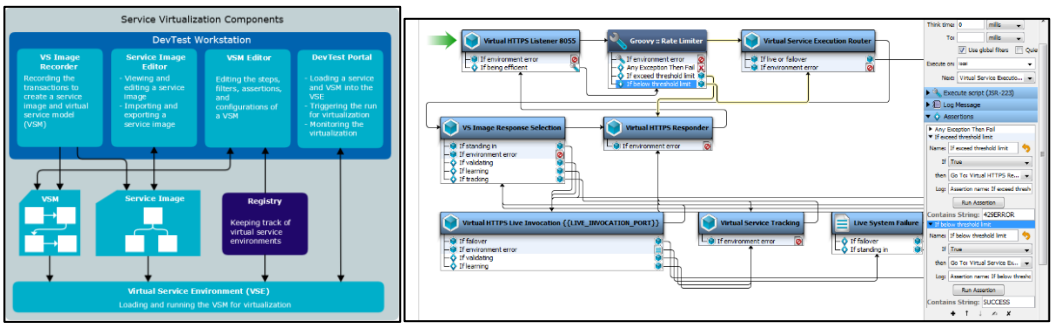
2) CA DevTest

CA Service Virtualization, crafted by Broadcom, is instrumental in advancing continuous testing and enhancing software excellence. It automates the generation of holistic software environments that mirror real-world behaviors, transactions, and performance metrics, transcending basic stubs or responders. These environments are perpetually accessible and demand minimal setup.

Within the DevTest suite, CA Application Test scrutinizes application layers, while CA Service Virtualization replicates the dynamics of inaccessible systems, fostering a robust testing milieu that bolsters application testing and quality.

The tool's advantages include facilitating concurrent development and testing, promoting early 'Shift Left' testing to detect issues promptly, and offering extensive protocol support. It adeptly mimics realistic system behaviors and simplifies virtual service management, thus streamlining testing and development processes.

Moreover, it equips testing teams with a suite of automated API testing tools, enhancing the maintenance and execution of continuous testing strategies.



Above picture explains the components and Virtual model is being used in CA DevTest

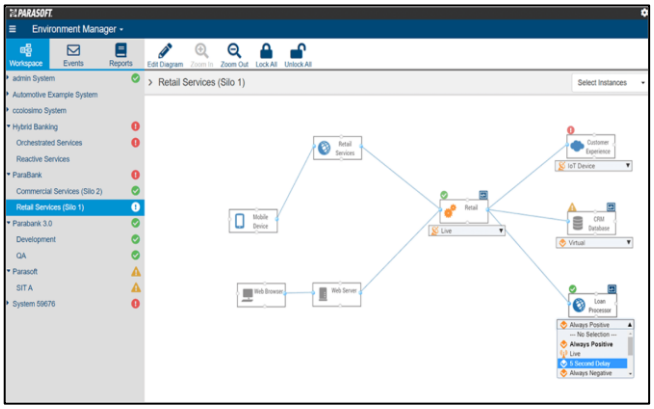
3) Parasoft

Parasoft Virtualize offers significant benefits for service virtualization, including:

- **Anytime Access**: It enables the creation of virtual services that mimic real dependencies for application testing, ensuring access even when actual services are unavailable.
- **Dependency Isolation**: The tool replicates the behavior of interdependent services, allowing for accurate testing without the need for the actual services to be accessible.
- **Stable Testing Environment**: Virtual services contribute to a more stable and reliable testing process, reducing the risk of test failures due to unstable or incomplete services.
- **Codeless Interface**: With its user-friendly interface, Parasoft Virtualize allows testers to create virtual services without scripting. For those who prefer or require scripting, it supports multiple programming languages.
- **Realistic Service Replication**: The tool studies and models the interactions between your application and live services to ensure that the virtual services behave realistically.

Overall, Parasoft Virtualize is a powerful ally for testers and developers, providing a robust platform for simulating services and data, facilitating seamless integration with

testing teams, and enhancing the efficiency of failure detection and debugging.



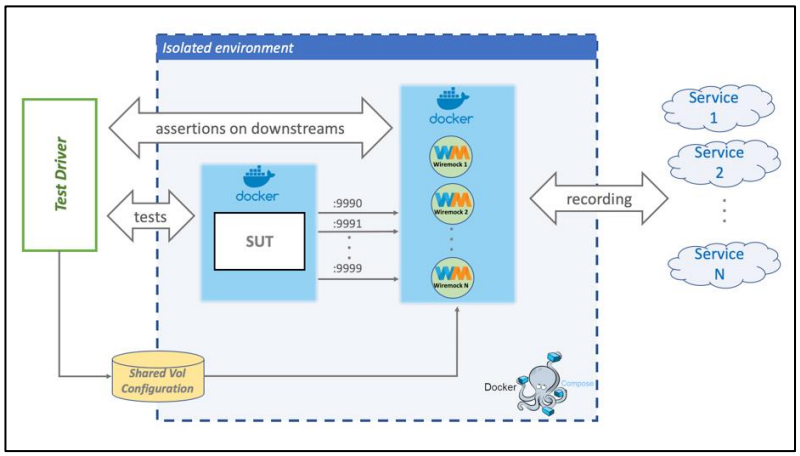
The above picture explains the Virtual model is being used in parasoft.

4) Wire Mock

WireMock is an indispensable tool for creating mock APIs, particularly useful during the development and integration testing phases. It is essentially a library that allows for the stubbing and mocking of web services, creating an HTTP server that mimics a real web service. Users can establish expectations, interact with the service, and confirm behaviors while the WireMock server is operational.

The system boasts several key features:

- Advanced Request Matching: This feature allows users to set intricate rules that precisely match incoming requests, ensuring a high level of specificity and control.
- Dynamic Response Templating: It provides the flexibility to tailor responses based on the parameters of the request, enabling a more dynamic and personalized interaction.
- Fault and Latency Injection: With this, users can introduce faults or delays into the system, simulating real-world scenarios for testing purposes.
- Record/Playback: This functionality captures real API interactions, which can then be replayed for testing or documentation purposes.
- Multi-Language Support: WireMock is accessible as a standalone service or as a Java library, with integrations for various modern languages and technology stacks.



The above picture explains the Virtual model is being used in Wiremock.

5) Mountebank:

This open-source tool facilitates multi-protocol testing and is developed using Node.js, which simplifies the creation of stubs and mocks. It offers support for SMTP, HTTP, TCP, and HTTPS protocols. Additionally, it is fully cross-platform compatible and benefits from regular updates.

6) Hoverfly Cloud:

Crafted for seamless integration, automation, and enhanced performance, this solution is engineered to optimize virtualized services, ensuring efficient management of system loads. It is versatile and deployable across major cloud platforms including Google Cloud, AWS, and Azure. Additionally, it features the capability to automatically provision virtualized services, streamlining the process as an integral part of the test setup.

4. Implementing DB Virtualization

Setting up a database service virtualization environment is a crucial step in ensuring the success of your software development life cycle (SDLC) projects. Managing data quality in virtualized environments has become a critical aspect of ensuring the success of software development life cycle (SDLC) projects. By defining clear guidelines for data collection, storage, and usage, SDLC professionals can ensure that the data used in virtualized environments is accurate, consistent, and reliable. This includes implementing data validation processes to identify and correct any errors or inconsistencies in the data before it is used for testing or development purposes.

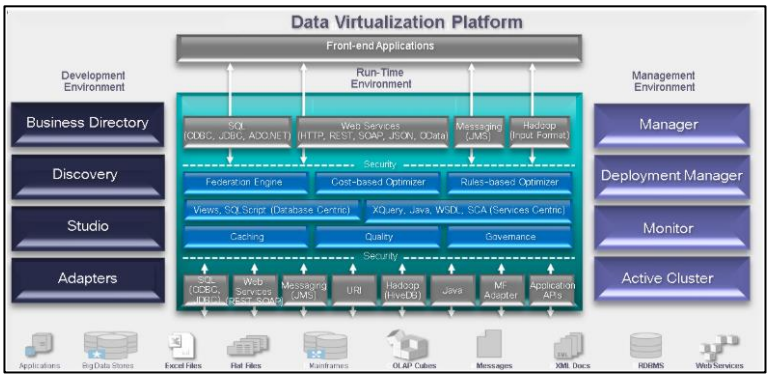
Tools for DB virtualization

Data virtualization tools simplify and expedite access to data stored in data warehouses, databases, and files located on-premises and in the cloud. These tools connect multiple data sources and centralize data acquisition logic in a metadata layer, creating a single source of data for data consumers1. Here are some notable data virtualization tools:

1) **CData**: CData provides embedded data virtualization solutions by focusing on eliminating data bottlenecks and providing a unified data access layer across various data sources. It has the following capabilities. CData's Embedded Data Virtualization (DV) solution is designed to integrate a data virtualization layer within existing systems, establishing a universal data access layer that

ensures seamless connectivity among diverse data sources. This embedded DV layer serves as a logical data warehouse, managing real-time or near-real-time data from various sources, both on-premises and cloud-based. It effectively removes data bottlenecks and facilitates self-service data services tailored for various business scenarios and workloads. Additionally, CData's Driver technologies bolster virtualized data connectivity, aligning with widely used data access standards including ODBC, JDBC, ADO, and Python. These drivers are easily adaptable for use with federated queries, SQL Server linked servers, Polybase, and other prevalent data virtualization technologies.

2) **Denodo Data Virtualization** provides a robust platform for users to collect, manage, and analyze data from diverse sources, including databases, APIs, and systems, in real-time. This approach streamlines data integration, minimizes complexity, and boosts data management agility.
3) **IBM Cloud Pak** for Data encompasses data virtualization features, offering a cohesive view of data from multiple sources. This facilitates effortless data access and integration, complemented by data discovery, governance, and metadata management capabilities.
4) **Oracle Virtualization** delivers a data virtualization solution that enables the integration and access of data from assorted sources without necessitating physical data movement. It ensures a uniform data perspective, simplifying the handling of data assets for organizations.
5) **TIBCO Data Virtualization** is geared towards data integration, presenting a consolidated data view across different platforms. It aids organizations in cutting costs, accelerating processes, and increasing the overall agility of their data management practices. The TIBCO Data Virtualization is a data virtualization server that connects to existing data, federates disparate data, abstracts complex data, and delivers the information as data services. The server includes a graphical development environment that lets you rapidly design and develop database-centric objects, including relational views and service-oriented objects. The TIBCO Data Virtualization also includes a complete set of management capabilities.



The above picture explains the Virtual model is being used in TIBCO data virtualization.

5. Results and Discussion

This paper explains the summary of what is being discussed as part of model and analysis.

API service virtualization

Service virtualization has revolutionized software development and testing, particularly for API testing. It enables teams to emulate external systems, facilitating application testing in a real-world setting without actual dependencies. This approach is instrumental in microservices architecture, where applications are segmented into smaller, intercommunicating components. Teams can mimic these components' behaviors, allowing for comprehensive interaction testing without full system deployment.

For cloud-based applications, service virtualization is invaluable, simulating dependencies to test applications in a controlled environment. This ensures performance meets expectations upon cloud deployment. Furthermore, service virtualization streamlines API testing automation within the SDLC, simplifying the creation and dismantling of test environments, enhancing testing frequency and depth, and ultimately improving software quality.

Integrating service virtualization with continuous testing practices further solidifies API testing within the SDLC, ensuring extensive and consistent testing throughout the development process.

Database Service Virtualization

In virtualized database design, performance impact is a critical factor. It's important for SDLC professionals to evaluate and ensure that virtualized databases meet application performance needs without detriment. This may require configuration optimization, resource tuning, and the use of performance monitoring tools.

Data quality management is vital in virtualized environments for SDLC project success. Establishing governance policies, utilizing machine learning tools, and continuous data quality monitoring can help maintain accuracy and consistency.

Addressing performance issues in database service virtualization involves a thorough approach, including understanding the root causes, optimizing resources, enhancing query processing, and resolving network issues. Machine learning tools can aid in proactive optimization and timely problem-solving.

For SDLC professionals, mastering data synchronization in machine learning-enhanced database service virtualization is crucial. This entails grasping synchronization challenges and establishing strong processes for synchronization.

6. Conclusion

The paper discusses the necessity of API and database service virtualization, various methodologies, and the tools utilized in the market. It also examines how these elements contribute to enhancing the Software Development Life Cycle (SDLC) process and facilitate an earlier transition to production.

When explains about **API service virtualization**, this paper gives the direction for choosing correct tool based on the need of the hour. It also talks about emerging trends in service virtualization like the use of **cloud-based tools.** Cloud-based service virtualization tools offer a more scalable and flexible solution for testing teams, allowing them to quickly spin up virtual services in a cloud environment without the need for expensive hardware or infrastructure. This trend is particularly beneficial for organizations looking to streamline their testing processes and reduce costs. Another trend is **automation of virtual services** in the SDLC. Automation of service virtualization allows testing teams to create and manage virtual services more efficiently, enabling them to quickly adapt to changing requirements and scale their testing efforts as needed. This trend is essential for organizations looking to increase their testing speed and efficiency while maintaining high-quality standards.

When explaining **database service virtualization**, the paper gives direction that database service virtualization is a technique that allows SDLC professionals to simulate the behavior of databases in a controlled environment. By creating virtual copies of databases, developers can test their applications without relying on the actual database, leading to faster and more efficient software development cycles.

Then discussed the benefits of using machine learning-enhanced database service virtualization. By harnessing the power of machine learning algorithms, SDLC professionals can automate the process of creating virtual databases and predicting the behavior of real databases. This can lead to more accurate testing results and improved software quality.

Then explored best practices for implementing database service virtualization in the software development lifecycle. These include creating realistic virtual databases, integrating virtualization into existing testing processes, and leveraging automation tools to streamline the virtualization process. By following these best practices, SDLC professionals can ensure the success of their virtualization efforts.

In conclusion, provides a comprehensive overview of database service virtualization and its applications in software development. By understanding the key takeaways from this book, SDLC professionals can enhance their skills and improve the efficiency of their software development processes. Whether you are new to database service virtualization or looking to enhance your existing knowledge, this book is a valuable resource for all SDLC professionals.

7. Recommendation for future result

API service virtualization

Incorporating service virtualization within the Software Development Life Cycle (SDLC) significantly enhances testing procedures and boosts software quality. This subchapter outlines essential strategies for effective integration of service virtualization into the SDLC.

Engaging all stakeholders in the implementation process is crucial. Developers, testers, project managers, and other pertinent personnel must understand the advantages of service virtualization to foster support and commitment.

Selecting appropriate service virtualization tools is also vital. Cloud-based options provide scalability and adaptability beneficial for testing. It's imperative to assess various tools against your specific needs and criteria.

Automating service virtualization within the SDLC is fundamental for realizing its full potential. Automating the generation, deployment, and management of virtual services conserves time and resources, enhancing test efficiency. Integrating service virtualization into the continuous testing pipeline optimizes the process and facilitates earlier bug detection.

Moreover, service virtualization should be part of API testing. Virtualizing APIs enables the simulation of real-world conditions and interdependencies, leading to more comprehensive and precise testing, thus elevating software quality and dependability.

Finally, service virtualization's role in performance and microservices testing within the SDLC should be examined. Simulating diverse performance scenarios and dependencies aids in early identification and resolution of performance issues.

Database service virtualization

The outlook for database service virtualization is highly optimistic for professionals in the Software Development Life Cycle (SDLC). As technological advancements persist, the demand for more efficient and effective testing methodologies is set to rise. Database service virtualization presents a viable solution to the hurdles that developers and testers encounter when establishing realistic testing environments. It achieves this by emulating the behaviors of dependent systems, thereby enabling SDLC professionals to refine their testing procedures and elevate the caliber of their software products.

A significant trend on the horizon for database service virtualization is the incorporation of machine learning. By utilizing machine learning algorithms, the precision and efficiency of virtualized databases can be augmented through the analysis of data trends and the anticipation of results. This advancement is expected to yield simulations that are more lifelike and comprehensive in terms of test coverage, which in turn, is anticipated to enhance the overall quality of software applications.

Scalability is another critical factor in the realm of database service virtualization. With the growing adoption of agile and DevOps methodologies, the capacity to expand virtualized databases in response to the fast-paced changes in the environment is imperative. SDLC professionals must be

ready to modify their virtualization tactics to support the continuous evolution of their organizational needs.

Moreover, the progression of database service virtualization is likely to see a more profound integration with cloud computing technologies. Cloud-based virtualization solutions provide superior flexibility and accessibility, permitting SDLC professionals to effortlessly establish and administer virtualized databases across various platforms. This integration is poised to simplify the testing workflow and foster enhanced cooperation among development teams.

In summary, the prospective developments in database service virtualization promise substantial benefits for SDLC professionals. Keeping abreast of upcoming trends and technological innovations will be essential for these professionals to fully harness the advantages of virtualization within their testing frameworks.

References

[1] Antova, L., Bryant, D., Cao, T., Duller, M., Soliman, M A., & Waas, F. (2018, May 27). Rapid Adoption of Cloud Data Warehouse Technology Using Datometry Hyper-Q. https://doi.org/10.1145/3183713.3190652

[2] Bala, M., Boussaïd, O., & Alimazighi, Z. (2016, October 1). Extracting-Transforming-Loading Modeling Approach for Big Data Analytics. https://doi.org/10.4018/ijdsst.2016100104

[3] Barrett, D., & Kipper, G. (2010, January 1). How Virtualization Happens. https://doi.org/10.1016/b978-1-59749-557-8.00001-1

[4] Benguria, G., Alonso, J., Etxaniz, I., Orue-Echevarria, L., & Escalante, M. (2018, January 1). Agile Development and Operation of Complex Systems in Multi-technology and Multi-company Environments: Following a DevOps Approach. https://doi.org/10.1007/978-3-319-97925-0_2

[5] Kumar, R., & Goyal, R. (2020, October 1). Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC). https://doi.org/10.1016/j.cose.2020.101967

[6] Gunawan, F., & Yazid, S. (2020, September 19). Improving Digital Forensic Readiness in DevOps Context: Lessons Learned from XYZ Company. https://doi.org/10.1109/isemantic50169.2020.9234194

[7] What is Data Virtualization? (2014, August 18). https://www.denodo.com/en/data-virtualization/overview

[8] Why Data Masking. (2020, December 14). https://mask-me.net/datamaskingwiki/wiki/44/why-data-masking

[9] The Total Economic Impact™ of Data Virtualization Using the Denodo Platform. (2021, October 20). https://www.denodo.com/en/document/analyst-report/forrester-study-total-economic-impact-data-virtualization-using-denodo

[10] Best DevOps Online Course | DevOps Training Course | ITGuru. (2023, January 23). https://onlineitguru.com/devops-training.html

[11] Brust, A J. (2018, December 10). Revenge of the Data Warehouse: How a Classic Tech Category Has Evolved, and Triumphed. https://research.gigaom.com/report/revenge-of-the-data-warehouse-how-a-classic-tech-category-has-evolved-and-triumphed/

[12] Business Continuity Plan. (2016, February 2). https://www.slideshare.net/RageshRNair2/business-continuity-plan-57764822

[13] Customer Success. (2014, September 22). https://www.denodo.com/en/denodo-platform/services/overview

[14] Data governance overview. (2023, January 10). https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/scenarios/cloud-scale-analytics/govern

[15] Data Masking in the World of GDPR. (2020, May 15). https://dbexamstudy.blogspot.com/2020/05/data-masking-in-the-world-of-gdpr.html

[16] Data masking: techniques & best practices. (2020, February 3). https://www.datprof.com/data-masking/

[17] Data security. (2004, November 11). https://en.wikipedia.org/wiki/Data_security

[18] Database Virtualization – Driving the Next Big Transformation. (2023, April 4). https://devops.com/database-virtualization-driving-the-next-big-transformation/

[19] Datamatica - Accelerated data intelligence. (n.d). https://data-matica.com/

[20] DataVantage. (2023, January 1). https://datavantage.com/

[21] Denodo - The leading platform for delivering data in the language and speed of business. (2014, September 3). https://www.denodo.com/en

[22] Denodo Platform. (2018, April 11). https://www.denodo.com/en/denodo-platform/denodo-platform-80

[23] Devops Development and Services. (2019, August 14). https://www.ogsoftwaresolutions.com/devops_development_company/

[24] distributed | Scrum Agile Project Management Expert. (2022, December 5). https://www.scrumexpert.com/tag/distributed/

[25] Grześ, B. (2023, January 1). Managing an agile organization – key determinants of orgnizational agility. https://doi.org/10.29119/1641-3466.2023.172.17

[26] Guide Your Test Environment Management Through Service Virtualization. (2017, August 9). https://www.slideshare.net/Enov8ecosystem/guide-your-test-environment-management-through-service-virtualization-enov8

[27] HCL | Denodo. (2015, August 31). https://www.denodo.com/en/partner/hcl

[28] Home. (2021, December 19). https://www.vmcompute.com/

[29] How Cloud Operations Services Are Changing the Business Landscape. (2023, January 17). https://www.zymr.com/blog/the-evolution-of-business-world-with-cloud-operations-services

[30] How to Analyze Your MDM Strategy for Better Business Outcomes. (2022, May 24).

https://www.digitaldoughnut.com/articles/2022/may-2022/how-to-analyze-your-mdm-strategy
[31] I am a DBA! ¿what do I need to know to understand the application of DevOps practices for Databases?. (2021, January 26). https://cmcarlosmoreno.medium.com/i-am-a-dba-913f92c0987c
[32] Informatica dynamic data masking pdf. (2014, June 9). https://lesscompchowsvogt.web.app/1432.html
[33] Ivanov, T., Petrov, I., & Buchmann, A. (2012, July 1). A Survey on Database Performance in Virtualized Cloud Environments. https://doi.org/10.4018/jdwm.2012070101
[34] Lange, T A P., Cemim, P., Xavier, M G., & Rose, C A F D. (2013, July 1). Optimizing the management of a database in a virtual environment. https://doi.org/10.1109/iscc.2013.6755012
[35] Limoncelli, T A. (2018, December 19). SQL is no excuse to avoid DevOps. https://doi.org/10.1145/3287299
[36] Personal data anonymization: key concepts & how it affects machine learning models. (2020, June 11). https://tryolabs.com/blog/2020/06/11/personal-data-anonymization-key-concepts--how-it-affects-machine-learning-models
[37] Privacy Enhancing Technologies. (2022, March 1). https://magedata.ai/platform/privacy-enhancing-technologies/
[38] Product and Services Overview. (2017, March 3). https://www.denodo.com/en/denodo-platform/overview
[39] SAP HANA Cloud. (2022, November 16). https://www.sap.com/india/products/technology-platform/hana.html
[40] SAP HANA Modelling. (2019, November 8). https://www.automaticinfotech.com/sap_hana_modelling/
[41] Save Time and Money with Data Virtualization. (2018, June 8). https://www.denodo.com/en/document/brochure/infographic-save-time-and-money-data-virtualization
[42] Smart Decision Analytics. (2000, January 1). https://sdanalytics.in/
[43] Sotiropoulos, D M. (2013, January 1). ARTICLE 29 Data Protection Working Party. https://ec.europa.eu/justice/data-protection/article-29/documentation/other-document/files/2013/20131211_letter_to_greek_presidency_en.pdf
[44] TheDBAdmin. (2022, November 25). https://thedbadmin.com/