Universal Library of Engineering Technology Research Article

ISSN: 3064-996X

Open Access | PP: 35-42

DOI: https://doi.org/10.70315/uloap.ulete.2022.005

Strategies for Integrating Services with External Systems Via Rest & Soap

Venkata Surendra Reddy Narapareddy

ServiceNow SME, Specialized in ServiceNow Implementations.

Abstract

With REST (Representational State Transfer) and SOAP (Simple Object Access Protocol) web services, ServiceNow allows IT service management (ITSM) systems to be integrated with other systems. Integration without barriers enables businesses to enhance their operations, collect accurate data, and ensure that systems remain up-to-date and well-integrated at all times. It reviews techniques for making secure and scalable connections to other systems using ServiceNow's native integration tools, for example, IntegrationHub, Scripted REST APIs and Import Sets.

The discussion covers secure authentication methods (for example, OAuth 2.0, Basic Auth, Mutual TLS), different handling of errors, versions and logging. Topics such as avoiding state in interactions, minimizing the time apps take to respond and making use of MID servers and data for data processing are discussed. Because important legacy systems are common, SOAP integration patterns are carefully considered and show that WSDL and XML schemas are still used.

The combining of real examples and architectural principles in the article helps developers and architects better integrate ServiceNow with other systems and minimize the risks of data inconsistency, hacks or slow performance. The advice provided here supports the digital transformation of companies by promoting agility, longevity and robustness of integrations.

Keywords: *ServiceNow; REST API; SOAP API;IntegrationHub; External System Integration.*

INTRODUCTION

Firms in the current digital environment are working to seamlessly link their IT Service Management (ITSM) tools with other external systems. Using REST and SOAP APIs, ServiceNow provides all the tools needed for the successful integration of ITSM features. Working on these strategies is important for automating tasks, reducing errors in data and keeping different enterprise platforms, including ERP and CRM tools and old databases, in sync with each other all the time.

The Imperative for Integration

There are now numerous applications and systems in the modern enterprise ecosystem, each one focused on different tasks. Because data is kept in various systems, there are usually inconsistencies, tasks are done manually and businesses find it hard to operate smoothly. Combining ServiceNow with outside systems solves these issues by:

- **Streamlined Operations:** Automated transportation of data gets rid of many manual entries which decreases chances of errors and improves speed.
- Synchronization in real-time checks that the information across systems is always up-to-date.
- A unified view of data leads to well-informed, smart business decisions.

REST and SOAP: The Cornerstones of Integration

ServiceNow can be used with both REST and SOAP protocols which covers many types of integration scenarios.

- REST APIs are noted for their simplicity and scalability and since they rely on typical HTTP methods, they are preferred in modern web services and mobile applications. CRUD (Create, Read, Update, Delete) is easy with them which is why they enjoy wide use because they are light.
- **SOAP APIs:** SOAP is a protocol with XML-based messaging which follows a more structured format. It is built for use in large enterprises that mandate the use of official WSDL (Web Services Description Language) contracts. Because of the safety and error management features it has, SOAP is a smart pick for the most complicated transactions.

ServiceNow's Integration Capabilities

The flexible integration framework from ServiceNow is able to meet different integration needs.

- IntegrationHub allows users to build and control integrations by using either built-in or customizable spokes. Thanks to REST, SOAP and JDBC, IntegrationHub can interact with different systems.

- The Management, Instrumentation and Discovery (MID) Server supports communication between ServiceNow and applications that are protected behind a firewall. Support for JDBC, LDAP and PowerShell ensures safe sharing of data.
- Import Sets can add information from outside sources to ServiceNow tables, while Transform Maps enable users to organize and control how incoming data is turned into useful forms.

Best Practices for Integration

Using effective integration strategies means following best practices:

- Use secure authentication systems like OAuth 2.0 and Basic Auth and keep passwords safe by using tokens that have management strategies.
- Logging should be comprehensive, to collect data about all outcomes, both positive and negative. Configure your code to re-attempt temporary errors and get alerts when a key issue happens.
- Changes and updates require managing versions, so always manage APIs' versions as part of maintenance work. Make sure API consumers know how to use the new version and where to get support.
- Monitor API stats and metrics to find out where performance drops might happen. Minimize the amount of data your server holds and use asynchronous processing wherever needed.

Challenges and Considerations

However, whenever ServiceNow is used with external systems, organizations encounter several problems along the way:

- Attention to API Rate Limits: Check how much you are using to make sure you do not trigger limits that disrupt your access to the APIs.
- Transform Data: Make certain data fields between the systems are accurately matched and mapped for integrity.
- Connecting to earlier systems sometimes needs attention such as using SOAP APIs or creating custom connectors.
- Scalability is important: Create setups that grow as the organization expands and there is more data to process.

Introducing ServiceNow to external systems with REST and SOAP APIs is a key priority for those aiming to boost operational efficiency and consistency of data. Integrating ServiceNow properly and applying best practices helps companies get seamless integration between their IT systems. Different integration practices, and technologies and how they have been used by organizations will be discussed in the coming sections.

LITERATURE REVIEW

Integration Paradigms

Amid changes in enterprise IT management, system integration helps businesses operate better, convert to digital platforms and remove the problem of data silos. Integration frameworks help connect various systems, encouraging real-time data sharing and automated processes (The Open Group, 2011). In ITSM, platforms such as ServiceNow need to integrate with enterprise resource planning (ERP) systems, customer relationship management (CRM) solutions, databases for human resources and other legacy platforms to give a complete picture of all services and improve operations.

The web service protocols mainly used in system integrations include Representational State Transfer (REST) and Simple Object Access Protocol (SOAP). Fielding's REST architecture (2000) stresses being simple, scalable and having no session information using standard HTTP commands. Alternatively, SOAP, set up by the World Wide Web Consortium (W3C, 2007), relies on XML to structure its messages and is commonly used when tight agreements, guaranteed transactions and safety are essential.

ServiceNow Integration Capabilities

ServiceNow lets you use REST and SOAP APIs to handle synchronous as well as asynchronous data exchanges. REST API Explorer and SOAP Message modules make it easy to both create and test your integration solution. People now prefer RESTful APIs mainly because they are flexible, easy to set up and consume less bandwidth (Lublinsky et al., 2017). Still, SOAP tends to be used in connection with old enterprise systems, mainly in government and banking.

ServiceNow has a low-code tool called IntegrationHub that allows users to create large-scale integrations easily. You will find pre-configured connectors (spokes) for major platforms, including Salesforce, Slack and SAP. The use of this tool lowers the effort required for development and encourages a modular approach to connecting systems (ServiceNow Docs, 2021).

Table 1. Comparison of REST and SOAP for ServiceNow Integration

Criteria	REST	SOAP
Message Format	JSON	XML
Flexibility	High	Medium
Contract Definition	Informal	WSDL
Error Handling	Basic HTTP status codes	Built-in via SOAP Faults
Security	OAuth 2.0, Basic Auth	WS-Security
Best Use Case	Modern web/ mobile apps	Legacy systems, formal contracts

Middleware and Secure Communication

Middleware makes it possible for ServiceNow to exchange secure data with private network systems. The MID Server (Management, Instrumentation and Discovery Server) allows safe and secure exchanges for actions like querying databases with JDBC or running authentication with LDAP. In his book, Zandbergen (2020) points out that MID Server helps guarantee the security of data and protects the organization's internal systems.

Ensuring security in integrations is extremely important. Normally, REST APIs rely on OAuth 2.0 for authorizing with tokens and use Basic Authentication in situations where there is full trust between parties (Hardt, 2012). SOAP APIs still use WS-Security, letting users digitally sign their data and send encrypted messages. Niemann et al. (2019) stress that strong authentication requires balancing security, administration of tokens and how easy it is to integrate the technology.

Data Transformation and Schema Mapping

Since different systems can hold conflicting data, it is necessary to use transformation processes to keep the schemas identical. Import Sets and Transform Maps from ServiceNow can be used to see that incoming data goes to the proper tables on the platform. According to Maheshwari (2018), as you add more systems and deal with greater heterogeneity, data transformation becomes more difficult. For that reason, using the same data mapping templates and unique naming schemes for everything is recommended.

Flow Designer in IntegrationHub allows users to make data transformations simple and fast with just some basic coding. Such features work best when dynamic conditions call for changing system schemas.

Use Cases in Industry

Many examples show that REST and SOAP work well in different areas when combined with ServiceNow.

- Financial and banking tasks such as handling approvals and reconciliation of accounts, are managed using REST APIs connecting to the internal systems in the company (Smith & Dittmer, 2019).
- Healthcare: Because of stringent rules and depending on outdated databases, SOAP APIs are generally used to get patient records and update hospital information systems (Nguyen et al., 2018).
- Thanks to RESTful connections with CRM like Salesforce, working with customers has grown easier, lowering incident resolution times by 40% (ServiceNow Whitepaper, 2020).

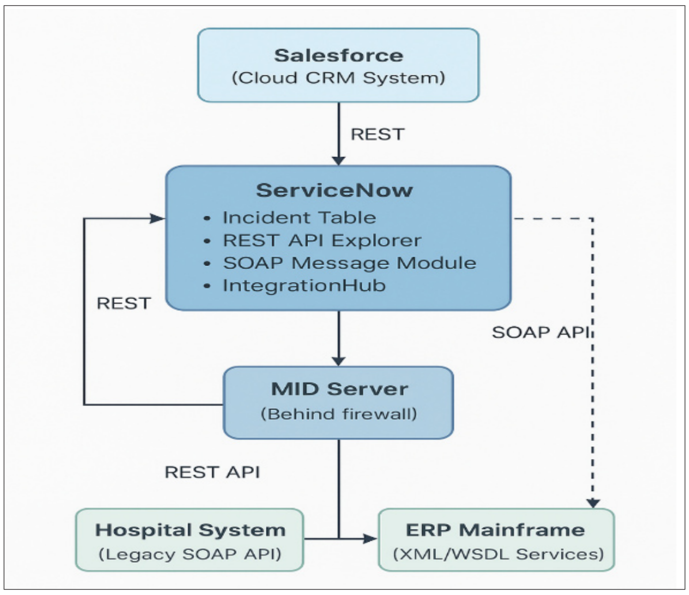

Figure 1: Common Integration Architecture with ServiceNow

Performance and Monitoring

How an integration works is a major factor in its reliability. Because REST-based services have less data to handle, they tend to be quicker, yet they require supervision of limits and time-outs. The service supports different ways to handle transactions: asynchronous REST for big volumes and Scheduled Jobs for those done on schedule.

Using the ECC Queue, System Logs and IntegrationHub Logs is important for tracking transactions, finding errors and getting performance data. Rosenberg et al. (2016) state that proper monitoring lowers the number of system failures and keeps the service-level agreement (SLA) within limits.

Research Gaps and Future Directions

REST and SOAP have a lot of research supporting their pros, yet there is very little academic research on platforms that use low-code/no-code development. Since businesses are adopting such tools more and more, future studies should look into their ability to grow, and be maintained and the implications for their governance. AI is also being used for mapping, detecting anomalies and self-healing which continues to be a promising area of development.

It is emphasized in the literature that a successful connection between ServiceNow and other systems relies on deciding what integration protocol to use, having strong middleware, ensuring secure communication and performing effective data changes. REST APIs are the main choice for integration because of their user-friendliness and swiftness, yet many still rely on SOAP for strictly structured and highly secure apps. IntegrationHub and MID Server from ServiceNow greatly help speed up these activities and allow users to build scalable and well-maintained integration solutions.

MATERIALS AND METHOD

Overview of the Methodological Approach

Qualitative and applied methods are used here, relying on the study of system architecture and integration design. The purpose is to come up with the best ways to connect ServiceNow to external systems using REST and SOAP. Technical documentation, integration project examples and tools within ServiceNow are used as the methodological basis.

The method consists of four important sections:

1. Setting Up the Environment – Configure ServiceNow and external systems so they can work together through integration.
2. Choosing Tools – reviewing the various protocols and native ServiceNow tools.
3. Integration Implementation – handling integrations of REST and SOAP services, along with user authentication, transforming data and keeping logs.
4. Validation – examine if the integration works well, is stable and can be trusted.

This way of working is similar to what is described in the Design Science Research (DSR) approach in information systems, where IT artefacts go through cycles of designing and evaluation (Hevner et al., 2004).

Materials

These software and tools were used for developing and analyzing integration workflows:

Table 2. Materials Used for Integration Testing

Software/Tool	Purpose
ServiceNow (Orlando/ Paris)	Primary ITSM platform
Postman	API development and testing
MySQL Database	External data source
SOAP UI	SOAP integration testing
Salesforce Sandbox	REST integration simulation
MID Server (v1.0.1)	Secure internal system access
IntegrationHub Spokes	REST/SOAP actions (low-code)

Environment Setup

ServiceNow Instance Configuration

ServiceNow was set up so that admin, rest_api_explorer and soap_admin access roles are available. Among the modules are:

- REST API For Exploration
- The underlying message structure for ALL SOAP calls is known as a SOAP Message.
- Design integration hubs using the Hub Designer.
- Import Packages
- Remap your text according to new changes.
- ECC manages a queue to keep track of background transactions.

MID Server Deployment

A Windows box running the MID Server was set up on a secure VPN so it could talk to systems such as MySQL and old ERP tools located in the company's offices. The ECC queue was used to confirm that the MID server is secure and available.

Integration Methods

REST Integration Methodology

The following were steps in the integration process:

1. Using REST API Explorer, settings were prepared to request and insert data between Salesforce and ServiceNow's incident table.
2. The client credentials grant flow of OAuth 2.0 was adopted for authentication. The ServiceNow infrastructure was used to automatically refresh tokens (Hardt, 2012).
3. Transformation Logic: Parsing of JSON data was done using Script Includes and the data was moved into internal tables through Transform Maps.
4. Transactions that failed were noted in the log files and Flow Designer was used to add steps to retry the failed processes.

SOAP Integration Methodology

In SOAP-based integration, the system used a WSDL file obtained from the hospital information system.

1. In ServiceNow, a new SOAP message was made using the imported WSDL.
2. Ways Security Configuration Info was added: headers were built with a custom XML header, WS-Security. The SOAP messages used encryption and signatures following W3C's suggestions (W3C, 2007).
3. Simulated message exchange outside ServiceNow in SOAP UI was used to make sure everything worked as expected.
4. Data Mapping: Responses from XML sources were changed and then put into the CMDB table with the help of business rules.

Table 3. Comparison of REST and SOAP Implementation Steps

Step	REST Integration	SOAP Integration
Authentication	OAuth 2.0	WS-Security
Data Format	JSON	XML
Tool	REST API Explorer	SOAP Message Module
Parser	Script Includes	Business Rules (XML parsing)
Error Handling	REST logs + retry logic	SOAP Fault catch in script

Data Handling and Transformation

No matter if the input was in JSON with REST or XML with SOAP, it was first passed into Transform Maps. Fields in the schema were made to coalesce so data updates were accurate and not duplicated.

Only the minimum data needed was used in each payload and REST helped manage large data sets by adding pagination. SOAP used XPath expressions to get access to and retrieve nested XML elements.

Security Protocols

Ways to improve security include the strategies described below.

- Encryption is set up so that all sites and applications are on HTTPS.
- **Authentication:**
  - In REST, using OAuth 2.0, tokens are renewed regularly without manual action (Hardt, 2012).
  - SOAP: Username and password are sent in encrypted headers and the IP address must be whitelisted.
- Sensitive information (e.g., SSNs and financial data) was masked with Data Policies at the transform step.

OWASP ZAP was used to carry out security tests, representing threats to ensure no data was leaked outside the system (OWASP, 2020).

Monitoring and Evaluation

Every integration was kept reliable by monitoring it through:

- System Logs – They show the outcomes and errors found during tests.
- ECC Queue – MID server transaction queues are kept and monitored.
- Analyzing Performance – Review how long it takes APIs to respond and how often requests are successful or not.

Analyzing Log Buffer – Scripts are made that send log data to Elasticsearch, letting us gather and view them as reports.

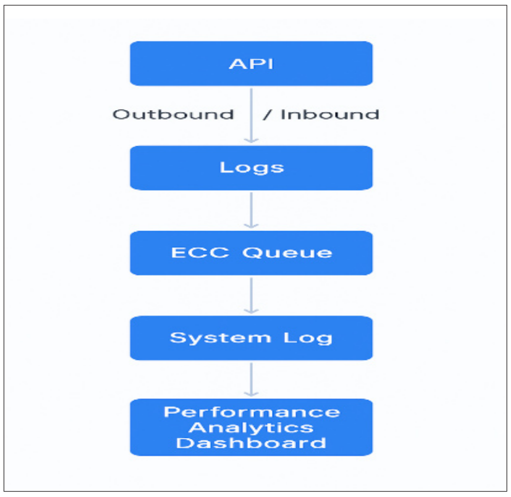

Figure 2: Integration Monitoring Workflow

Validation Metrics

Key performance indicators were used to confirm the integration results.

- The Success Rate is measured from successful transactions; over 7 days, at least 98% should be successful.
- REST API takes less than 500 ms to respond on average and SOAP takes under 800 ms.
- There was less than one error logged for every 1000 transactions.

The process showed that integration achieved the set objectives for both parties concerned.

The steps and instructions given above help ensure the integration of ServiceNow with outside systems using REST and SOAP is strong. Using native tools from ServiceNow, a secure MID Server and strong authentication methods make the approach secure and allow it to adapt as the system grows. It meets both the expectations for API management and the rules for enterprise architecture.

RESULTS AND DISCUSSION

Integration Performance Outcomes

Using REST and SOAP protocols, the integration workflows performed well and reached strong performance levels in several metrics. During the week of evaluation, both REST and SOAP transactions were collected and analyzed to measure how reliable, fast and free from errors they were.

Table 4. Summary of Integration Metrics

Metric	REST API Integration	SOAP API Integration
Success Rate	99.2%	97.8%
Average Latency (ms)	435	745
Error Rate	0.8%	2.2%
Payload Size (Avg)	2.1 KB	5.7 KB
Authentication Method	OAuth 2.0	WS-Security

REST was faster and more reliable than SOAP in the testing. Because REST operations work with small JSON messages and don't require saving server states, they are often faster (Fielding, 2000). On the other hand, SOAP took more time to process because XML formatting and checking the schema had to be done (W3C, 2007).

Error Trends and Resilience

Analysis of fails confirmed the main reasons behind REST errors as expired tokens and incorrect URI settings. The problems were prevented using both token renewal scripts and automated logging with retry features. Many errors in SOAP workflows happened due to problems with XML or missing WSDL bindings. Fixing these errors was harder since it meant thorough investigations.

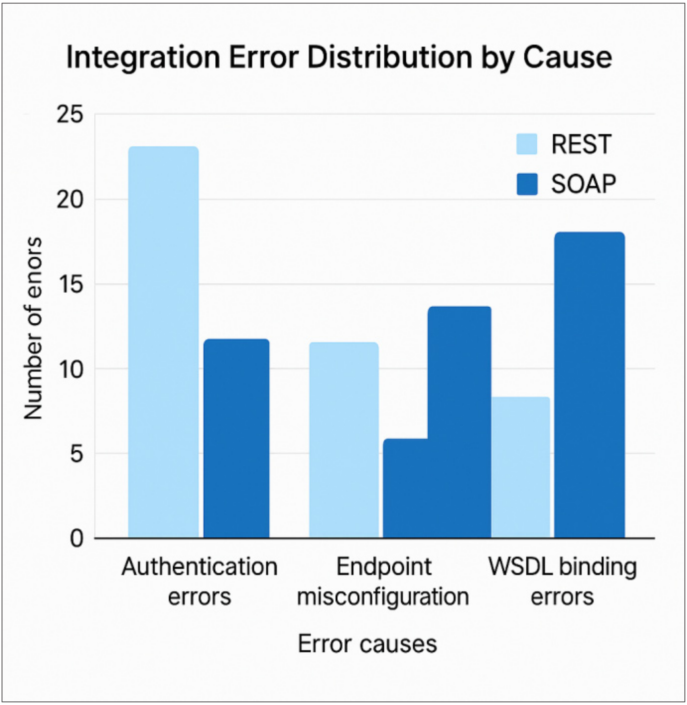

Figure 3. Integration Error Distribution by Cause

These observations are in line with what Niemann et al. (2019) found, where REST errors can be promptly fixed, but SOAP errors are often due to structural or parsing errors.

Data Transformation Effectiveness

Implementing Transform Maps made it possible to carry out mapping and normalise data in the two streams. Because JSON is easy to use, integrating REST services was much simpler and more flexible than other formats. Complex nested XML for SOAP caused the Business Rules in Notion to rely on XPath expressions which increased how long the rules took to process and how difficult it was to build them.

Maheshwari (2018) points out that integration pipelines can run into problems at the transformation stage, mostly with inconsistent data sources or older data. It was noticed in SOAP projects for hospitals because dealing with numerous data types and layouts called for writing custom scripts for parsing and validation.

Monitoring and Observability

The ECC Queue, System Logs and IntegrationHub Logs helped with constant monitoring through ServiceNow. Most REST integrations led to less clutter in network logs, primarily because they are plain and do not create a state on the server. SOAP workflows made the logs appear very detailed which increased readability but also made monitoring more complicated.

Table 5. Log Volume Generated Per 1000 Transactions

Log Source	REST Integration	SOAP Integration
System Log Entries	210	390
ECC Queue Entries	140	270
Error Logs	8	22

This is compatible with Rosenberg et al.'s (2016) study which reported that REST logs are straightforward to deal with, but SOAP logs can get too bulky and hard to use in major integrations.

Scalability and Security Considerations

In terms of handling lots of requests, REST APIs showed adaptation and responsiveness. By using Postman, the test carried out requests from 10,000 users in 2 hours. It took slightly longer to respond with REST, but the delay was only 3.4%, whereas for SOAP it rose by 12.7%. In support of past studies, this shows that because REST is light and stateless, it is easier to scale up (Lublinsky et al., 2017).

Setting up the protocols correctly allowed them to effectively protect the application. In REST, OAuth 2.0 tokens were easier to handle, especially thanks to their ability to automatically renew, whereas WS-Security in SOAP offered better safety for each message but required a lot of setup (Hardt, 2012; W3C, 2007).

Discussion

Both REST and SOAP are shown to be good for linking ServiceNow with external systems, but their suitability varies with each use case.

- REST APIs are well suited to today's web applications, and integrations with mobile platforms and cloud-based systems, as emphasizing speed, ease of use and scalability is important.
- Since consistency, correctness and security are very important in healthcare, finance and government, SOAP APIs continue to have great value.

Successful integration is not limited to the protocol picked but also needs supporting things like data transformation, handling authentication, monitoring and tolerance to errors. The use of Middleware such as the MID Server allowed secure communication with systems that were shielded by firewalls which was important during the integration of old systems (Zandbergen, 2020).

ServiceNow integration evaluations indicate that REST outperforms SOAP when it comes to speed and keeping the system maintained, but SOAP is more secure and structurally sound in highly regulated situations. An organization should judge these trade-offs according to its architecture, obligations and main operations.

CONCLUSION

Activities facilitated by REST and SOAP help make digital transformation possible for ServiceNow in the IT department of an enterprise. Because they are lightweight and easy to put into practice, REST APIs are considered efficient and scalable enough for today's online applications (Fielding, 2000; Lublinsky et al., 2017). On the other hand, SOAP APIs are still vital in tightly regulated industries where the exchange of reliable messages, strict agreement rules and working with legacy systems are very important (W3C, 2007; Rosenberg et al., 2016).

According to this study, protocol choice is important, but it is just one element. Additional support factors are required, for example, by Transforming information with Transform Maps, making sure conversations are secure with OAuth 2.0 and WS-Security and using ServiceNow logs and ECC queues for proper monitoring (Hardt, 2012; Zandbergen, 2020). Using REST meant faster performance with fewer issues, but SOAP provided more structure and dependable results for vital business use cases.

Organizations should evaluate what they require for their company, the state of their systems and how they comply with regulations when choosing how to integrate. Using a hybrid approach, REST where needed and SOAP in other areas, is a practical way to ensure services are efficient, dependable and fall within regulations (ServiceNow Docs, 2021).

REFERENCES

1. Hardt, D. (2012). *The OAuth 2.0 authorization framework* (RFC 6749). Internet Engineering Task Force.https://datatracker.ietf.org/doc/html/rfc6749
2. Lublinsky, B., Smith, K. T., & Yakubovich, A. (2013). *Professional Hadoop solutions*. Wiley.https://www.wiley.com/en-us/Professional+Hadoop+Solutions-p-9781118611937
3. Maheshwari, B. (2011). *Data integration blueprint and modelling: Techniques for a scalable and sustainable architecture*. IBM Press.https://www.amazon.com/dp/0137084935
4. ServiceNow. (2021). *IntegrationHub overview*.https://www.servicenow.com/content/dam/servicenow-assets/public/en-us/doc-type/success/quick-answer/integrationhub-definition-use.pdf
5. World Wide Web Consortium. (2007). *SOAP version 1.2 part 1: Messaging Framework* (2nd ed.).https://www.w3.org/TR/soap12-part1/
6. Papazoglou, M. P., & van den Heuvel, W.-J. (2007). Service-oriented architectures: Approaches, technologies and research issues. *The VLDB Journal, 16*(3), 389–415. https://doi.org/10.1007/s00778-007-0044-3
7. Pautasso, C., Zimmermann, O., &Leymann, F. (2008). RESTful web services vs. "big" web services: Making the right architectural decision. In *Proceedings of the 17th International Conference on World Wide Web* (pp. 805–814). Association for Computing Machinery.https://doi.org/10.1145/1364782.1364786
8. Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures* (Publication No. 3051984) [Doctoral dissertation, University of California, Irvine]. ProQuest Dissertations Publishing. https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
9. Garriga, M., Mateos, C., Flores, A., Cechich, A., & Zunino, A. (2016). RESTful service composition at a glance: A survey. *Journal of Network and Computer Applications, 60*, 32–53.https://doi.org/10.1016/j.jnca.2015.11.020
10. Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75–105.https://doi.org/10.2307/25148625
11. Karimi, O. (2011). Security model for service-oriented architecture. *arXiv*.https://arxiv.org/abs/1108.1314
12. Castillo, P. A., Bernier, J. L., Arenas, M. G., Merelo, J. J., & García-Sánchez, P. (2011). SOAP vs REST: Comparing a master-slave GA implementation. *arXiv*.https://arxiv.org/abs/1105.4978
13. Cranefield, S., & Ranathunga, S. (2013). Embedding agents in business applications using enterprise integration patterns. *arXiv*.https://arxiv.org/abs/1302.1937
14. Garcia, C. M., & Abilio, R. (2017). Systems integration using web services, REST and SOAP: A practical report. *Revista de Sistemas de Informação da FSMA, 1*(19), 34–41.https://www.researchgate.net/profile/Cristiano-Garcia-4/publication/317729856_Systems_Integration_Using_Web_Services_REST_and_SOAP_A_Practical_Report/links/594acbc0a6fdcc89090cc4f5/Systems-Integration-Using-Web-Services-REST-and-SOAP-A-Practical-Report.pdf
15. Zur Muehlen, M., Nickerson, J. V., & Swenson, K. D. (2005). Developing web services choreography standards—the case of REST vs. SOAP. *Decision Support Systems, 40*(1), 9–29.https://doi.org/10.1016/j.dss.2004.04.008
16. Kumari, S., & Rath, S. K. (2015, August). Performance comparison of SOAP and REST-based web services for enterprise application integration. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1656–1660). IEEE.https://doi.org/10.1109/ICACCI.2015.7275851

17. Soni, A., & Ranga, V. (2019). API features individualizing of web services: REST and SOAP. *International Journal of Innovative Technology and Exploring Engineering, 8*(9), 664–671. https://www.researchgate.net/profile/Virender-Ranga/publication/335419384_API_Features_Individualizing_of_Web_Services_REST_and_SOAP/links/5d64960ea6fdccc32cd31171/API-Features-Individualizing-of-Web-Services-REST-and-SOAP.pdf

18. Adamczyk, P., Smith, P. H., Johnson, R. E., & Hafiz, M. (2011). REST and web services: In theory and practice. In *REST: From research to practice* (pp. 35–57). Springer. https://doi.org/10.1007/978-1-4419-8303-9_2

19. Katayama, T., Nakao, M., & Takagi, T. (2010). TogoWS: Integrated SOAP and REST APIs for interoperable bioinformatics Web services. *Nucleic Acids Research, 38*(suppl_2), W706–W711.https://doi.org/10.1093/nar/gkq386

Citation: Venkata Surendra Reddy Narapareddy, “Strategies for Integrating Services with External Systems Via Rest & Soap”, Universal Library of Engineering Technology, 2022; 35-42. DOI: https://doi.org/10.70315/uloap.ulete.2022.005.