API Security

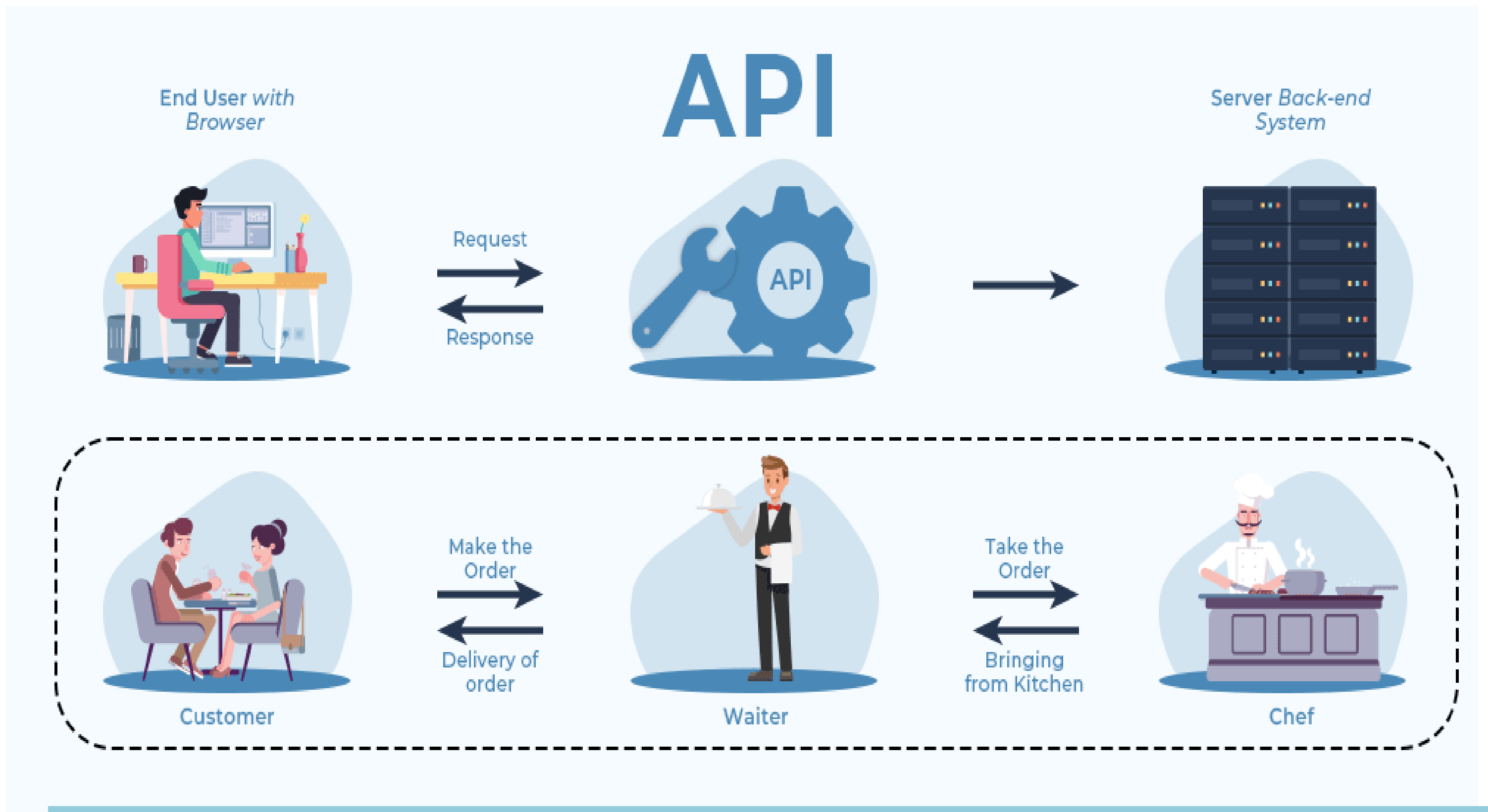
Kenneth Cher, Computer Information Systems
Mentor: Dr. Weijun Zheng
Student RSCA Conference 2025

RSCA
Research, Scholarship and Creative Activities

Introduction

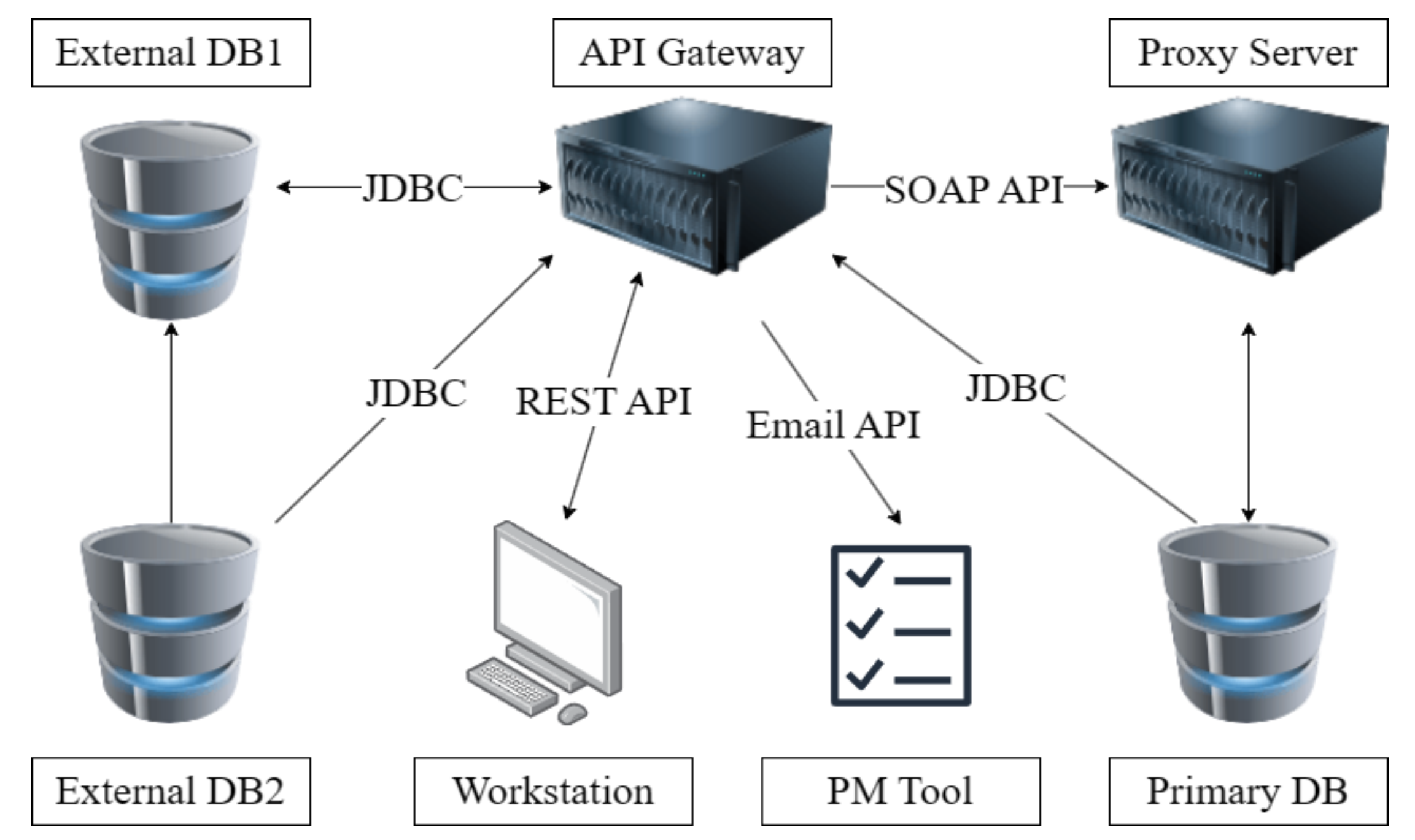
The purpose of this research is to give an introductory definition of what APIs are as well as an understanding of what they are and what they do. There are many different types of APIs so I will be comparing two popular APIs to see which one is better fitted for an enterprise environment in most cases. Once an API has been chosen, I hope to propose the best practices in using it as well as potential future use cases. Finally, I have developed and tested a simulation for students to perform penetration tests against a vulnerable API to give them a well-rounded experience in API security.

An application programming interface (API) is a software that performs a specific task and is a point where two applications can communicate. One side being the client and the other being the server. The client sends requests for data and can be a variety of devices ranging from cell phones to laptops and personal computers. A server, which is a large computer that stores lots of information, takes the client request and returns the appropriate data (See Figure 1). Although useful to many people and businesses, APIs can present new risks and vulnerabilities for bad actors to exploit. Companies should consider the following authentication methods, best practices and user access management to further increase the security of APIs in organizations.



API Gateways

There can be lots of APIs being used in an organization and one way to manage them is through an API gateway. This is a tool that serves as a reverse proxy between the user and the SP (RedHat). When a user sends in a request, the gateway will break the request down into multiple requests, interpret it, and ensure everything goes where it needs to go; all while logging everything. Gateways help centralize API usage for both the client and organization. They also act as an additional layer of security for APIs as further authentication services can be implemented. If an API gets abused or overused then client connection starts to get throttled, which means API processing gets slowed down. This does not mean the connection gets severed but rather if it slowed down enough, the connection could time out and could open an attack vector for Denial-of-Service Attacks. To counter throttling, many API gateways implement rate limits. Rate limiting is a threshold for how much requests an API gateway can handle and is measured by Transaction per Second (TPS) (Defranchi). The amount of TPS a backend can process is called backend rate limiting while the amount of TPS a client can send is called application rate limiting. Gateways can also use quotas to determine how often an API can be used and its history is tracked over time. Quotas are more for long-term use and are measured in Transactions per day or month whereas a rate limit is measured in TPS. Quotas can be placed on the clients themselves or even applications. Sometimes, clients might need to use APIs above their limit or quotas, and this can be enabled through API bursts (Defranchi). Clients can send in calls above their limit for a certain amount of time. To determine what clients can access the APIs, policies and rules can be configured within the gateway.



SOAP and REST APIs

Representational State Transfer (REST) is an architectural style used in modern applications like microservices and containers (GeeksforGeeks). REST APIs are versatile, commonly used for public applications, and require less security while still supporting TLS encryption for secure data exchange. They use JavaScript Object Notation (JSON) for flexible data transfer.

To be RESTful, an API must follow key principles:

1.Uniform Interface – The server consistently relays information, enforcing four constraints (AWS):

1. Clients request resources via a uniform resource identifier.
2. Servers send metadata to allow modification or deletion of resources.
3. Self-descriptive messages guide clients on resource usage.
4. Hyperlinks to related resources aid task completion.

2.Statelessness – Each request is processed independently (AWS).

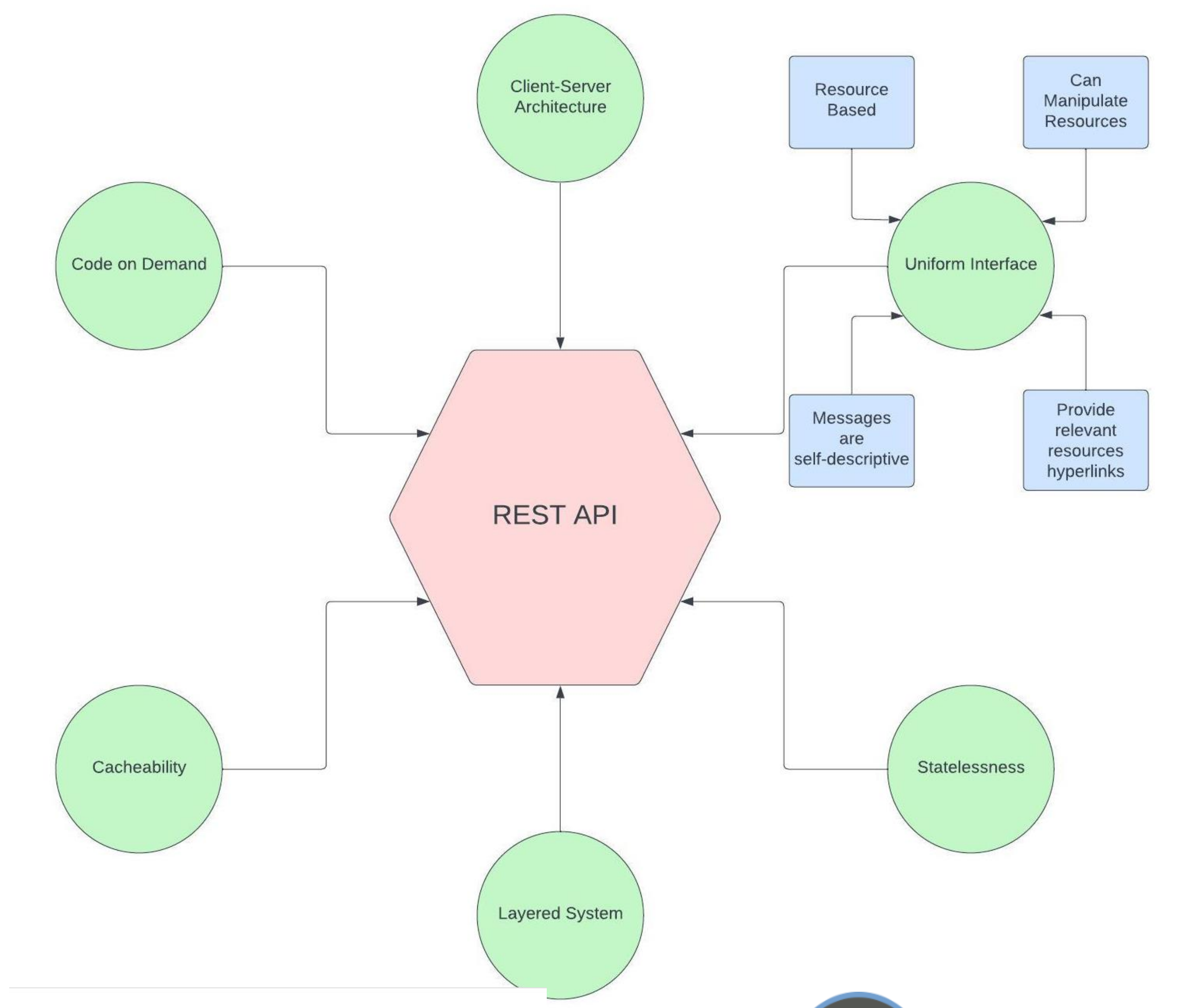
3.Layered System – Only adjacent layers share knowledge, with intermediaries between client and server.

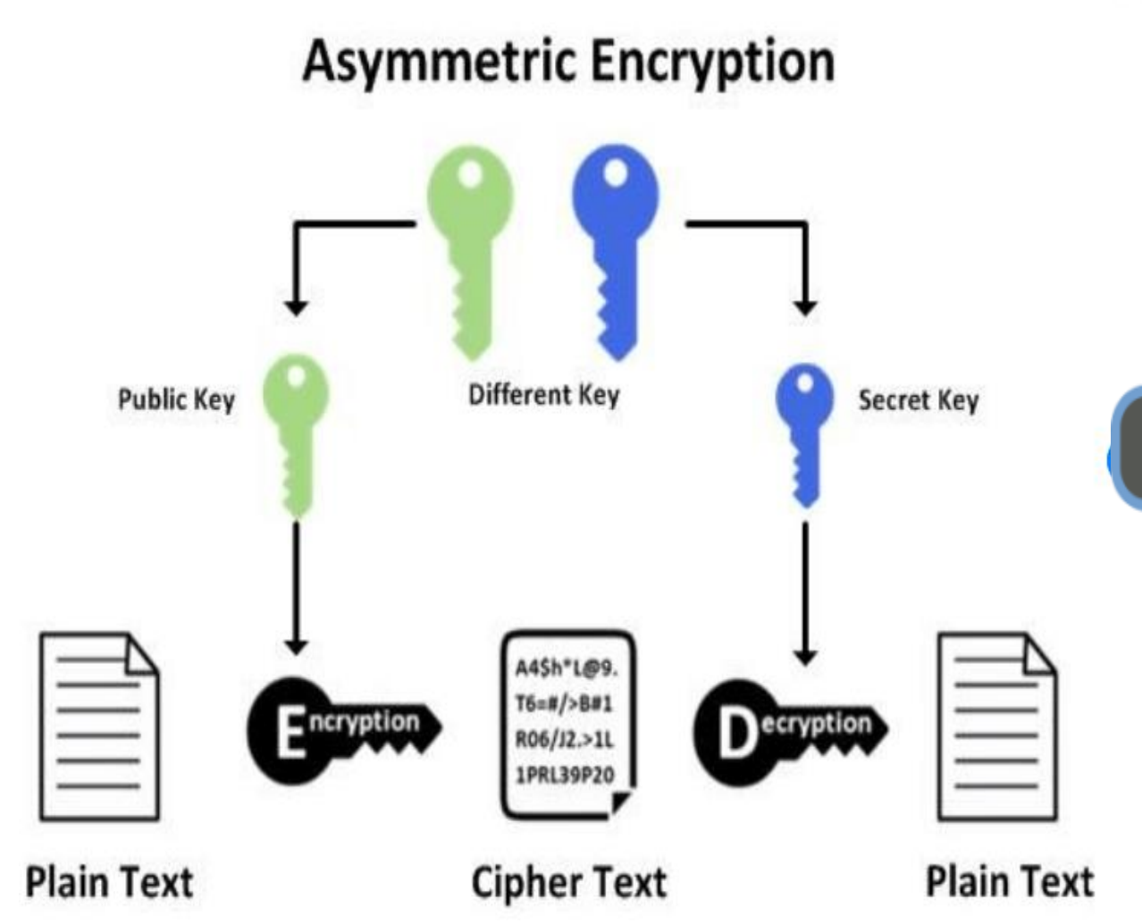
4.Caching – Responses are stored to improve performance.

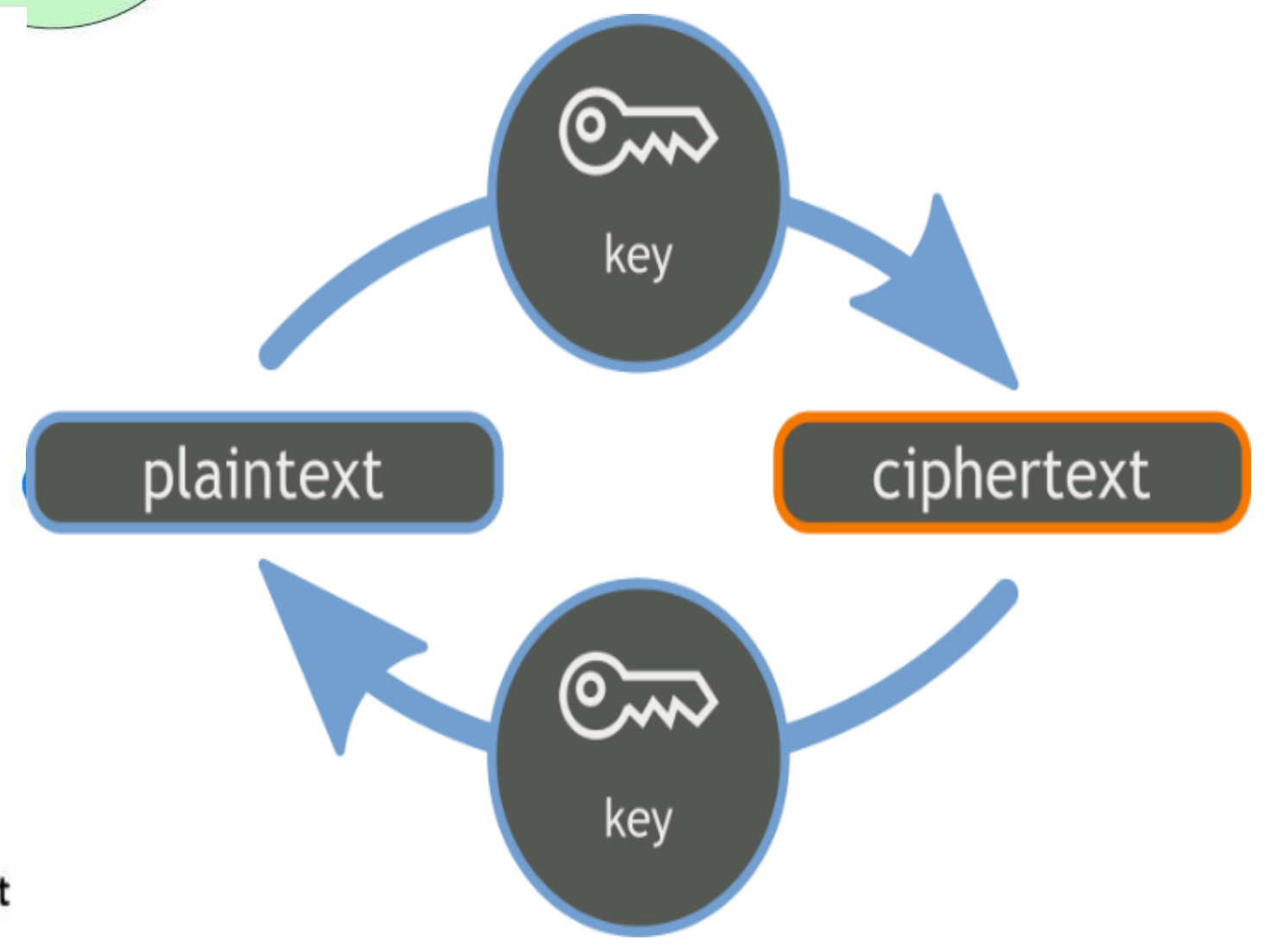
5.Code on Demand – Clients can receive executable code from the server for enhanced functionality.

REST APIs are scalable due to statelessness and caching, reducing server load. Their layered structure allows flexibility, enabling independent modifications without affecting application logic. They also support multiple programming languages.

Authentication methods include HTTP authentication (basic and bearer tokens), API keys (less secure due to network transmission), and OAuth, which employs Multi-Factor Authentication (MFA) for stronger security.







Simple Object Access Protocol (SOAP) predates REST and is better suited for legacy system integration. It includes Web Services Security (WS Security), emphasizing confidentiality, authentication, and compliance with OASIS and W3C standards (RedHat). SOAP exclusively uses XML and secures data through XML encryption, XML signatures, and Security Assertion Markup Language (SAML) tokens (IBM). It is also ACID-compliant, ensuring data integrity.

WS Security enhances SOAP by securing message confidentiality, integrity, and authentication. XML encryption, developed by W3C, ensures data remains protected, with options to encrypt entire messages or specific parts like credit card details. Encryption can be **symmetrical** (faster, using a shared private key) or **asymmetrical** (more secure, using private/public key pairs). Symmetrical encryption suits enterprises that can securely share keys, while asymmetrical encryption ensures stronger security with private key signatures, covering two elements of the CIA triad: confidentiality and integrity.

SAML facilitates authentication and authorization using tokens exchanged between a service provider (SP) and an identity provider (IdP). Once verified, a SAML response grants the vendor access to requested databases. While SAML remains a leading authentication method for SOAP APIs, it can be vulnerable to attacks. Using an IdP like Okta with Multi-Factor Authentication (MFA) mitigates risks.

An alternative authentication method, **OpenID Connect (OIDC)**, allows third-party authentication, simplifying account management for users while reducing developers' responsibility for password security (openid.net). Enterprises can use **SAML** for workforce authentication, while **OIDC** is better for customer authentication, improving access through services like Google. Internal government departments can authenticate using Okta rather than maintaining multiple accounts.

Industry Trends

As society continues to use and adapt to new technologies, so does the usage of APIs. They are an important part of what connects us together and the security of them will become of utmost importance. APIs are relatively new in the already novel tech industry as seen by a report from Salt Security, an API security platform. In this report, it was observed that "95% of respondents experienced security issues in their production APIs in" 2024 (Taylor). Instead of developing tools to combat these issues manually, the industry is turning to artificial intelligence (AI) to help automate API security. AI can be trained to recognize malicious traffic within a network and ban that activity before an attack happens. Beyond security, AI has been used in API development for tasks such as documentation and testing (Taylor). Technology companies have invested lots of resources to train their own AI and have already started to implement them within their products. Now, the challenge is securing any vulnerabilities that the AI might have produced.

Enterprise Recommendations

Each API design has its strengths and weaknesses; REST APIs are more flexible and faster but sacrifice security. SOAP APIs, on the other hand, are more secure, but are also slower and rigid. Each API should be used where it is appropriate. SOAP APIs will provide more security in exchanging data and pose less of a risk to the department. REST APIs would provide better compatibility and future use for the department as it becomes more prominent in cyberspace. Overall, I would recommend SOAP APIs to an enterprise in this instance. Since they could have a few different business groups they would like to make APIs available on the public internet to be used by vendors and other government bodies in order to share internal data.

Lab Application

To further solidify my understanding of APIs, I developed a workshop lab for my fellow students with the help of my club Student's With an Interest in the Future of Technology (SWIFT). The core components of the lab consisted of a vulnerable python database and Postman (an API platform). I had club members Postman to perform the penetration test to find flags in a Capture the Flag (CTF).

Conclusion

APIs are important technology; it is used by everyone for anything, anywhere. To bring about awareness on this topic, I taught my club, SWIFT, about it and prepared a lab for them to perform penetration tests on a vulnerable API. They can be exploited in many ways, including some that can have code written by AI. Using AI responsibly and ensuring that a production API is secure will always be a challenge for security professionals in the field. One reason for this is due to the numerous amounts of APIs that can and will be developed in the future; there is no one API. Each company has different goals, and therefore a different API that will fulfill their needs. But one thing remains constant, the prioritization of API security.

Sources

"How Openid Connect Works - Openid Foundation." *OpenID Foundation - Helping People Assert Their Identity Wherever They Choose*, 20 July 2023, openid.net/developers/how-connect-works/.

"REST API Architectural Constraints." *GeeksforGeeks*, GeeksforGeeks, 9 Jan. 2025, www.geeksforgeeks.org/rest-api-architectural-constraints/.

"What Are the Benefits and Challenges of Using SAML for Single Sign-on (SSO)?" *SAML for SSO: Benefits and Challenges Explained*, 1 Sept. 2023, www.linkedin.com/advice/1/what-benefits-challenges-using-saml-single-sign-on#:~:text=SAML%20is%20a%20complex%20protocol,with%20multiple%20IdPs%20or%20SPs.

"What Does an API Gateway Do?" *Red Hat - We Make Open Source Technologies for the Enterprise*, www.redhat.com/en/topics/api/what-does-an-api-gateway-do. Accessed 28 Jan. 2025.

Digital Signature Overview, www.ibm.com/docs/en/b2badv-communication/1.0.1?topic=overview-digital-signature. Accessed 28 Jan. 2025.

Lydia Defranchi. "API Rate Limiting, Throttling, API Quota & API Bursts Defined." *Axway Blog*, 14 Oct. 2024, blog.axway.com/learning-center/apis/api-design/api-quota.

Soap vs Rest - Difference between API Technologies - AWS, aws.amazon.com/compare/the-difference-between-soap-rest/. Accessed 29 Jan. 2025.

What Is API Security?, www.redhat.com/en/topics/security/api-security#what-is-web-api-security. Accessed 28 Jan. 2025.

What Is Restful API? - Restful API Explained - AWS, aws.amazon.com/what-is/restful-api/. Accessed 29 Jan. 2025.

XML Encryption Overview, www.ibm.com/docs/en/b2badv-communication/1.0.1?topic=overview-xml-encryption. Accessed 28 Jan. 2025.

XML Encryption, www.ibm.com/docs/en/was/8.5.5?topic=wsspmica-xml-encryption-1. Accessed 28 Jan. 2025.

Picture Sources

Comment, et al. "What Is an API (Application Programming Interface)." *GeeksforGeeks*, 7 Aug. 2024, www.geeksforgeeks.org/what-is-an-api/.

Wikimedia Commons, commons.wikimedia.org/wiki/File:Orange_blue_symmetric_cryptography_en.svg. Accessed 29 Jan. 2025.

Wikimedia Commons, commons.wikimedia.org/wiki/File:End_to_end_encryption.png. Accessed 29 Jan. 2025.

Taylor, Twain. "What's next for Apis? 4 API Trends for 2025 and beyond: TechTarget." *Search App Architecture*, TechTarget, 2 Dec. 2024, www.techtarget.com/searchapparchitecture/tip/Whats-next-for-APIs-API-trends.