DevNet Associate Network Programmability Module

Module ACCB5029

November 18, 2025

COMPUTER NETWORK AND CYBERSECURITY
UWTSD

Assignment Title:
Reflection on Technical reading and Case Study about Network Programmability

Table of Contents

Task 1: Developer Network Associate Chapters 2
Abstract 2
Chapter 1: Introduction to Network Programmability 2
Understanding Network Programmability 2
Case Study Insight 3
Chapter 2: The DevNet Developer Environment 3
Key Components of the DevNet Environment 4
Chapter 3: Software Development Design 4
Software Development Lifecycle (SDLC) 5
Chapter 4: Using and Understanding APIs 5
Conclusion 11
Task 2: Case Study about Network Programmability and Automation in modern Enterprises. 12
Introduction 12
Organization Background 14
Objectives 15
Tools and Technologies. 16
Implementation Methodology 16
Lessons Learned 16
Conclusion 17
References (IEEE Style) 17
References 17

Task 1: Developer Network Associate Chapters

Abstract

Cisco Developer Network (DevNet) is the centre of developer program that helps network engineers, cybersecurity, software developer, IT professionals, specially students learn how use Cisco technologies, programmability tools and Application Programming Interfaces.

Chapter 1: Introduction to Network Programmability

Network Programmability The role is tools to enabling technology and definition is the ability to use software, code Application Programming Interfaces to directly manage and automate network functions. One of the foundational topic is model driven programmability which enables automation by defining network configurations in a structured way. Network Programmability is the ability to use software to interact with network devices. It can apply to both traditional and modern network using a Python script to configure a legacy router. Software Defined Networking is concept of a centralised, smart network, and programmability is the method used by the centralised controller to achieve it. the controller uses programmability tools to implement decisions across all decoupled devices in the data plane. [1]

Understanding Network Programmability

The network programmability is referring to the ability to control and automate network through code and software interfaces rather than manual device configuration. This can be accomplished using scripting languages

like Python or through model driven configurations such as YANG and NETCONF. The main goal is to automate routine tasks and reduce human errors, allowing networks to adjust quickly and efficiently as business needs change [2].

Case Study Insight

The big organisation used Cisco's Developer Network tools to automate data centre operations, reducing human errors and improving response time. And below are some keys:

- ✓ Skills: Engineers must know Python, Application Programming Interfaces, and Git
- ✓ Efficiency: Automation reduces manual errors with reliable code.
- ✓ Scalability: Enables rapid, uniform configuration across devices.
- ✓ Troubleshooting: Automation speeds up fault detection and fixes.

These case shows that network programmability enhances not just efficiency, but also agility and consistency at scale.

Chapter 2: The DevNet Developer Environment

The Developer Network Environment is a platform that provides tools and resources for learning to help network engineers, software developers, cybersecurity professionals, and IT students build their skills in network programmability and automation using Cisco technologies and Application Programming Interfaces, and software development practices.

The Cisco platform and development section or Developer Network associate it focuses on working with Cisco various platforms. This all about understanding how Cisco devices and services expose Application Programming Interfaces for managing networks and security and computing infrastructure. So, a key skill here is writing Python scripts using Cisco SDK and SDK stands for software development kits and they are pre-built tools that simplify interacting with Cisco platforms. NeXT I was exploring

Cisco's network management platforms such as Mirage, Cisco DNA centre, ACI, Cisco SDWAN. And these platforms provide centralised automation and monitoring systems and understanding their Application Programming Interfaces helps in network automation. Need to know how to leverage Cisco Developer Network resources such as sandboxes and learning labs. These are great tools for testing and learning about Cisco Application Programming Interfaces.

Key Components of the DevNet Environment

Sandboxes for testing, a code exchange with shared scripts, learning labs and certifications for skill development, comprehensive Application Programming Interfaces documentation and tools, and a strong community for collaboration and knowledge sharing. [3]

Developer Network's Educational Impact, encourages hands on learning and collaboration theoretical concepts from the classroom to real world networking problems, and provides supports learning labs, Developer Network Sandbox, and Code Exchange [4]. Cisco Developer Network, 2023.

Overall The Developer Network prepares professionals to design automated, secure, and scalable networks essential in today's DevOps driven, digitally transforming IT landscape, leveraged Cisco DevNet, Python, and Cl/CD to automate network, cut downtime, and boots efficiency. It provides resources to build, automate, and integrate solutions with Cisco platforms centre. [1, 5]

Chapter 3: Software Development Design

This chapter introduces core programming concepts and software design principles that are crucial for building scalable, maintainable applications

The software development and design section is the Cisco Developer Network associate exam is all about helping network engineers become comfortable with programming concept that are essential for automation. It

is important to understand the basics of software development event if I do not consider myself a programmer. First I learned about the data formats like YAML, JSON, AND XML. And these are used a lot in networking especially when with Application Programming Interfaces or automation tools. JSON and YAML for example are commonly used to send configuration data or response from Application Programming Interfaces while the Xml is often seen in older systems. The exam also expects to convert or parse these formats into Python data structures so I can work with them in scripts [6].

Software Development Lifecycle (SDLC)

The software is known as the software development life cycle. Is the high-quality software that fulfils client requirements and is finished on schedule and within budget? Software development lifecycle. The most crucial stage of software development life cycle is the initial requirements gathering phase. According to the client's business needs, business analysis gathers and documents the client's requirements [7] , [8].

The SDLC defines a structured process for building reliable, scalable software:

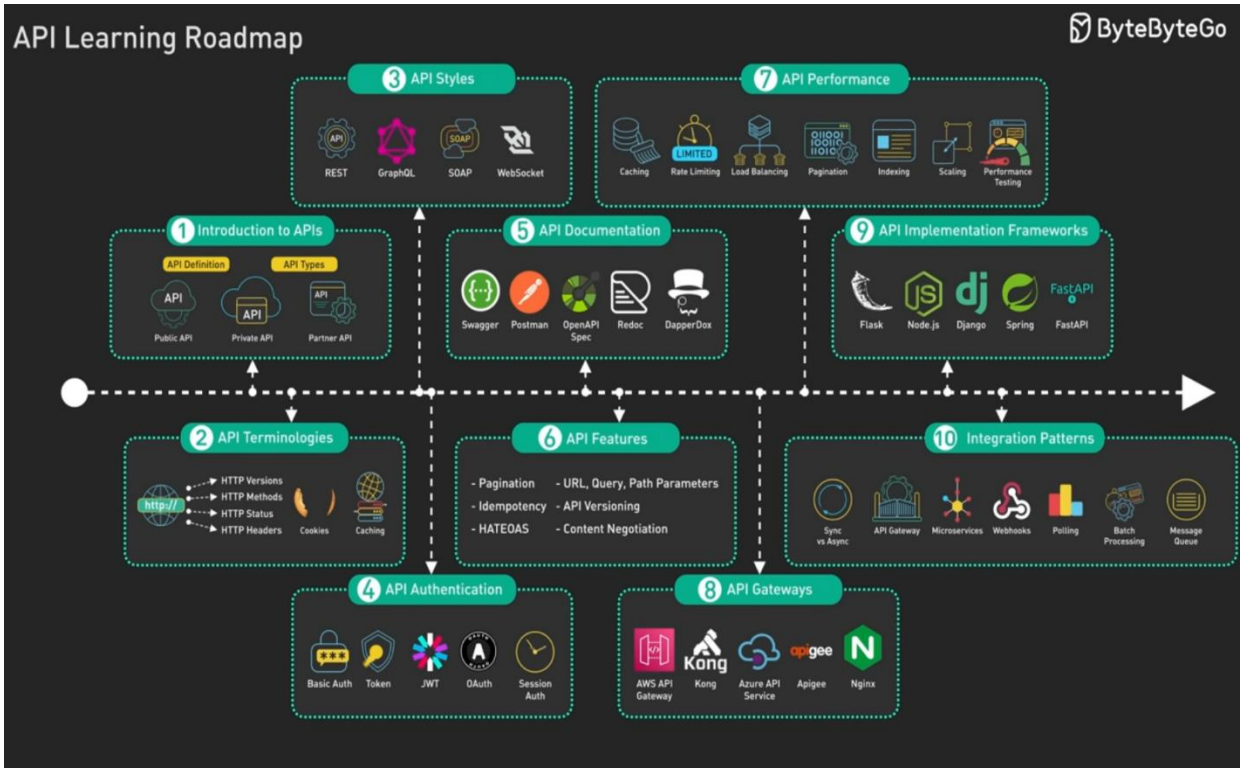
1. Planning: Set goals, resources, and timelines.
2. Design: define architecture and data models.
3. Implementations: is developing or writing code using languages like Python, C#, JavaScript, and go on.
4. Testing: is ensure functionality and performance as intended.
5. Deployment: is the launch in the production environment.
6. Maintenance: is the apply updates and fixing security issues.

Understanding SDLC ensures that network automation scripts and tools are developed systematically and are scalable.

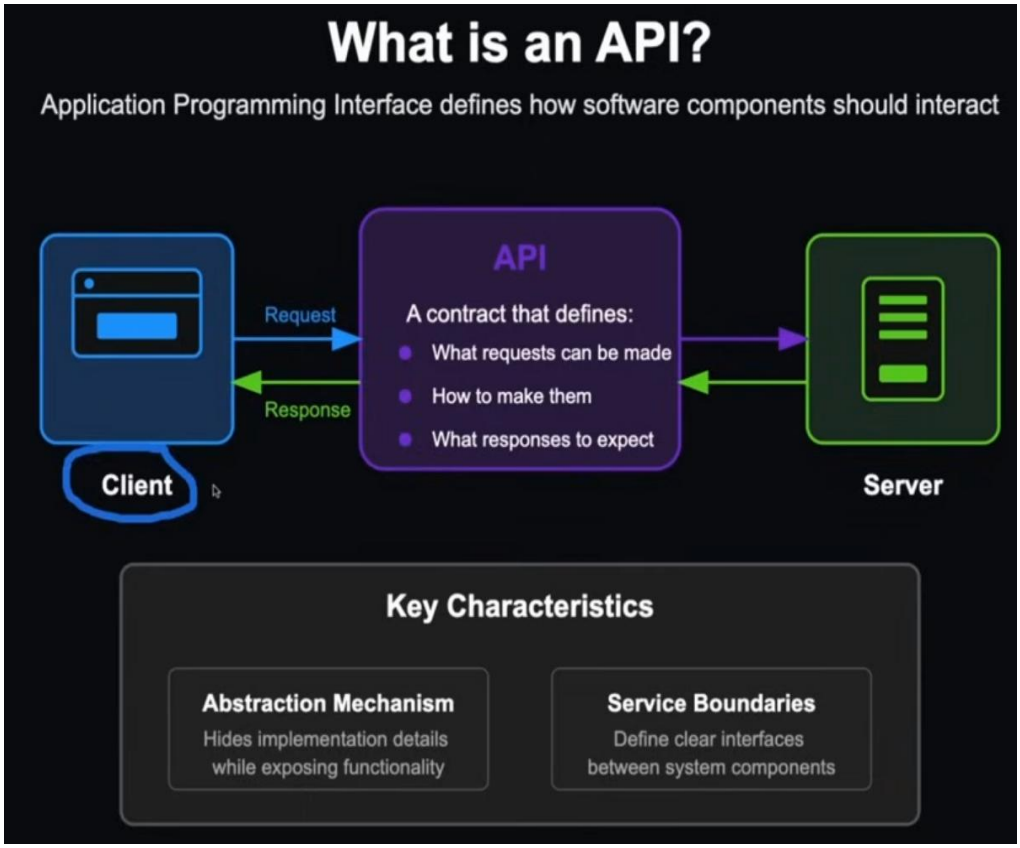
Chapter 4: Using and Understanding APIs

My understanding and using Application Programming Interfaces, a cornerstone of network programmability and modern software integration. These chapter focus on representational state transfer, using Developer Network programmable network platforms such as Cisco DNA Centre. Enabling automated configuration, monitoring, and control. I also learned that APIs use GET, PUT, POSTMAN [9], [10]. Also introduces model driven programmability, using data as Yet Another Next Generation. The backbone of intelligent, automated scalable network infrastructures. [11]

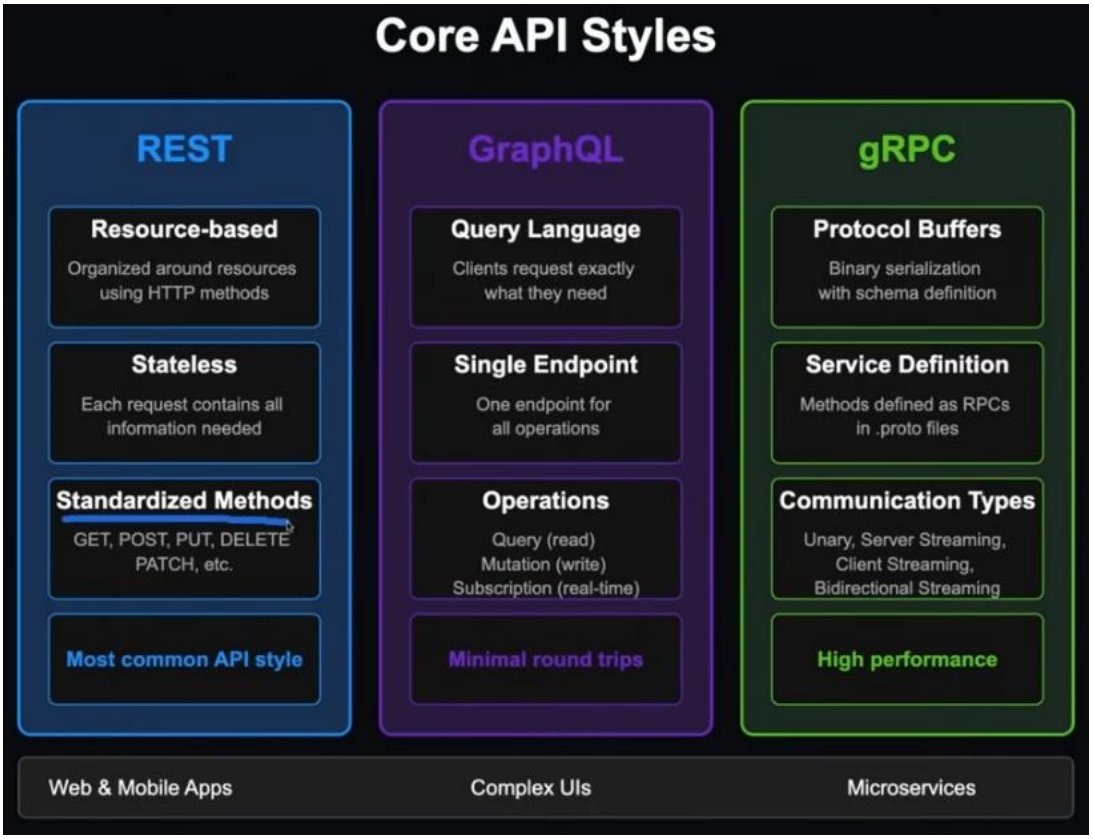
Below screenshot is from APl Learning Roadmap [12] [13]

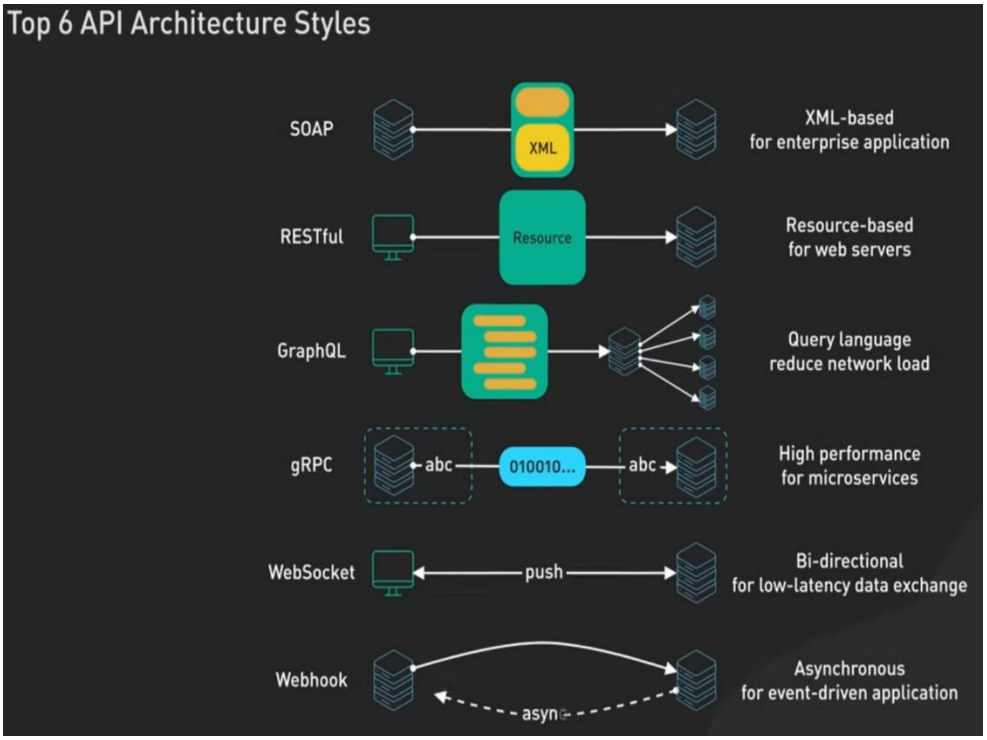


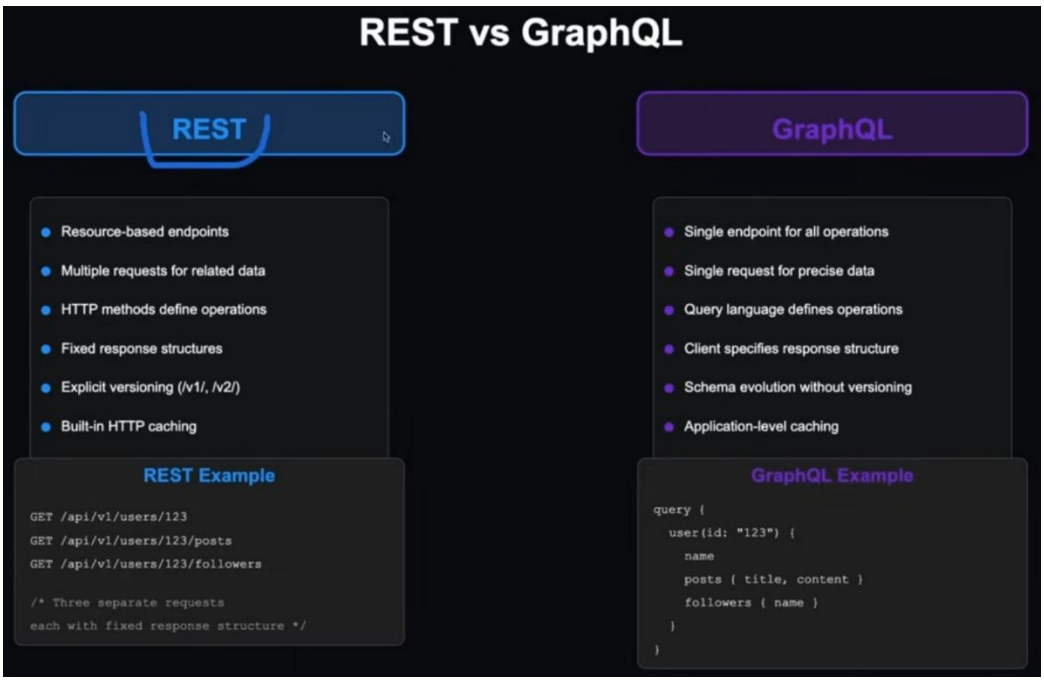
My understanding is APl design which will enable to create efficient, scalable and also maintainable interfaces between software systems [14].



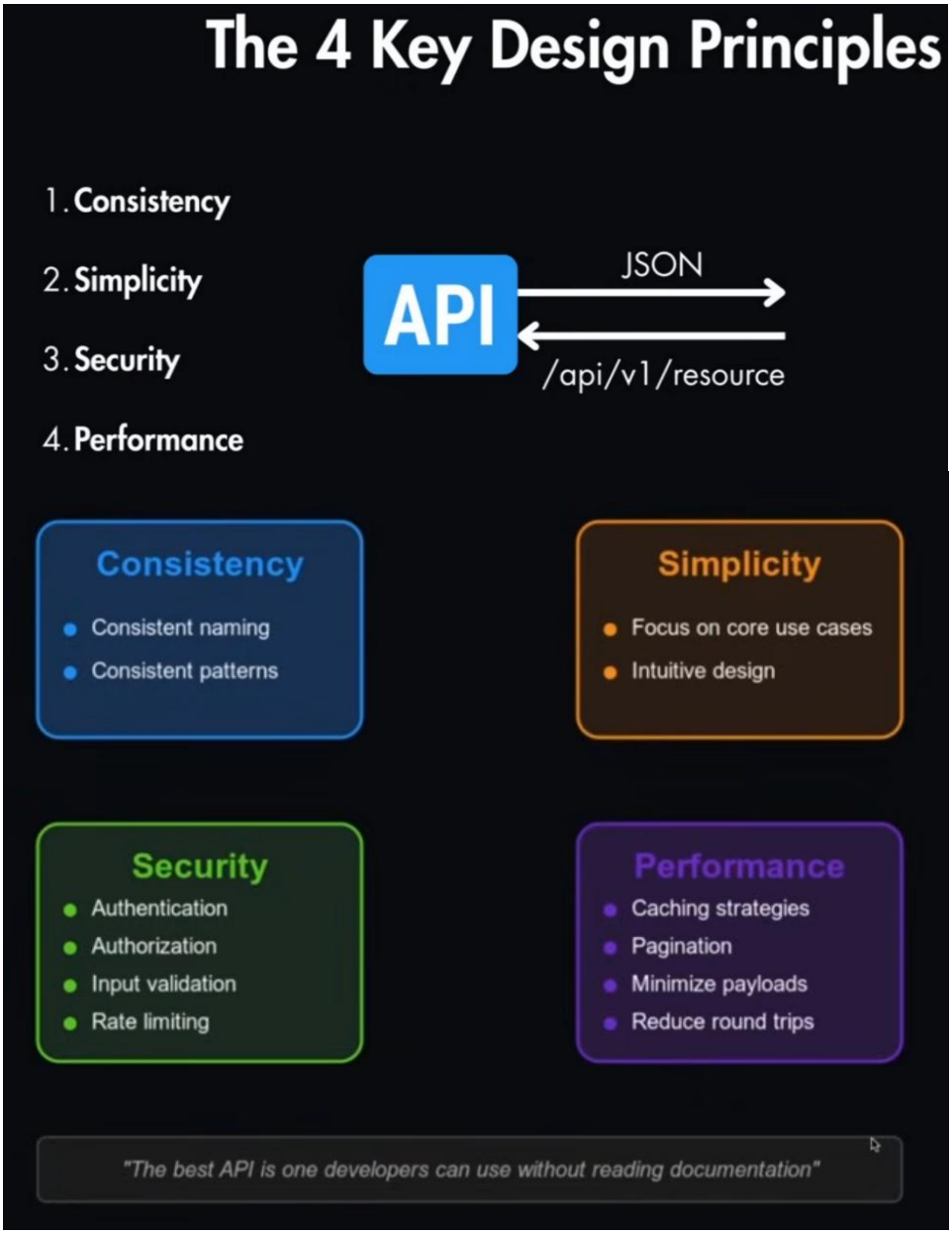
I also learned module objective, like API Design style, Web hooks [15]

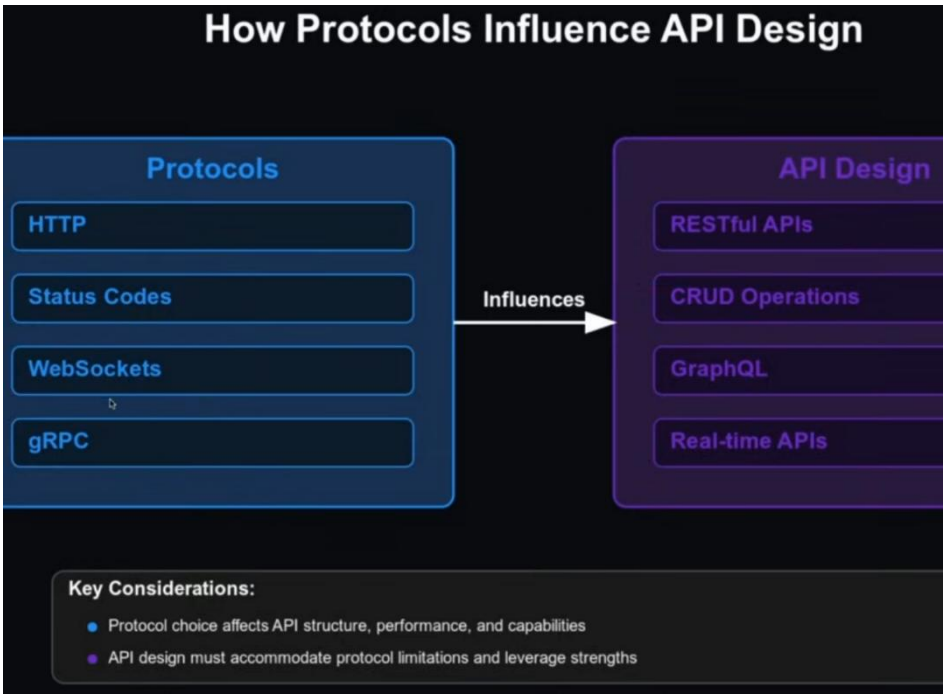

Core API Styles
REST
Resource-based
Organized around resources using HTTP methods
Stateless
Each request contains all information needed
Standardized Methods
GET, POST, PUT, DELETE PATCH, etc.
Most common API style
GraphQL
Query Language
Clients request exactly what they need
Single Endpoint
One endpoint for all operations
Operations
Query (read)
Mutation (write)
Subscription (real-time)
Minimal round trips
gRPC
Protocol Buffers
Binary serialization with schema definition
Service Definition
Methods defined as RPCs in .proto files
Communication Types
Unary, Server Streaming, Client Streaming, Bidirectional Streaming
High performance
Web & Mobile Apps
Complex UIs
Microservices


Top 6 API Architecture Styles
SOAP
XML
XML-based
for enterprise application
RESTful
Resource
Resource-based
for web servers
GraphQL
Query language
reduce network load
gRPC
abc
010010...
abc
High performance
for microservices
WebSocket
push
Bi-directional
for low-latency data exchange
Webhook
Asynchronous
for event-driven application


REST vs GraphQL
REST
Resource-based endpoints
Multiple requests for related data
HTTP methods define operations
Fixed response structures
Explicit versioning (/v1/, /v2/)
Built-in HTTP caching
REST Example
GET /api/v1/users/123
GET /api/v1/users/123/posts
GET /api/v1/users/123/followers
/* Three separate requests
each with fixed response structure */
GraphQL
Single endpoint for all operations
Single request for precise data
Query language defines operations
Client specifies response structure
Schema evolution without versioning
Application-level caching
GraphQL Example
query {
user(id: "123") {
name
posts { title, content }
followers { name }
}
}

The four key design principles are one developers can use without reading documentation.





It is important to understand the basics of software development and how protocols influence APl design works,event if you do not consider yourself a programmer. I learned about the data formats like YAML, JSON, AND XML. And these are used a lot in networking especially when with Application Programming Interfaces or automation tools. JSON and YAML for example are commonly used to send configuration data or response from Application Programming Interfaces while the Xml is often seen in older systems.

Conclusion

Overall The Developer Network prepares professionals to design automated, secure, and scalable networks essential in today's DevOps driven, digitally transforming IT landscape, leveraged Cisco DevNet, Python, and Cl/CD to automate network, cut downtime, and boots efficiency. The Cisco Developer Network Associate curriculum integrates or unifies networking, software development, and automation into a modern, practical skill set.

Chapter 1: Introduces network programmability and Application Programming Interfaces driven automation.

Chapter 2: Explores the Developer Network environment with tools and community support.

Chapter 3: Builds programming, version control, and secure coding skills.

Chapter 4: Enhanced my understanding of how APIs and model driven programmability form the backbone of intelligent, automated, and scalable network infrastructures.

Task 2: Case Study about Network Programmability and Automation in modern Enterprises.

Introduction

Welcome to the next task2. So on this task I am going to create case study about network programmability and automation in modern enterprises. Let's look at the contents of this chapter. So first I am going to talk about the introduction to why do we need network programmability, the problem, difficulties, and also the automation part. Why do we need automation? Then I will talk about the common programming languages, which is available in the industry.

Network programmability replaces manual configurations with automated, software driven control, boosting and scalability in modern IT. Cisco's DEVASC certification supports this shit, merging networking with skills. A case study shows how a midsized enterprise benefits from adopting programmable networks. Traditional networks, rely on manual (CLI) that stand for command line interface configurations. The risks of manual data processing errors, inconsistencies between teams. My understanding between manual and automated configuration.

Manual data processes have error prone human data entry, person dependency, limited scalability, hours/days for insights, while automated

data validation, error handling growing data volumes, real time or scheduled automation, visual documented. And for example, such as they need to execute a certain command, they need to perform a certain configuration, they need to change the password. So just imagine if you are handling just like three network devices, which is fine no problem, because network engineer can easily just tell net to treat different network equipment. But imagine if you are handling 100 devices, this is gone be very, very time consuming and also very efficient because you need to keep track one by one, maybe you need to do an Excel spreadsheet to keep track one by one. Maybe you need to do an Excel spreadsheet to keep track which one have you been updated and stuff like that and also very difficult to audit.

You want to do a configuration audit, for example, company says we need to execute a new policy. All the device today cannot be enabled with telnet and you need to enable all of them with SSH, which is also mean as Telnet. And after that, configuration changes. So again, as I mentioned, sometimes you need to change the password for all the devices. This is actually the difficulties. All right, so what is network automation? So let's look at the definition. Network automation, this is tools that are used to implement automated network deployment, operation, and also operation and maintenance. And this is actually the objective, very simple, is to reduce the dependency on the human. So there are many open source tools available currently. Example, Puppet, Shed.

These are all very popular in the industry currently. And they are not only meant for network automation, they are also meant to be used for, let's say, software deployment for operating system to automate the process, to do provisioning of virtualization and stuff like that. So these are some examples. Ansible, Chef, Python is more like programming languages that can actually include a lot of modules to do the automation [16]. so now let's look at the programming-based network automation. So here you can see this example using Python below. So this is a very typical configuration steps. So usually when we want to configure a device, you will type says name and then followed by the device name, then followed by VLAN, then you want to perform a configuration for VLAN and maybe you want to do description. So instead of typing this over and over again for. So below is Python scripts screenshot as evidence [17].

```python
from netmiko import ConnectHandler

device = {
    'device_type': 'cisco_ios',
    'ip': '192.168.10.1',
    'username': 'admin',
    'password': 'cisco123'
}

connection = ConnectHandler(**device)
commands = [
    'hostname Branch-Router',
    'vlan 10',
    'name Sales',
    'exit'
]

connection.send_config_set(commands)
connection.disconnect()
```

Organization Background

Trends in Recruiting Automation, adopted network programmability to resolve issues from manual CLl management across its three offices [18].

The recruitment trends are experiencing a tech driven change. AI and automation are making hiring processes faster is becoming more important than tradition qualifications. The increase in remote and talent pools. There is also a greater emphasis on candidate experience, diversity, equity and employee wellbeing. The technologies like block chain are further changing the landscape.

By adopting network programmability, TRA aimed to create a dynamic and responsive IT environment capable of supporting its growing customer base and distributed workforce.

Addressing Organisational Challenges through Industry Technologies and Practices

Modern organisations face more complexity in managing digital infrastructures, which leads to the use of automation and new industry practices. Tools like Python scripting, and YANG (Yet Another Next Generation) models help automate repetitive tasks [19]. This ensures consistency and reduces human error in network operations.

Collaboration and efficiency improve through CI/CD and Git version control [20] [21]. These tools simplify testing, deployment, and teamwork. They also improve reliability, speed up configuration changes, and ensure accountability among teams.

Automation greatly boosts workflow performance by speeding up provisioning, lowering downtime, and increasing scalability. Beyond operational advantages, it lowers costs, adds flexibility, and enhances customer experience with faster, more reliable service delivery.

However, challenges still exist. The initial costs of implementation, the skill gap among staff, and the risk of over automation which can limit human oversight are significant issues that organisations need to manage carefully.

Looking ahead, AI-driven networking and Zero Touch Provisioning (ZTP) are emerging trends that will further change the industry. They will enable smarter, self-managing networks and represent a major step toward fully automated digital operations [22].

Problem Statement

- Manual configuration challenges
- Lack of centralised visibility
- Compliance and security risks.

Objectives

The objectives focus on automating network tasks with Python and Cisco Als, centralising management via DNA Centre, enabling RESTFUL communication, integrating Cl/CD pipelines, and improving security and efficiency through automation [23] [24].

Tools and Technologies.

During implementation were used the following tools.

Category	Technology
Version Control	GitHub
Programming Language	Python 3.10
APls	RESTCONF, NETCONF
Automation Framework	Ansible
Cl/CD Tools	Jenkins, Docker
Monitoring	Cisco Model Driven Telemetry
Testing	Postman

Implementation Methodology

- APl Exploration and Authentication, using Cisco's DevNet Sandbox.
- Configuration Automation, using Python and Ansible playbook, VLANs, ACLs, and interface configurations were deployed.
- CLCD Integration, were implemented Jenkins pipelines for automated testing.
- Real Time Monitoring, to deliver real time device health updates.
- Security Hardening, all communications were encrypted via HTTPS/TLS.

Lessons Learned

- Train network staff
- Begin automation with repetitive tasks before scaling
- All workflows document and Adopt a DevOps mind set. [26]

Network programmability encompasses a set of tools designed for deploying, managing, and troubleshooting network devices. An intelligent software system drives a programmability-enabled network, capable of handling individual nodes or groups of nodes, allowing it to address the network as a cohesive whole.

Conclusion

Overall The Developer Network prepares professionals to design automated, secure, and scalable networks essential in today's DevOps driven, digitally transforming IT landscape, leveraged Cisco DevNet, Python, and Cl/CD to automate network, cut downtime, and boots efficiency [27]. Programmability proved vital for scalable, secure operations and future AI driven innovation.

Coming to term network programmability can have a different meaning or depending on a perspective of person who is using or where the area it is used to. Which are having the driver configurations, troubleshoot. so on the device. The programmer, APls or the device will be communicating between the software sitting in the device based on the technology or based on the person who is using it, the network programmability comes into.

References (IEEE Style)

References

[1] D. Cisco, "Cisco, DevNet," Cisco, DevNet. [Online].

[2] [Online]. Available: https://developer.cisco.com/docs/nx-os/model-driven-programmability-with-yang/#the-rise-of-network-automation. [Accessed 18 10 2025].

[3] D. Sandbox, 2024. [Online]. Available: https://developer.cisco.com/site/sandbox/?utm_source=chatgpt.com. [Accessed 02 11 2025].

[4] [Online]. Available: https://developer.cisco.com/docs/sandbox/getting-started/#what-is-devnet-sandbox. [Accessed 28 10 2025].

[5] Cisco. [Online]. Available: https://developer.cisco.com/docs/nx-os/model-driven-programmability-with-yang/.

[6] developer.cisco.com. [Online]. Available: https://developer.cisco.com/. [Accessed 29 10 2025].

[7] aws.amazon. [Online]. Available: https://aws.amazon.com/what-is/sdlc/. [Accessed 30 10 2025].

[8] C. Quiroz-Vázquez. [Online]. Available: https://www.ibm.com/think/topics/software-development. [Accessed 30 10 2025].

[9] [Online]. Available: https://bytebytego.com/guides/the-ultimate-api-learning-roadmap/.

[10] [Online]. Available: https://www.geeksforgeeks.org/system-design/application-protocol-system-design/#common-application-protocols. [Accessed 30 10 2025].

[11] vishalbhandari, 17 01 2025. [Online]. Available: https://community.cisco.com/t5/cisco-cafe-blogs/automating-network-operations-with-cisco-switching-and-apis/ba-p/5248734.

[12] F. &. Beyond. [Online]. Available: https://bytebytego.com/. [Accessed 30 10 2025].

[13] 2025. [Online]. Available: https://bytebytego.com/. [Accessed 31 10 2025].

[14] O. Naseem, 2 2 2024. [Online]. Available: https://www.datasciencecentral.com/building-robust-api-step-by-step-guide/. [Accessed 30 10 2025].

[15] [Online]. Available: https://www.geeksforgeeks.org/software-testing/what-is-an-api/. [Accessed 30 10 2025].

[16] R. H. A. A. Platform, 2024. [Online]. Available: https://www.redhat.com/en/technologies/management/ansible. [Accessed 02 11 2025].

[17] [Online]. Available: https://www.redhat.com/en/technologies/management/ansible. [Accessed 02 11 2025].

[18] “Wellpin,” 11 07 2025. [Online]. Available: https://wellpin.io/blog/recruiting-automation-in-2025-tools-trends-and-results. [Accessed 10 31 2025].

[19] techtarget, “techtarget,” 30 09 2025. [Online]. Available: https://www.techtarget.com/searchapparchitecture/definition/RESTful-API. [Accessed 02 11 2025].

[20] M. Fowler, “ martinfowler.com,” 14 01 2024. [Online]. Available: https://martinfowler.com/articles/continuousIntegration.html. [Accessed 02 11 2025].

[21] J. Fernandes, “Red Hut,” 2023. [Online]. Available: https://www.redhat.com/en/about/press-releases/red-hat-brings-distributed-ai-inference-production-ai-workloads-red-hat-ai-3?intcmp=RHCTG0250000464627. [Accessed 02 11 2025].

[22] “shttps://www.se.com/ww/en/insights/next-generation-automation/industrial-software/unlocking-the-potential-of-5g-for-industrial-automation/,” 2023. [Online]. Available: https://www.se.com/ww/en/insights/next-generation-automation/industrial-software/unlocking-the-potential-of-5g-for-industrial-automation/. [Accessed 02 11 2025].

[23] f. scale, “fullscale,” 21 05 2025. [Online]. Available: https://fullscale.io/blog/cicd-pipeline-automation-guide/. [Accessed 31 10 2025].

[24] “developer.cisco.com/docs/network-automation-delivery-model/cicd/,” 2025. [Online]. Available: https://developer.cisco.com/docs/network-automation-delivery-model/cicd/#continuous-integrationcontinuous-delivery-cicd. [Accessed 01 011 2025].

[25] Edward, “Well Pin,” 11 07 2025. [Online]. Available: https://wellpin.io/blog/recruiting-automation-in-2025-tools-trends-and-results. [Accessed 31 10 2025].

[26] J. Fernandez, “Cisco Blogs / Developer / What Workflows for Cisco Networking Can Do,” 08 10 2025. [Online]. Available: https://blogs.cisco.com/developer/cisco-workflows-live. [Accessed 02 11 2025].

[27] 22 October 2022. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-19-3029-4_18. [Accessed 31 10 31].

[28] C. system, 2023. [Online]. Available: https://www.cisco.com/#tabs-35d568e0ff-item-4bd7dc8124-tab. [Accessed 02 11 2025].