18/10/2017

CA400 Functional Specification

ASDN Platform

Filip Nikolic – ID: 14470852

Supervisor – Mr. Brian Stone

SCHOOL OF COMPUTING, DCU

Table of Contents

1 INTRODUCTION

1.1 OVERVIEW

The project being developed is a network automation platform which will initially aid and eventually replace network engineers in configuring and making changes on a live network. This will subsequently reduce operating costs as well as the possibility of human introduced errors. The devices this platform will communicate with include, but are not limited to switches, routers and firewalls. These are not consumer, but industry grade networking appliances, ranging from thousands to hundreds of thousands of euro in value.

The system will use a variety of technologies to accomplish the above outlined goals. It will adopt a client-server architecture and is intended to run on a wide range of devices and operating systems which can display web-pages. The platform consists of several separate modules, each providing a specific set of functionalities. The two main ones will be the Software Defined Networking and Assisted Machine Learning Module.

1.2 BUSINESS CONTEXT

I will be developing this project with help from Agile Networks, as they will provide me with access to high-end network devices and servers to develop and test the project.

They are a Network Engineering Company, based in Dublin. They build and support IT networks across 1,600 sites and with over 1.8 million end users, despite being a small to medium sized company with around forty employees.

I have calculated that, on average a company such as Agile Networks can decrease engineer time spent making network changes significantly. It would normally take multiple minutes for a task that can be done by the ASDN platform in a matter of seconds.

1.3 AVAILABLE SOLUTIONS

One of the most common tasks that a network engineer carries out daily is making changes on networks, whether they are user requested or maintenance based. These repetitive tasks are extremely hard to automate, require specialist software that may not work in many scenarios or an in-house developed solution.

Most solutions available today are vendor specific, meaning that each network device vendor will offer their own automation solution which may cost thousands per month and is limited only to the devices manufactured by the same company. As many enterprise networks use a mix of manufacturers, these solutions are very impractical and only useful for a green field deployment scenario.

Other possible solutions are open-source, however require a lot of programming knowledge to deploy successfully and do not take into consideration version changes on devices. In addition, scalability is another one for its drawbacks, especially important in large enterprise environments.

The time an engineer spends making changes on a network decreases company efficiency as well as revenue. Unlike large multinational companies such as Facebook and Google, most wouldn't have the resources to develop effective in-house automation solutions.

1.4 GLOSSARY

- **OSI Model** – This is a reference network layering model, which has 7 layers, starting from the Physical at the bottom and moving towards the Application layer at the top.
- **Network Switch** – connects various devices together on a computer network. It uses packet switching to receive, process and forward data to the destination device, operating on the data link layer of the OSI Model.
- **Network Router** – forwards data packets between different networks. Packets are usually forwarded between routers, until it reaches its destination node. They operate on the Network Layer of the OSI Model.
- **Firewall** – a network security device that monitors incoming and outgoing traffic, making decisions in real-time whether to allow or block traffic based on a defined set of security rules.
- **LLDP** – Link Layer Discovery Protocol is a vendor-neutral link layer protocol used by network devices to advertise their identity, capabilities and neighbours.
- **SNMP** – Simple Network Management Protocol is used for collecting information from network devices such as switches, routers, firewalls, etc.
- **CDP –** Cisco Discovery Protocol is a proprietary protocol that accomplishes a similar goal to LLDP, used to share information over the network about interconnected Cisco equipment.
- **SSH** – s a cryptographic network protocol for operating network services securely over an unsecured network. The best-known example application is for remote login to computer systems by users.
- **XML** – is a markup language that defines rules for encoding data in a human-readable and machine-readable code.
- **VPN – Virtual Private Network** is a way of securely interconnecting networks over the internet, whereby the remote network will appear as if it was located internally, allowing users to securely access protected resources.
- **Ansible** – Ansible is an open-source automation engine that allows for software provisioning, configuration management, and application deployment.
- **VLAN – Virtual local Area Network** is a group of devices on one or more local area networks configured to communicate as if they were attached to the same wire
- **DHCP – Dynamic Host Configuration Protocol** is a network protocol that enables a server to automatically assign IP addresses to hosts on the same network.
- **XSS – Cross Site Scripting** is a computer security vulnerability normally found in web applications, where the attacker injects malicious data.
- **IPSec – Internet Protocol security** is a framework of open standards that help to ensure privacy and secure communication over IP.

- **PyCharm** – This Python IDE will provide a robust and scalable Integrated Development Environment. Some of its advantages are a graphical debugger, an integrated unit tester and it also enables the integration of code with version control systems, such as GitHub and GitLab.
- **Angular –** This is the web front end used to develop the user interface, which will be responsive and adopt a Restful API backend integration approach.
- **PyBrain –** This is a modular Machine Learning Library for Python, flexible, easy-to-use and used in a variety of development environments to test and compare algorithms.
- **Paramiko** – It is a Python Library used interact with other devices on the network via the SSHv2 protocol and requires **pip** to run. It also uses the Cryptography Python Library ton encrypt the SSH connections.
- **xmltodict** – This is an XML Manipulation Library which is lightweight and simple, to implement and will help to easily parse device output.
- **mysqldb** – A library used to interact with SQL databases.
- **ESXi –** Virtualisation OS running on a bare-bones server allowing me to create a dynamic development environment.
- **Docker –** This is a software virtualisation technology providing an additional layer of abstraction and automation of client operating systems, such as Windows and Linux.
- **Ansible –** An open-source automation engine, helping automate software provisioning and configuration management. I will be using it to help me develop a standardised networking language.

2. GENERAL DESCRIPTION

2.1. SYSTEM FUNCIONALITY

The platform will allow network engineers as well general technical staff to easily make changes on the network. Over time the Assisted Machine Learning Module will aid the system in making the same changes automatically.

Initially they will include simple ones, such as interface address assignment and general Layer 2 functionality, VLAN setup and DHCP services. Support for more complex tasks such as static and dynamic route configuration will be added further in the development cycle. Finally, the most challenging feature will be automating network security, such as firewall zones, policies, access control, etc.

The user interface for this system will be easily reachable via a Web Application and gives users access to their network from any remote site. Device status alterations can be easily made by using an intuitive network map and pushing real time changes to it.

Unlike many systems in the networking industry, it will be developed and distributed as a subscription model whereby the potential customers will not be investing in any hardware upfront.

2.2. USER CHARACTERISTICS AND OBJECTIVES

The main user group, most likely to benefit from this network automation platform are network engineers, however general IT personnel would as well.

No features of the system will directly benefit the end user, however as it will eliminate human introduced errors, the likelihood of downtime is reduced, therefore positively impacting user experience.

I am aware that the target market is quite narrow and to further define the most common users I have developed two sample personas. There will be background information on each fictional user and more detail on what the expectations and requirement they might have from this automation platform.

** The photographs used are public domain, royalty-free.*

Edward	**Story**
	Edward is a 40-year-old Network Engineer, with over fifteen years of experience in the industry. Preferring a more hands on approach to interacting with and configuring devices, using a CLI instead of Web Interfaces. He has a lot of networking knowledge, however has limited experience scripting or automating network tasks. He has many certifications from vendors such as Cisco, Juniper Networks, etc.
Demographics **Age** 40 **Occupation** Network Engineer **Tech literacy** Very High **Residence** Ireland	**System expectations and requirements:** • Looks to automate time intensive tasks, which have normally been delegated to other engineers • Improve the efficiency of manually configuring devices, by using network maps offered by the system instead of obscure CLIs. • Push changes to multiple devices at once using the platform, instead of one at a time.
Sean	**Story**
	Sean is a 25-year-old IT Administrator in a finance firm. He has recently been promoted from an IT helpdesk job. He has a diverse knowledge of IT, ranging from enterprise software, server expertise to networking skills, however all are limited. To carry out any significant changes on the network, he must contact the contracted network support company and wait for up to three days for the change to be implemented. He finds this very frustrating, as it directly impacts his productivity. He has requested a quicker turnaround from the network support company, however due to a much higher cost, it has been rejected by the finance department.
Demographics **Age** 28 **Occupation** IT Admin **Tech literacy** High **Residence** UK	**System expectations and requirements:** • Easily make changes on the network using the system, without contacting the support company. • Take advantage of the automated device reconfiguration functionality, increasing productivity. • The system will provide him with a failsafe, to avoid network downtime.

2.3. OPERATIONAL SCENARIOS

The scenarios outlined below, give a brief description of the most common user's objectives, as well as actions they will have to take. However, it is not an in-depth table, missing more rudimental fields such as action source and step number. This would greatly increase the table complexity and size.

Scenario ID	1
User Objective	To sign up online for the services offered by ASDN and set up the initial VPN connection to their network.
User Action	The user will enter their personal details and receive a confirmation email which will allow them to set up two-factor authentication using the Google Authenticator App. Thereafter, they will be greeted with a VPN setup screen on their account dashboard. They will fill the form in and a secure connection will be established between their network and our system.
Comments	There are some pre-requisites for this Objective to be successful. Firstly, the user requires to have a valid email address. Secondly, Google's Authenticator App is needed to complete the two-step verification setup. Finally, the user needs to correctly configure their VPN device to establish a successful connection between the two sites.

Scenario ID	2
User Objective	To make manual network changes using the provided network diagram. For example, change an interface IPv4 address.
User Action	The user will log onto the ASDN dashboards and click on the device on their network map they wish to configure. Then, enter the address they want and click on the commit button. They will be notified within seconds whether the configuration change has been approved and applied by the system.
Comments	There main assumption made here is that the user has some networking knowledge, however even if they make a mistake, the system will prevent this action.

Scenario ID	3
User Objective	To monitor the changes that the system automatically applies to the network.
User Action	The user will log onto the ASDN dashboards and look at recent changes the system has made. If some of them are incorrectly applied there will be a rollback option available.
Comments	The platform can be used as a monitoring tool, even though that is not its main intended purpose. There will be very little user input required when the fully autonomous mode is operational.

2.4. CONSTRAINTS

There are many constraints related to this approach of developing a network automation system. I will elaborate further on the most impactful ones below:

1. The user must have elementary networking knowledge to take full advantage of the system functionality offered. It is however possible to allow the system to run fully autonomously for extended periods of time.

2. A VPN capable firewall is also needed on the user side to ensure the most secure possible connection technology is used to accommodate communication between the two networks. This a device deployed on most sites, where a company has internet access.

3. As mentioned previously, a limitation of the system is that it will initially support only the largest network device manufacturers, however this is an addressable limitation, which can be improved upon in future software releases.

4. Automation of some of the more niche tasks will not be supported. These mainly include very vendor specific ones and do not directly impact core network operation and functionality.

5. The system will not be able to operate correctly without a constant connection to the remote customer network. This is because of the client-server deployment model, a constant communication stream needs to be maintained between the application server and the remote customer network.

3. FUNCTIONAL REQUIREMENTS

The functional requirements shown below give a general description of each requirement, as well as issues that are very likely to be encountered during development.

Requirement ID	1
Description	The web application must allow the user to securely create an account and log in using two-factor authentication. In addition, an email validation token needs to be confirmed by the user.
Criticality	This is the key requirement of the system, as without this feature the user would not be able to access the application UI.
Technical Issues	This involves deploying a web-server, a mail server and implementing two factor authentication libraries to securely log the user in. Additionally, all the dependencies will have to be deployed, such as a web, mail, database server, etc. Most of the work will be spent making the registration and login process secure, with SSL other technologies and by preventing XSS and other similar attacks.
Dependencies	N/A

Requirement ID	2
Description	A back-end system will be needed that will configure a local VPN capable firewall. The user will enter VPN connection details as soon as they initially log into the web app. This information will be used to enable the back-end of the system to communicate with the remote devices on the customer network.
Criticality	This is also a curtail requirement, as the system would not operate at any capacity without being able to communicate with network devices.
Technical Issues	This requires a program to be developed that can automatically configure a site to site IPSec VPN between the local and remote customer site. This is challenging as if the automated configuration malfunctions a user will need to fix it.
Dependencies	This depends on requirement 1 as if there is no web server, the system has no remote network details and cannot be set up. In addition, there is another pre-requisite, whereby the customer is required to configure a VPN connection on their side. This might look complex. However, most larger companies have staff capable of accomplishing such a task or can outsource it relatively cheaply.

Requirement ID	3
Description	The system must be able to connect to and communicate with devices located on a customer's network using a secure protocol such as SSH running over the previously configured VPN connection.
Criticality	This is a very important requirement, as without the ability to communicate with devices, the system cannot operate.
Technical Issues	This involves developing a formatted I/O stream between the system and the network devices. The data will most likely be in an XML format. It will also need to be able to parse output of many different device vendors such as Cisco, HP and Juniper Networks.
Dependencies	This depends heavily on requirement 1 and 2 as there is nothing to send or receive from the device if we can't make a connection.

Requirement ID	4
Description	System must be able to update necessary data stores.
Criticality	This is less important than establishing a connection as any data collected needs to be stored for future use.
Technical Issues	One of the main challenges is ensuring the database is used correctly, as well as formatted and accessible. It must conform to the 3rd normal form. In addition, security is paramount, as the database will contain very sensitive customer network details There might also be a need for two different database technologies, SQL and NoSQL. The former will be used for general data storage, whereas the later for the Assisted Machine Learning System.
Dependencies	This depends heavily on requirement 2 and 3, as there no data available for storage if connection to the remote network isn't hade and no data was transmitted.

Requirement ID	5
Description	A Networking Module will have to be developed to carry out several important tasks. The main ones include, network mapping, network translation and monitoring. There are many sub-requirements of this module, for example, to recognise device type, it's neighbouring nodes, translate a configuration of a device of any brand into a standardised format, etc.
Criticality	This module is very important as it is a core part of how the system operates. There would be no possibility of automation without it.
Technical Issues	There will be a variety of challenges tied to the Networking Module, however the main ones include the following: 1. Standardisation of network device configurations and operational states. 2. Creating a network status feature, that will be able to tell whether a neighbour device is up, traffic is transmitted successfully, etc. This will most likely be done with a variety of protocols, such as CDP, LLDP, SNMP, etc. 3. Mapping a network infrastructure for it to be visually represented in the Web UI.
Dependencies	This depends heavily on requirement 2, 3 and 4.

Requirement ID	6
Description	This is the requirement encompassing the Assisted Machine Learning Module. It will be used to automate the process of reconfiguring the devices on the network as well as checking the validity of human requested changes. It will use a variety of technologies and many sub-systems to accomplish this task.
Criticality	This module is the main differentiator between this system and its competitors. While being very important, the platform would still be very usable and useful even without it.
Technical Issues	Changing a device configuration requires the system to take many different parameters into consideration. For this it will require two rulebooks, one containing network specific rules, such as IP addressing best practices, VLAN numbering and device type consideration, etc. Whereas the second will contain the rules which the system must follow to change and adapt the first network specific rulebook. This will be the most challenging aspect of this project. This is due to it being able to damage very expensive devices, therefore any errors must be caught quickly and dealt with gracefully. This can be achieved by constantly monitoring the effects of the changes and allowing for configuration rollbacks.
Dependencies	This depends heavily on requirement 2, 3, 4 and 5 to operate correctly.

Requirement ID	7
Description	Cross-platform client support and multi-vendor nature are necessary for this platform to be commercially viable. However, as my access to devices from most vendors is somewhat limited, I will attempt to cover as many with a large market share as possible. Initially I will concentrate on Juniper Networks and Cisco and will further expand to devices from HP, Palo Alto and Extreme Networks.
Criticality	This is not a critical feature, however the larger the vendor support portfolio, the more usable the system is in an actual deployment scenario.
Technical Issues	This will be somewhat challenging, as some vendors do not employ a hierarchical configuration structure which is what the system will be leveraging to build a tree-set of possible command completions.
Dependencies	This requirement is independent of any previous as it can be developed separately with only requirement being access to different brands of networking equipment.

4. SYSTEM ARCHITECTURE

To develop a platform that will solve the outlined problems, a client-server architecture will be used. It will mainly be made in Python, as it's an excellent scripting language and recommended as a network automation language by many different device manufacturers, such as Juniper, Cisco etc. My development environment will run on top of an ESXi server.

My IDE of choice is PyCharm, as it is very powerful and automates some testing processes such as code coverage. I will use Angular as my Web-Front End, due to its versatility and Restful API calls. Most user testing will be performed using Docker, due to its consistency and reliability.

4.1. NETWORKING MODULE

This module is responsible for communicating with devices located on the customer's network using a library called paramiko to establish an SSH connection and xmltodict to encode and decode XML data streams.
Another responsibility will be to create a standardised network map, save it (using the mysqldb library) and send it on to the Assisted Machine Learning Module.

The map creation will be done with the use of Ansible, as this is a standardised automation language. It will also need to track the status of the remote network to prevent any possible issues, which could potentially bring down the links between devices.

4.2. ASSISTED MACHINE LEARNING MODULE

This module will manly leverage the PyBrain, xmltodict and Ansible dependencies. There will be a great deal of research vested into developing this module and its use of Rulebooks and self-governing principles.

My current approach is to use two rulebooks for the Assisted Machine Learning Process. One will dictate how the program can change the other rulebook and can be only edited manually, whereas the second will change constantly based on user's actions on the network. They will evolve and improve over time, becoming more accurate and self-sufficient.

The results that are made from these rulebooks are translated to the standardised format and passed back to the Networking Module to be pushed out to a device on the customer's network.

5. HIGH-LEVEL DESIGN

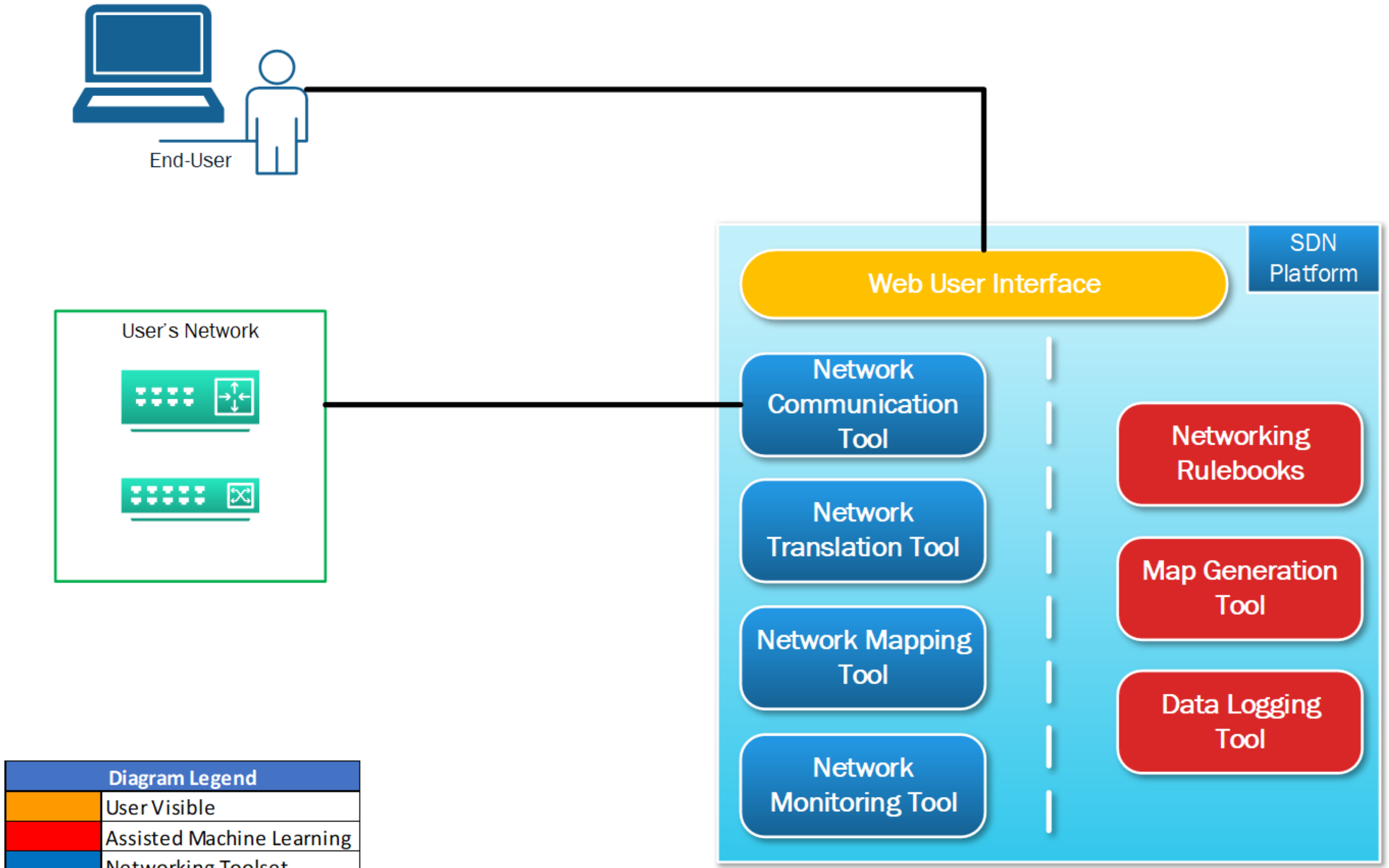
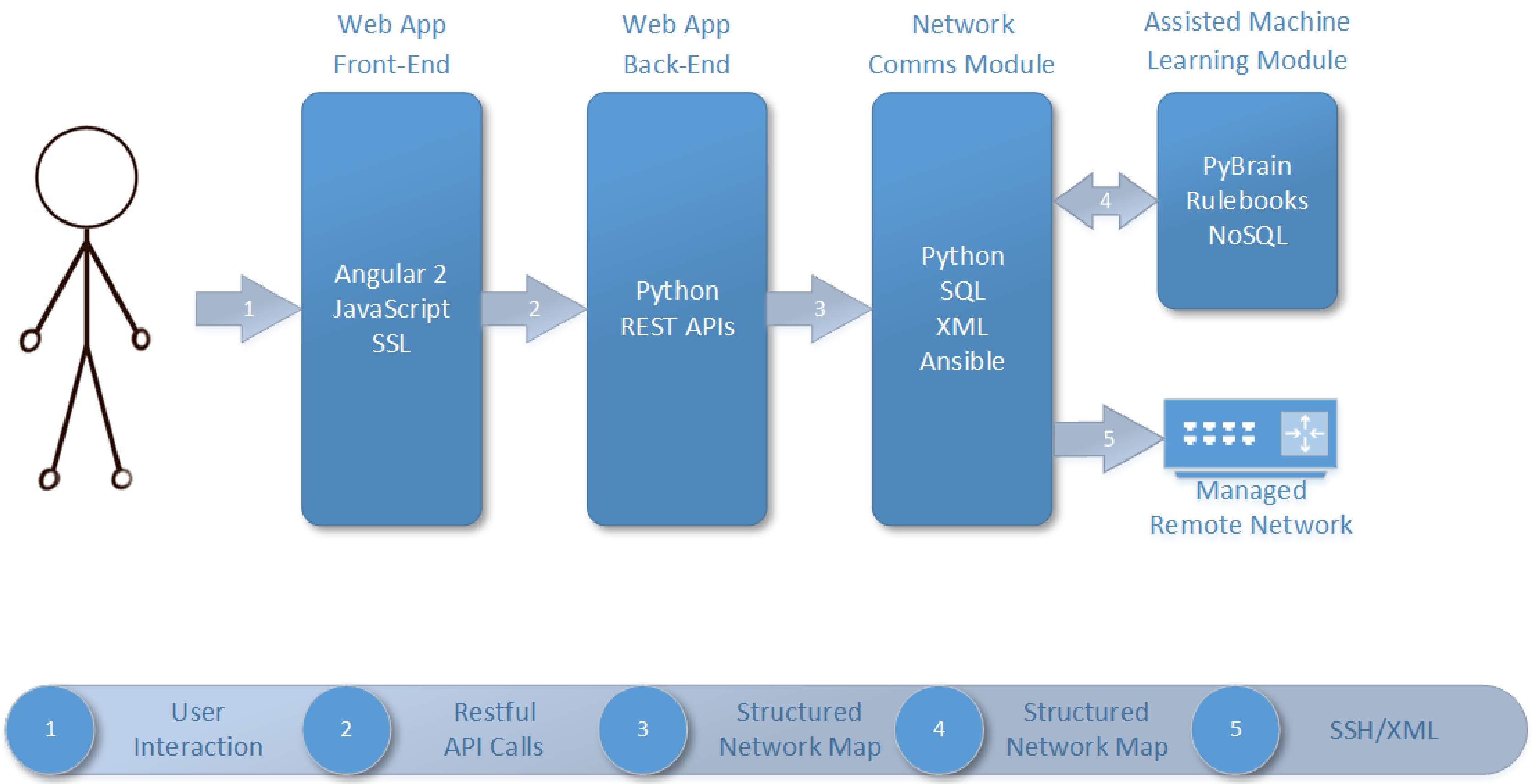
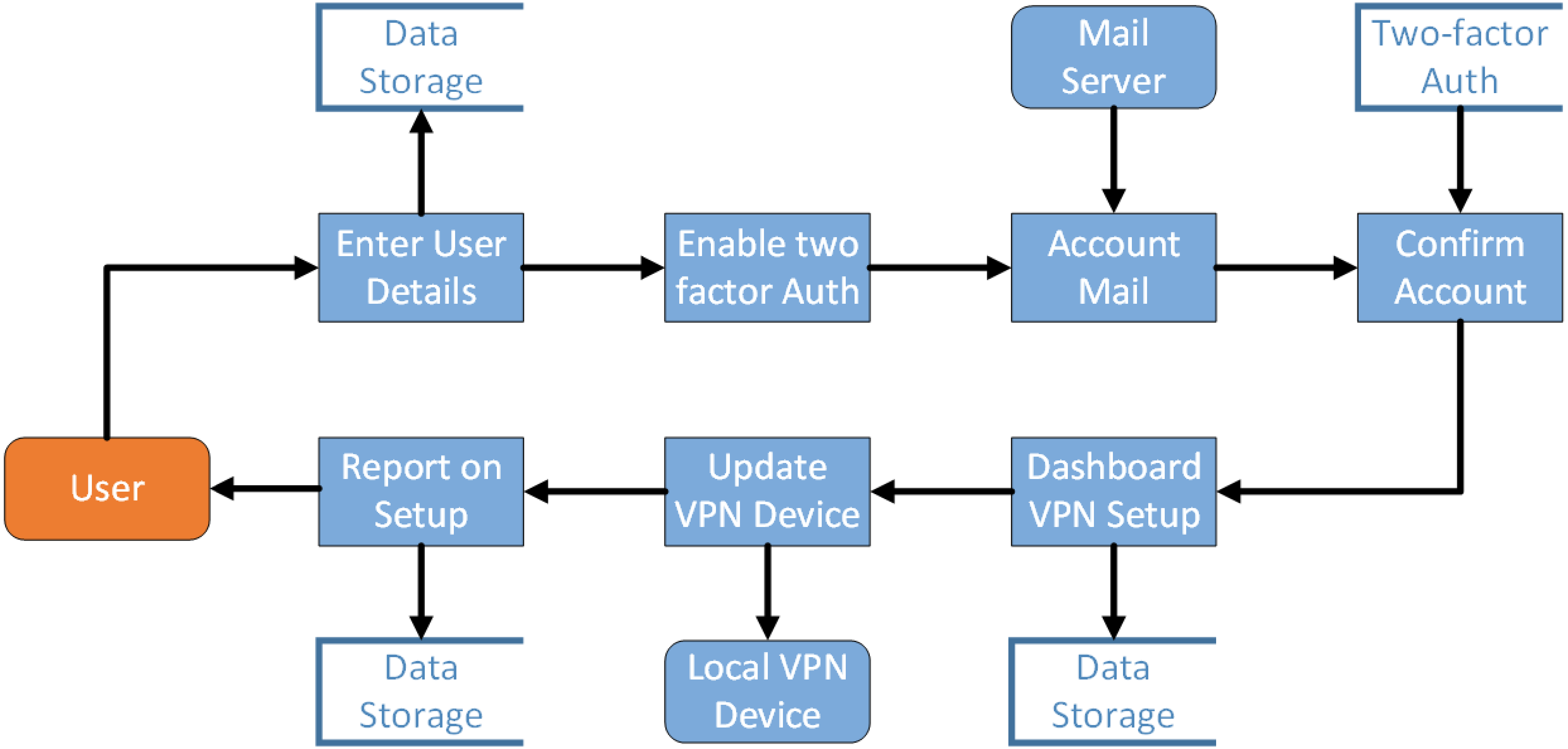


Diagram Legend	
🟧	User Visible
🟥	Assisted Machine Learning
🟦	Networking Toolset
🟩	External Netwrok

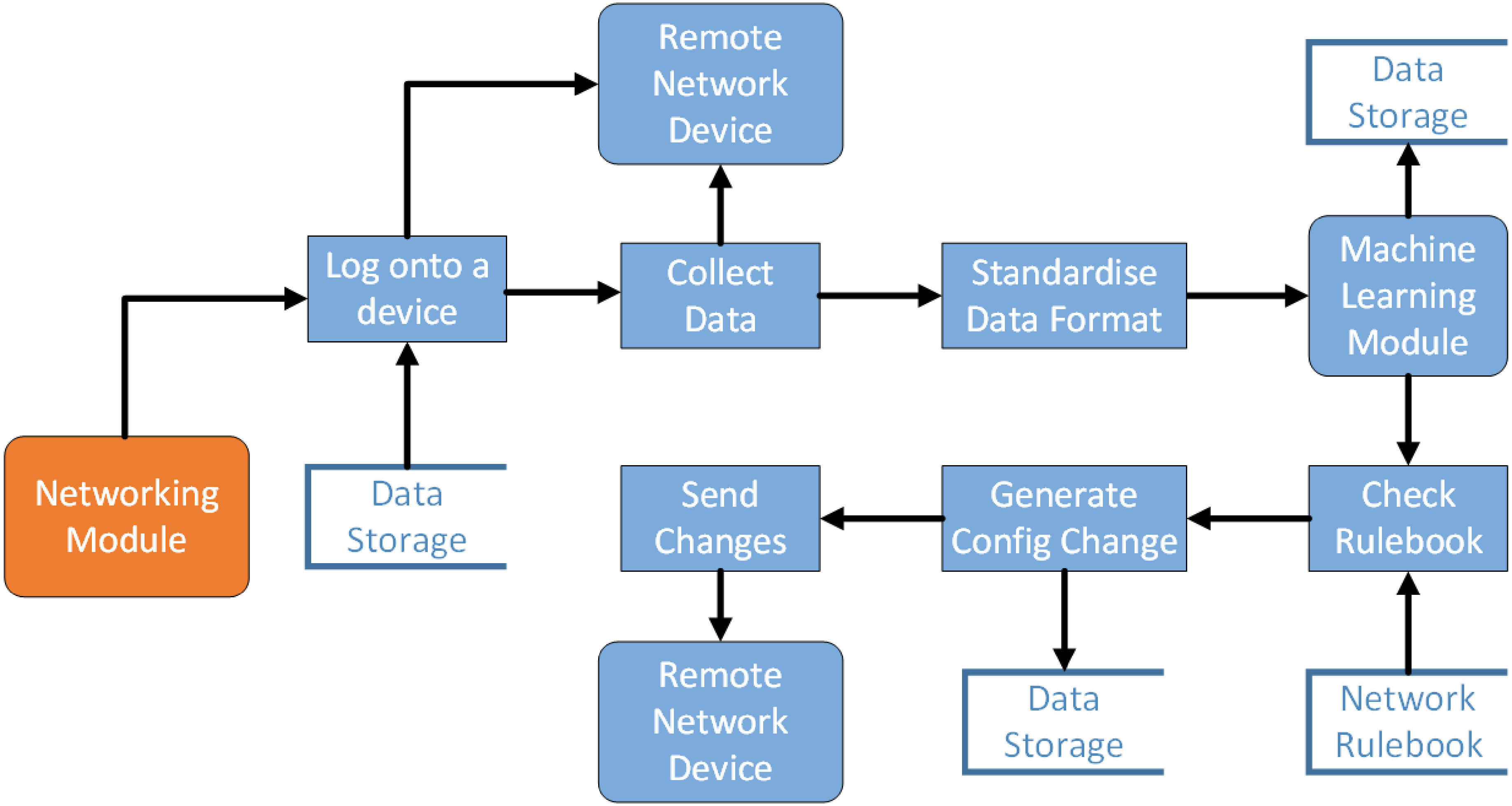
5.1. APPLICATION ARCHITECTURE DIAGRAM



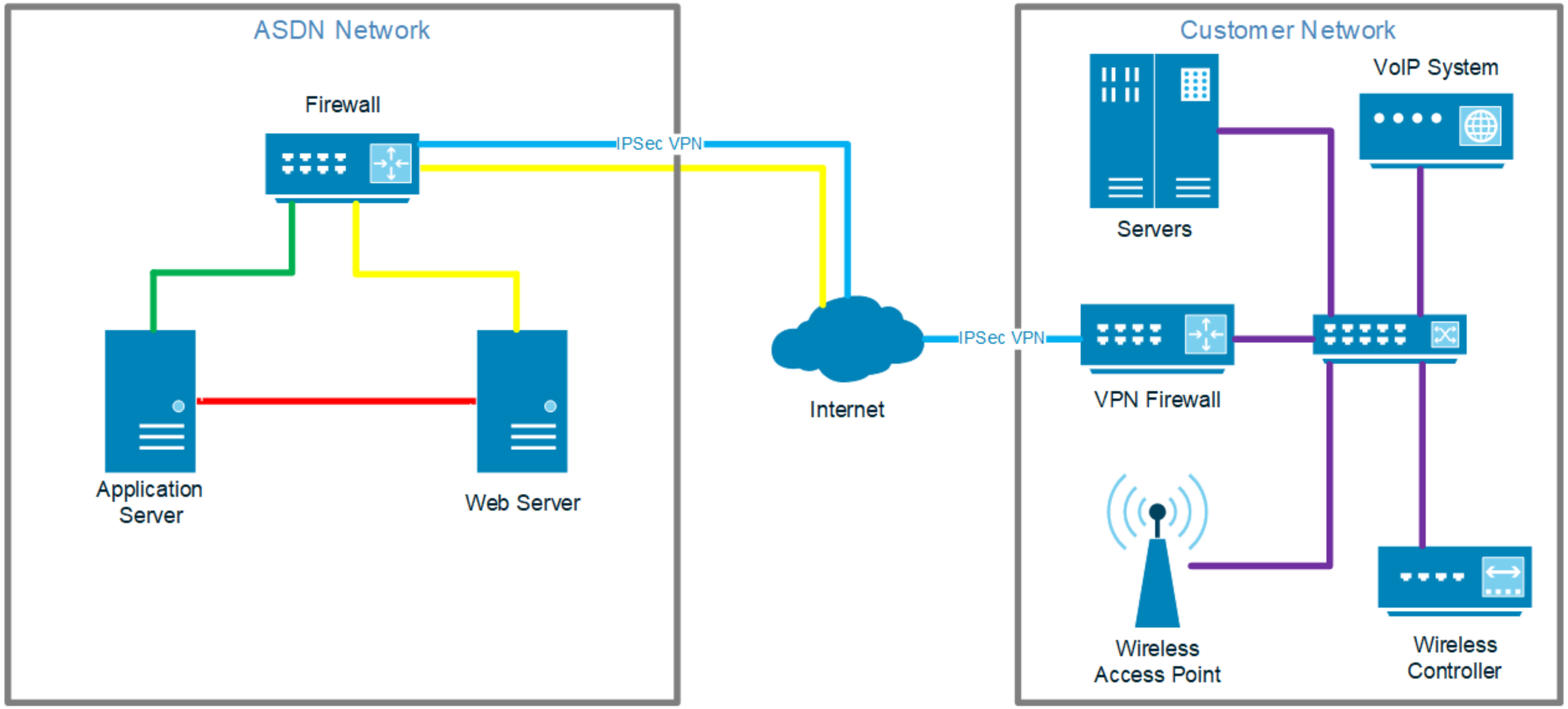
5.2. DFD – USER REGISTATION AND SETUP



5.2. DFD – SYSTEM OPERATION



5.3. NETWORK DIAGRAM



Network Legend			
🟥	Hardware Conncection	🟪	Customer Network
🟩	VPN Access	🟦	Site to Site VPN
⬛	Network Boundary	🟨	Web Access

6. PRELIMINARY SCHEDULE

6.1. SCHEDULE OVERVIEW

- The project plan timeline has been structured in such a way, that activities are carried out in the most efficient manner. This is achieved by using the Critical Path Method and allowing for the shortest time possible to complete the project. In addition, I will attempt to work on as many tasks in parallel as possible

- A Scrum development approach has been adopted and I will use trello to keep track of all remaining and completed tasks. There also are weekly supervisor meetings with Mr. Brian Stone.

- There are no time-bound technology dependencies. Most software and hardware licencing I have purchased work for more than a year. The only exception being, the SSL certificate for the Web Server, which allows me to have https capabilities. This is very important if the project was to be released, however for the time being I will apply a 90-day licence on, before demonstrating the project next year.

- A lot of time will be spent testing using a lab network that I will have to set up off site in an Agile Networks Managed Services Datacentre.

6.2. GANTT CHART

Starting 02/10/17	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11
Requirements Analysis	🟧										
Design Analysis	🟦										
Project Planning	🟧										
Project Proposal	🟦	🟦									
Set up dev environment			🟧	🟧							
UI Prototype			🟦	🟦							
Functional Specification					🟧	🟧					
Network Module Prototype							🟦	🟦	🟦		
Automated VPN Prototype							🟧	🟧	🟧		
Launcher Prototype									🟦		
Interconnecting Systems										🟧	
Preliminary User Testing										🟦	🟦
Documentation	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧
Project Minutes	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦

Starting 02/10/17	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21	Week 22
Networking Module Build	🟦	🟦									
Automated SND Prototype	🟧	🟧									
GUI Build		🟦	🟦								
Launcher Build			🟧	🟧							
Overall Build				🟦							
Test Planning				🟧							
Test Development					🟦						
Boundary Value Testing					🟧	🟧					
Equivalence Testing					🟦	🟦					
Decision Tables						🟧	🟧				
Functional Testing						🟦	🟦				
Structural Testing							🟧	🟧			
Path Testing								🟦			
Data Dependence								🟧	🟧		
Integration Testing									🟦		
Testing Review & Write-up									🟧	🟧	
Code Review										🟦	🟦
Documentation	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧
Project Minutes	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦