PPPoE Implementation Example

[.Fluidity: *Bring your telecoms infrastructure to the future*]

The Computer Hardware Lab

By **Charalampos Kapolonaris (Technical Lead)**
& **Vasilios Koutlas (Software Tester)**

.Fluidity PPPoE Implementation Example

The Addressing Scheme

.FLUIDITY PPPoE EXAMPLE SYNOPSIS			
.Fluidity Devices	**External IP Address/External Interface**	**.Fluidity Device Password**	**List of Internal Interfaces on each .Fluidity Device**
.Fluidity Server	static_ip_from_isp/ppp0	_Flu1d1ty-S3rv3r-2020	Network 1: eth0
.Fluidity Client	static_ip_from_isp/ppp0	_Flu1d1ty_Cli3nt	Network 2: eth0

PPPoE VPN TUNNELING ADDRESING SCHEME					
Tunnel Connections	**Server Address**	**Client Address**	**Subnet Mask**	**Network Address**	**Broadcast Address**
Server - Client (VPN NETWORK)	192.168.54.1	192.168.54.2	255.255.255.252	192.168.54.0	192.168.54.3

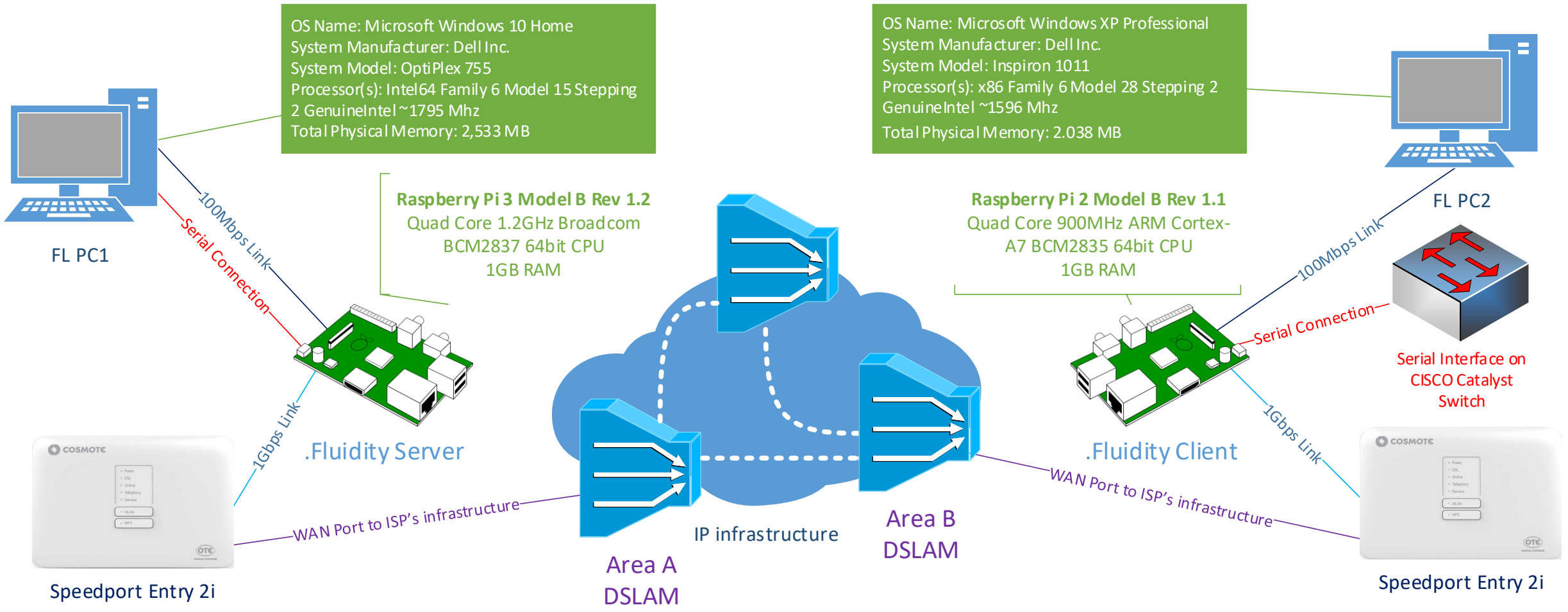
SERIAL TUNNELING SCHEME	
Serial Tunnel to .FL Server	**Serial Tunnel from .FL Client 1**
FL Server PC_1 -> .Fluidity Server	.Fluidity Client 1 -> Remote Console System (FL 1.1)

.Fluidity PPPoE Implementation Example

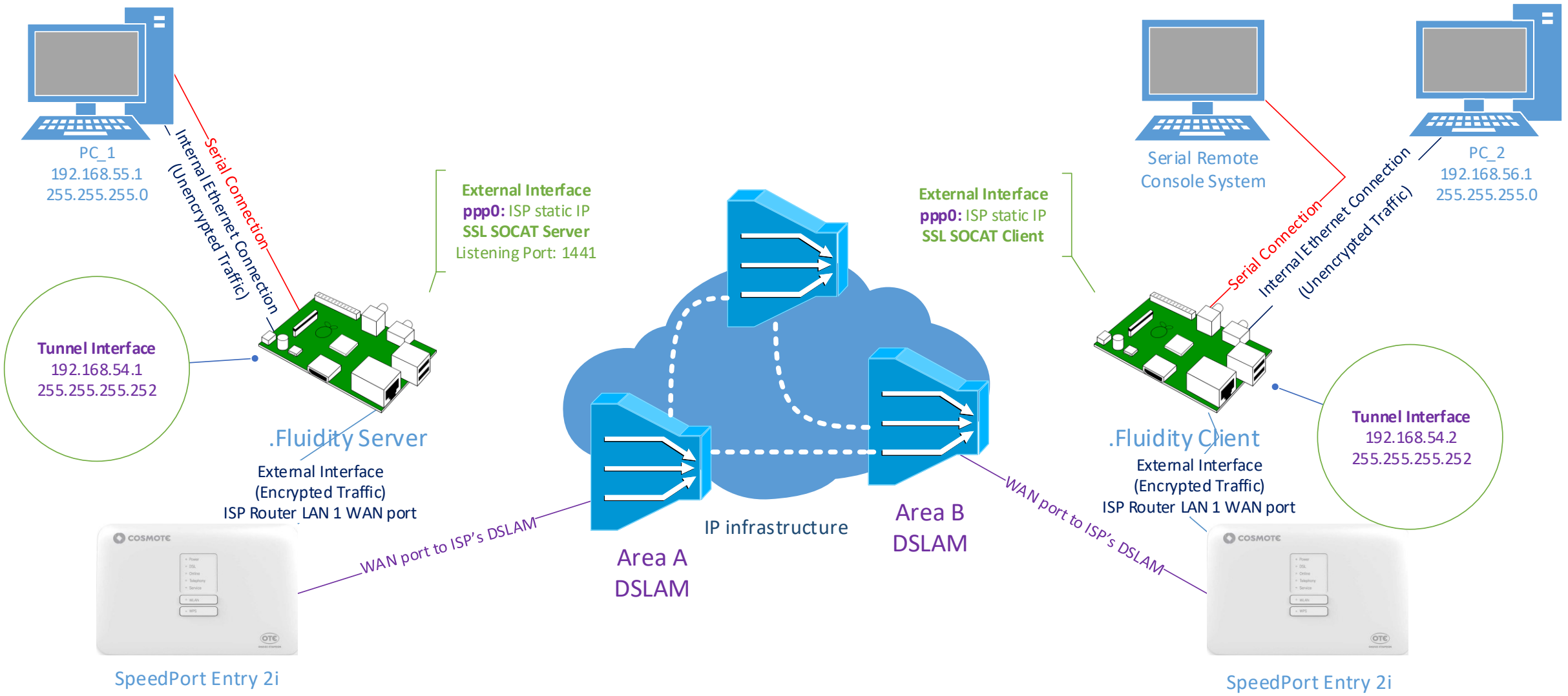
The Networking Scheme in Detail

Devices Attached and the Networking Addressing Scheme in Detail	
Network 1	
CIDR Network Address	**192.168.55.0/24**
PC_1:	**192.168.55.1**
.Fluidity Server:	**192.168.55.254**
Broadcast Address	**192.168.55.255**
Network 2	
CIDR Network Address	**192.168.56.0/24**
PC_2:	**192.168.56.1**
.Fluidity Client 1:	**192.168.56.254**
Broadcast Address	**192.168.56.255**

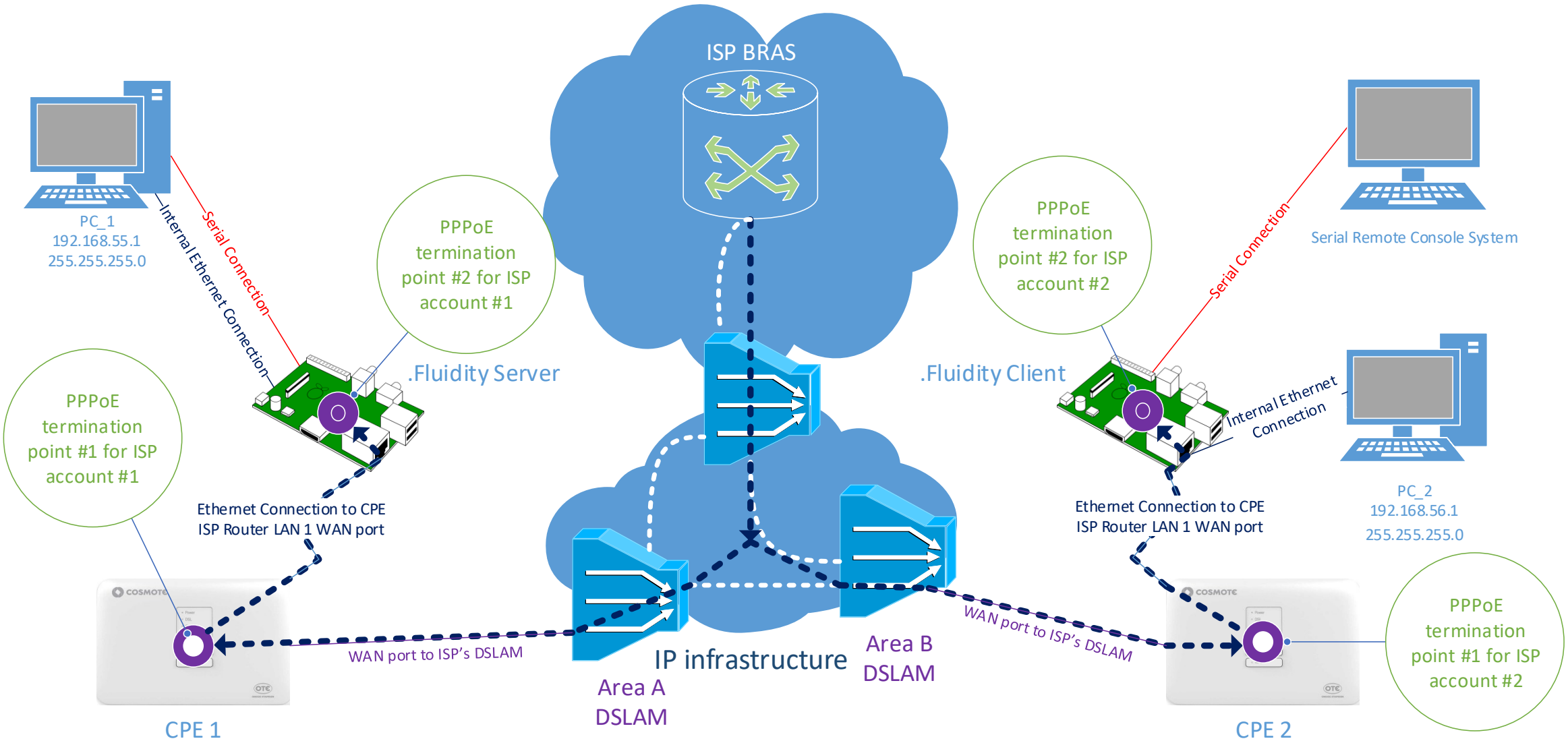
.Fluidity PPPoE Example – The Lab's Technical Specs



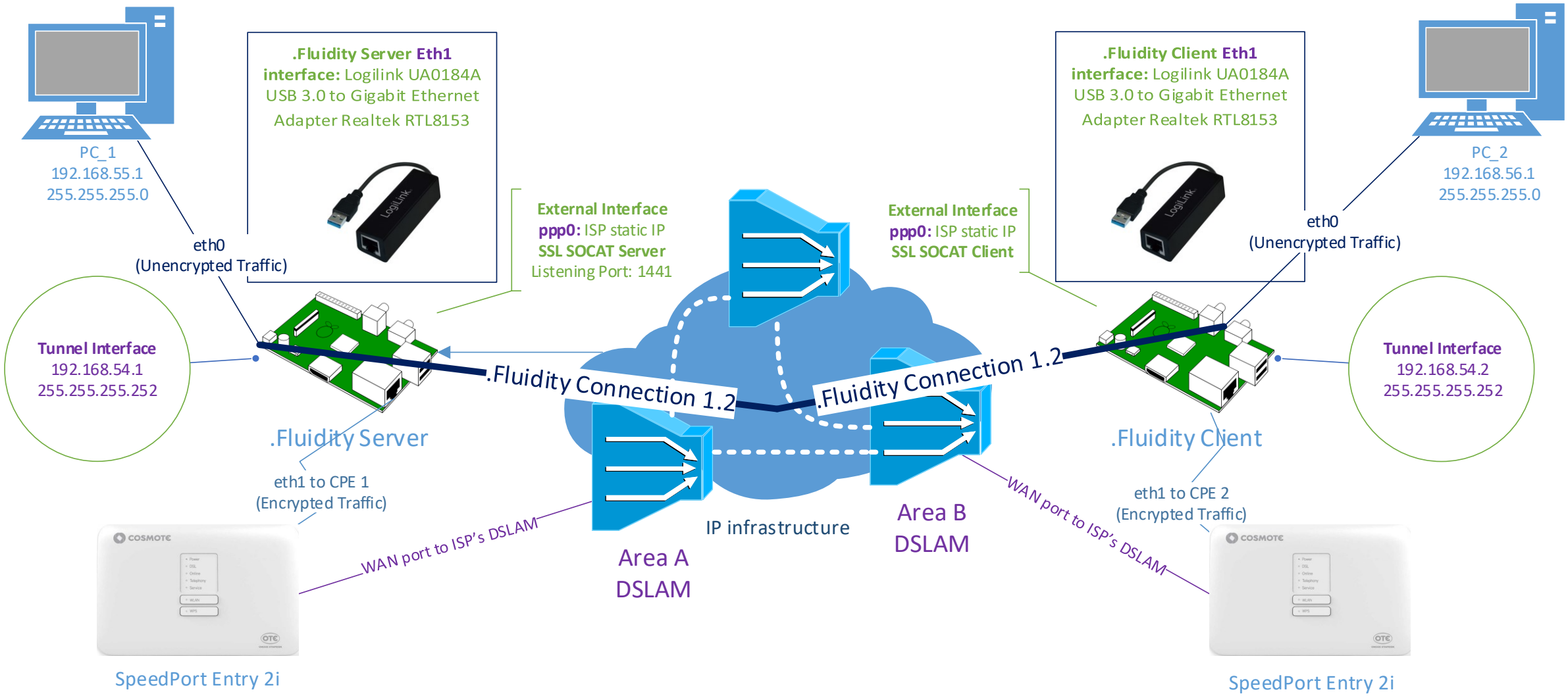
.Fluidity PPPoE Example - Visual Overview



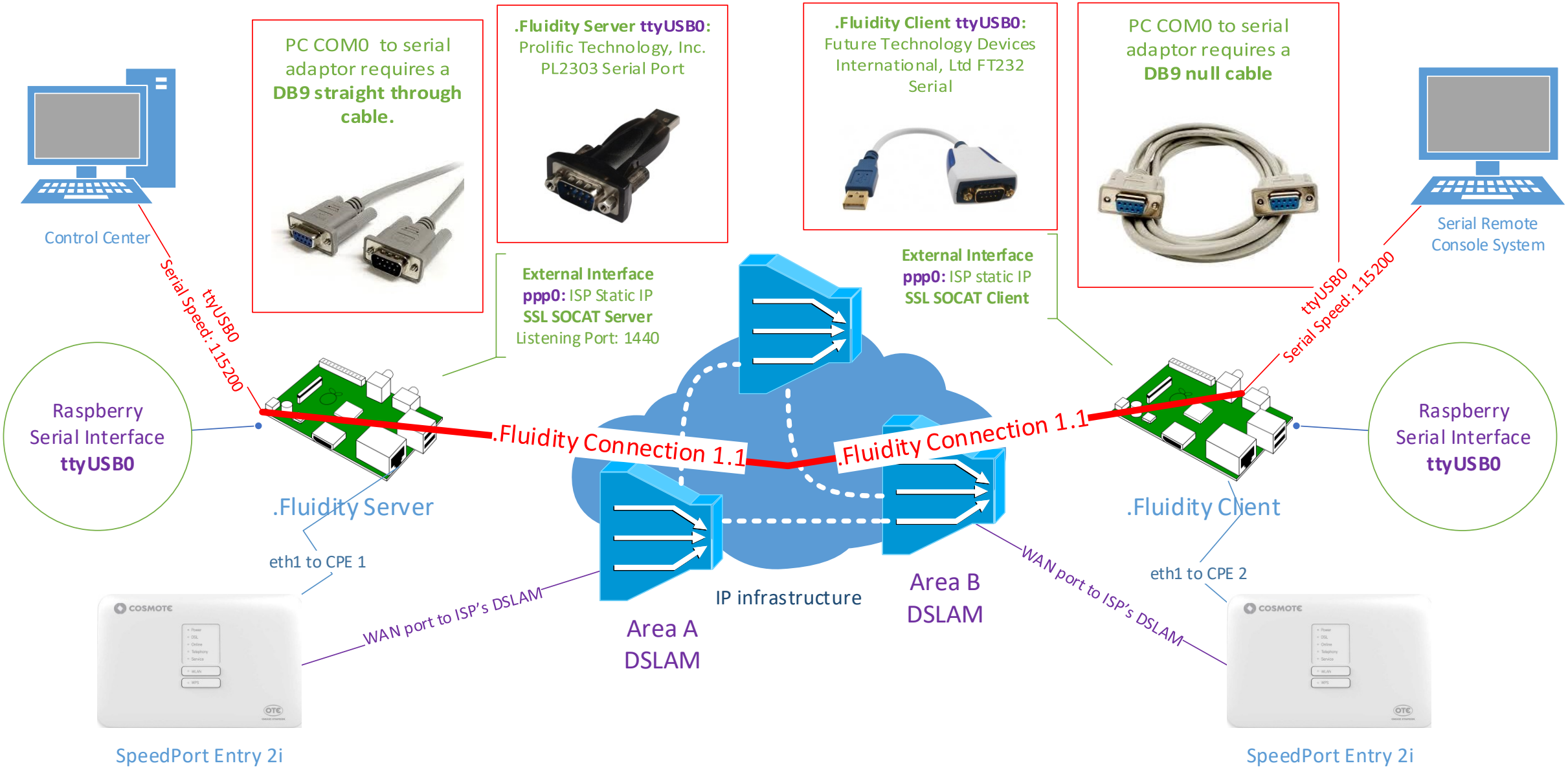
.Fluidity PPPoE Example PPPoE Termination Points



.Fluidity PPPoE Example - VPN Networking Visual Overview

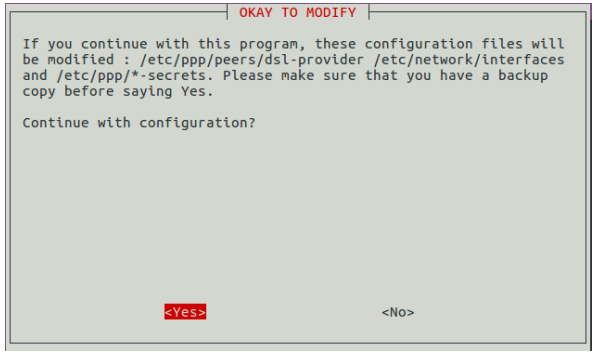


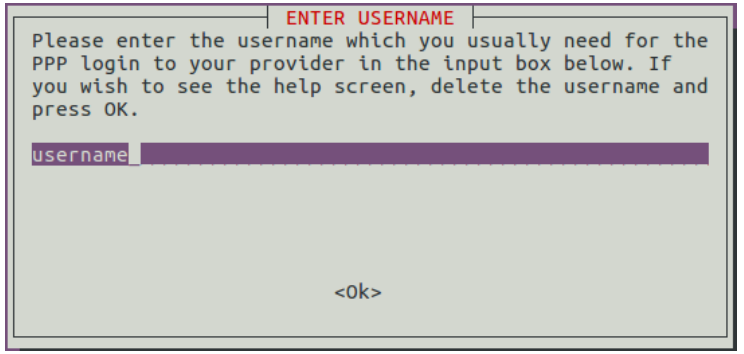
.Fluidity PPPoE Example - Serial Connectivity Visual Overview

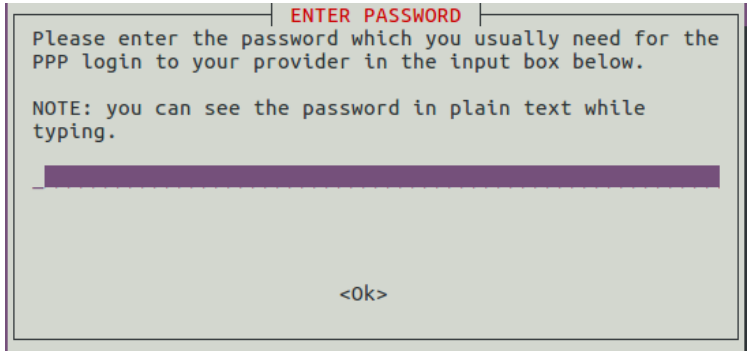


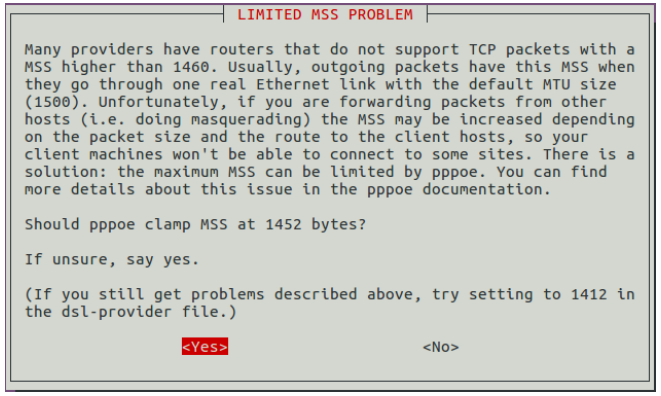
.Fluidity PPPoE Example. Install and Setup pppoeconf on each .Fluidity device (leave default settings).

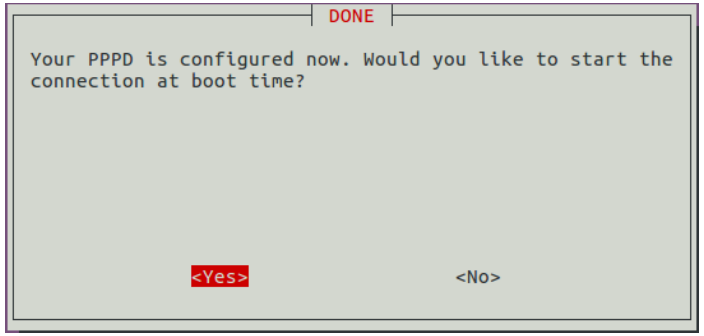
Actions Performed	BASH Commands
Install PPPoE	`sudo apt-get install pppoeconf`
Setup PPPoE	`pppoeconf`











.Fluidity VLAN Implementation Example

Step by step .Fluidity installation and configuration…

Actions Performed	BASH Commands
Load .Fludity's Command Line Interface (CLI)	`source Fluidity.sh`

Actions Performed	.Fluidity Commands
Install the .Fluidity Server to device	`installFluidity`
Add the .Fluidity clients	
Add Client	`addFluidityClient 1 static_server_ip static_client_ip _Fluidity_Client_1 pi`
Attach networks to .Fluidity VPN network	
Attach network 1 to Server which is on eth0	`setInternalInterface eth0`
Attach network 2 to Client which is on eth0	`Execute locally on Client 1: setInternalInterface eth0`
Add .Fluidity client connections	
Add a connection to Client	`addFluidityConnection 1 1`
Add a second connection to Client	`addFluidityConnection 1 2`
Do routing from .FL Server (Network 1)	
to Network 2	`addServerRoute ip route add 192.168.56.0/24 via 192.168.54.1`
Do routing from .Fluidity Client (Network 2)	
to Network 1	`addClientRoute 1 2 ip route add 192.168.55.0/24 via 192.168.54.2`

.Fluidity VLAN Implementation Example

Lighting the Spark…

List of VPN Connections					
.Fluidity Client 1 Connection Parameters					
Connection Number	**Connection Type**	**Server Listening Port**	**Server Tunnel Interface IP**	**Client Tunnel Interface IP**	**Tunnel Network Subnet Mask**
2	Tunnel	1441	192.168.54.1	192.168.54.2	30
Initiate .Fluidity VPN connection to Client 1		`runFluidity -t 1 2 1441 192.168.54.1 192.168.54.2 30`			
Stop .Fluidity VPN connection to Client 1		`stopFluidity 1 2`			

List of Serial Connections					
.Fluidity Client 1 Connection Parameters					
Connection Number	**Connection Type**	**Server Listening Port**	**Server Serial Interface**	**Client Serial Interface**	**Serial Speed**
1	Serial	1440	ttyS1	ttyS1	115200
Initiate .Fluidity serial connection to Client 1		`runFluidity -s 1 1 1440 ttyUSB0 ttyUSB0 115200`			
Stop .Fluidity serial connection to Client 1		`stopFluidity 1 1`			

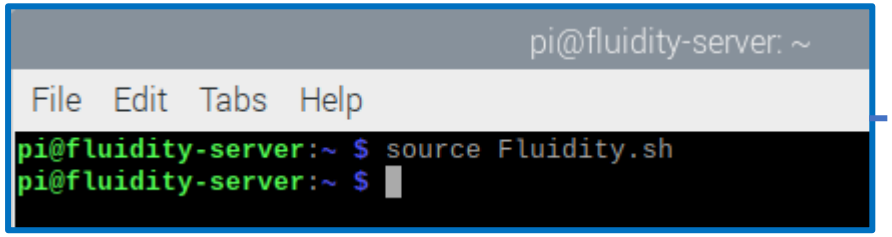
.Fluidity PPPoE Implementation Example

Step by step .Fluidity removal…

Actions Performed	.Fluidity Commands
Remove routes from .FL Server (Network 1)	
to Network 2	`removeServerRoute ip route add 192.168.56.0/24 via 192.168.54.1`
Remove routes from .Fluidity Client 1 (Network 2)	
to Network 1	`removeClientRoute 1 2 ip route add 192.168.55.0/24 via 192.168.54.2`
Remove .Fluidity client connections	
Remove the #1 Connection from Client 1	`removeFluidityConnection 1 1`
Remove the #2 Connection from Client 1	`removeFluidityConnection 1 2`
Remove the .Fluidity clients	
Remove Client 1	`removeFluidityClient 1`
Detach the networks from the .Fluidity VPN network	
Detach network 1 from Server which is on eth0	`removeInternalInterface eth0`
Detach network 2 from Client which is on eth0	`Execute locally on Client 1: removeInternalInterface eth0`
Remove the .Fluidity server	
ecryptFs unmount the .Fluidity Server folder	`BASH: sudo umount Fluidity_Server`
Remove .Fluidity main Server folder	`BASH: rm -r Fluidity_Server`

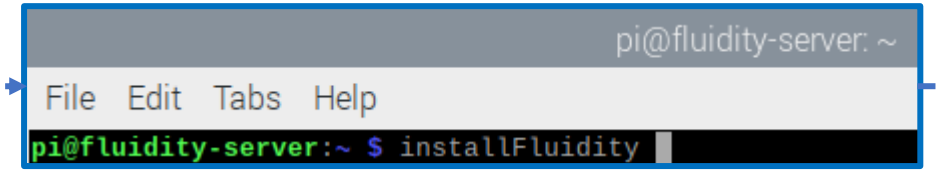
.Fluidity PPPoE Example – Visual Guided Tour

Step-by-step installation & configuration



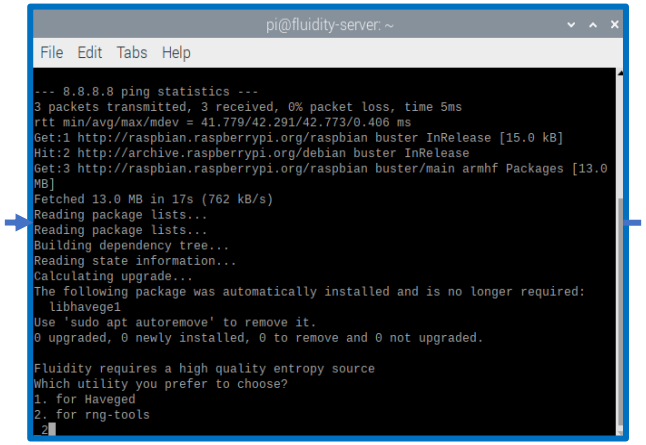
1. source Fluidity.sh

Import .Fluidity's command set



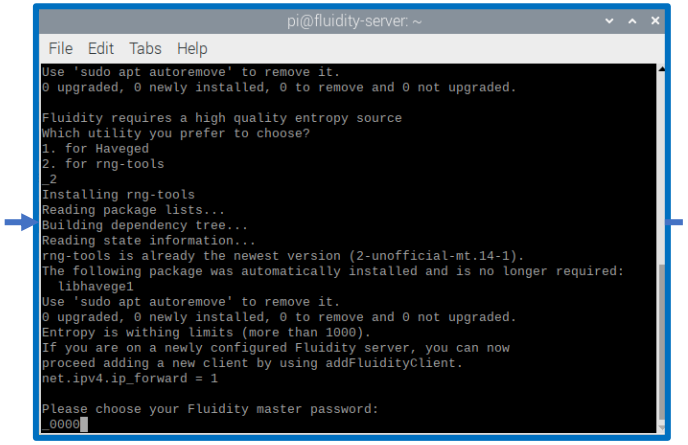
2. installFluidity

Begin the installation process



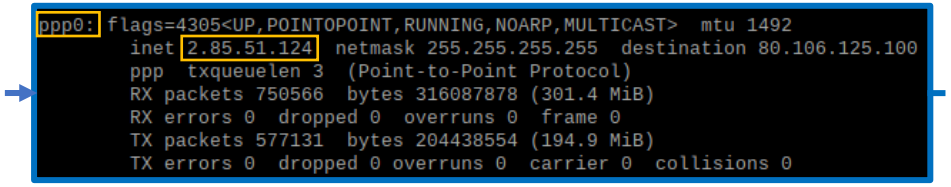
2.1 installFluidity

Install rng-tools (choice 2)



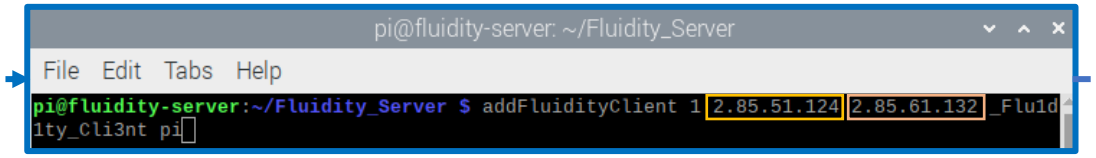
2.2 installFluidity

Give the master password (i.e. 0000)



3. ifconfig

For the ppp0 interface, observe the server's IP address



4. addFluidityClient 1

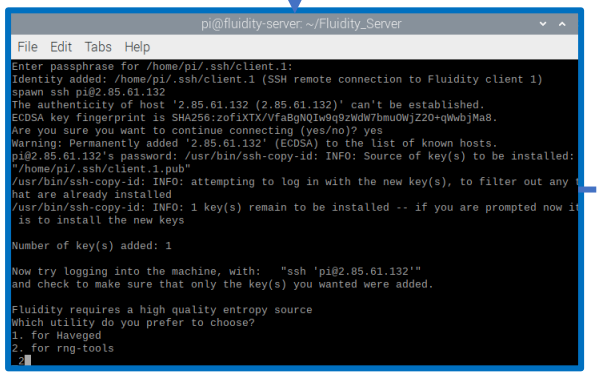
Add .Fluidity client 1 by using local and remote ppp0 IP address

Next Slide

.Fluidity PPPoE Example – Visual Guided Tour

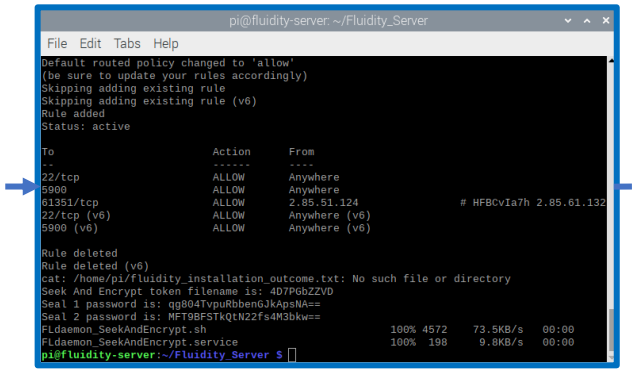
Step-by-step configuration

Previous Slide



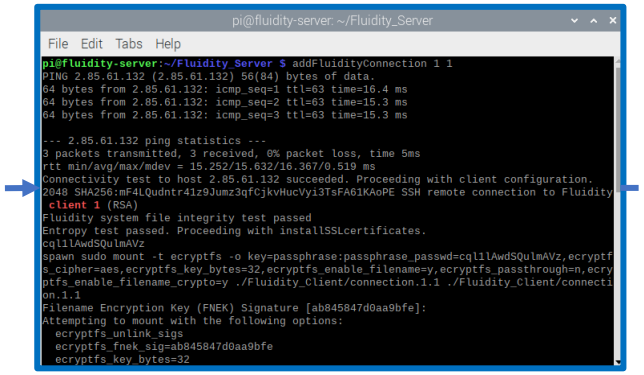
4.1 addFluidityClient 1

Client Side: Install rng-tools (choice 2)



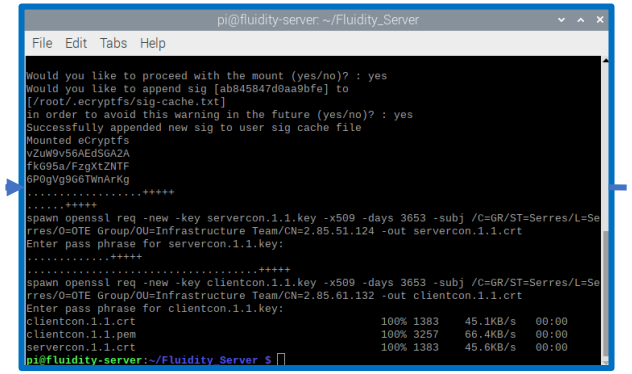
4.2 addFluidityClient 1

Finalizing client 1 installation



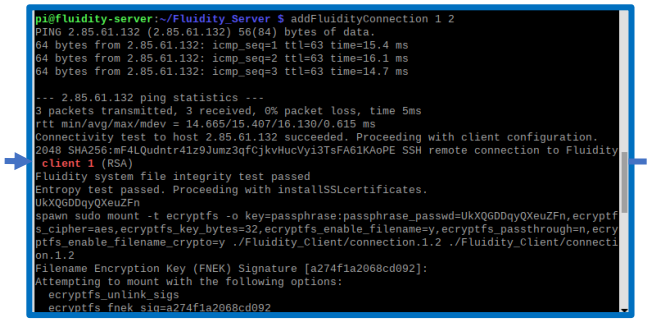
5. addFluidityConnection 1 1

Add connection 1 to client 1



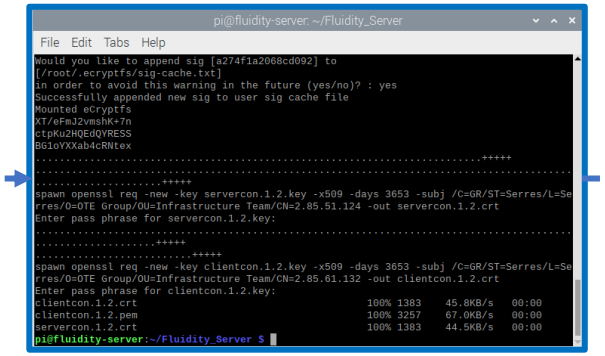
5.1 addFluidityConnection 1 1

Finalizing connection 1 1 installation



6. addFluidityConnection 1 2

Add connection 2 to client 1



6.1 addFluidityConnection 1 2

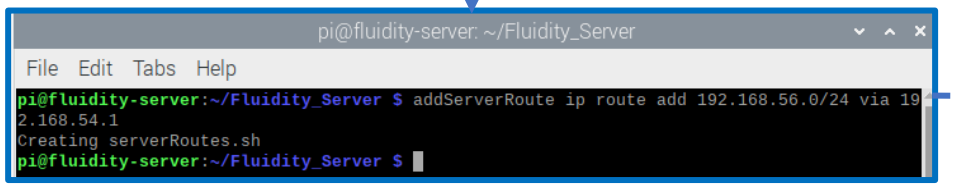
Finalizing connection 1 2 installation

Next Slide

.Fluidity PPPoE Example - Visual Guided Tour

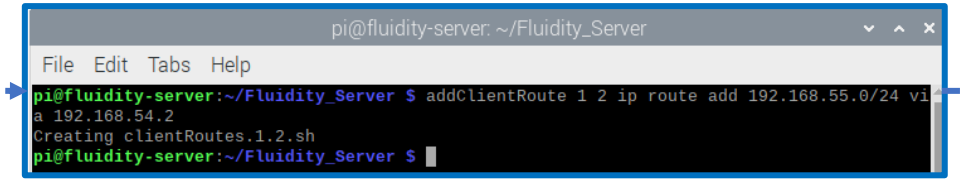
Step-by-step configuration

Previous Slide



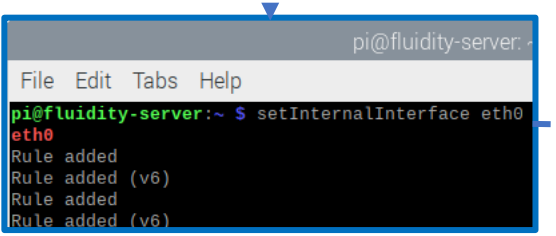
7. addServerRoute

Instruct the server's OS to reach network 192.168.56.0/24 via the IP address 192.168.54.1 (this will be the IP address of tun0 interface when runFluidity -t is executed).



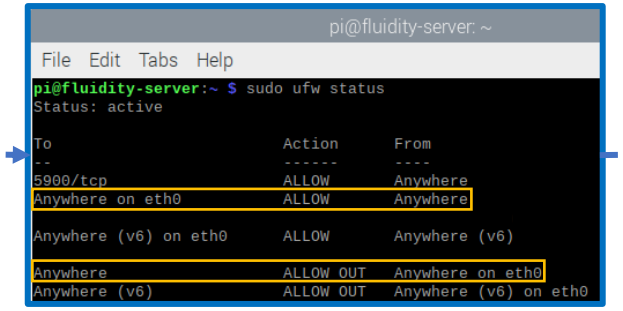
8. addClientRoute

Instruct client 1 connection 2 to reach network 192.168.55.0/24 via the IP address 192.168.54.2 (this will be the IP address of the tun0 interface when runFluidity -t is executed).



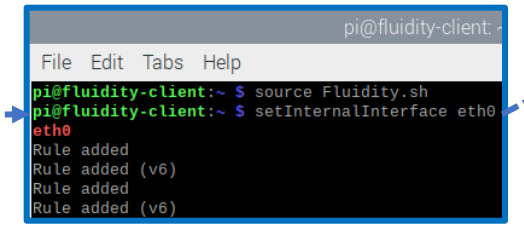
9. setInternalInterface eth0

Server Side: Designate interface eth0 as an internal interface, so that traffic can go through it freely.



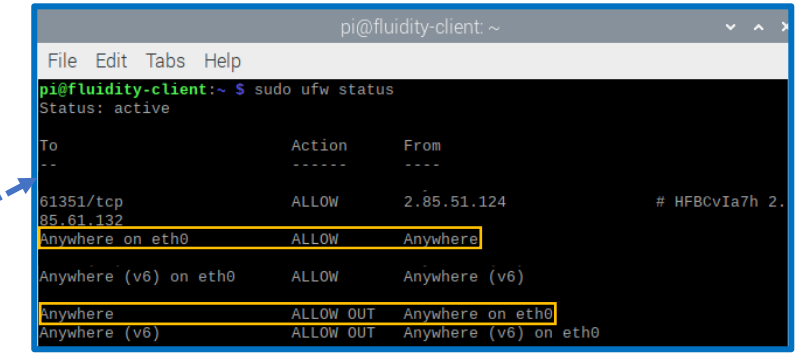
9.1 setInternalInterface eth0

Server Side: Observe the orange frames indicating the change on firewall rules.



10. setInternalInterface eth0

Client Side: Designate interface eth0 as an internal interface, so that traffic can go through it freely.



10.1 setInternalInterface eth0

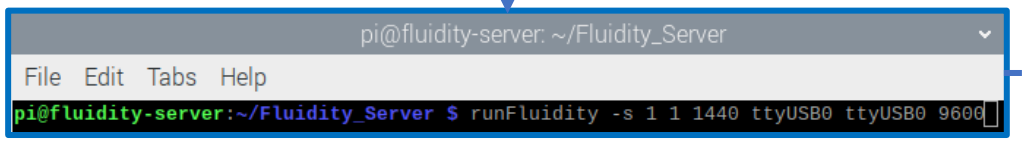
Client Side: Observe the orange frames indicating the change on firewall rules.

Next Slide

.Fluidity PPPoE Example – Visual Guided Tour

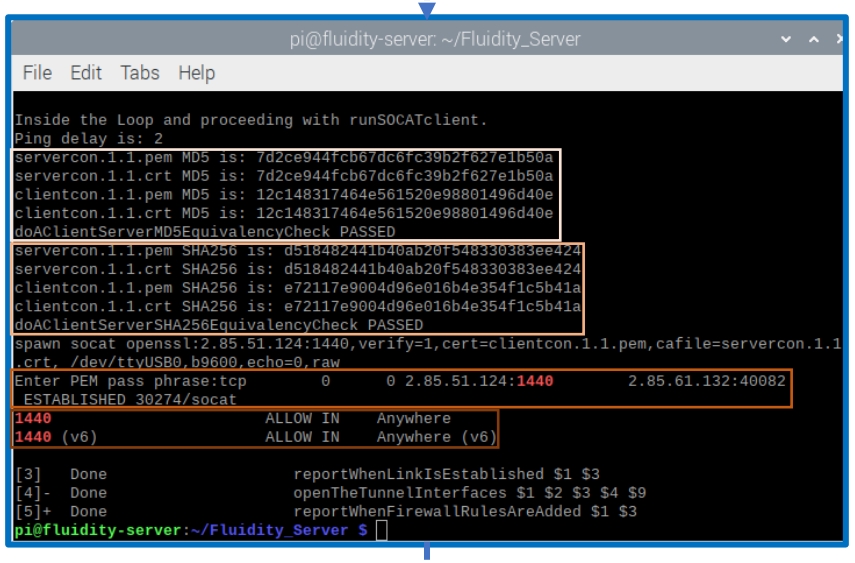
.Fluidity execution

Previous Slide



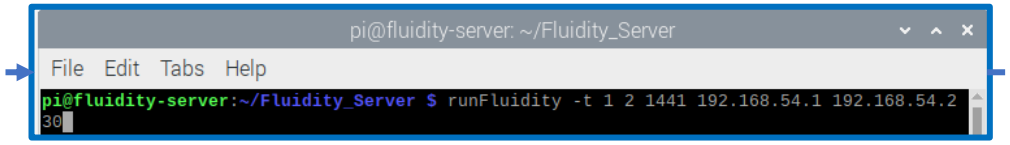
11. runFluidity –s

Execute .Fluidity and link serial devices through an ethernet tunnel. This is .Fluidity for client 1 and connection 1, through port 1440, for local serial device ttyUSB0 and client serial device ttyUSB0 for 9600 symbols/sec.



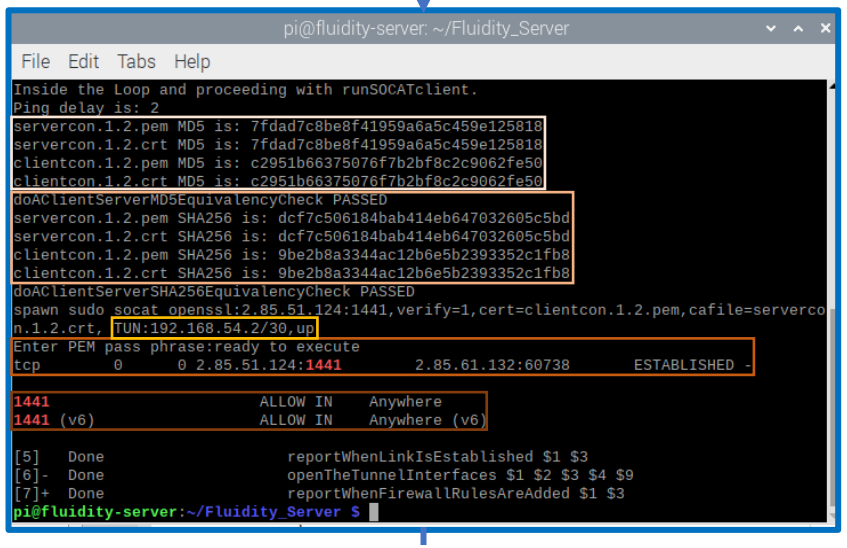
11.1 runFluidity –s

.Fluidity reports that both MD5 and SHA256 SSL certificate hashes match, that for port 1440 a TCP link is established and that a firewall rule was added allowing traffic through port 1440.



12. runFluidity –t

Execute .Fluidity and establish a tunnel link. The tunnel will consist of IP addresses 192.168.54.1 and 192.168.54.2 on server and client, respectively, to securely connect the networks attached on each .Fluidity device.



12.1 runFluidity –t

.Fluidity reports that both MD5 and SHA256 SSL certificate hashes match, that for port 1441 a TCP link is established and that a firewall rule was added allowing traffic through port 1441.

Next Slide

.Fluidity PPPoE Example – Visual Guided Tour

A closer look to the SOCAT generated tunnel, active link firewall statuses and .Fluidity's reporting commands.

Previous Slide

```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 192.168.54.1  netmask 255.255.255.252  destination 192.168.54.1
        inet6 fe80::d597:f3bc:55b2:1065  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500
```

13.1 SERVER SIDE: ifconfig

Once the link is up, tun0 interface is created. Observe tun0 IP address and network mask for the demonstrated ifconfig output.

```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 192.168.54.2  netmask 255.255.255.252  destination 192.168.54.2
        inet6 fe80::85d4:a2f:6b24:70fe  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500
```

13.2 CLIENT SIDE: ifconfig

Respectively, observe the tun0 IP address and network mask for the demonstrated ifconfig output on client's side

pi@fluidity-server: ~

File Edit Tabs Help

```
pi@fluidity-server:~ $ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----

Anywhere on eth0           ALLOW       Anywhere
1440                       ALLOW       Anywhere
1441                       ALLOW       Anywhere
Anywhere on tun0           ALLOW       Anywhere

Anywhere (v6) on eth0      ALLOW       Anywhere (v6)
1440 (v6)                  ALLOW       Anywhere (v6)
1441 (v6)                  ALLOW       Anywhere (v6)
Anywhere (v6) on tun0      ALLOW       Anywhere (v6)

Anywhere                   ALLOW OUT   Anywhere on eth0
Anywhere                   ALLOW OUT   Anywhere on tun0
Anywhere (v6)              ALLOW OUT   Anywhere (v6) on eth0
Anywhere (v6)              ALLOW OUT   Anywhere (v6) on tun0
```

14. SERVER SIDE: ufw status

Observe the rules created allowing traffic through ports 1440 and 1441 for the active .Fluidity connections [1,1] and [1,2].

pi@fluidity-server: ~

File Edit Tabs Help

```
pi@fluidity-server:~ $ showLinkStatus 1 1
Fluidity Connection ID: 1.1
Fluidity Flavour: Serial Link
Serial Link Speed: 9600
Server Listening Port: 1440
Server Serial Interface: ttyUSB0
Server IP Address: 2.85.51.124
Client Serial Interface: ttyUSB0
Client IP Address: 2.85.61.132
Client Username: pi
Netstat Reports: ESTABLISHED
Fluidity Connection Status: ACTIVE
```

15.1 showLinkStatus 1 1

showLinkStatus enables us to see the current .Fluidity link status.

pi@fluidity-server: ~

File Edit Tabs Help

```
pi@fluidity-server:~ $ showLinkStatus 1 2
Fluidity Connection ID: 1.2
Fluidity Flavour: Tunnel Link
Server Listening Port: 1441
Server IP Address: 2.85.51.124
Server Tunnel IP Address: 192.168.54.1
Client IP Address: 2.85.61.132
Client Tunnel IP Address: 192.168.54.2
Network Subnet Mask: 30
Client Username: pi
Netstat Reports: ESTABLISHED
Fluidity Connection Status: ACTIVE
```

15.2 showLinkStatus 1 2

showLinkStatus enables us to see the current .Fluidity link status.

Enjoy your .Fluidity!