AN OPENSSL SOCAT CONNECTION ADMINISTRATION TOOL

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

THE PROBLEM WE ARE TRYING TO SOLVE

1

Secure general-purpose IP communication over any unsecured network.

2

Secure serial communication over any unsecured network for specialized industrial applications.

3

Interconnect serial legacy communications equipment over any modern IP network.

4

Low cost secure IP and serial communication between end-points.

OUR SOLUTION

.Fluidity is a **scalable, low-cost, open-source** solution that utilizes the SSH protocol and openSSL SOCAT connections, to **securely** deliver data over any unsecured network.

- **Scalable:** For a single point-to-point connection you can just use two Raspberry computers. Use more machines as the number of the required end-points increase. Use more powerful equipment as bandwidth requirements increase.
- **Low-cost:** A minimum client-server setup can be implemented with just two Raspberries PIs. The OS can be any Debian-based distro.
- **Open-Source:** Source code written in BASH.
- **Open Architecture:** Use .Fluidity in any DEBIAN based supported system, unhinged from specific hardware restrictions.
- **Customizable:** Feel free to write new code to cover your specialized needs.
- **Securely:** Use the latest encryption technologies to secure your data.

WHO WILL USE IT?

Internet Service Providers

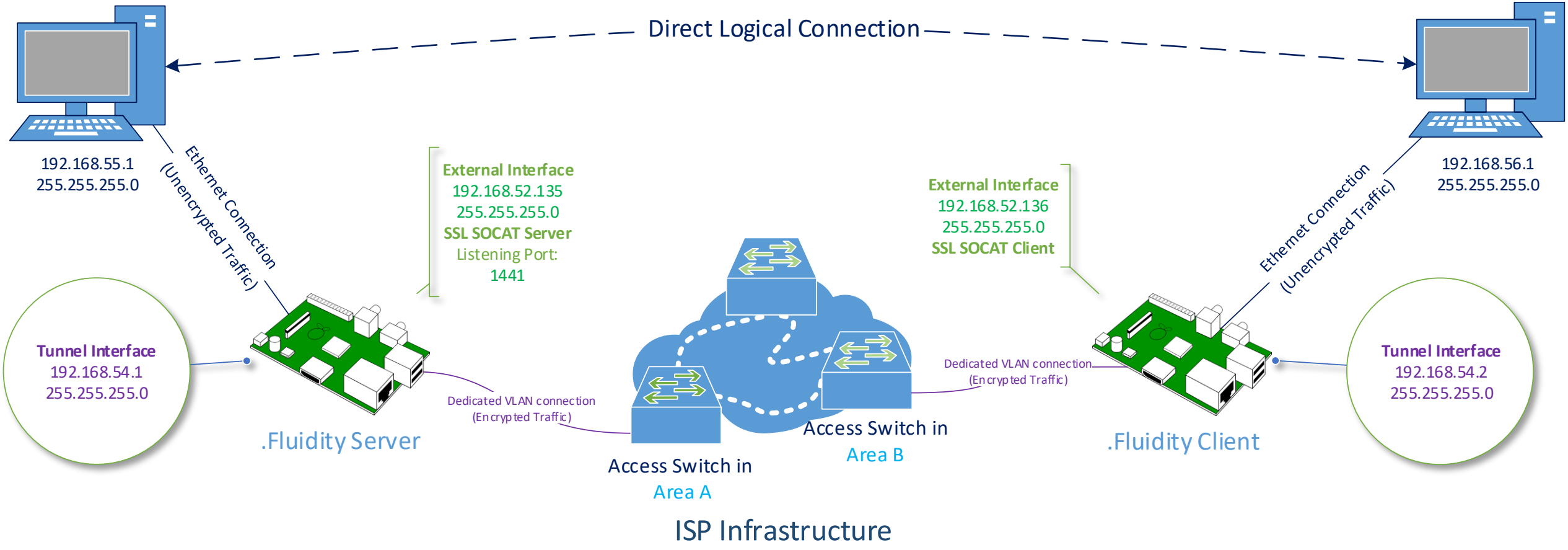
Manufacturing Plants

Government Agencies

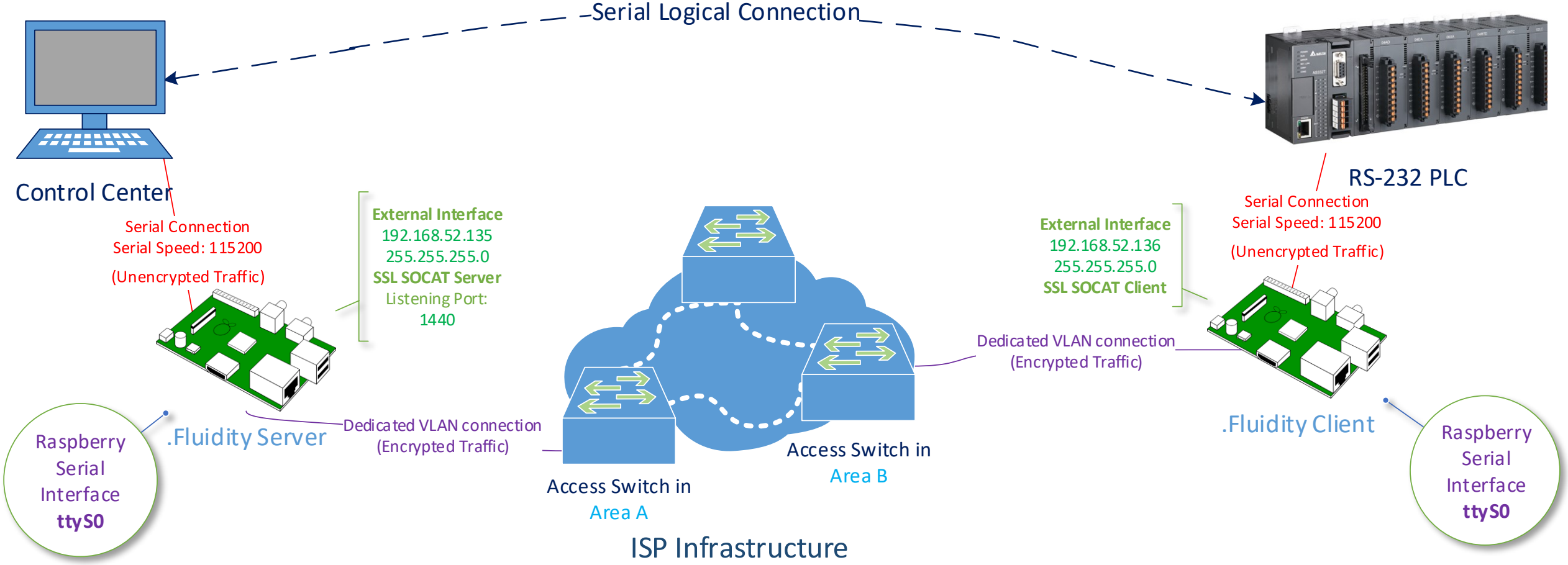
Private Citizens

Any entity that wishes to move data securely over a private or a public network.

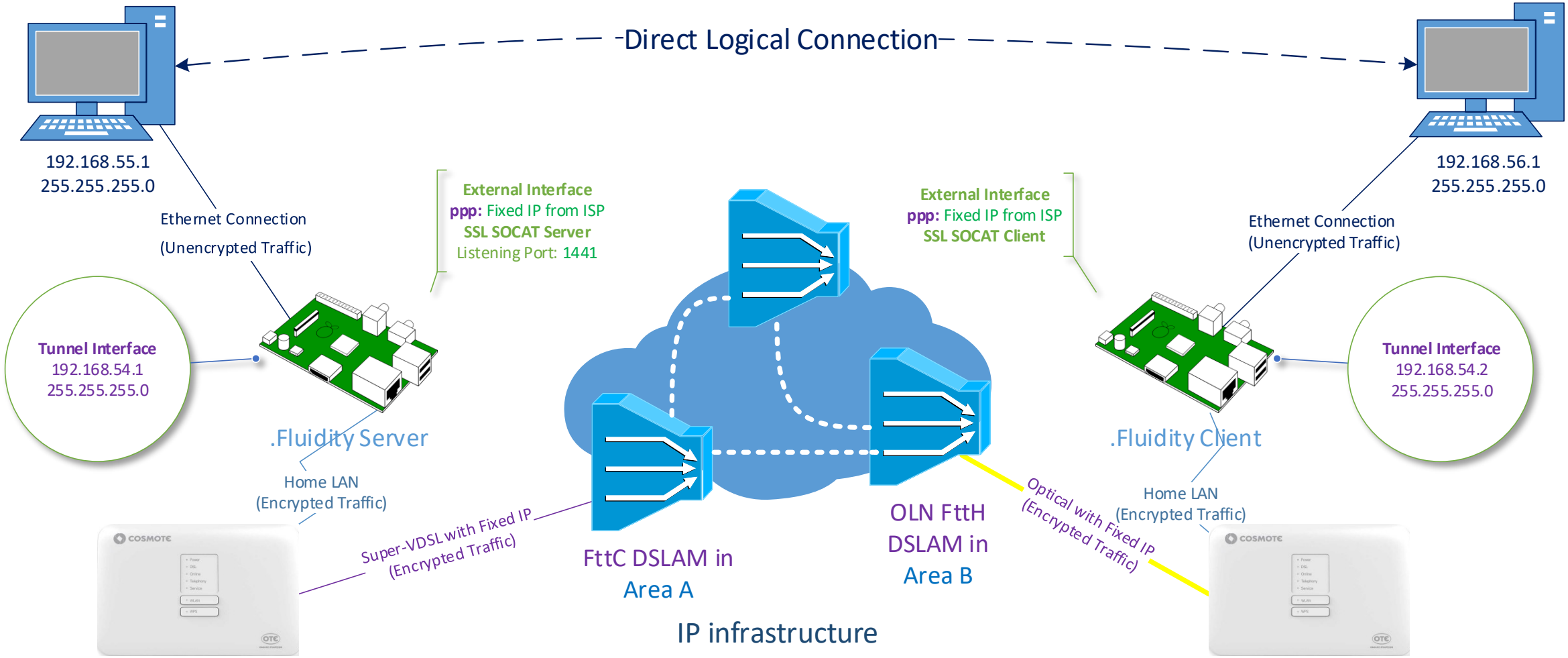
BASIC EXAMPLE – .FLUIDITY IP TUNNELING
COMMUNICATION OVER ISP'S INFRASTRUCTURE (PRIVATE NETWORK, PRIVATE IP)



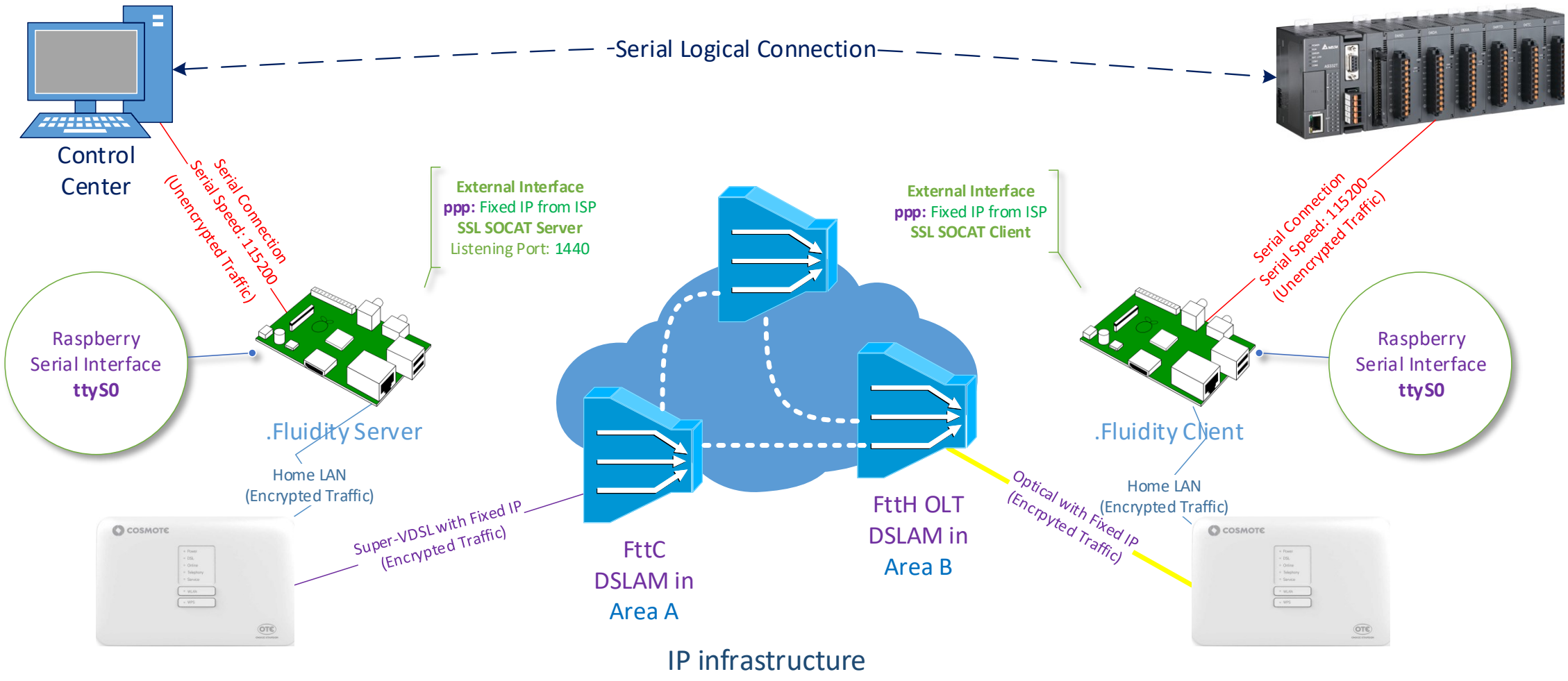
BASIC EXAMPLE – .FLUIDITY SERIAL COMMUNICATION OVER ISP'S INFRASTRUCTURE (PRIVATE NETWORK, PRIVATE IP)



BASIC EXAMPLE – .FLUIDITY IP TUNNELING OVER PPPoE (THROUGH THE INTERNET, PUBLIC IP)

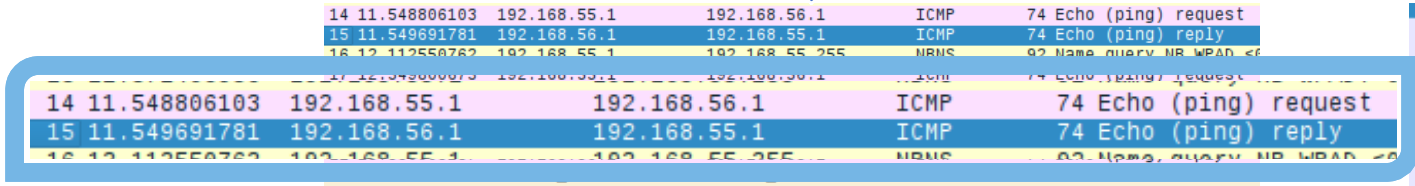


BASIC EXAMPLE – .FLUIDITY SERIAL OVER PPPoE (THROUGH THE INTERNET, PUBLIC IP)

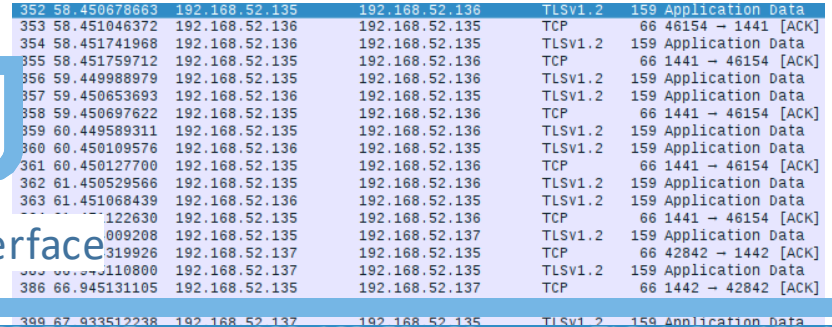


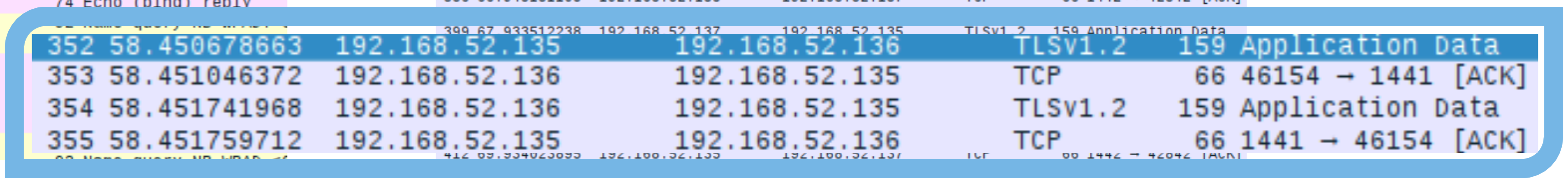
BASIC EXAMPLE – ENCRYPTION ENCAPSULATION - SERVER-SIDE

Ping 192.168.56.1
Ping 192.168.57.1



Internal Interface





FL Server PC_1

.Fluidity Server

BASIC EXAMPLE – ENCRYPTION ENCAPSULATION – CLIENT SIDE

322	46.808989585	192.168.52.135	192.168.52.136	TLSv1.2	159 Application Data
323	46.809092519	192.168.52.136	192.168.52.135	TCP	66 46154 → 1441 [ACK]
324	46.809866531	192.168.52.136	192.168.52.135	TLSv1.2	159 Application Data
325	46.810026692	192.168.52.135	192.168.52.136	TCP	66 1441 → 46154 [ACK]
326	47.807618682	192.168.52.135	192.168.52.136	TLSv1.2	159 Application Data
327	47.808046101	192.168.52.136	192.168.52.135	TLSv1.2	159 Application Data
328	47.808274978	192.168.52.135	192.168.52.136	TCP	66 1441 → 46154 [ACK]
329	48.806582446	192.168.52.135	192.168.52.136	TLSv1.2	159 Application Data
330	48.806932948	192.168.52.136	192.168.52.135	TLSv1.2	159 Application Data
331	48.807111533	192.168.52.135	192.168.52.136	TCP	66 1441 → 46154 [ACK]
332	49.806884503	192.168.52.135	192.168.52.136	TLSv1.2	159 Application Data
333	49.807205399	192.168.52.136	192.168.52.135	TLSv1.2	159 Application Data
334	49.807438446	192.168.52.135	192.168.52.136	TCP	66 1441 → 46154 [ACK]

322	46.808989585	192.168.52.135	192.168.52.136	TLSv1.2	159 Application Data
323	46.809092519	192.168.52.136	192.168.52.135	TCP	66 46154 → 1441 [ACK]
324	46.809866531	192.168.52.136	192.168.52.135	TLSv1.2	159 Application Data
325	46.810026692	192.168.52.135	192.168.52.136	TCP	66 1441 → 46154 [ACK]

Internal Interface

.Fluidity Client 1

Ping Reply

4	0.000422455	192.168.56.1	192.168.55.1	ICMP	74 Echo (ping) reply
5	0.989596175	192.168.55.1	192.168.56.1	ICMP	74 Echo (ping) request
6	0.989823153	192.168.56.1	192.168.55.1	ICMP	74 Echo (ping) reply
7	1.990001514	192.168.55.1	192.168.56.1	ICMP	74 Echo (ping) request

			192.168.56.1	ICMP	74 Echo (ping) request
			Broadcast	ARP	60 Who has 192.168.56.2
			Vmware_5e:70:a9	ARP	42 192.168.56.254 is at
			192.168.55.1	ICMP	74 Echo (ping) reply
			192.168.56.1	ICMP	74 Echo (ping) request
			192.168.55.1	ICMP	74 Echo (ping) reply
			192.168.56.1	ICMP	74 Echo (ping) request
			192.168.55.1	ICMP	74 Echo (ping) reply
			192.168.56.1	ICMP	74 Echo (ping) request
10	2.989125115	192.168.56.1	192.168.55.1	ICMP	74 Echo (ping) reply
11	5.064772348	Vmware_e1:d2:f5	Vmware_5e:70:a9	ARP	42 Who has 192.168.56.1
12	5.065006883	Vmware_5e:70:a9	Vmware_e1:d2:f5	ARP	60 192.168.56.1 is at 0

.FL 1.2 PC_2

.FLUIDITY SECURITY FEATURES IN SHORT…

Key folders are encrypted with ecryptFS. Its security specs are:

- Encryption method: AES
- Key length: 256bits
- With folder name encryption enabled.

.Fluidity client management and connection session management is achieved through an SSH administrative channel. SSH is utilized with the following security specs:

- Encryption method: RSA
- Key length: 2048 bits.

Data are exchanged between .Fluidity devices with SOCAT SSL connections. The Client – Server SSL public/private certificate pair uses the following features:

- Encryption method: RSA
- Key length: 2048 bits
- Password protected SSL certificates

.FLUIDITY SECURITY FEATURES IN SHORT…

There is a safeguard mechanism that allows SSH and SSL certificate creation only when system entropy exceeds a 1000 bits.

There is also a safeguard mechanism that assigns to each .Fluidity (SOCAT SSL) connection a random port number.

Connection to a .Fluidity client is only allowed with an SSH private key. This feature prevents SSH brute-force attacks.

Key policies in SSH server /etc/ssh/sshd_config are changed, in order to fine-tune .Fluidity client security.

Firewalling is performed with the Uncomplicated Firewall (UFW), an easy to use front-end interface to Linux IP-tables.

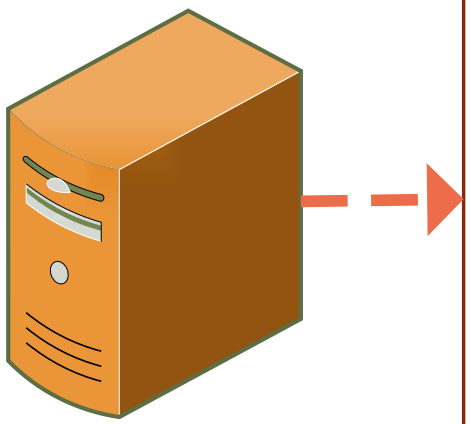
.FLUIDITY SECURITY FEATURES IN SHORT…

A safeguard mechanism exists that performs SSL client-server certificate validation by comparing their generated MD5 hashing values.

A safeguard mechanism exists that performs SSL client-server certificate validation by comparing their generated SHA256 hashing values.

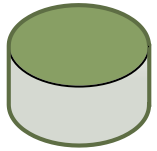
Client data, in the client connection folder, are automatically re-encrypted when the respective .Fluidity connection is detected inactive.

There is an obfuscation mechanism that ensures that the SSL certificate password is encrypted when htop command is executed on a .Fluidity client.

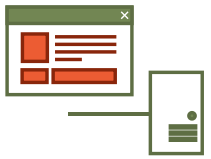


.Fluidity Server

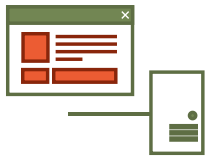
.Fluidity Server: Main systems and sub-systems (not in complete detail):



OS entropy pool
/proc/sys/kernel/random/entropy_avail

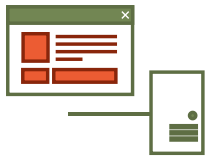


.Fluidity Generated Scripts
(~/Fluidity_Server/
Generated_Scripts)

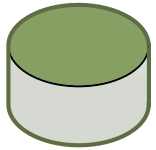


.Fluidity Script
(source Fluidity.sh)

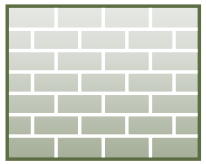
.Fluidity SSL
Certificate Manager



eCryptFs (.Fluidity Folder
Encryption – Decryption)

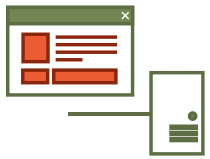


List of Server VPN Routes
listOfServerRoutes.sh

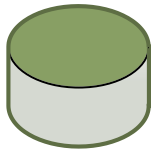


Uncomplicated Firewall (UFW)

.Fluidity Server
File Structure



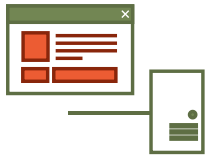
OpenSSL library



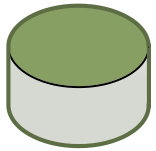
List of Client VPN Routes
for each client
listOfClientRoutes.$1.$2.sh

SSH Server

.Fluidity Clients



SOCAT Multipurpose relay
tool

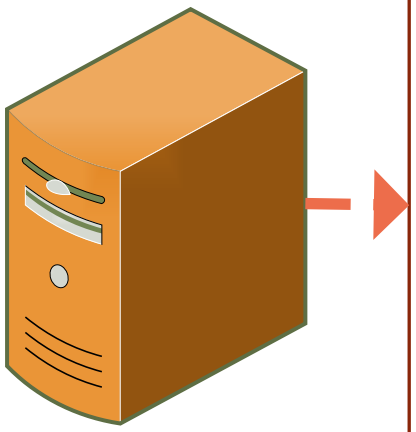


.Fluidity Connections
(Files, folders and
State Variables)

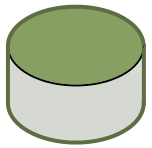
Fluidity

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

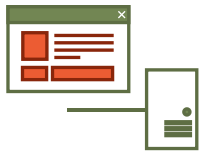
.Fluidity Client: Main systems and sub-systems (not in complete detail):



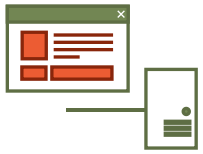
.Fluidity Client



OS entropy pool
/proc/sys/kernel/random/entropy_avail

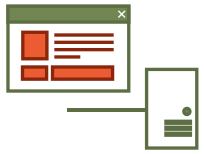


FLdaemon_SeekAndEncrypt.sh

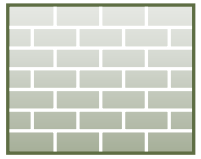


.Fluidity Script
(source Fluidity.sh)

.Fluidity SSL Certificate Manager
Managed by .Fluidity Server

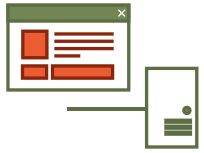


OpenSSL library



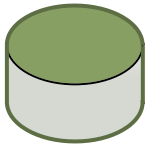
Uncomplicated Firewall (UFW)
Managed by .Fluidity Server

.Fluidity Client
File Structure
Managed by .Fluidity Server

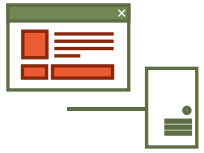


eCryptFs (.Fluidity Folder
Encryption – Decryption)
Managed by .Fluidity Server

SSH Server
Managed by .Fluidity Server



.Fluidity Client
Connection Data

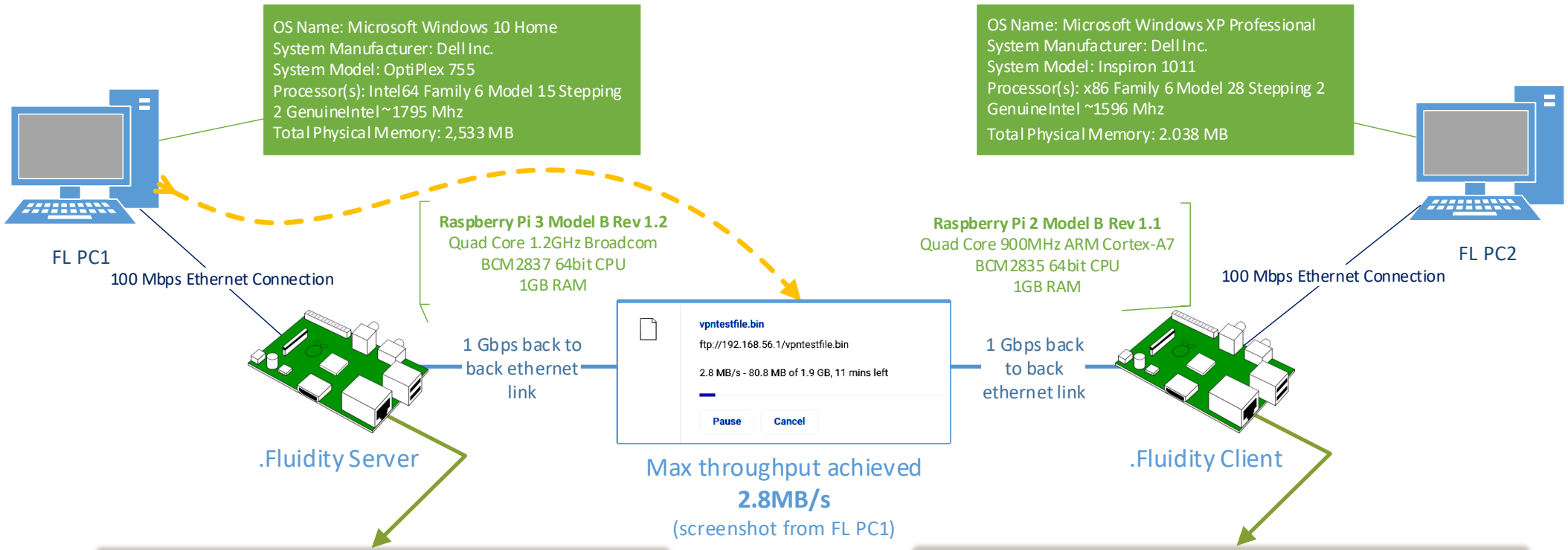


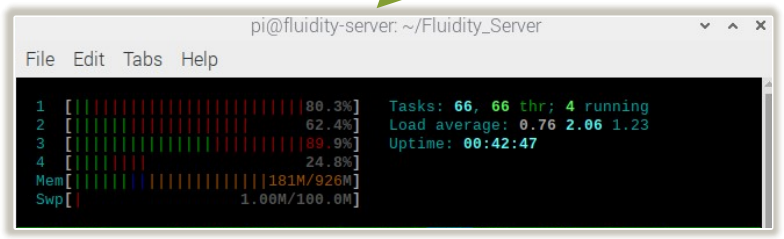
SOCAT Multipurpose relay
tool

Fluidity

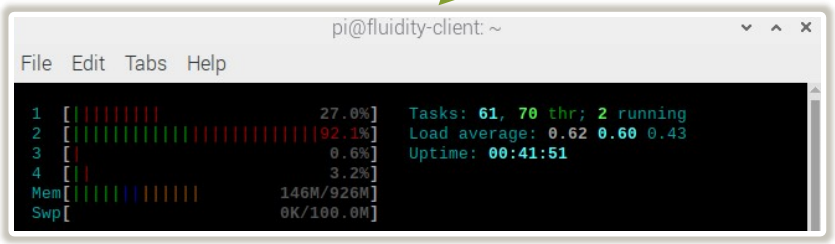
[.Fluidity: *Bring your telecoms infrastructure to the future*]

LAB IMPLEMENTATION– .FLUIDITY MAX THROUGHPUT EXPERIMENT FOR A SINGLE CONNECTION



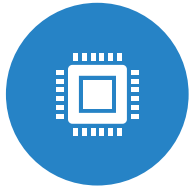


Max CPU utilization on .Fluidity Server



Max CPU utilization on .Fluidity Client

WHY SHOULD PEOPLE USE IT?



Low cost alternative to much more expensive systems (Cisco 800s loaded with crypto licenses).

Open-Source. You know what you get (review the code yourself).

Fully customizable. Add your own code, if you wish.

Uses Debian LINUX Free operating system. Linux source code is open source and gets regularly reviewed by an active community.

Protect your competitive advantage by encrypting your data.

Use cheap public connections to implement a high-end VPN network.

WHO WE ARE

ckapolonaris@ote.gr
LinkedIn

Charalampos Kapolonaris is a Telecoms Electronics Engineer, is currently employed by the OTE Group of Companies (HTO) and holds the position of the Customer Experience Technician.

Charalampos believes in the transformative power that Open Source Software brings to modern society.

Based on the challenges that modern telecommunications organizations are currently facing, Charalampos drew inspiration for this project.

These can be summarized as follows:

- The convergence of the legacy technologies to the new unified telecommunications IP infrastructure.
- The need for innovation in a continuously changing technological environment.

But most of all:

- safeguarding the telecommunications privacy and data security in systems that service critical subsystems of the corporate infrastructure.

vkoutlas@ote.gr

Vasileios Koutlas holds a degree in Electrical Engineering and adheres to the idea of Ethical and White Hat Hacking.

He has broad experience in the use of Linux OS and participated in many projects, mainly related to IP and VoIP networks.

Being highly sensitive in matters concerning data security and data confidentiality, he enthusiastically accepted his participation in this project.

He believes that his involvement in the Open Source Community may result in the development of safer methods for data transfer.

He has been working in OTE Group of Companies (HTO) occupying various positions in the organization, currently occupying the position of CX Technicians Supervisor of Filed Technical Operations in Xanthi Greece.

FIND US ON GITHUB

https://github.com/ckapolonaris/.Fluidity

THANK YOU FOR YOUR ATTENTION