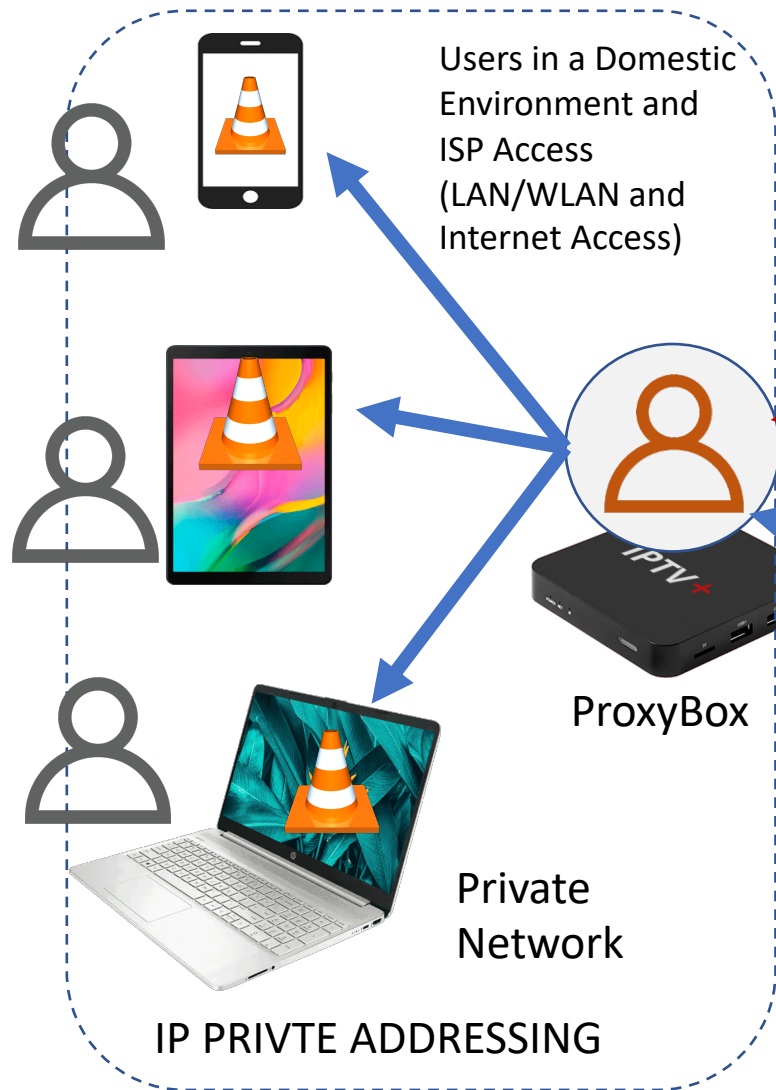


# PA#1

**A secure “pay-per-view” real-time media streaming system**

# System Model



**SAPKDP PROTOCOL**

**SRTSP Protocol**

Internet  
(Public  
Addressing)

## Cloud-Provided Pay-Per-View Streaming as a Service

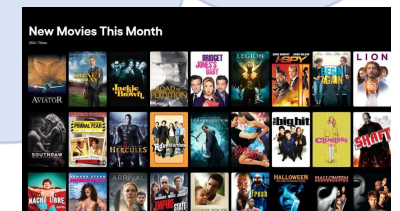
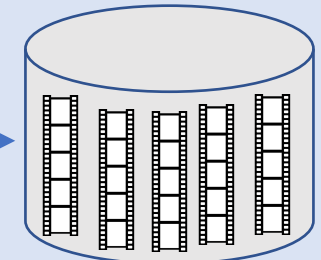


Signaling  
Server

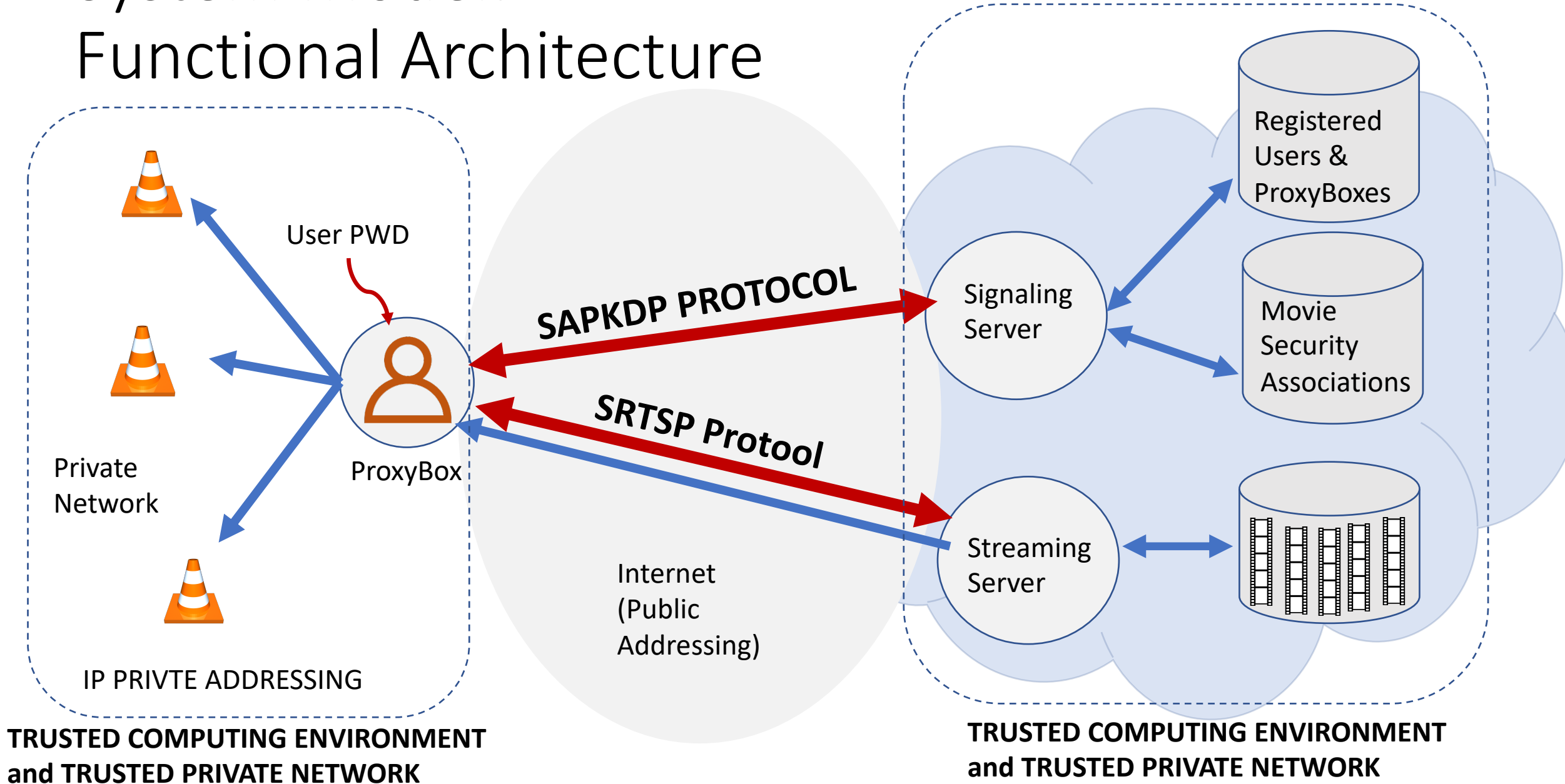
Streaming  
Server

Registered  
Users &  
ProxyBoxes

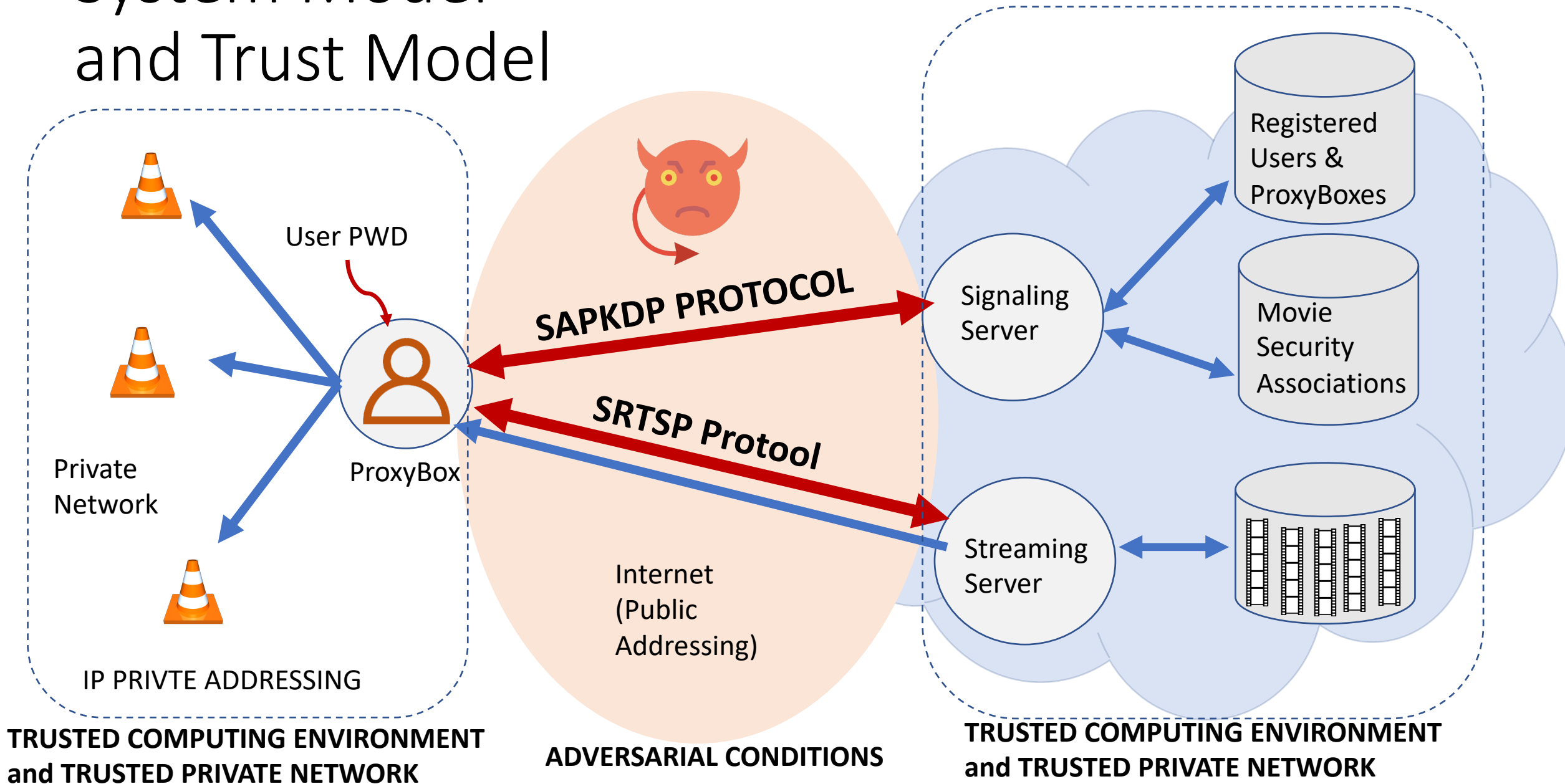
Movie  
Security  
Associations



# System Model: Functional Architecture



# System Model and Trust Model



# Adversary Model

## X509 Framework Attack Types Considered:

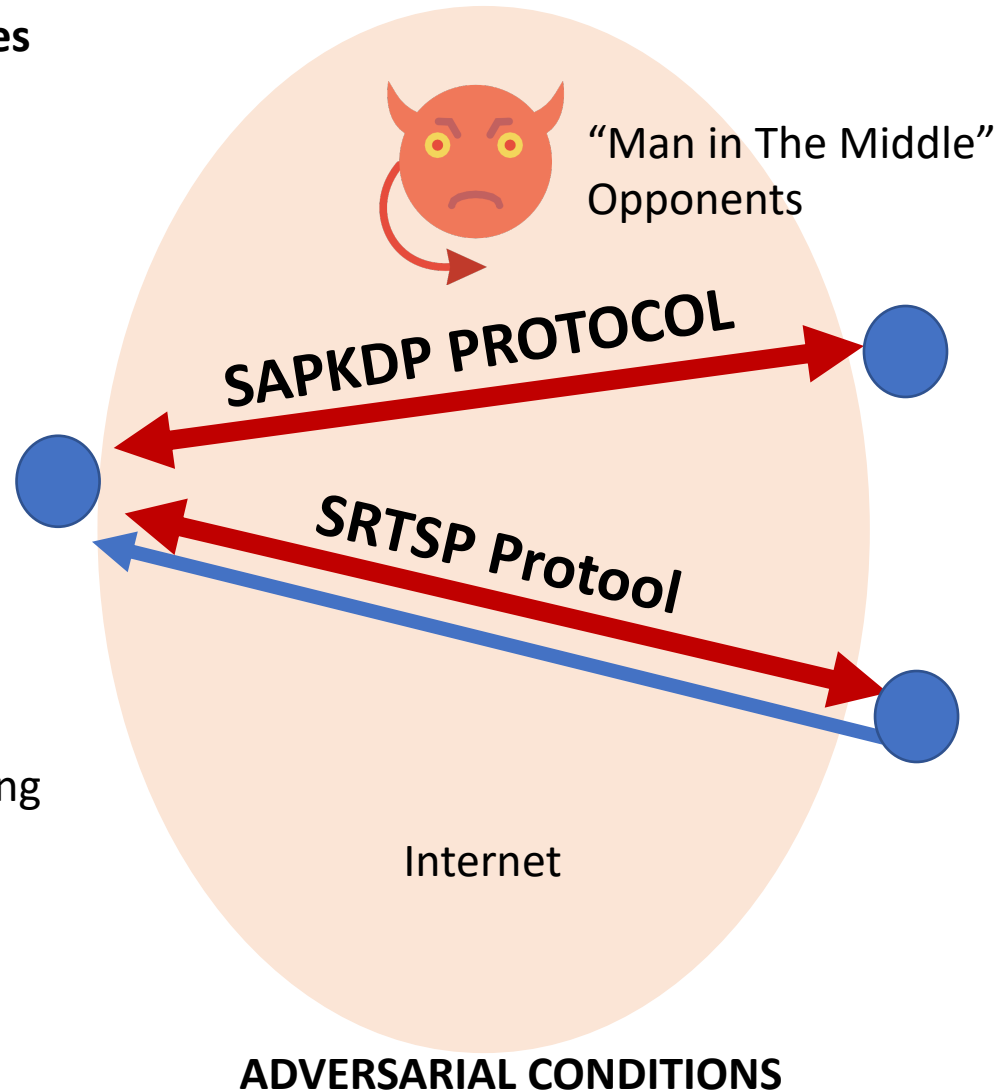
Identity and IP Spoofing or  
Maquerading  
(Peer-Authenticity Breaks)

Data-Leakage  
(Confidentiality Breaks)

Message/DataFlows  
Authenticity Breaks

Message/DataFlows Tampering  
(Integrity Breaks)

Can do Traffic Analysis



## Out of the Adversary Model Scope

DoS, DDoS

Ex:

- Network Congestion and/or Saturation
- Availability and Correctness of Endpoints

# The protocols involved

- **SAPKDP**

- Secure Authentication, Payment and Key-Distribution Protocol**

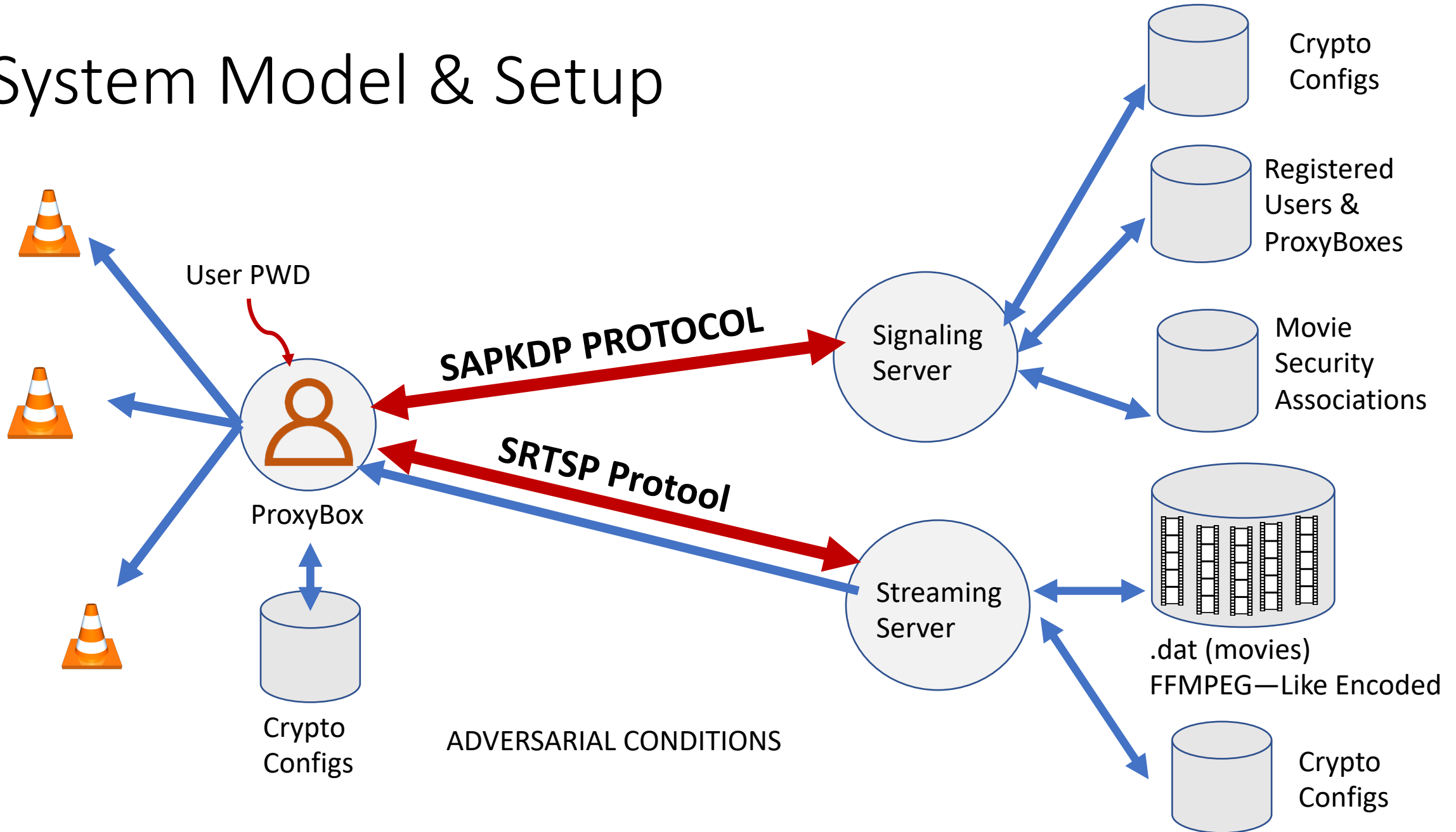
- Can be implemented over UDP, TCP or HTTP (you can choose the encapsulation in your design and implementation)
    - Important: for the PA#1 delivery you will not use TLS or HTTPS – anyway the protocol will be secure by design and implementation
    - Optionally and later on ... you can support it over TLS, DTLS or HTTPS if you want !

- **SRTSP**

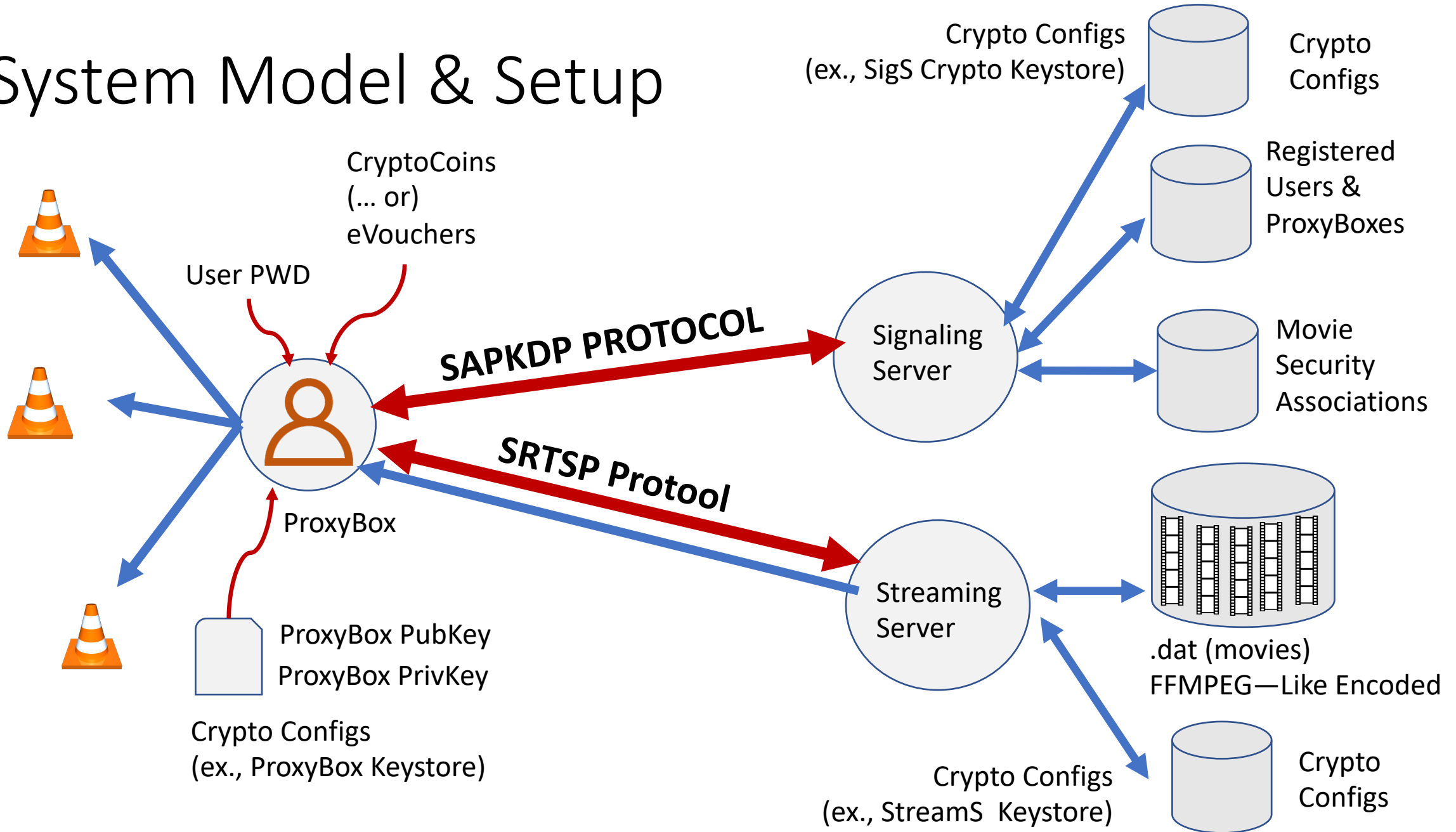
- Secure Real Time Streaming Protocol**

- Must be implemented over UDP !

# System Model & Setup

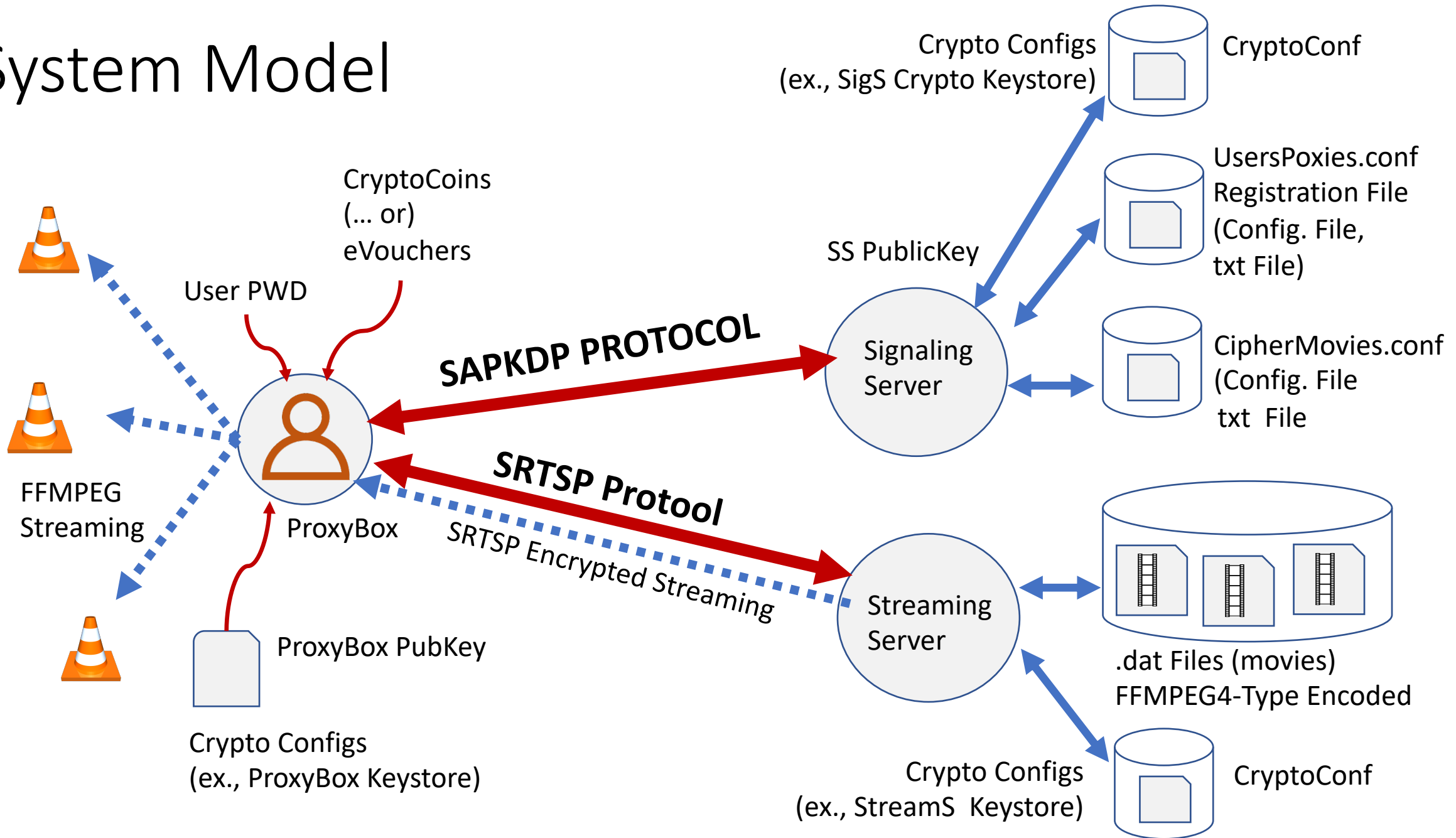


# System Model & Setup





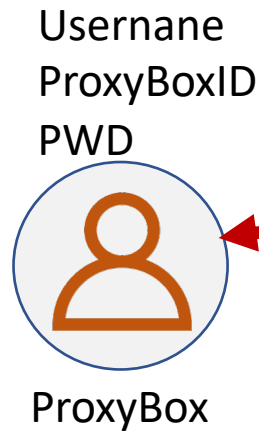
# System Model



# SADKDP Functional Discussion

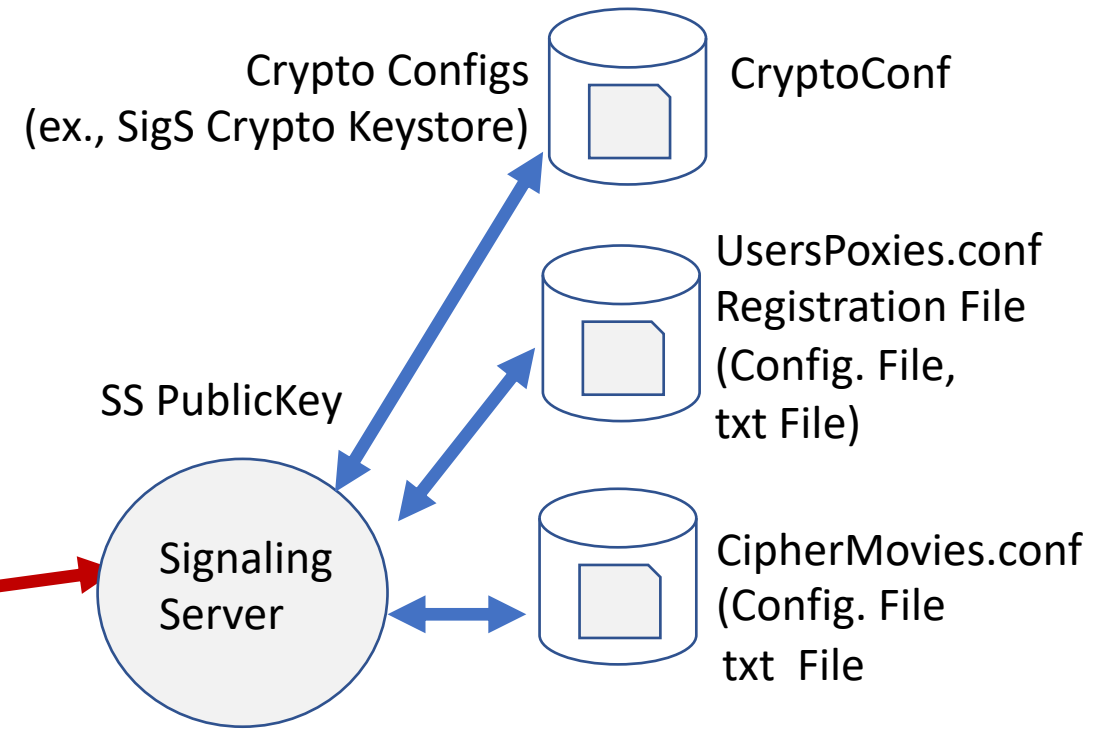
*(Specification formalized in a specific reference doc)*

# System Model: The SAPKDP Protocol



CryptoCoins  
(... or)  
eVouchers

SAPKDP Protocol



# The SAPKDP Protocol



Username  
ProxyBoxID  
PWD  
PrivateKey



ProxyBox



CryptoCoins  
(... or)  
eVouchers

SAPKDP Protocol

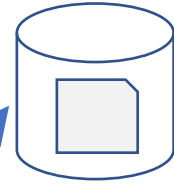
Hi, I am userX,ProxyBoxID

Crypto Configs  
(ex., SigS Crypto Keystore)

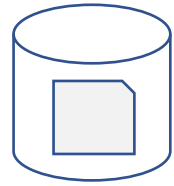


CryptoConf

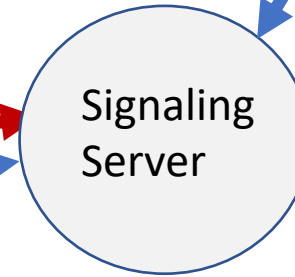
SS PublicKey



UsersPoxies.conf  
Registration File  
(Config. File,  
txt File)



CipherMovies.conf  
(Config. File  
txt File)



Signaling  
Server

# The SAPKDP Protocol



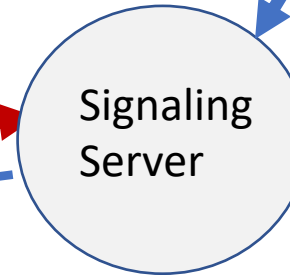
Username  
ProxyBoxID  
PWD  
PrivateKey



ProxyBox

SAPKDP Protocol

SS PublicKey



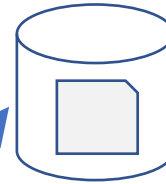
Signaling  
Server

Crypto Configs  
(ex., SigS Crypto Keystore)

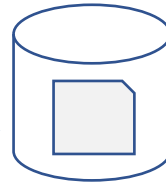


CryptoConf

UsersPoxies.conf  
Registration File  
(Config. File,  
txt File)



CipherMovies.conf  
(Config. File  
txt File)



OK, here you have a NONCE challenge  
a SALT and a Counter for you PBE Proof

CryptoCoins  
(... or)  
eVouchers



# The SAPKDP Protocol



Username  
ProxyBoxID  
PWD  
PrivateKey



ProxyBox



CryptoCoins  
(... or)  
eVouchers

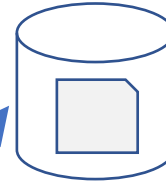
SAPKDP Protocol

Crypto Configs  
(ex., SigS Crypto Keystore)

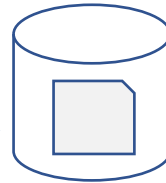


CryptoConf

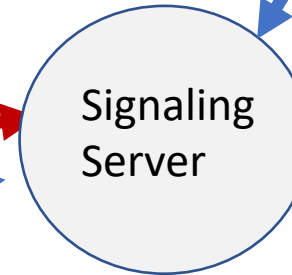
SS PublicKey



UsersPoxies.conf  
Registration File  
(Config. File,  
txt File)



CipherMovies.conf  
(Config. File  
txt File)



Here you have my PBE Auth Proof  
I want to see the Movie "CARS", can I ?

# The SAPKDP Protocol



Username  
ProxyBoxID  
PWD  
PrivateKey



ProxyBox



CryptoCoins  
(... or)  
eVouchers

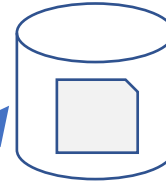
SAPKDP Protocol

Crypto Configs  
(ex., SigS Crypto Keystore)

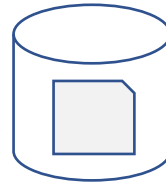


CryptoConf

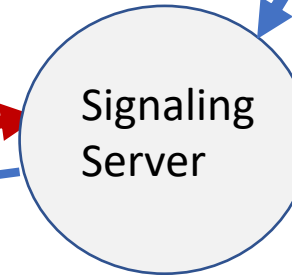
SS PublicKey



UsersPoxies.conf  
Registration File  
(Config. File,  
txt File)



CipherMovies.conf  
(Config. File  
txt File)



Yes you can ... must pay 1 cryptocoin  
Here you have another NONCE  
Send the valid payment and sign your  
pay-per-view order (with your valid  
digital signature)

# The SAPKDP Protocol



Username  
ProxyBoxID  
PWD  
PrivateKey



ProxyBox



CryptoCoins  
(... or)  
eVouchers

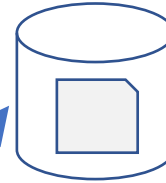
SAPKDP Protocol

Crypto Configs  
(ex., SigS Crypto Keystore)

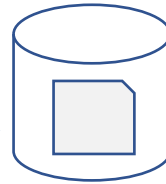


CryptoConf

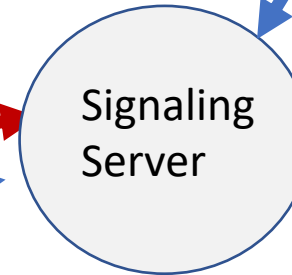
SS PublicKey



UsersPoxies.conf  
Registration File  
(Config. File,  
txt File)



CipherMovies.conf  
(Config. File  
txt File)



This is my signed transaction of  
1 crypticoins for the payment  
... You can validate the payment  
Is correct and valid



# The SAPKDP Protocol



Username  
ProxyBoxID  
PWD  
PrivateKey



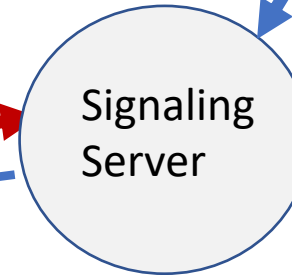
ProxyBox



CryptoCoins  
(... or)  
eVouchers

SAPKDP Protocol

SS PublicKey

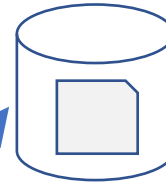


Crypto Configs  
(ex., SigS Crypto Keystore)

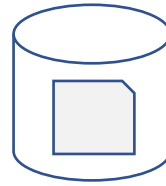


CryptoConf

UsersPoxies.conf  
Registration File  
(Config. File,  
txt File)



CipherMovies.conf  
(Config. File  
txt File)



OK, the payment is verified and it  
is correct

I am sending all the info your need for the  
movie you want, protected and just for you and  
signed by me :

- ENDPOINT (IP & Port)
- ciphersuite conf
- cryptographic materials ad keys
- Opaque Info (encrypted ticket> you must send  
to the stream server

# Use model: The SRTSP Protocol



Username  
ProxyBoxID  
PWD



ProxyBox

CryptoCoins  
(... or)  
eVouchers

*SRTSP Protocol*

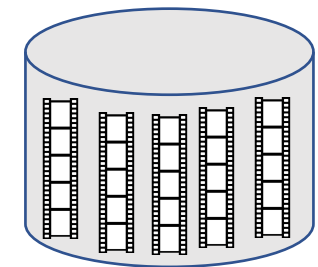
Signaling  
Server

Streaming  
Server

Registered  
Users &  
ProxyBoxes

Movie  
Security  
Associations

Configuratuion Files



.dat Files (movies)

# SRTSP Functional Discussion

*(Specification formalized in a specific reference doc)*

# The SRTSP Protocol



Username  
ProxyBoxID  
PWD



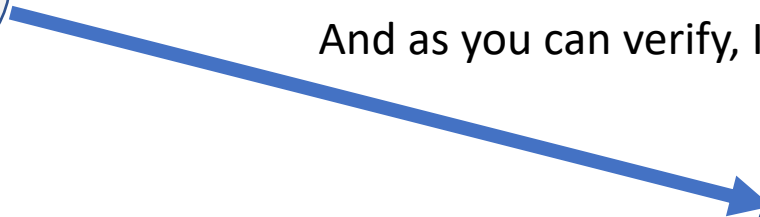
ProxyBox

Hi Streaming Server ...  
I want to see the movie CARS

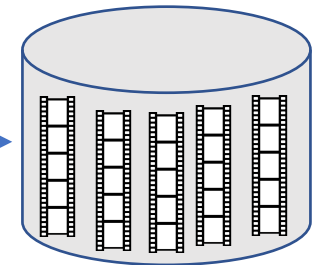
You have here an Opaque Ticket for  
You (I obtained from the Signaling Server)

It was delivered just for me by the Signalling Server,  
to forward it for you

And as you can verify, I am sending this signed by me



Streaming  
Server



.dat Files (movies)

# The SRTSP Protocol



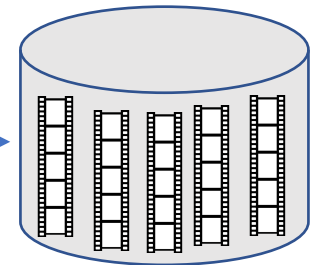
Username  
ProxyBoxID  
PWD



ProxyBox

Ok, From my verification, it is fine  
... Here is the confirmation that everything is ok  
Are you ready to receive ?  
See that this is signed by me  
If it is ok, send me an ACK to this "nonce challenge"

Streaming  
Server



.dat Files (movies)

# The SRTSP Protocol



Username  
ProxyBoxID  
PWD

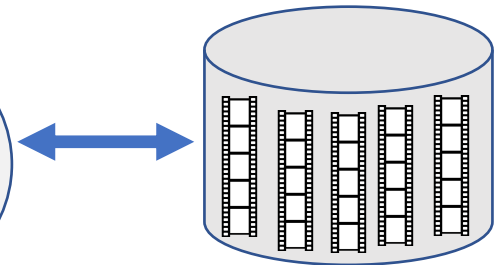


ProxyBox

Yep, I recognize your signature ...  
I send the answer to your challenge ...  
So Yep we are now eager, ready, with our  
“popcorns” ready to start playing!

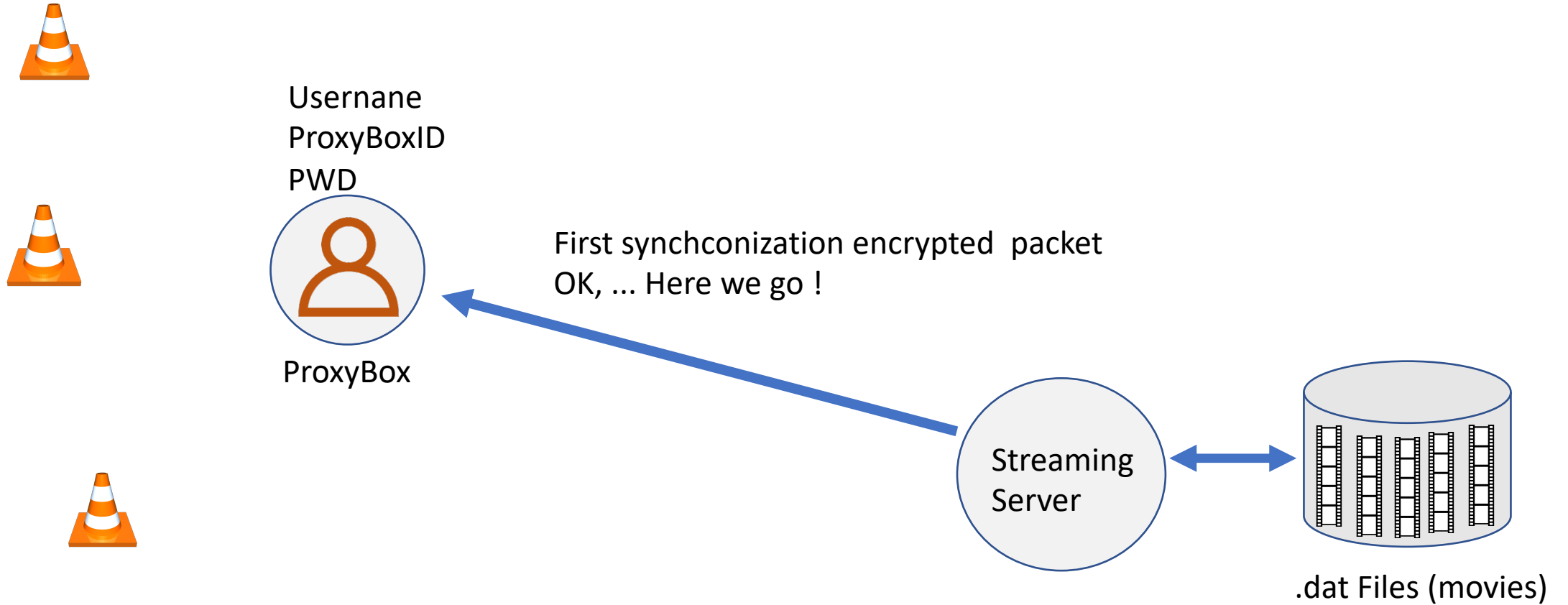


Streaming  
Server

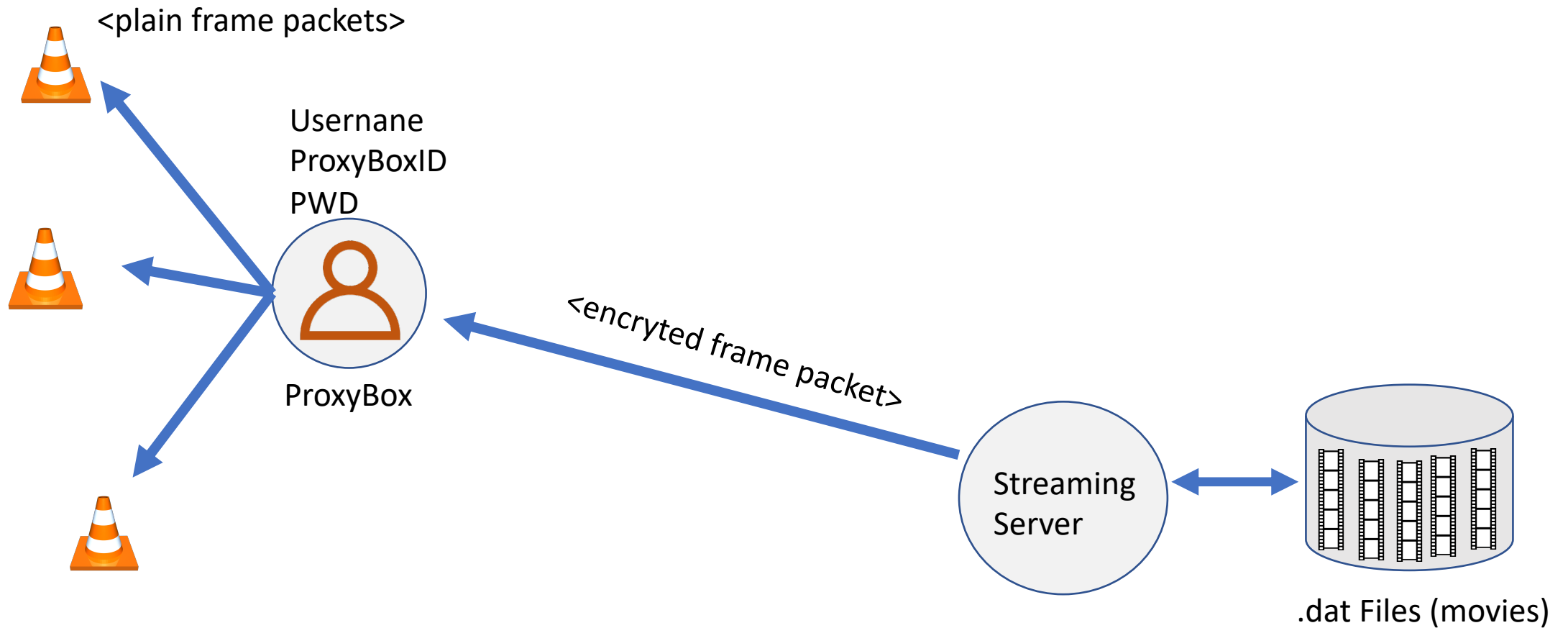


.dat Files (movies)

# The SRTSP Protocol



# The SRTSP Protocol and Real-Time Playing





# Implementation

PA#1 developed in 3 Stages:

Step 1 (or STAGE 1)

Initial Approach to a “simplified” version of SRTSP Protocol

**No Signaling Server, No SAPKDP Protocol, no Coins and no PayPerView**

Step 2 (or subsequent STAGE 2)

Signaling Server

SAPKDP

SRTSP

(SAPKDP and SRTSP Complete Integration and everything else)

# Ref. Evaluation

12/20 Points

~8-12 / OCT

(Prelim.

Demos on

LABs

15-17 / OCT)

20/20 Points

Delivery + Quiz:

Ref: 24/NOV

“Hands On”: Here you go: Stage 1 !  
(Simple SRTSP Version and simplified vision of  
the desired system)

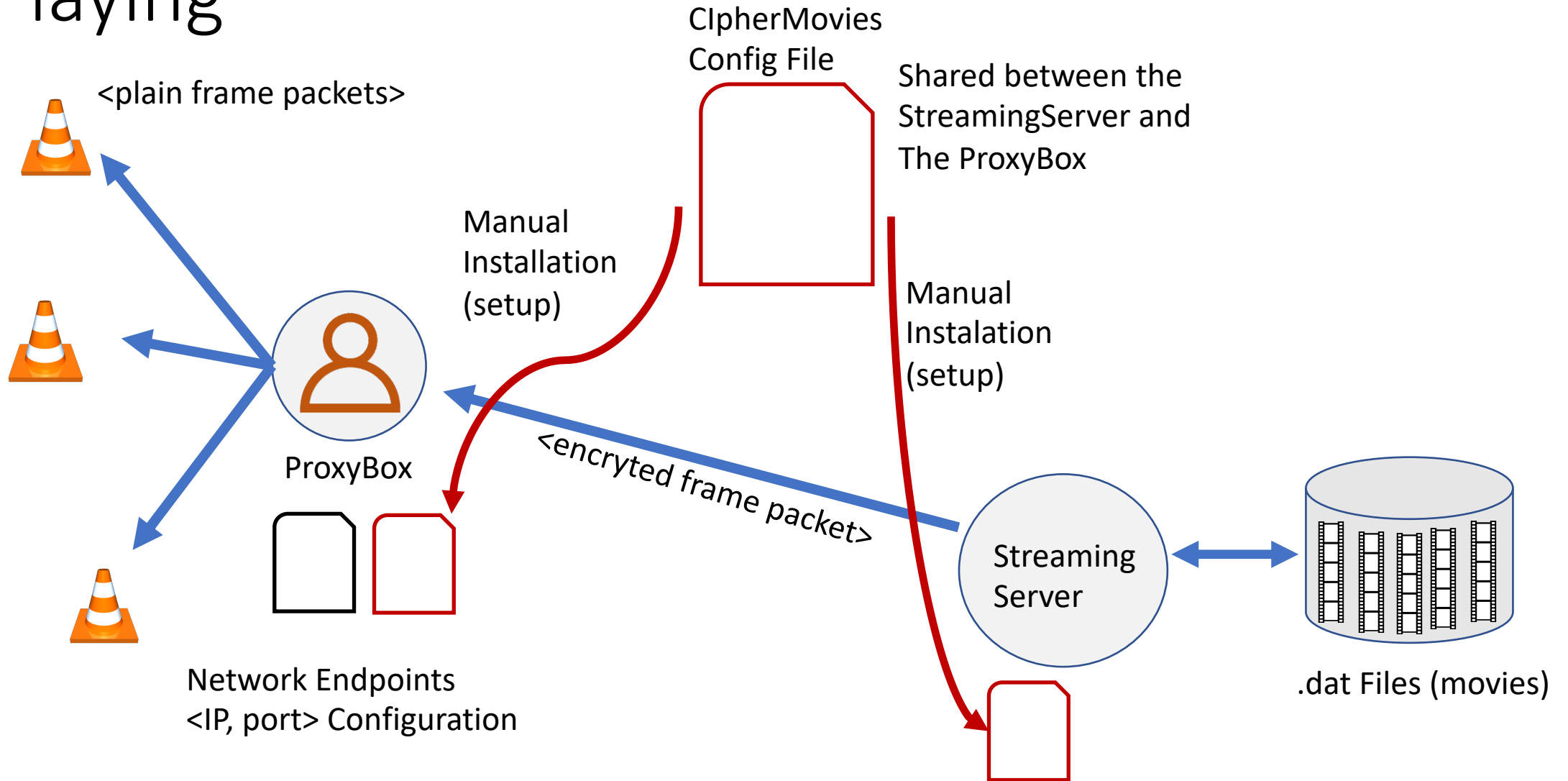
# STEP 1 Implementation

# Your first step challenge (suggestion)

- Start with the SRTSP Protocol
- A direct “step” forward to the provided material (Lab 3, Streaming and Materials)
  - You have Movies (.dat) and the initial (unsecure) StreamingServer and Proxy implementation
  - Ready to be used (unsecurely)!
- You can develop the ProxyBox and StreamingServer extending the provided implementations
  - You have the base materials and of course the VLC tool
- Can start by using “static” configurations
  - Configuration files manually installed in ProxyBOX and Streaming Server
- The SRTSP for the streaming phase only requires Symmetric Crypto and MACs
  - BUT the USABLE CRYPTOGRAPHY MUST BE CONFIGURABLE !!!!

Later on, the static configurations must be removed after the complete implementation of the SRTSP and SAPKDP protocols, as well as the Signaling Server

# The “simplified” SRTSP Protocol for Real-Time Playing



# Initial static configuration for the SRTSP Protocol and its implementation in the Streaming Server and in the ProxyBox

- Can choose any valid “static” configurations
  - CRYPTOGRAPHICALLY CONFIGURABLE !!!!

Later on (Next Steps), this file will be not shared between the Streaming Serve and the ProxyBox

The setup will be dynamic after the complete implementation of the SRTSP and SAPKDP protocols, as well as the Signaling Server

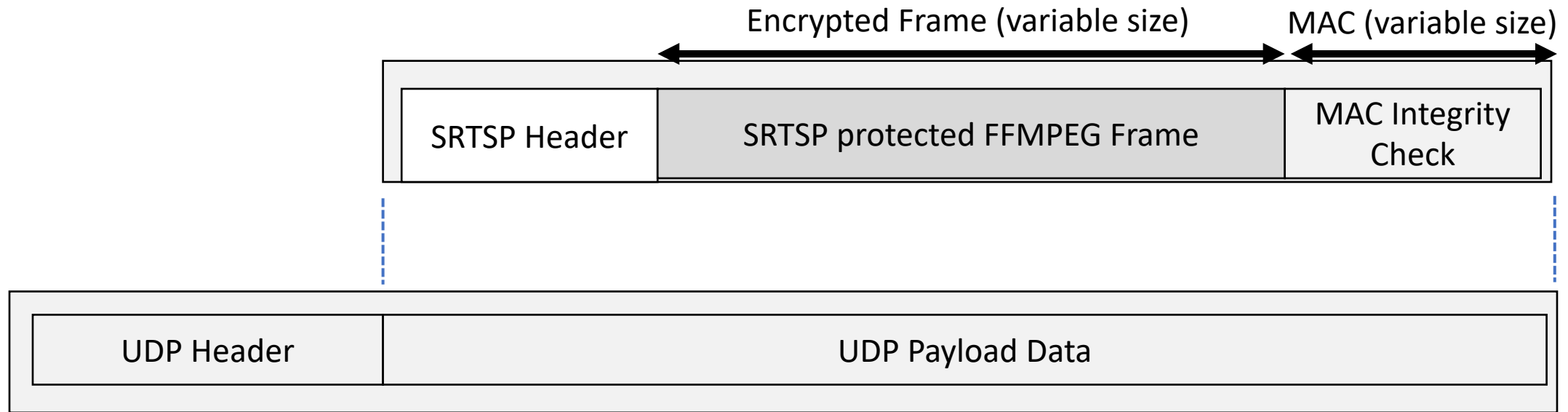
# Simplified version of SRTSP and SRTSP encapsulation over UDP and Datagram Packets

SRTSP Header contains:

4 bits: contains the version id of the Simplified SRTSP protocol: 0001

4 bits: contains an indication that the setup is manual: 0000 means manual configuration of endpoints

16 bits: integer, contains the size in bytes of the encapsulated encrypted frame



# For this week ! (1)

- Try to address the first proposed challenge (STEP 1) for the SRTSP
- You only need to start from the materials you already have

StreamingServer

Proxy

- Try to avoid complexity in the base StreamingServer and Proxy
  - You must implement with a modular approach !
  - Extend the DatagramSocket or Datagram Packet Classes
  - Ex: MySRTSPDatagramSocket ..... MySRTSPDatagramPacket
  - The minimum number of lines changed in the StreamingServer and Proxy Class, the better and more modular will be your first solution
  - VERY IMPORTANT FOR THE NEXT PHASE !



# For this week (2)

Try to avoid complexity in the base StreamingServer and Proxy

- You must implement with a modular approach !
- Extend the DatagramSocket or Datagram Packet Classes
- Ex: MySRTSPDatagramSocket ..... MySRTSPDatagramPacket
- The minimum number of lines changed in the StreamingServer and Proxy Class, the better and more modular will be your first solution
- VERY IMPORTANT FOR THE NEXT PHASE !

How Many LoC do you have in your Secure Proxy and Secure Stream Server implementation ? +/- 10% diff max. comparing w/ the original code ?

Is your configuration neutral ? Can we use any cupsuete configuration ?