PA#2

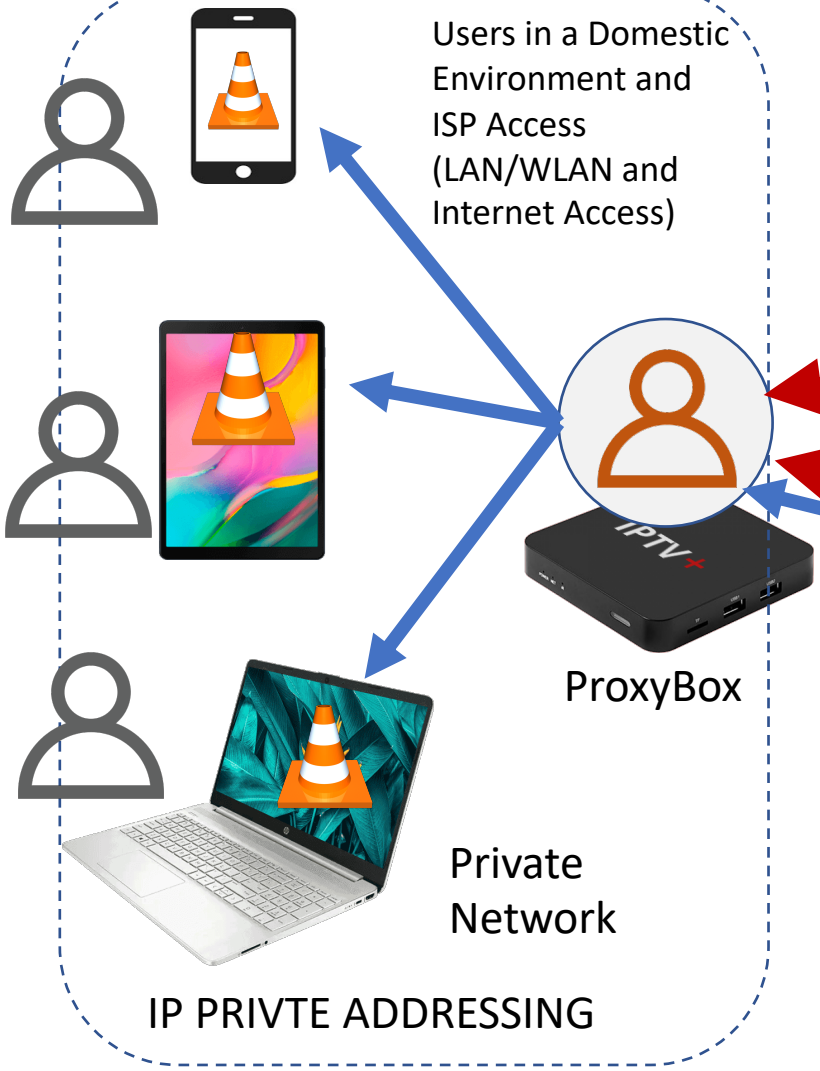
Overlayed secure “pay-per-view” real-time media streaming system supported by configurable TLS Channels

Initial framework for PA#2

- Designed and implemented as an extension form the PA#1 implementation
- Each student can use the base PA#1 implementation to select different PA#2 implementation options
  - Depending on the PA#1 Implementation State (as delivered)
  - Or extending the PA#1 implementation delivered, to address the selected PA#2 option
- Overlaying solutions for SAPKDP, or SRTSP or BOTH:
  - Leveraged from the JSSE Programing support:
    - JSSE Sockets for TLS / TCP Support
    - JSSE Sockets for DTLS / UDP Support
  - Configurable TLS or DTLS Operation behavior in the respective endpoints

System Model (Rev. PA#1)

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



SAPKDP PROTOCOL

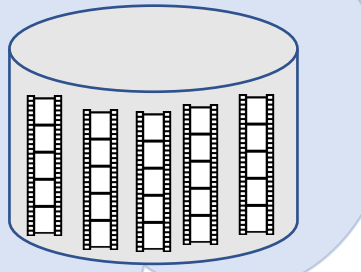
SRTSP Protool

Signaling Server

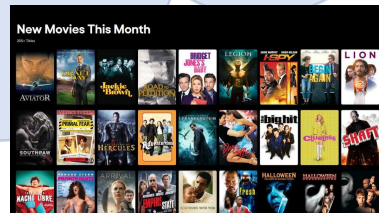
Registered Users & ProxyBoxes

Movie Security Associations

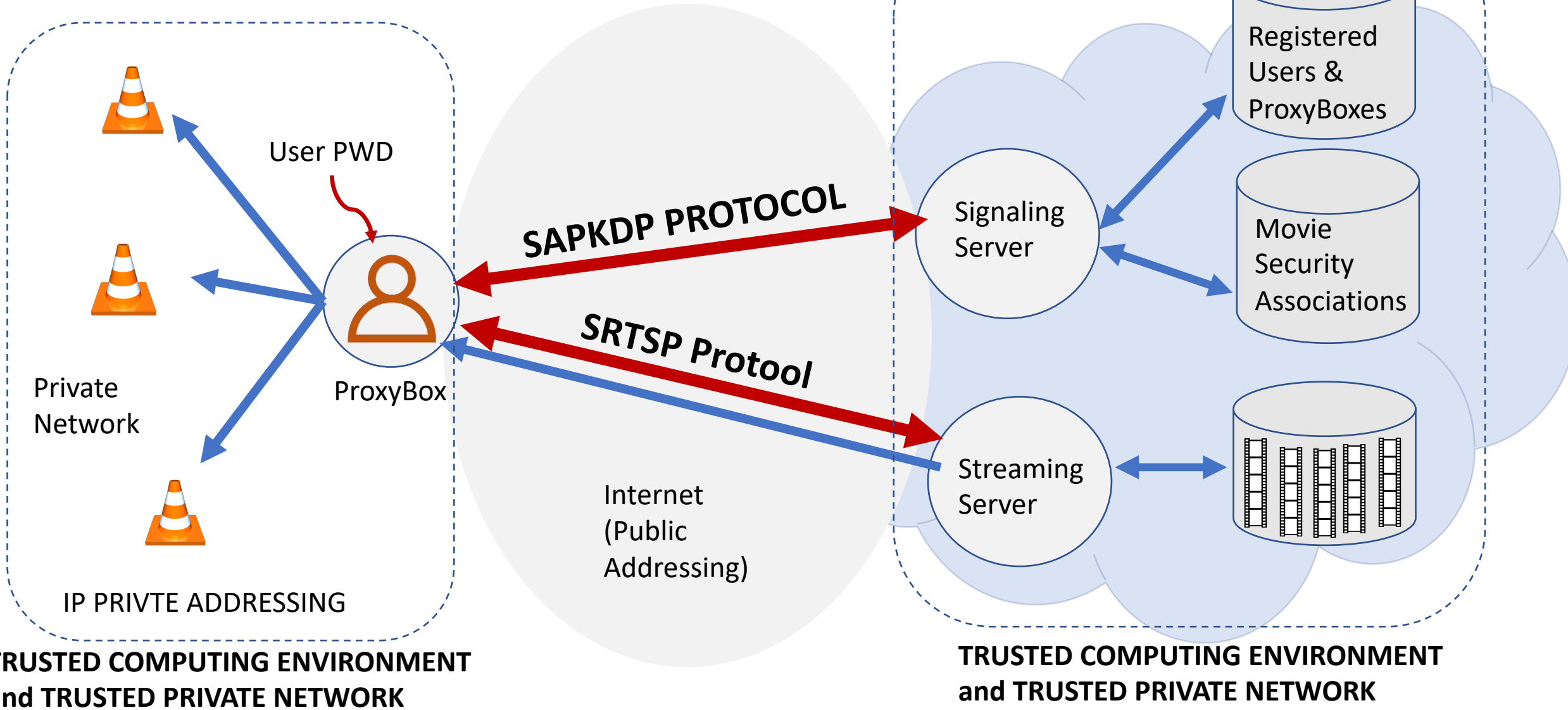
Streaming Server



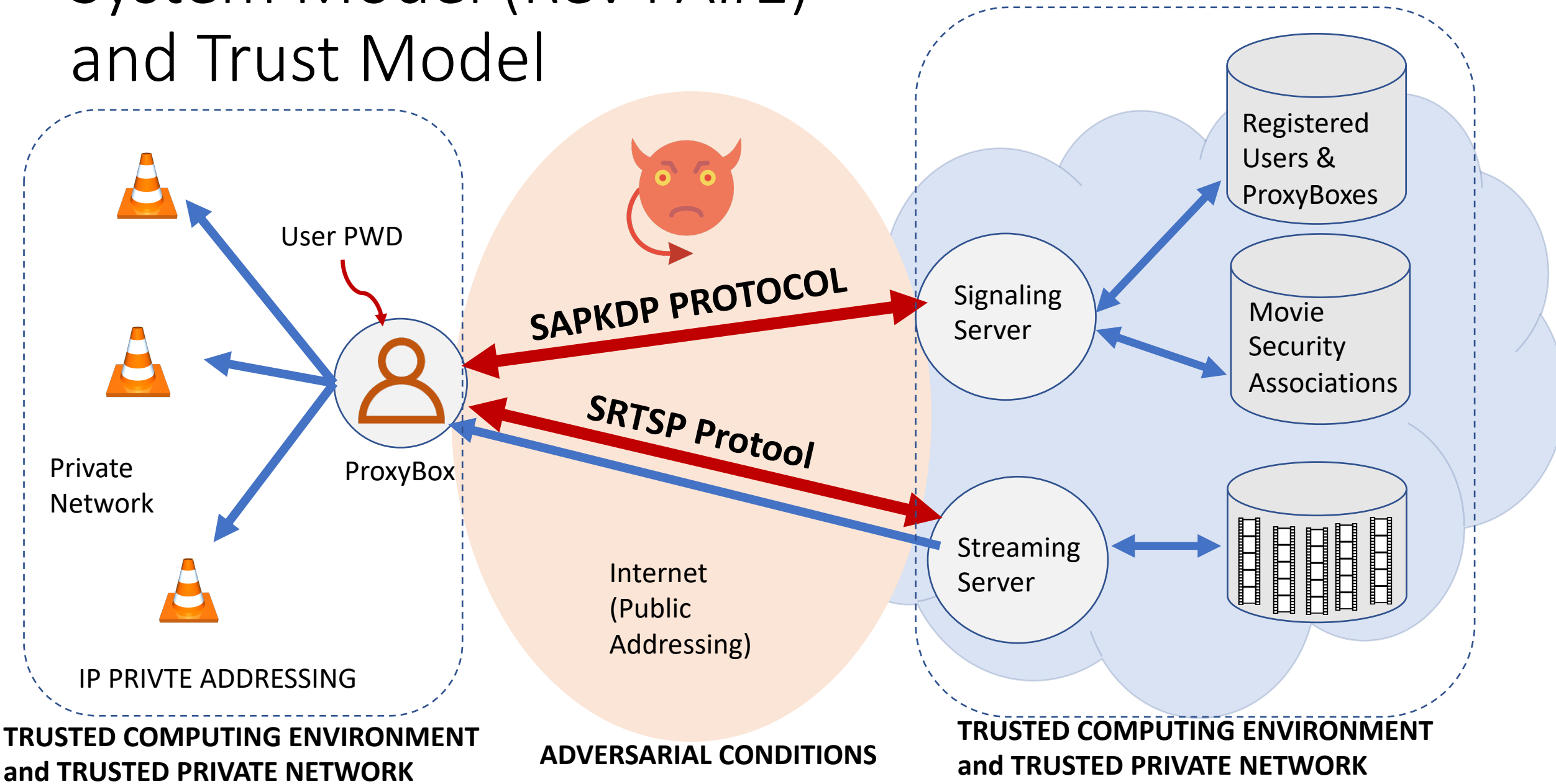
Internet (Public Addressing)



System Model (Rev. PA#1) Functional Architecture



System Model (Rev PA#1) and Trust Model



Adversary Model (Rev PA#1)

Communication Endpoints

X509 Framework Attack Types Considered:

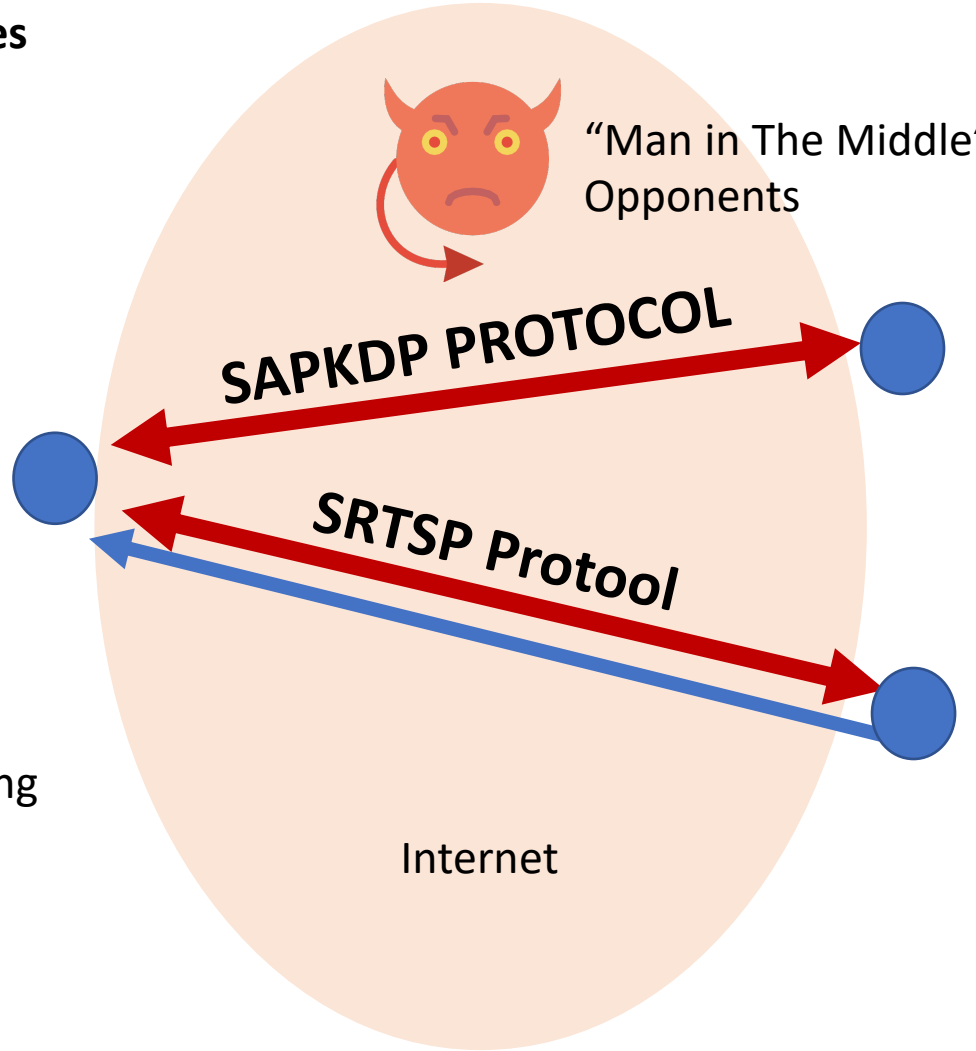
Identity and IP Spoofing or Maquerading
(Peer-Authenticity Breaks)

Data-Leakage
(Confidentiality Breaks)

Message/DataFlows Authenticity Breaks

Message/DataFlows Tampering
(Integrity Breaks)

Can do Traffic Analysis



ADVERSARIAL CONDITIONS

Out of the Adversary Model Scope

DoS, DDoS

Ex:

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

The protocols involved in PA#1

- **SAPKDP**

  Secure Authentication, Payment and Key-Distribution Protocol

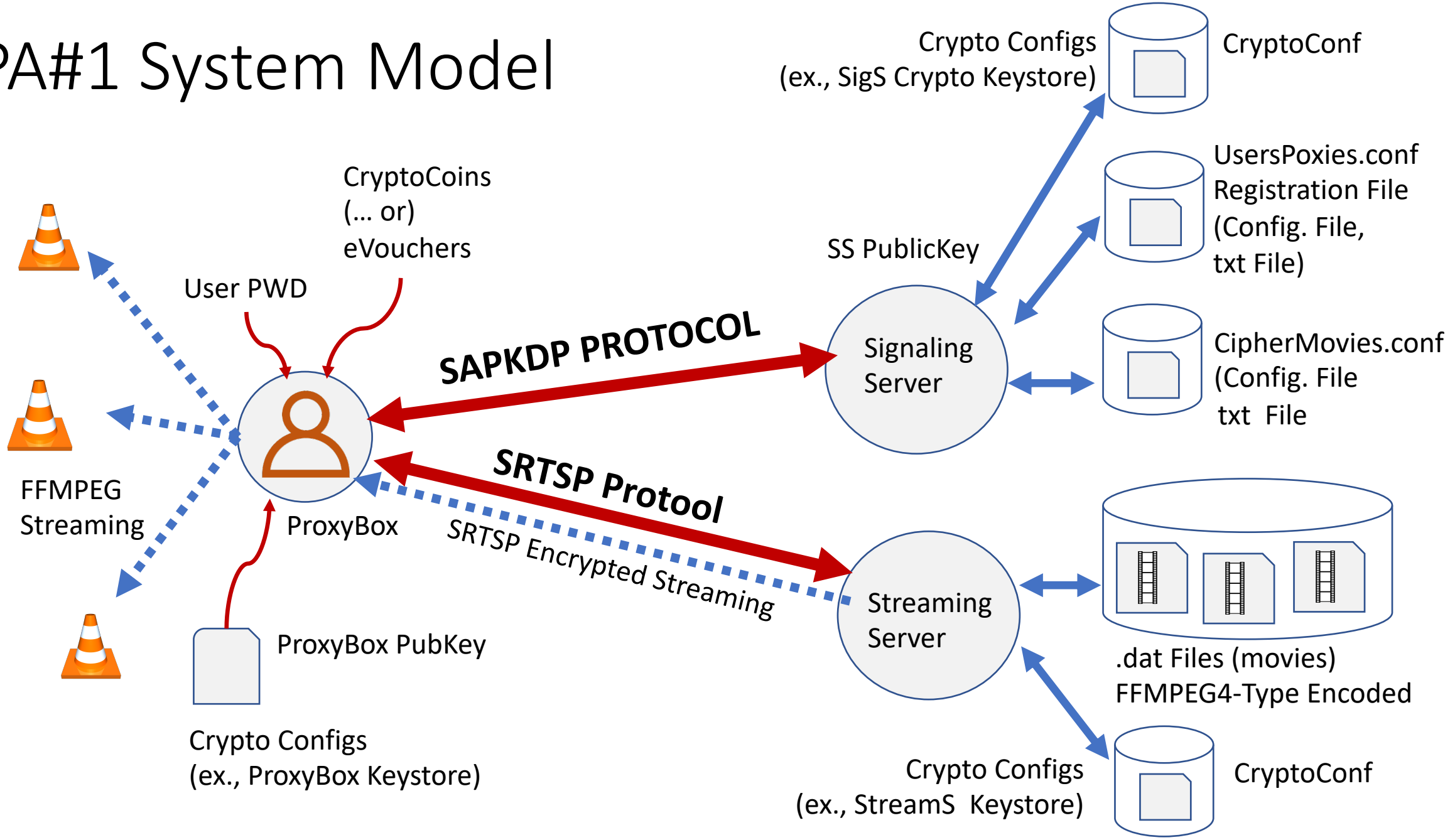
  - Can be implemented over UDP, TCP or HTTP (you can choose the encapsulation in your design and implementation)
  - Important: for the PA#1 delivery you will not use TLS or HTTPS – anyway the protocol will be secure by design and implementation
  - Optionally and later on … you can support it over TLS,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

- **SRTSP**

  Secure Real Time Streaming Protocol

  - Must be implemented over UDP !

PA#1 System Model

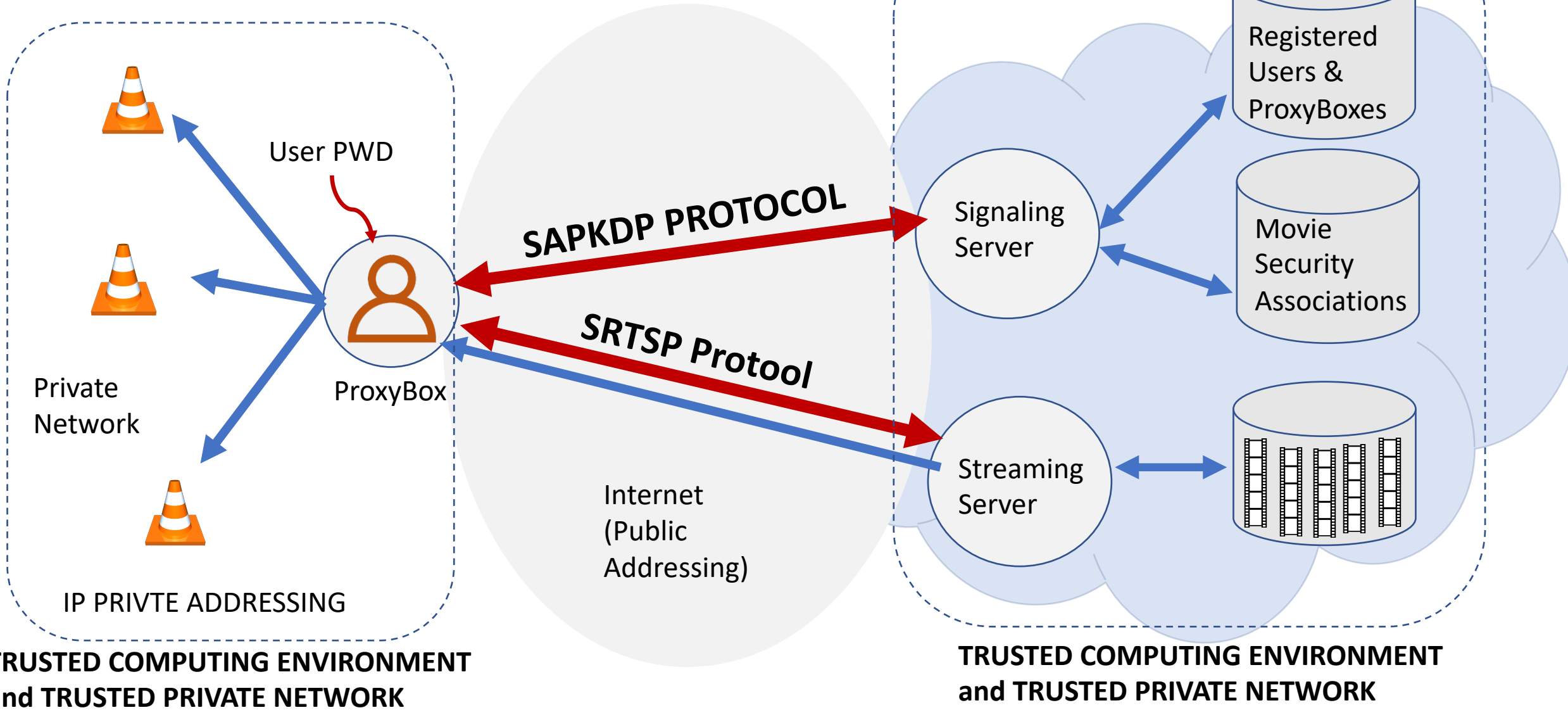


“Hands On”: PA#2

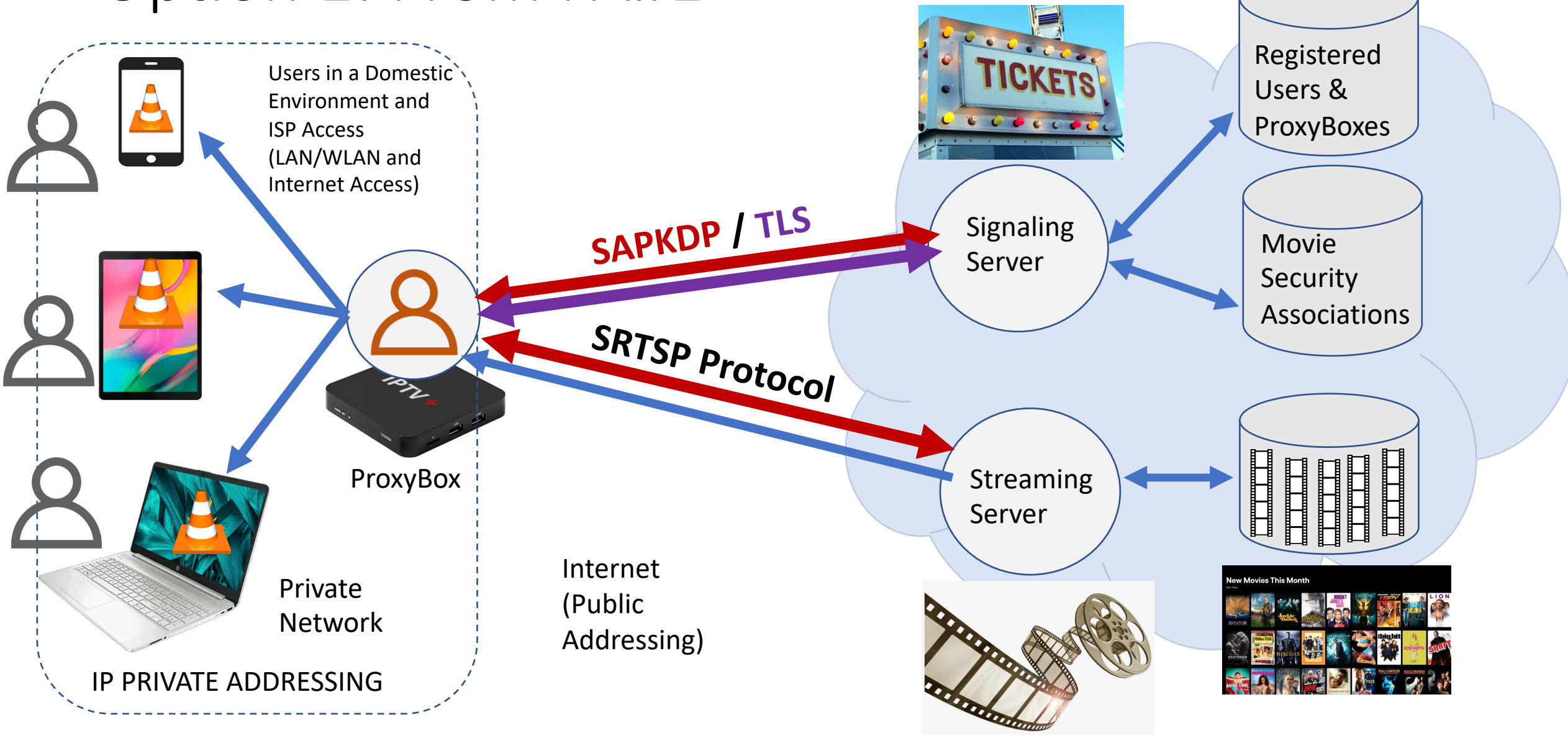
PA#2 Implementation Options

OPTION 1
TLS Configuration for SAPKDP/TLS using JSSE

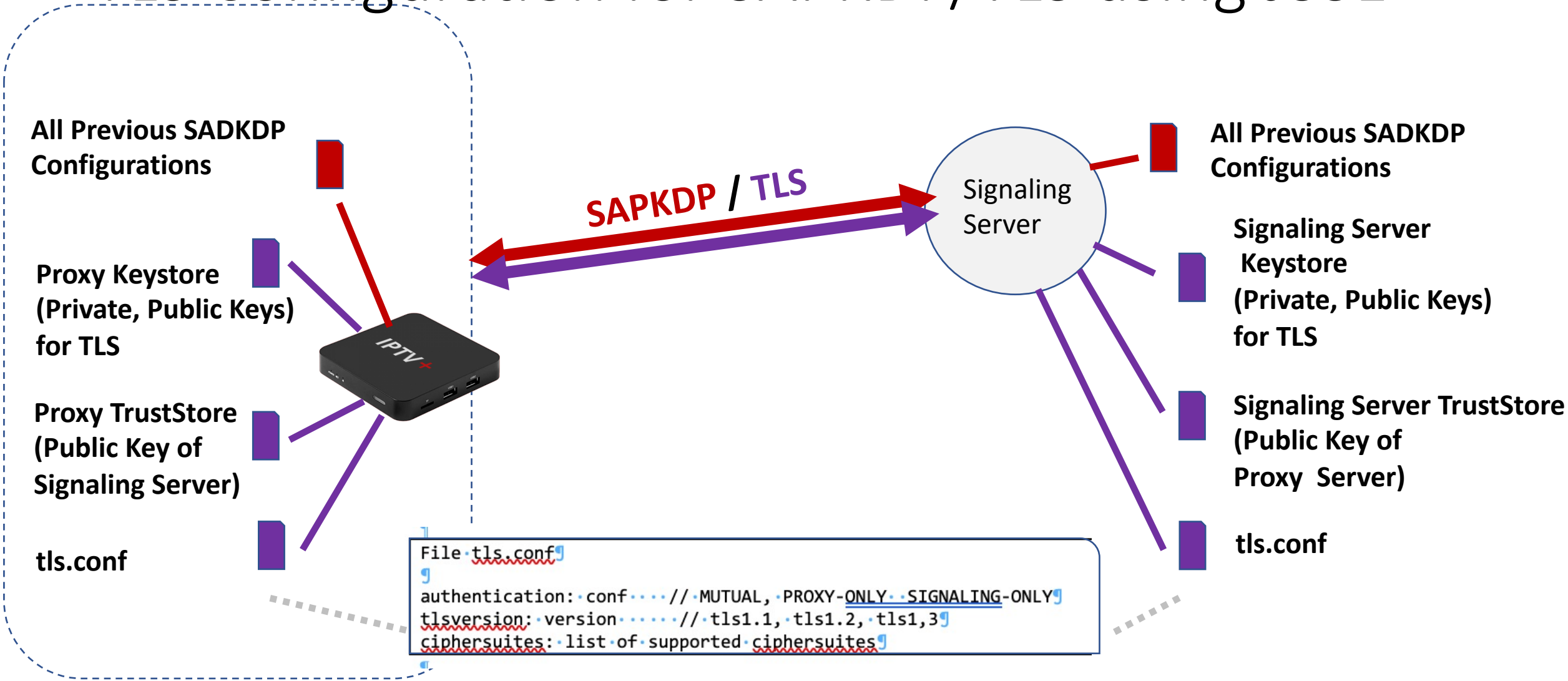
System Model (Rev. PA#1) Functional Architecture



Option 1: From PA#1



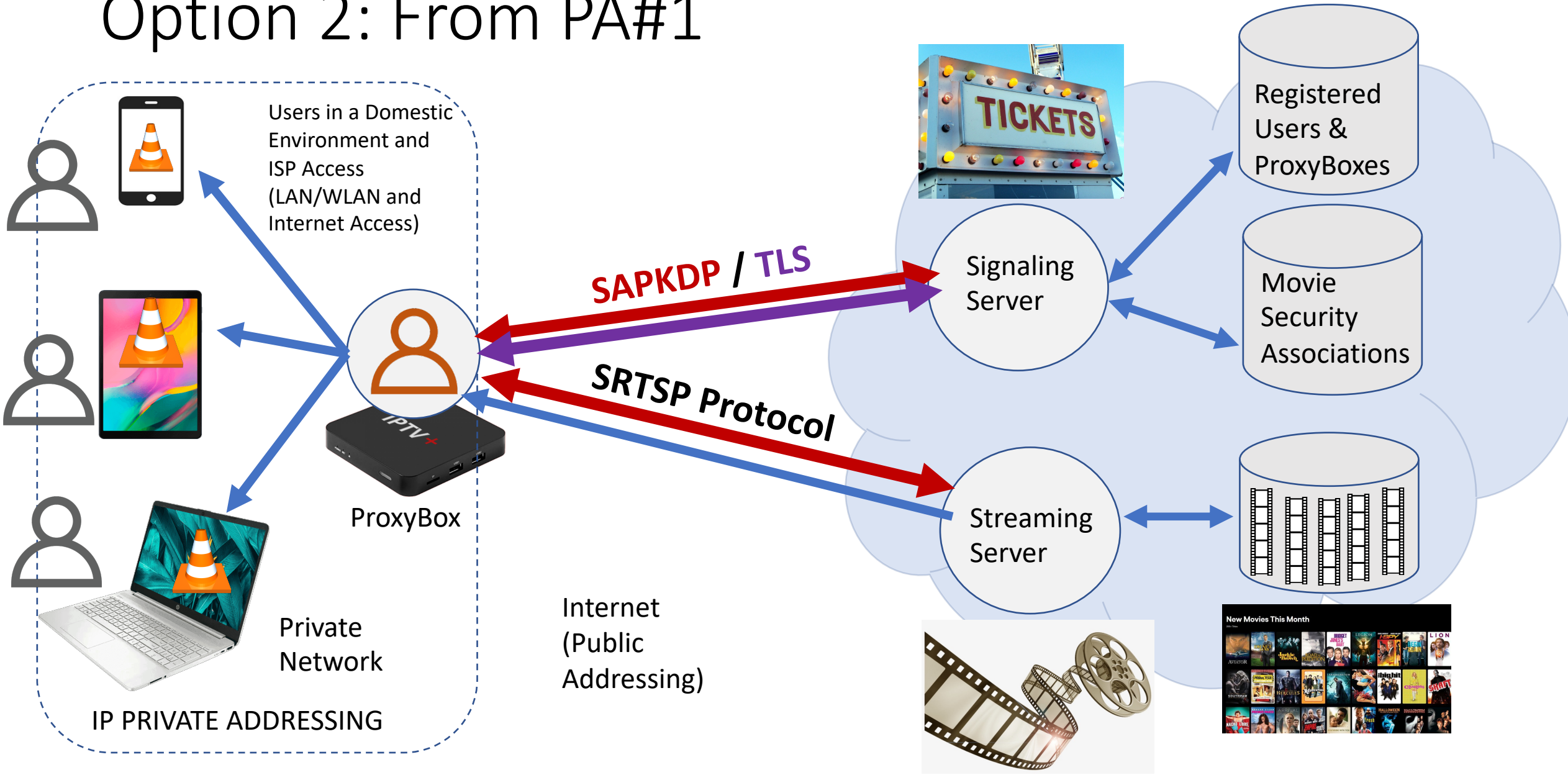
Option 1:
TLS Configuration for SAPKDP/TLS using JSSE



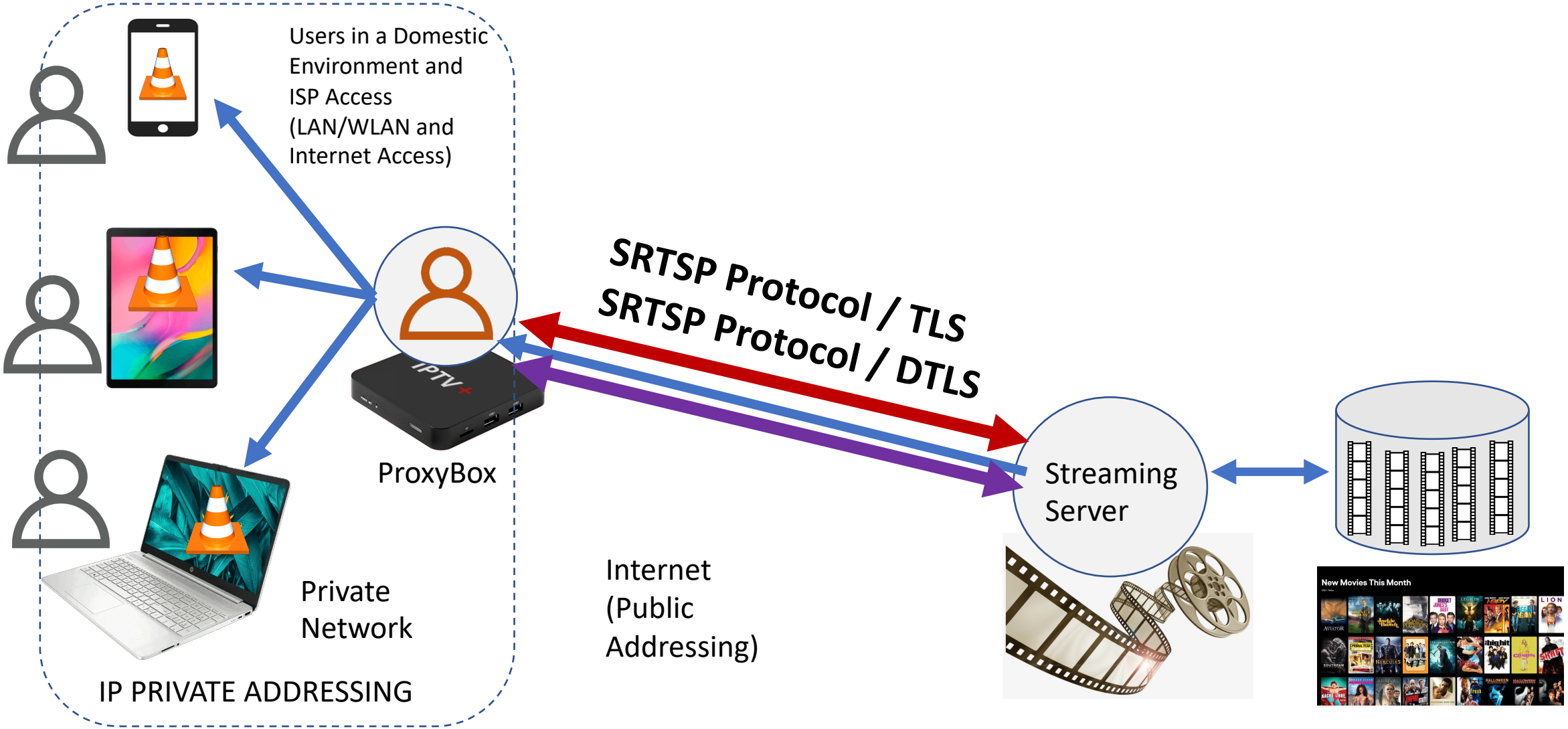
OPTION 2

TLS and DTLS Configurations for SRTSP/TLS and SRTSP/DTLS

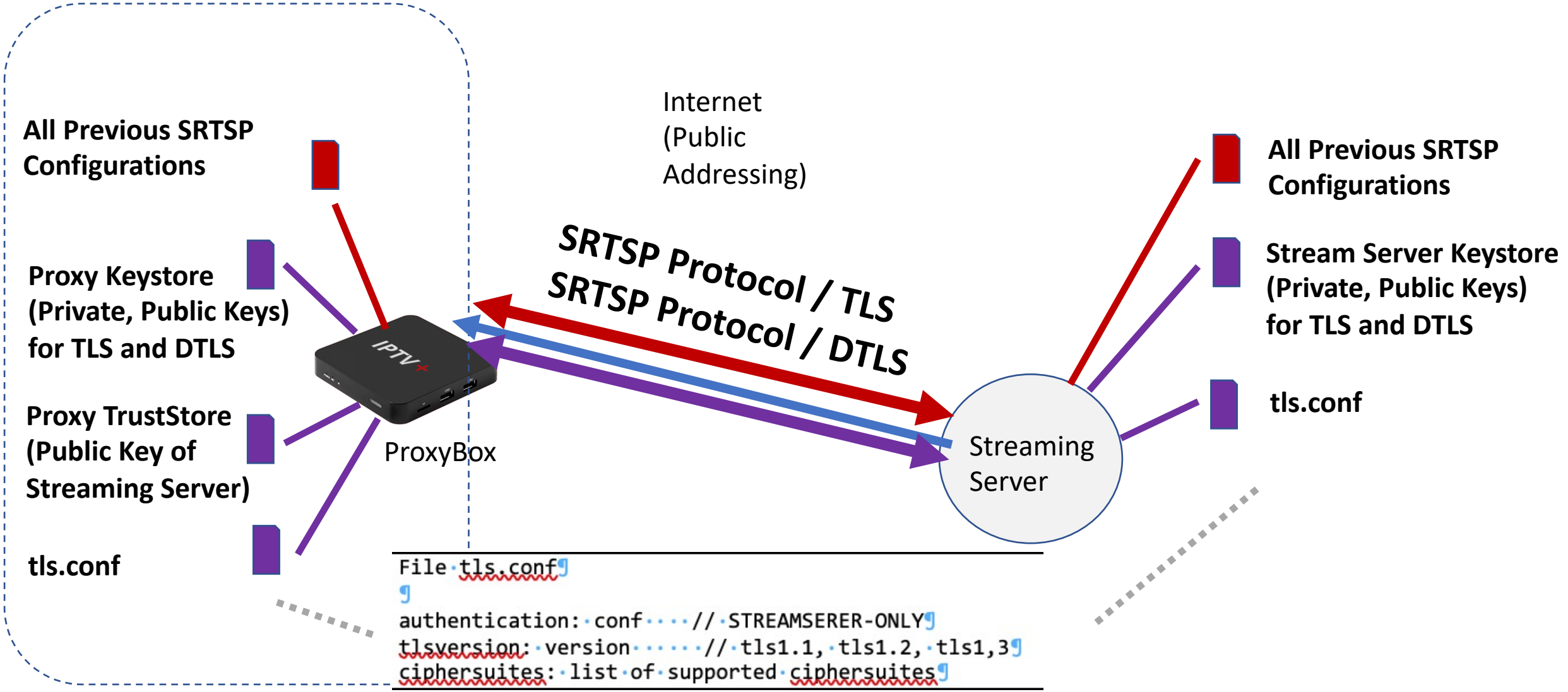
Option 2: From PA#1



Option 2: TLS and DTLS Configurations for SRTSP/TLS and SRTSP/DTLS

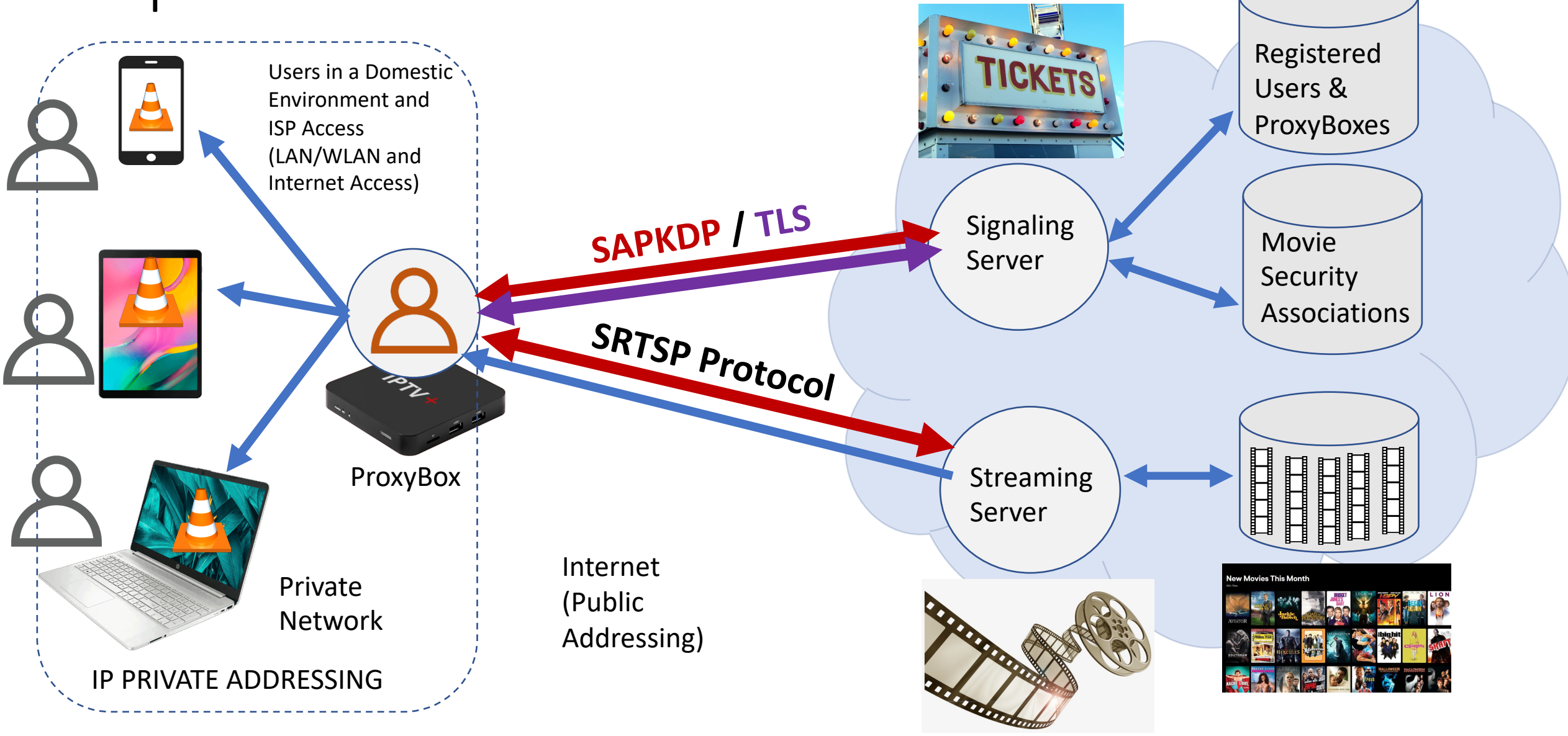


Option 2: TLS and DTLS Configurations for SRTSP/TLS and SRTSP/DTLS

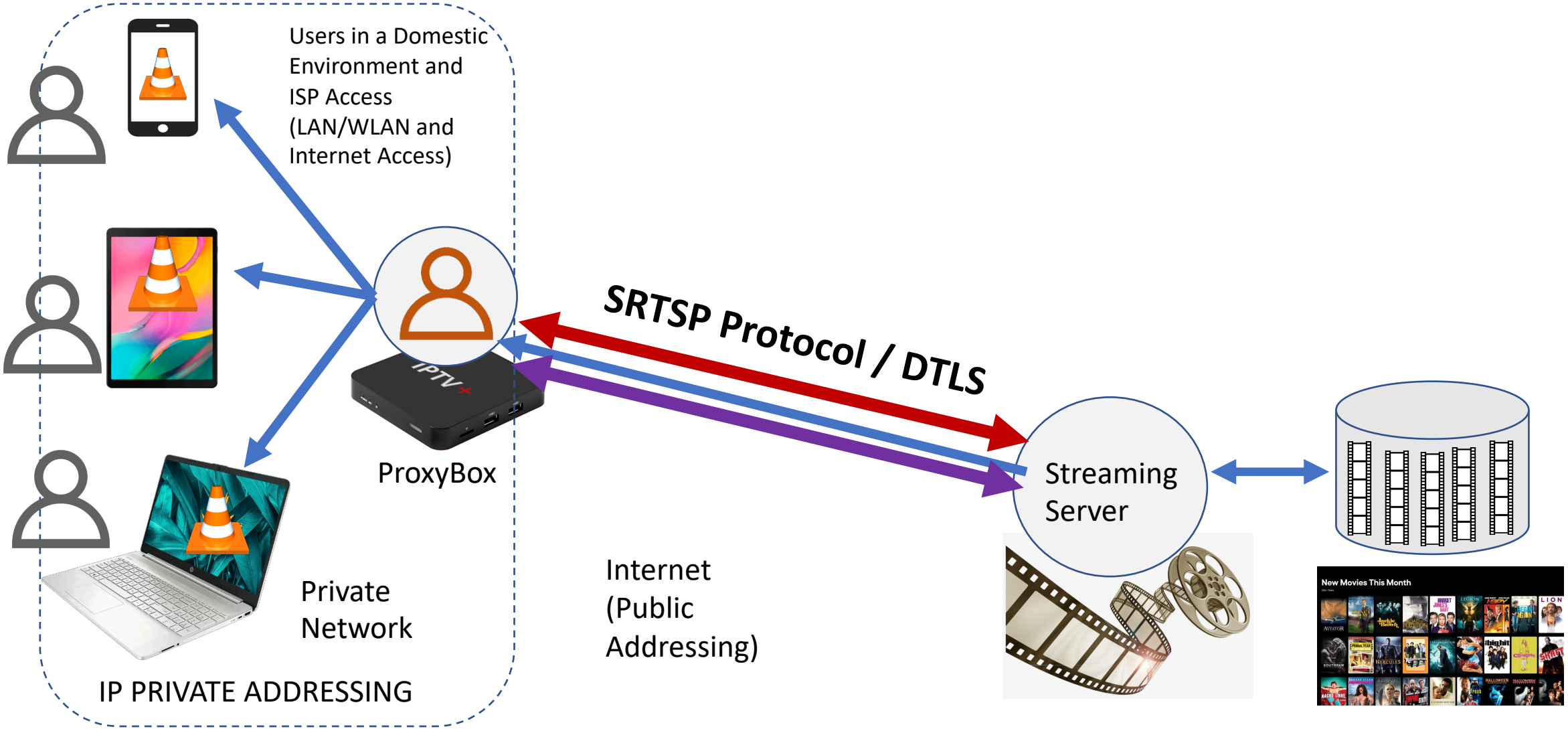


OPTION 3
DTLS Configurations for SRTSP/DTLS

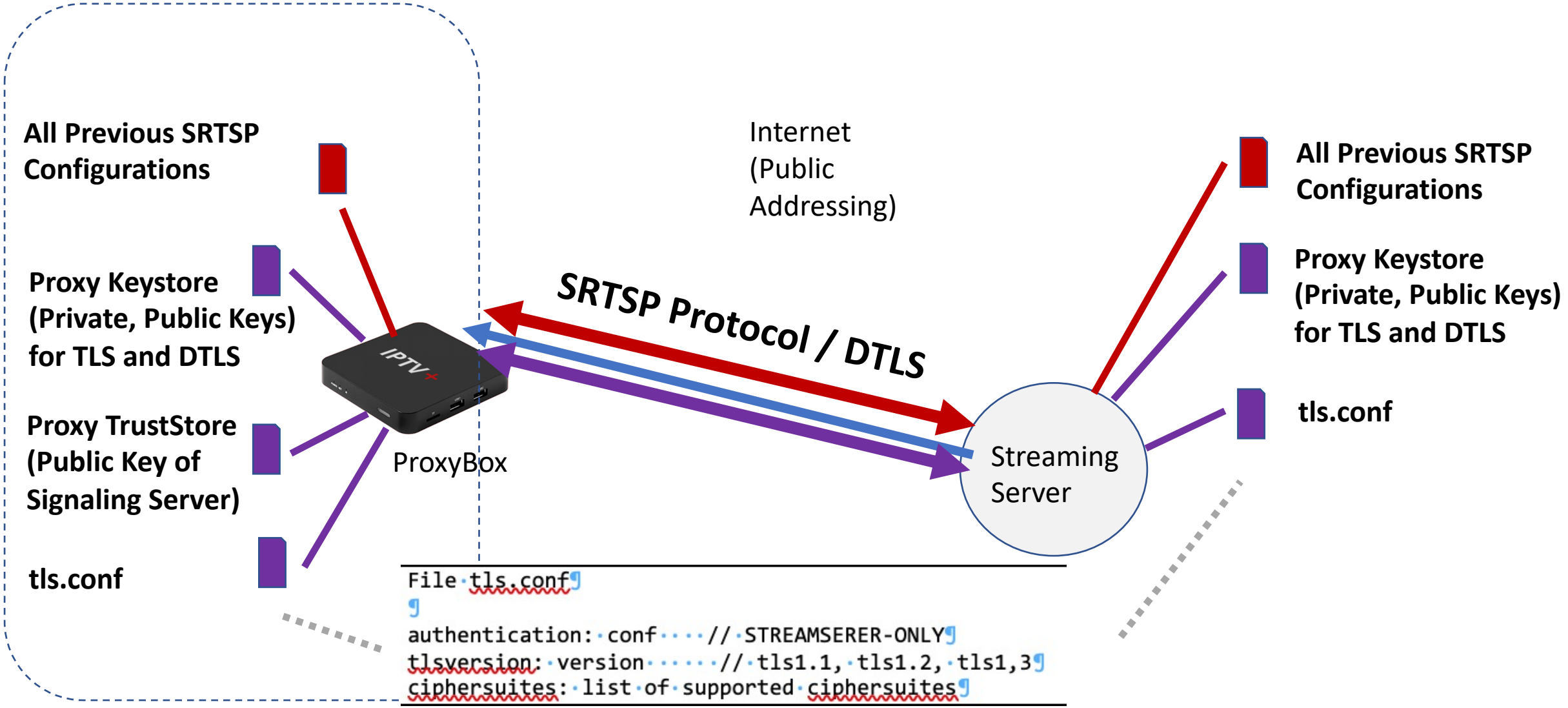
Option 3: From PA#1



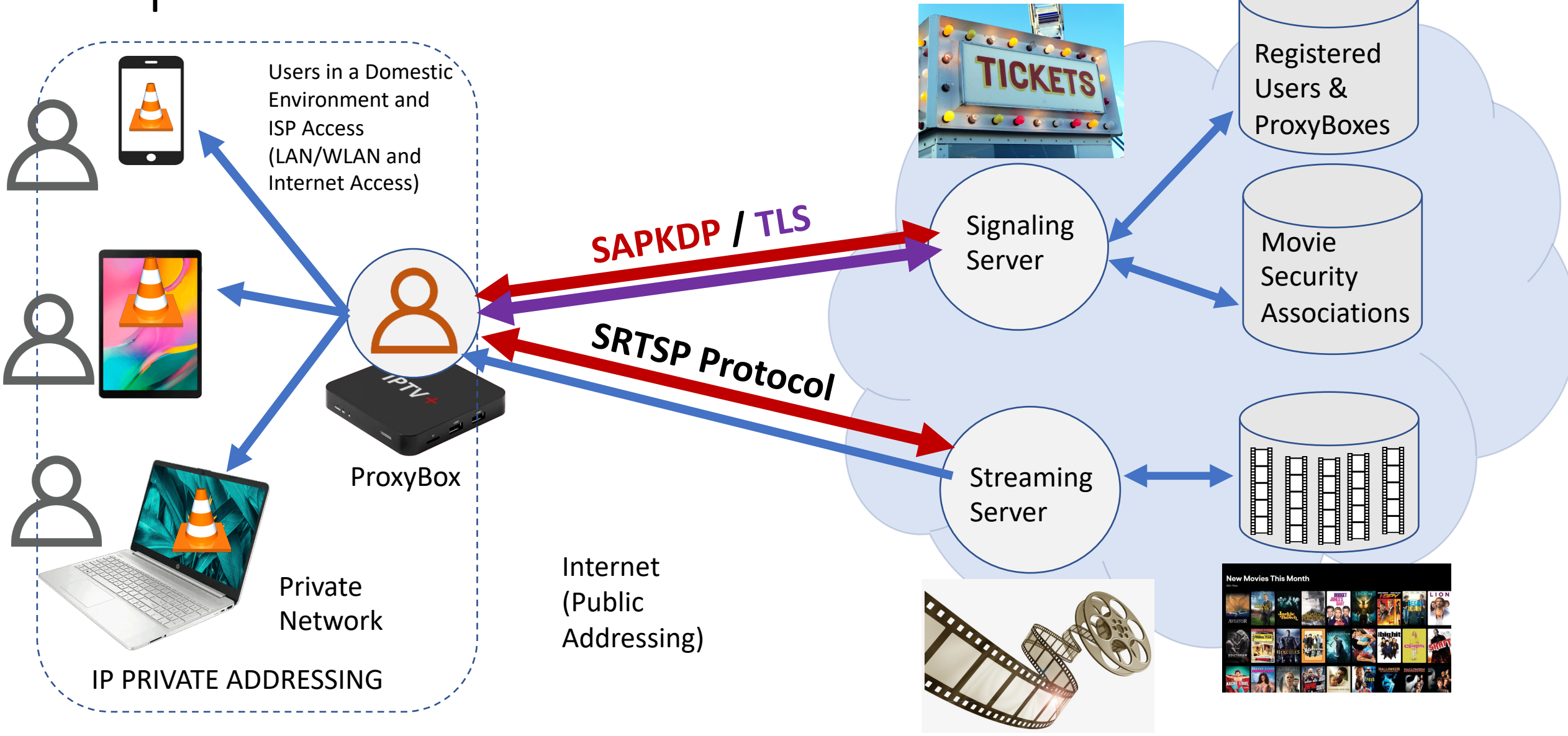
Option 3: DTLS Configurations for SRTSP/DTLS



Option 3: TLS and DTLS Configurations for SRTSP/TLS and SRTSP/DTLS



Option 3: From PA#1



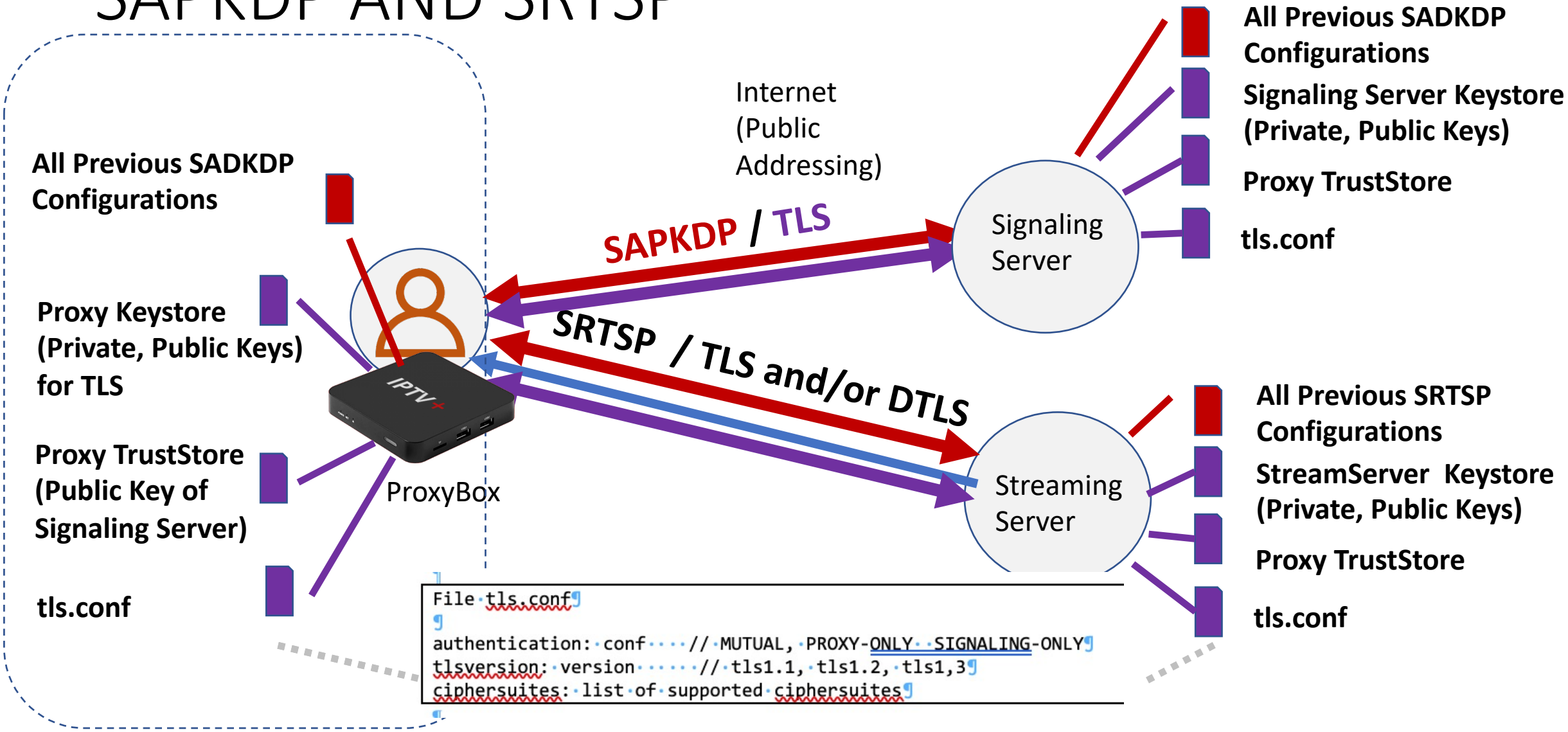
OPTION 4

TLS Configuration for SAPKDP/TLS

and

TLS or DTLS Configuration
for SRTSP/TLS and/or SRTSP/DTLS

Option 4: TLS and DTLS Configurations for SAPKDP AND SRTSP

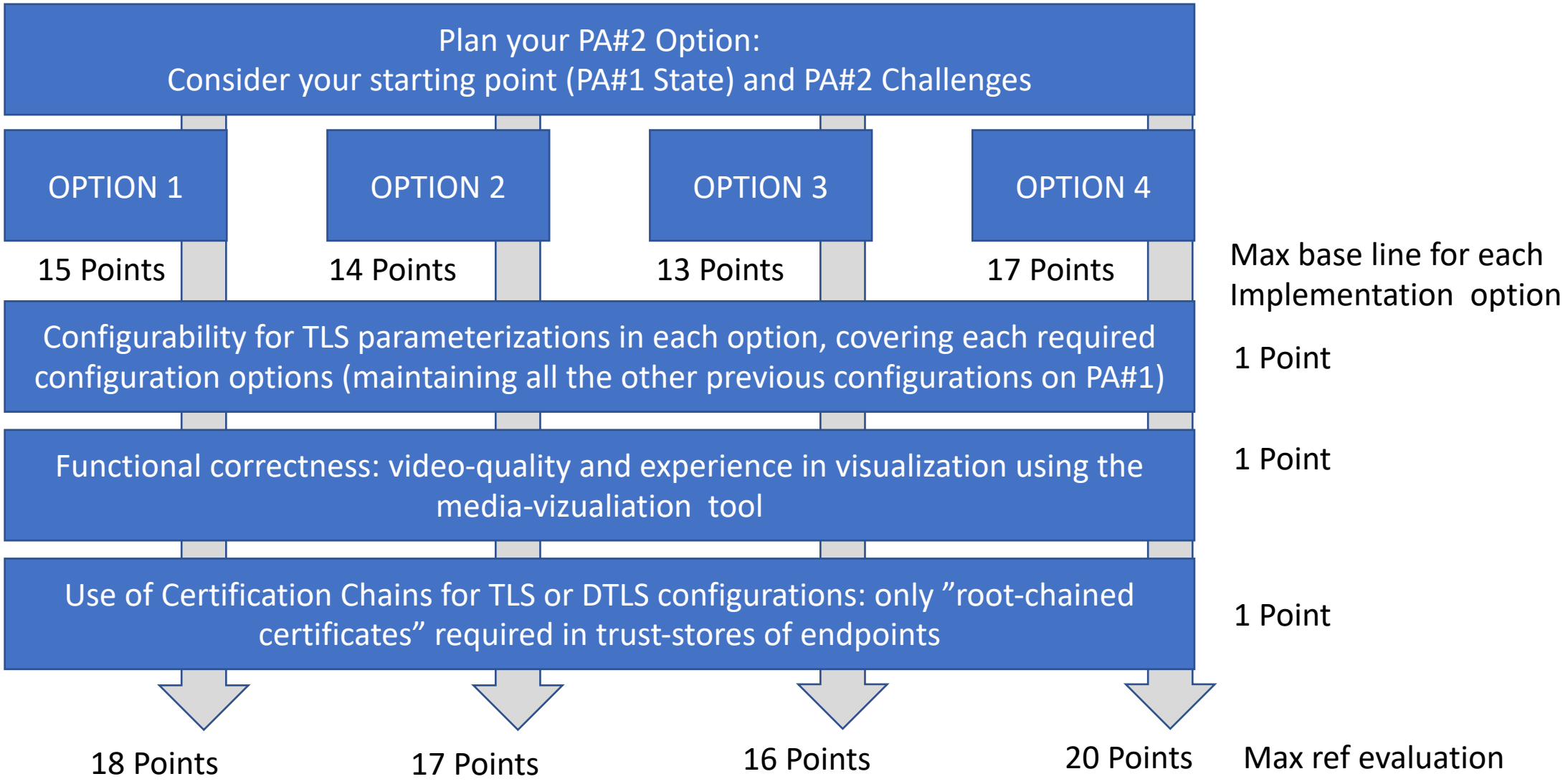


Evaluation: Reference Criteria

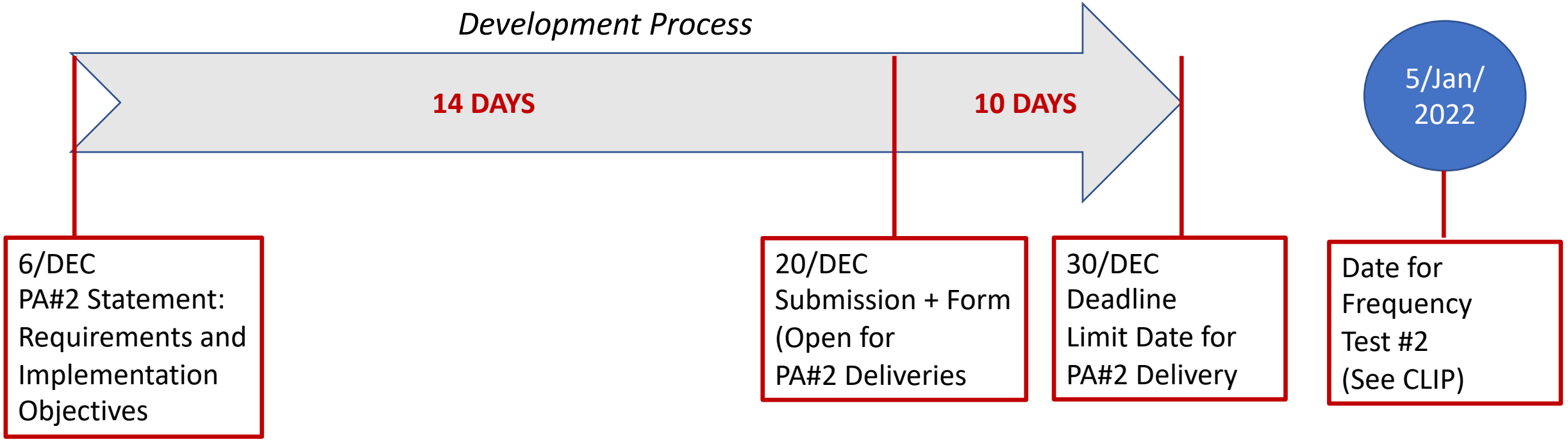
Ref. Criteria

- CRITERIA-1: Option 3 with the provided configurations and correct operation and demonstration, including the quality of the video stream visualization with the used media tool: 13 points
- CRITERIA-2: Options 1, 2, or 3 with the provided configurations and correct operation and demonstration, including the quality of the video stream visualization with the used media tool: 15 points
- CRITERIA-3: Robustness of the solution, applicable for the options 1, 2 or 3: 1 point
- CRITERIA-4: Modularity of the solution, applicable for the options 1, 2 or 3: 1 point
- CRITERIA-5: Use of Certification Chains (with at least two certificates and only one root-certificate used in trusted stores (or trusted keystores) in the the TLS or DRLS endpoints: 1 point
- CRITERIA-5: Option 4 with the provided configurations and correct operation and demonstration, including the quality of the video stream visualization with the used media tool: 17 points

Ref. Criteria



Dates / Plan and Deadlines
PA#2 Dev/Delivery and Frequency Test#2



References and Materials

- See the OA#2 Statement / Requirements and Reference for Evaluation Criteria
- Java and JSSE Documentation
  - **From JAVA 8** … https://docs.oracle.com/javase/8/docs/technotes/guides/security/jsse/JSSERefGuide.html
  - **To … Java 17**: https://docs.oracle.com/en/java/javase/17/security/java-secure-socket-extension-jsse-reference-guide.html
    // See for your Java Version
- Tools and Management of Keystores
  - Tools you can use: keytool, openssl tool, KeyStore Explorer
- Use of Wireshark and openssl tool for your Debug/Experimental Observations
- See also Lab Materials/Examples:
  - Lab-6**:** Practice with X509 Certificates, use of keytool, openssl tool, test of TLS endpoints and how to address Certification Chains (using keytool and openssl tool)
  - Lab-7**:** TLS: Analysys, Java Progfamming and Tools : Programming with JSSE,
  - Lab-8: Use of Wireshark tool for HTTPS and/or TLS traffic traces and debug and auditing of TLS endpoints with some available tools. You have also a shell script tool to test security of TLS endpoints