Location Based Access Control for P2P Video Streaming

Giovanni Simoni - PEER
2016-01-29
Milano

Background and Motivation

LBAC for P2P Video Streaming

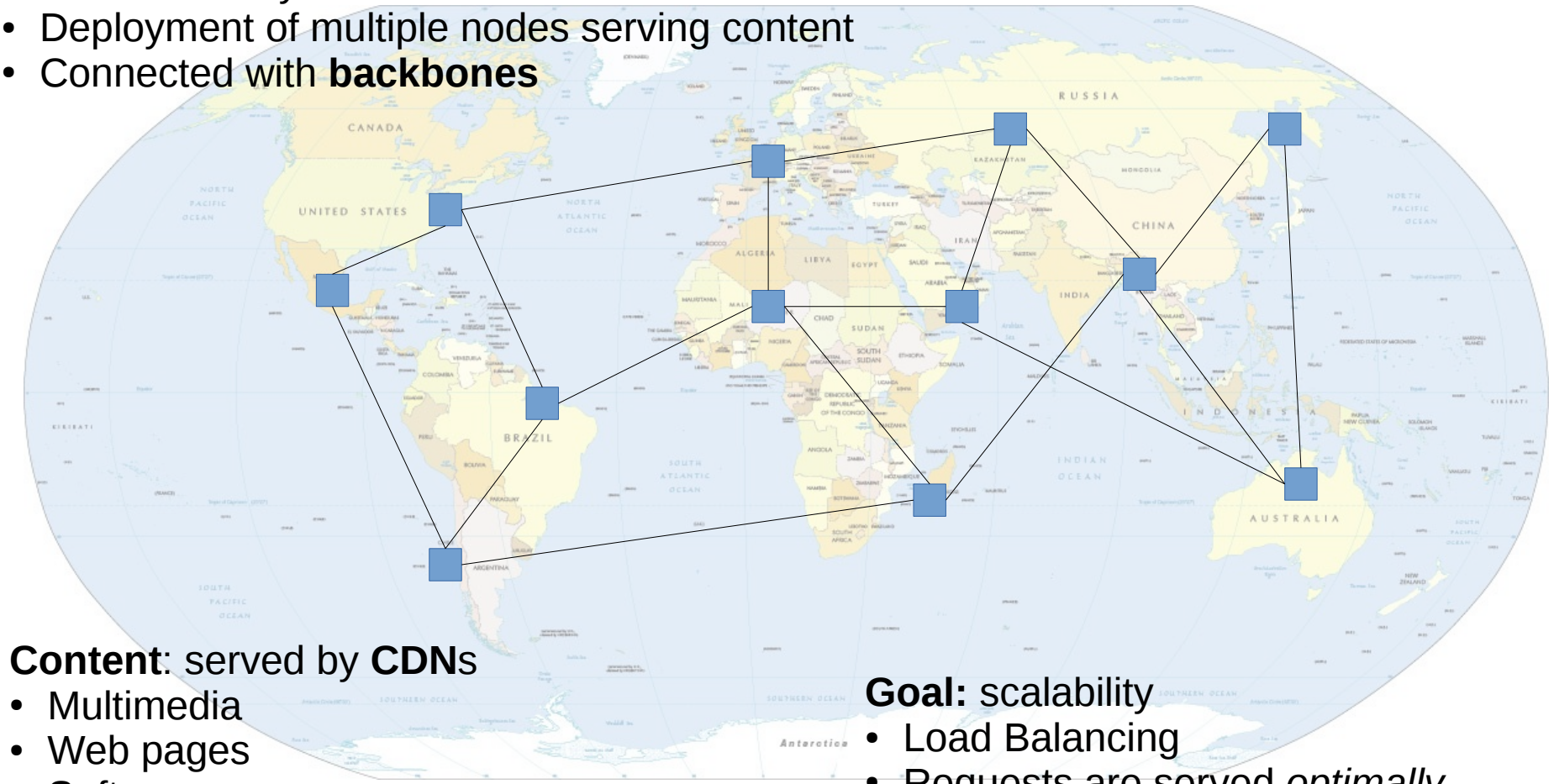
(L stands for “Location”)

- Video streaming is popular in Social Networks
- Also in “dedicated” social networks:
  - Enterprises OSN
  - e.g. Yammer
- We exploit the P2P technology to improve video streaming in Enterprises
- Motivation: improve security with LBAC

Under the hood: CDNs

Content **D**elivery **N**etworks:

- Deployment of multiple nodes serving content
- Connected with **backbones**



Content: served by **CDN**s

- Multimedia
- Web pages
- Software
- …

Goal: scalability

- Load Balancing
- Requests are served *optimally*

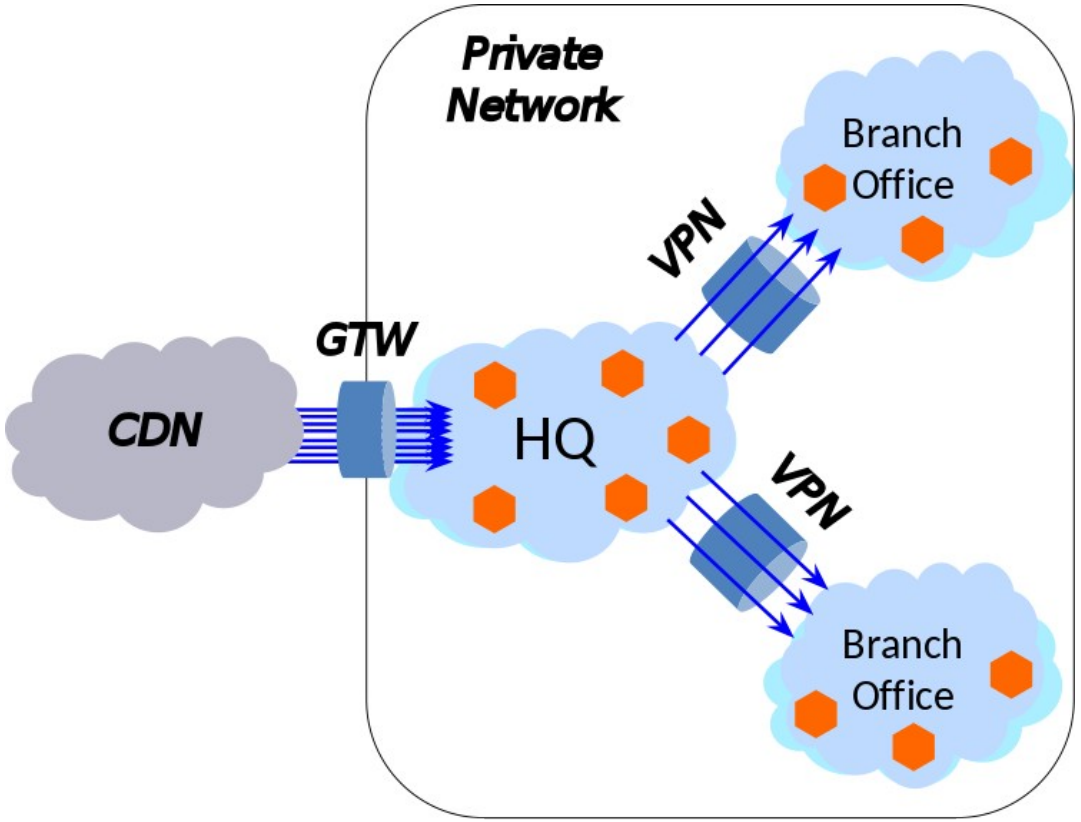
CDNs and Enterprise Networks

Hierarchical structure

- Central headquarter
- Many sites interconnected by Virtual Private Networks
- Security: all the traffic goes through VPNs
  - Defeating the purpose of CDNs

Live Video Events

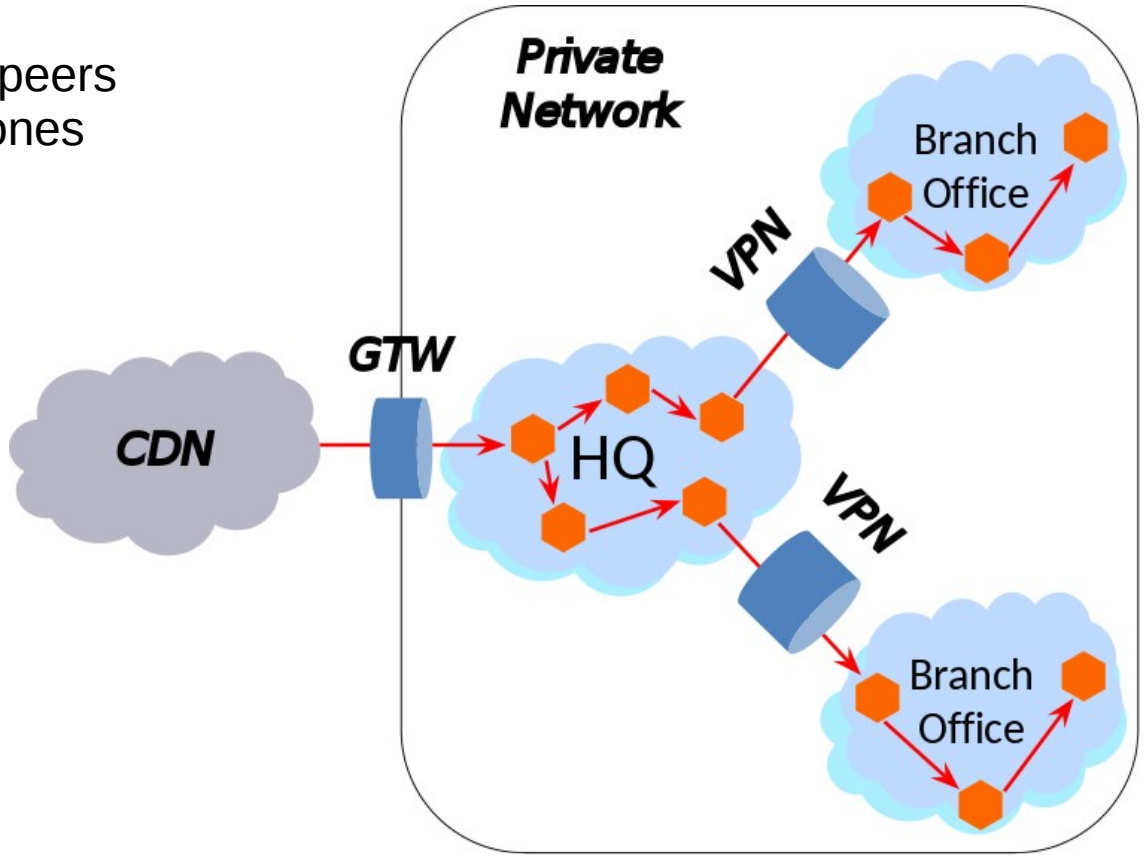
- Each node runs independent requests
- Redundant transmissions of the same video stream
- **Bottleneck!**



Hive: distributed CDN
(using P2P)

Principles:

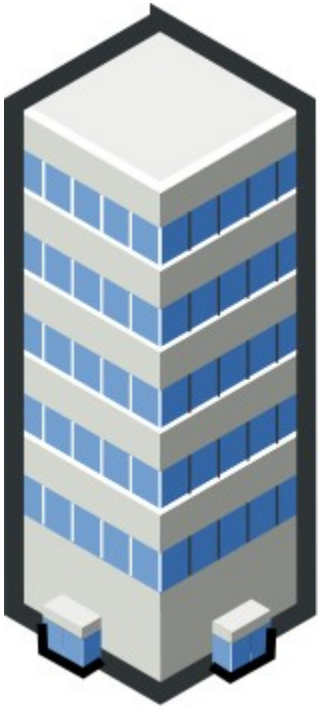
- The content is retrieved by peers and re-distributed to other ones
- **Peers** take the role of **CDN** endpoints.



Confidential Video content in Enterprise OSN

- Confidential data shared through OSNs
  - Inclusive of knowledge base, in form of Videos
- **R**ole **B**ased **A**ccess **C**ontrol
  - But people have *{laptops, smartphone, tablets, whatever}*
- **L**ocation **B**ased **A**ccess **C**ontrol
  - **Additionally** we consider the user location as context

LBAC



"The Corporate Building"

Many flavours of LBAC

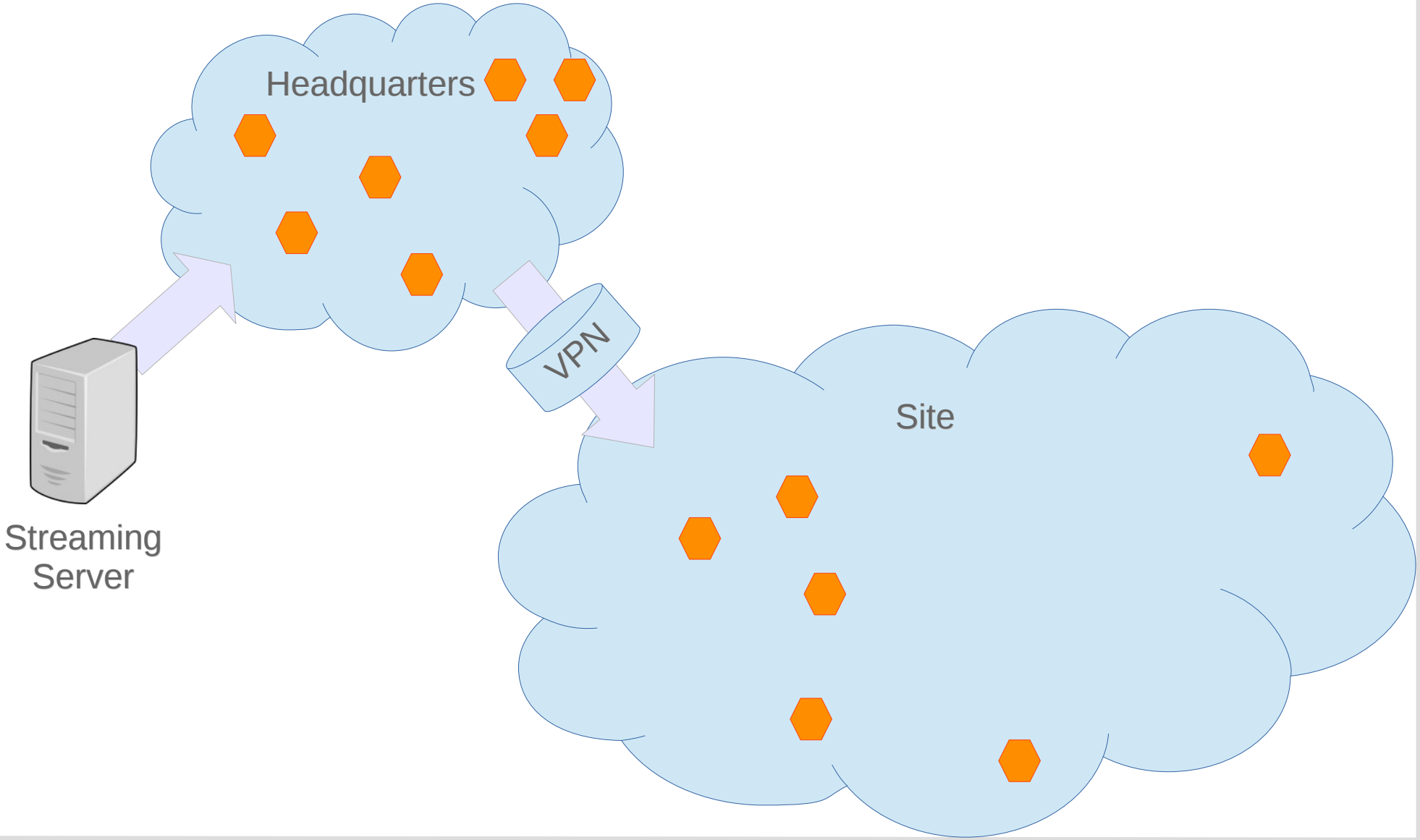
- Location detection
  - GPS, GSM antennas, RFID, **WiFi**
- Different granularity
  - Geographical region ← to → Meters

Contribution

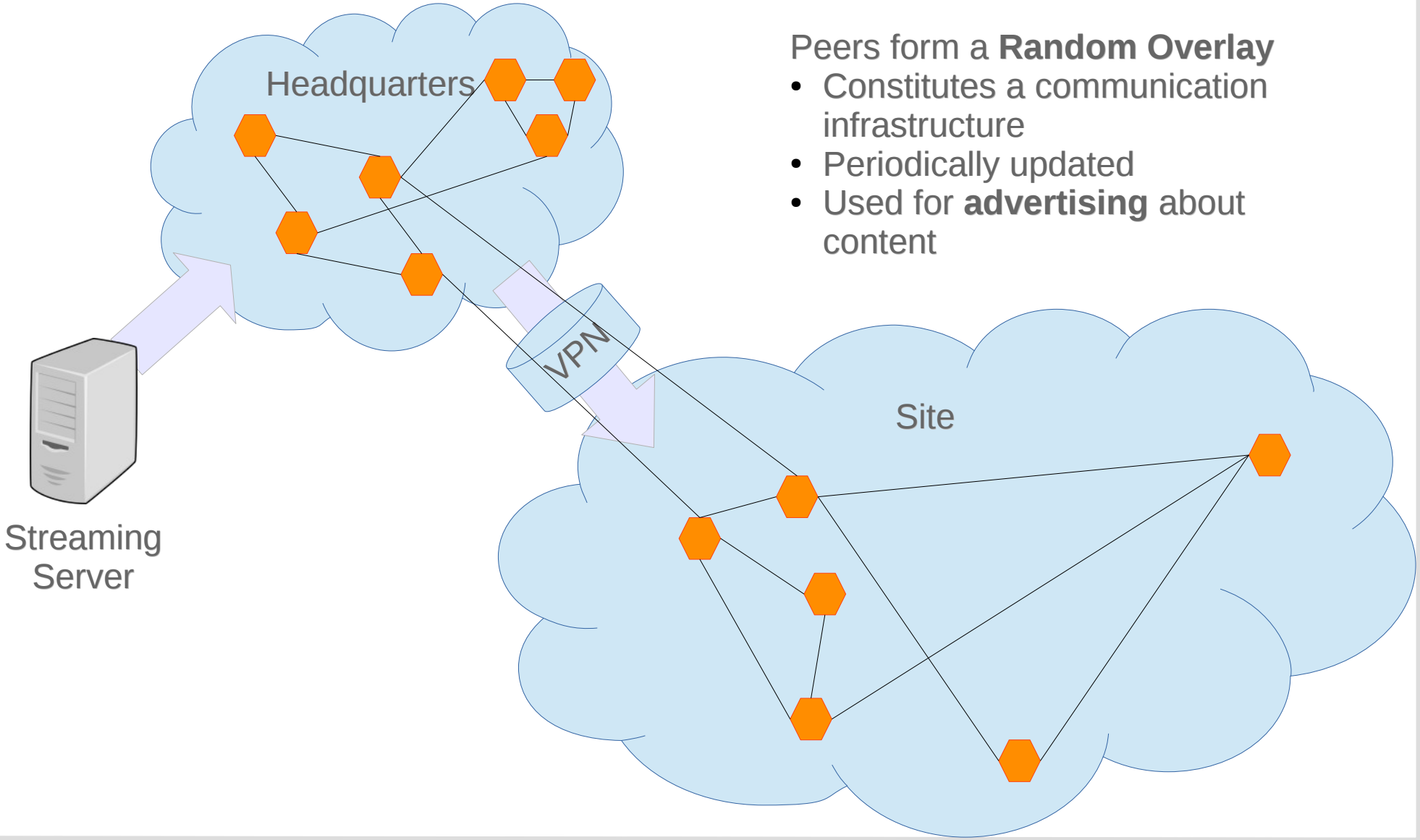
- Design of a **simple** LBAC distributed protocol
  - On the top of Hive
- Testing in simulation
- Measure of performances
  - User experience
  - Effectiveness of the enforcement

The Hive Client

The Hive System



The Hive System



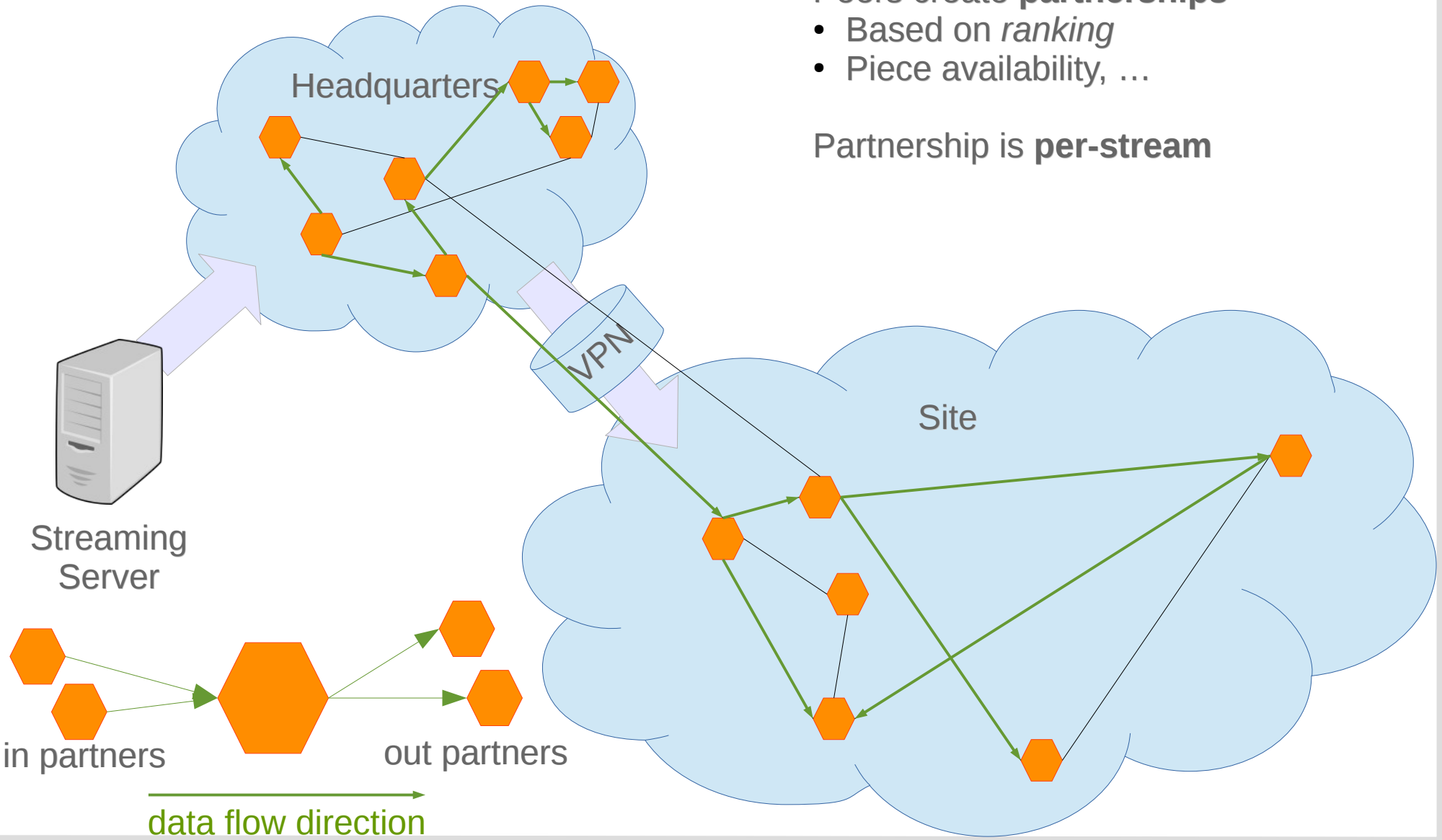
Peers form a **Random Overlay**

- Constitutes a communication infrastructure
- Periodically updated
- Used for **advertising** about content

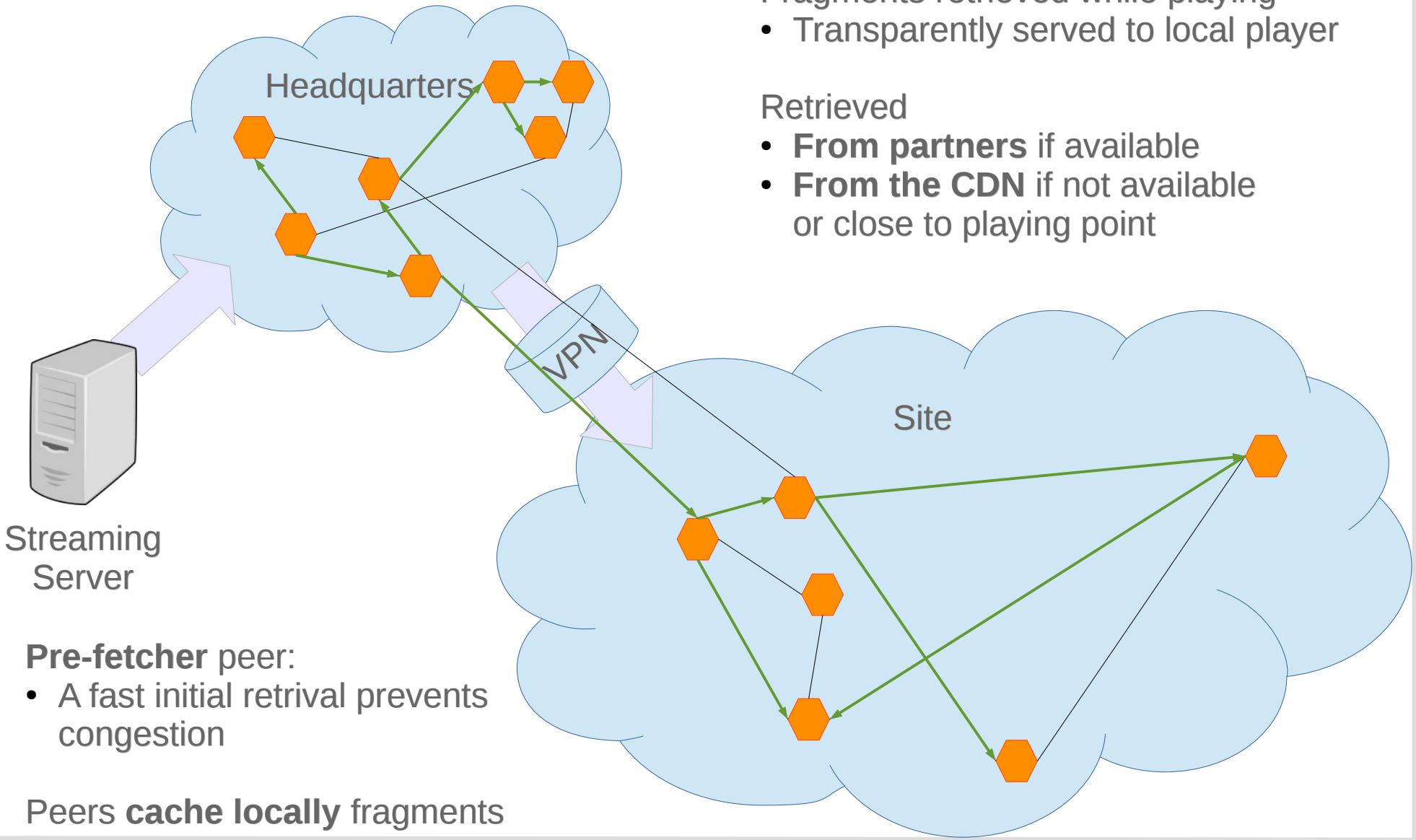
The Hive System

Peers create **partnerships**
- Based on *ranking*
- Piece availability, …

Partnership is **per-stream**



The Hive System



Fragments retrieved while playing
- Transparently served to local player

Retrieved
- **From partners** if available
- **From the CDN** if not available or close to playing point

Pre-fetcher peer:
- A fast initial retrival prevents congestion

Peers **cache locally** fragments

LBAC:
Extension to Architecture

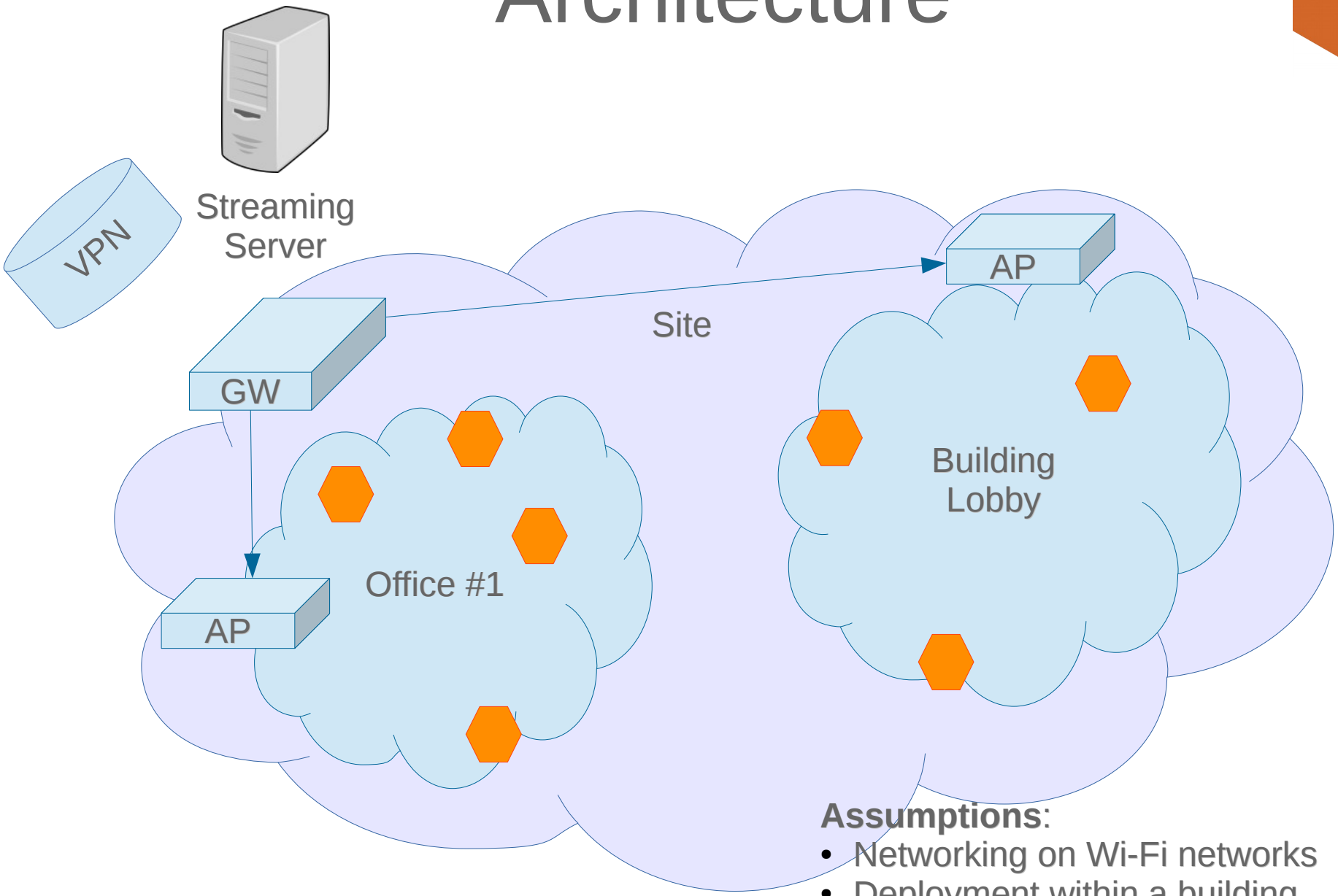
Architecture

Streaming Server

VPN

Site

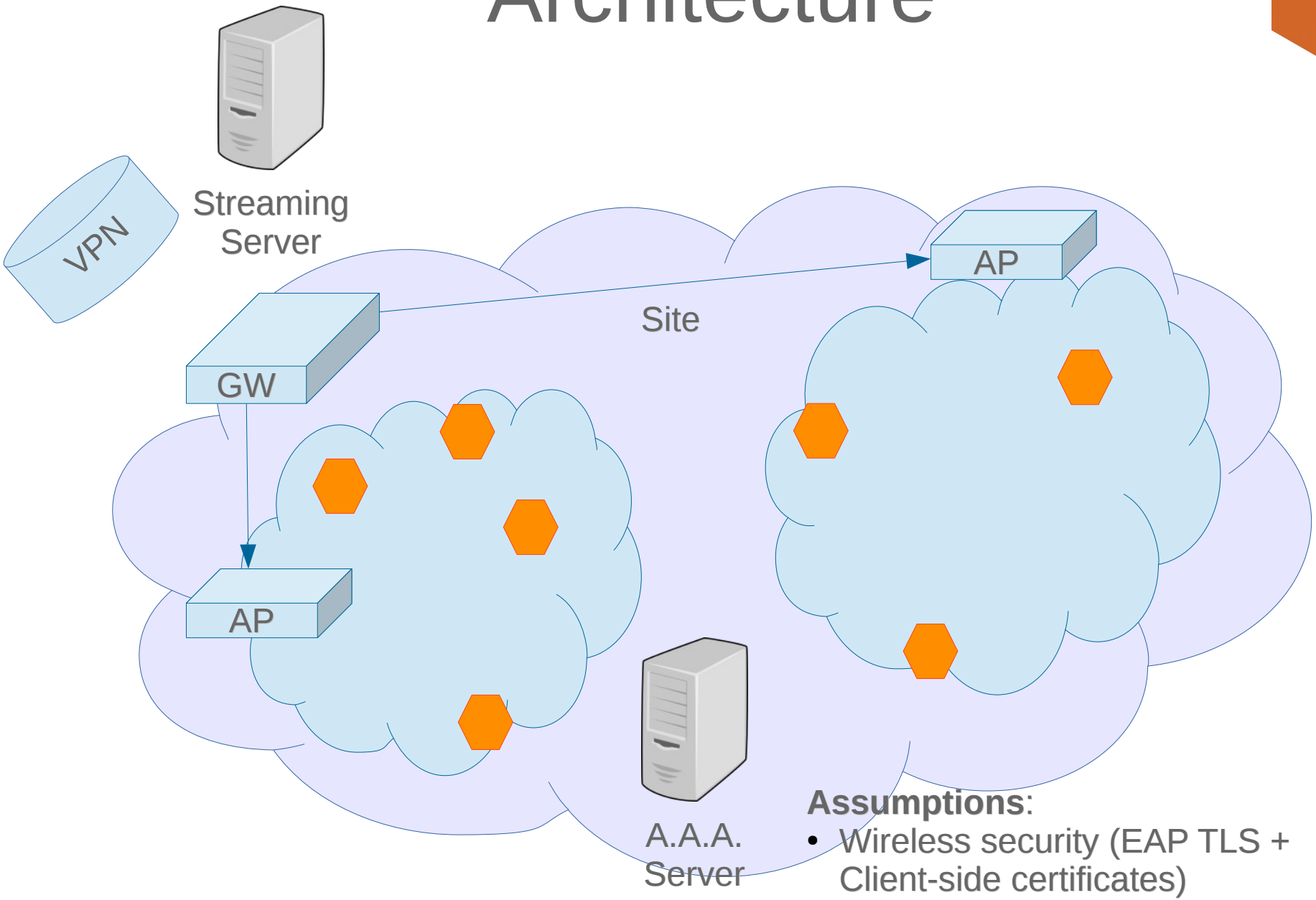
Architecture



Assumptions:

- Networking on Wi-Fi networks
- Deployment within a building
- We identify locations w.r.t. APs

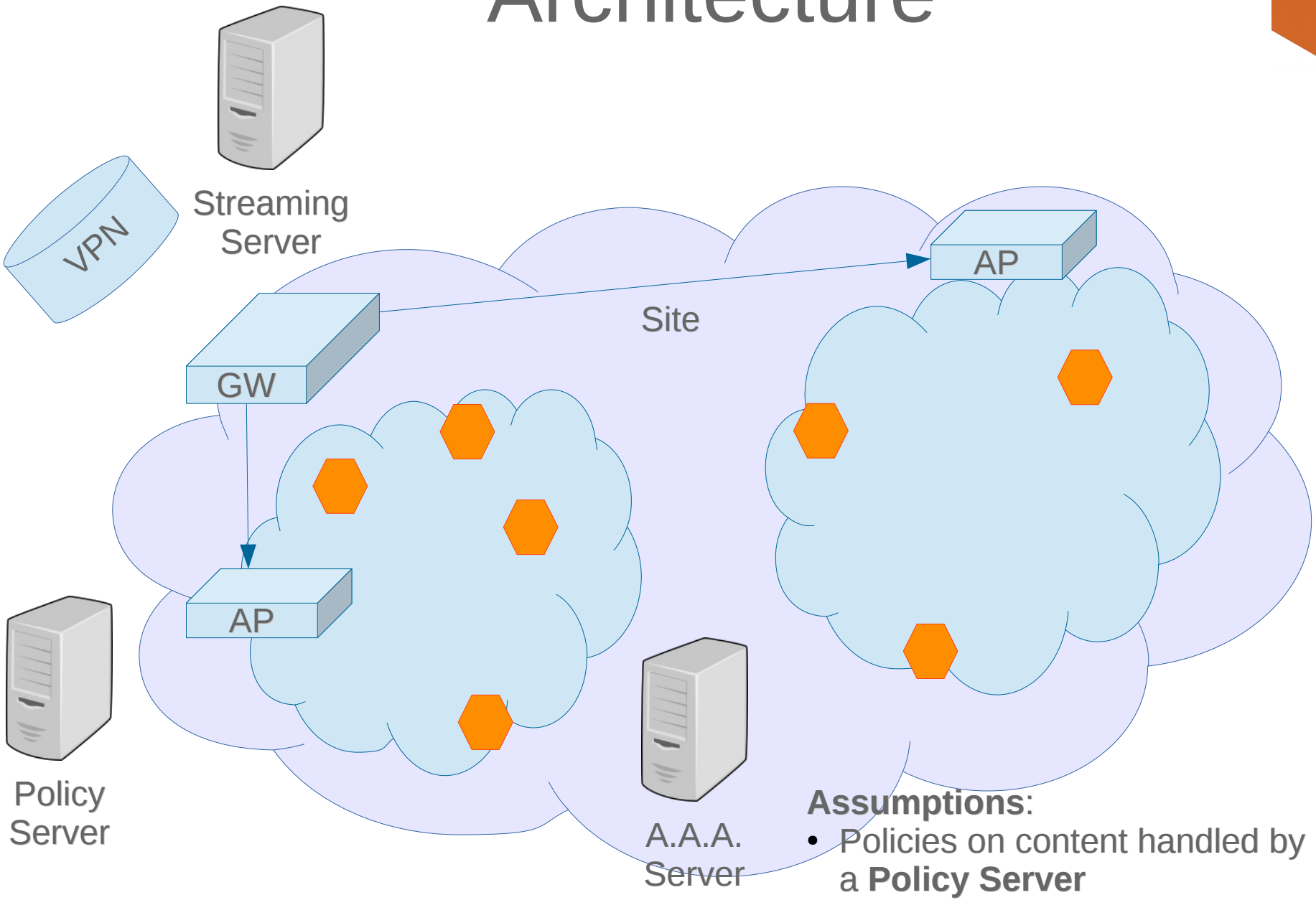
Architecture



Assumptions:

- Wireless security (EAP TLS + Client-side certificates)

Architecture



Assumptions:

- Policies on content handled by a **Policy Server**

LBAC:
Extension to the Protocol

The protocol

- Assuming simple policies

Content URL	Location: Lobby	Location: Office #1	Location: Office #2	Location: Relax Area
https://...	yes	yes	yes	yes
https://...	no	yes	yes	no
https://...	no	no	yes	no

- We can add orthogonally different flavours of access control
  - e.g. Role Based Access Control

Simple Protocol

- Based on certified authorizations
  - Policy server emits **signed** **C**ertified **A**ccess **G**rant
    - CAG(who=user1, loc=loc1, item=http://..., expire=1454189691)
  - AAA server emits a **signed** **C**ertified **L**ocaiton **P**roof
    - CLP(who=user1, loc=loc1, expire=1454189691)
- Proof have an expiration
  - CAG => **P**olicy **C**ertification **V**alidity parameter
    - Accounting for changes in the policy
    - Medium to long expiration (e.g. hours or days)
  - CLP => **L**ocation **C**ertification **V**alidity parameter
    - Accounting for user movement
    - Short expiration (e.g. seconds or minutes)
  - Absolute expiration time: Emission Time + Period

Enforcement points

The Client can obtain data from three sources

- The CDN connection
- In-Partners
- The local Cache

P2P Retrieval

- Client modification: prevent **out-partnersip** towards unauthorized peers
- Partnership establishment:

  Candidate **out-partner** user1 must provide certificates:

  - CAG(who=user1, loc=loc1, item=V, expire=future1)
  - CLP(who=user1, loc=loc1, expire=future2)
- Partnership maintenance:
  - Expires at time T_e = min {future2, future1}
  - Client user1 must keep certificates up to date
  - Change of location:
    - CLP(who=user1, loc=loc2, expire=future2+...)
    - Provided with matching CAG(who=user1, loc=loc1, item=V, expire=future1+...)

CDN Source Retrieval

- CDN data is served through HTTP[S]
  - Many different server implementations could be used
  - Trick: using regular HTTPS credentials, temporarily
- Enabled and periodically maintained
  - Request to AAA
  - Must provide
    CAG(who=user1, loc=loc1, item=V, expire=future1)
  - Must be renewed before future1 with
    CAG(who=user1, loc=loc1, item=V, expire=future2)
- Setup during Manifest retrieval: fallback must be fast

Local Retrieval

- The client aggressively fetches fragments for future needs
  - Content is stored in encrypted form
- Requirement: check the authorization before deliverying to the player
  - As simple as: check my own permission

Evaluation

Simulation Scenario

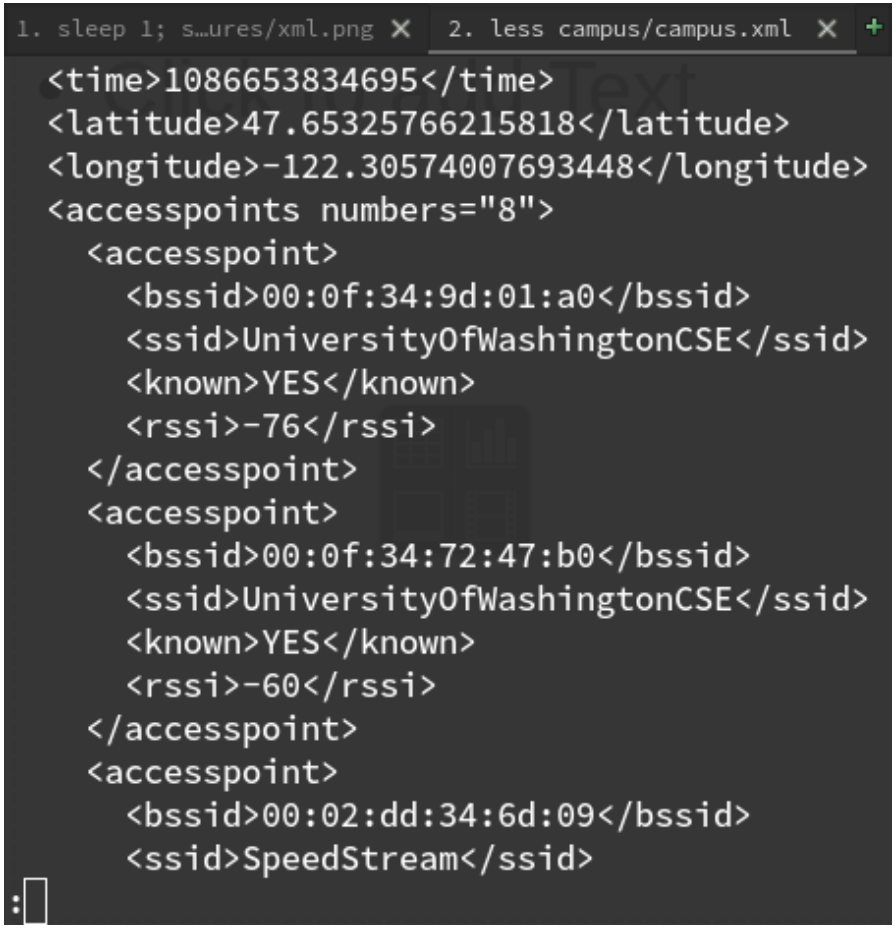
From the streaming perspective

- Simple scenario used for unit testing
  - Ingredients: network structure + streaming pattern
  - Focus on one site
- Goals:
  - User experience perspective
  - How fast is the enforcement

Simulation Scenario

From the movement perspective

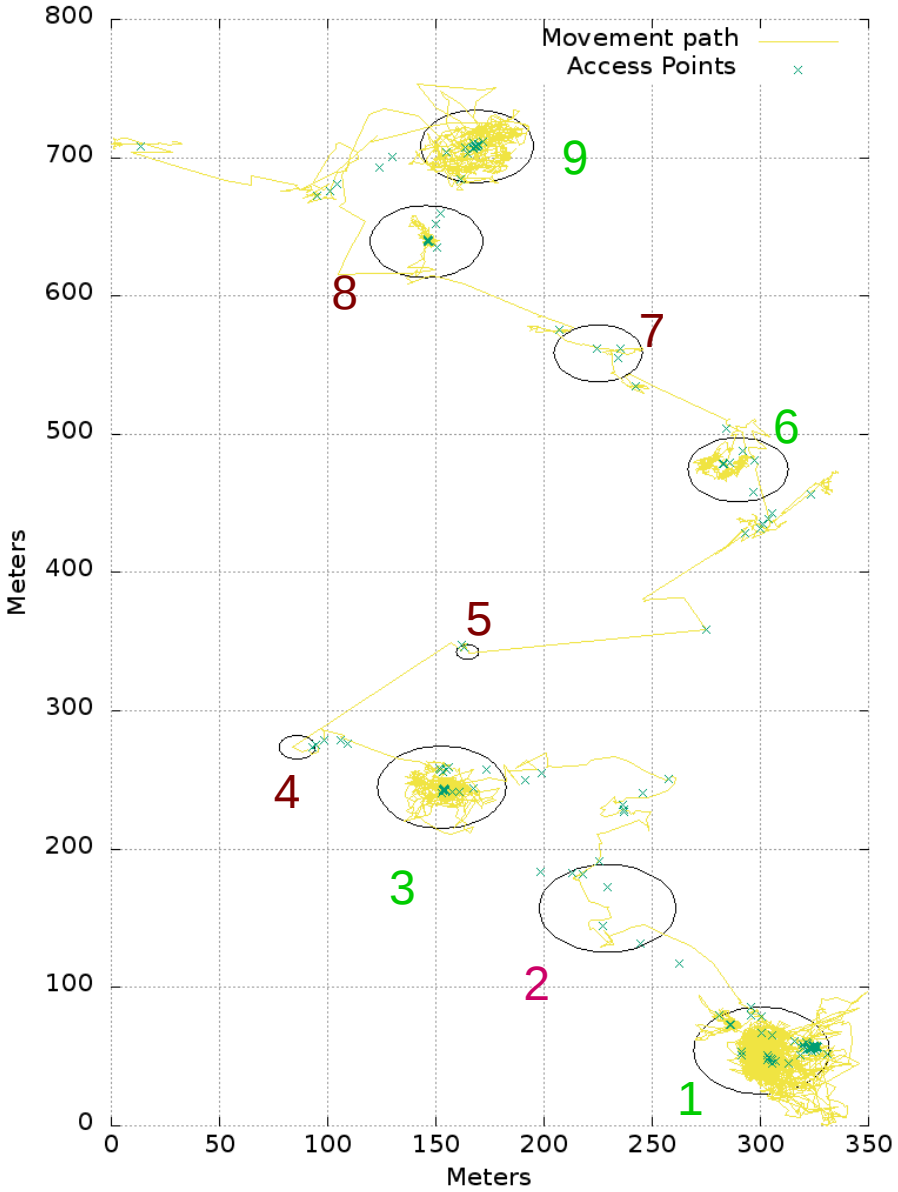
- Datasets are difficult to find, but I was lucky
  - Crawdad archive *uw/places*[1]
- Companion paper on *extraction of relevant places from locations*[2]



[1]: Jong Hee Kang, Gaetano Borriello, William Welbourne, and Benjamin Stewart. CRAWDAD dataset uw/places (v. 2006-05-02). Downloaded from http://crawdad.org/uw/places/20060502

[2] Jong Hee Kang, William Welbourne, Benjamin Stewart, and Gaetano Borriello. Extracting places from traces of locations.

Simulation Scenario

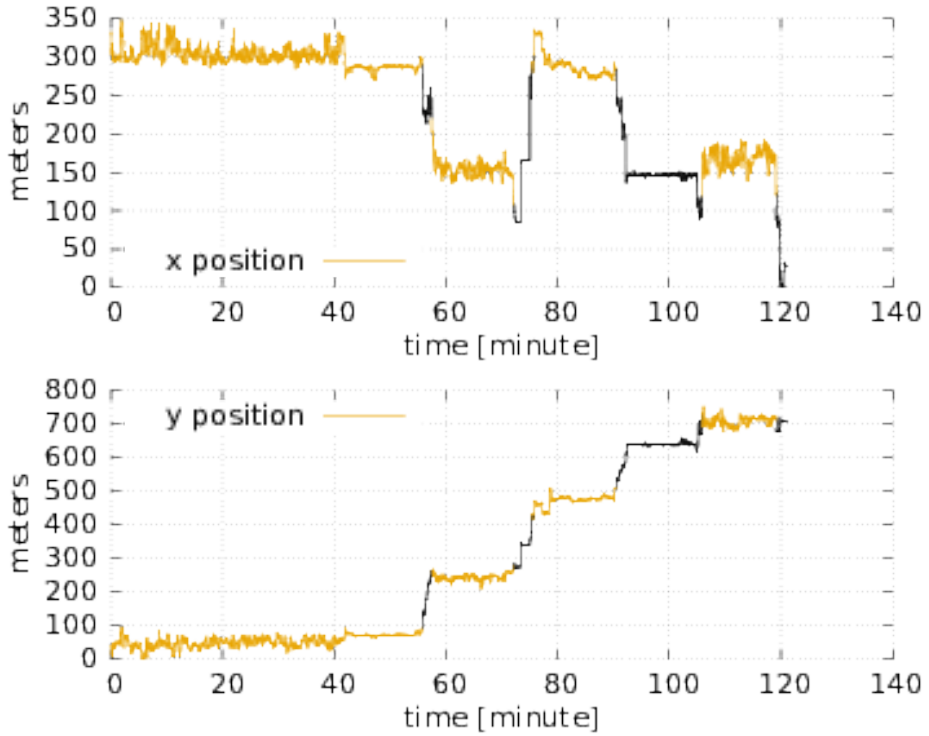


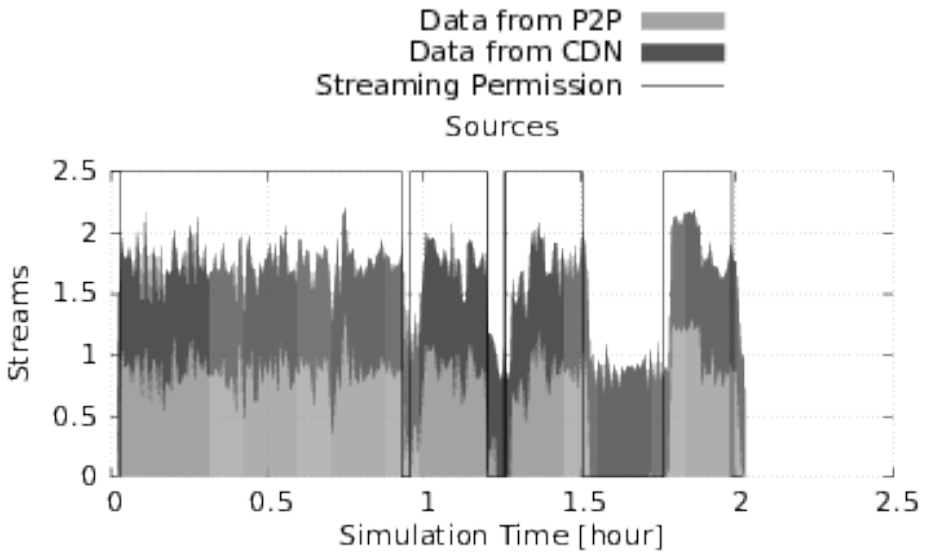
Movement in time

(from bottom-right to top-left)

- Tracing regular activity of a volunteer
  - 2+ hours long
- Clustering algorithm was parametrized and applied
  - 9 “relevant” locations
- Arbitrary selection of 4 areas as *Allowed*
  - Access points marked accordingly

Permission in Time





Simulation parameters

How does it behave?

- What impact on user experience?
  - Streaming from source
  - Streaming from a peer
- How soon is the enforcement applied?
  - Same cases

Evaluation in different setting:

- **nolbac**: Original system (for comparison)
- **perlbac**: Permissive (checks are done, but all Locations are allowed)

  (for user experience)
- **lbac**: Enforcing (selected locations are forbidden)

Simulation Parameters

Available settings for LBAC:

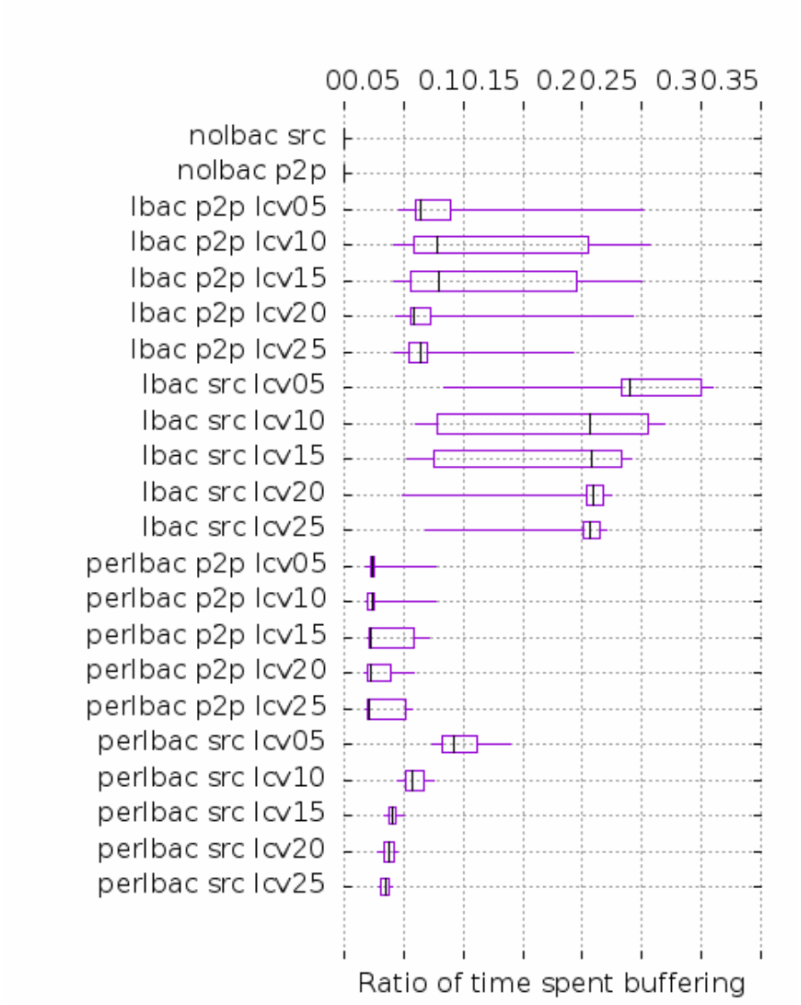
- **P**olicy **C**ertification **V**alidity
- **L**ocation **C**ertification **V**alidity

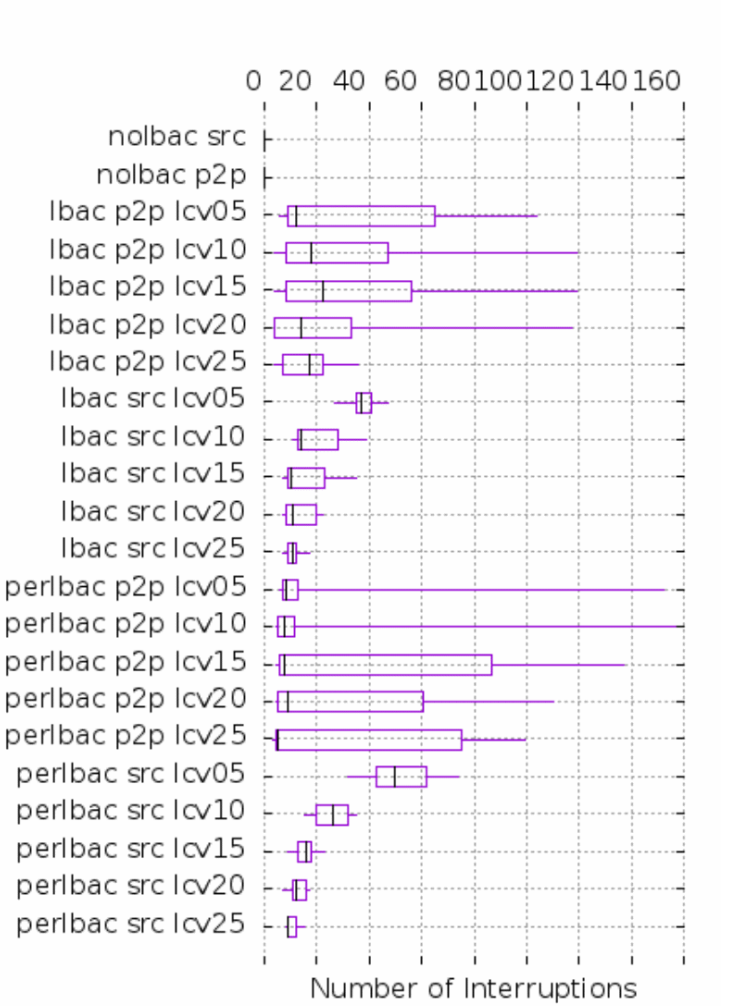
Expiration of certificates (intuitively)

- High → Lower disruptive effect on streaming
- Low → Faster reactions to system changes

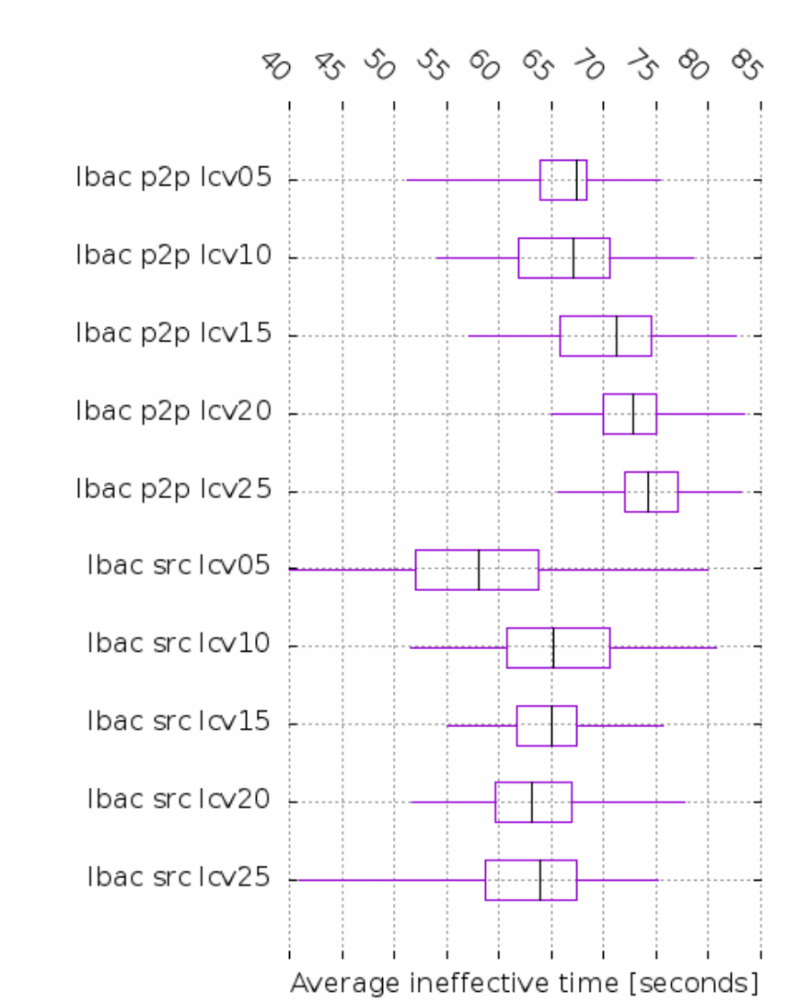
- PCV not really relevant
  - Probably we want higher values for it (as in hours or days)
- The disruptive effect of the shortest one (LCV) dominates

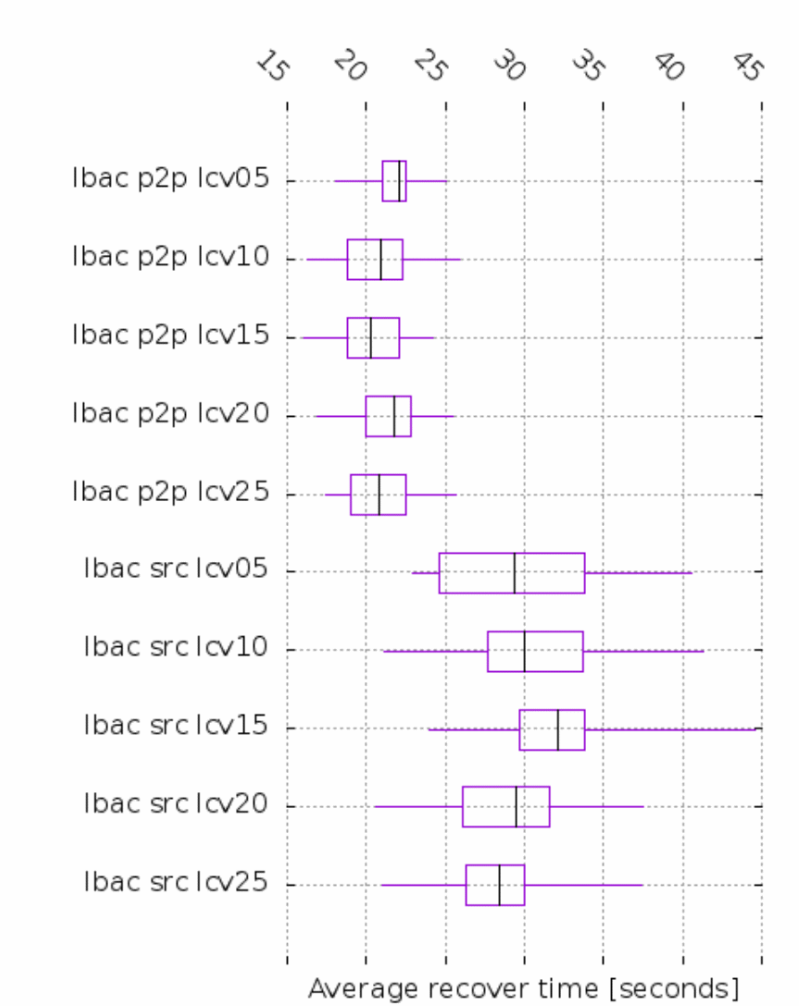
Results: user experience





Results: effectiveness





Results

- Trade-off between user experience and security
  - Although not a deal-breaker
- Streaming from source is more prone to disruption
  - A new credentials setup is always needed
  - Recommended a static node pre-fetching from an authorized area

Conclusions

Conclusions

- Status:
  - Paper is ready
  - Targeting SACMAT conference
    - Deadlines soon (beginning of February)

Thanks!

Q&A