

Information Security

Networking 1: Eve on the Wire

Winter 2022/2023



Jakob Heher, www.iaik.tugraz.at

he/his

What do *you* hope to see in this lecture?

Lecture ground rules

- We color technologies, algorithms, etc. for your convenience
 - State-of-the-art tech, no known vulnerabilities ✓
 - This is generally safe to use!
 - Outdated tech, known issues, covered for demonstration purposes ✗
 - You should not use this!
- Coloring provides a very quick-and-dirty categorization for you
 - Want to know *why*? That's what the lecture is for 😊

Meet the players



Alice
she/hers



Bob
he/his



Eve
????

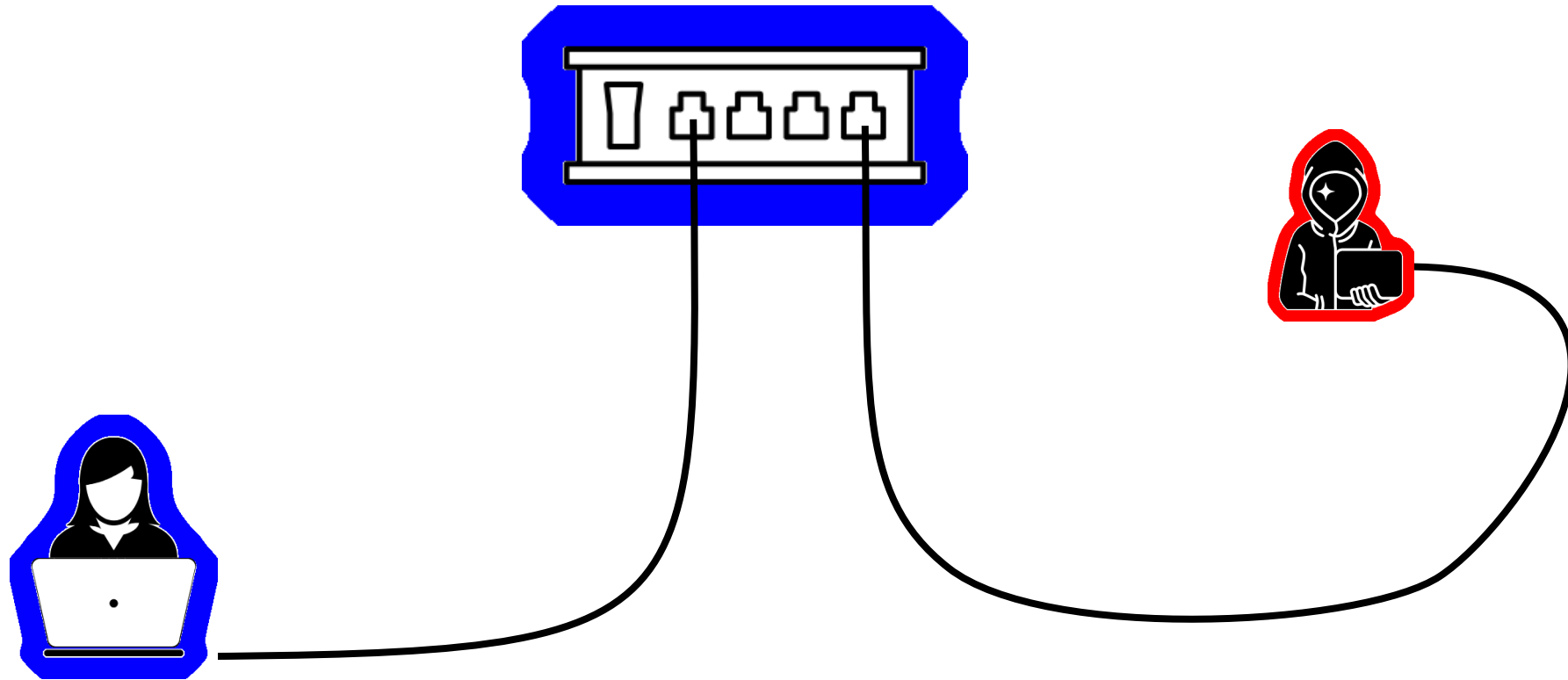


Smith
she/hers

Recall: Computer Organization and Networks

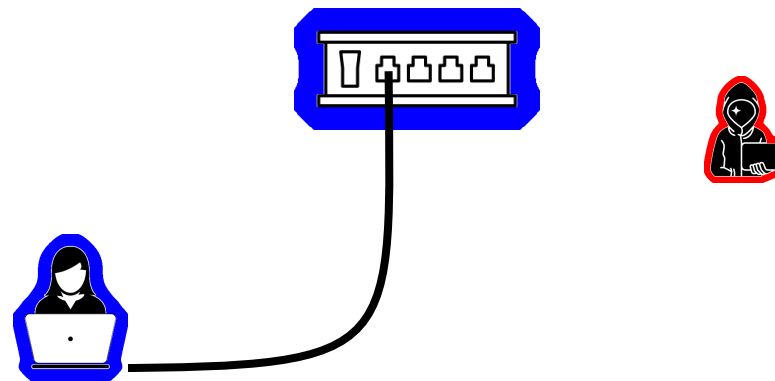
CON recap

- Different **Layers** divide tasks:
 - **Data Link Layer:** send data locally (Ethernet, Wi-Fi)
 - **Network Layer:** send data far away (IP)
 - **Transport Layer:** structured transport & multiplexing (TCP, UDP)
 - **Application Layer:** actual productive data (HTTP, DNS, SSH, NTP, BGP, ...)
- **Today's lecture:** what if CON, but everyone is evil



Data Link Layer: Ethernet

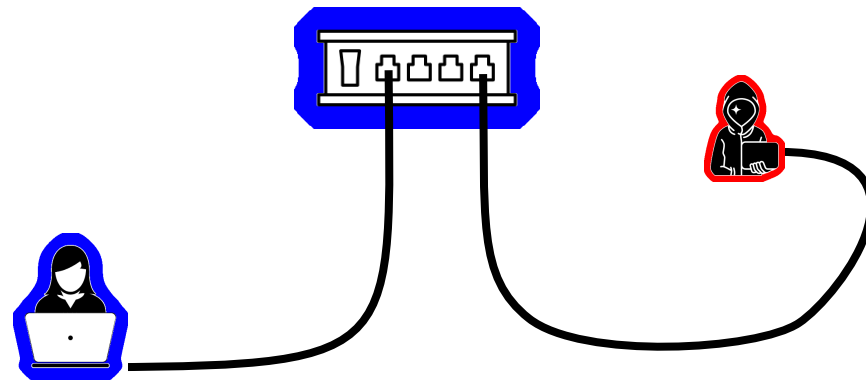
- Wired connections provide inherent physical security
 - Without a connection to the network, Eve can't do much!
- This is why workplaces don't let you plug in foreign devices
 - Your private laptop is a malware vector
- Going forward we'll assume Eve has a physical network connection





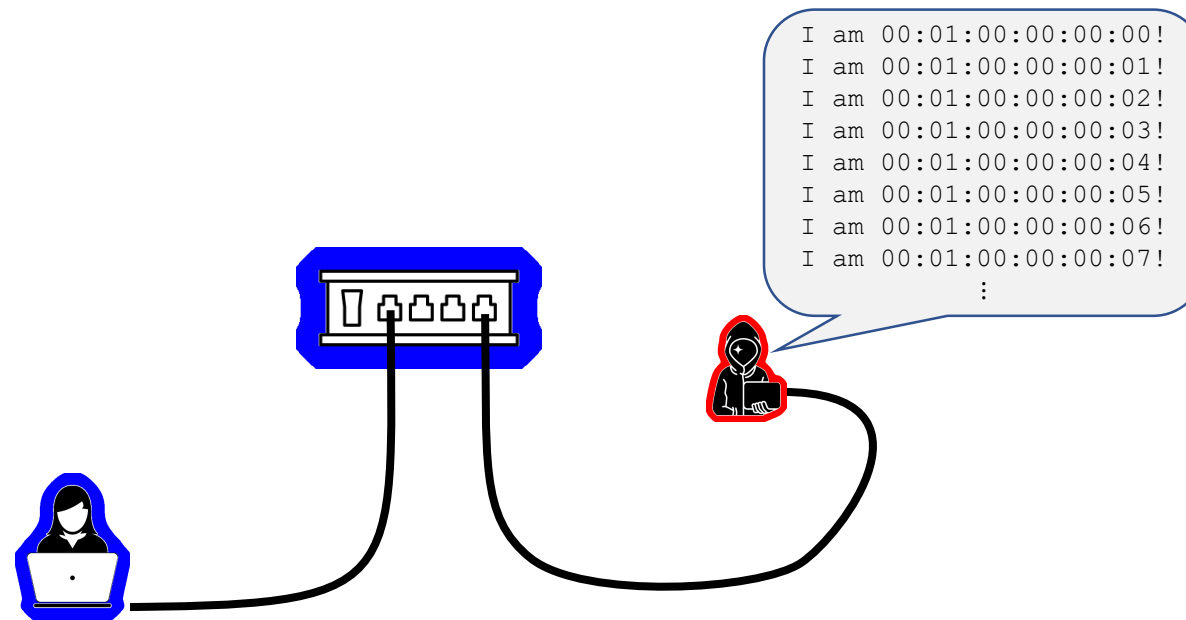
CON recap

- Ethernet *switches* learn a mapping of switch port to MAC address
- Unicast frames are only forwarded to the “correct” switch port



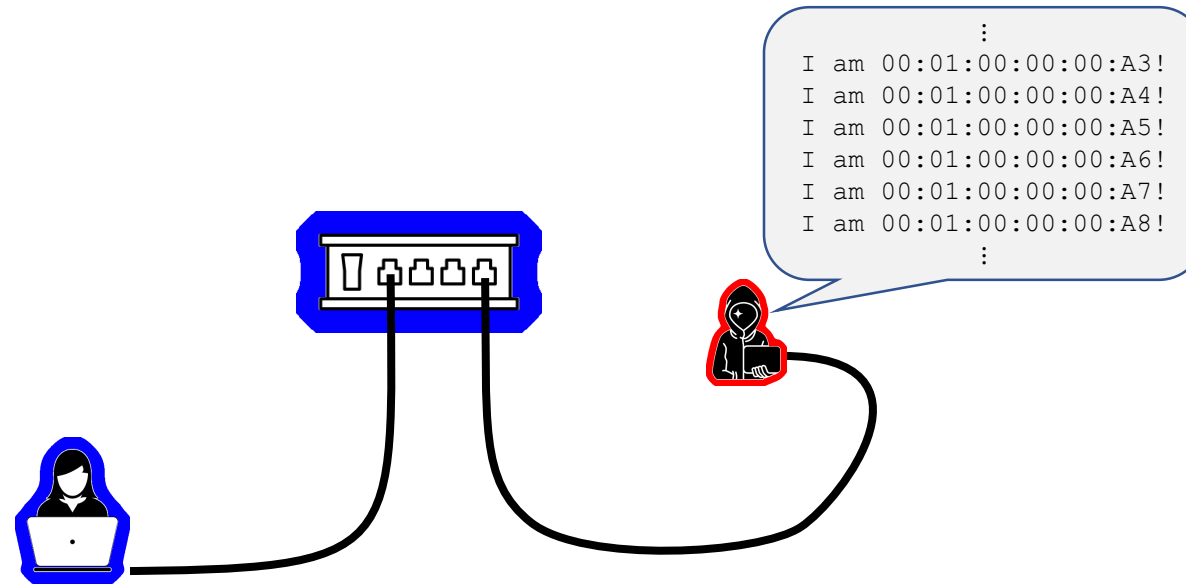
MAC flooding attack

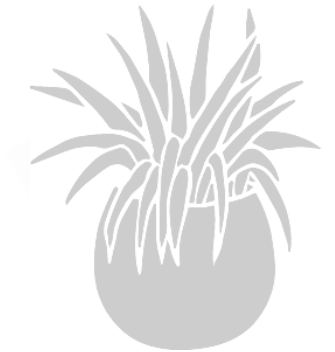
- Switches are embedded devices with limited memory
 - What happens if the switch runs out of space for its port \leftrightarrow MAC map?



MAC flooding attack

- Switches are embedded devices with limited memory
 - Many switches just drop the oldest entries when they run out of memory
 - After legitimate entries are dropped, unicast frames are flooded to all ports

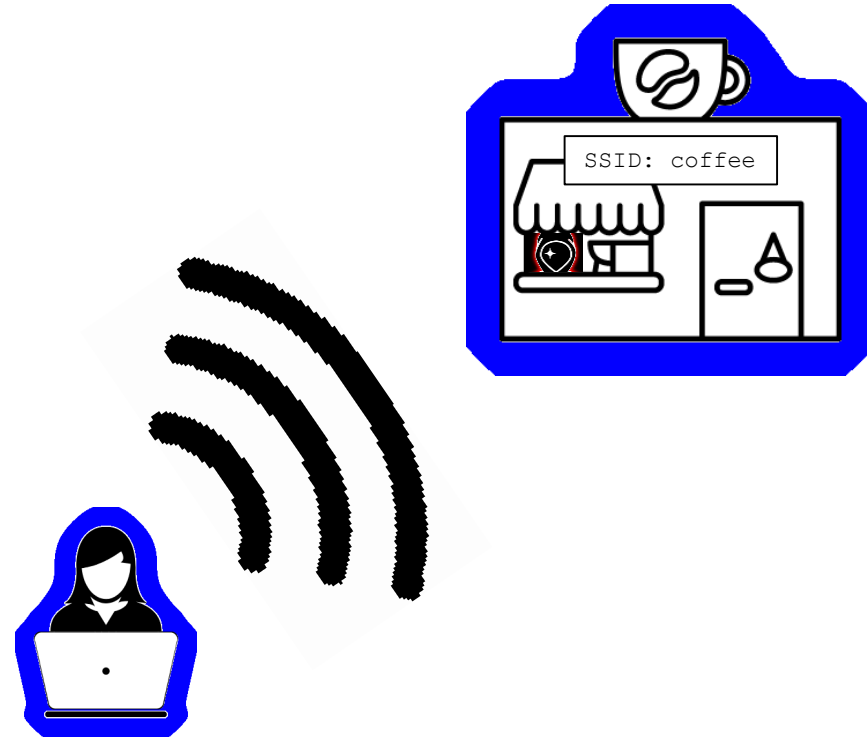




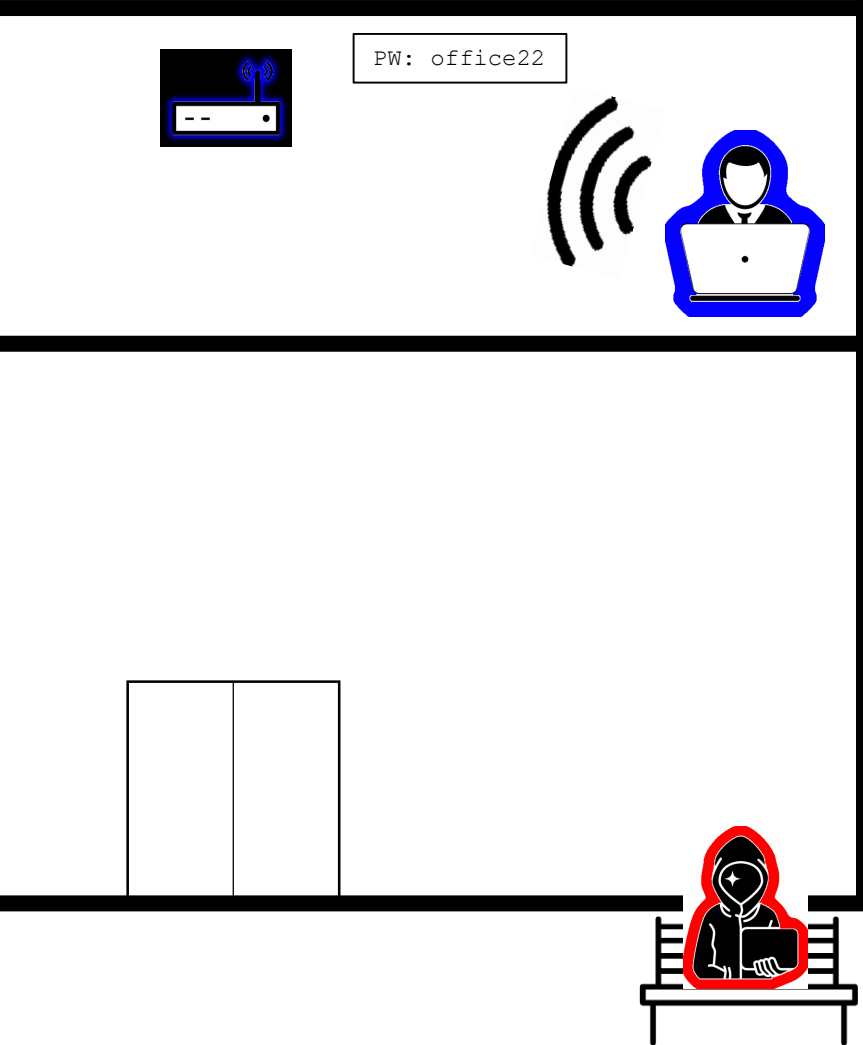
Data Link Layer: Wi-Fi



- Public Wi-Fi uses a shared medium
 - Every client can listen to all packets



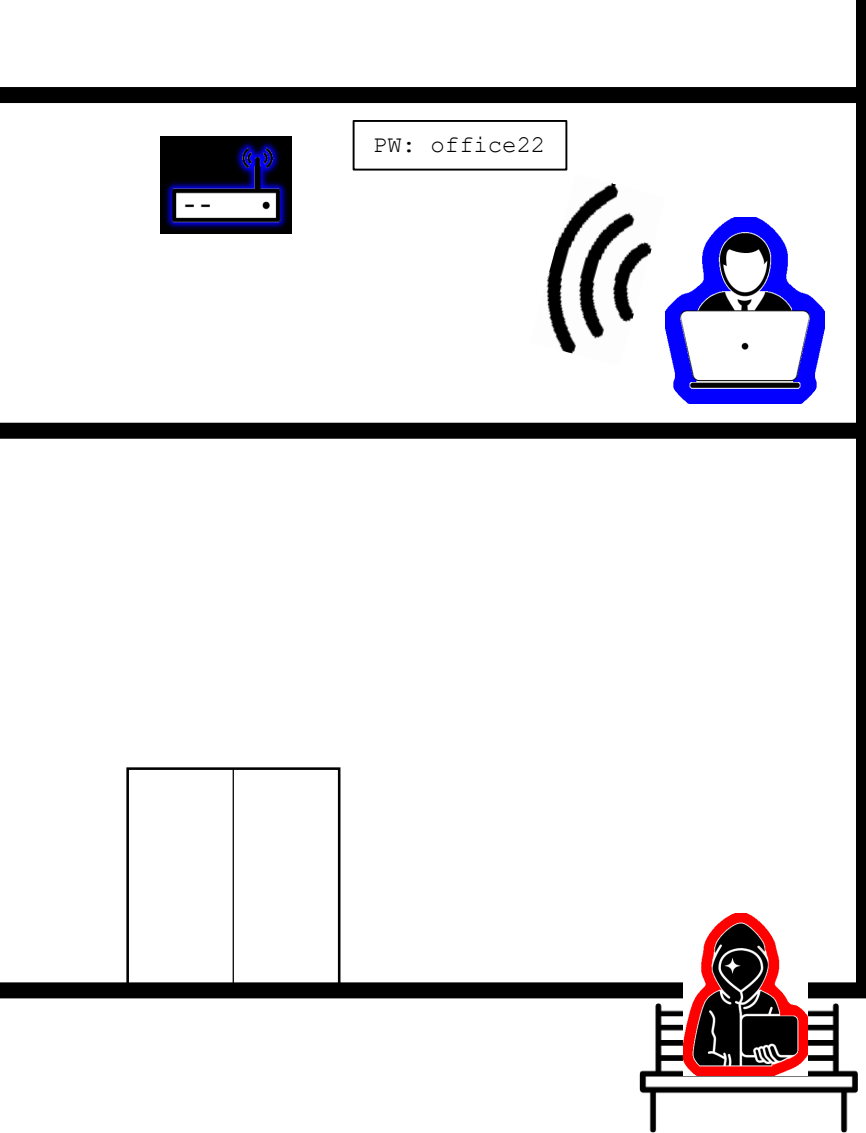
- Public Wi-Fi uses a shared medium
 - Every client can listen to all packets



- What if the network isn't public?
 - Packets are encrypted!

- What standard is used?
 - **WEP X**
 - Broken since 2001
 - **WPA-TKIP X**
 - Broken since 2015
 - **WPA2 ✓**
 - Crypto not broken
 - Available on all modern devices
 - **WPA3 ✓**
 - Crypto not broken
 - Improved protocol-level security



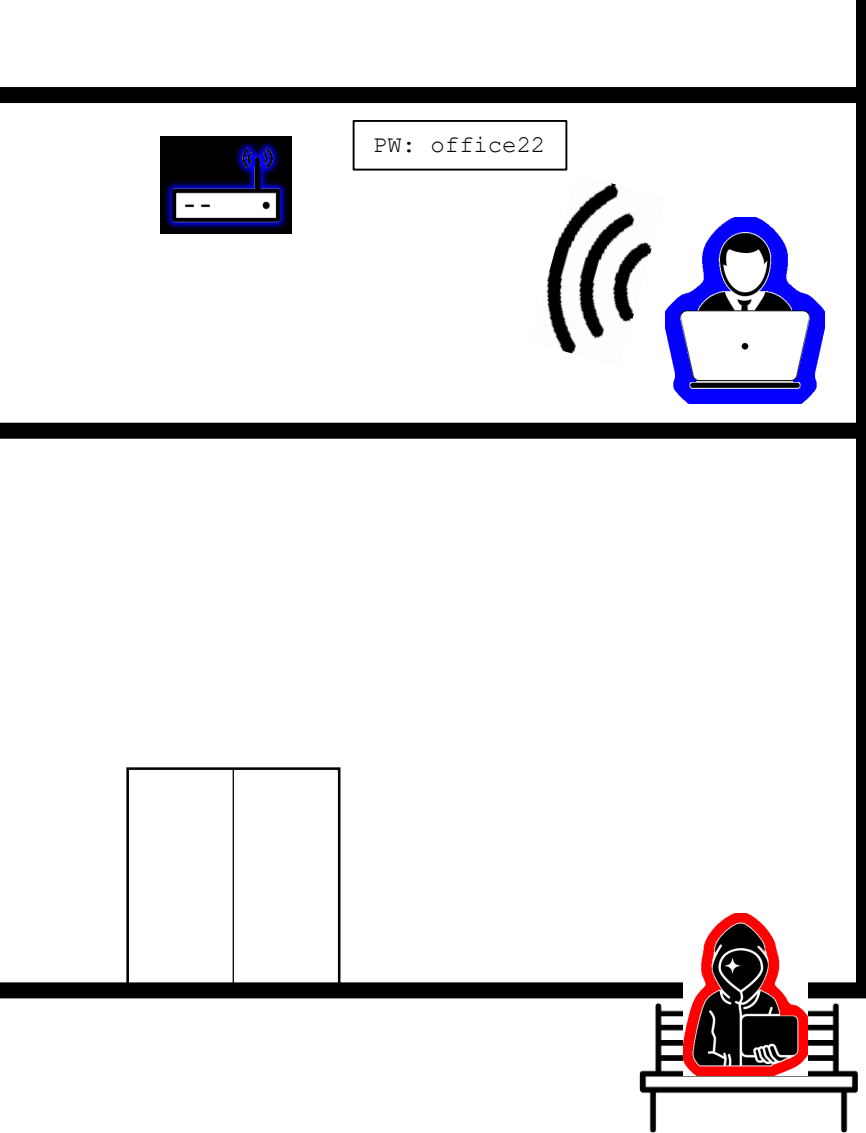


WPA2-PSK security

- WPA2 using **Pre-Shared Key**

 - Using a network password
 - Alternative: WPA2-Enterprise
 - Uses a dedicated authentication server

- Most devices don't support WPA3 yet
 - WPA2-PSK might be the best you can get



WPA2-PSK security

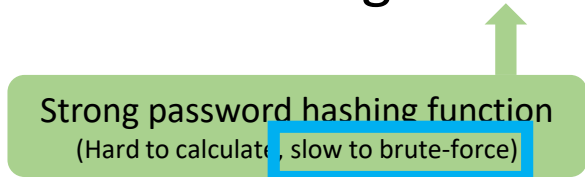
- The good:
 - Traffic inaccessible without password
- The bad:
 - Password can be brute-forced offline
 - After recording a genuine user's handshake
 - Connection control not authenticated
 - No forward secrecy
 - Weak per-user key derivation
 - If the master password is known

(a very reductive overview of)

WPA2-PSK crypto

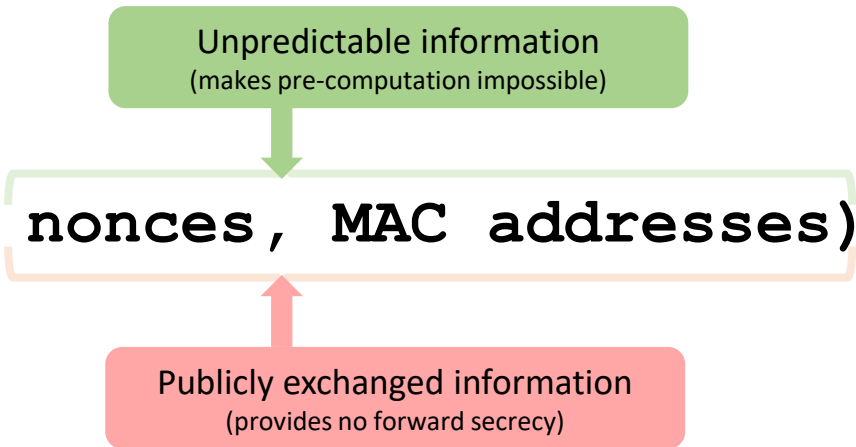
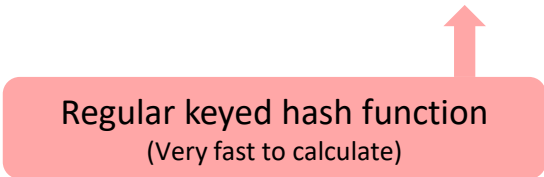
- **Pairwise Master Key** (global)

- Derived using **PBKDF2 (SSID, password)**



- **Pairwise Transient Key** (per-user)

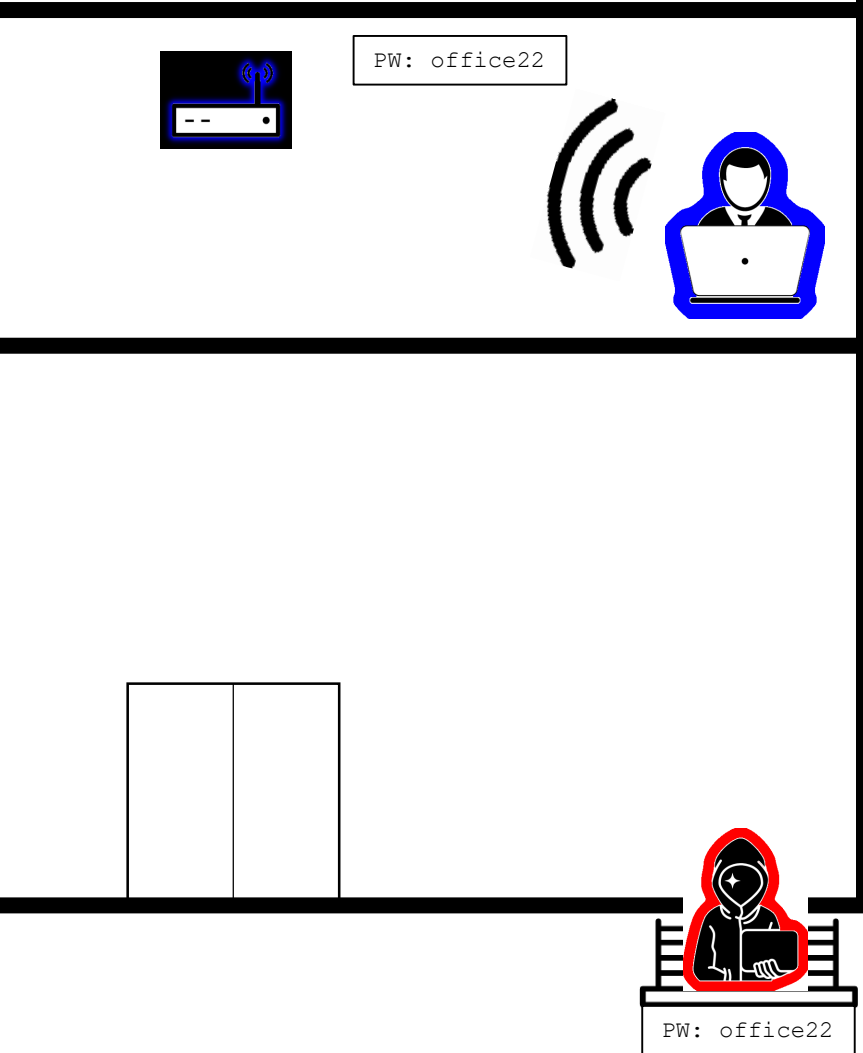
- Derived using **HMAC_SHA1 (PMK, nonces, MAC addresses)**



Attacking WPA2-PSK crypto

```
1 linksys
2 <no ssid>
3 default
4 NETGEAR
5 Wireless
6 WLAN
7 Belkin54g
8 MSHOME
9 home
10 hpsetup
11 smc
12 tsunami
13 ACTIONTEC
14 orange
15 USR8054
16 101
17 tmobile
18 <hidden ssid>
19 SpeedStream
20 linksys-g
21 3Com
22 WaveLAN Network
23 Wayport_Access
24 hhonors
25 pi07490509x
26 pi07490509x09
27 Motorola
28 SST-PR-1
```

- Verifying a password guess is slow...
- Verifying a **Pairwise Master Key** guess is fast!
 - PMK only depends on SSID + password
 - We can pre-calculate this!
- These *rainbow tables* already exist
 - 1 million common passwords
 - 1,000 common SSIDs
 - **≈ 33GB worth of PMKs**
- Solution: **Don't use a common SSID/password!** ✓



WPA2-PSK security

- What if Eve knows the password?

- Pairwise Transient Key = HMAC_ SHA1 (PMK, nonces, MACs)

Derived from SSID & password

Publicly available

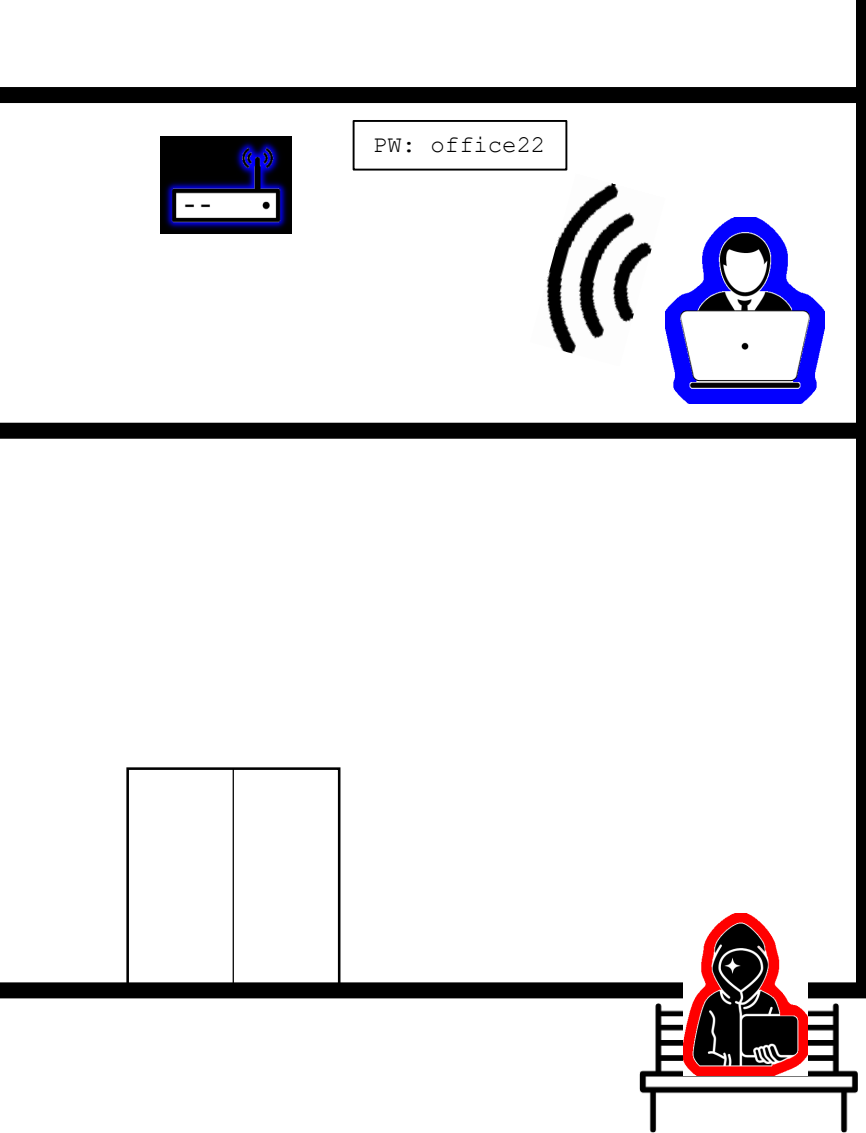
Exchanged in plaintext during handshake

- Eve can passively read all exchanged data!



WPA2-PSK security

- Eve wants to observe a handshake
- WPA2 does not encrypt control frames!
 - Eve can de-authenticate Bob
 - Bob's computer will then typically re-connect
 - Bingo!



WPA3 security

- The good:
 - Traffic inaccessible without password
 - Password cannot be attacked offline
 - Authenticated connection control frames
 - Forward secrecy is provided
 - Strong per-user key derivation
- The bad:
 - Not widely available (yet!)

- Do you turn off Wi-Fi on your phone when you go out?

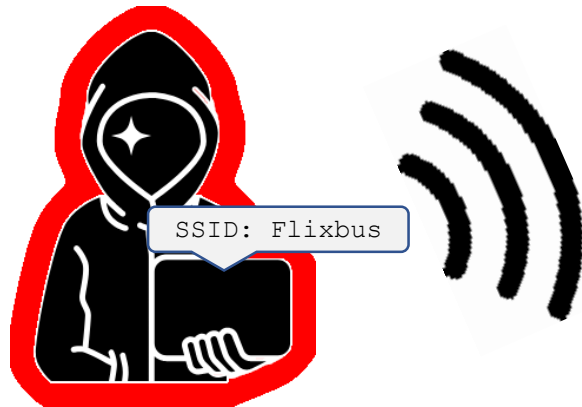
or ÖBB trains

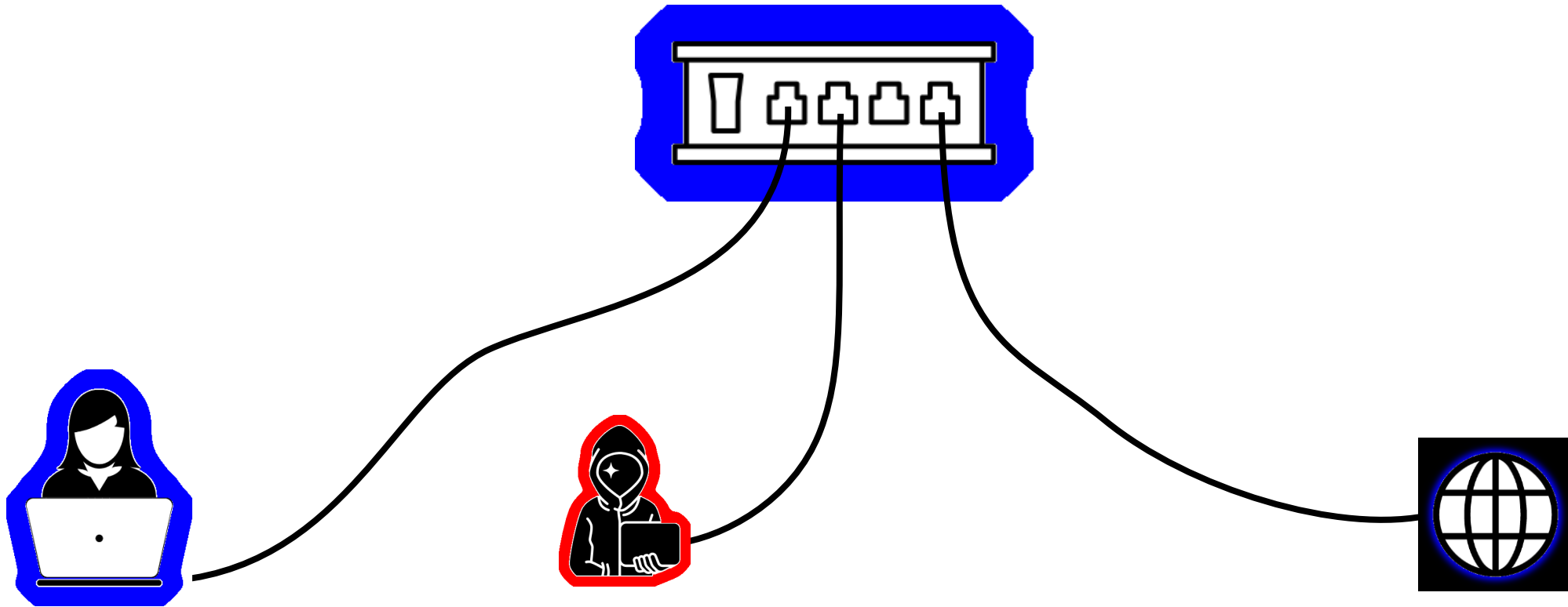


- Have you ever been on Flixbus & used Wi-Fi there?

or *OEBB*, for the train-minded

- What happens if Eve sets up a Wi-Fi AP with SSID *Flixbus* near you?
- Your phone will automatically use it to connect to the internet
 - You probably won't notice
 - Eve can read all the packets along the way!



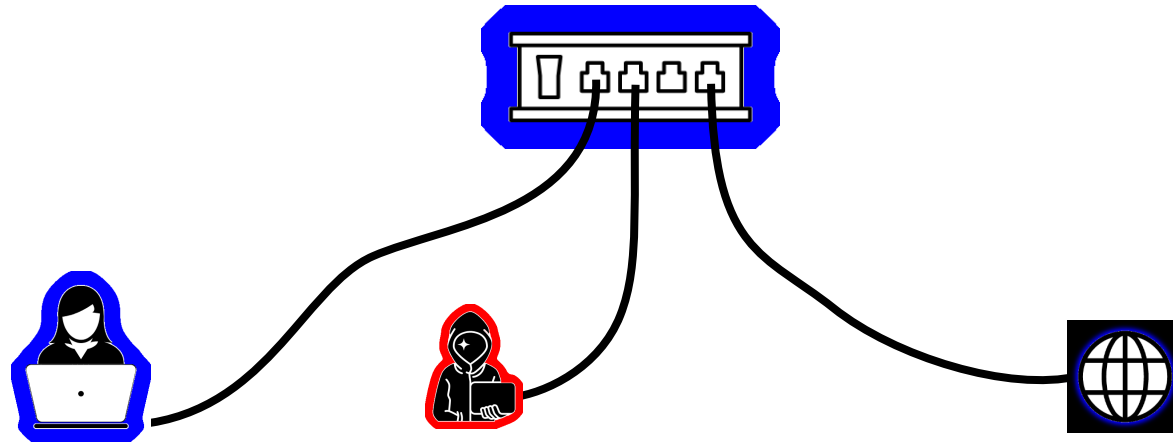


The Internet Layer

(in a local network)

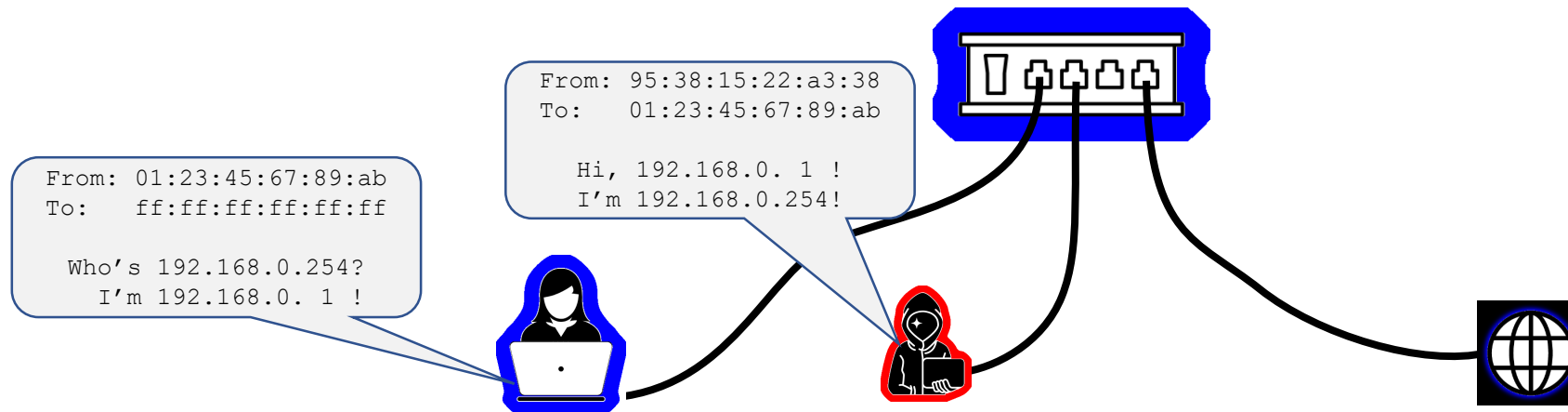
CON recap

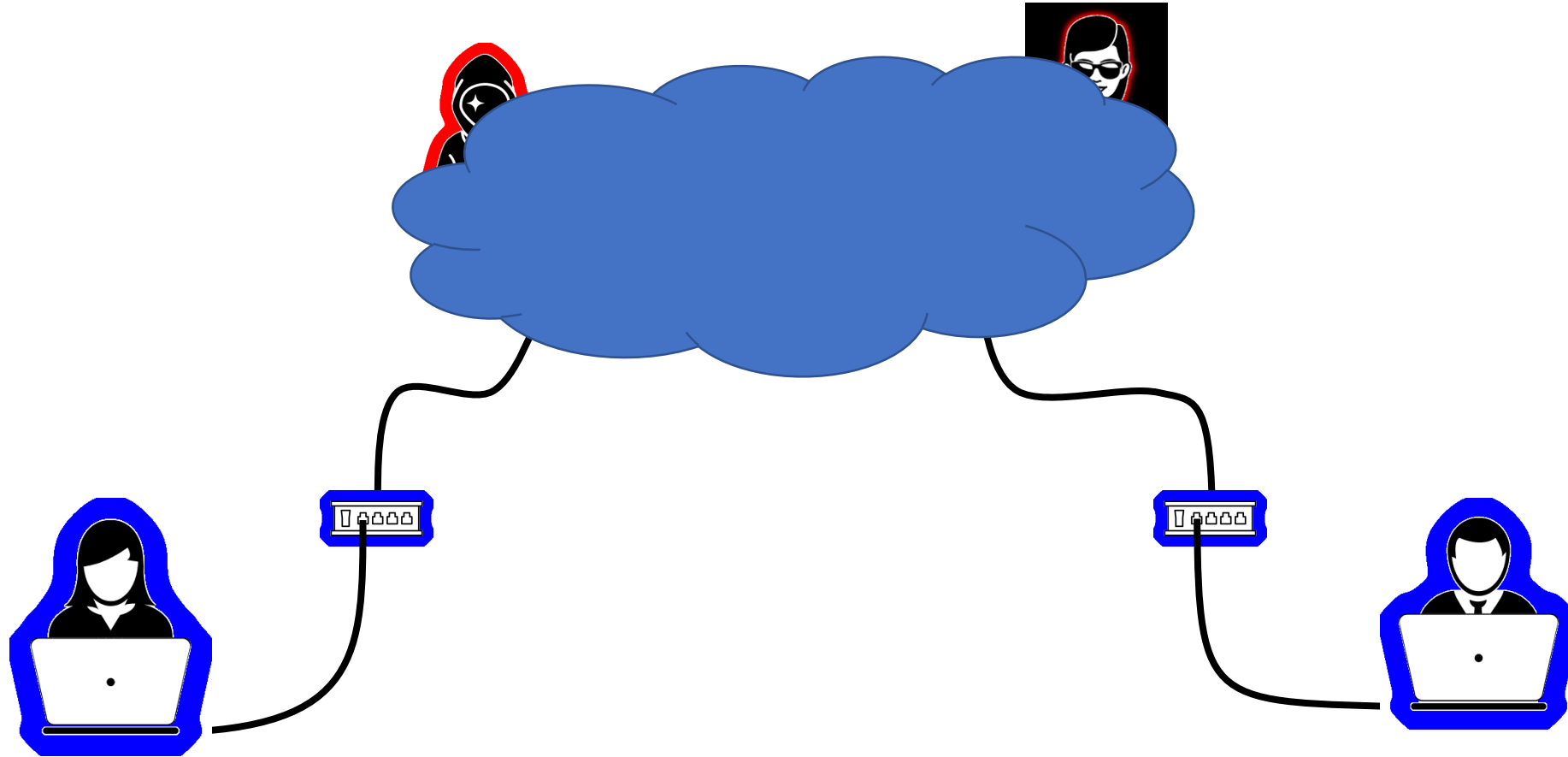
- **Address Resolution Protocol**
 - Translates Network Layer (IP) addresses to Data Link Layer (MAC) addresses
- Simple stateless query-response protocol
 - ARP request to Link Layer broadcast address
 - ARP reply from the host with the desired IP address



ARP spoofing attack

- The Address Resolution Protocol does not provide authentication
 - Eve can mislead both sides of the IP traffic
 - Eve forwards packets to the correct MAC address afterwards





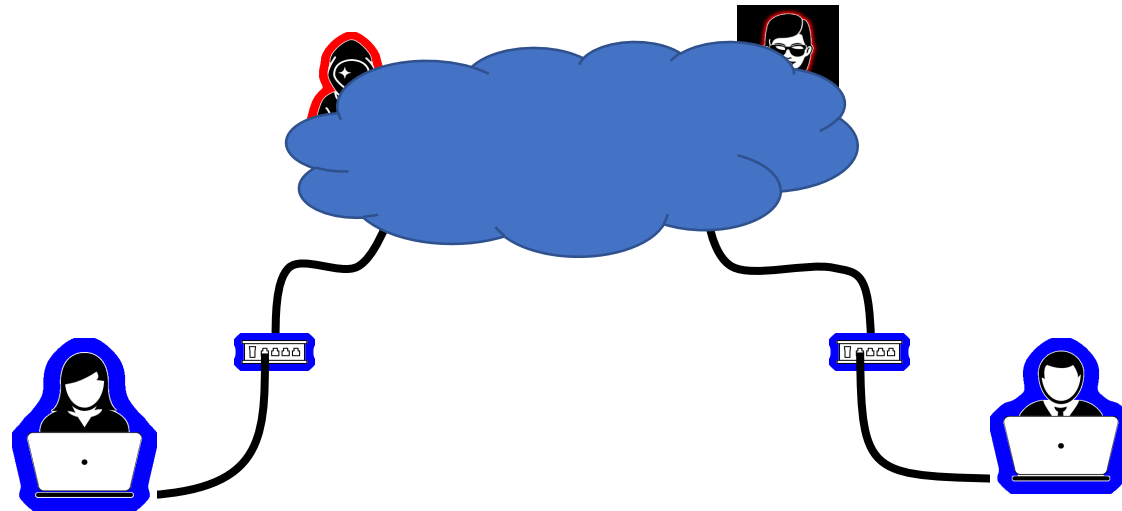
The Internet Layer

(on a global scale)

Who can see your data?

CON recap

- IP packets are passed onward by routers
 - Two packets to the same destination might take different routes
- Any router that forwards your packets can see your data!



Who can see your data?

- Any router that forwards your packets can see your data!

```
Tracing route to google.com [216.58.201.78]
over a maximum of 30 hops:
```

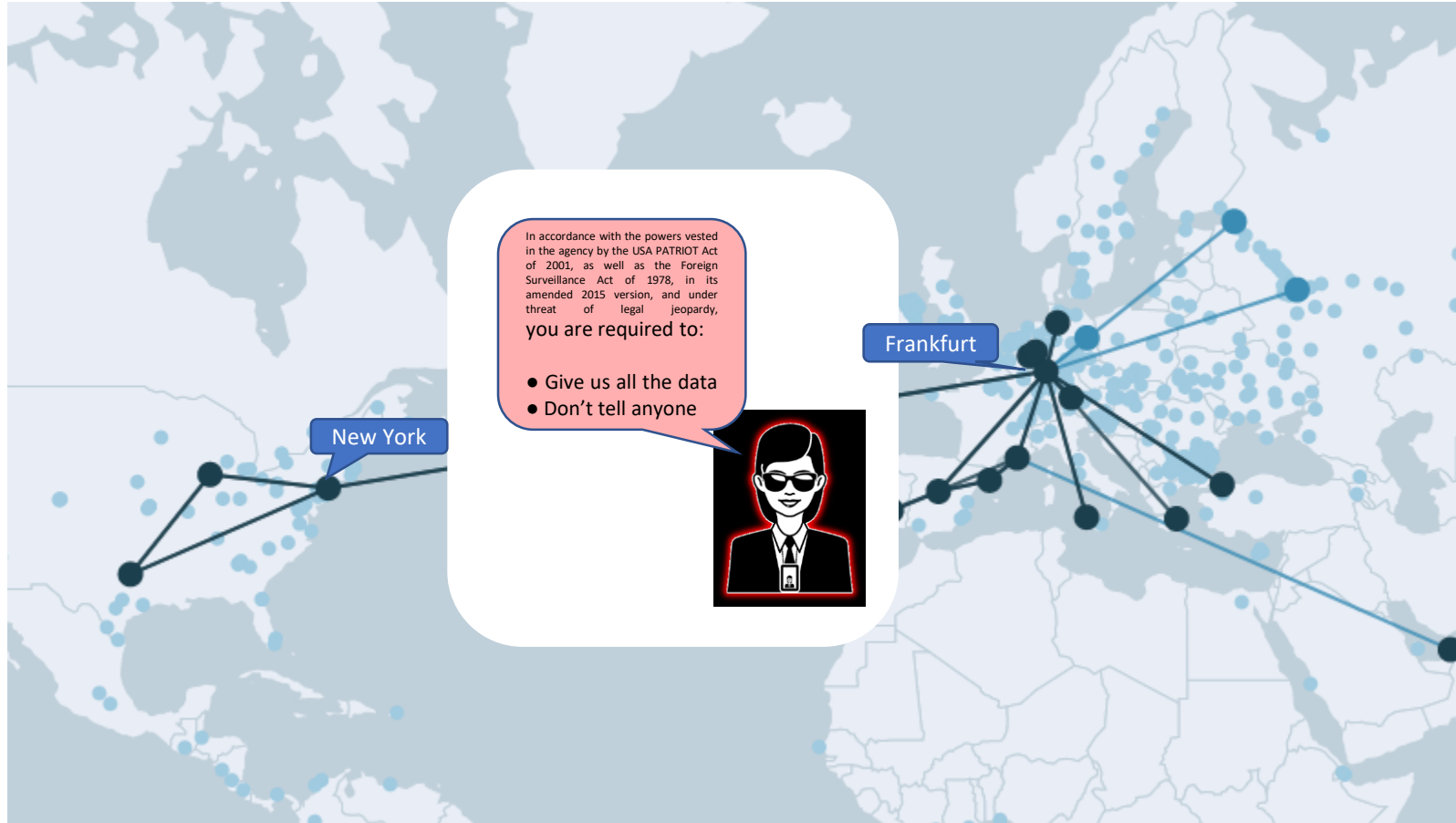
1	<1 ms	<1 ms	<1 ms	10.27.152.1	
2	<1 ms	<1 ms	<1 ms	129.27.200.161	TU Graz public address range
3	*	*	*	Request timed out.	
4	1 ms	1 ms	1 ms	graz1.aco.net [193.171.21.41]	Austrian research & education network
5	5 ms	4 ms	4 ms	195.113.179.150	Czech research & education network
6	14 ms	8 ms	10 ms	r98-bm.cesnet.cz [195.113.179.149]	
7	15 ms	14 ms	14 ms	195.113.235.109	
8	12 ms	12 ms	12 ms	r2-r93.cesnet.cz [195.113.157.70]	Google public address range
9	10 ms	10 ms	10 ms	172.253.50.255	
10	10 ms	10 ms	10 ms	108.170.236.229	
11	10 ms	10 ms	10 ms	prg03s01-in-f14.1e100.net [216.58.201.78]	

```
Trace complete.
```

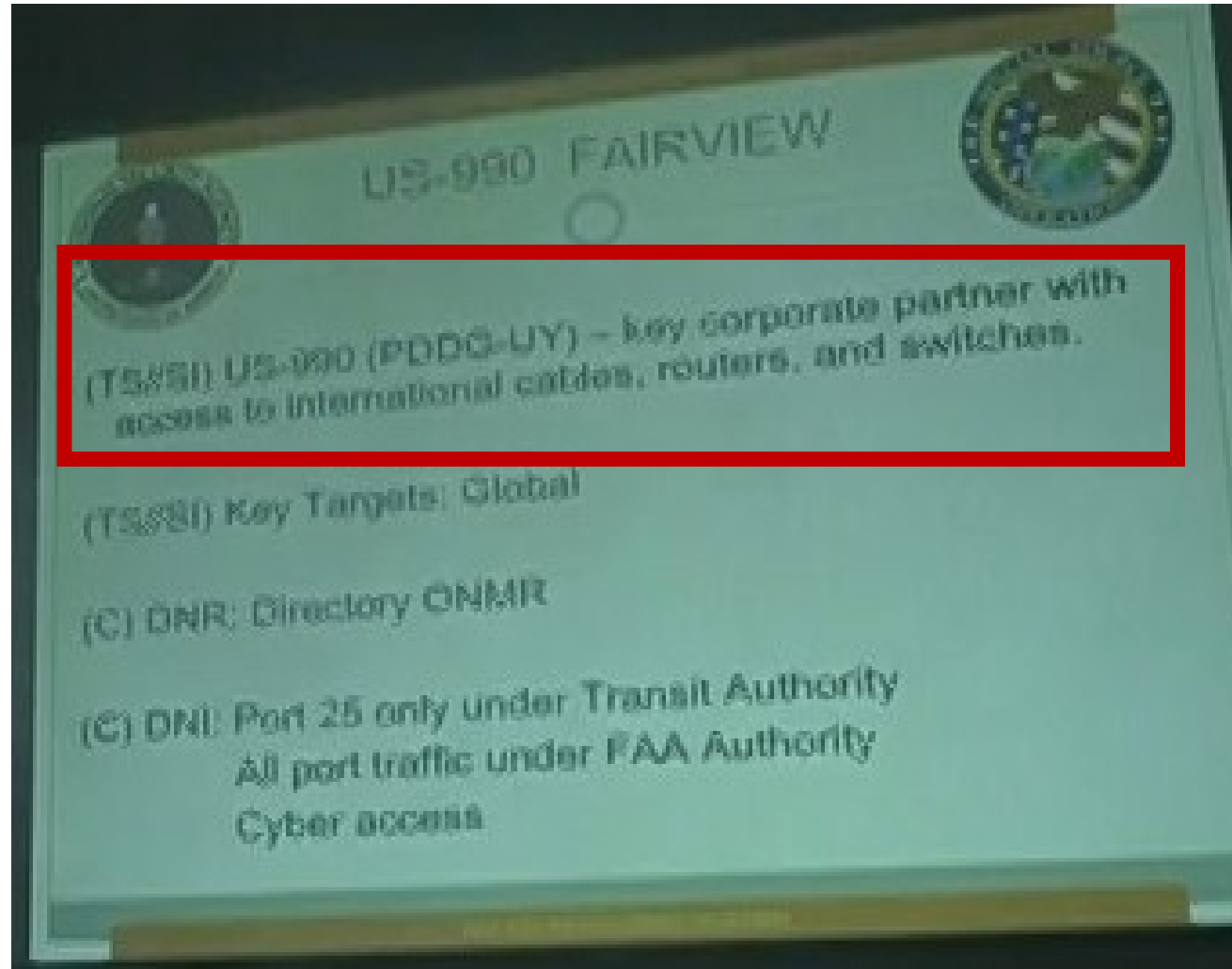
How is the internet interconnected?

- Direct peering
 - Very fast dedicated links to high-traffic destinations
 - Example: Czech research & education network <-> Google
 - 1.25 gigabytes/second
- Internet exchange points
 - Interconnect many participant networks
 - Example: DE-CIX
 - 10.4 terabytes/second between 2385 connected networks

How is the Internet interconnected?



But surely they wouldn't...



But surely not in the EU...

Frankfurter Allgemeine

ZEITUNG ● FAZ.NET

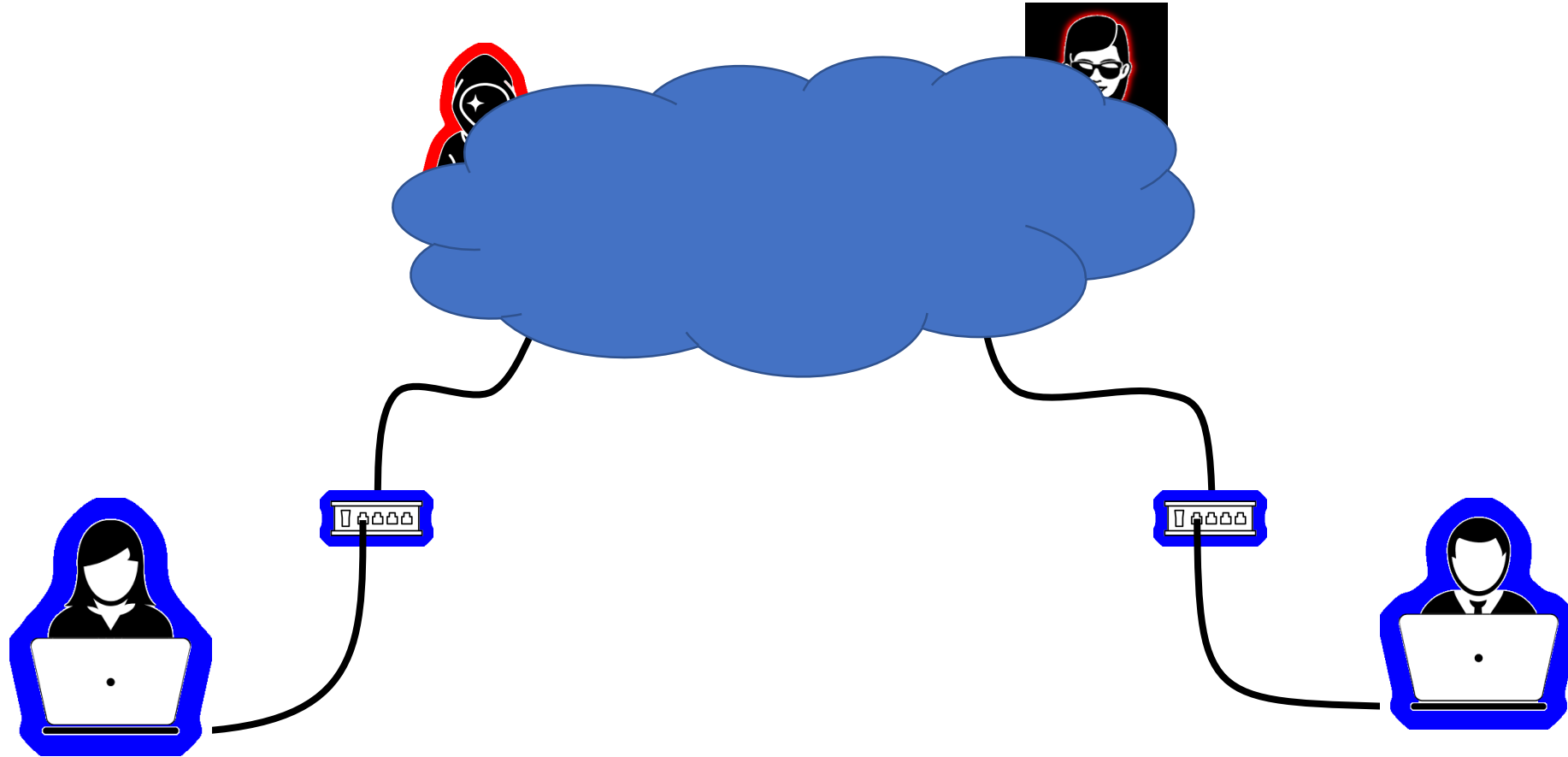
KLAGE VON DE-CIX

Nachrichtendienst darf weiter Daten von Internet-Knoten abzapfen

AKTUALISIERT AM 31.05.2018 - 08:17



Der Bundesnachrichtendienst bedient sich seit Jahren an den Daten, die den weltweit größten Internetknoten De-Cix durchlaufen. Ein Gerichtsurteil hat ihm die Möglichkeit nun bestätigt.

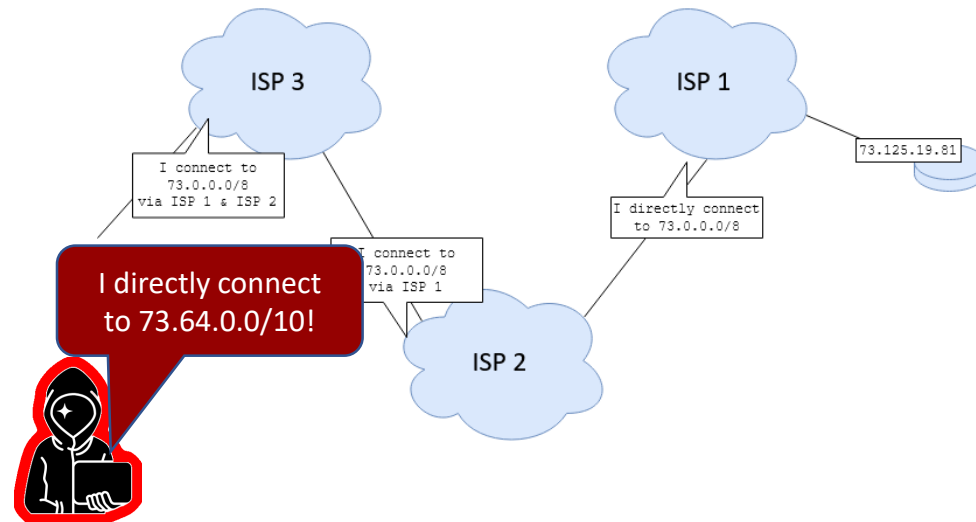


The Internet Layer

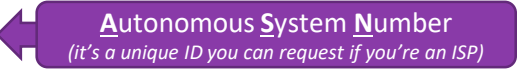
(more shenanigans on a global scale)

CON recap

- IP packets are passed onward by routers
 - Two packets to the same destination might take different routes
- The **B**order **G**ateway **P**rotocol lets **network providers** advertise routes
 - It's a big, collaborative, distributed shortest-path algorithm!

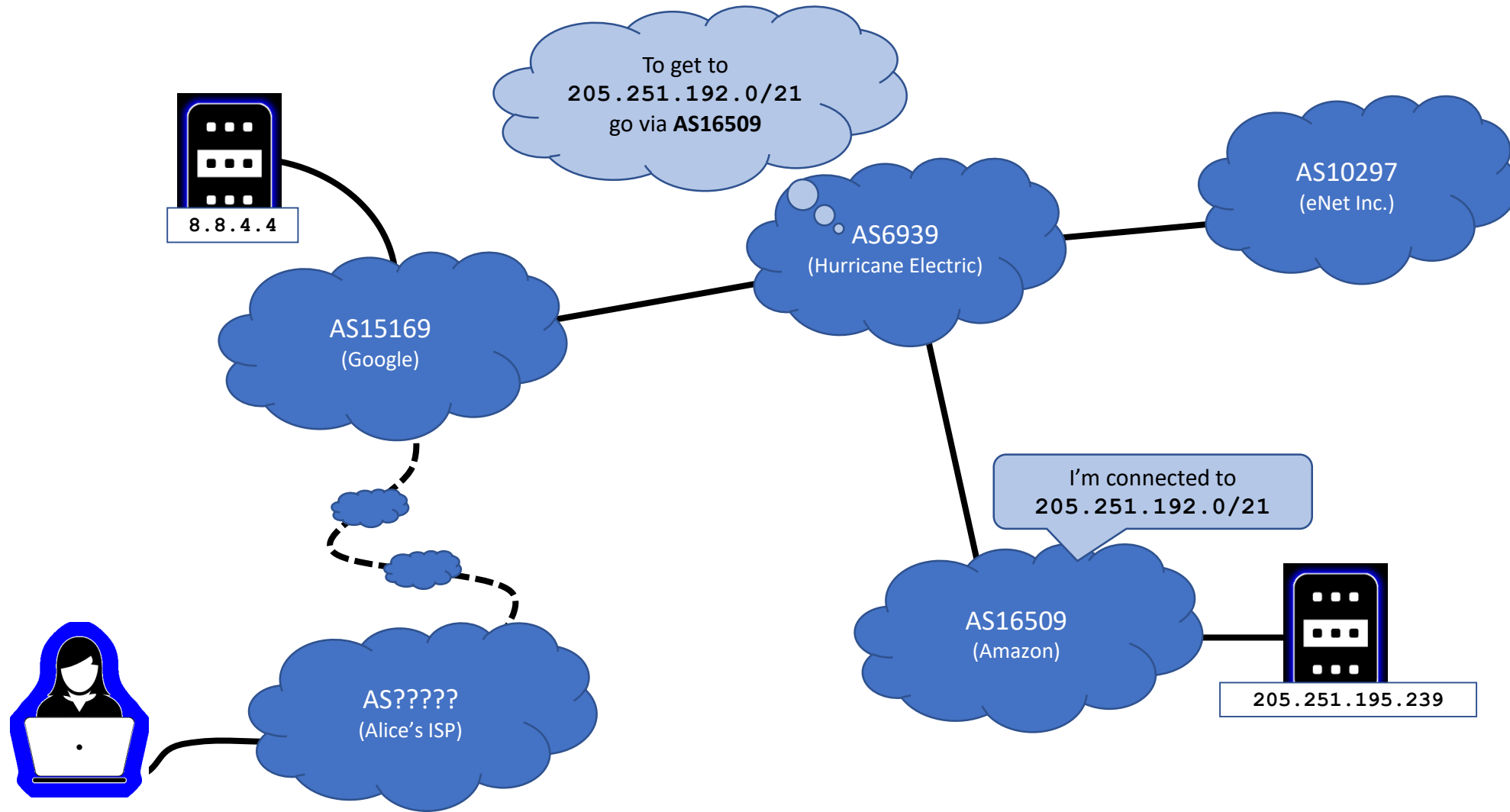


BGP hijacks

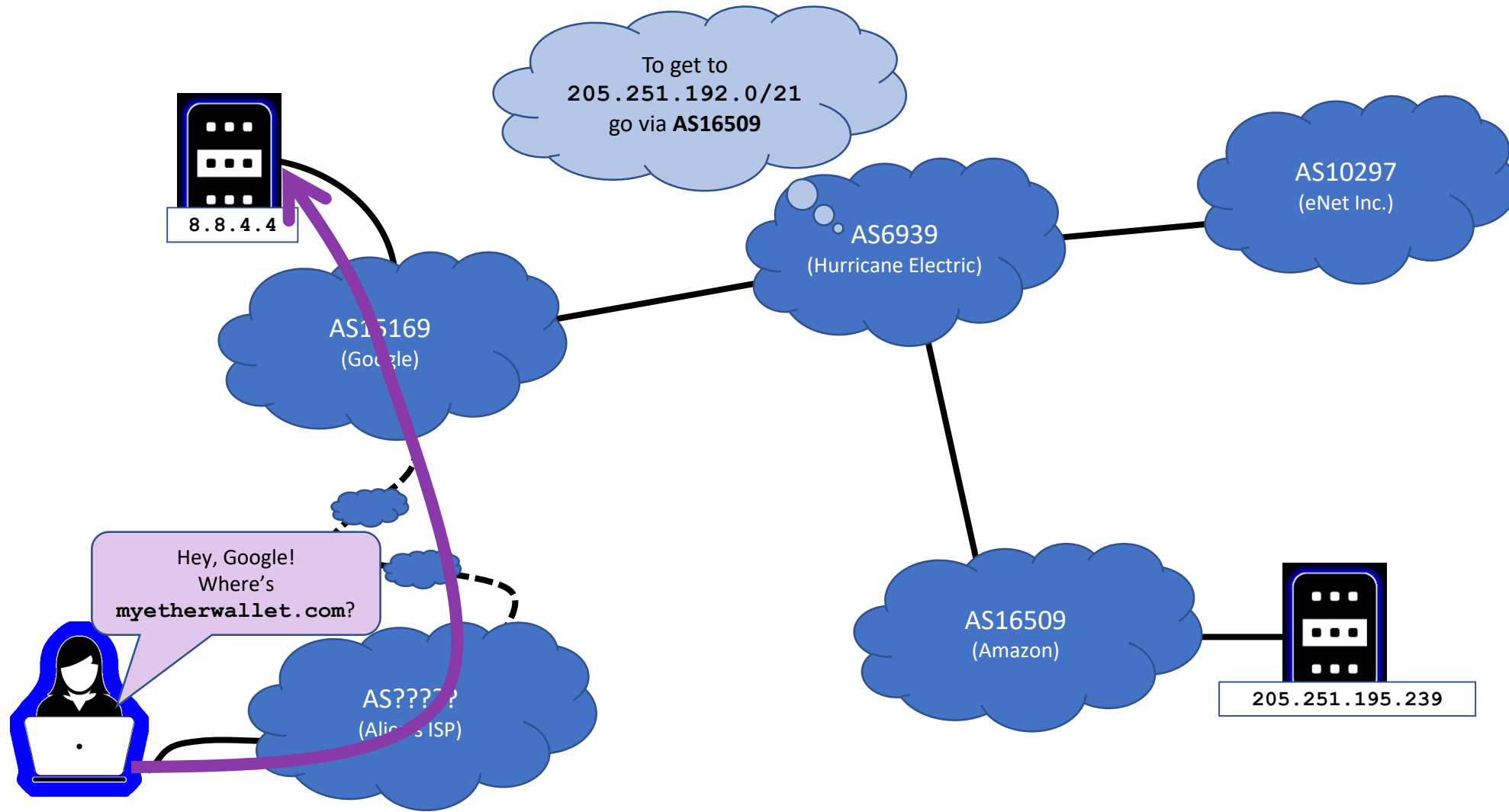
- Basic BGP assumes that connected ISPs are trustworthy
 - There are $\approx 100,000$ ASNs 
 - Even if all of them can be trusted – are all of them secure?
- Once a route is broadcast, it is picked up and forwarded
 - More specific routes are preferred

BGP hijacks – Example

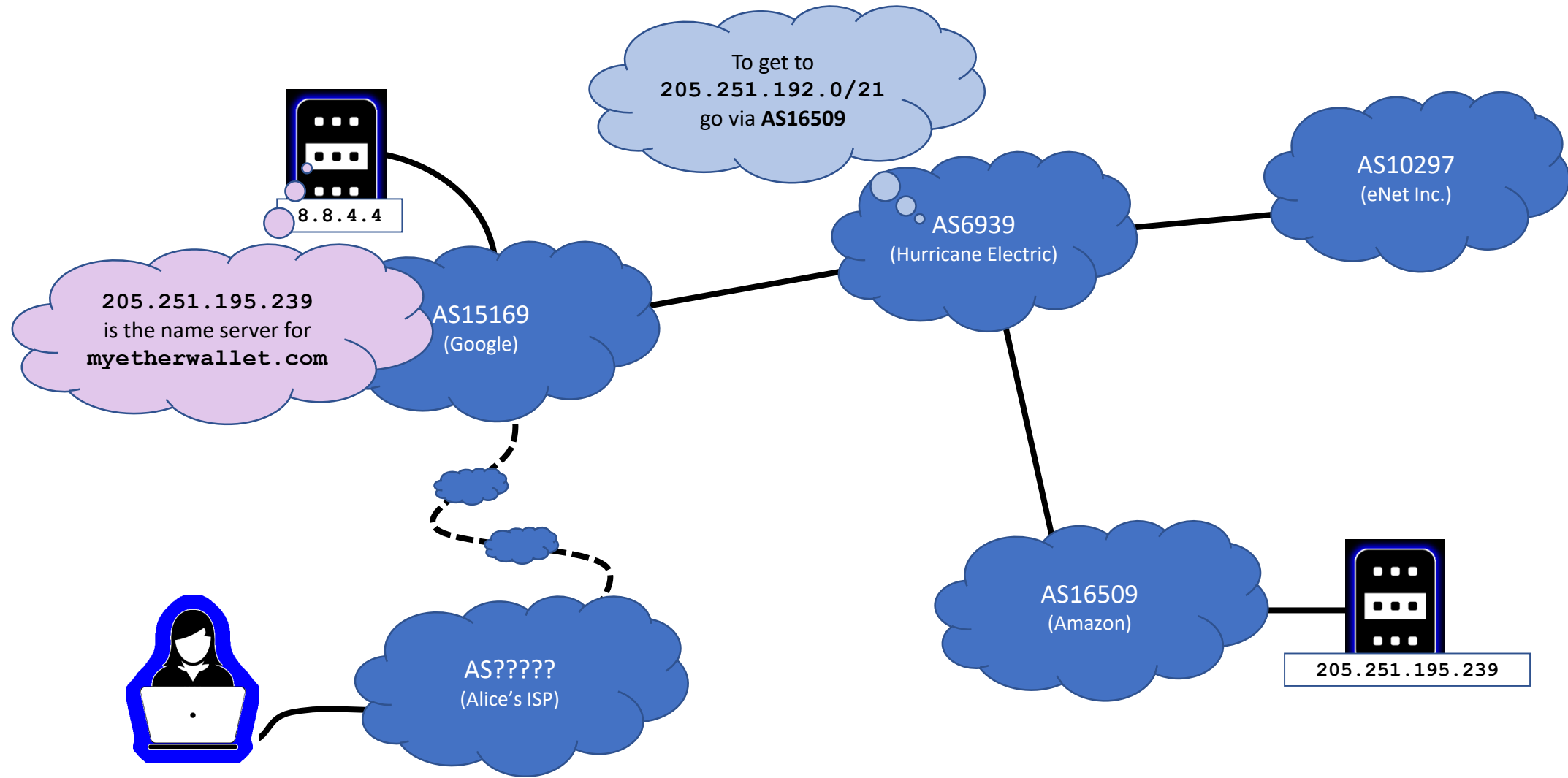
- **205.251.192.0/21** – Amazon Route53 DNS servers
 - On Apr 24 2018, AS10297 starts announcing **205.251.195.0/24**
 - This is more specific than the actual Amazon (AS16509) announcement
 - It makes its way via AS6939 to the wider internet...
- The hijack captures traffic directed to Amazon's DNS
 - Any requests for the IP of `myetherwallet.com` return fake results
 - Users are directed to a phishing site hosted in Ukraine...



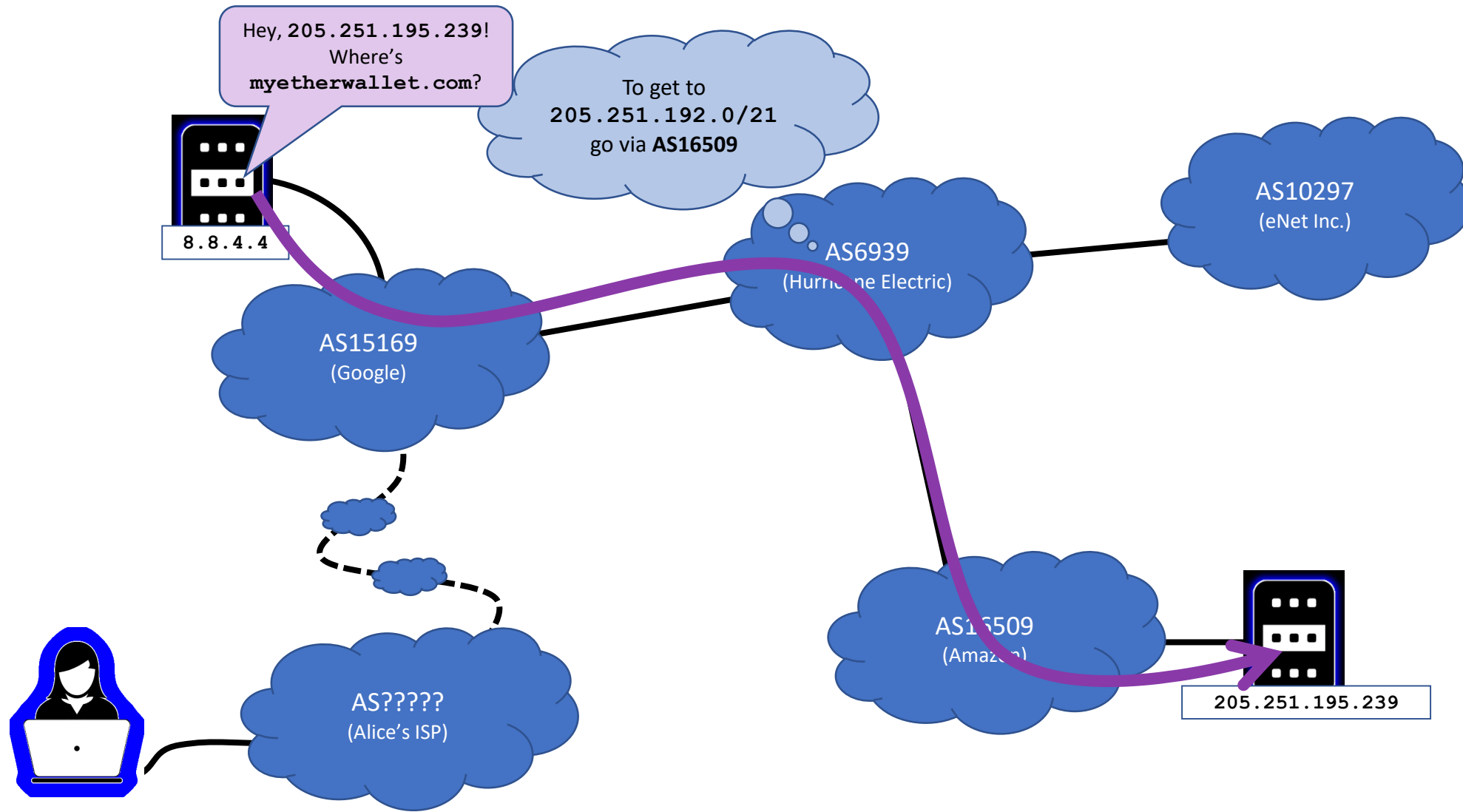
The Internet on any other day...



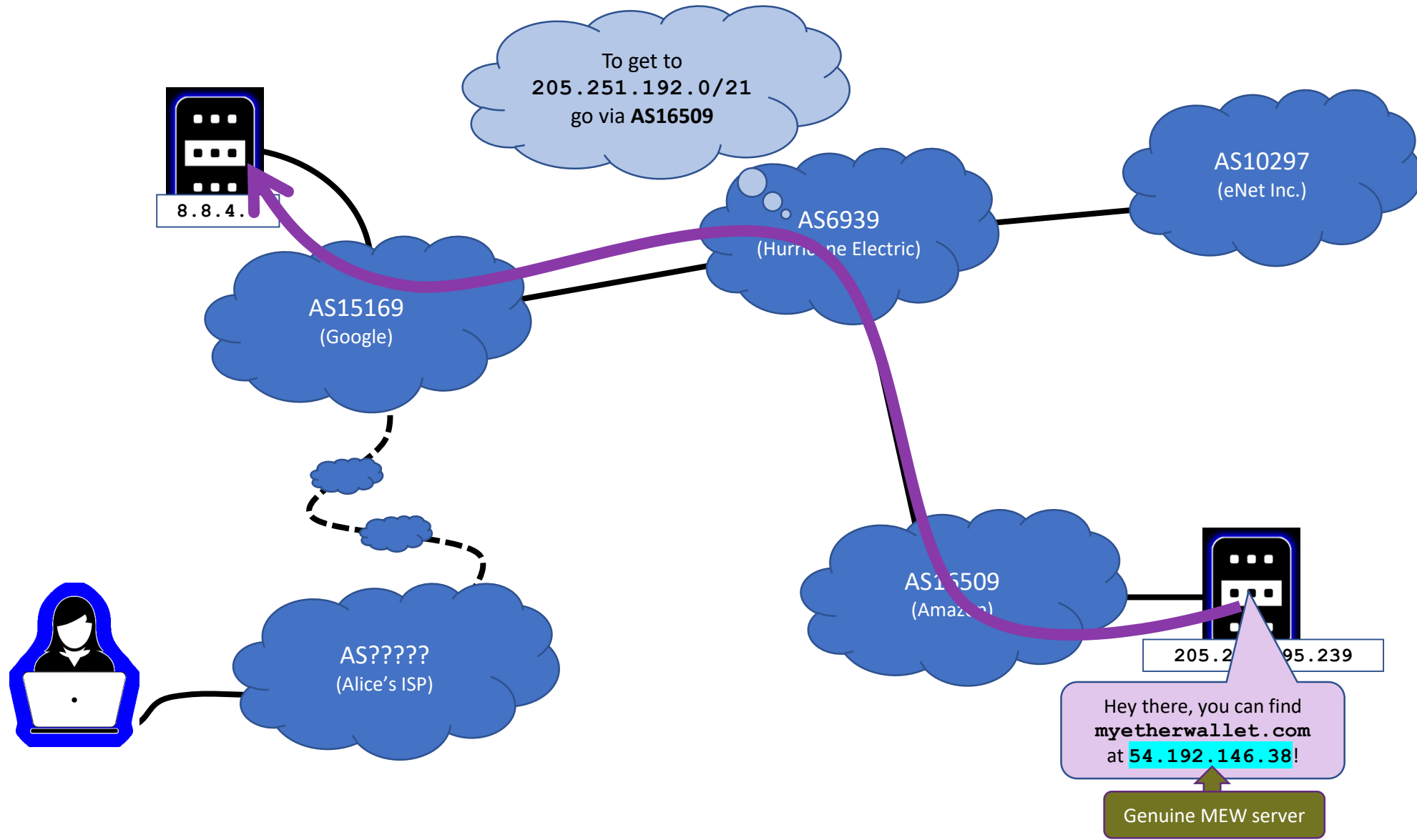
The Internet on any other day...



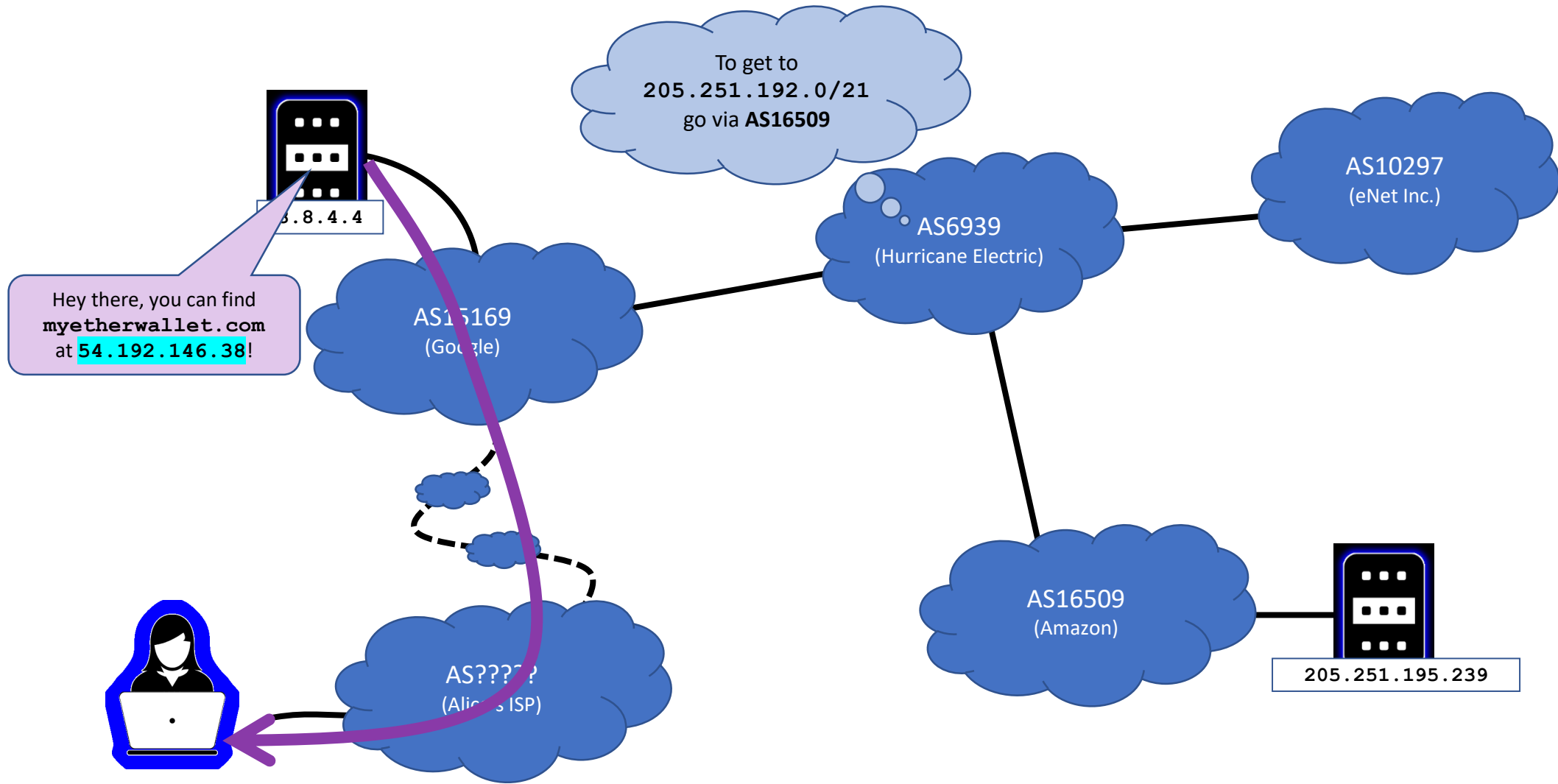
The Internet on any other day...



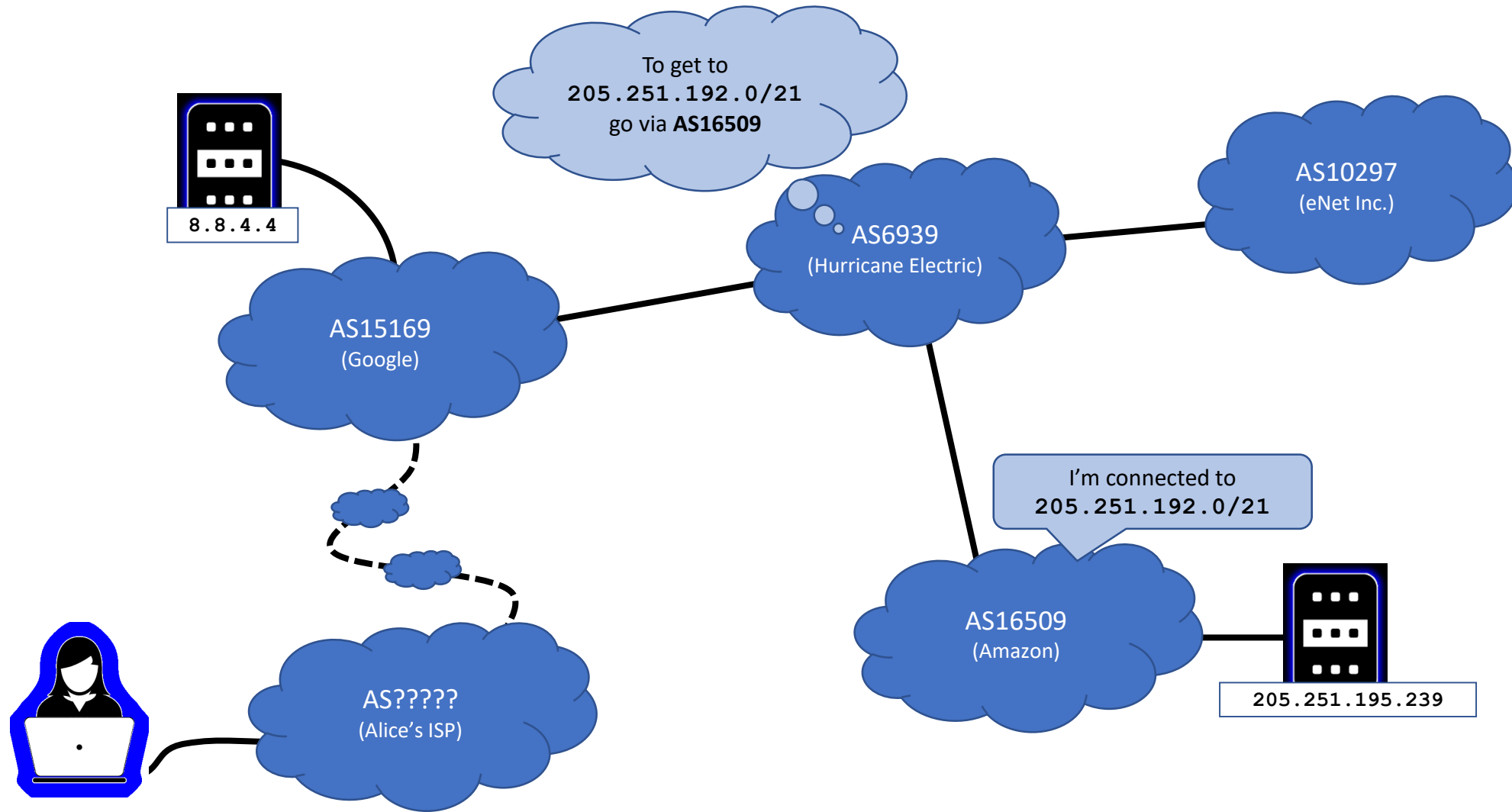
The Internet on any other day...



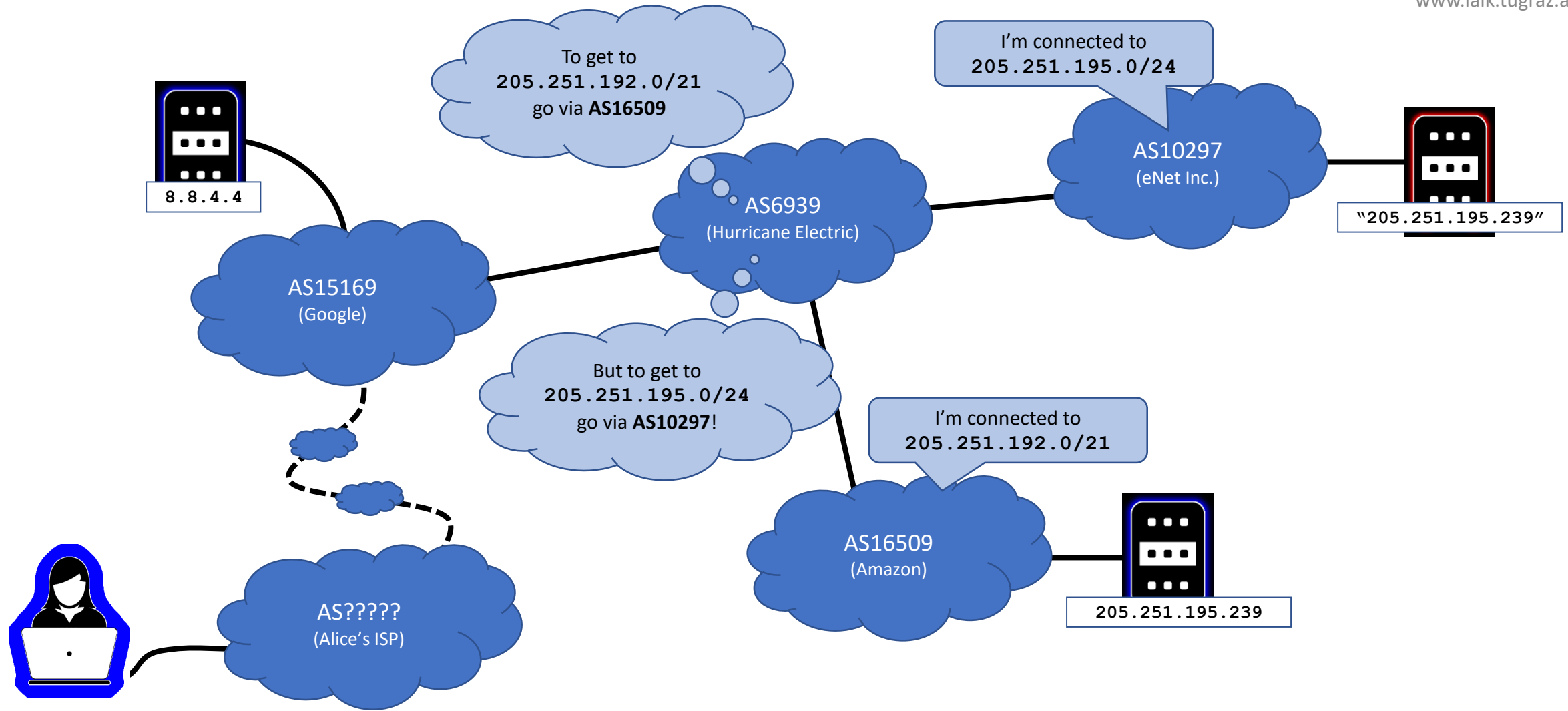
The Internet on any other day...



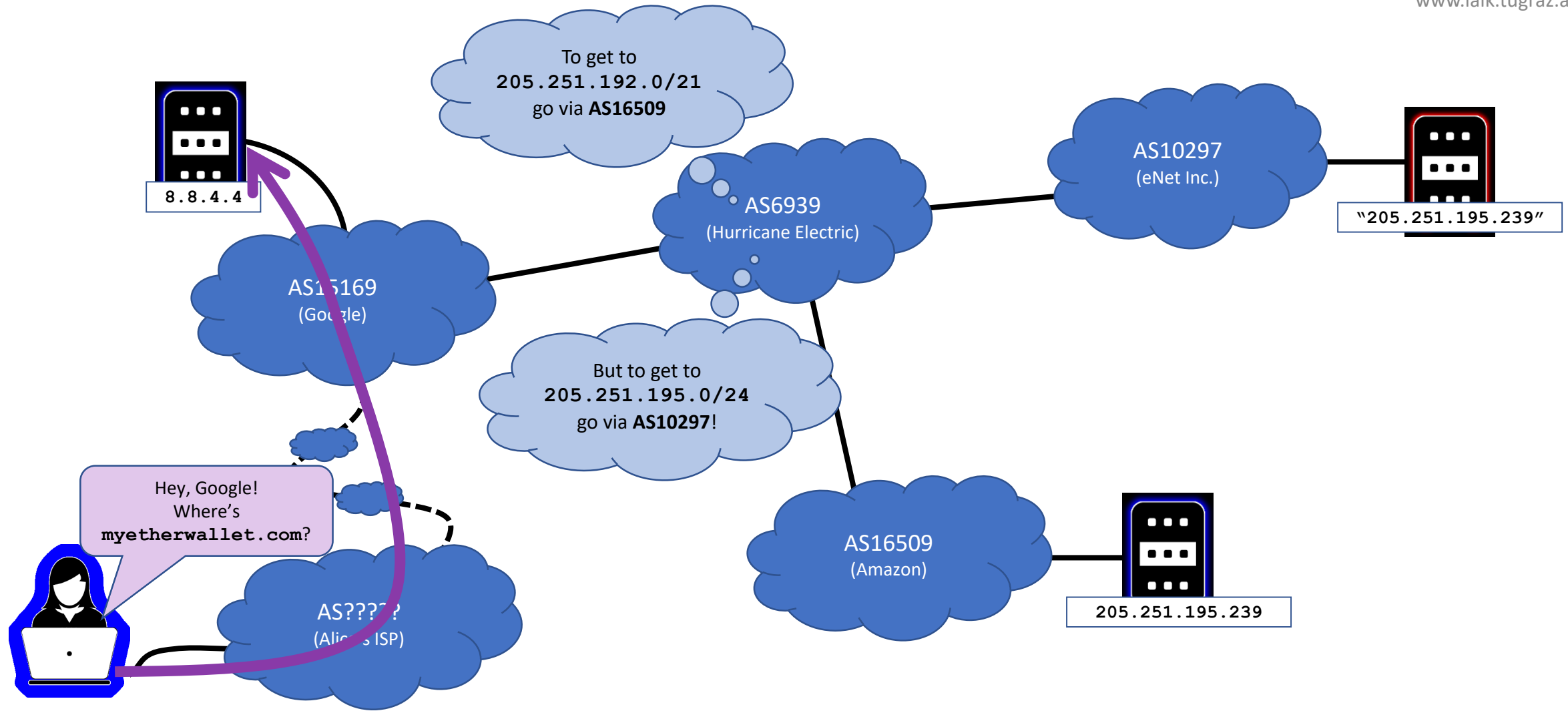
The Internet on any other day...



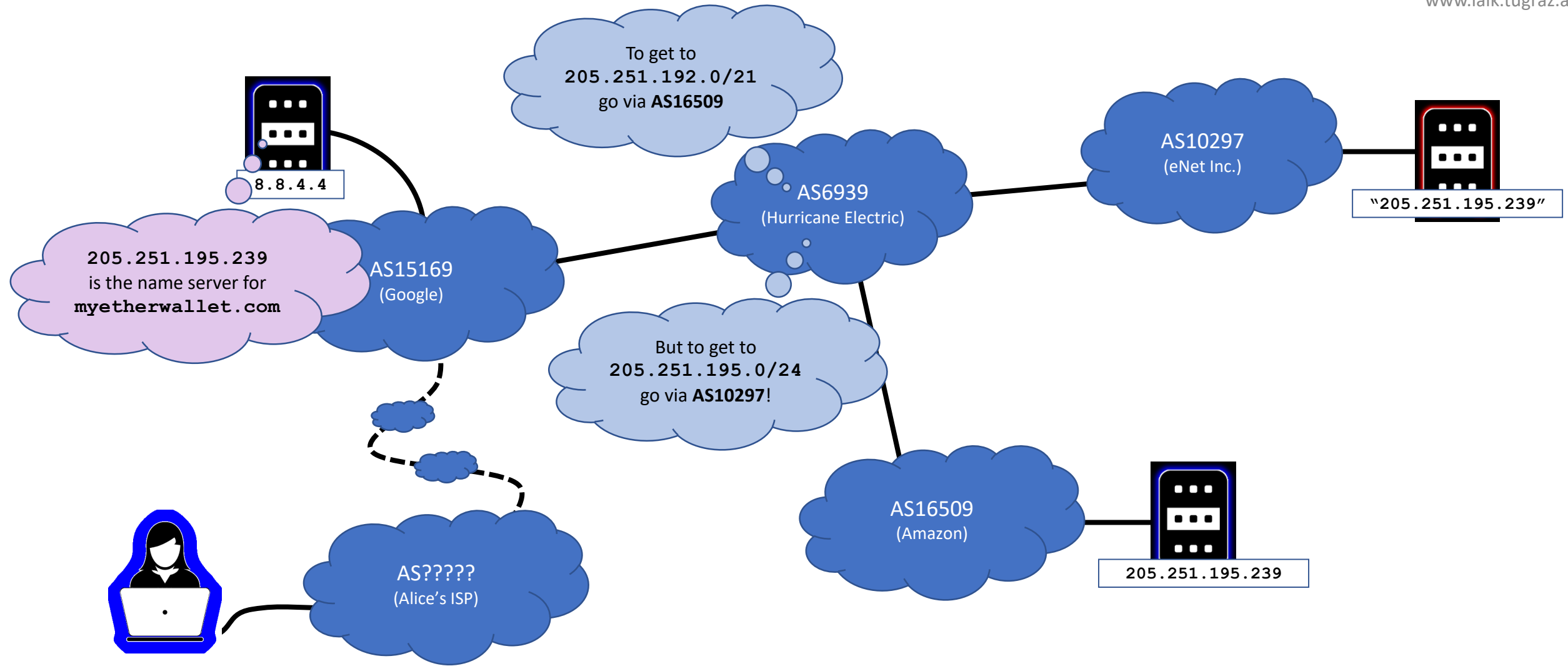
The Internet on Apr 24, 2018...



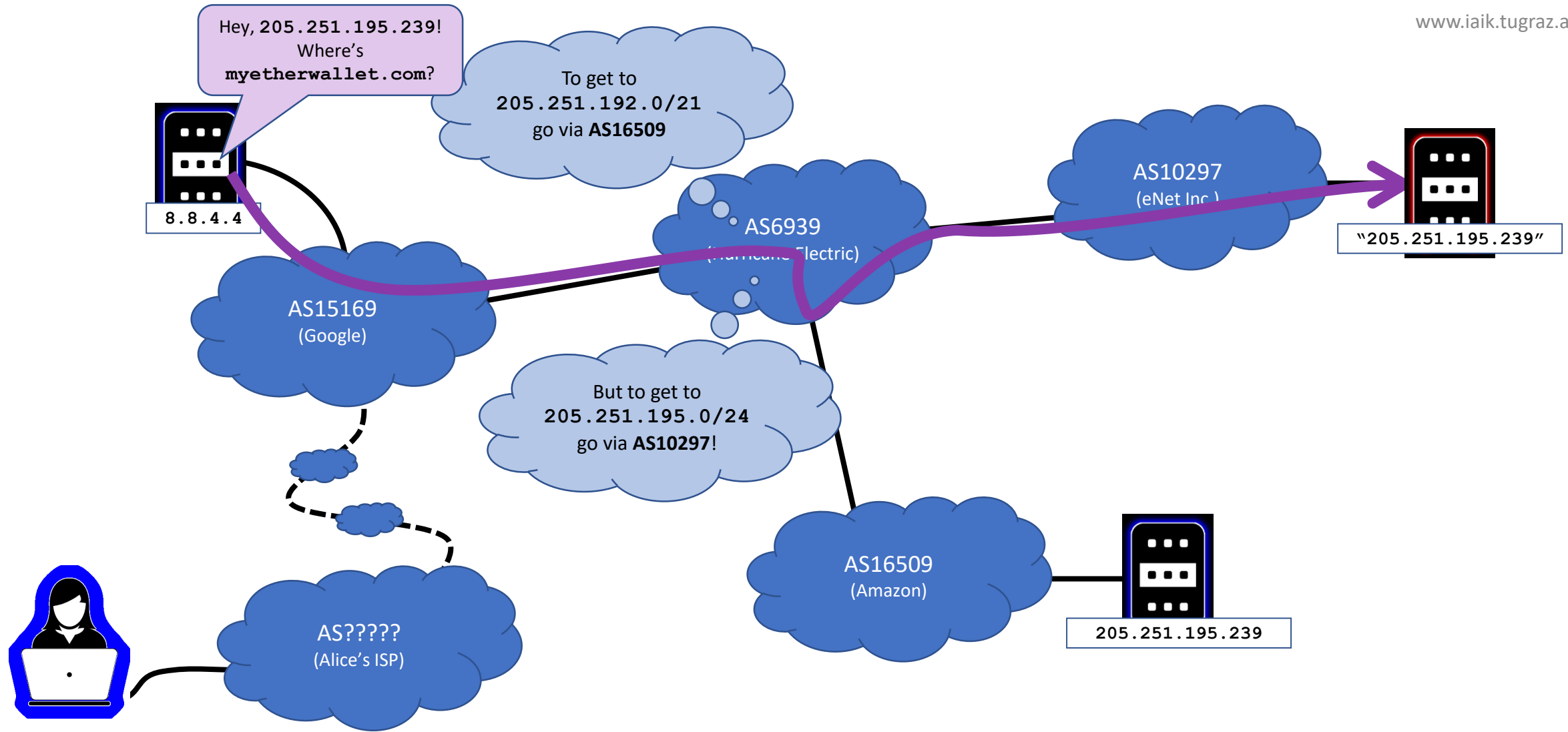
The Internet on Apr 24, 2018...



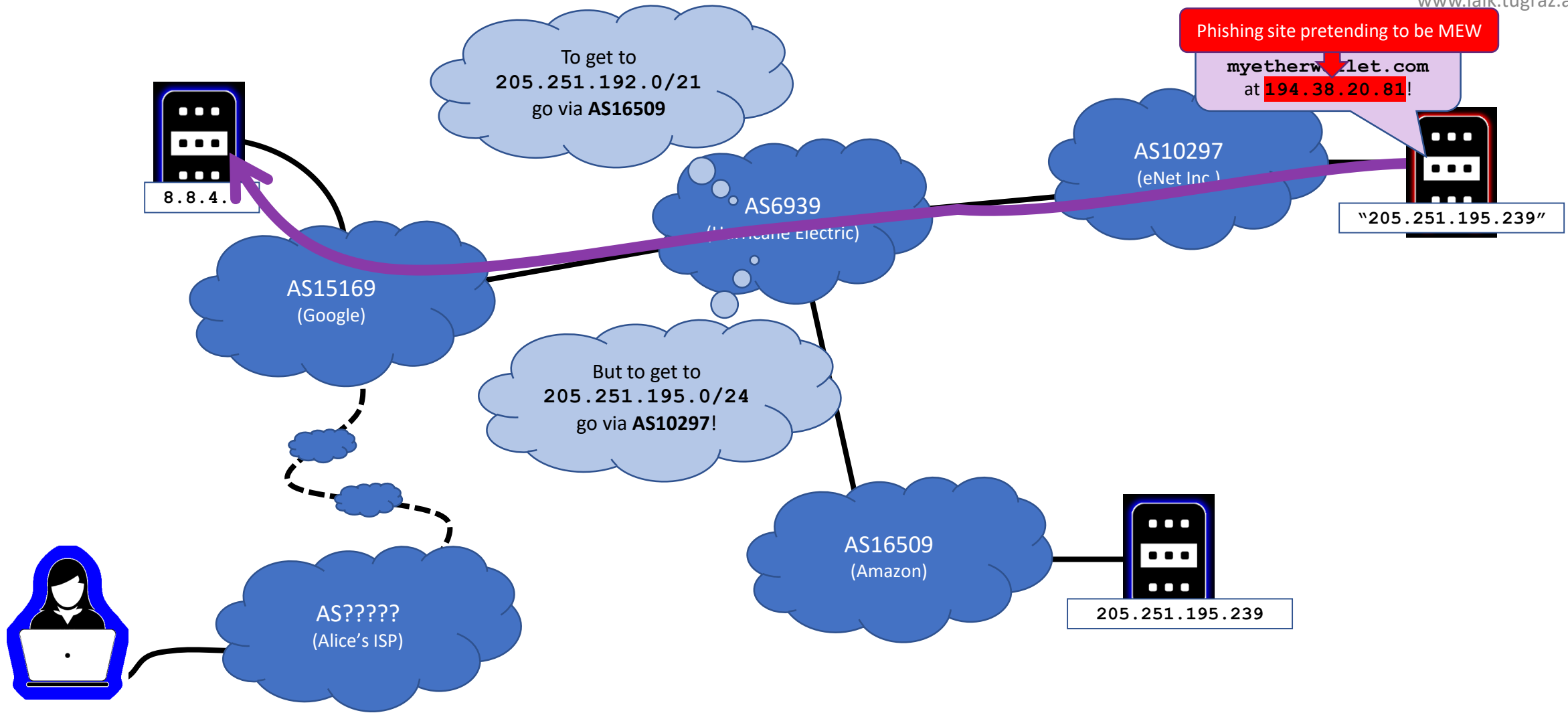
The Internet on Apr 24, 2018...



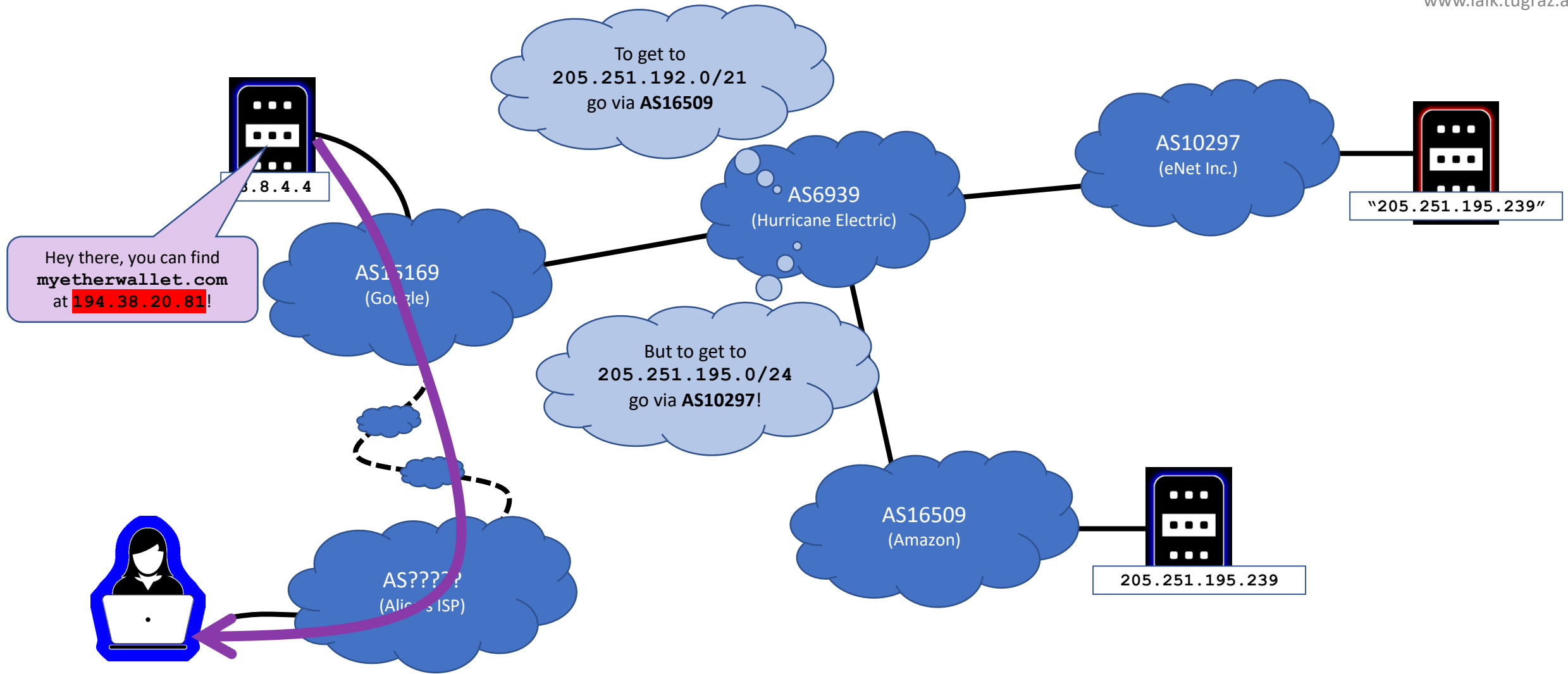
The Internet on Apr 24, 2018...



The Internet on Apr 24, 2018...



The Internet on Apr 24, 2018...



The Internet on Apr 24, 2018...

BGP hijacking – Solutions?

- DNSSEC
 - Digital signatures for DNS information
 - Would prevent the imposter DNS from sending forged replies
- BGP filtering
 - Would prevent the imposter advertisement from being picked up
- HTTPS
 - Did its job, the phishing site would've popped up a security warning
 - But users click past those...

Lower Layers – Recap

- Excellent at reliably delivering your data if everyone cooperates
- Not so excellent in the face of malicious actors

- Take-aways:
 - You cannot inherently trust that you are talking to the right person
 - You cannot inherently trust that your data is confidential
 - You cannot inherently trust that your data is unaltered
- The application layer has to take care of these things!

Bonus: Certificate Transparency

(If we have time, otherwise next week...)

HTTPS recap

- You open **https://online.tugraz.at/** in your browser
 - DNS lookup for **online.tugraz.at**
 - Browser connects to the returned IP address
 - Browser indicates that it wants to connect to **online.tugraz.at**
 - Server sends certificate **proving** that it is the **online.tugraz.at** server



Certificate


online.tugraz.at GEANT OV RSA CA 4 USERTrust RSA Certification Authority

Subject Name

Country	AT
	8010
State/Province	Steiermark
Locality	Graz
	Rechbauerstraße 12
Organization	Technische Universität Graz
Organizational Unit	Zentraler Informatikdienst
Common Name	online.tugraz.at

Issuer Name

Country	NL
Organization	GEANT Vereniging
Common Name	GEANT OV RSA CA 4



OK, so why should we trust them to not be lying...?

Certificate

online.tugraz.at

GEANT OV RSA CA 4

USERTrust RSA Certification Authority

Subject Name

Country	NL
Organization	GEANT Vereniging
Common Name	GEANT OV RSA CA 4

Issuer Name

Country	US
State/Province	New Jersey
Locality	Jersey City
Organization	The USERTRUST Network
Common Name	USERTrust RSA Certification Authority



OK, so why should we trust them to not be lying...?

Certificate

online.tugraz.at

GEANT OV RSA CA 4

USERTrust RSA Certification Authority

Subject Name

Country	US
State/Province	New Jersey
Locality	Jersey City
Organization	The USERTRUST Network
Common Name	USERTrust RSA Certification Authority

Issuer Name

Country	US
State/Province	New Jersey
Locality	Jersey City
Organization	The USERTRUST Network
Common Name	USERTrust RSA Certification Authority



Source: Don't worry, I got this.

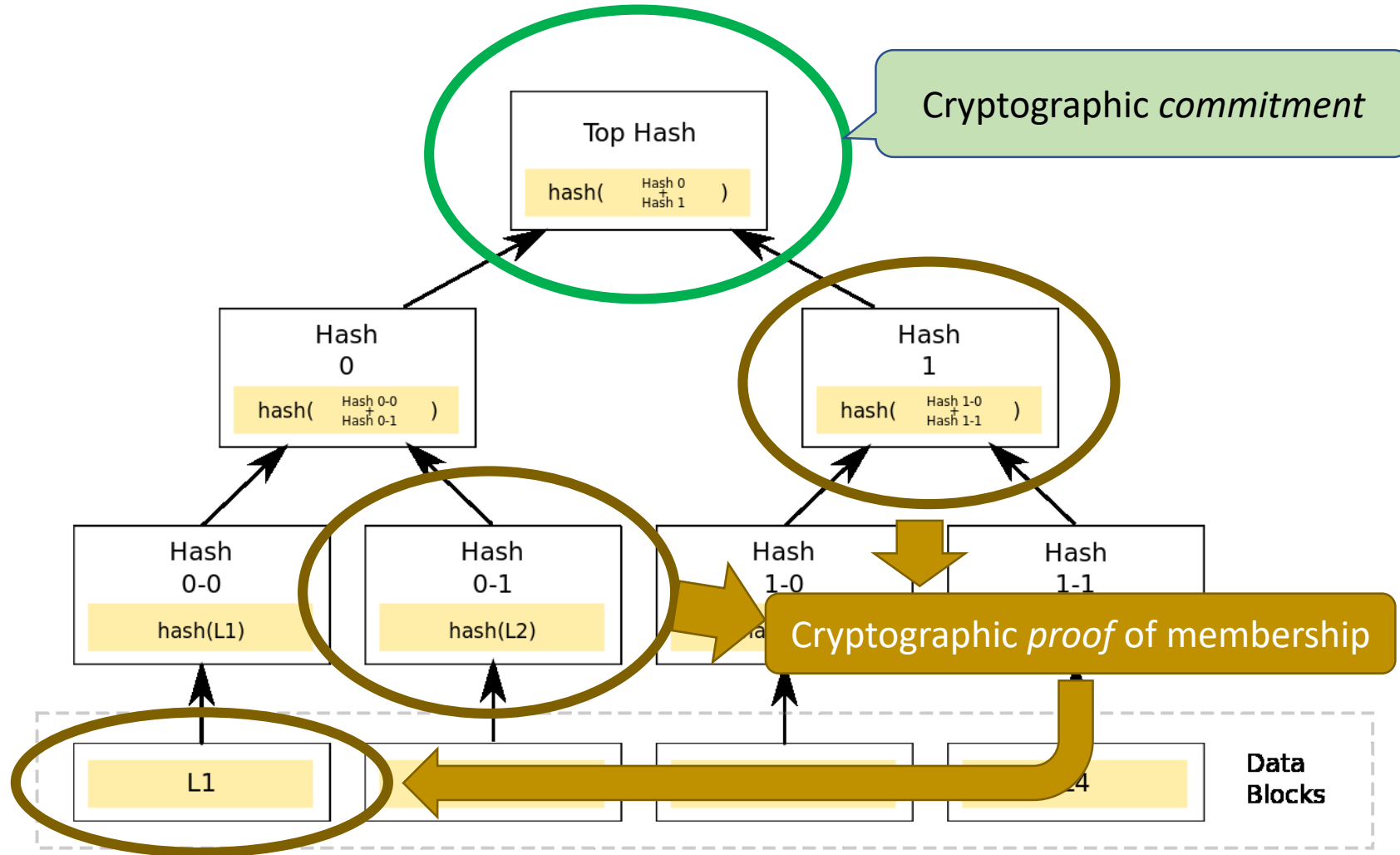
So why do we trust these people?

- Getting into a major browser's default trust store is *hard*
 - Think *years* of lead-up time
- Browsers care a lot about the security of their trust store
 - There's a very, *very* long list of security and auditing requirements
 - Violations will get you removed from browsers' trust stores quickly
 - If SSL certificates are your business, that means you no longer have a business

- Still not good enough for you?

(a very quick overview of)

Merkle Trees



Certificate Transparency

- Append-only log of *all* issued certificates using Merkle trees
 - Publicly available for monitoring
- Certificates can include a proof of inclusion
 - Mandatory in major browsers since 2021

Embedded SCTs	
Log ID	46:A5:55:EB:75:FA:91:20:30:B5:A2:89:69:F4:F3:7D:11:2C:41:74:BE:FD:49:B8:85:AB...
Name	Google "Xenon2022"
Signature Algorithm	SHA-256 ECDSA
Version	1
Timestamp	Thu, 09 Jul 2020 08:02:42 GMT
Log ID	DF:A5:5E:AB:68:82:4F:1F:6C:AD:EE:B8:5F:4E:3E:5A:EA:CD:A2:12:A4:6A:5E:8E:3B:1...
Name	Let's Encrypt Oak 2022
Signature Algorithm	SHA-256 ECDSA
Version	1
Timestamp	Thu, 09 Jul 2020 08:02:42 GMT
Log ID	6F:53:76:AC:31:F0:31:19:D8:99:00:A4:51:15:FF:77:15:1C:11:D9:02:C1:00:29:06:8D...
Name	Sectigo (Comodo) "Mammoth" CT
Signature Algorithm	SHA-256 ECDSA
Version	1
Timestamp	Thu, 09 Jul 2020 08:02:42 GMT