Modern Privacy Initiatives

Protocols for privacy partitioning

Tommy Pauly / **CS249i Winter 2025**

What do we mean by "privacy" in modern Internet protocols?

"Some privacy threats are already considered in Internet protocols as a matter of routine security analysis.

> *Surveillance, Stored Data Compromise, Intrusion, Misattribution*

Others are more pure privacy threats that existing security considerations do not usually address."

> ***Correlation***, ***Identification***, *Secondary Use, Disclosure, Exclusion*

RFC 6973, Section 5

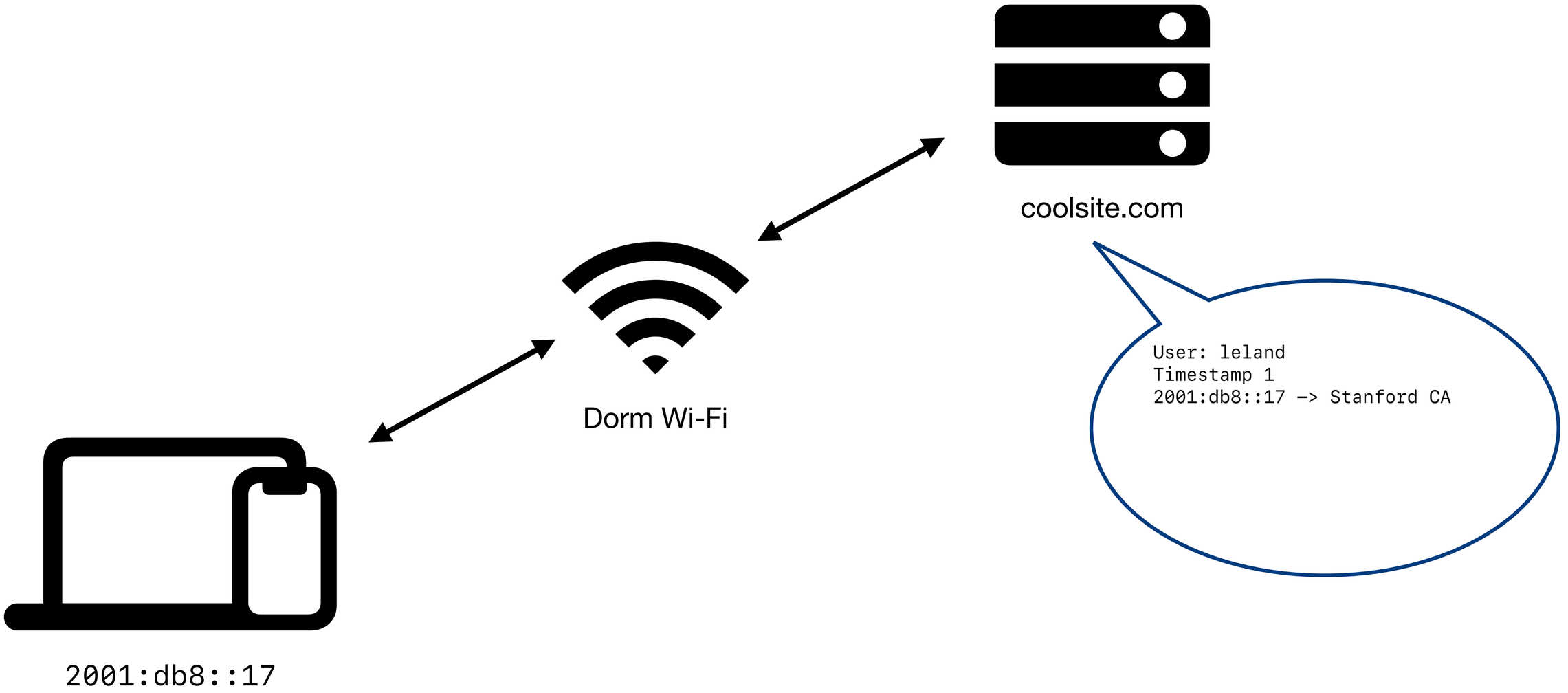
Privacy ≠ Security

Privacy ≠ Security

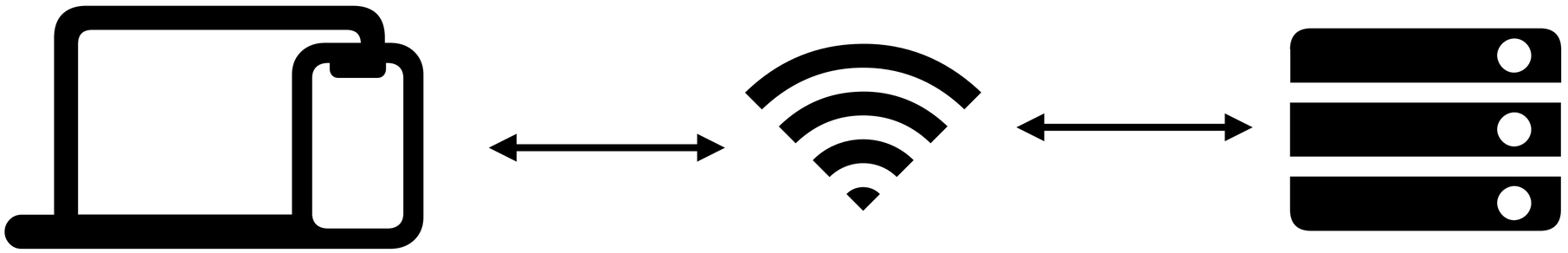
Security is a component of and prerequisite for privacy, but isn't sufficient

Metadata that is visible along the network path can be used to **correlate** activity and **identify** users, destinations, and content

Location tracking



Location tracking



2001:db8::cafe:9

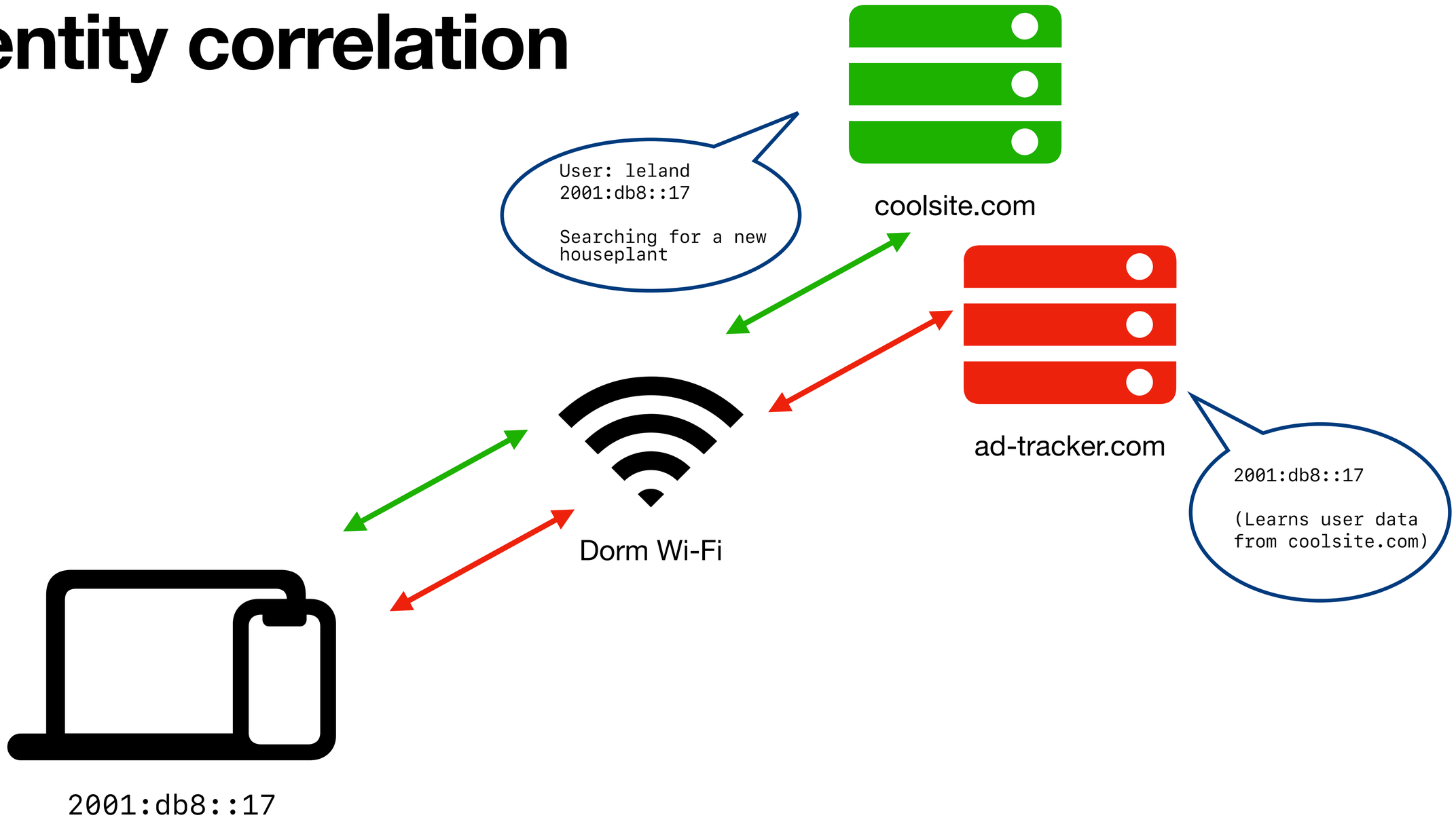
Coffeeshop Wi-Fi

coolsite.com

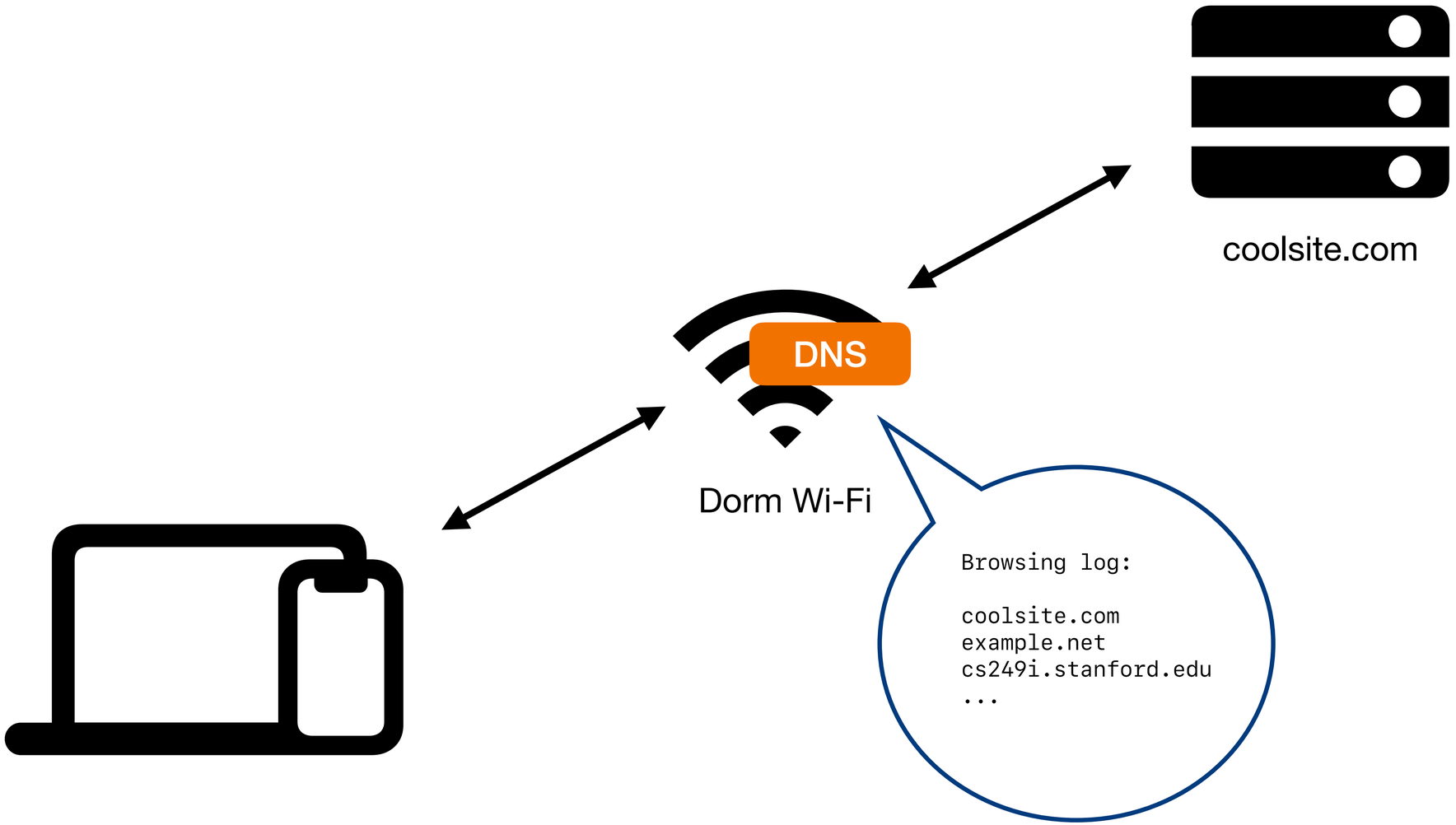
```
User: leland
Timestamp 1
2001:db8::17 -> Stanford CA


User: leland
Timestamp 2
2001:db8::cafe:9 -> Palo Alto CA
```

Identity correlation



Activity Tracking



"**Correlation** is the combination of various pieces of information related to an individual or that obtain that characteristic when combined....

Correlation is closely related to identification. Internet protocols can facilitate correlation by allowing individuals' activities to be **tracked and combined** over time....

Pseudonymity is strengthened when less personal data can be linked to the pseudonym; when **the same pseudonym is used less often and across fewer contexts**; and when independently chosen pseudonyms are more frequently used for new actions (making them, from an observer's or attacker's perspective, unlinkable)."

RFC 6973, Section 5.2.1

Correlation exists at *all* layers

Managing HTTP cookies is not sufficient

The **fundamentals** of the Internet (addresses, DNS, etc) are a key part of the privacy problem

Why isn't "incognito mode" enough?

Browsers offer "incognito mode" / "private browsing"

- Users often think of this as "Internet privacy" — it's not!

Historically was about not saving history on the local machine, and not storing cookies

- ✅ Stops obvious identifiers (log-in credentials, etc) from being shared with servers unintentionally
- ❌ Doesn't prevent servers from using IP addresses to tag location or recognize users
- ❌ Doesn't prevent network operators from observing traffic and recording history

Some browsers are improving what "private browsing" means

IP addresses and **hostnames** are **pseudonyms** that can represent users, groups of users, and content

Every network operation is littered with these identifiers *(client and server IP addresses in transport connections, hostnames in DNS and TLS SNI)*

Removing correlation requires using **partitioning** these identifiers into **different contexts**

Privacy partitioning aims to separate *who* someone is from *what* they do

RFC 9614, Section 2

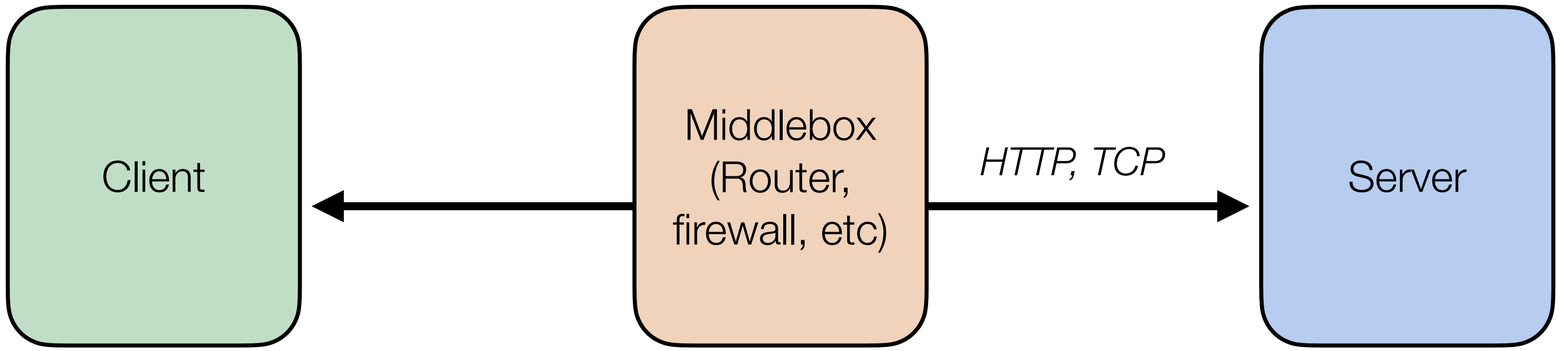
"In order to prevent the correlation of user-specific information across contexts, partitions need to ensure that **any single entity** (other than the client itself) **does not participate in more than one context** where the information is visible.

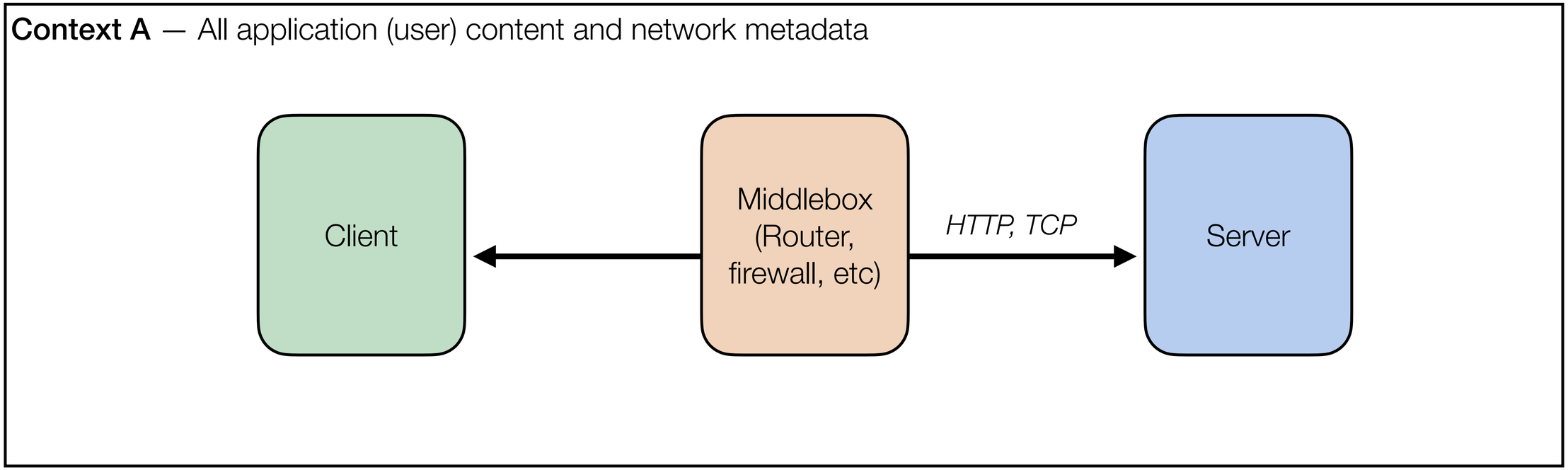
...any identifier at any layer that is common across different contexts can be used as a way to correlate activity."

RFC 9614, Section 2.1

How do we think about contexts?

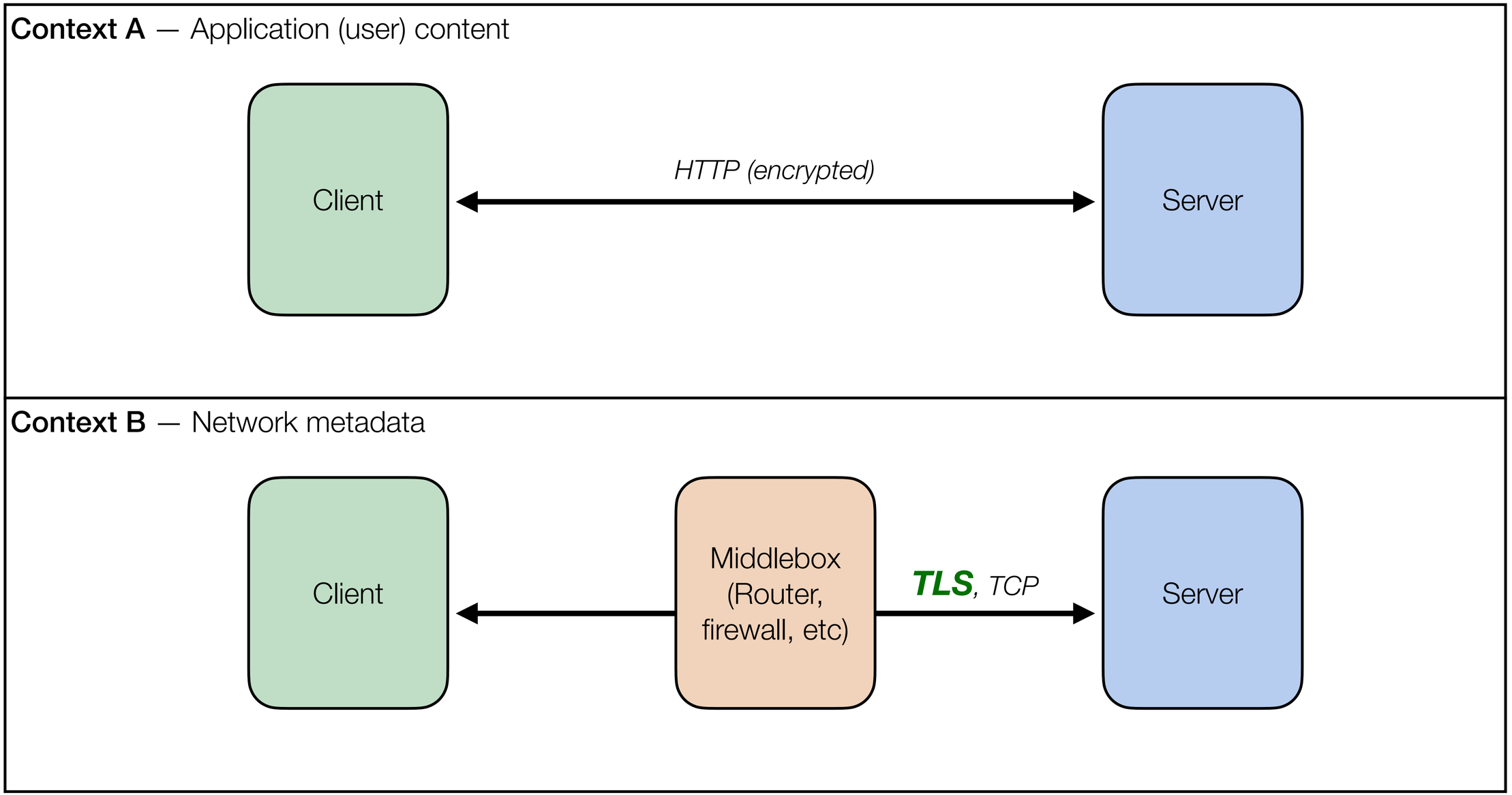
Context A — All application (user) content and network metadata



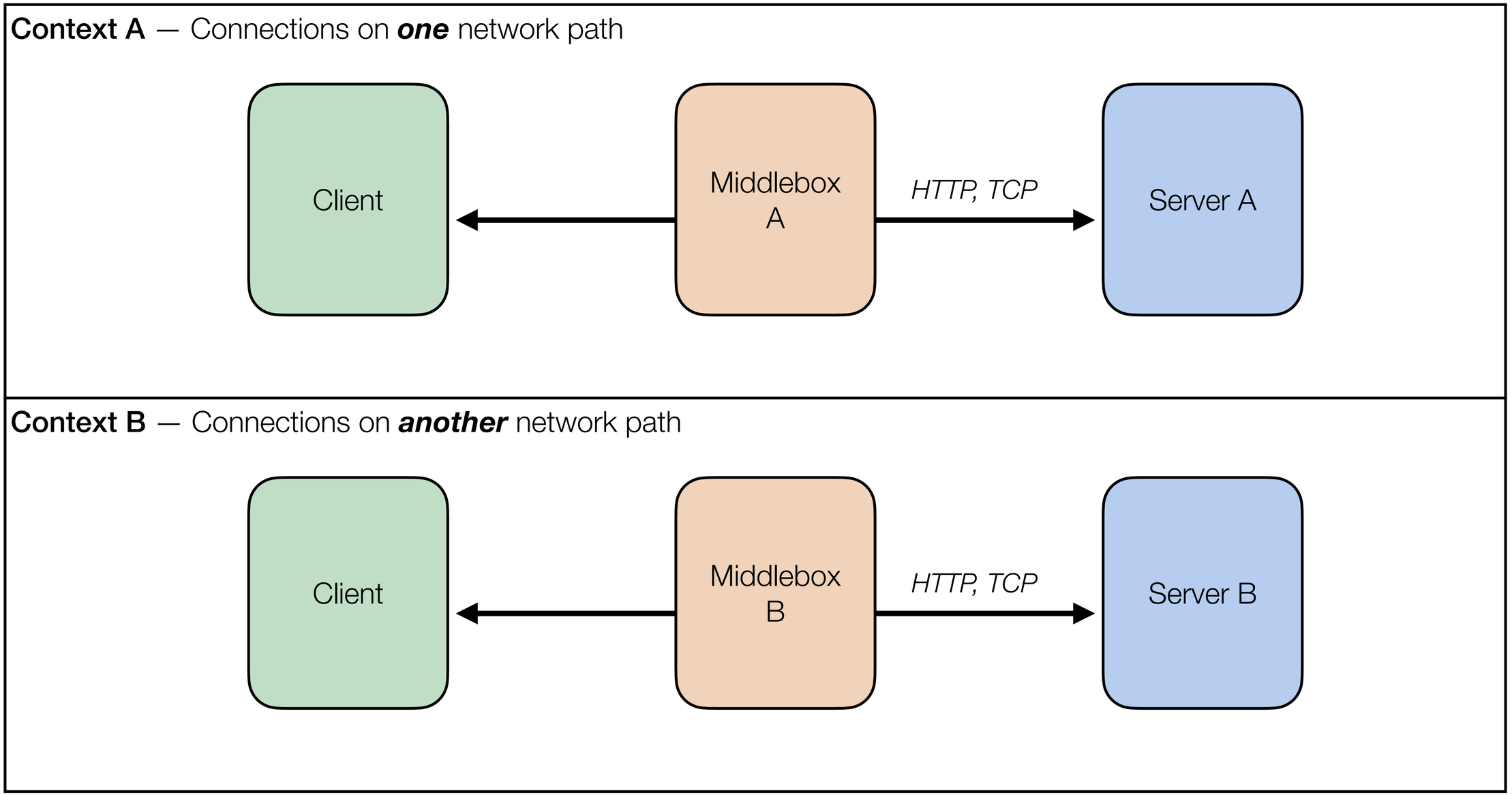


What mechanisms can we use to break up this context?

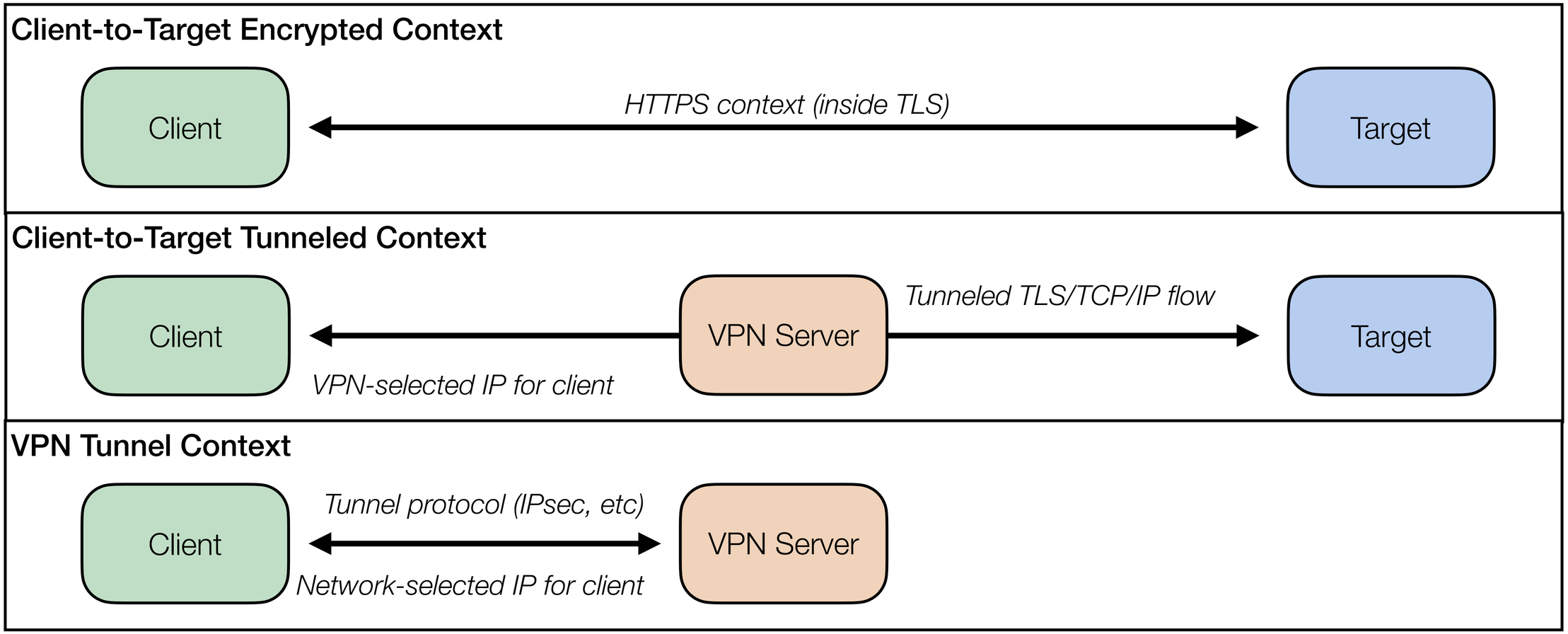
1. Cryptographic protection


Context A — Application (user) content
Client
HTTP (encrypted)
Server
Context B — Network metadata
Client
Middlebox
(Router,
firewall, etc)
TLS, TCP
Server

2. Connection separation

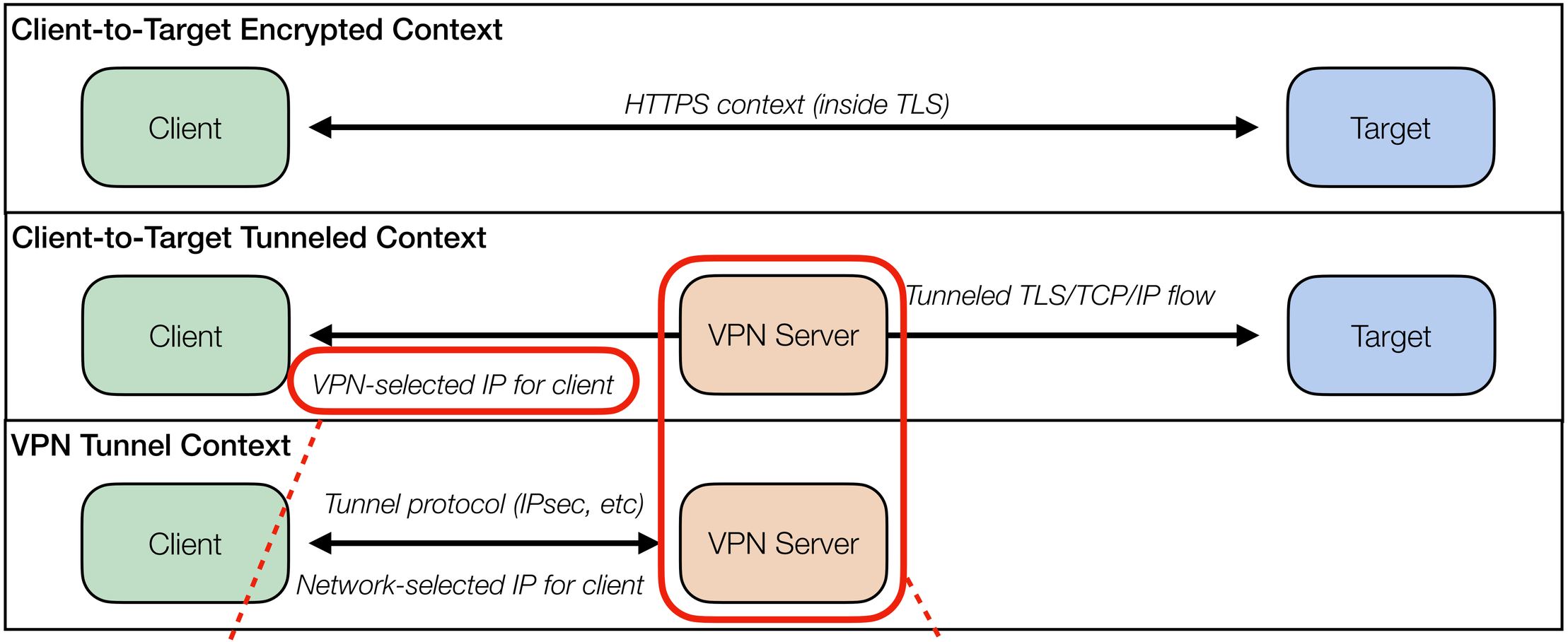

Context A — Connections on one network path
Client
Middlebox A
HTTP, TCP
Server A
Context B — Connections on another network path
Client
Middlebox B
HTTP, TCP
Server B

Case study: VPN



...what are some privacy problems with this "partitioning?

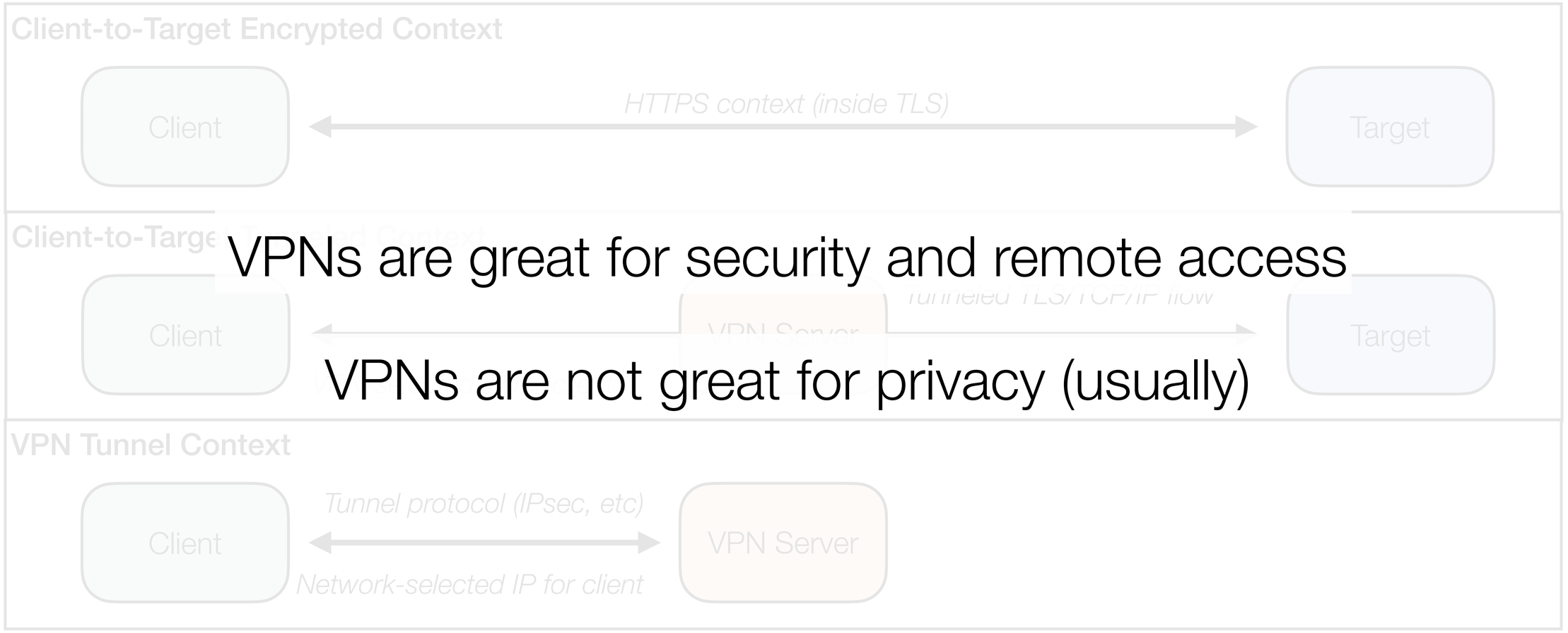
Case study: VPN



The target can see a stable client identifier; if this gets linked back to the original client IP, it can be used to correlate all traffic

VPN server is in too many contexts; It can observer all client and target identifiers

Case study: VPN



VPNs are great for security and remote access

VPNs are not great for privacy (usually)

Use cases for privacy partitioning

Why would we design protocols specifically to make things more private?

Prevent servers from collecting user identity or location without permission

Prevent networks from collecting user activity history without permission

Prove that a user has permission to access content without correlating their identity with the specific piece of content they are accessing

Allow anonymous collection of aggregated metrics without collecting user activity

Modern protocols using privacy partitioning

MASQUE: RFC 9298 (Proxying UDP in HTTP), RFC 9484 (Proxying IP in HTTP)

Oblivious HTTP: RFC 9458 (and Oblivious DNS over HTTP, RFC 9230)

Privacy Pass: RFC 9576, RFC 9577, RFC 9578

Privacy Preserving Metrics (PPM) / DAP: draft-ietf-ppm-dap

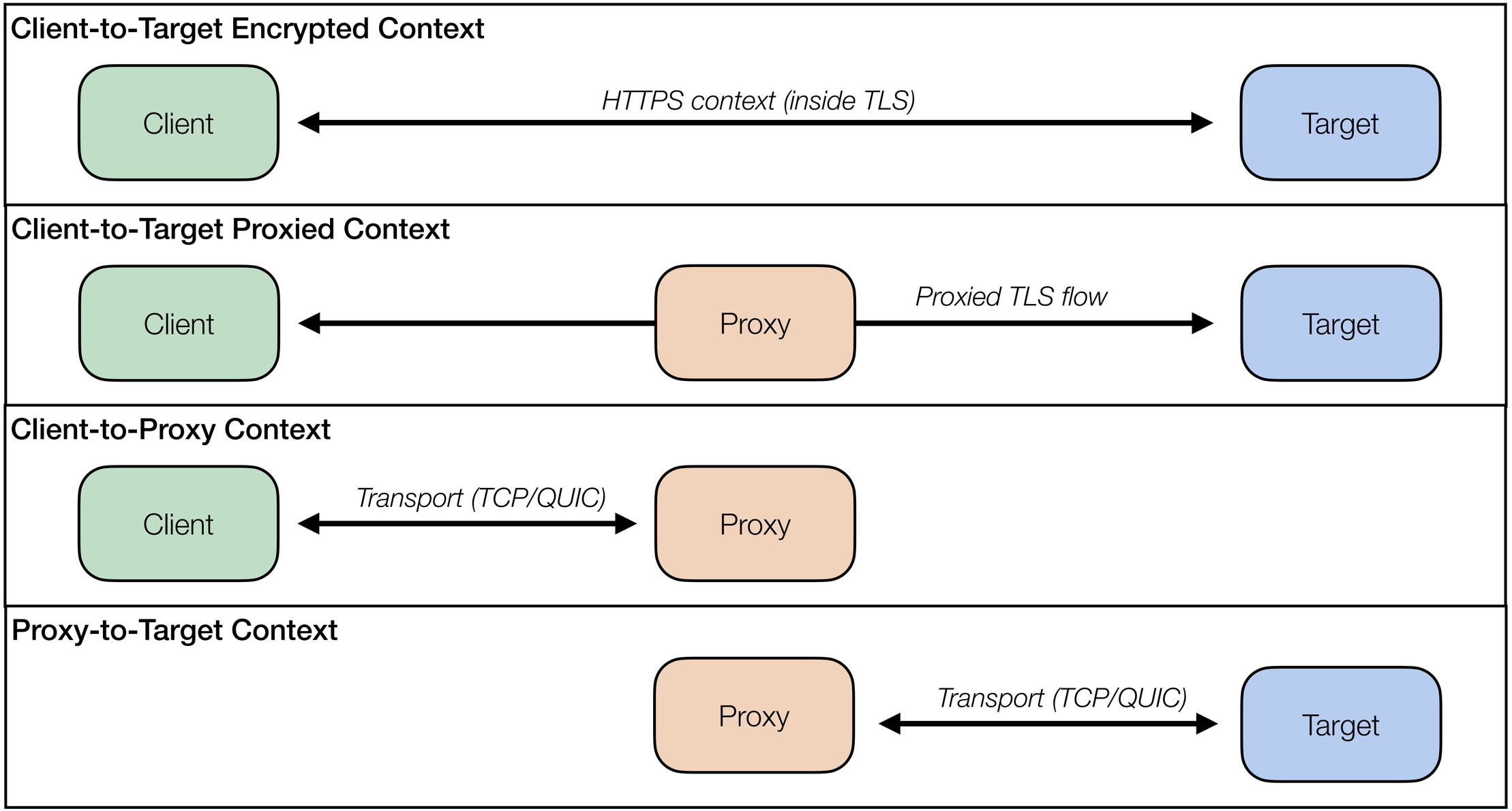
MASQUE

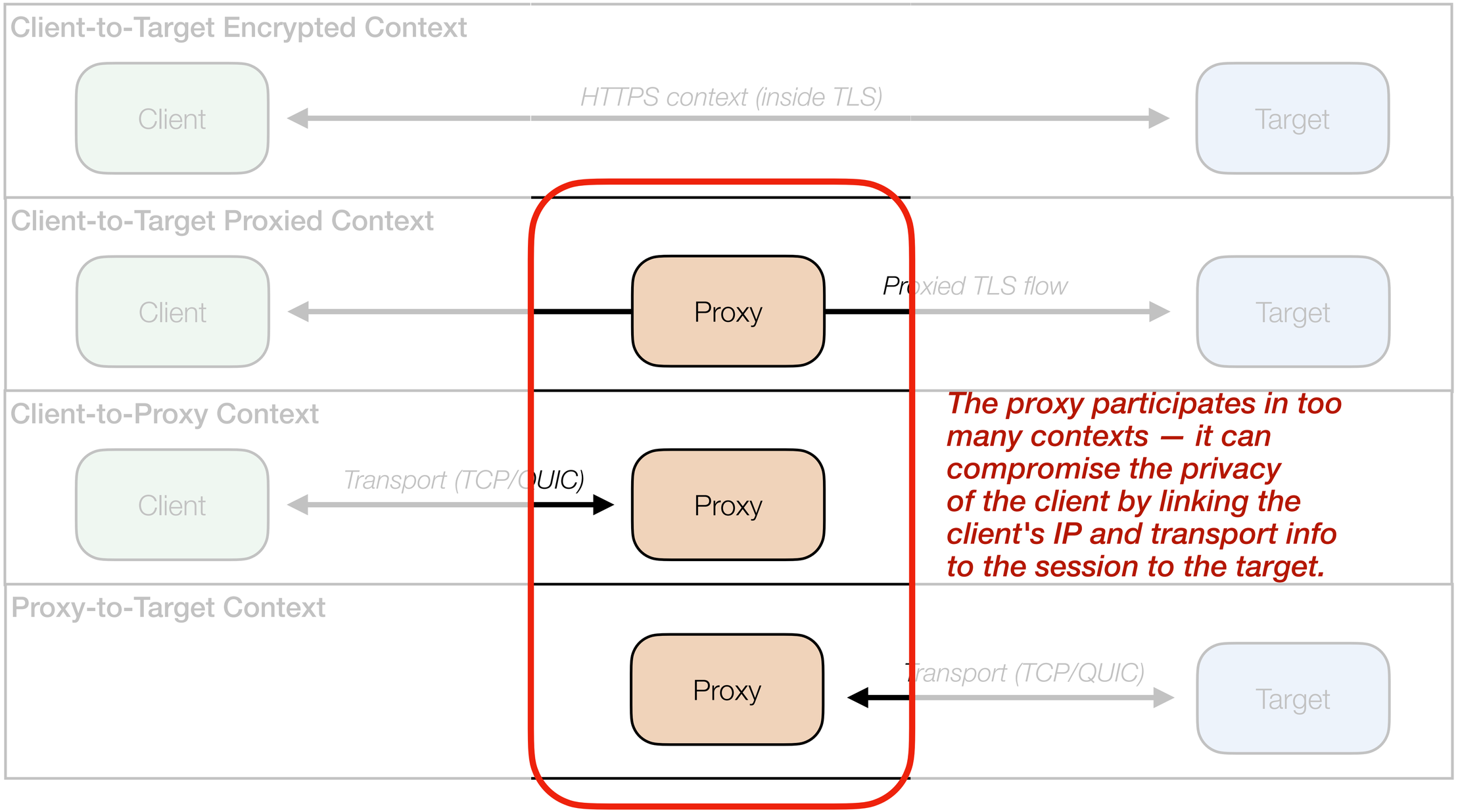
HTTP CONNECT (forward) proxy

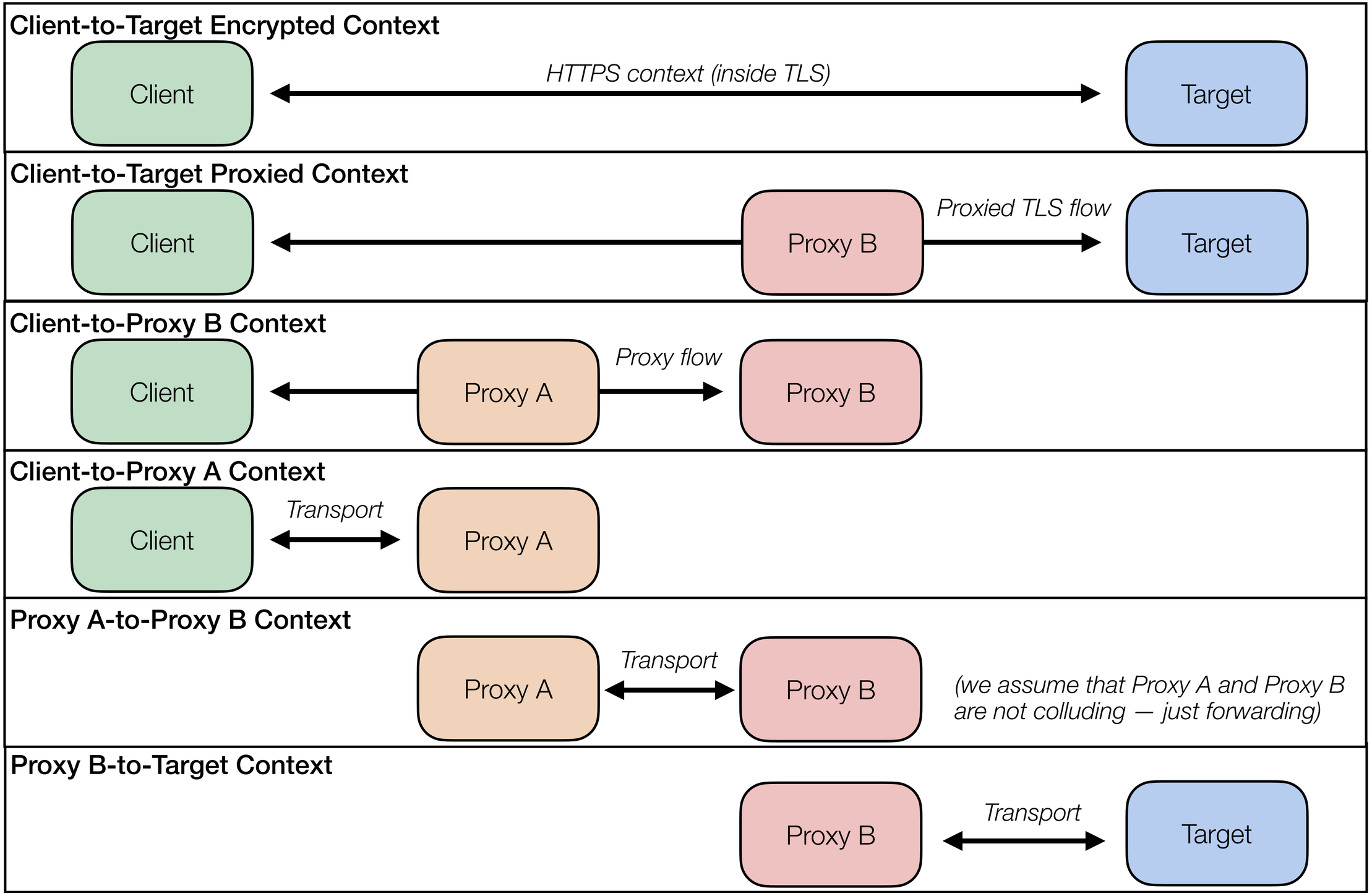
- Connections to proxies are encrypted with TLS
- Existing CONNECT method allowed proxying TCP streams
- "CONNECT-UDP" allows proxying UDP
- "CONNECT-IP" allows proxying IP
- Prefers using HTTP/3, leveraging QUIC DATAGRAMs

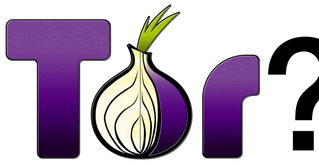
Proxy to any TCP/UDP/QUIC/IP endpoint!

RFC 9298, RFC 9484


Client-to-Target Encrypted Context
Client
HTTPS context (inside TLS)
Target
Client-to-Target Proxied Context
Client
Proxy
Proxied TLS flow
Target
Client-to-Proxy Context
Client
Transport (TCP/QUIC)
Proxy
Proxy-to-Target Context
Proxy
Transport (TCP/QUIC)
Target


Client-to-Target Encrypted Context
Client
HTTPS context (inside TLS)
Target
Client-to-Target Proxied Context
Client
Proxy
Proxied TLS flow
Target
Client-to-Proxy Context
Client
Transport (TCP/QUIC)
Proxy
Proxy-to-Target Context
Proxy
Transport (TCP/QUIC)
Target
The proxy participates in too many contexts — it can compromise the privacy of the client by linking the client's IP and transport info to the session to the target.


Client-to-Target Encrypted Context
Client
HTTPS context (inside TLS)
Target
Client-to-Target Proxied Context
Client
Proxy B
Proxied TLS flow
Target
Client-to-Proxy B Context
Client
Proxy A
Proxy flow
Proxy B
Client-to-Proxy A Context
Client
Transport
Proxy A
Proxy A-to-Proxy B Context
Proxy A
Transport
Proxy B
(we assume that Proxy A and Proxy B
are not colluding — just forwarding)
Proxy B-to-Target Context
Proxy B
Transport
Target

Why not just use  ?

Making privacy ubiquitous

The best privacy solution is the privacy solution that many people actually use

- Little to no performance impact (user-perceived latency)
- Scale to many millions of users
- Avoid being blocked by sites due to poor IP reputation

HTTP proxies that can be deployed by major CDNs meet these needs

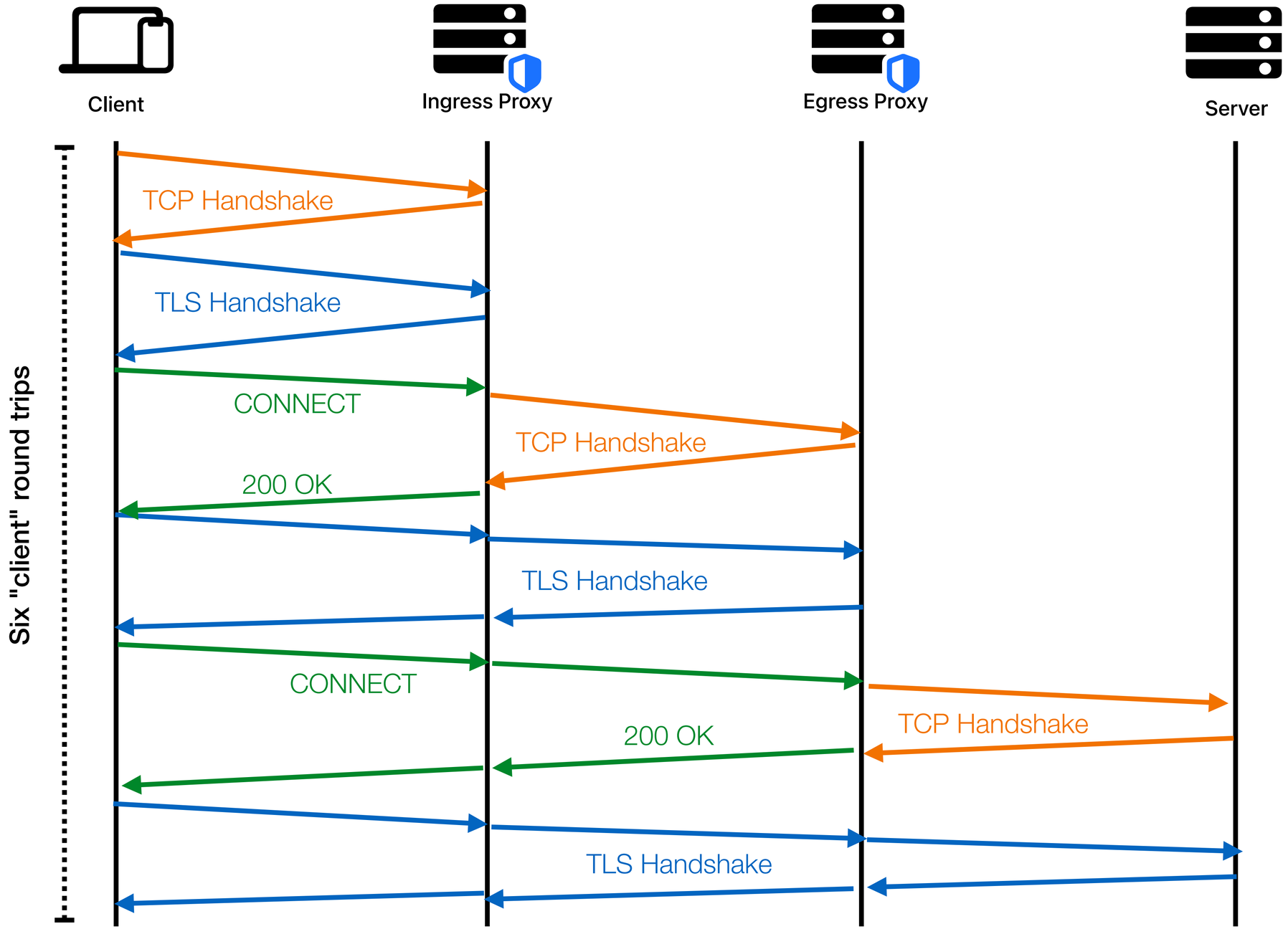
- HTTP/3 is a particularly useful protocol here (datagrams, multiplexing)

Privacy ≠ Slowness

Naive approach using TCP connections

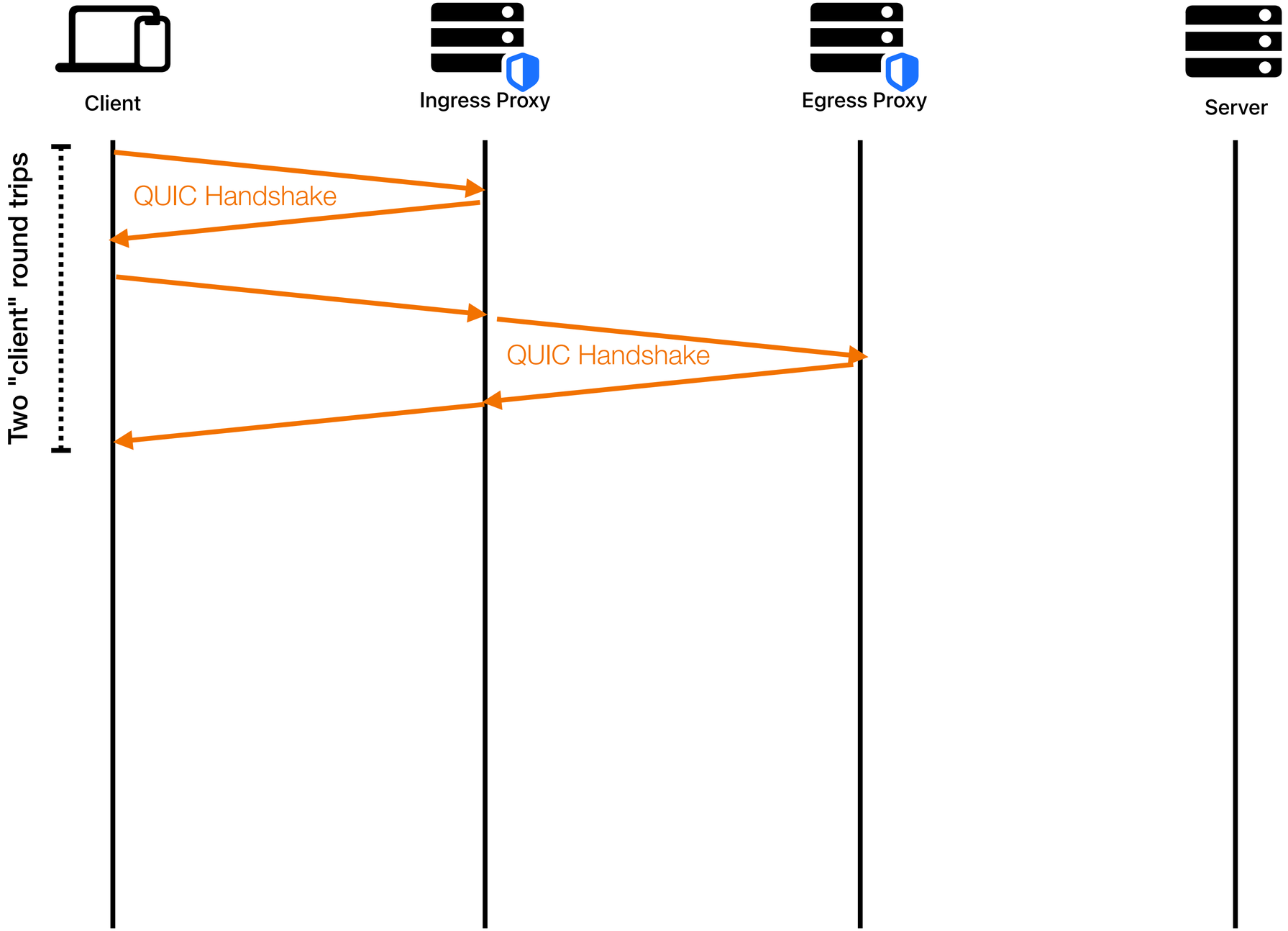
(HTTP/1.1 or SOCKS)

Each end-to-end connection requires new proxy connections



With QUIC (HTTP/3), proxy connection setup takes only two round trips

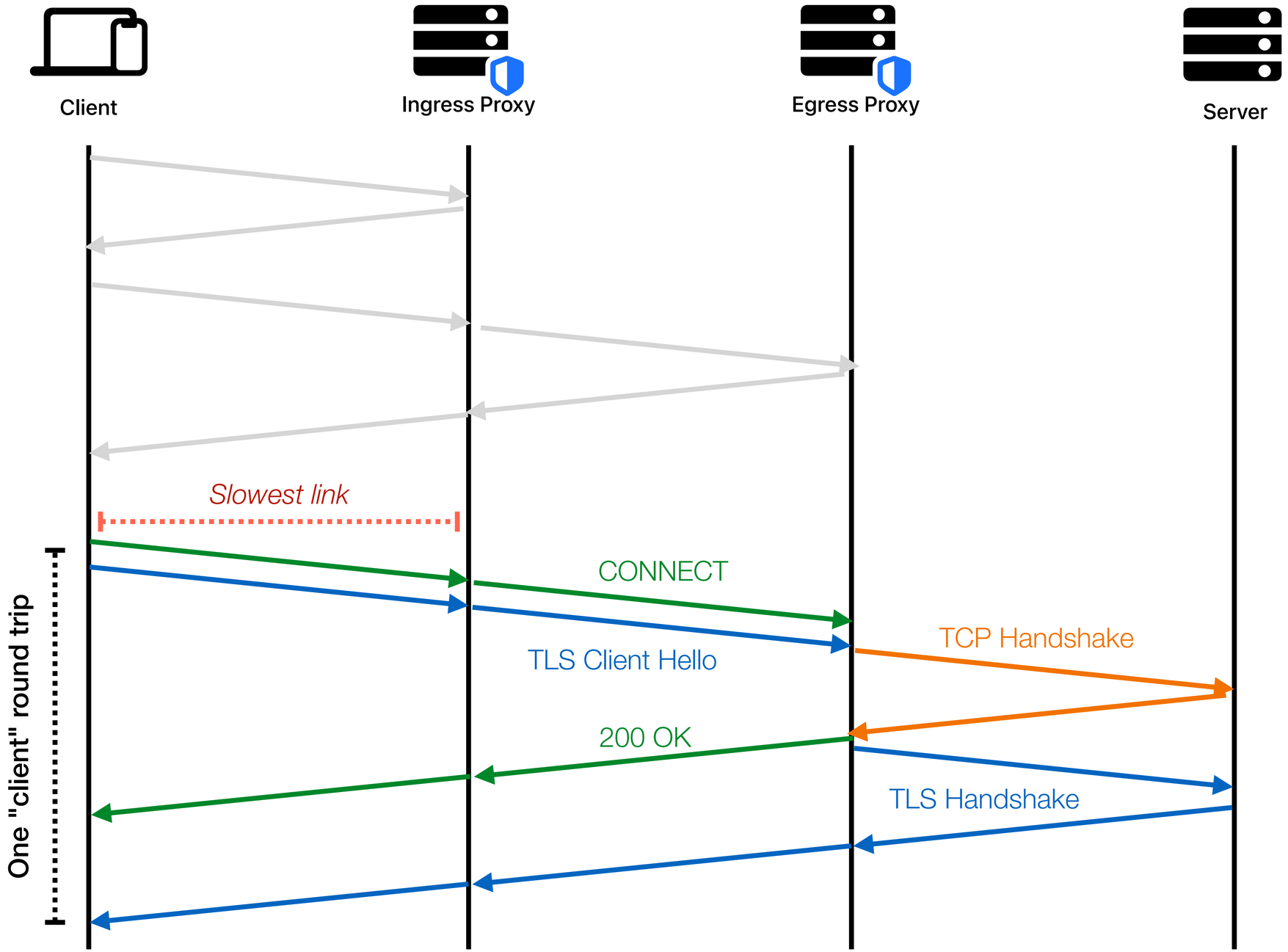
With 0-RTT resumption, can be even faster!



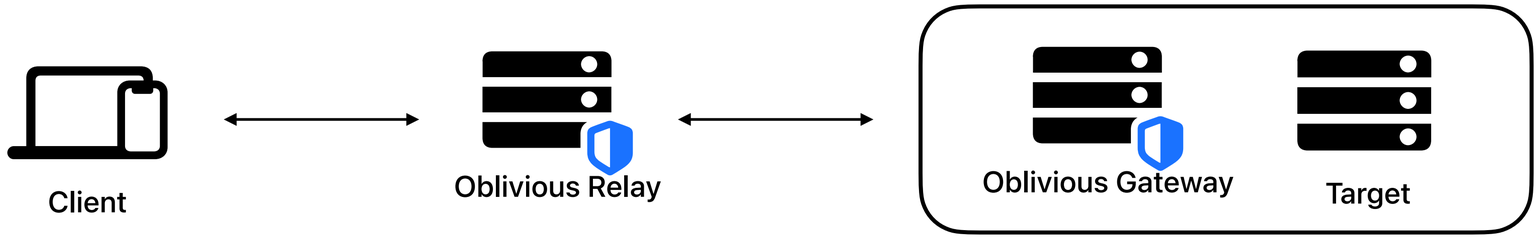
Each end-to-end TLS connection then only takes a single round trip

End-to-end connections are multiplexed on existing QUIC tunnels

Faster than non-proxied connections!



Oblivious HTTP



Application-level encryption, send over a relay (reverse proxy)

- Uses Hybrid Public Key Encryption (HPKE)
- Client knows the public key of the gateway, and uses it to send messages
- Connections to the relay are long-lived and reused for many messages

Proxy to specifically cooperating endpoints (unlike MASQUE)

RFC 9458

MASQUE vs Oblivious HTTP

MASQUE	OHTTP
Medium-weight (Requires end-to-end TLS handshakes)	Very lightweight! (Per-message encryption)
Good for long-lived sessions (web browsing, video streaming)	Good for short, bursty messages (DNS, metrics, AI inference)
Requests from the same client can be linked when multiplexing	Messages from the same client cannot be linked, even with multiplexing
Communicate with arbitrary servers	Coordinating servers only
Has perfect forward secrecy	No perfect forward secrecy 😢

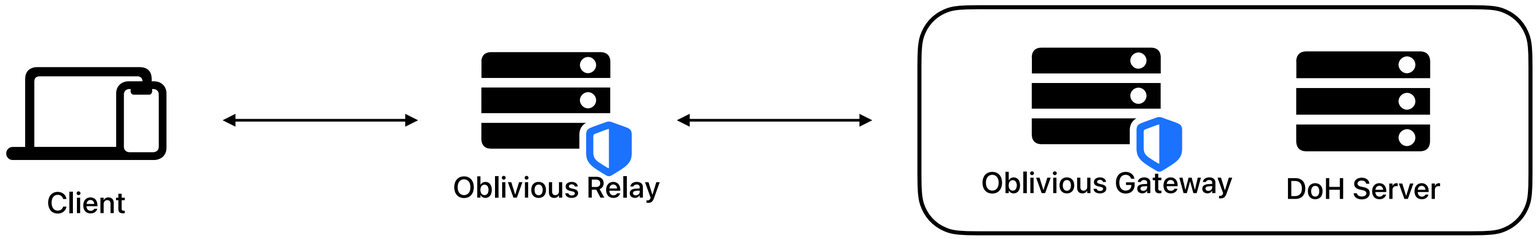
Oblivious DNS

DNS was one of the original motivations for "oblivious" protocols

- DNSSEC can provide integrity and authentication, but not confidentiality
- Encrypted DNS (DoH / DoT) can provide confidentiality and integrity from a network attacker, but not from the encrypted resolver itself

Oblivious DNS → Oblivious DoH → Oblivious HTTP

- Confidentiality and anonymity; resolver can't track client history, or hand specific answers to specific clients



Privacy ≠ Easy

Collusion

Partitioning relies on non-collusion across contexts

- If parties collude and share information, they can violate the privacy
- Technical mechanisms only go so far; policies and incentives are key

Mitigations to prevent collusion:

- Policies to not generate logs/history for each context; multiple policy violations necessary to re-join data
- Protocol restrictions to make it harder to share data across contexts
- Adding more partitions and using them unpredictably to make collusion harder and less effective

Faulty partitioning

There are many ways to incorrectly apply privacy partitioning!

Including client identifiers (like IP addresses, email addressees, etc) in application-level messages over a privacy proxy

Not including enough parties or partitions; VPNs often only have a single server which then becomes a single observation point from which to see all client locations and full history of server access

Traffic analysis and side channel attacks; observing timing and size information can let a passive attacker still link activity across proxies

Insufficient anonymity sets (too few clients, or too unique patterns)

Impacts of partitioning

Difficult to triage and debug!

Network operators lose traffic observability

Performance impairment if protocols are not designed to partition efficiently

More protocol participants → more points of failure

Incentives towards centralization

- Many clients using common infrastructure increases anonymity sets, but can put increased reliance on a few operators

Questions?