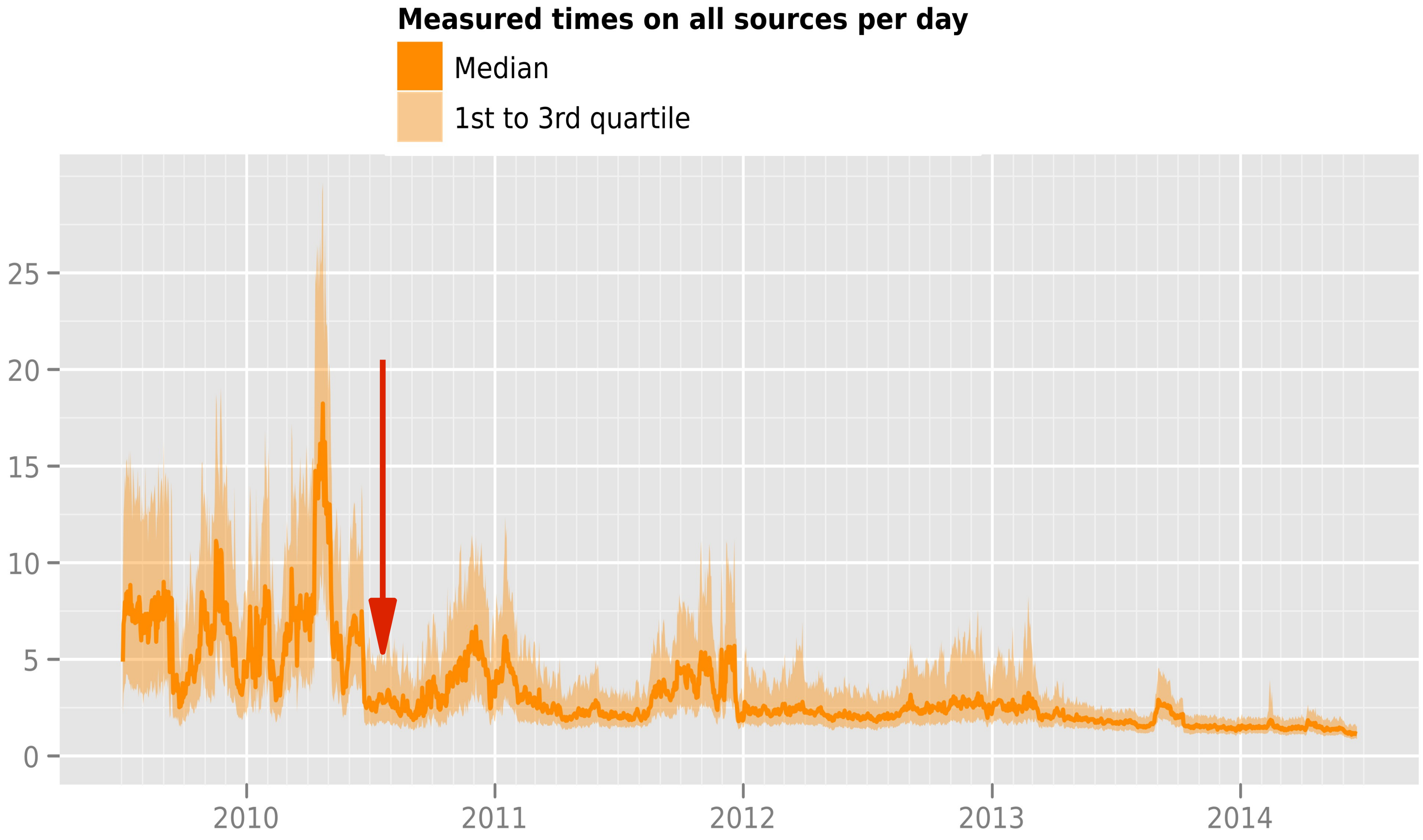
The Tor Project, Inc.

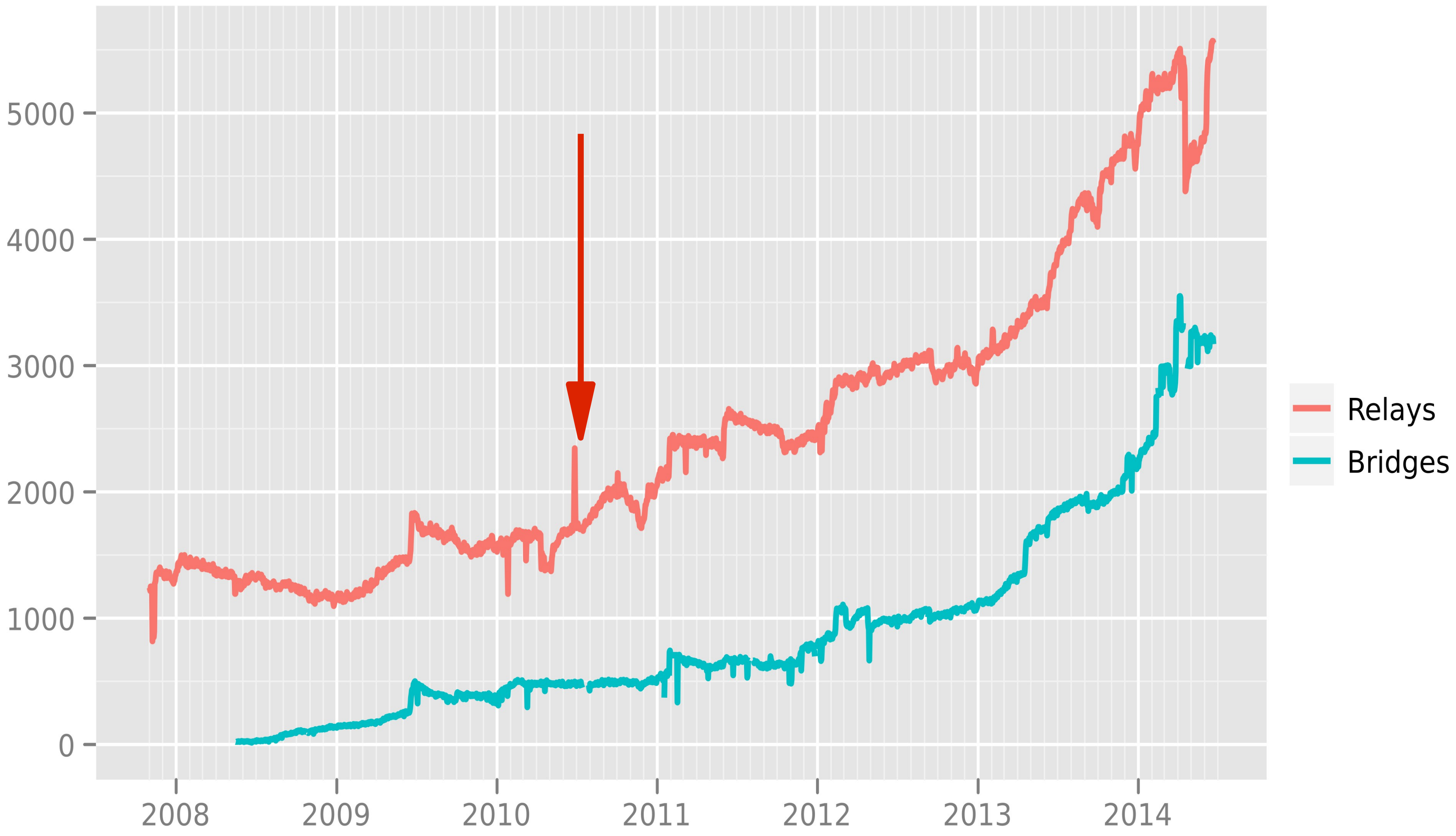
Our mission is to be the global resource for technology, advocacy, research and education in the ongoing pursuit of freedom of speech, privacy rights online, and censorship circumvention.

Time in seconds to complete 50 KiB request



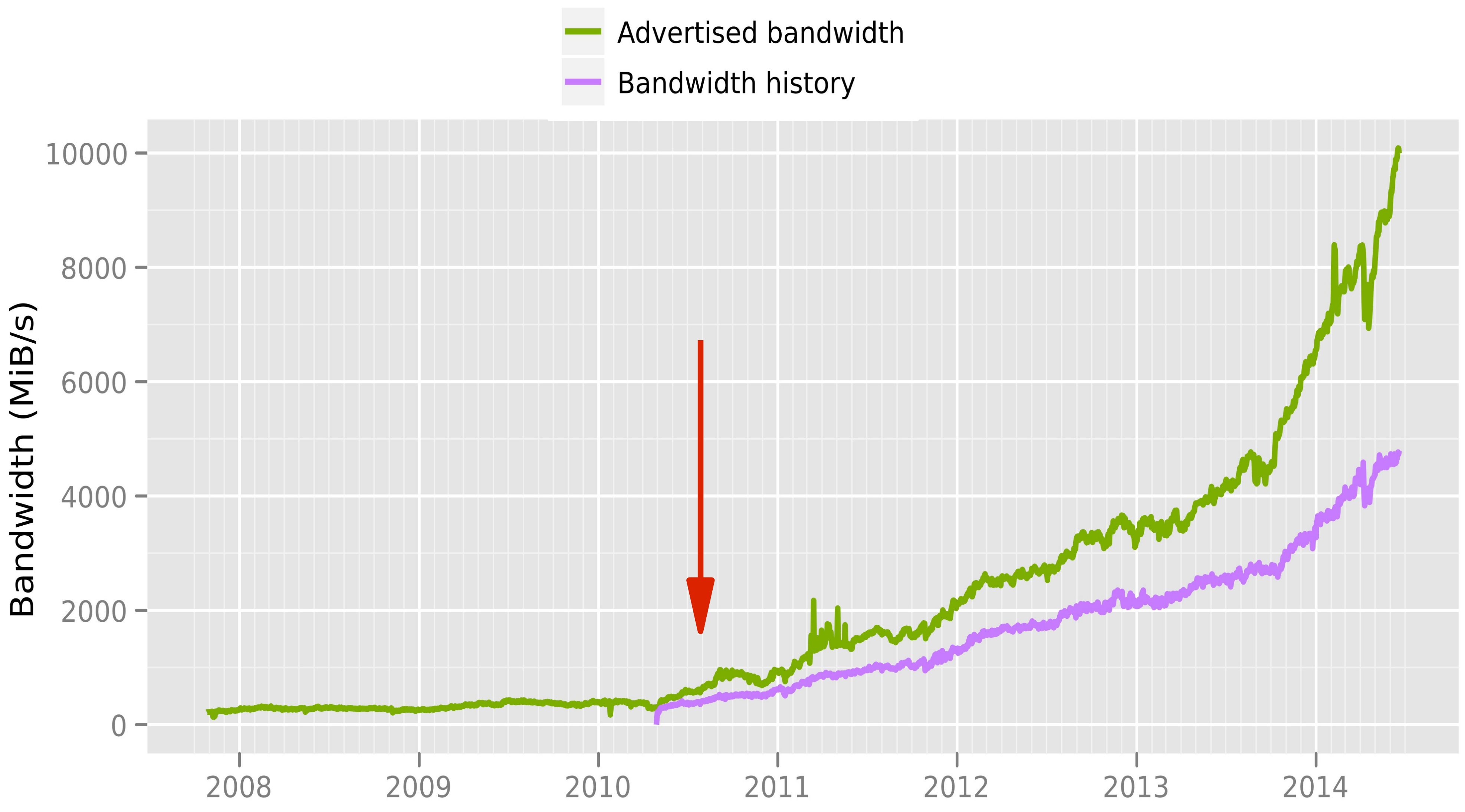
The Tor Project - https://metrics.torproject.org/

Number of relays



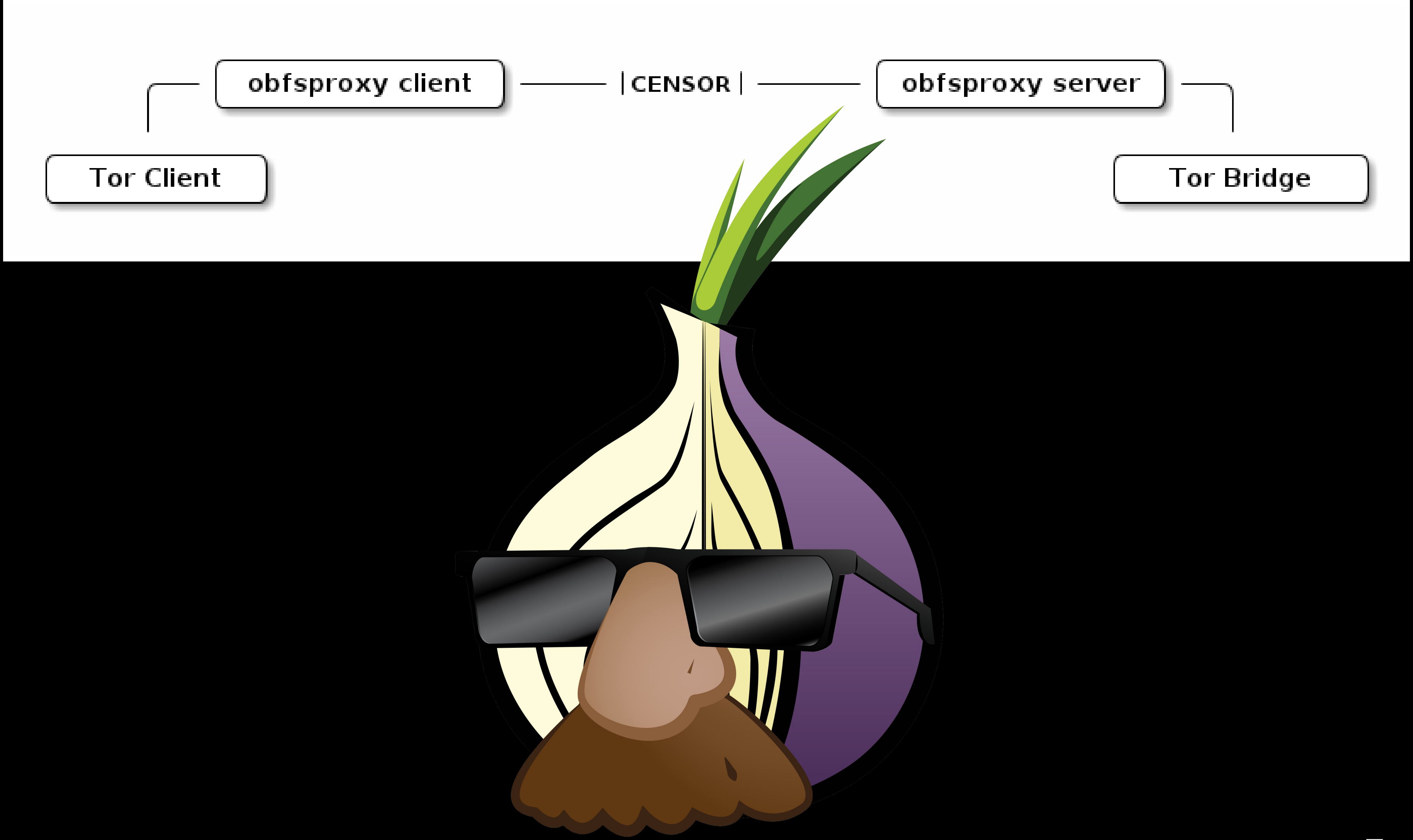
The Tor Project - https://metrics.torproject.org/

Total relay bandwidth



The Tor Project - https://metrics.torproject.org/

Deployed pluggable transports



Tor Client

obfsproxy client

|CENSOR|

obfsproxy server

Tor Bridge

Deployed #1: Obfs2

- Look-like-random-noise transport
- Part of Obfsproxy
- Broken and being deprecated

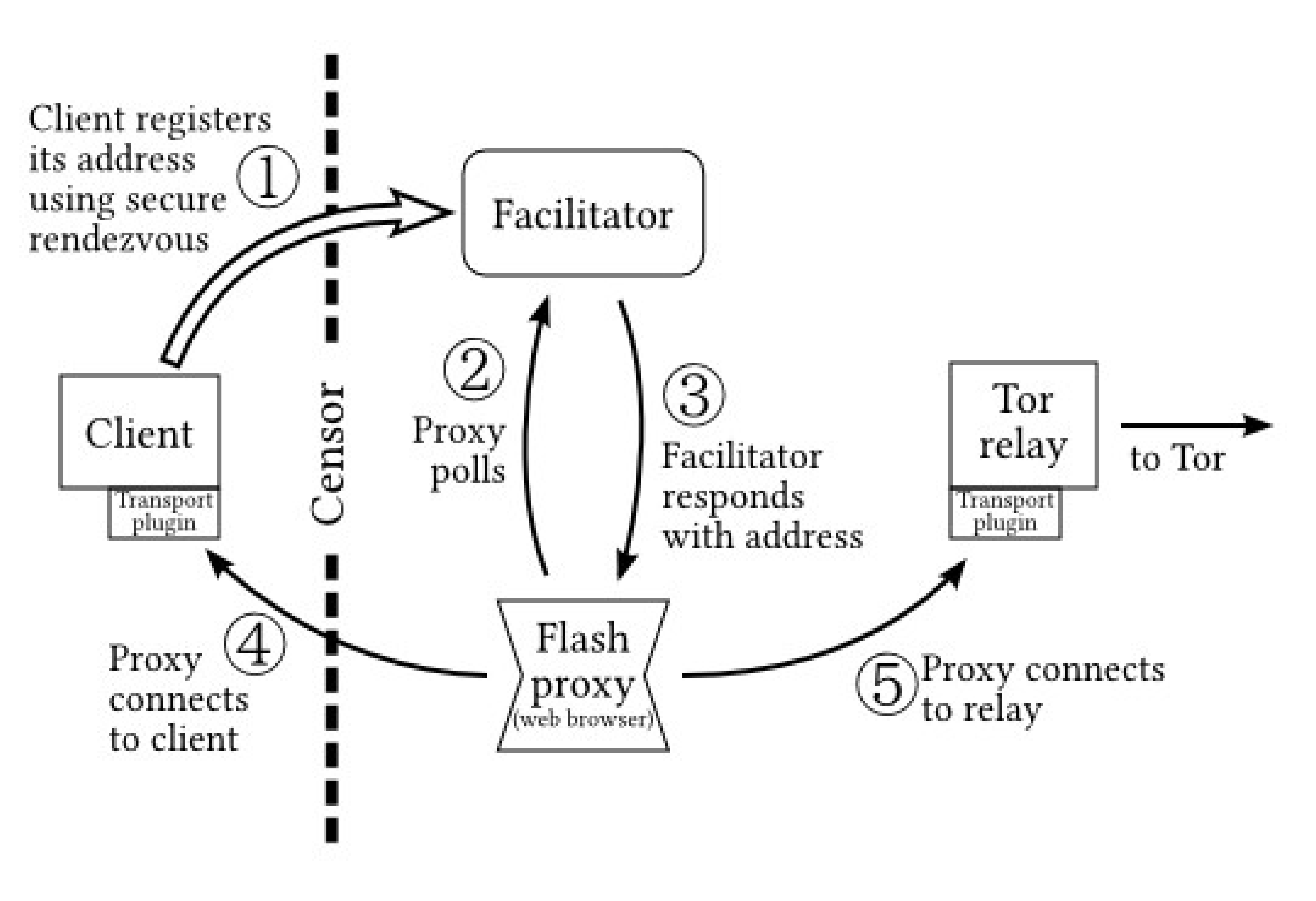
Deployed #2: Obfs3

- Look-like-random-noise transport
- Part of Obfsproxy
- Uses Diffie-Hellman key exchange to make DPI harder
- Current front-line protocol

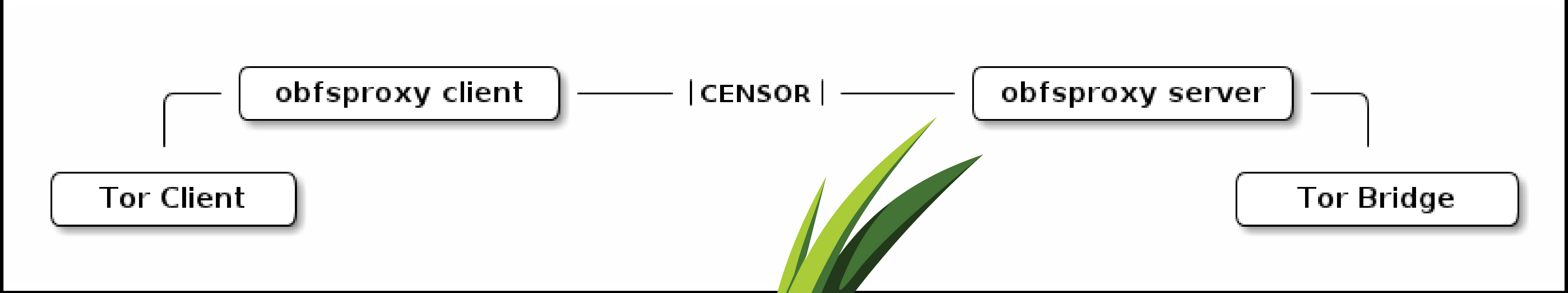
Deployed #3: FTE

- Look-like-something-else transport
- “Format-transforming encryption” + DFA to generate flows that match regexps
- But traffic statistics don't match...

  and it doesn't match the protocol?

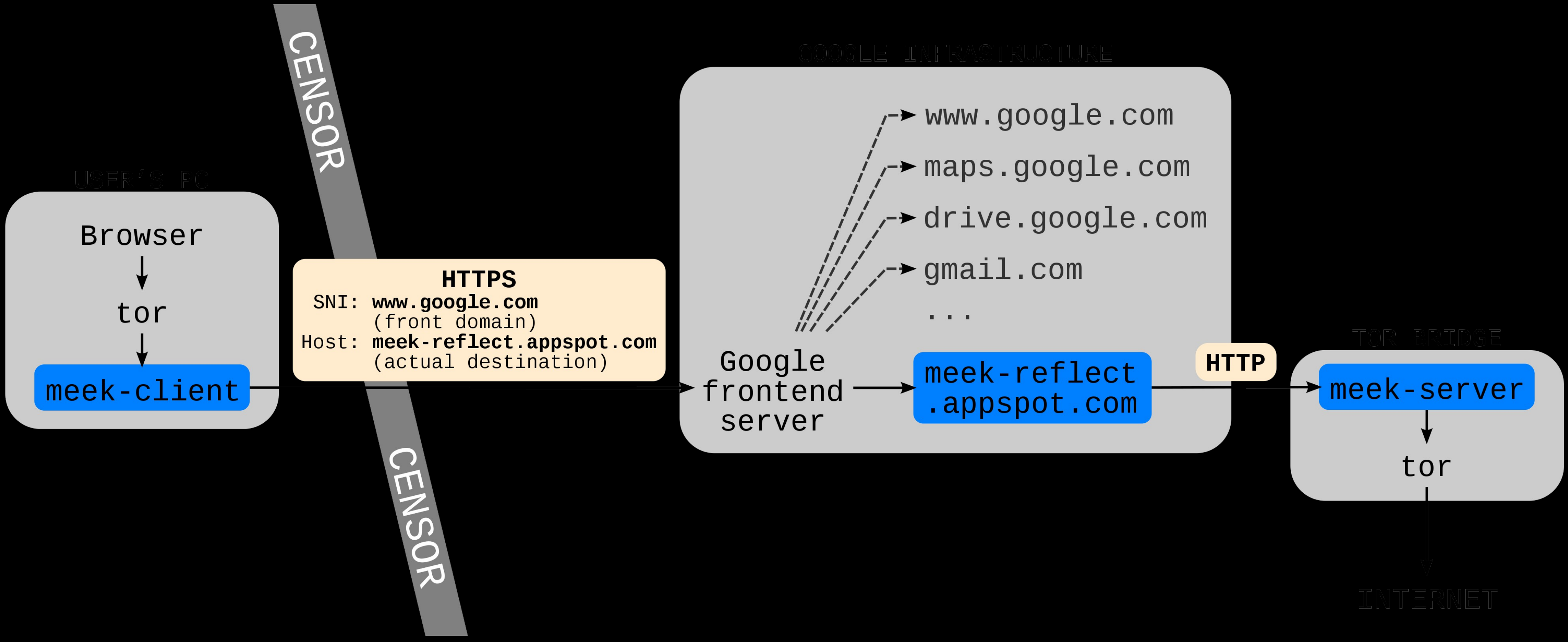

Client registers its address using secure rendezvous
①
Facilitator
Censor
②
Proxy polls
③
Facilitator responds with address
Client
Transport plugin
Tor relay
Transport plugin
to Tor
④
Proxy connects to client
Flash proxy
(web browser)
⑤
Proxy connects to relay

Near-term pluggable transports



ScrambleSuit and Obfs4

- Look-like-random-noise transport
- Use a shared secret between client and bridge to stymie active probing attacks
- Can do traffic padding to reduce traffic signatures, or inter-packet delays
- Obfs4 uses ECC (djb-crypto + NTor)



USER'S PC

Browser

tor

meek-client

CENSOR

CENSOR

HTTPS

SNI: **www.google.com**
(front domain)
Host: **meek-reflect.appspot.com**
(actual destination)

GOOGLE INFRASTRUCTURE

www.google.com
maps.google.com
drive.google.com
gmail.com
...

Google
frontend
server

meek-reflect
.appspot.com

HTTP

TOR BRIDGE

meek-server

tor

INTERNET

“Fronting”

Drives Tor Browser for realistic TLS

- Google
- Amazon S3
- Cloudflare
- Akamai
- Azure

Not deployed #1: SkypeMorph

- Look-like-something-else transport
- Characterize Skype traffic, generate flows that statistically match them

Not deployed #2: HexChat

- Route through XMPP

Not deployed #3: StegoTorus

- Look-like-something-else transport
- E.g. embed content in web objects
- Client side embeds in e.g. json, cookies, headers, etc
- Bridge side needs a library of objects

Not deployed #4: uProxy

- Google + UW collaboration
- Discovery: Google Plus contacts
  - But only one hop away (abuse)
- Transport: WebRTC (udp + sctp)

Not deployed #5: Dust

- Look-like-nothing transport
- Generates UDP packets with widely varying characteristics

Not deployed #6: Decoy Routing

- Route toward “innocent” destinations

Attack #1: Address enumeration

- Break into bridge authority
- Solve challenges from BridgeDB
- Vulnerable: everything that uses a standard Bridge line
- Immune: meek, flashproxy

Attack #2: Active probing

- Vulnerable: obfs2, obfs3, fte, flashproxy (pointless?)
- Immune: obfs4, ScrambleSuit

Attack #3: Broad DPI

- Accepts high collateral damage
- E.g. blocking flows based on packet entropy
- Vulnerable: obfs2, obfs3, obfs4, ScrambleSuit
- Immune: meek, flashproxy, fte (?), StegoTorus

Attack #4: Protocol DPI

- Attacks to determine the protocol that's in use
- Vulnerable: obfs2, flashproxy (?)
- Immune: obfs3, obfs4, ScrambleSuit, meek, fte, StegoTorus

Attack #5: Parrot DPI

- Attacks to distinguish the apparent protocol from the underlying one
- Vulnerable: fte, SkypeMorph

Attack #6: Protocol whitelisting

- Only allow known protocols through. Includes Iran's aggressive throttling of unknown protocols.
- Vulnerable: obfs2, obfs3, obfs4, ScrambleSuit
- Immune: depends on whitelist config

Attack #7: Cut long connections

- Terminate/throttle non-whitelisted protocols after 60s
- Vulnerable: obfs2, obfs3, obfs4, ScrambleSuit, fte
- Immune: meek, StegoTorus, flashproxy (?)

Attack #8: Flow fingerprinting

- Determine underlying protocol by e.g. timing, data transfer size, etc
- Vulnerable: obfs2, obfs3, meek, fte(?), flashproxy
- Mitigated: ScrambleSuit, obfs4
- Immune: StegoTorus (?)

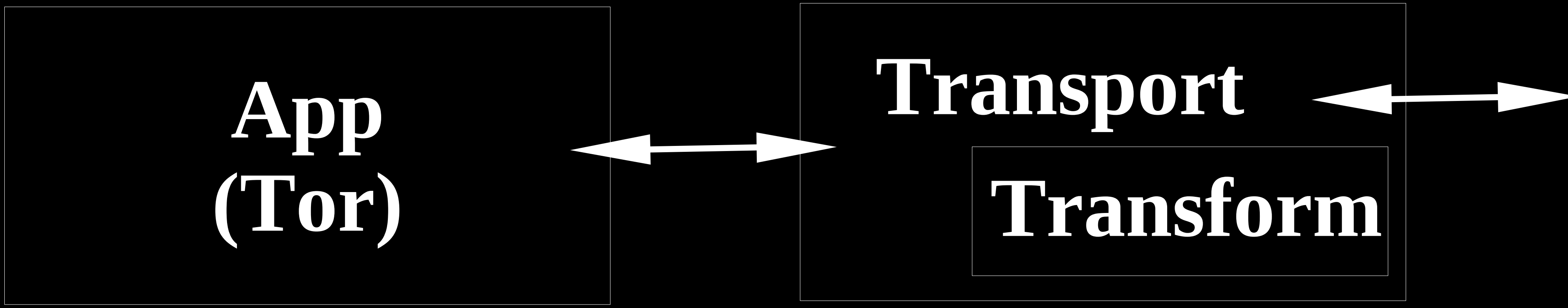
Other angles (1)

- Triangle Boy

Other angles (2)

- Fog – pluggable transport combiner. E.g. obfs3 + flashproxy
- Obfs4 bridge lines are a UX disaster

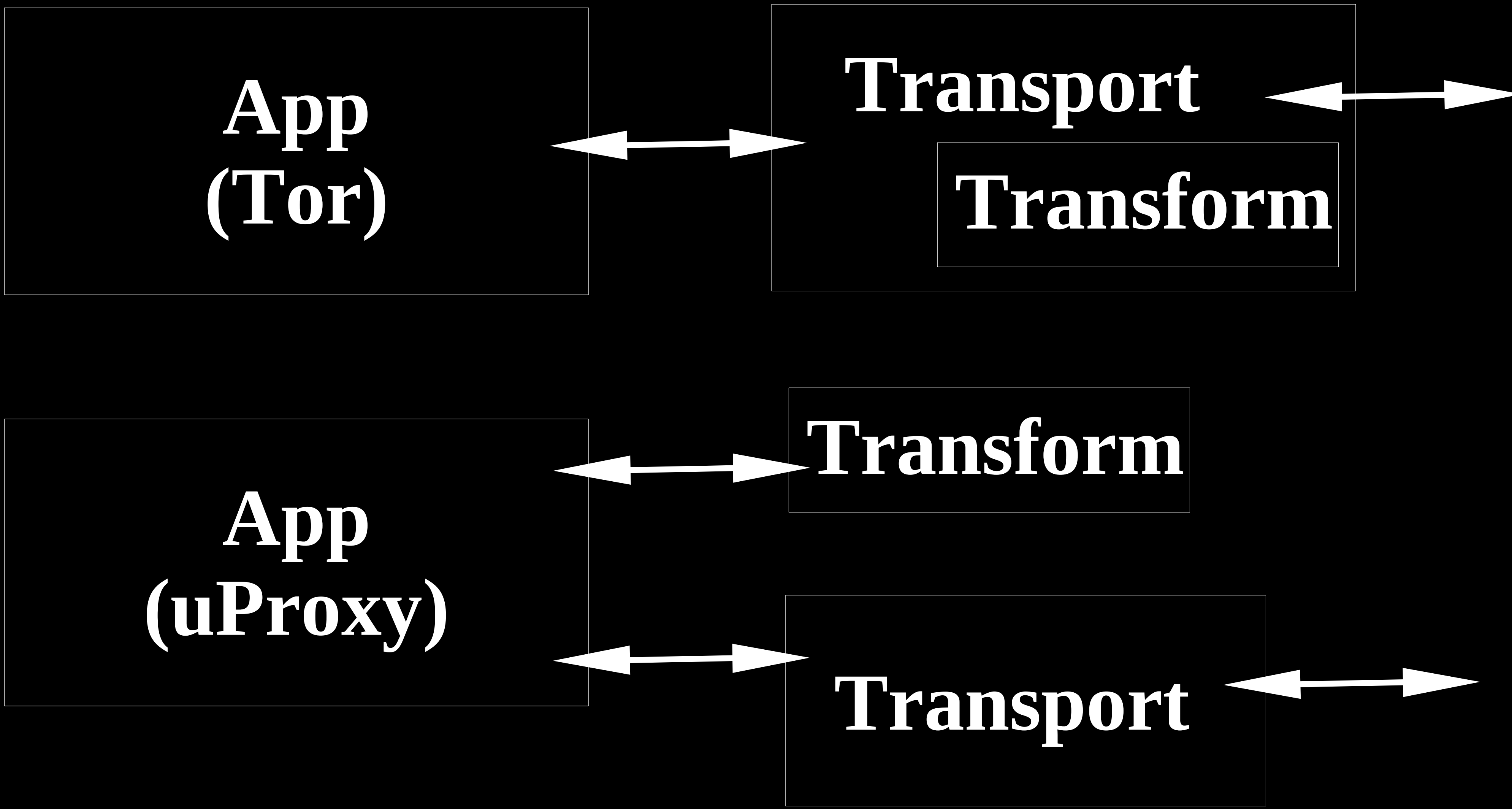
Composing and layering


App
(Tor)

Transport

Transform

Composing and layering



App
(Tor)

Transport

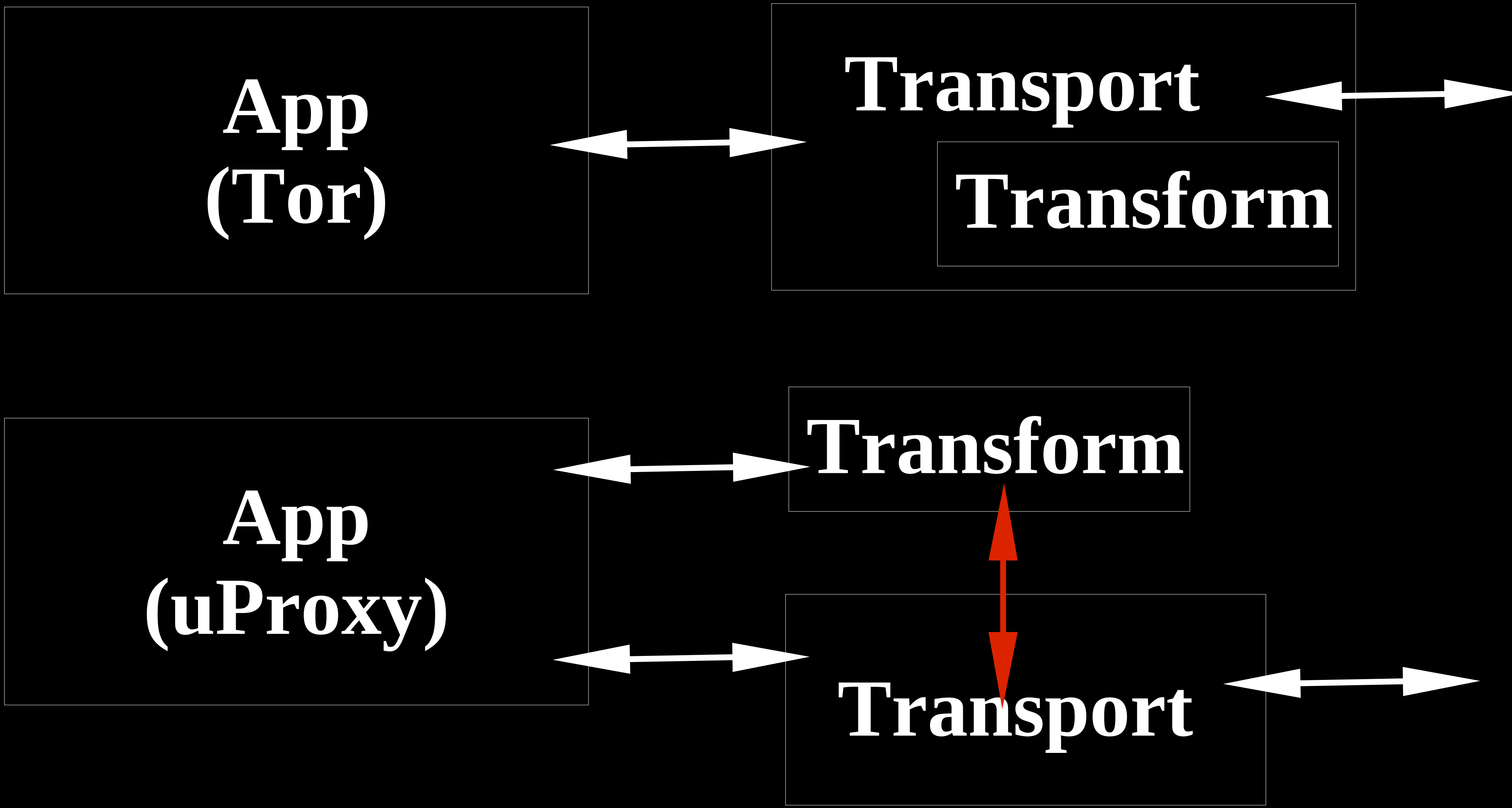
Transform

App
(uProxy)

Transform

Transport

Composing and layering



Other angles (3)

- Mobile: obfs4 is in Go, obfsclient is in C++, obfsproxy is in Python

Other angles (4)

- Host-based censorship
- “Lawful intercept” mandate

Other angles (5)

- Russia's Tor “contest”
- Black Hat / CERT talk

Other angles (6)

- Ad fetches in Tor Browser can harm anonymity

Other angles (7)

- Flashproxy as a savior vs Global surveillance?

Other angles (8)

- Don't forget effort involved in deployment arms race

Measurement Lab / Adversary Lab

- We need a set of benchmarks (“Iran 2011”) to test against – real attacks that we want to know how a given design fares against
- Background traffic issue
- Assessment needs to describe attributes, not conclusions. “China can't block this” vs “An adversary who does X would choose not to block this”

Measurement Framework

Need to extend the framework to include:

- Probing / active attacks
  - We need probe vectors! Skype connections, web connections, Tor connections, etc
- Pass traffic through transparent proxies

OONI:
Measuring interference in the wild

- Measuring censorship of destinations and protocols
- But just as importantly, preemptively tracking which protocols work where

Big open questions (1)

- Resisting address enumeration attacks

Big open questions (2)

- What protocols/services will remain open?

Big open questions (3)

- Who should be the exit relays?
- (For Tor, for uProxy, etc)

Big open questions (4)

- Realism of parrot attacks?
- FTE should be resistant, but in practice is incredibly vulnerable

Big open questions (5)

- Centralization of bridge operation?
- Or of blending services

Big open questions (6)

- What do we do when protocol whitelist + tls mitm?
- What other plausible censorship scenarios is our toolkit unprepared for?

Big open questions (7)

- Facebook is in Western jurisdiction
- Censorship encourages users to switch to Chinese Facebook equivalent
- ...outside of Western control
- Threatening information dominance

Big open questions (7)

- Facebook is in Western jurisdiction
- Censorship encourages users to switch to Chinese Facebook equivalent
- ...outside of Western control
- Threatening information dominance
- National security tie-in?