

Cypherpunks write code

Hacking on Tor related projects

\$ whoarewe

Arturo “hellais” Filastò

- Working mainly on censorship detection and measurements (OOONI)
- A Random GlobaLeaks Developer

Aaron “aagbsn” Gibson

- Working mainly on the Tor infrastructure
- Bridge distribution
- Anti-censorship related issues

What does the Tor Project do?

- Help people access information Anonymously (**Tor**)
- Help people publish information Anonymously (**Tor Hidden Services**)
- Help people circumvent censorship (**Bridges**, **Obfsproxy**)
- Measure censorship across the world (**OONI**)

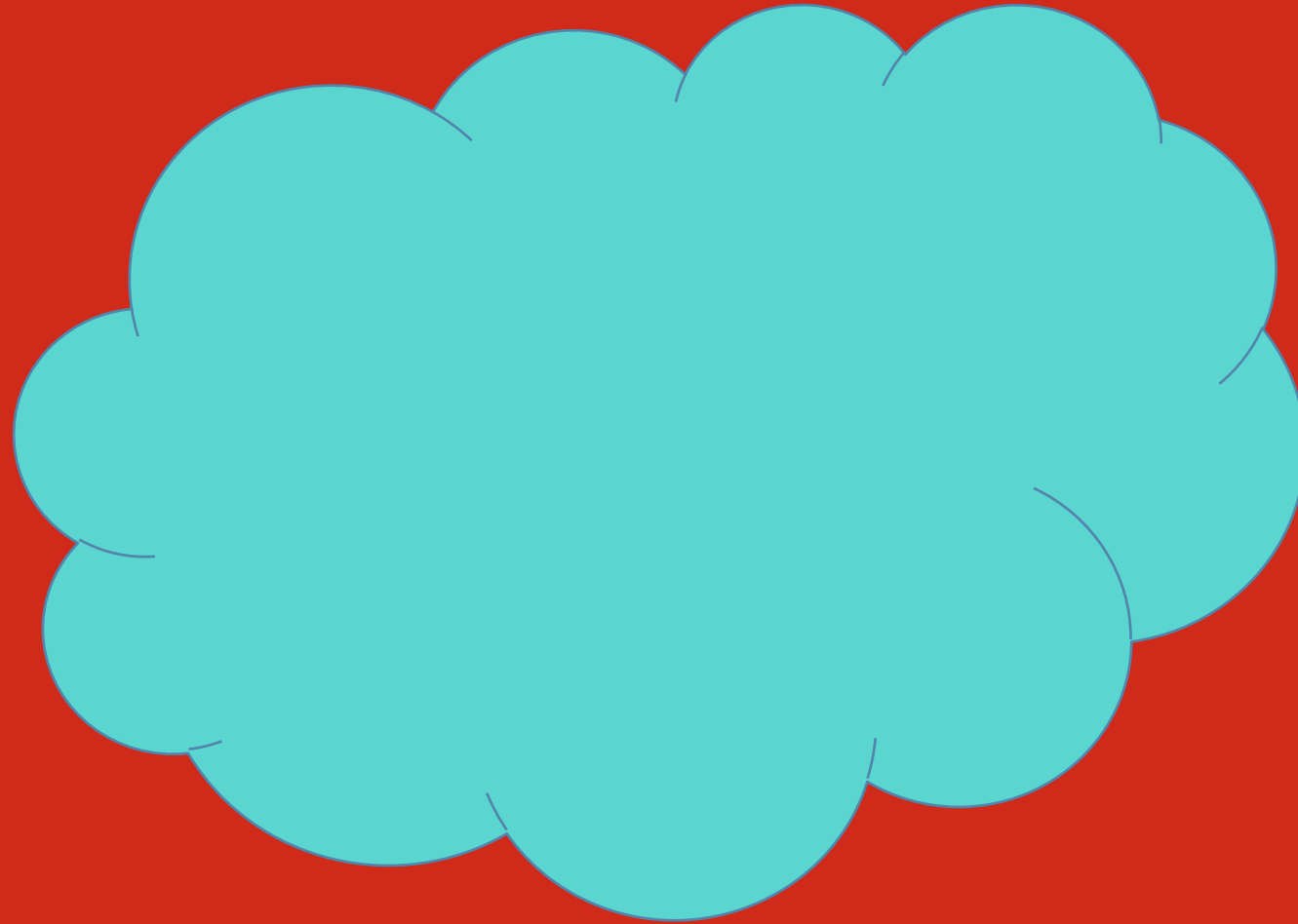
A brief intro

Some of the lesser known parts of the Tor Network

The Tor Architecture



**Tor
Client**



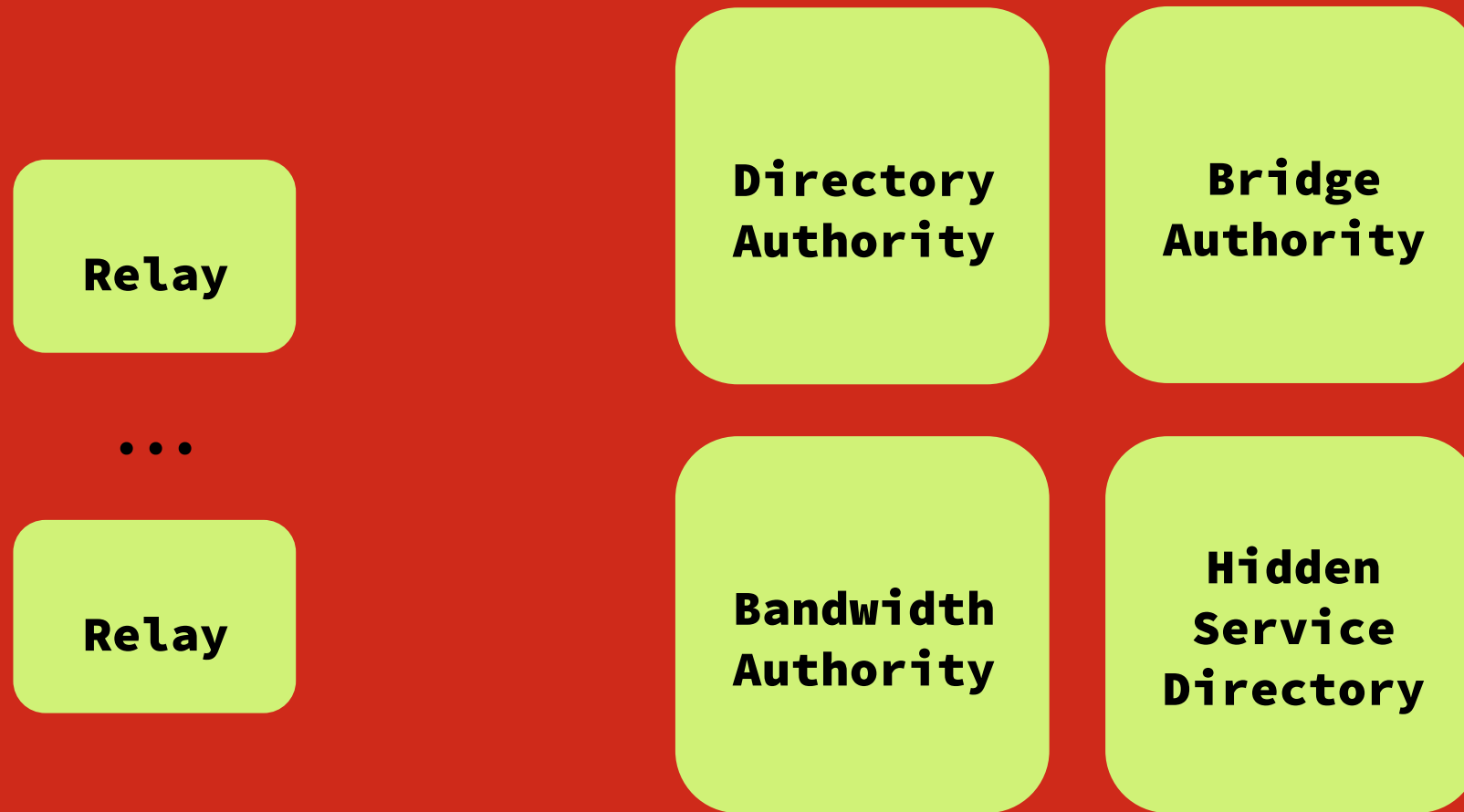
The Tor Architecture



**Tor
Client**



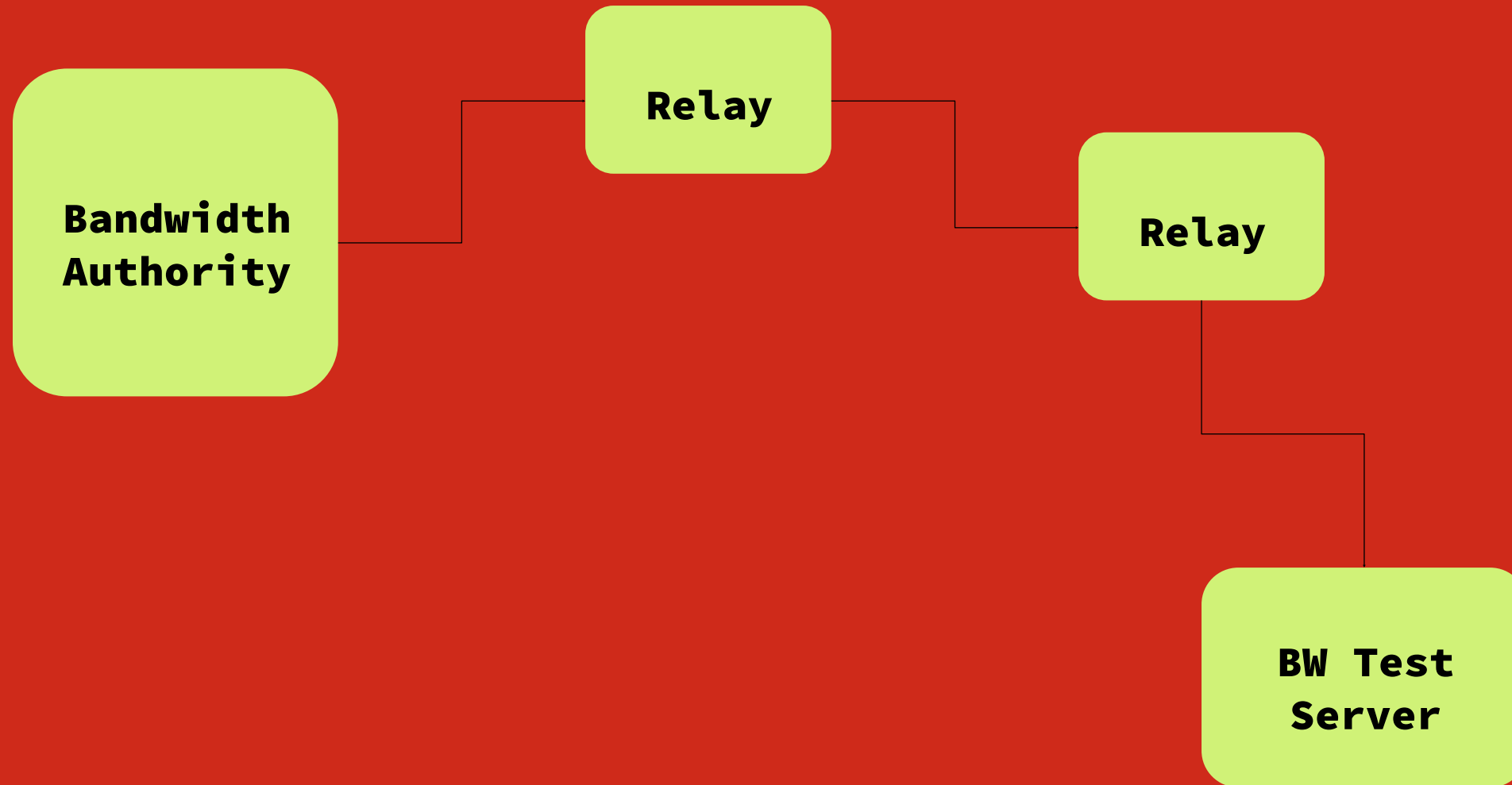
The Tor Architecture



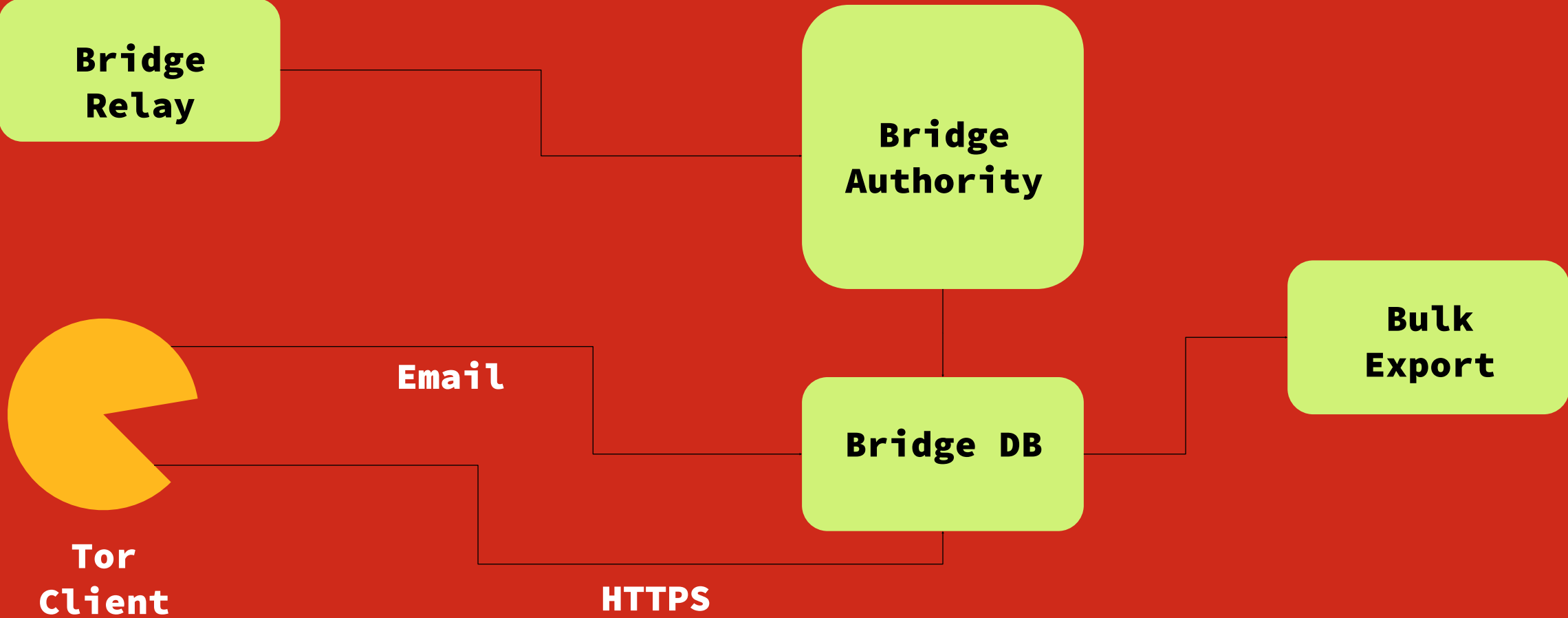
Directory Authority

- There are currently 9 Directory Authorities
- The core ones are shipped with every Tor binary
- Used as root of trust
- Discovery of the network
- DA's vote on stuff

Bandwidth Authority



Bridge Authority



Projects

Hope some will excite you!

Tor Button

- Is a Firefox extension that torifies your connections
- Currently Tor Button is a component of Tor Browser Bundle and should not be used as a standalone plugin.
- <https://www.torproject.org/torbutton/en/design/>

TorBirdy

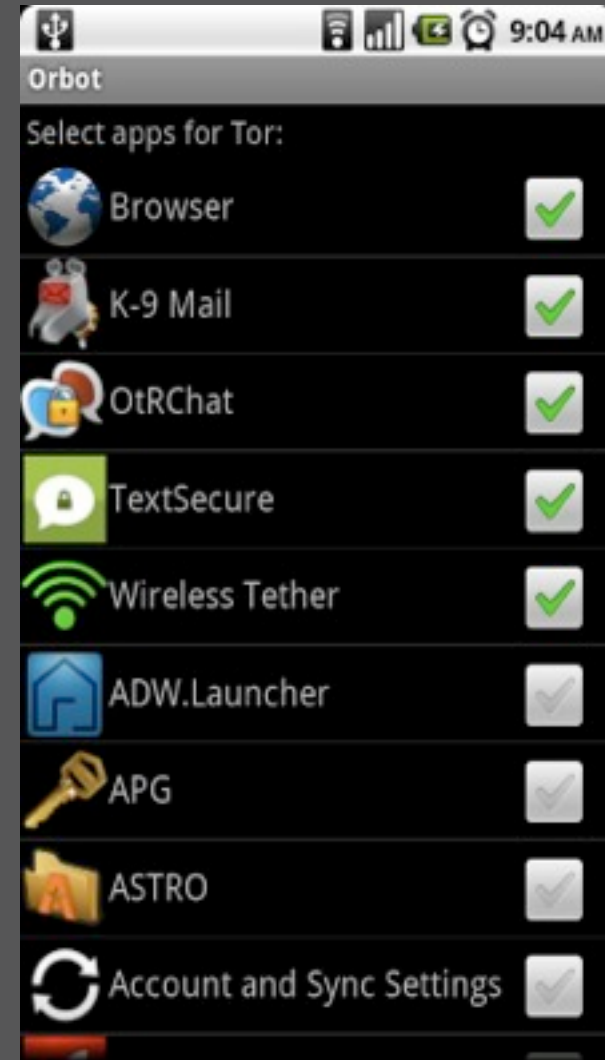
- Used to torify Thundebird
- It's a **Thunderbird Extension**
- <https://github.com/ioerror/torbirdy>
- There are some open tickets in ThunderBird bug tracker

Vidalia

- Written in C++
- Based on QT
- Is the default Tor GUI

Orbot

- Written in **Java**
- Android Tor controller allowing you to Torify apps on your phone



Tor Browser Bundle

- Vidalia + Tor Button + Firefox + Build automation
- Includes patches for Firefox (some of which are not going to be merged upstream ☹)

Arm

- An ncurses based interface to Tor
- Written in Python
- Based on Stem

```
arm - odin (Linux 2.6.28-18-generic) Tor 0.2.1.19 (unknown) cpu: 0.6% mem: 49 MB (1.2%) pid: 4714 uptime: 10-22:26:36
caerSidi - 76.104.132.98:9001, Control Port (password): 9051 fingerprint: A7569483857064B181A9C852EFF702D32E4553EB
flags: Fast, HSDir, Named, Running, Stable, Valid exit policy: reject "*"

page 1 / 3 - q: quit, p: pause, h: page help
Bandwidth (cap: 40 KB, burst: 100 KB)
Downloaded (506 bytes/sec - avg: 13.2 KB/sec, total: 11.8 GB):
Uploaded (506 bytes/sec - avg: 13.3 KB/sec, total: 11.9 GB):

Accounting (awake) Time to reset: 150:10:02
10 GB / 30 GB 10 GB / 30 GB

Events (INFO, BW):
18:49:57 [INFO] router pick published address(): Could not determine our address locally. Checking if directory headers provide any hints.
18:49:57 [INFO] resolve_my_address(): Address 'odin' resolves to private IP address '127.0.1.1'. Tor servers that use the default DirServers must have public IP addresses.
18:49:57 [INFO] resolve_my_address(): Interface IP address '192.168.1.20' is a private address too. Ignoring.
18:49:57 [INFO] resolve_my_address(): Guessed local hostname 'odin' resolves to a private IP address (127.0.1.1). Trying something else.
18:49:56 [BW] READ: 506, WRITTEN: 506
18:49:56 [BW] READ: 0, WRITTEN: 0
18:49:55 [BW] READ: 0, WRITTEN: 1758
18:49:54 [BW] READ: 1172, WRITTEN: 0
18:49:53 [BW] READ: 0, WRITTEN: 0
18:49:52 [INFO] router pick published address(): Could not determine our address locally. Checking if directory headers provide any hints.
18:49:52 [INFO] resolve_my_address(): Address 'odin' resolves to private IP address '127.0.1.1'. Tor servers that use the default DirServers must have public IP addresses.
18:49:52 [INFO] resolve_my_address(): Interface IP address '192.168.1.20' is a private address too. Ignoring.
18:49:52 [INFO] resolve_my_address(): Guessed local hostname 'odin' resolves to a private IP address (127.0.1.1). Trying something else.
18:49:52 [BW] READ: 506, WRITTEN: 0
18:49:51 [BW] READ: 1172, WRITTEN: 1758
18:49:51 [INFO] circuit n conn done(): or conn failed. Closing circ.
18:49:50 [BW] READ: 1172, WRITTEN: 1172
18:49:49 [BW] READ: 1758, WRITTEN: 1758
18:49:48 [INFO] run connection housekeeping(): Expiring non-used OR connection to fd 364 (213.19.185.146:443) [Not in clique mode].
18:49:48 [BW] READ: 1172, WRITTEN: 2930
18:49:47 [BW] READ: 506, WRITTEN: 0
18:49:47 [INFO] router pick published address(): Could not determine our address locally. Checking if directory headers provide any hints.
18:49:47 [INFO] resolve_my_address(): Address 'odin' resolves to private IP address '127.0.1.1'. Tor servers that use the default DirServers must have public IP addresses.
18:49:47 [INFO] resolve_my_address(): Interface IP address '192.168.1.20' is a private address too. Ignoring.
```

txtorcon

- Written in **Python**
- Based on **Twisted**
- Provides functionality for interacting with the Tor Control port, starting and stopping of Tor clients, Hidden Services.
- All providing nice Twisted compatible interfaces

Atlas

- Is used to search and view details on Tor relays
- Written in Javascript
- Based on Backbone.js and require.js

Onionoo

- Provides the backend HTTP API to Atlas
- Written in **Java**
- There is also a WIP version of Onionoo called **PyOnionoo** written in **Python** based on **Twisted** (**cyclone**)

Metrics Portal

- Written in **Java**, **R** and **Python**
- Used to generate all the statistics and charts you see on metrics.torproject.org

TorFlow: Bandwidth Authority

- Builds 2 hop circuits through relays of similar capacity and measures throughput
- Implements PID feedback
- Results are fed to a corresponding Directory Authority
- Directory Authorities advertise the media bw value as the consensus bw
- Clients probabilistically select higher capacity relays

TorFlow: Exit Authority

- Detects content manipulation
 - Of HTTP, HTTPS, SSH, DNS
- Builds circuits through all Tor Exits and compares content
- Misbehaving exits are flagged
- Pitfalls: Does not scan dynamic websites

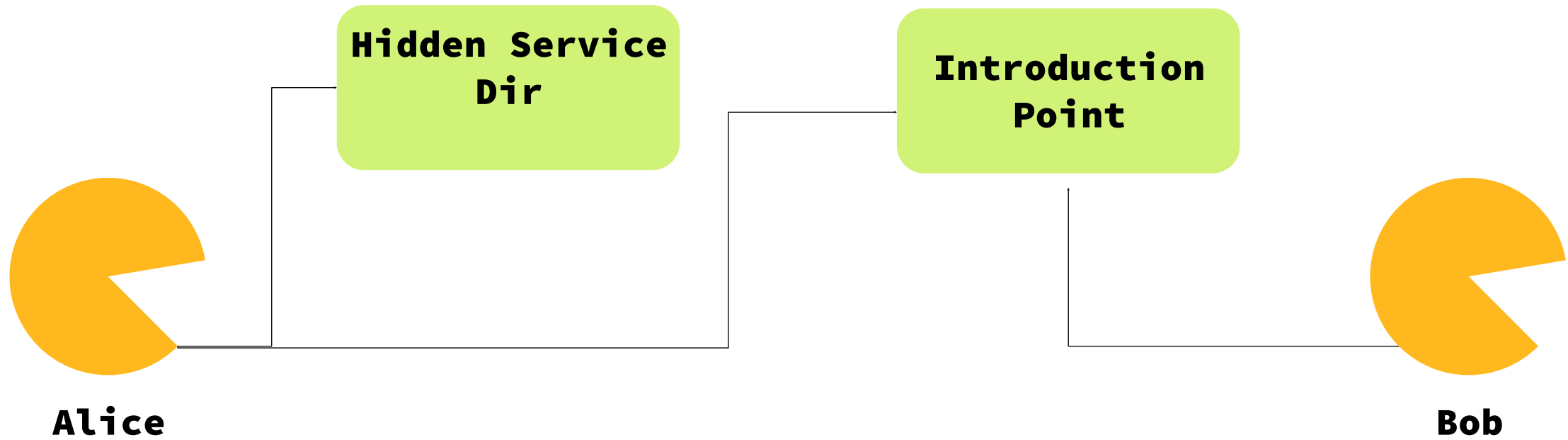
TorFlow: Other

- OpAddon, metatroller – modify Tor's path selection policy
- WARNING: May compromise your anonymity
- PathBias – Tools for measuring path bias

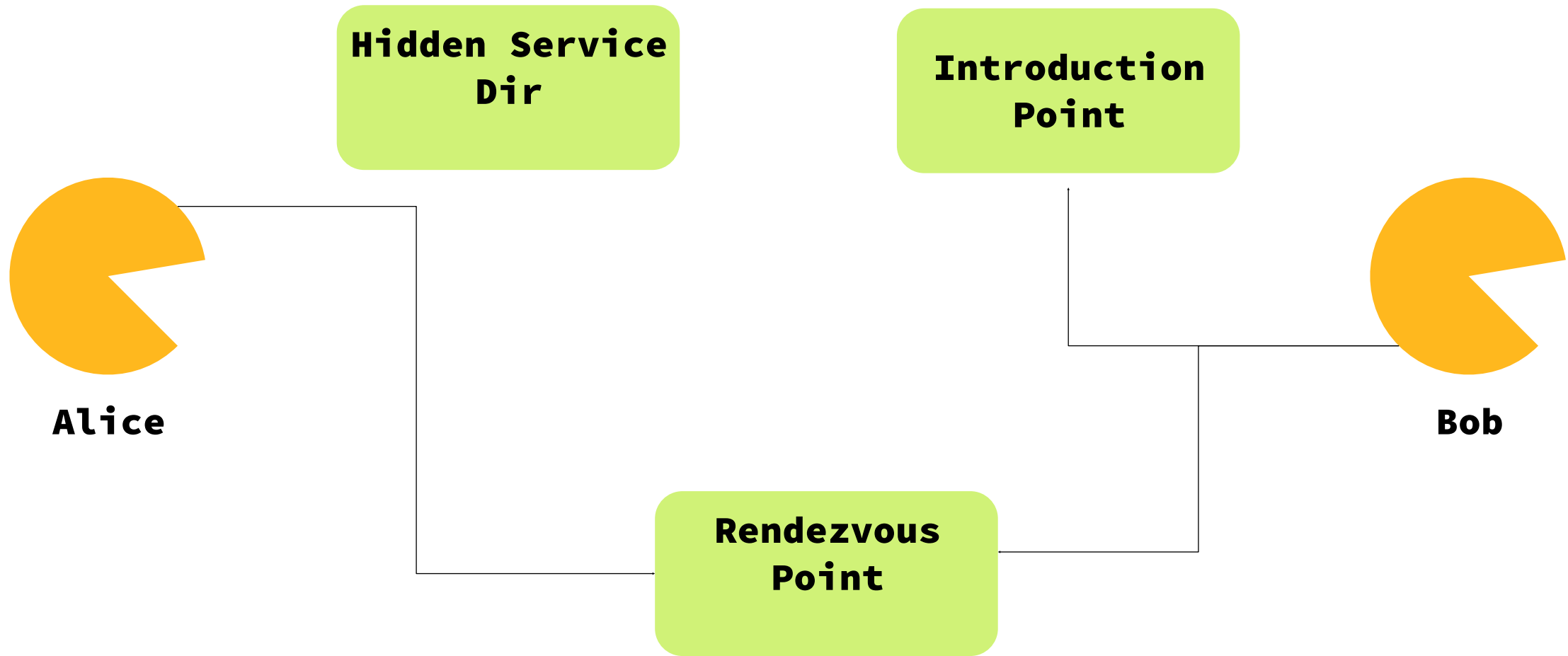
Tor Hidden Services

- Allows you to anonymously host server-side TCP services
- The .onion address is self authenticating
- Opens lot's of possibilities for self publishing

Tor Hidden Services



Tor Hidden Services



APAF: Anonymous Python Application Framework

- Written in **Python**
- Based on **Twisted** and **Storm**
- A build system for creating desktop oriented, **Tor Hidden Service** driven, python based **server side** applications

APAF: Anonymous Python Application Framework

- UI Related issues: <https://github.com/globalleaks/APAF/issues?labels=UserInterface&page=1&state=open>
- Security related issues: <https://github.com/globalleaks/APAF/issues?labels=Security&page=1&state=open>
- Enhancements: <https://github.com/globalleaks/APAF/issues?labels=enhancement&page=1&state=open>

Tor2web

- Makes HTTP based Tor Hidden Services accessible from the “surface web”
- Provides no anonymity for the client, but stills maintains anonymity for the publisher
- Written in Python
- Based on Twisted
- Some critical bugs:
 - Currently Internet Explorer does not work with tor2web

Shadow

- Simulates the Tor network
- Useful for testing and measurements
- Written in C
- Based on `foo` and `bar`

Tor and Censorship

- Tor is born as an Anonymity Tor
- Censorship circumvention is a side effect

Timeline of Tor censorship

- 2002 - Tor Source code released
- 2006 (April), Thailand - DNS filtering of tpo
- 2006, Websense/netfilter - Block Tor based on GET requests to Das
- 2007, Iran, Saudi - Block Tor thanks to Websense
- 2009, Iran throttles SSL
- 2009, Tunisia - Smartfilter to block all except 443 and 80
- 2009, China blocks public relays
- 2009, Tor Bridges are introduced
- 2010, China starts collecting and blocking bridges
- 2011, Iran by DPI on DH parameters of SSL
- 2011, Egypt selected targeted sites for
- 2011, Lybia throttling to limit use
- 2011, Syria, DPI on Tor's TLS renegotiation and killed connections
- 2011, Iran DPI on SSL and TLS certificate timeline
- 9 February 2012, Iran total SSL blockage
- 2012, China proactive censorship
- February - March 2012, Kazhakistan
- 22 May 2012, Ethiopia
- 25 June 2012, UAE Tor blocking via DPI
- Learn More: <https://trac.torproject.org/projects/tor/wiki/doc/OONI/censorshipwiki/>

OOONI-probe

- Written in **Python**
- Based on **Twisted** and **Scrapy**
- It aims at answering the questions:
 - What is censored?
 - Where is it censored?
 - How is it censored?

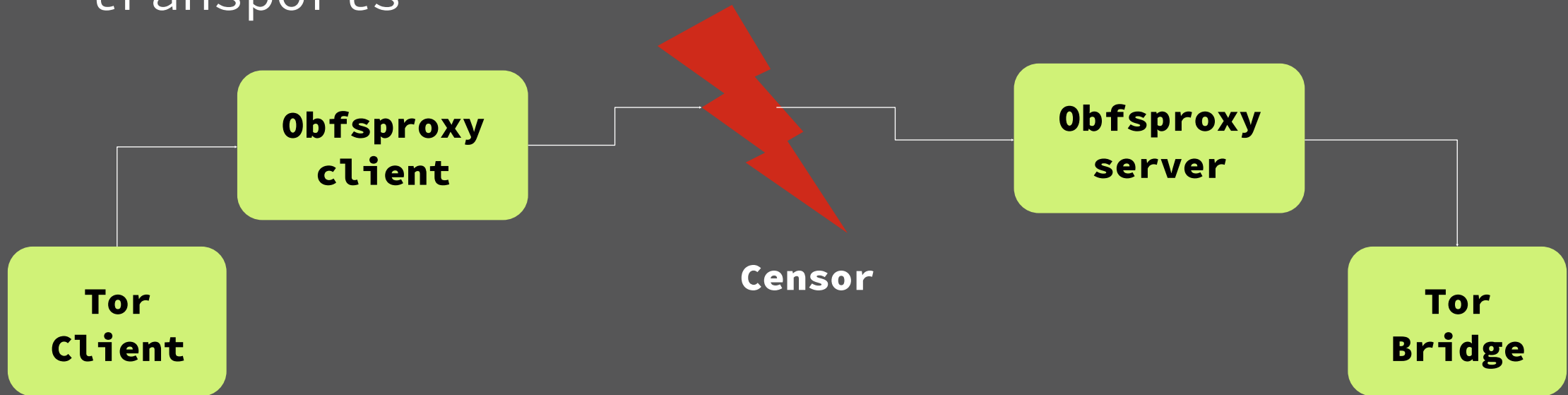
Tor Cloud

- The purpose is to simplify the setup of Tor Bridges



Obfsproxy

- A framework for creating pluggable obfuscated transports



Obfsproxy

- Useful for allowing Tor to circumvent censorship
- The bridge you are using must support your desired obfsproxy transport
- Written in C
- Based on libevent

BridgeDB

- Written in **Python**
- Uses **Twisted**
- Collects bridges and hands them out to clients
- It hands them out through distributors
 - Currently HTTPS, email and export to list

What next?

Come hack with us!

- We will have a hacking session in the workshop room, ping us if you are digging it!
- Join us on IRC:
irc.oftc.net #tor-dev
- Subscribe to the tor-dev and tor-talk mailing lists

Thanks for
Listening!

Questions?