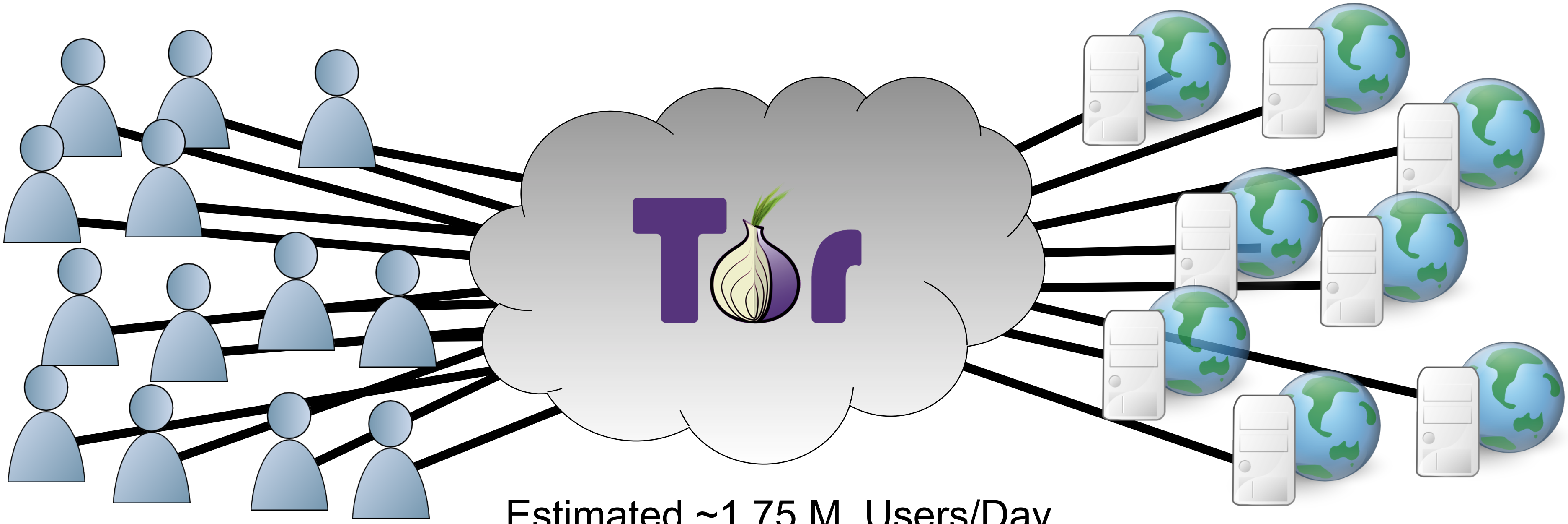
Safely Measuring Tor

"Safely Measuring Tor", Rob Jansen and Aaron Johnson, In the *Proceedings of the 23rd ACM Conference on Computer and Communication Security* (CCS 2016).

Rob Jansen
U.S. Naval Research Laboratory
Center for High Assurance Computer Systems

23rd Conference on Computer and Communication Security
Hofburg Imperial Palace, Vienna, Austria
October 27th, 2016

Talk Overview

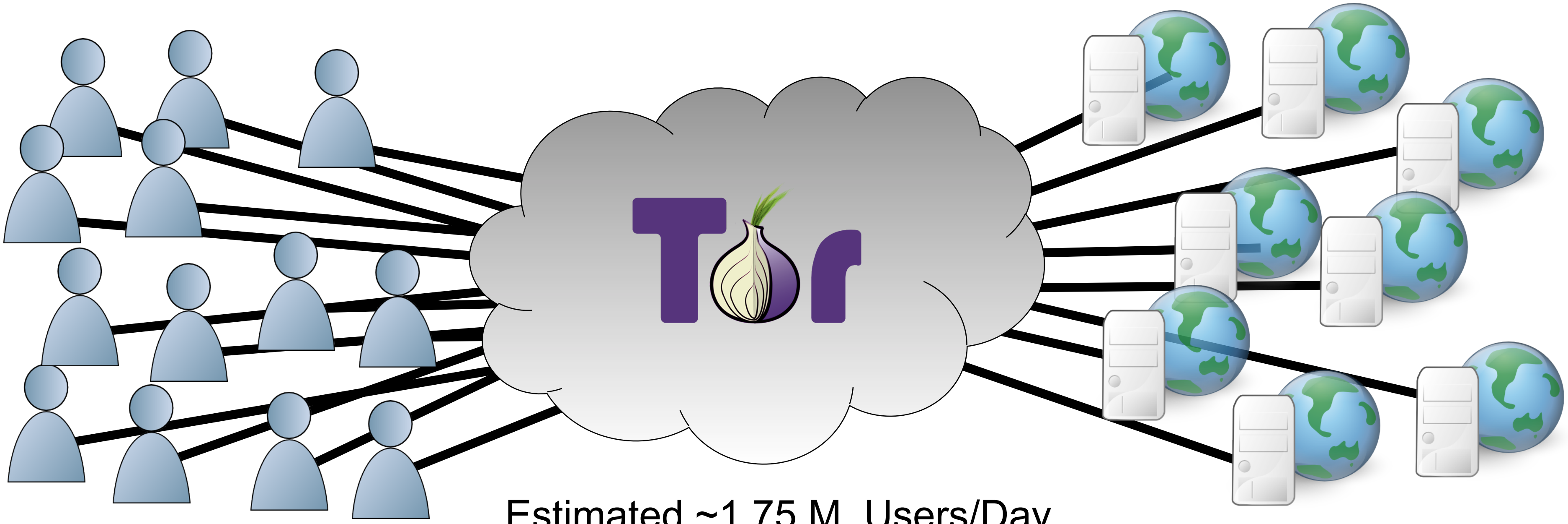


Estimated ~1.75 M. Users/Day
(metrics.torproject.org)

Tor: an anonymous communication, censorship resistant, privacy-enhancing communication system

- How is Tor being used? being misused? performing?

Talk Overview



Estimated ~1.75 M. Users/Day
(metrics.torproject.org)

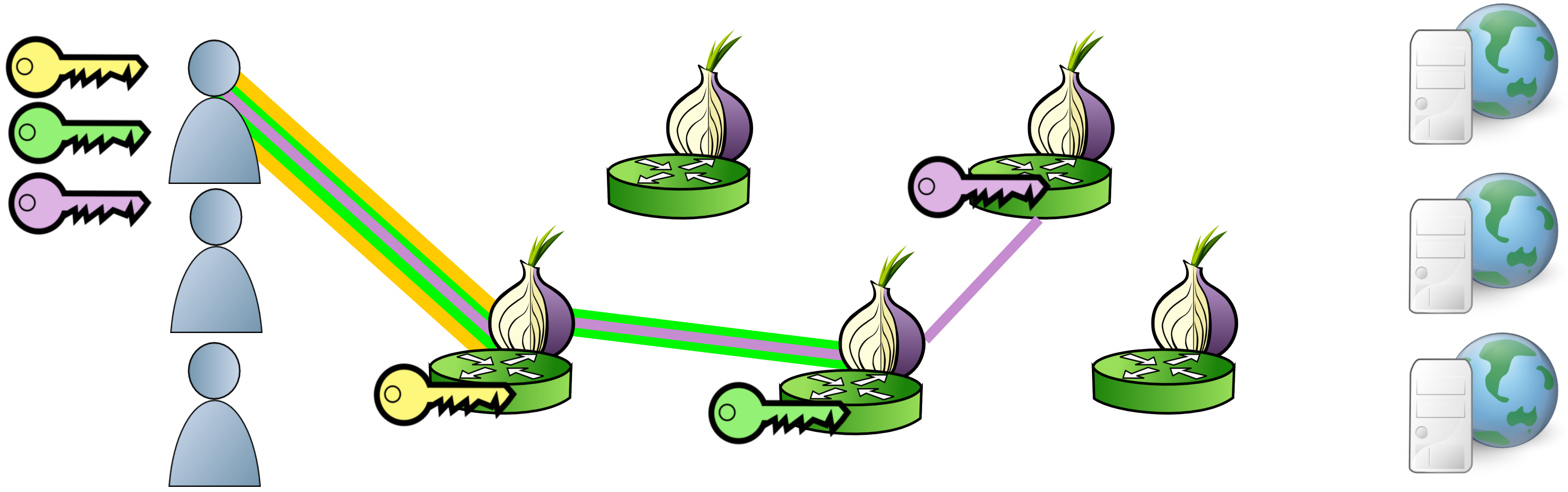
Tor: an anonymous communication, censorship resistant, privacy-enhancing communication system

- How is Tor being used? being misused? performing?
- Objective: To safely gather Tor network usage statistics
- Approach: Use distributed measurement, secure multiparty computation, and differential privacy

Background and Motivation

- How Tor works
- Why measurements are needed and what to measure
- Measurement challenges

Background: Onion Routing



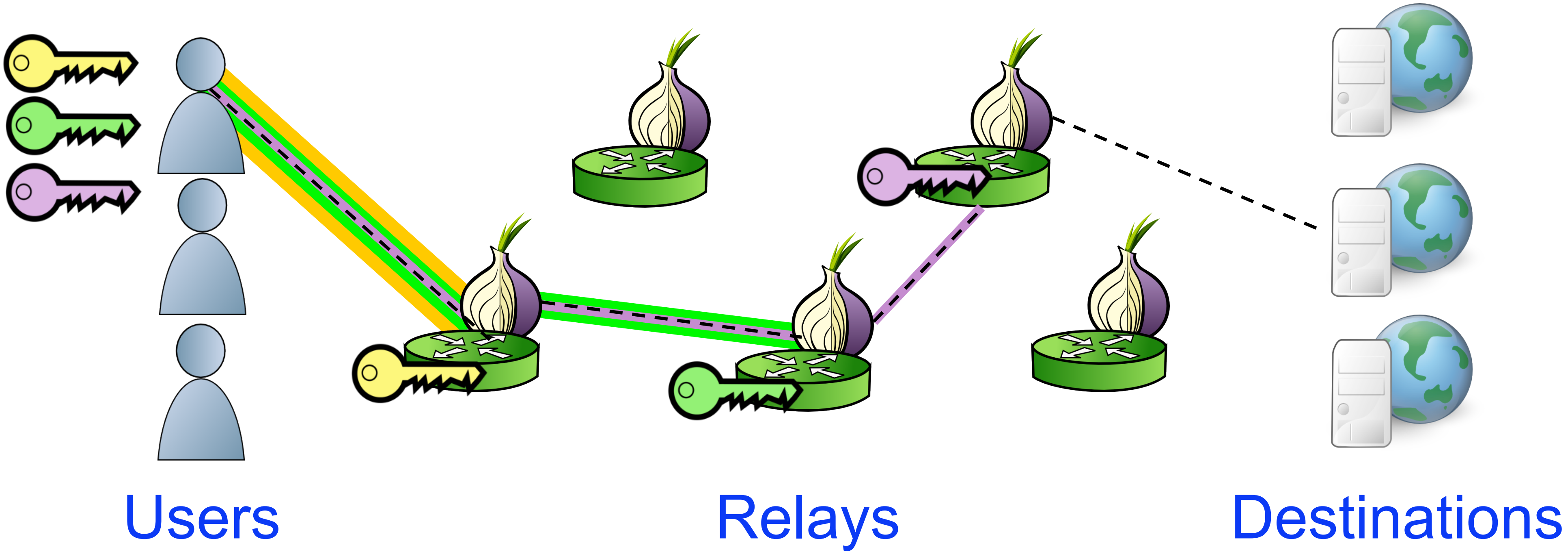
Users

Relays

Destinations

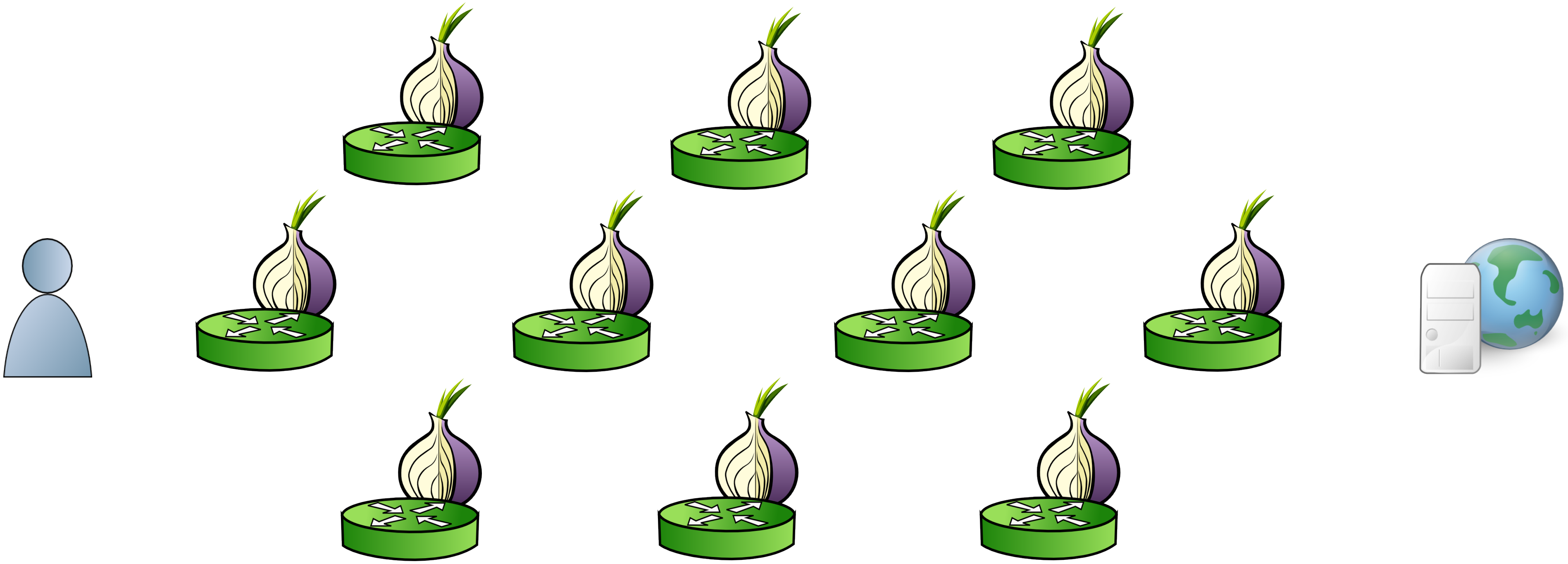
Circuit

Background: Onion Routing

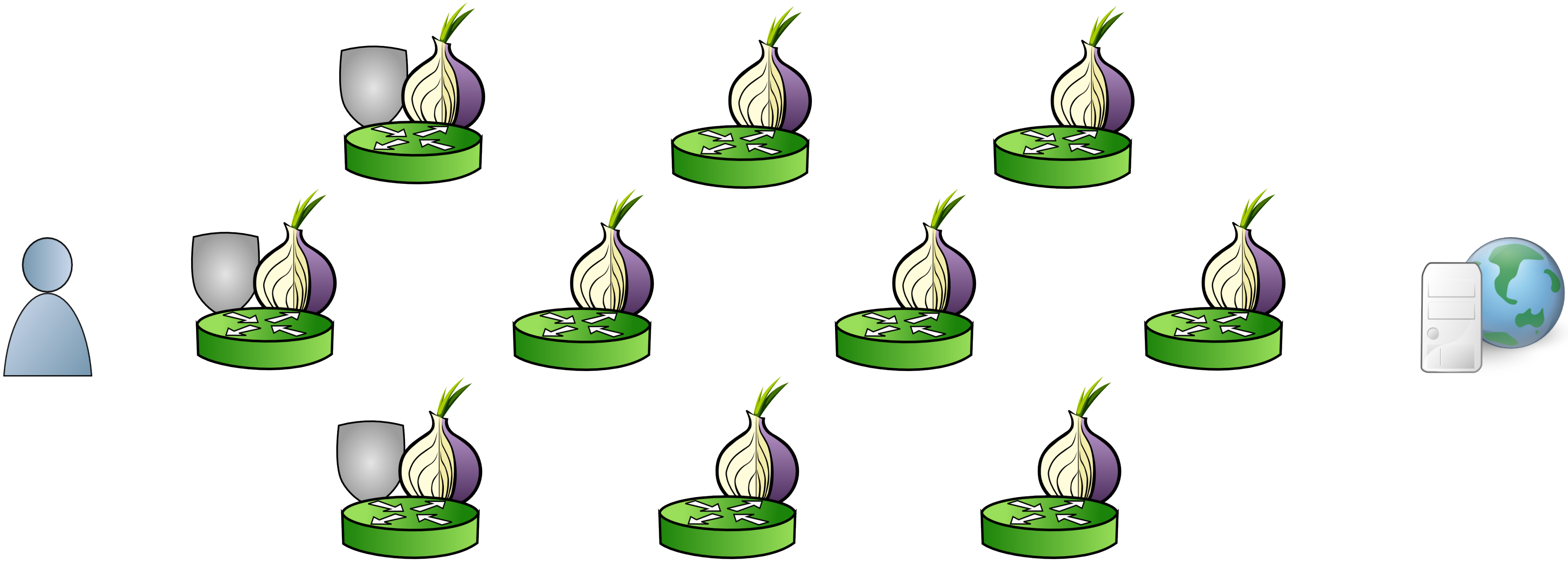


🟨	Circuit
- - - - - - - -	Stream

Background: Using Circuits

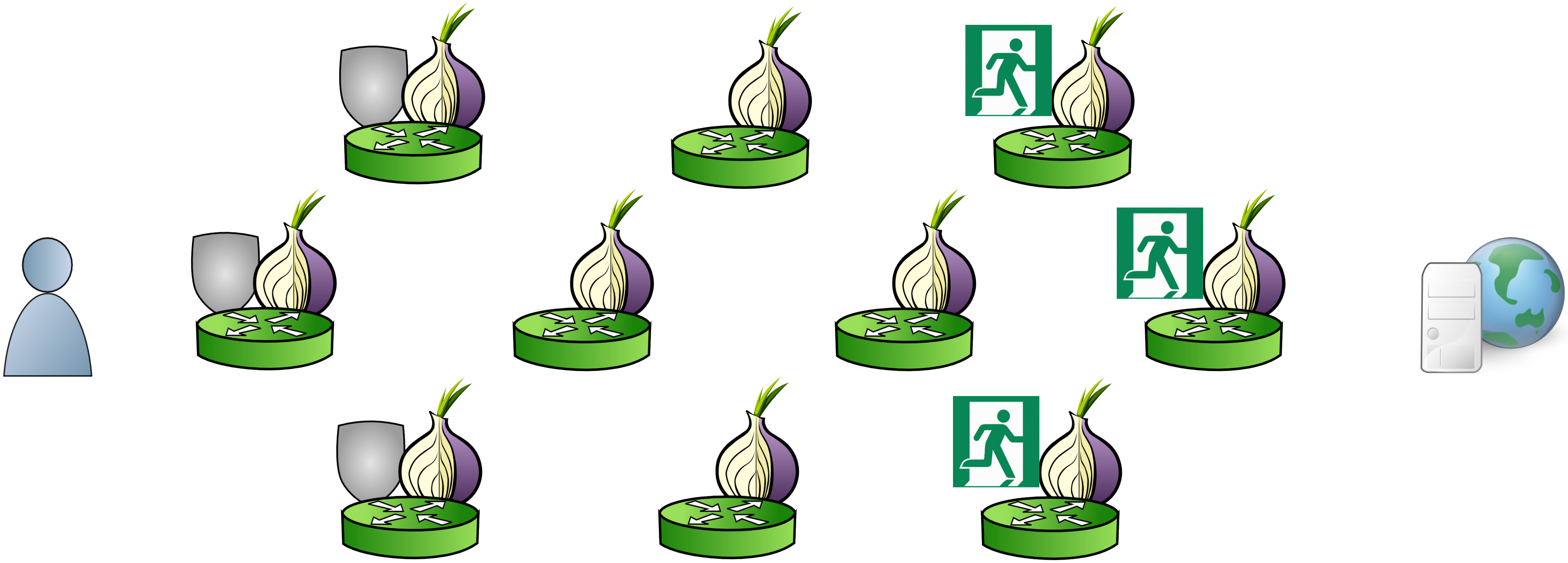


Background: Using Circuits



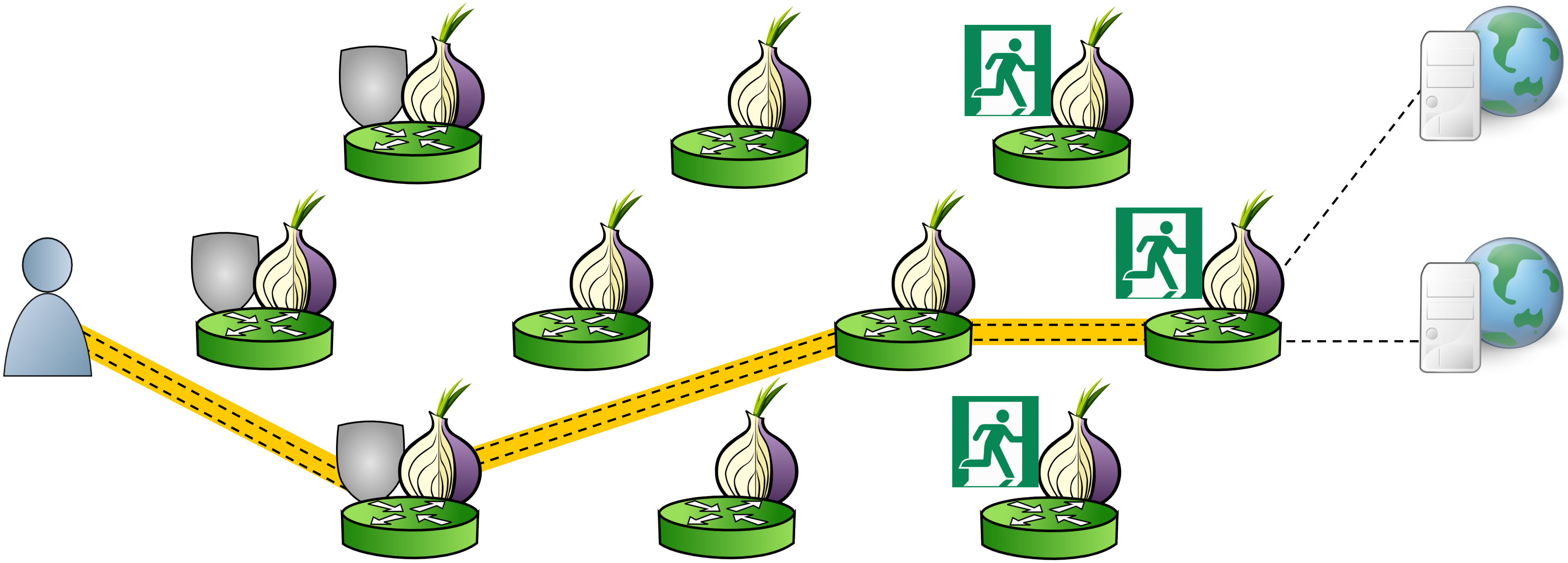
1. Clients begin all circuits with a selected guard

Background: Using Circuits



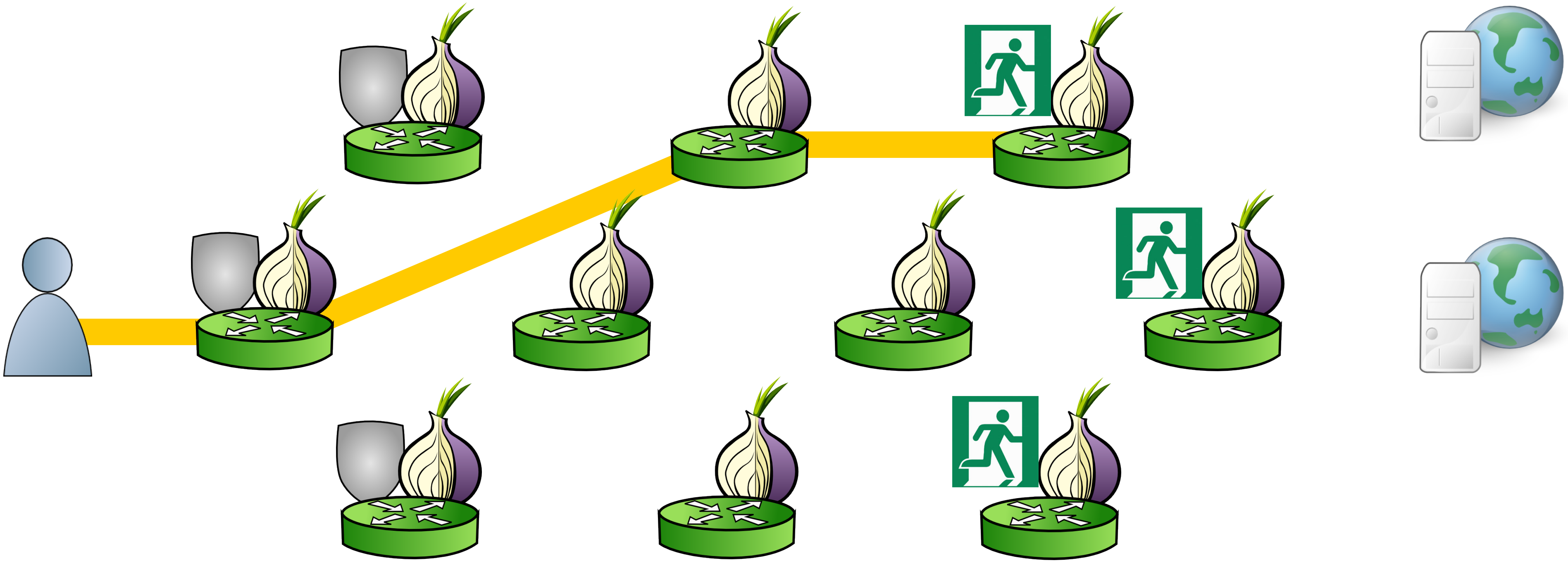
1. Clients begin all circuits with a selected guard
2. Relays define individual exit policies

Background: Using Circuits



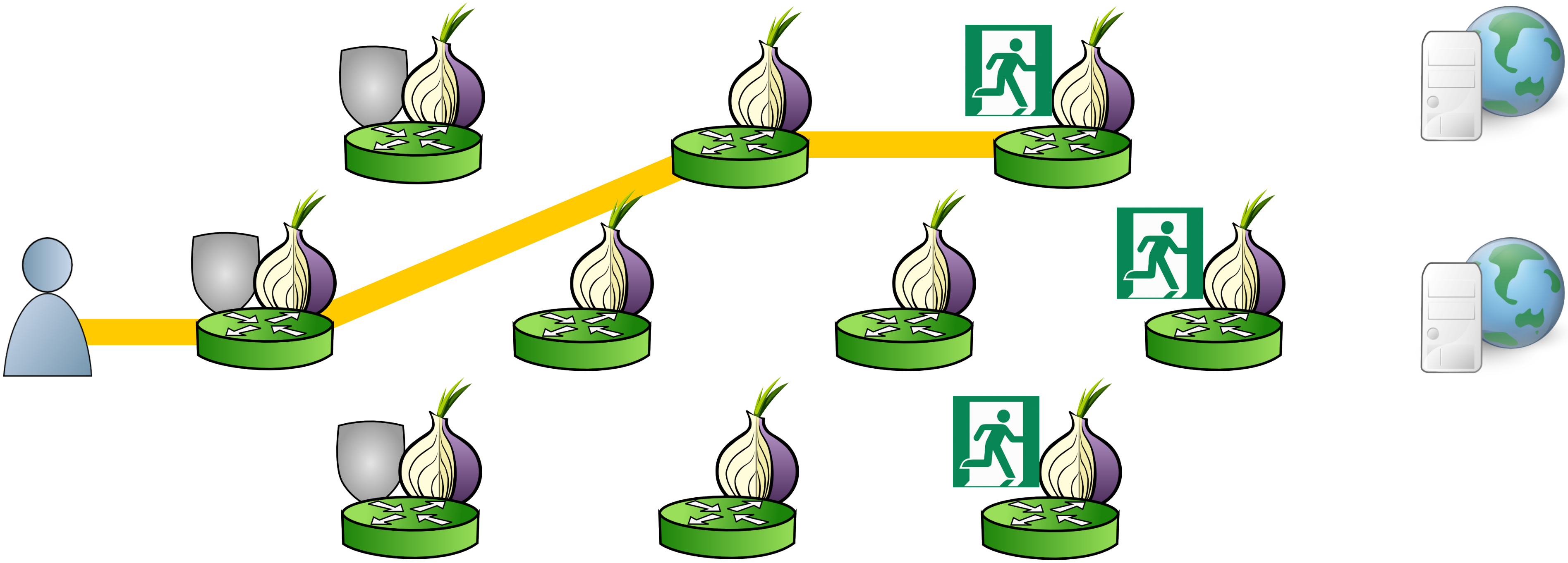
1. Clients begin all circuits with a selected guard
2. Relays define individual exit policies
3. Clients multiplex streams over a circuit

Background: Using Circuits



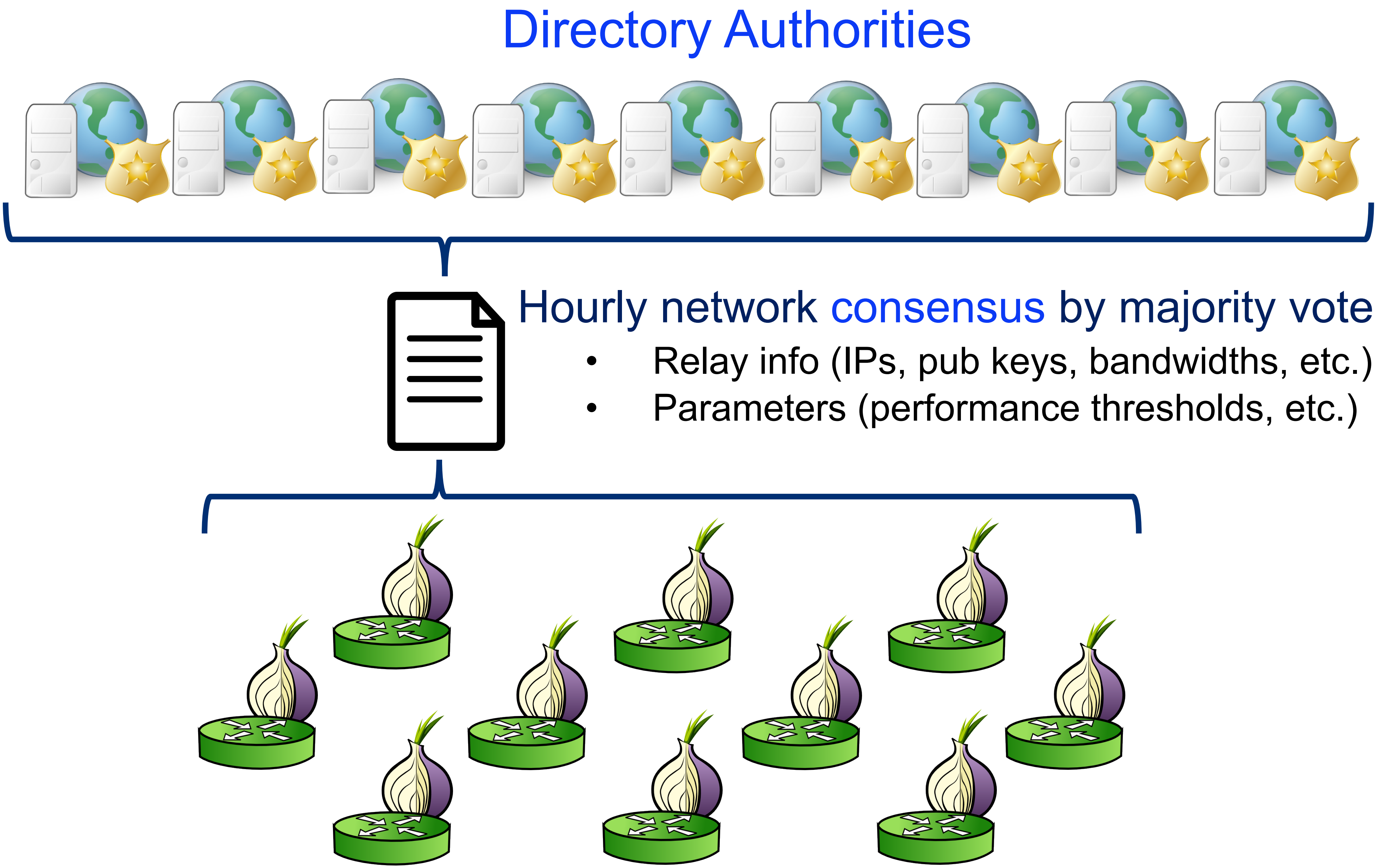
1. Clients begin all circuits with a selected guard
2. Relays define individual exit policies
3. Clients multiplex streams over a circuit
4. New circuits replace existing ones periodically

Background: Using Circuits



1. Clients begin all circuits with a selected guard
2. Relays define individual exit policies
3. Clients multiplex streams over a circuit
4. New circuits replace existing ones periodically
5. Clients randomly choose relays, weighted by bandwidth

Background: Directory Authorities



Directory Authorities

Hourly network consensus by majority vote

- Relay info (IPs, pub keys, bandwidths, etc.)
- Parameters (performance thresholds, etc.)

Motivation: Why Measure Tor?

Why are Tor network measurements needed?

- To understand usage behaviors to focus effort and resources
- To understand network protocols and calibrate parameters
- To inform policy discussion

U.S. NAVAL RESEARCH LABORATORY

Motivation: Why Measure Tor?

Why are Tor network measurements needed?

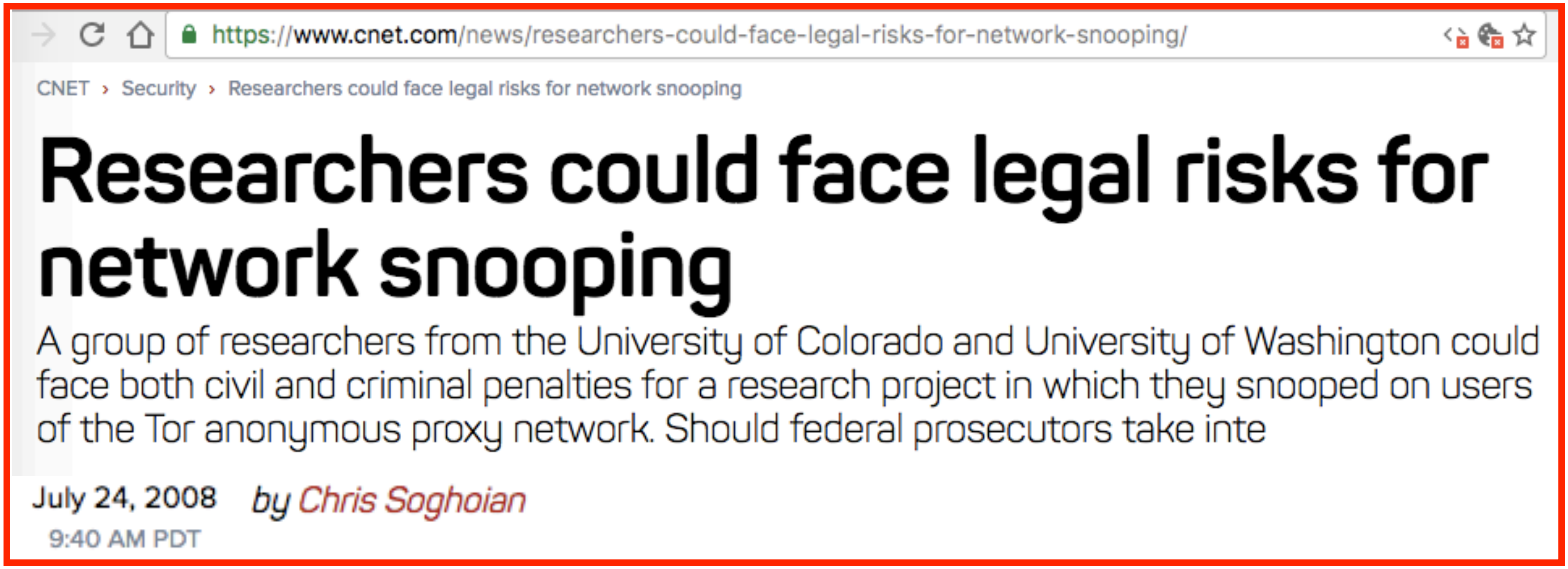
- To understand usage behaviors to focus effort and resources
- To understand network protocols and calibrate parameters
- To inform policy discussion

> *“Tor metrics are the ammunition that lets Tor and other security advocates argue for a more private and secure Internet from a position of data, rather than just dogma or perspective.”*
>
> – Bruce Schneier (June 1, 2016)
> (metrics.torproject.org)

Motivation: Previous Measurement Studies

Previous work collected, stored, and manually analyzed sensitive data

- McCoy *et. al.* (PETS 2008): tcpdump of first 150 bytes of packet (including 96 payload)
- Chaabane *et. al.* (NSS 2010): customized DPI software



https://www.cnet.com/news/researchers-could-face-legal-risks-for-network-snooping/

CNET › Security › Researchers could face legal risks for network snooping

Researchers could face legal risks for network snooping

A group of researchers from the University of Colorado and University of Washington could face both civil and criminal penalties for a research project in which they snooped on users of the Tor anonymous proxy network. Should federal prosecutors take inte

July 24, 2008 by Chris Soghoian
9:40 AM PDT

U.S. NAVAL RESEARCH LABORATORY

Motivation: Measurement Challenges

Tor METRICS

https://metrics.torproject.org

Some Existing Measurements

Data Published	Privacy Techniques	Unsafe	Inaccurate
Relay BW available	Test measurements		✖
Relay BW used	Aggregated ~ 4 hours	✖	
Total # daily users	Inferred (consensus fetches)		✖
# users per country	Aggregated ~ 24 hours, rounded, opt-in	✖	
Exit traffic per port	Aggregated ~ 24 hours, opt-in	✖	

Motivation: Measurement Challenges

https://metrics.torproject.org

Safety concerns:

- Per-relay outputs
- Data stored locally
- No privacy proofs

Some Existing Measurements

Data Published	Privacy Techniques	Unsafe	Inaccurate
Relay BW available	Test measurements		✖
Relay BW used	Aggregated ~ 4 hours	✖	
Total # daily users	Inferred (consensus fetches)		✖
# users per country	Aggregated ~ 24 hours, rounded, opt-in	✖	
Exit traffic per port	Aggregated ~ 24 hours, opt-in	✖	

Motivation: Measurement Challenges

Tor METRICS

https://metrics.torproject.org

Accuracy concerns:

- Per-relay noise
- Opt-in, limited vantage points

Some Existing Measurements

Data Published	Privacy Techniques	Unsafe	Inaccurate
Relay BW available	Test measurements		✖
Relay BW used	Aggregated ~ 4 hours	✖	
Total # daily users	Inferred (consensus fetches)		✖
# users per country	Aggregated ~ 24 hours, rounded, opt-in	✖	
Exit traffic per port	Aggregated ~ 24 hours, opt-in	✖	

Motivation: Missing Measurements

Many useful statistics are not collected for safety

Users

- Total number of unique users at any time, how long they stay online, how often they join and leave, usage behavior

Relays

- Total bandwidth capacity, congestion and queuing delays, circuit and other failures, denial of service and other attacks

Destinations

- Popular destinations, popular applications, effects of DNS, properties of traffic (bytes and connections per page, etc.)

The PrivCount Measurement System

- PrivCount system architecture
- Distributed measurement and aggregation protocol
- Secure computation and private output

PrivCount: Overview

Privacy-preserving counting system

- Consumes various new event types from Tor
  - Circuit end events
  - Stream end events
  - Connection end events

- Counts various statistics from event information, e.g.:
  - Total number of circuits, streams, connections
  - Data volume per circuit, stream
  - Number of unique users
  - …

- Based on PrivEx-S2 protocol of Elahi *et. al.* (CCS 2014)

PrivCount: Overview

Security goals for safer Tor measurements

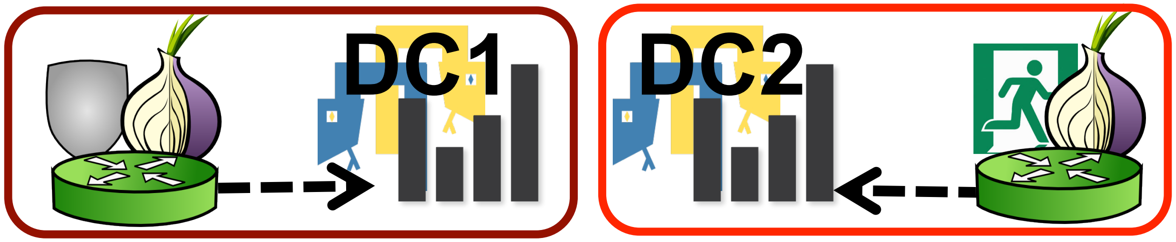
- Forward privacy
  - The adversary cannot learn the state of the measurement before time of compromise
- Differential privacy
  - Prevents confirmation of the actions of a specific user given the output
- Secure aggregation
  - Securely aggregates safe statistics across all measurement nodes
  - Only the safe, aggregated measurement results are released

U.S. NAVAL RESEARCH LABORATORY

PrivCount: Architecture

Data Collectors (DCs)

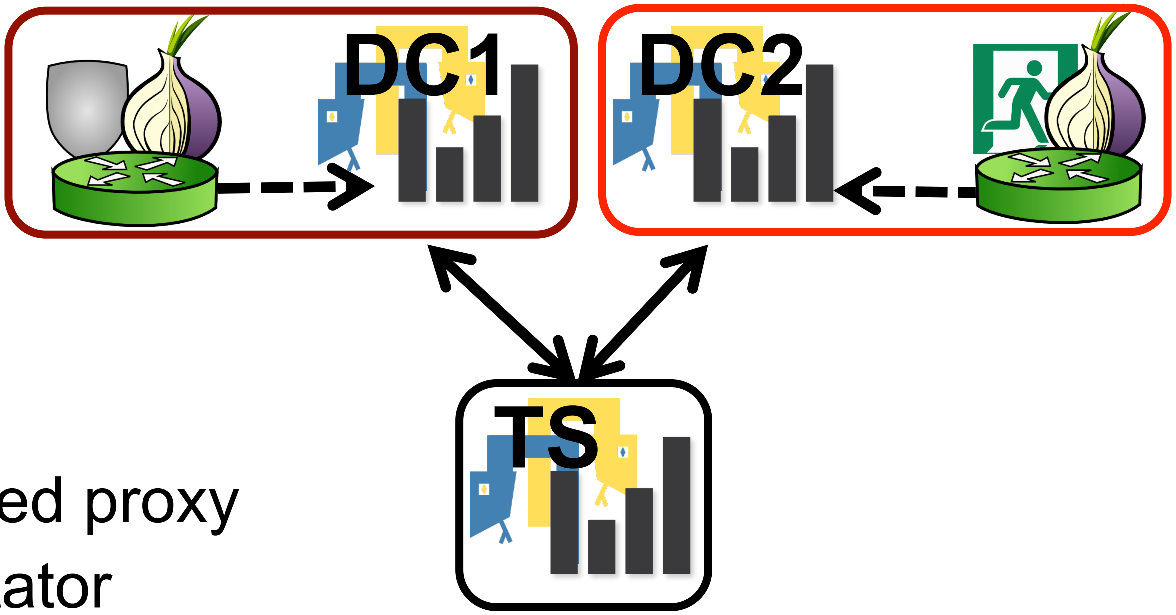
- Collect events
- Increment counters



PrivCount: Architecture

Data Collectors (DCs)

- Collect events
- Increment counters



Tally Server (TS)

- Central, untrusted proxy
- Collection facilitator

PrivCount: Architecture

Data Collectors (DCs)

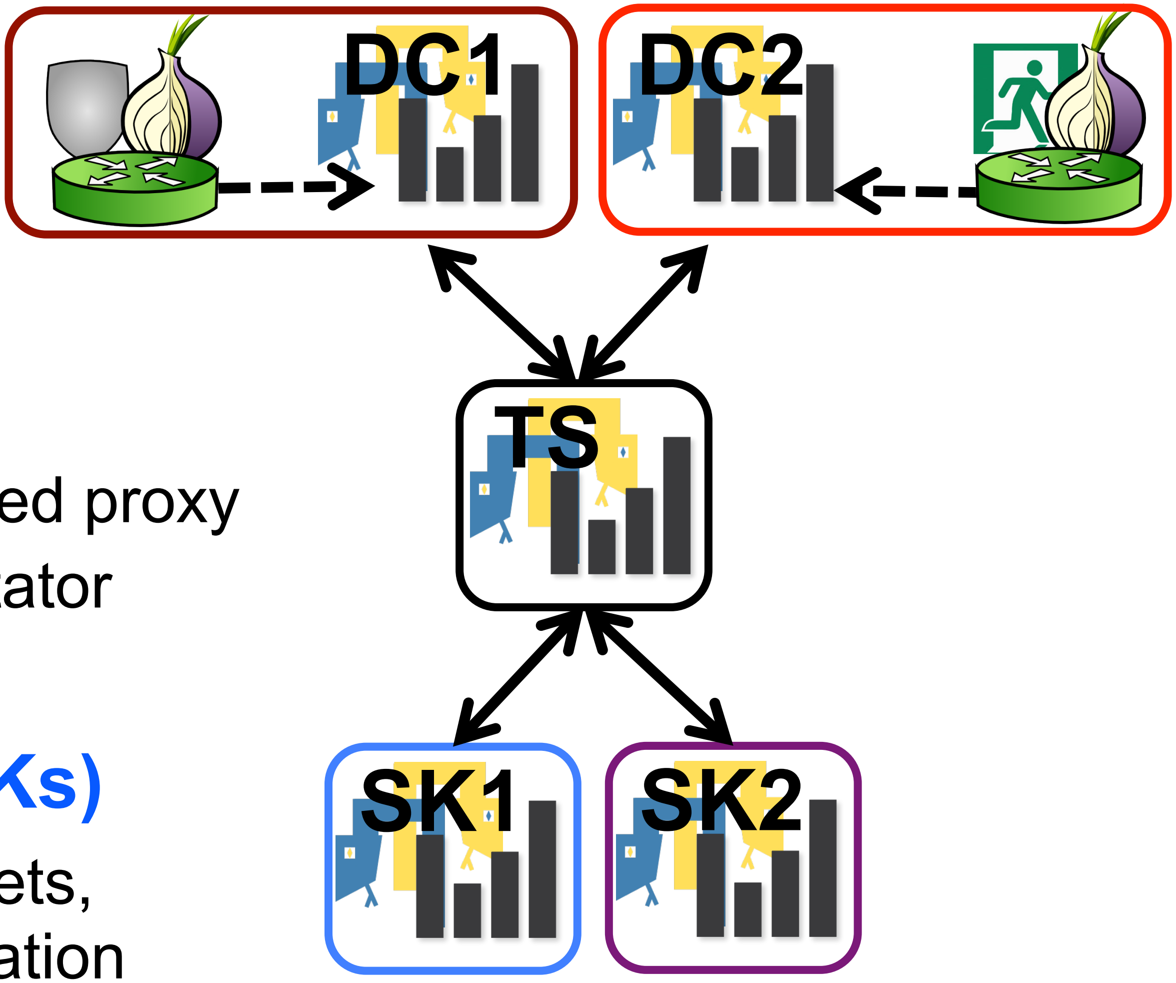
- Collect events
- Increment counters

Tally Server (TS)

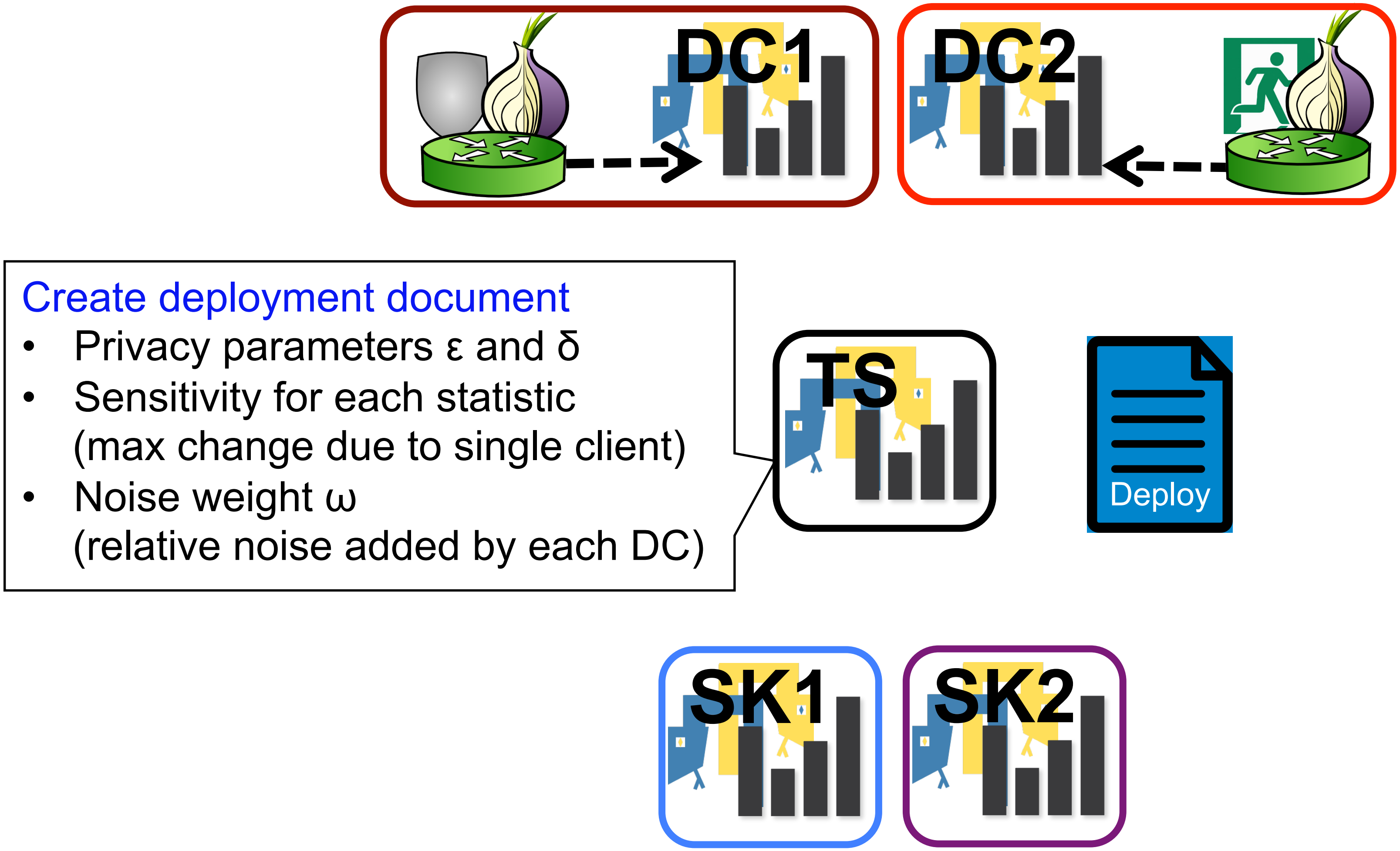
- Central, untrusted proxy
- Collection facilitator

Share Keepers (SKs)

- Stores DC secrets, sum for aggregation



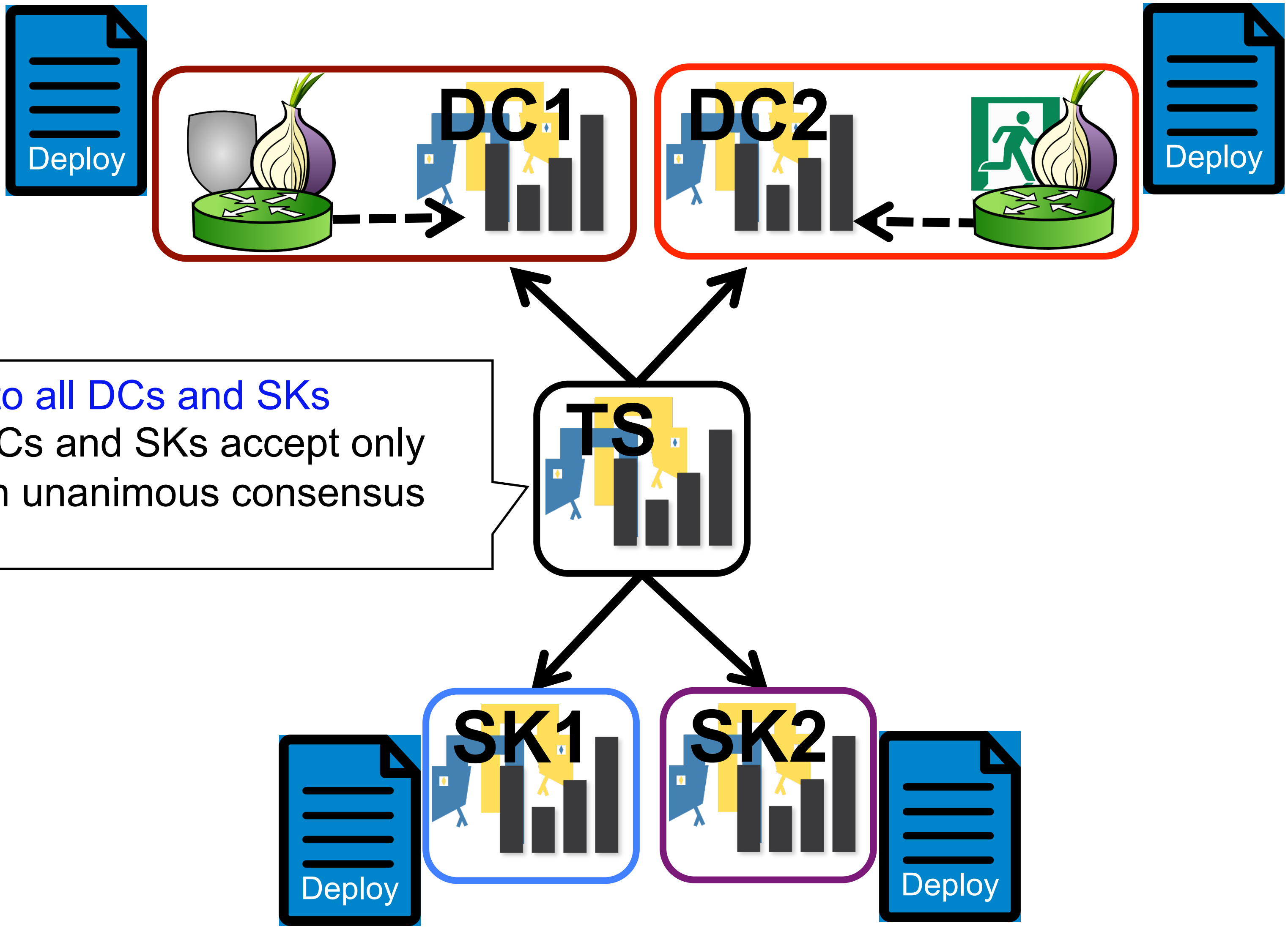
PrivCount: Initialization



Create deployment document

- Privacy parameters ε and δ
- Sensitivity for each statistic (max change due to single client)
- Noise weight ω (relative noise added by each DC)

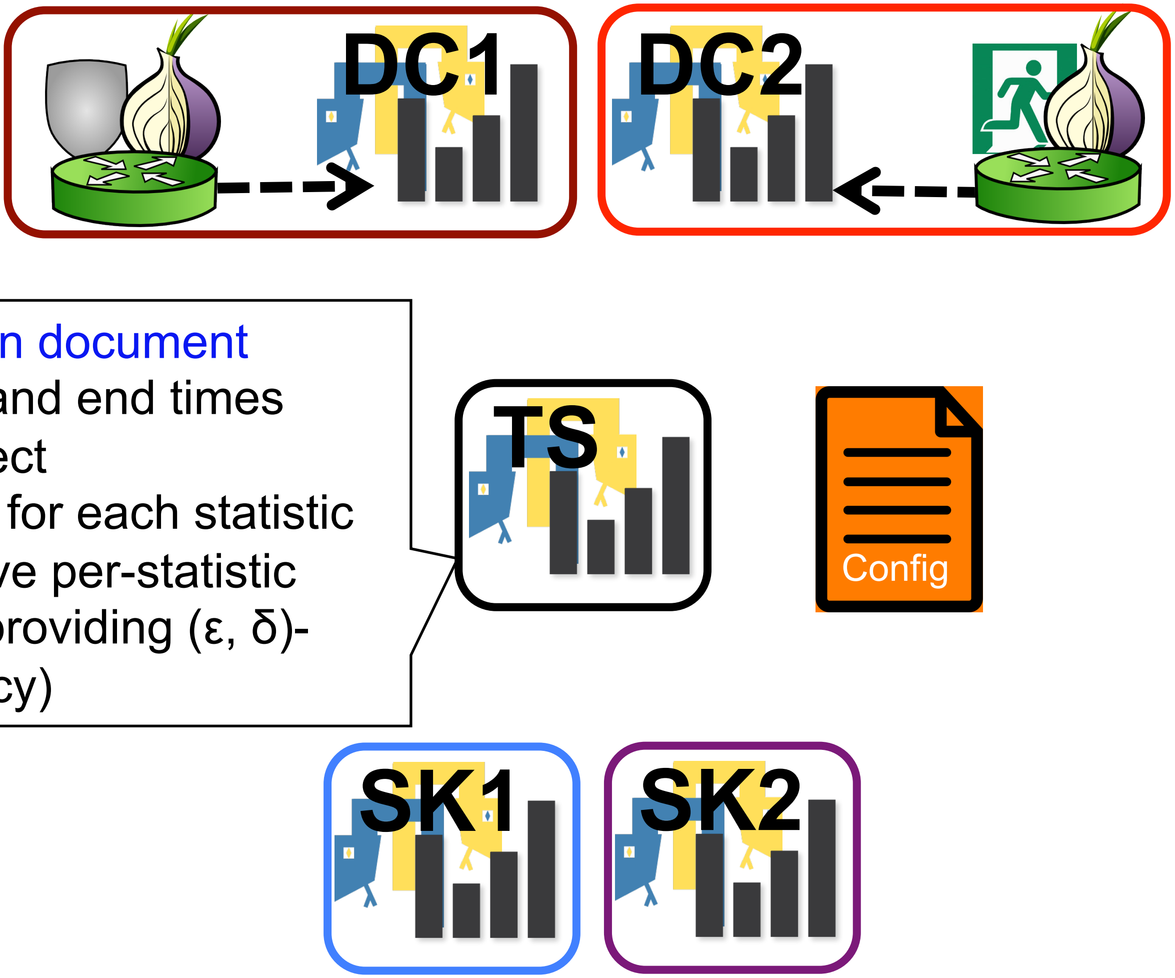
PrivCount: Initialization



Send to all DCs and SKs

- DCs and SKs accept only on unanimous consensus

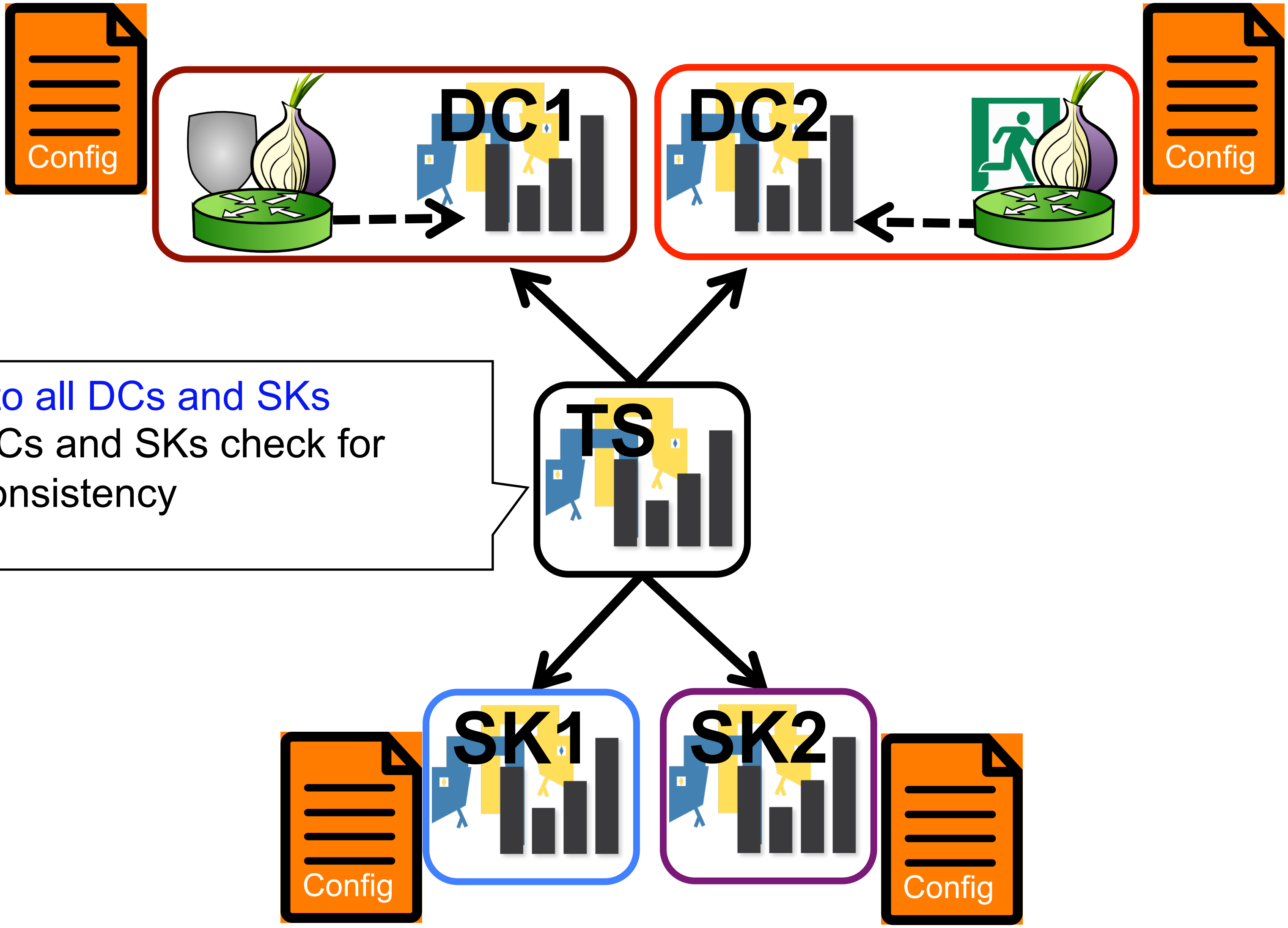
PrivCount: Configuration



Create configuration document

- Collection start and end times
- Statistics to collect
- Estimated value for each statistic (maximize relative per-statistic accuracy while providing (ε, δ)-differential privacy)

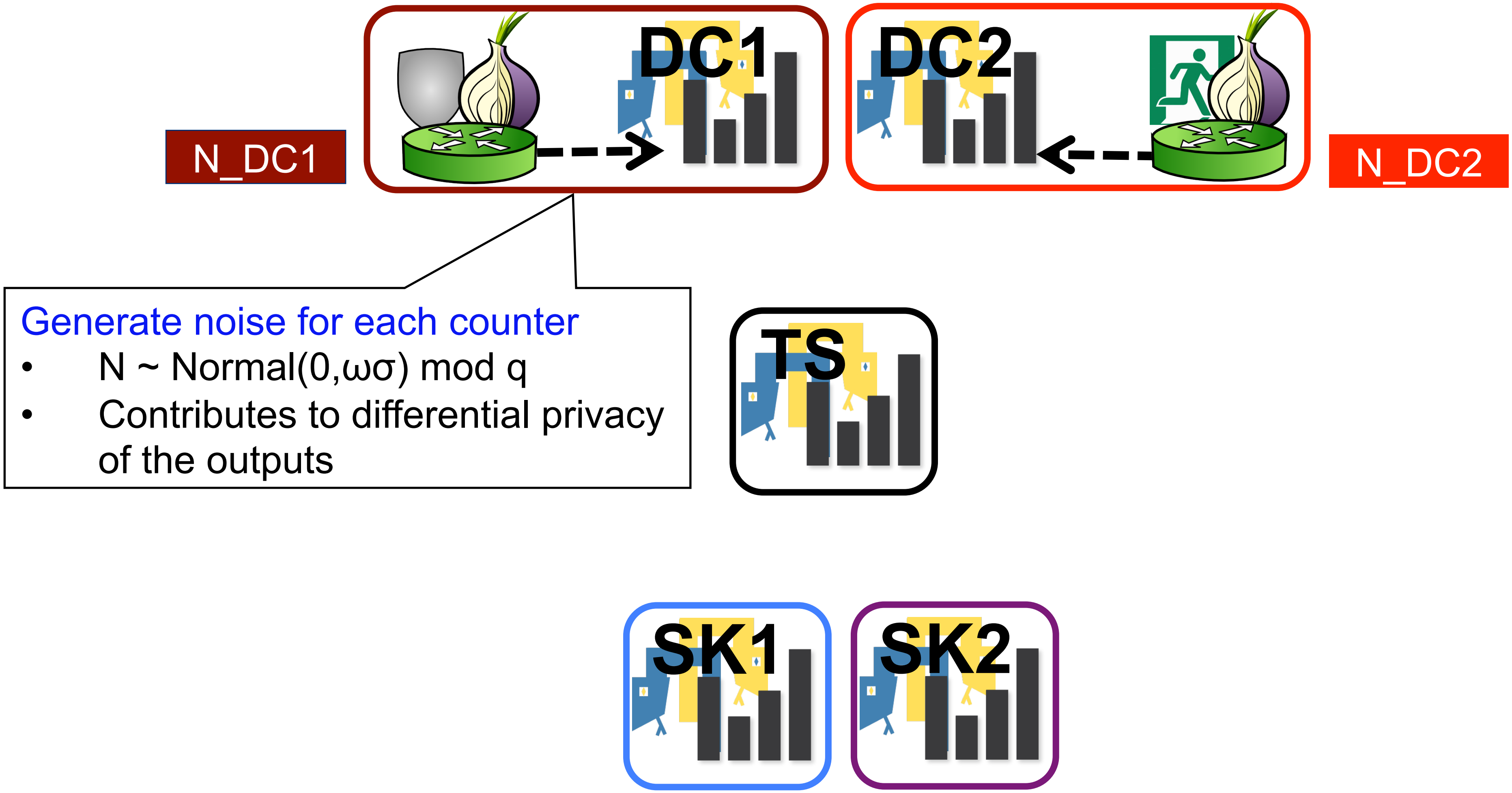
PrivCount: Configuration



Send to all DCs and SKs

- DCs and SKs check for consistency

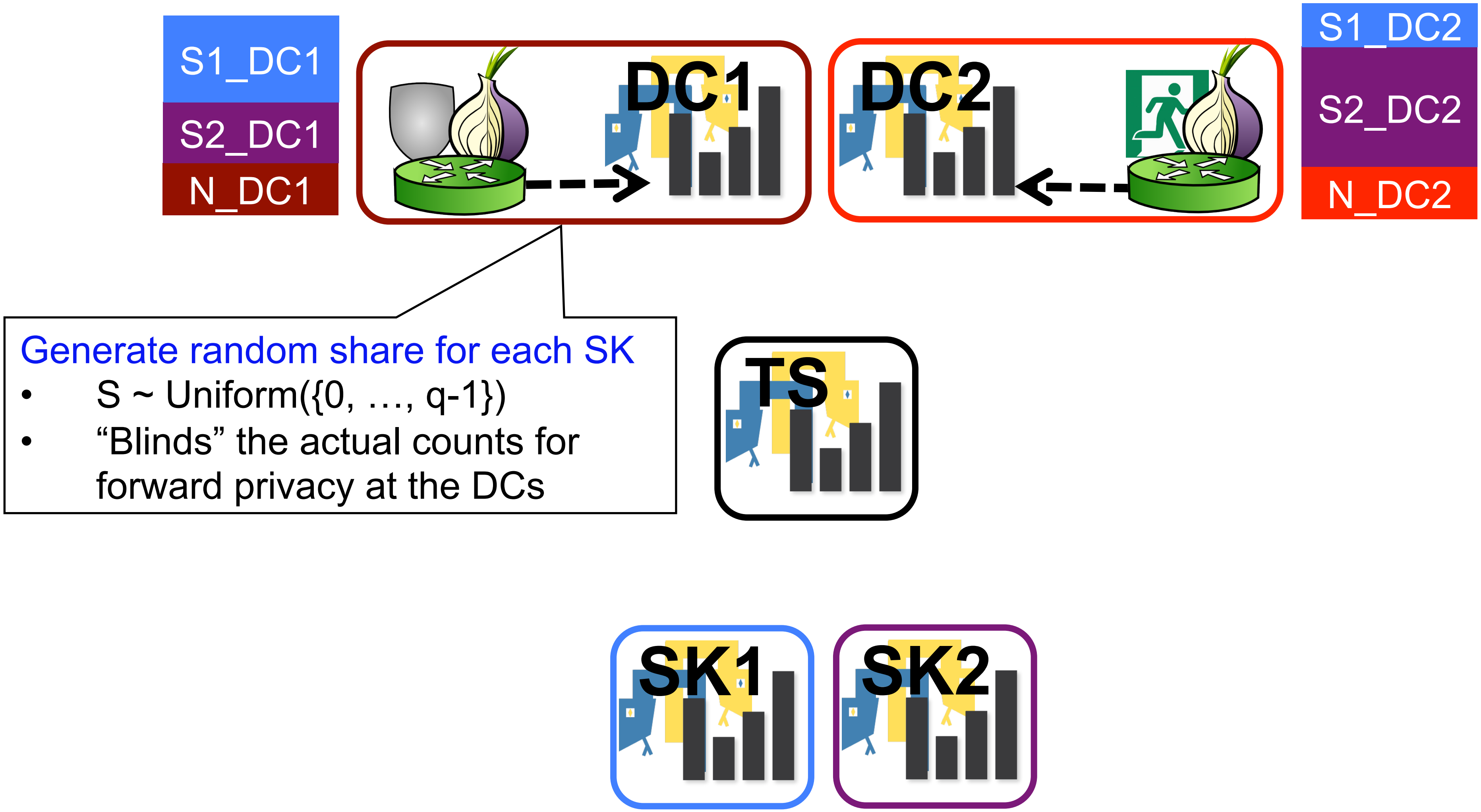
PrivCount: Execution - Setup



Generate noise for each counter

- N ~ Normal(0,ωσ) mod q
- Contributes to differential privacy of the outputs

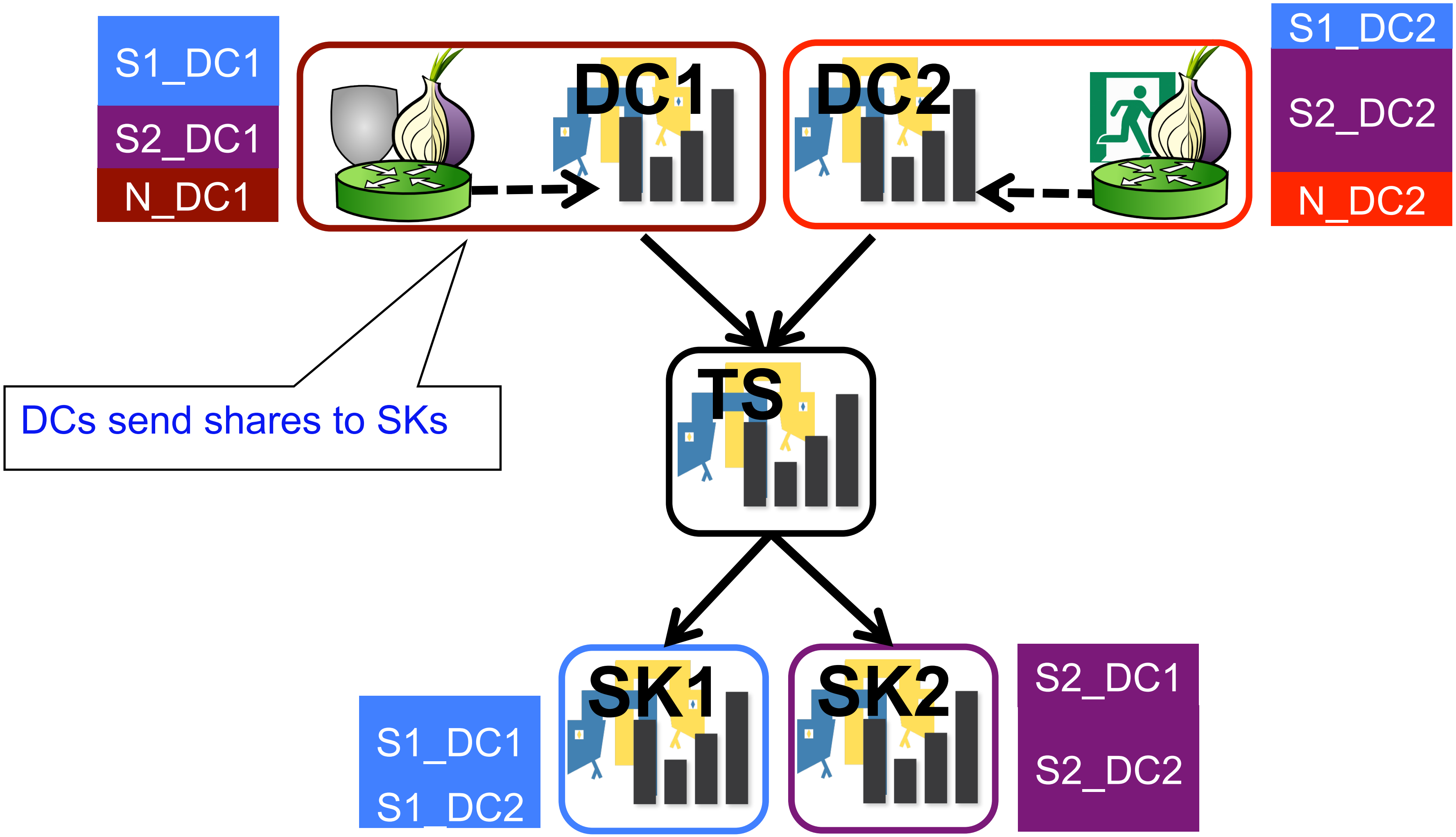
PrivCount: Execution - Setup



Generate random share for each SK

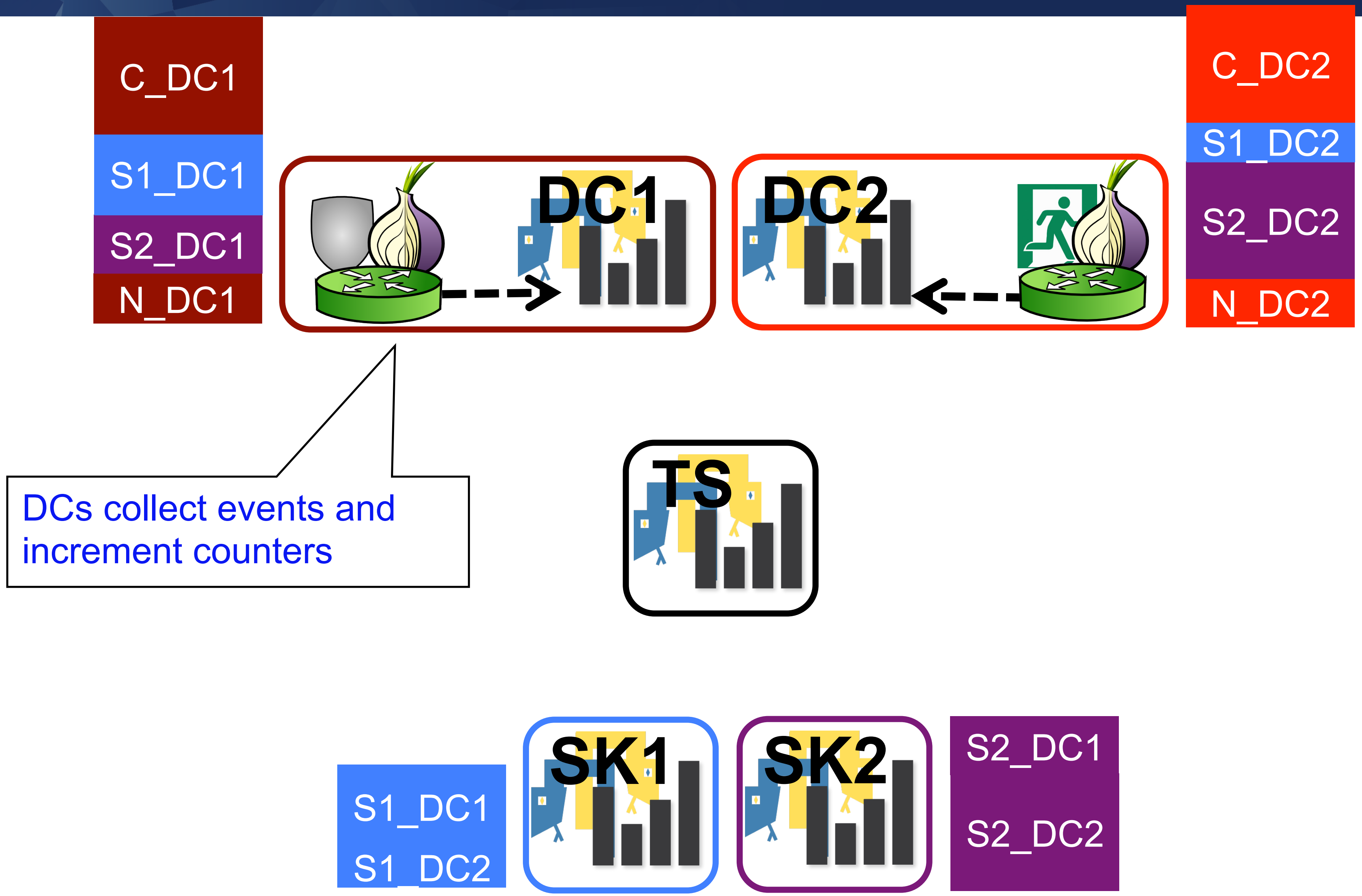
- S ~ Uniform({0, …, q-1})
- “Blinds” the actual counts for forward privacy at the DCs

PrivCount: Execution - Setup



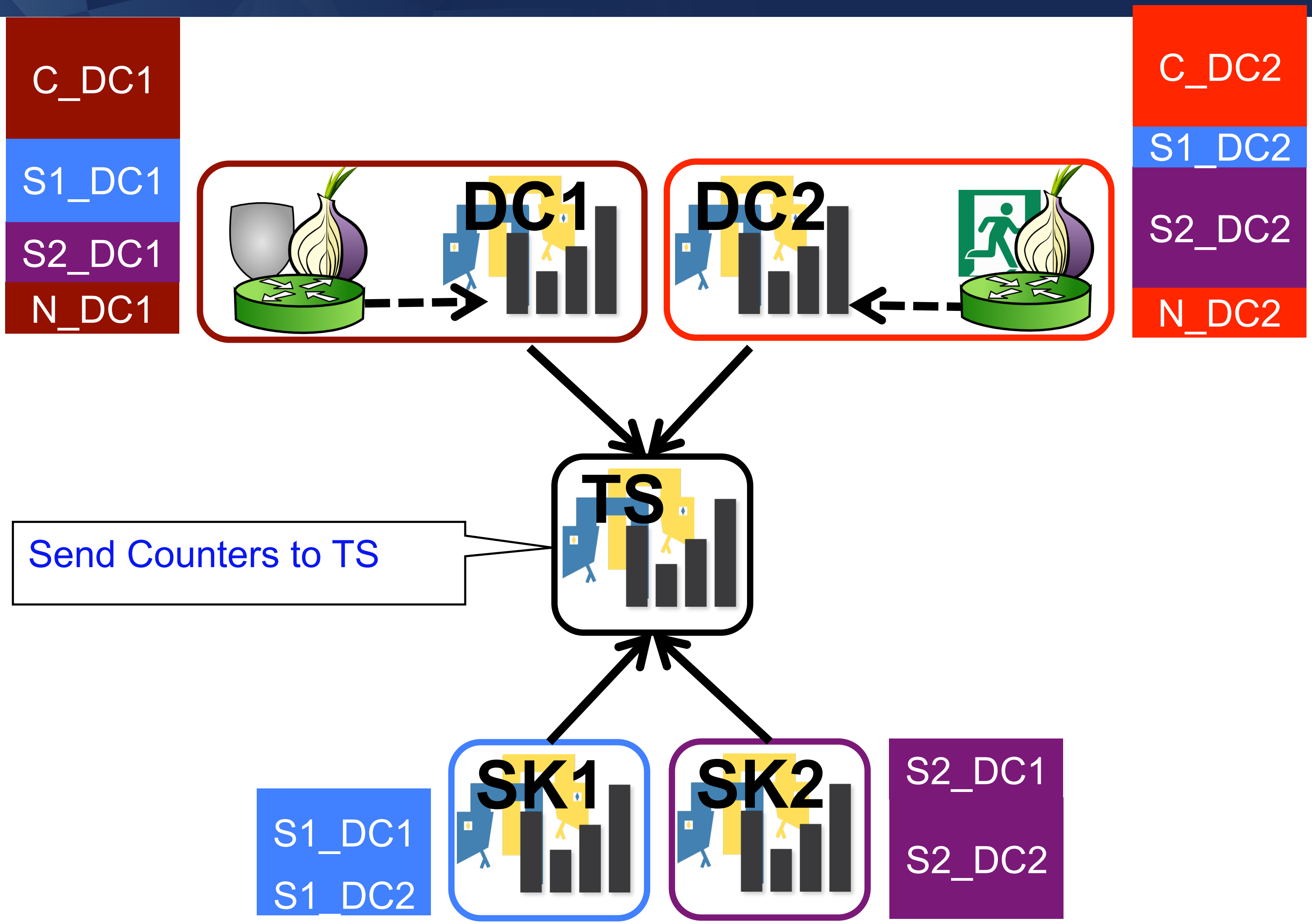
DCs send shares to SKs

PrivCount: Collection



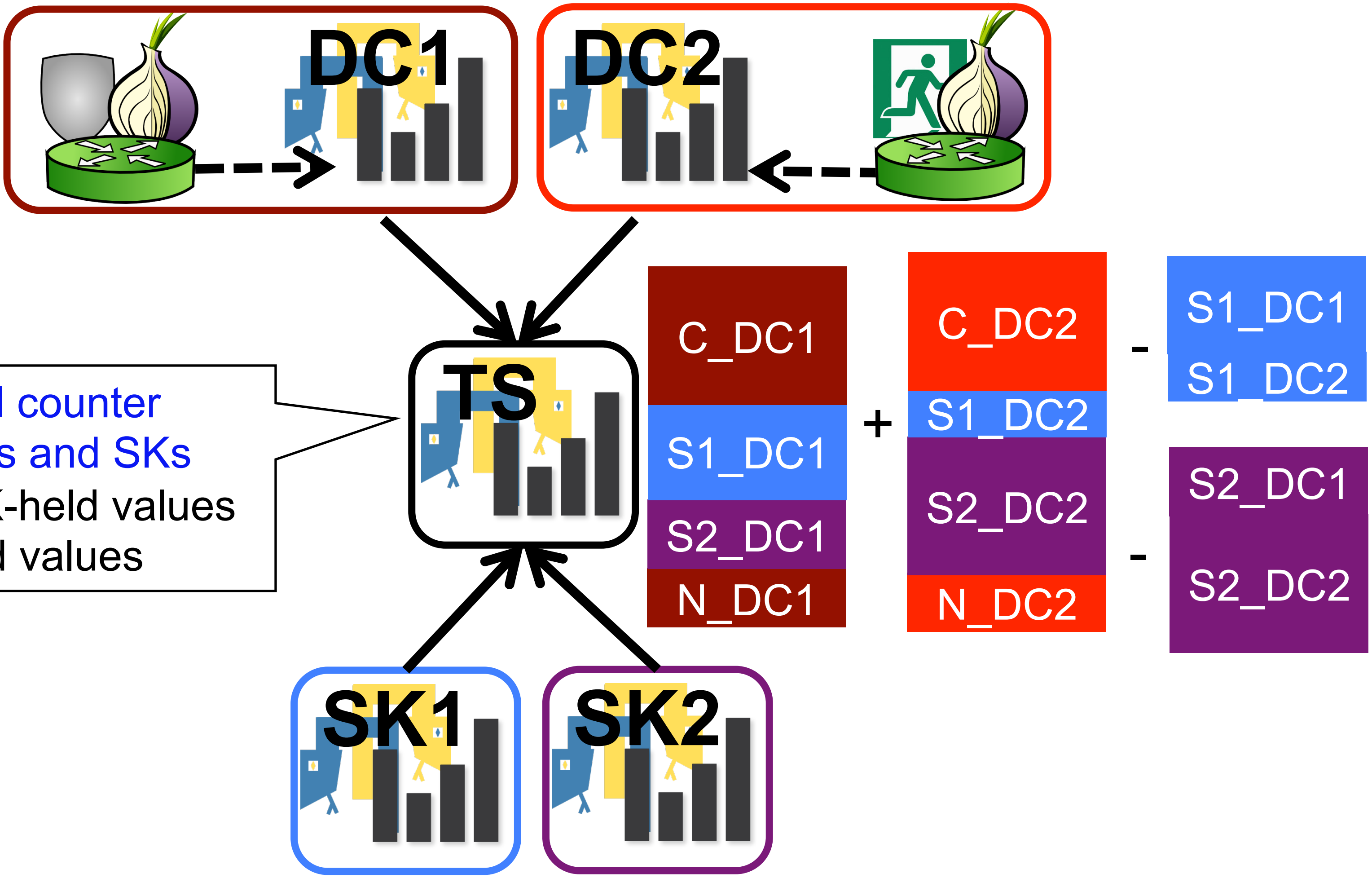
DCs collect events and increment counters

PrivCount: Collection



Send Counters to TS

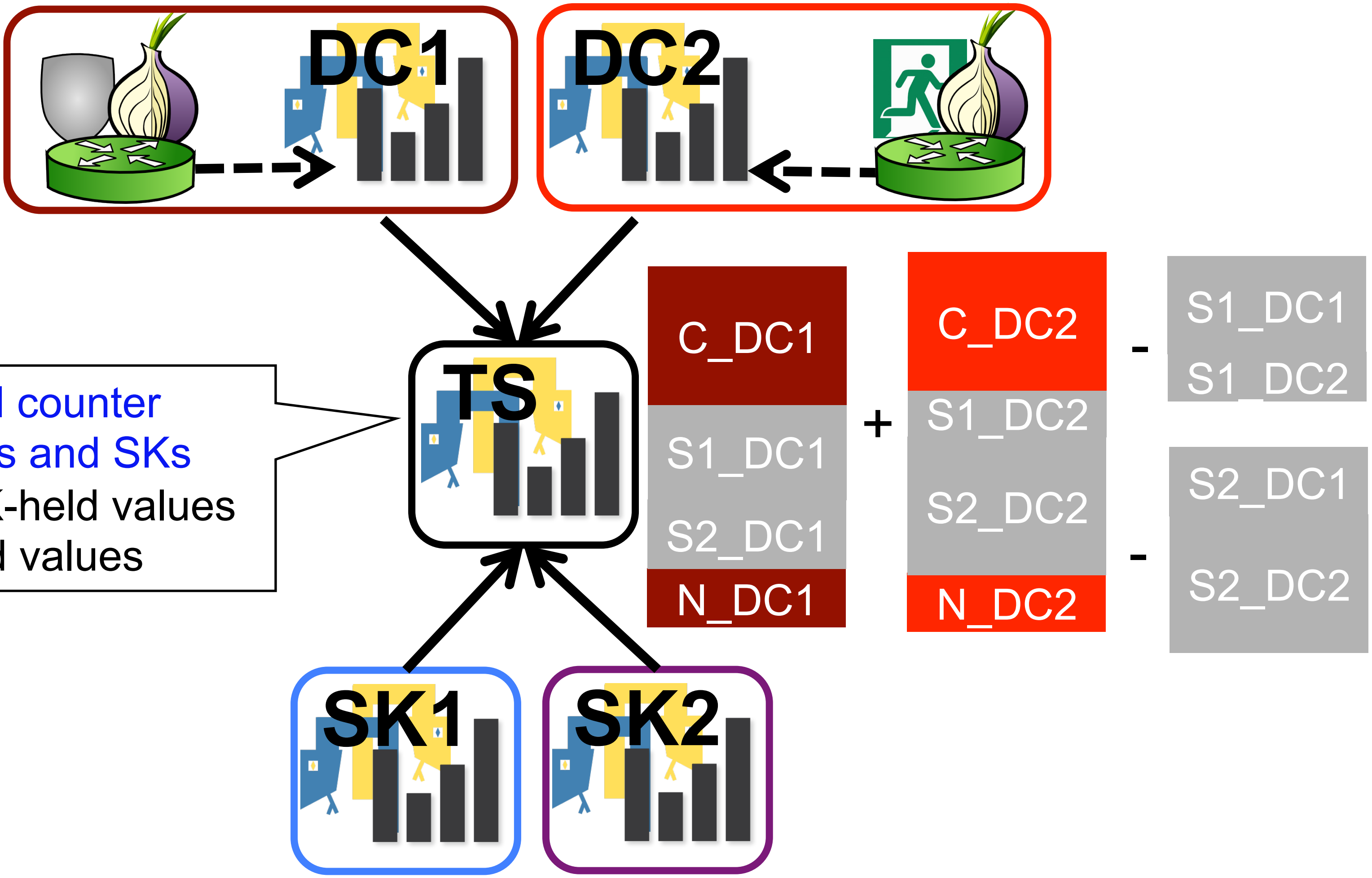
PrivCount: Aggregation



TS combines all counter values from DCs and SKs

- Subtracts SK-held values from DC-held values

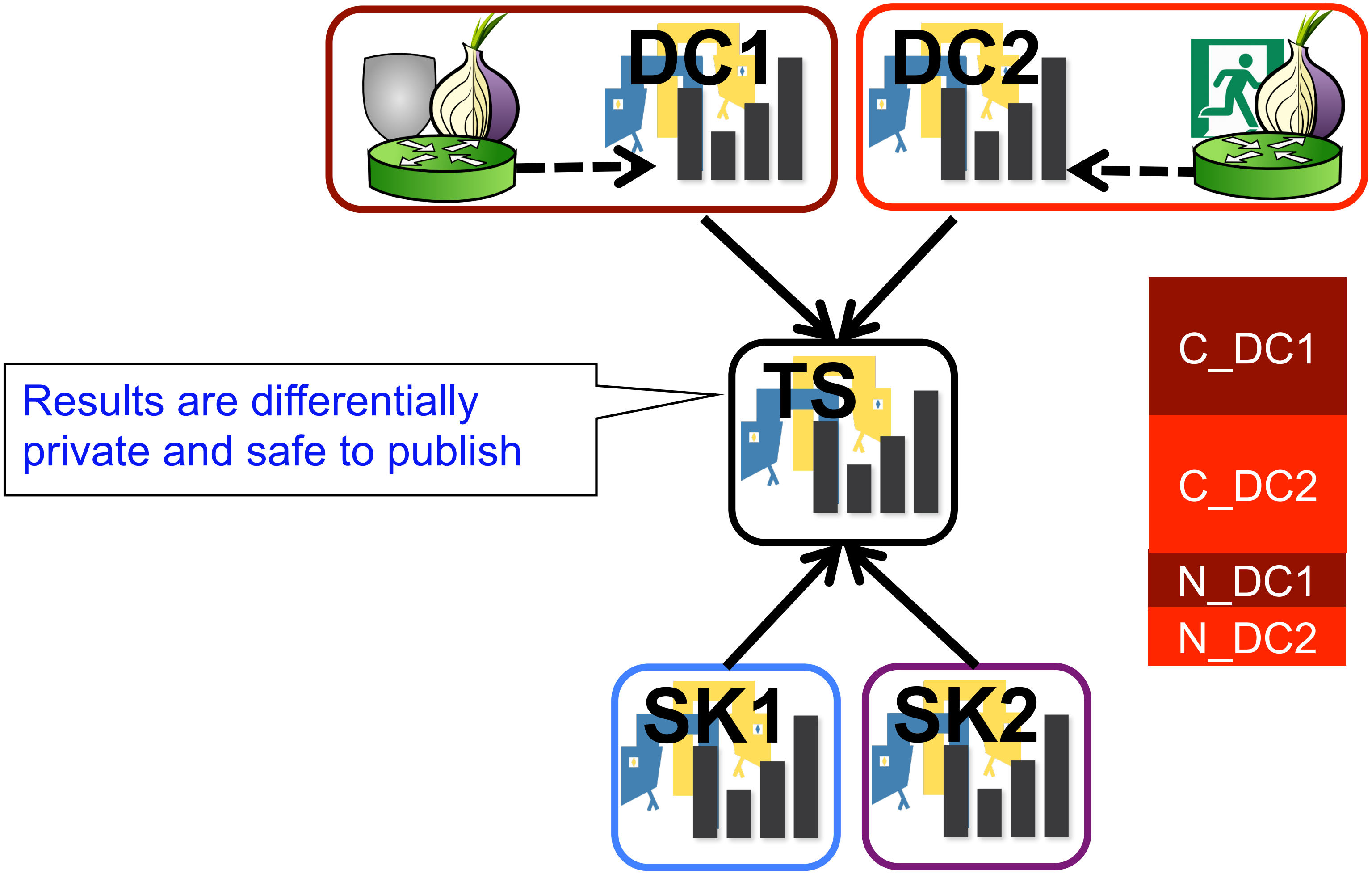
PrivCount: Aggregation



TS combines all counter values from DCs and SKs

- Subtracts SK-held values from DC-held values

PrivCount: Aggregation



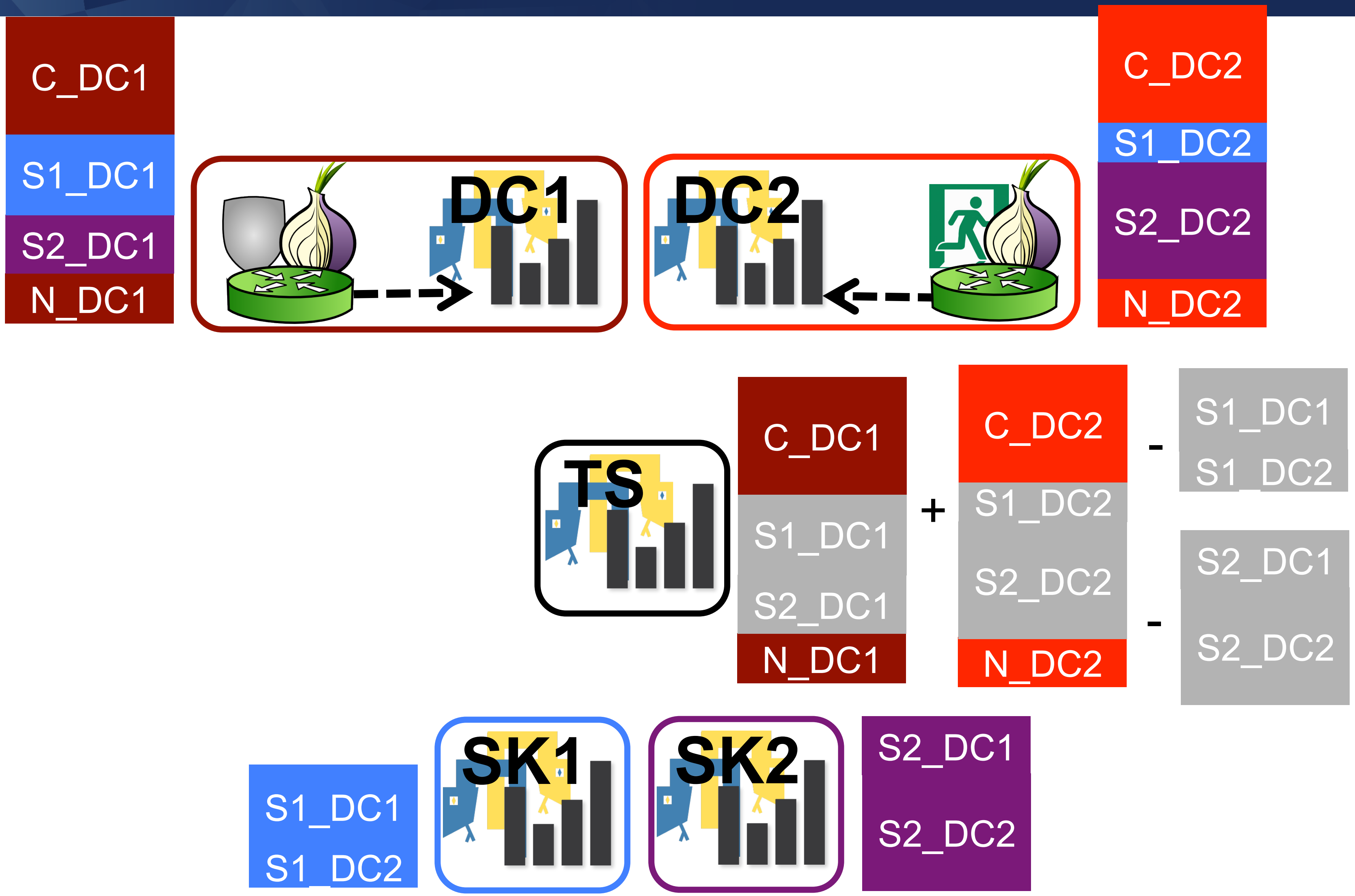
Results are differentially private and safe to publish

PrivCount: Security

Recall: Security Properties

- Forward privacy
    - The adversary cannot learn the state of the measurement before time of compromise
- Differential privacy
    - Prevents confirmation of the actions of a specific user given the output
- Secure aggregation
    - Securely aggregates safe statistics across all measurement nodes
    - Only the safe, aggregated measurement results are released

PrivCount: Security



C_DC1
S1_DC1
S2_DC1
N_DC1

DC1

DC2

C_DC2
S1_DC2
S2_DC2
N_DC2

TS

C_DC1
S1_DC1
S2_DC1
N_DC1

+

C_DC2
S1_DC2
S2_DC2
N_DC2

-

S1_DC1
S1_DC2

-

S2_DC1
S2_DC2

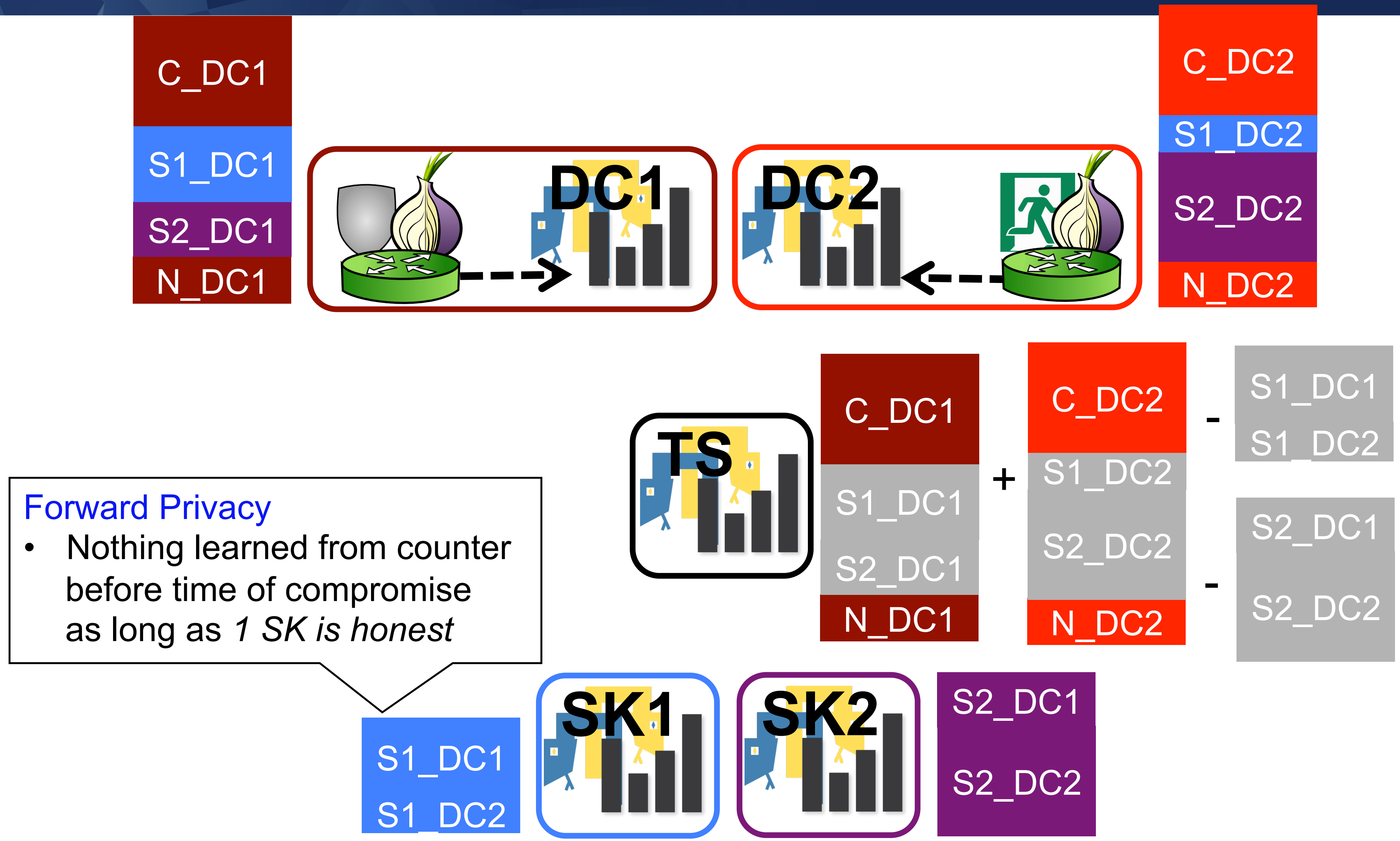
S1_DC1
S1_DC2

SK1

SK2

S2_DC1
S2_DC2

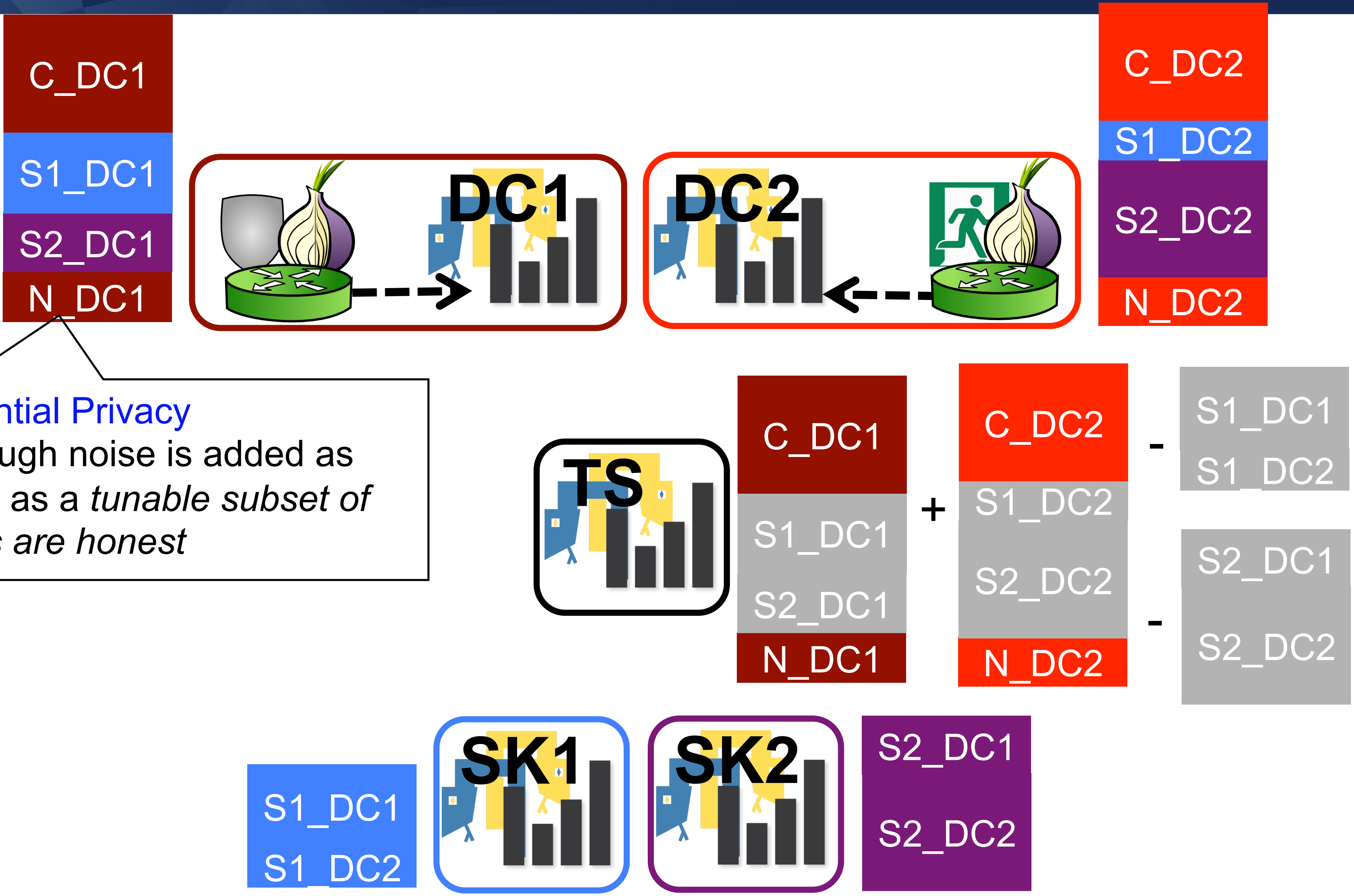
PrivCount: Security



Forward Privacy

- Nothing learned from counter before time of compromise as long as *1 SK is honest*

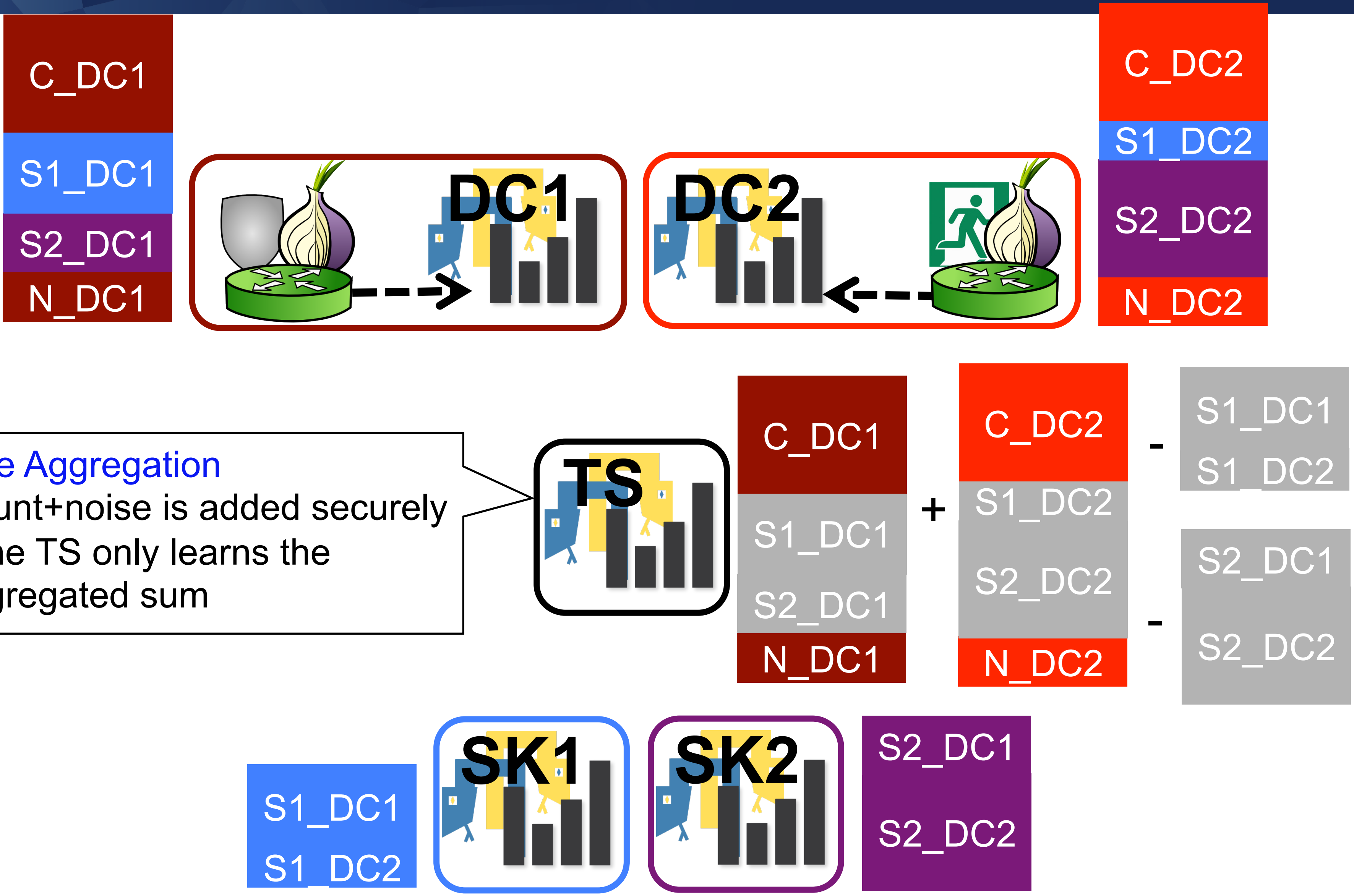
PrivCount: Security



Differential Privacy

- Enough noise is added as long as a *tunable subset of DCs are honest*

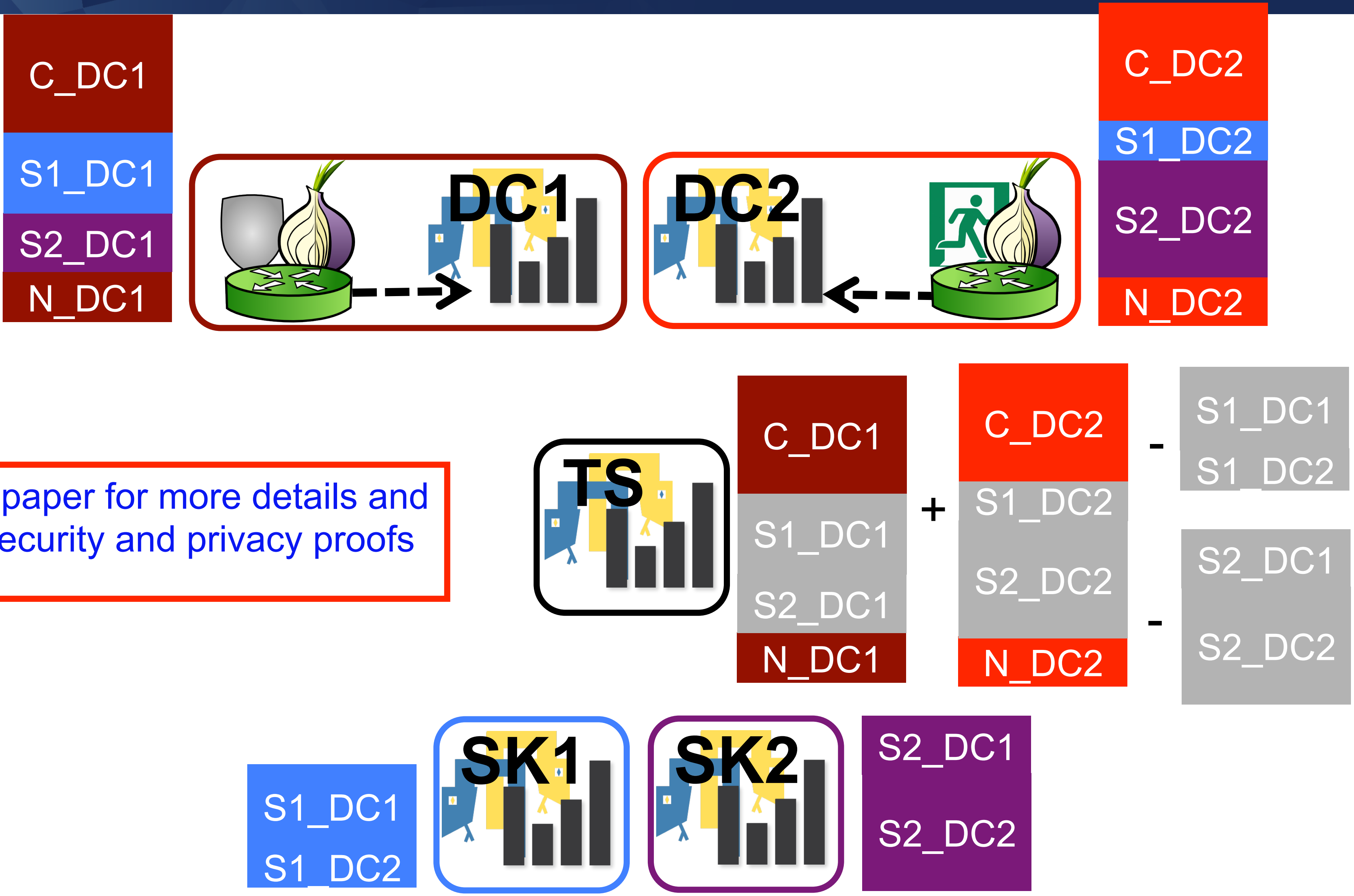
PrivCount: Security



Secure Aggregation

- Count+noise is added securely – the TS only learns the aggregated sum

PrivCount: Security

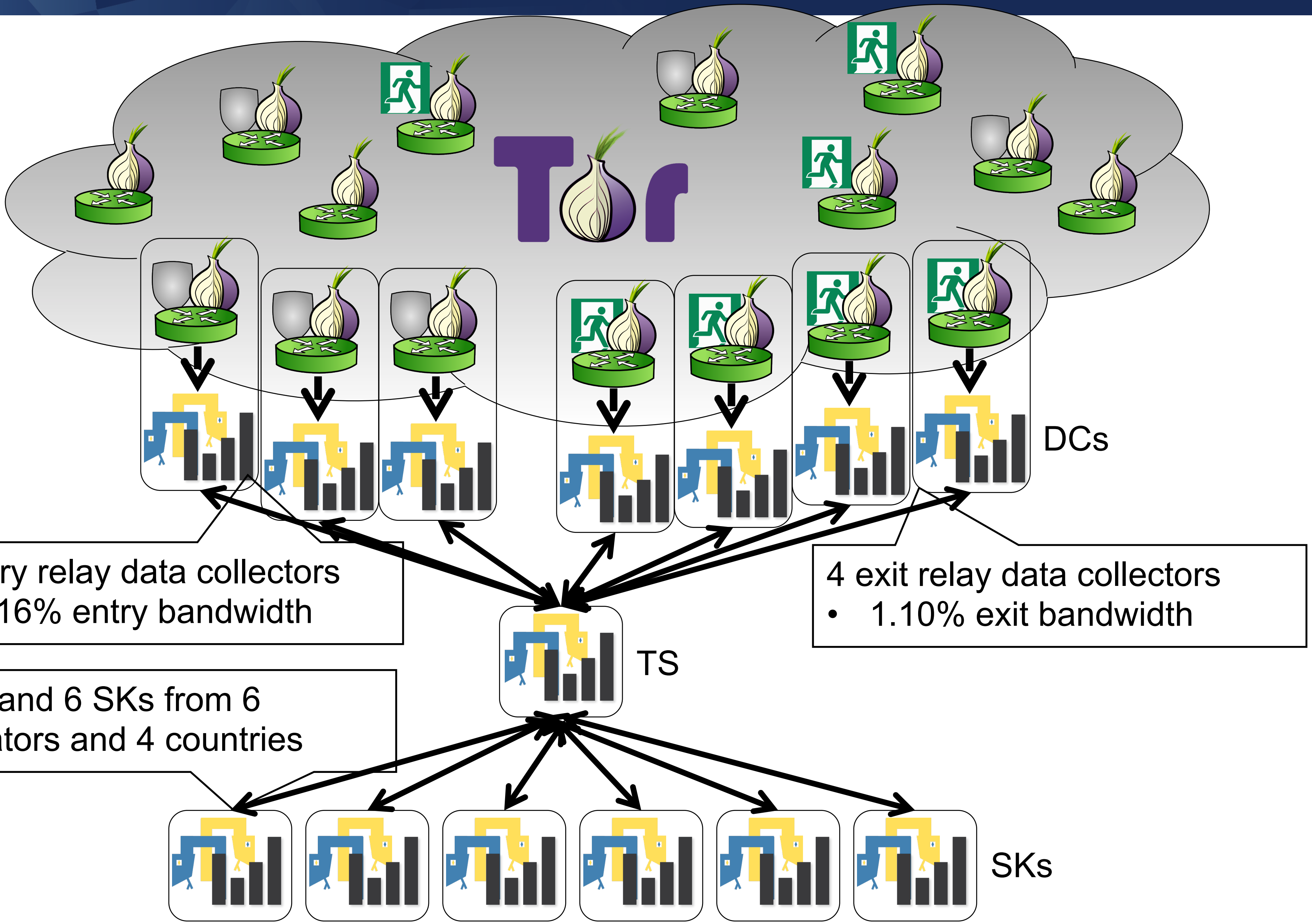


See paper for more details and for security and privacy proofs

Deployment and Measurement Results

- Configuring and running Tor relays
- “Exploratory” measurements using various exit policies
- “In-depth” measurements of most popular usage
- Network-wide measurement inference

Deploying PrivCount



3 entry relay data collectors

- 0.16% entry bandwidth

4 exit relay data collectors

- 1.10% exit bandwidth

1 TS and 6 SKs from 6 operators and 4 countries

Collection Phases

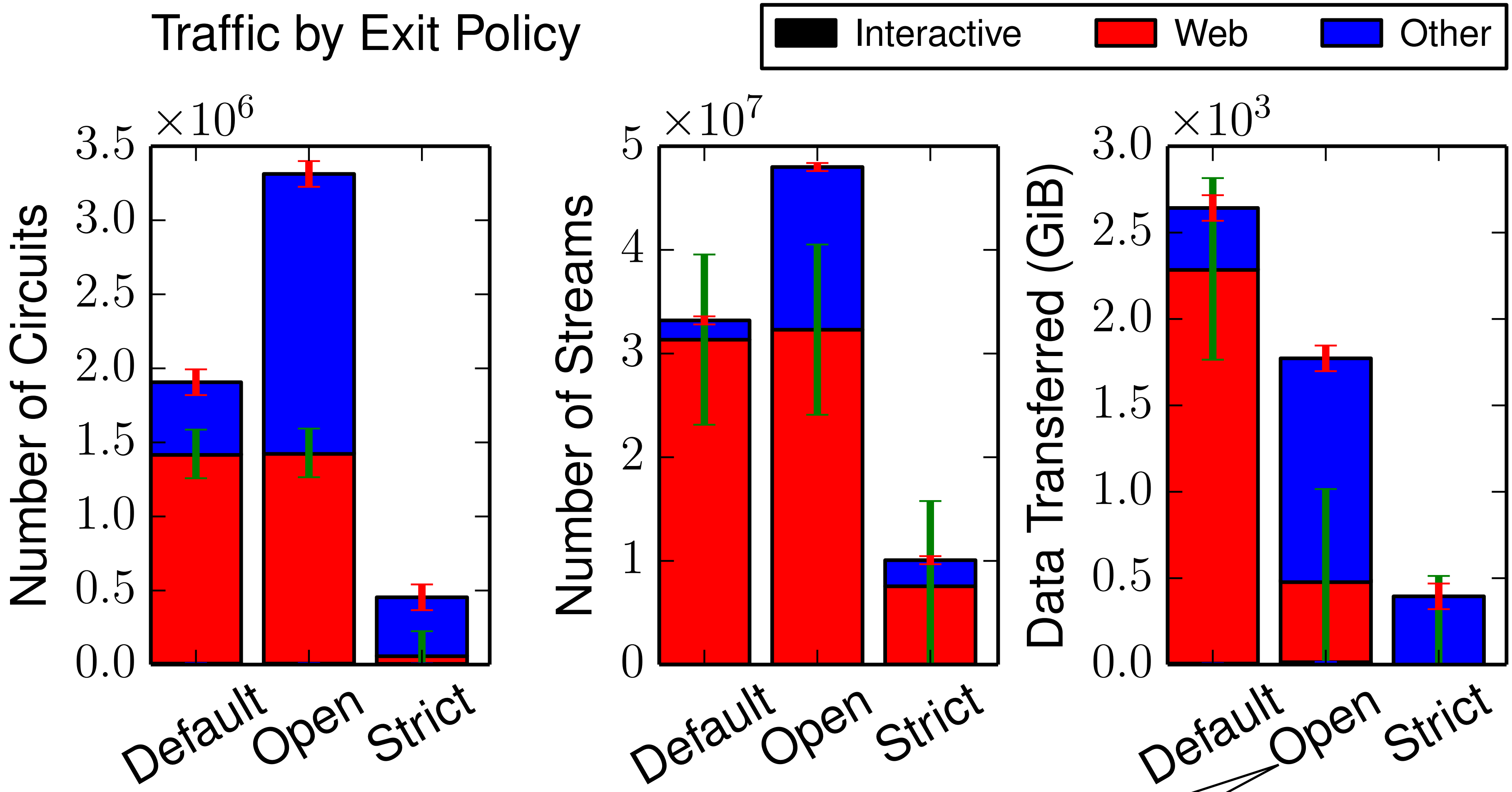
Exploratory phases

- Explore various exit policies (strict, default, open)
- Explore various applications (web, interactive, other)
- Gather only totals (circuits, streams, bytes)
- Use Tor metrics to estimate input parameters
- Run for 1 day, iterate

In-depth phases

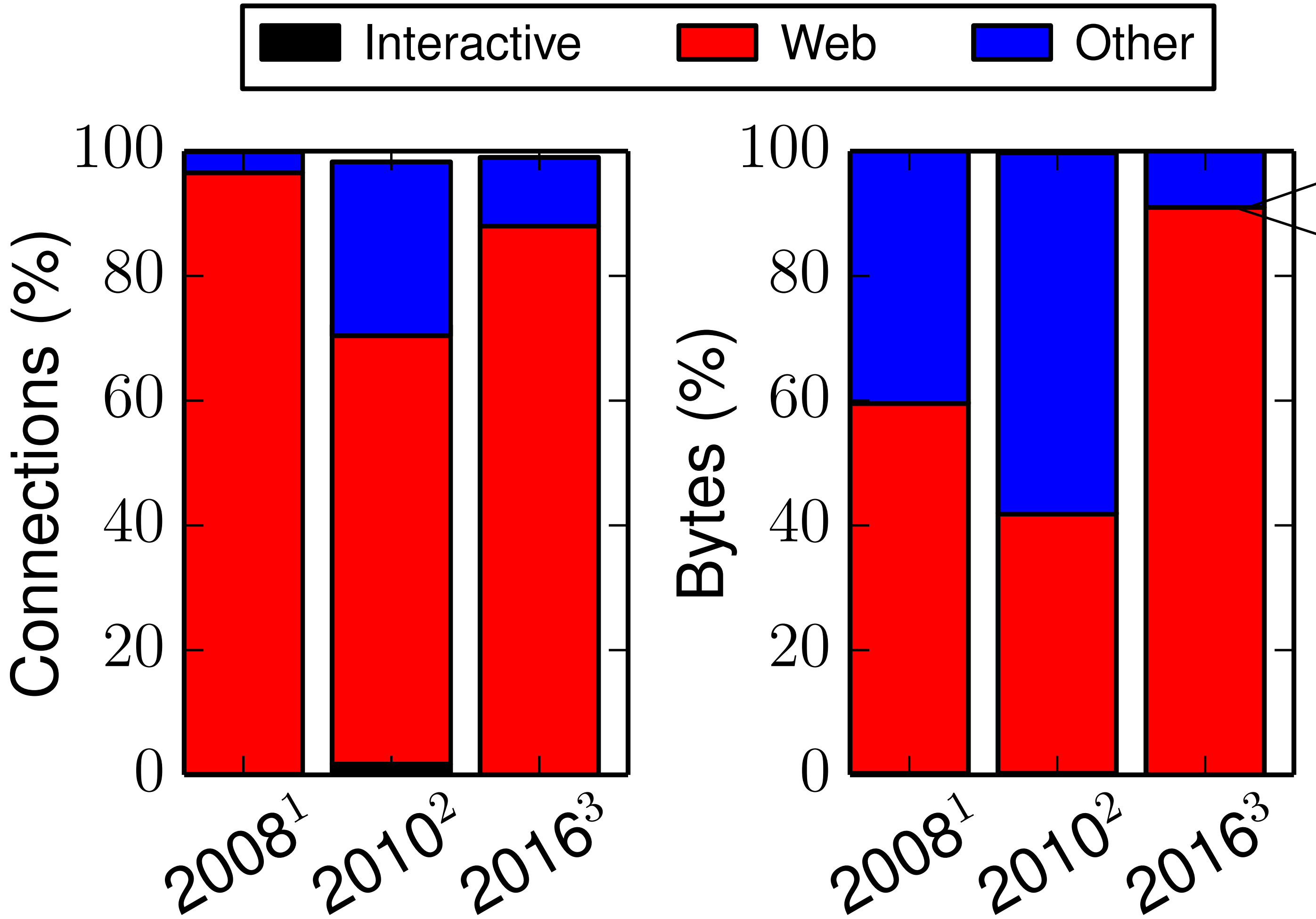
- Focus on most popular exit policy and applications
- Gather totals and histograms
- Use exploratory results to estimate input parameters
- Run for 4 days for client stats, 21 days for exit stats

Results: Exit Policies



Open file sharing ports reduce web data transferred

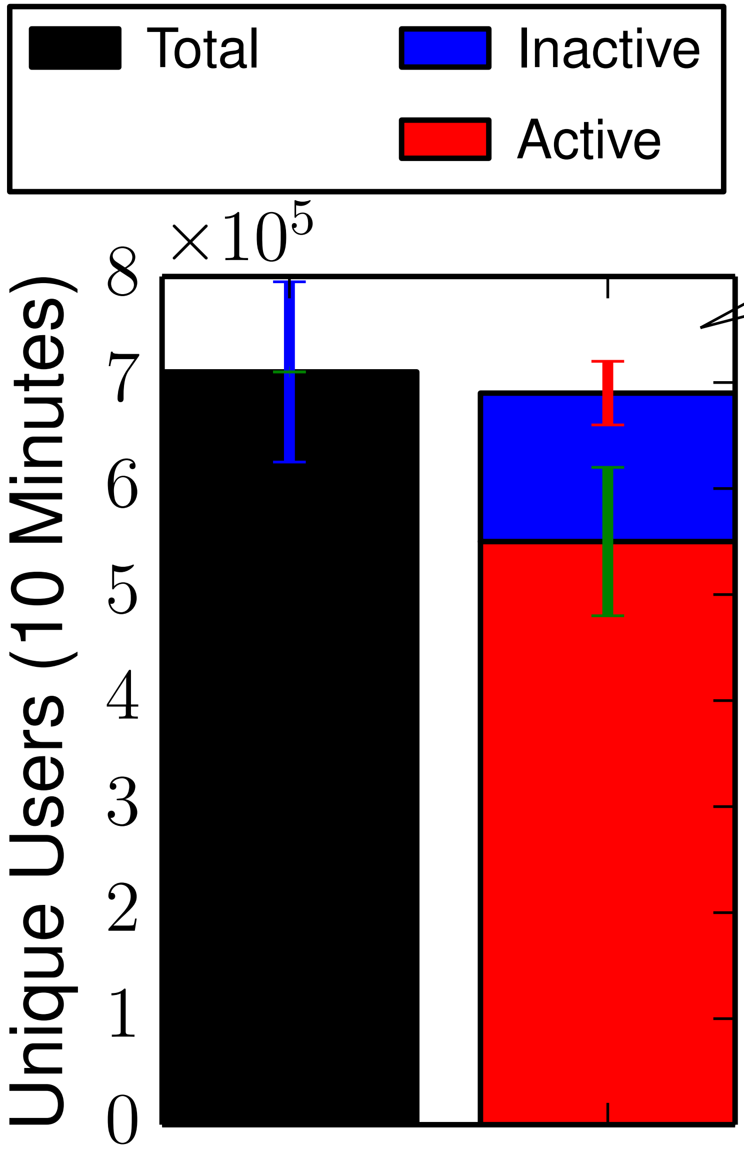
Results: Amount and Types of Traffic



Increase in web traffic – 42% in 2010 to 91% in 2016

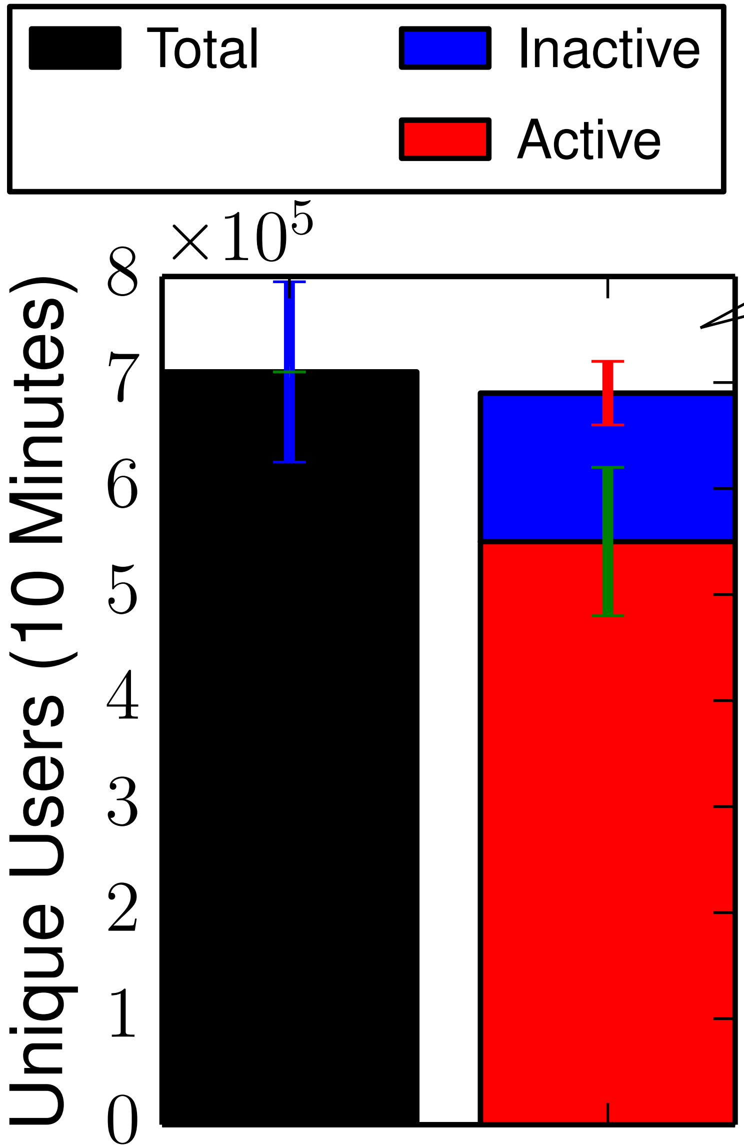
[1] PETS 2008, McCoy... [2] NSS 2010, Chaabane... [3] CCS 2016, Jansen...

Results: Number of Unique Users



710k total users
550k (77%) active users
In an average 10 mins.

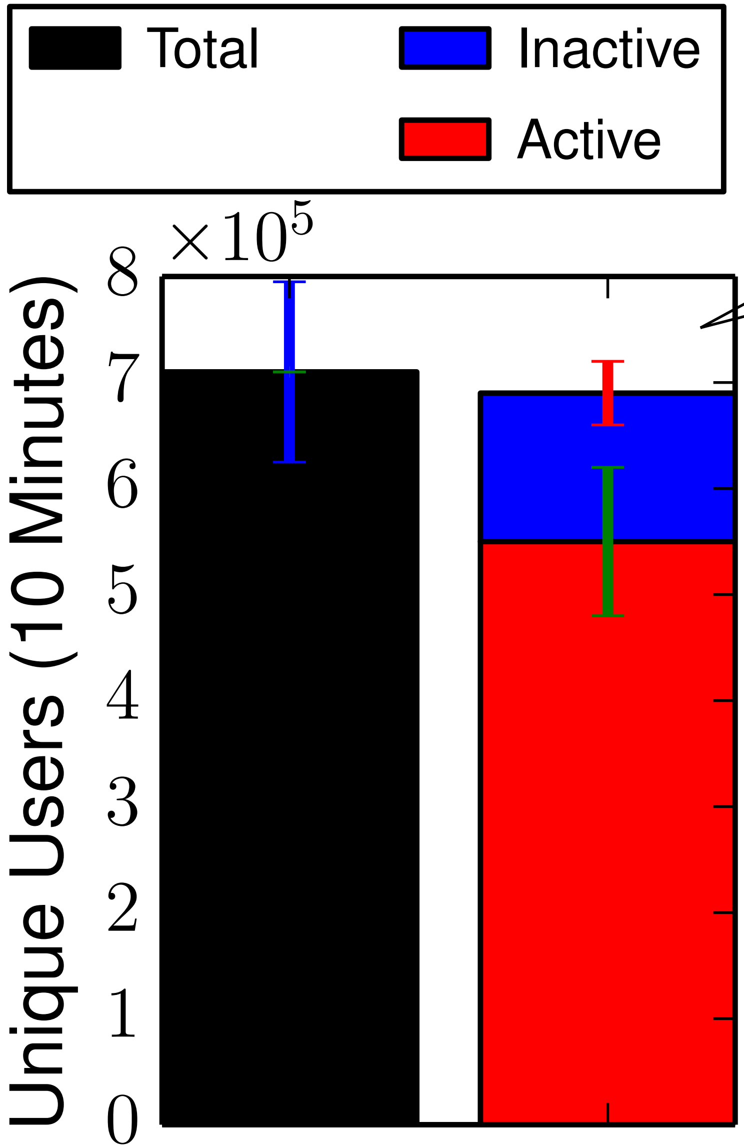
Results: Number of Unique Users



710k total users
550k (77%) active users
In an average 10 mins.

~800k – ~1.6m average *concurrent* users
(Tor Browser update pings – https://tor-metrics.shinyapps.io/webstats2/)

Results: Number of Unique Users



710k total users
550k (77%) active users
In an average 10 mins.

~800k – ~1.6m average *concurrent* users

(Tor Browser update pings – https://tor-metrics.shinyapps.io/webstats2/)

~1.75m *daily* users

(Consensus downloads – https://metrics.torproject.org)

Results: Traffic Modeling Statistics

More results in the paper!

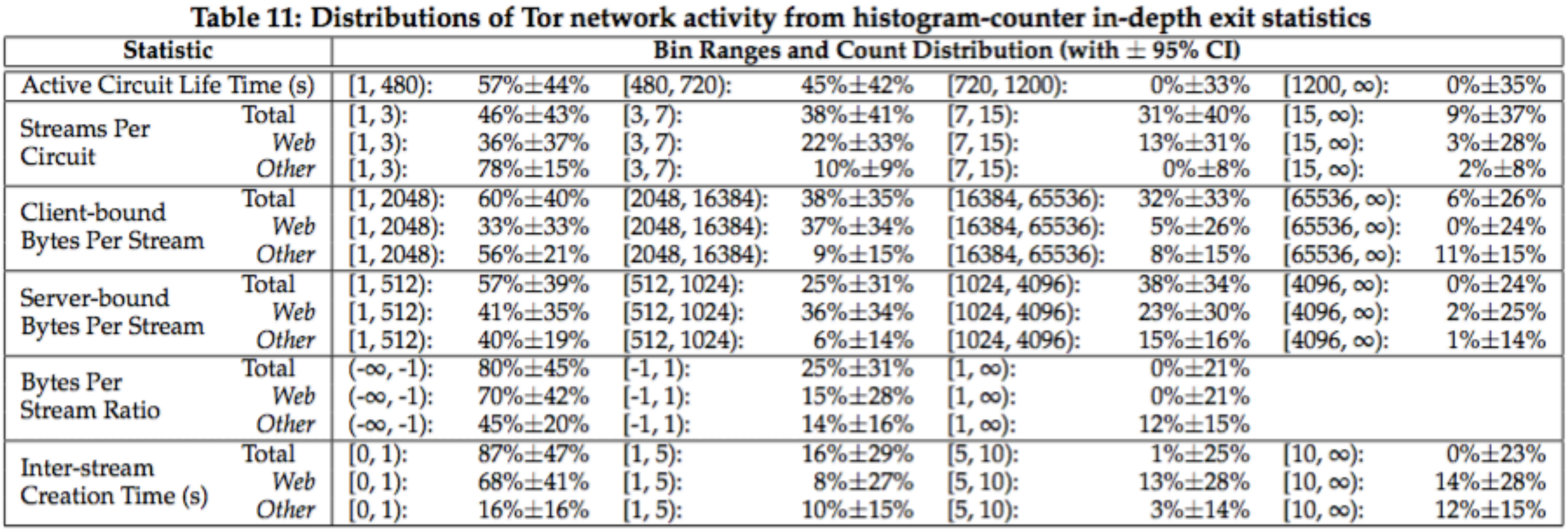


Table 11: Distributions of Tor network activity from histogram-counter in-depth exit statistics

Statistic		Bin Ranges and Count Distribution (with ± 95% CI)							
Active Circuit Life Time (s)		[1, 480):	57%±44%	[480, 720):	45%±42%	[720, 1200):	0%±33%	[1200, ∞):	0%±35%
Streams Per Circuit	Total	[1, 3):	46%±43%	[3, 7):	38%±41%	[7, 15):	31%±40%	[15, ∞):	9%±37%
	Web	[1, 3):	36%±37%	[3, 7):	22%±33%	[7, 15):	13%±31%	[15, ∞):	3%±28%
	Other	[1, 3):	78%±15%	[3, 7):	10%±9%	[7, 15):	0%±8%	[15, ∞):	2%±8%
Client-bound Bytes Per Stream	Total	[1, 2048):	60%±40%	[2048, 16384):	38%±35%	[16384, 65536):	32%±33%	[65536, ∞):	6%±26%
	Web	[1, 2048):	33%±33%	[2048, 16384):	37%±34%	[16384, 65536):	5%±26%	[65536, ∞):	0%±24%
	Other	[1, 2048):	56%±21%	[2048, 16384):	9%±15%	[16384, 65536):	8%±15%	[65536, ∞):	11%±15%
Server-bound Bytes Per Stream	Total	[1, 512):	57%±39%	[512, 1024):	25%±31%	[1024, 4096):	38%±34%	[4096, ∞):	0%±24%
	Web	[1, 512):	41%±35%	[512, 1024):	36%±34%	[1024, 4096):	23%±30%	[4096, ∞):	2%±25%
	Other	[1, 512):	40%±19%	[512, 1024):	6%±14%	[1024, 4096):	15%±16%	[4096, ∞):	1%±14%
Bytes Per Stream Ratio	Total	(-∞, -1):	80%±45%	[-1, 1):	25%±31%	[1, ∞):	0%±21%		
	Web	(-∞, -1):	70%±42%	[-1, 1):	15%±28%	[1, ∞):	0%±21%		
	Other	(-∞, -1):	45%±20%	[-1, 1):	14%±16%	[1, ∞):	12%±15%		
Inter-stream Creation Time (s)	Total	[0, 1):	87%±47%	[1, 5):	16%±29%	[5, 10):	1%±25%	[10, ∞):	0%±23%
	Web	[0, 1):	68%±41%	[1, 5):	8%±27%	[5, 10):	13%±28%	[10, ∞):	14%±28%
	Other	[0, 1):	16%±16%	[1, 5):	10%±15%	[5, 10):	3%±14%	[10, ∞):	12%±15%

Conclusion

PrivCount

- Distributed measurement system using secret sharing
- Safer Tor measurement study
- Open source: https://github.com/privcount

Future measurement plans

- Network traffic to create realistic traffic models
- Onion services to improve reliability and scalability
- Better techniques for cardinality (e.g., # unique users)
- Detecting denial of service attacks and other misbehavior

Contact

- rob.g.jansen@nrl.navy.mil, robgjansen.com, @robgjansen

Questions

PrivCount vs PrivEx

How does PrivCount enhance PrivEx

- Multi-phase iterative measurement
- Expanded privacy notion that simultaneously handles multiple types of measurements
- Optimal allocation of the ε privacy budget across multiple statistics
- Composable security definition and proof
- More capable and reliable tool
- Supports over 30 different types of Tor statistics
- Resilience to node failures and reboots
- Simpler configuration and setup

Privacy

Parameters for (ε, δ)-differential privacy

- ε = 0.3 : same as used by Tor onion service stats
- δ = 10^{-3} : upper bound on prob. of choosing noise value that violates ε-differential privacy
- DCs on 3 machines, add 3x noise

User action bounds

Action	Bound
Simultaneous open entry connections	1
Entry connection open time	24 hours
New entry connections	12
New circuits	146
New streams	30,000
Data sent or received	10 MiB