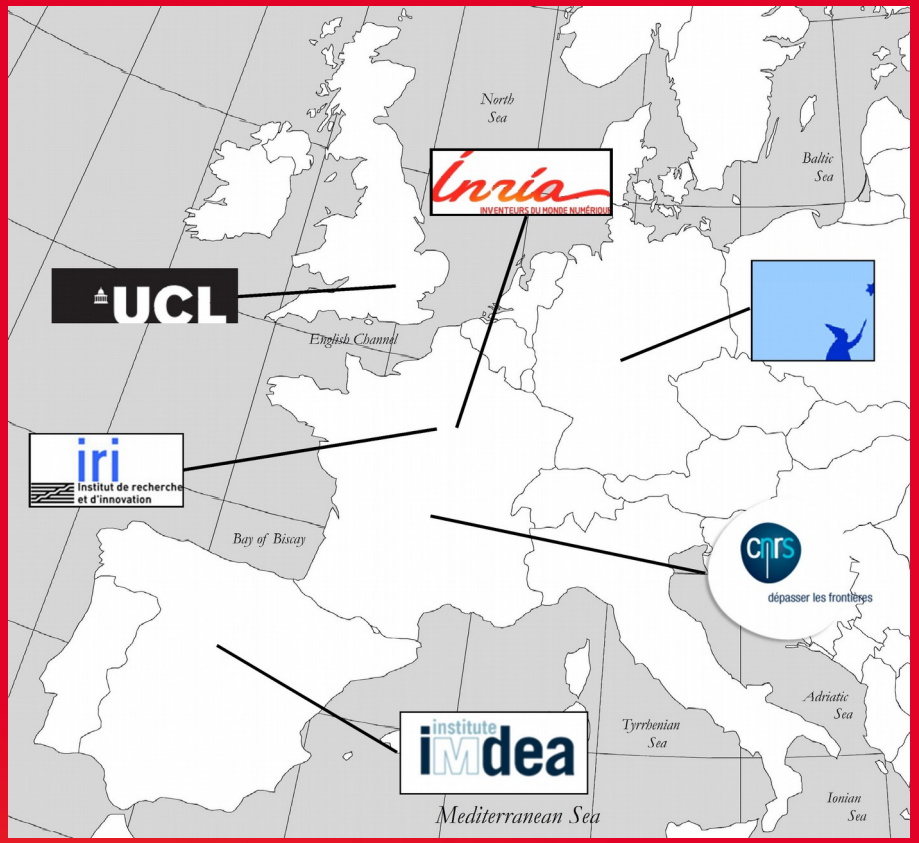


Decentralization

Privacy and Security:

W/i George Danezis (UCL),
Carmela Troncoso (IMDEA)

NEXTLEAP (nextleap.eu)
Harry Halpin
Prosecco

Harry.halpin@inria.fr
@harryhalpin

An Explosion of Decentralization

SEARCH

The New York Times

Woo Its App

Gawker, Filing for Bankruptcy After Hulk Hogan Suit, Is for Sale

Thomas J. Perkins, Pioneering Venture Capitalist in Silicon Valley, Dies at 84

Cable Industry Mobilizes Lobbying Army to Block F.C.C. Moves

TECHNOLOGY

The Web's Creator Looks to Reinvent It

By QUENTIN HARDY JUNE 7, 2016

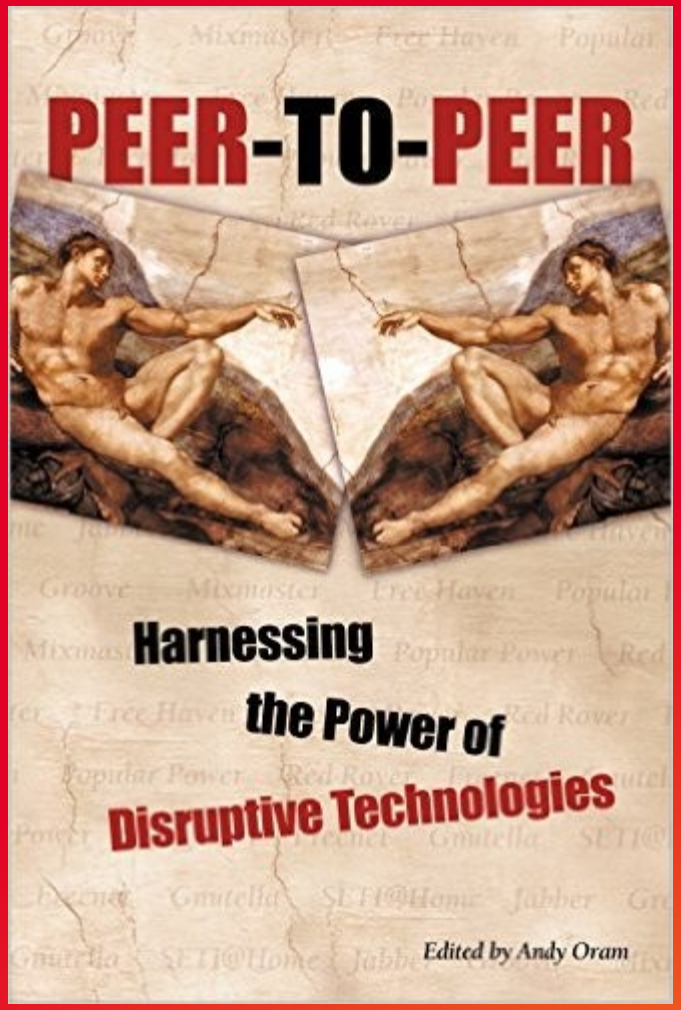
A group of top computer scientists gathered in San Francisco on Tuesday to discuss a new phase for the web.
Jason Henry for The New York Times

SAN FRANCISCO — Twenty-seven years ago, Tim Berners-Lee created the World Wide Web as a way for scientists to easily find information. It has since become the world's most powerful medium for knowledge, communications and commerce — but that doesn't mean Mr. Berners-Lee is happy with all of the

- Bitcoin
- W3C/IETF WebRTC
- Ethereum
- Patchwork
- Briar
- Tribler
- BitMessage
- Twister
- qTox
- SSB

- W3C Social Web WG (D-CENT)
- ActivityStreams
- Social Linked Data
- IndieWeb
- SwellRT (P2PValue)

The Hype Last Time : Peer-to-Peer



- **Napster (Bittorrent)**
- **Jabber**
- **SETI@Home**
- **… Skype**

Those with security/privacy considerations :

- **Freehaven (Tor)**
- **I2P**
- **Publius**
- **Jabber**
- **Mixmaster**
- **Freenet**
- **Red Rover**

Is Ross Anderson's *The Eternity Service* (1996 PRAGOCRYPT) the first system to discuss decentralization ?

Failures of central authorities

Napster (2000) → Bittorrent (2001)

Peers could share music through a centralized index.

Legal challenge in 2000 (RIAA). Ordered to keep track of activities to enforce copyright. Closes service in 2001.

E-Gold (2008) → Bitcoin (2008/2009)

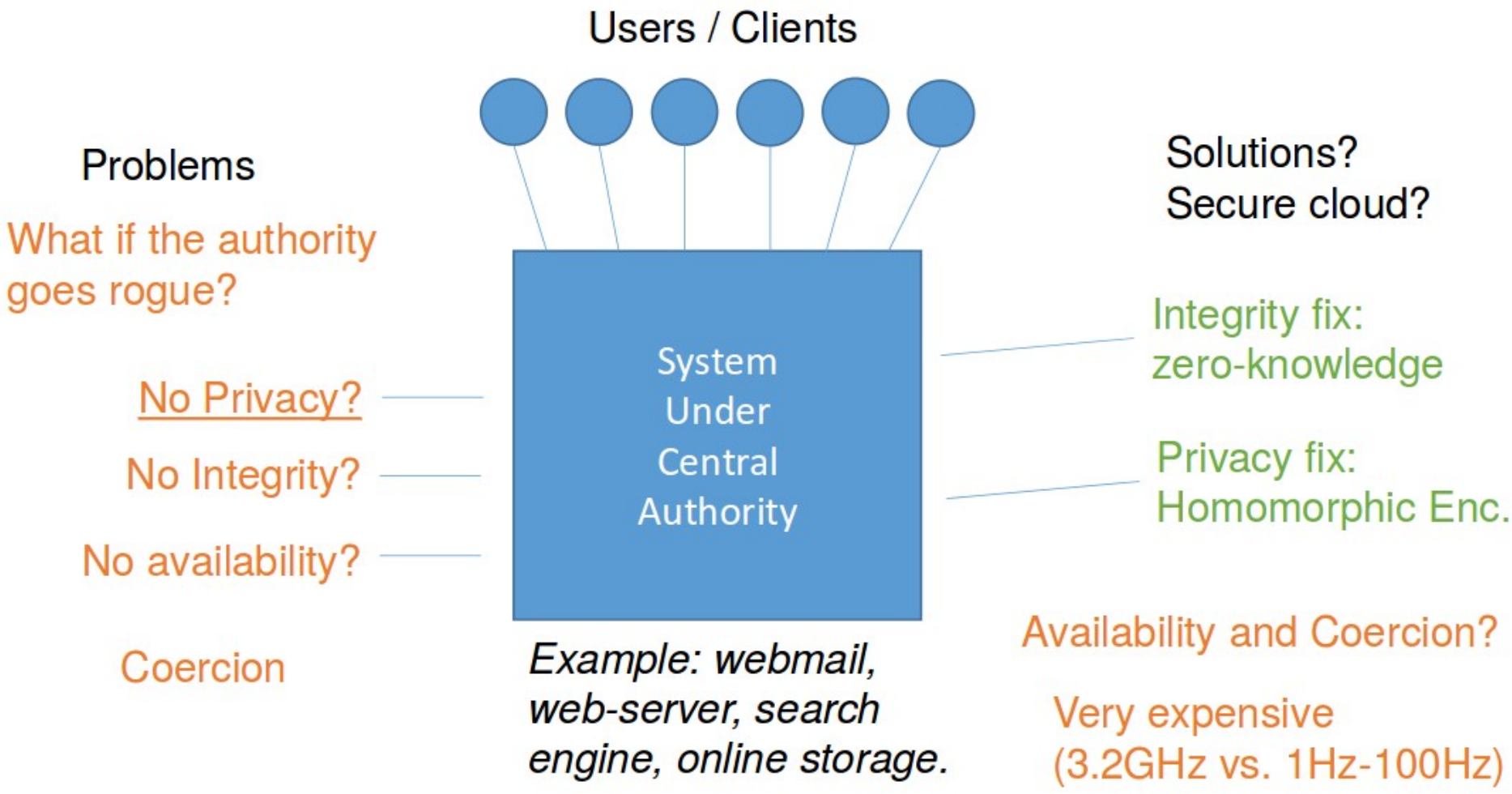
Online currency backed in grams of gold (launched 1996). Central entity kept balances and did instant trades.

2006-2008 United States Dept. of Justice prosectures as a “money transmitter” and closes 2008.

Centralized Provider Fails → Decentralized

Is secure messaging next?

A critical view of central authorities



Distributed and Decentralized

Distributed systems make use of multiple, possibly geographically separated, components usually interconnected through a network and co-ordinated via message-passing and without a single clock. Distribution is beneficial to support robustness to single component failure, scalability beyond what a single component could handle, high-availability and low-latency under distributed loads, and ecological diversity to prevent systemic failures. However, all those benefits can be achieved with a distributed system that is managed by a single authority (principal) but distributed for pratical reasons.

Google Cloud : Very large distributed system, paired datacenters, Chubby uses Paxos for distributed locks, BigTable for eventually consistent databases, Map-reduce for indexing, sharding for user availability.

Decentralized Systems are distributed systems in which multiple authorities control different components and no single authority is fully trusted by all others, in particular to perform a task with security or privacy implications. Tthere is no single entity that can act as a reference or monitor to enforce a global security or privacy policy; entities need to consider adversarial behaviour not simply by external parties, but by components of the system controlled by different authorities.

Gnutella : Many peers storing local files and flood fill search. Peers connect to other peers to ask for files. Peers download from othersSuper- peers can optimize some routing

A Mutually Beneficial Relationship or a Trade-Off for Privacy/Security?

- **Privacy:** no single trusted third-party → no (easy) surveillance?
- **Security: no** single trusted third-party → more difficult for a wide variety of attackers trying to violate confidentiality/integrity/authentication?
- **Availability:** no single entity controls file storage and retrieval → censorship resistance ?

Central Hypothesis : Is being vulnerable to a "random" subset of decentralized authorities better than being vulnerable to one for either security or privacy ?

- **How are systems decentralized ?**
- **What we gain from decentralizing?**
- **What may be lost with respect to privacy/security when decentralizing?**
- **What implicit centralized assumptions remain in decentralized systems?**

Literature review of 150+ papers from IEEE S&P, ACM CCS, Usenix SEC, NDSS, WPES, PETS, IEEE P2P.

How do nodes discover each other ?

Peer to Peer :

Open world, no central “admission control”

Vulnerabile to sybil attacks.

Examples: Bitcoin Miners, Torrent swarms

Social:

Relations of trust between nodes

Needs pre-existing social relationships ?

Examples : FreeNet, Drac

Distributed:

Well defined entities relating to each other including a distributed system with Byzantine failures.

Still has an element of centralization.

Examples: MPC, Distributed Storage, Tor relays.

Federated:

Multiple sources of authority representing users.

Centralizing tendencies

Example: Email (SMTP), Jabber

Auditing :

Commiter / Verifier distinction

Completely and availability of logs ?

Examples: Electronic voting systems, Certificate Transparency, Bitcoin

How do nodes route messages ?

Mesh:

O(n^2) channels,

run out of sockets.

Gossip:

No efficient routing

broadcast only

Example: Bitcoin mining, Gnutella, CT

Structured:

Nodes assume co-ordination positions to facilitate efficient routing.

Example: MainDHT Torrents, Tor HSDir

Restricted (Stratified, cascaded):

Specialization

Example : Tor routers (Exit, Middle, Guard

Social:

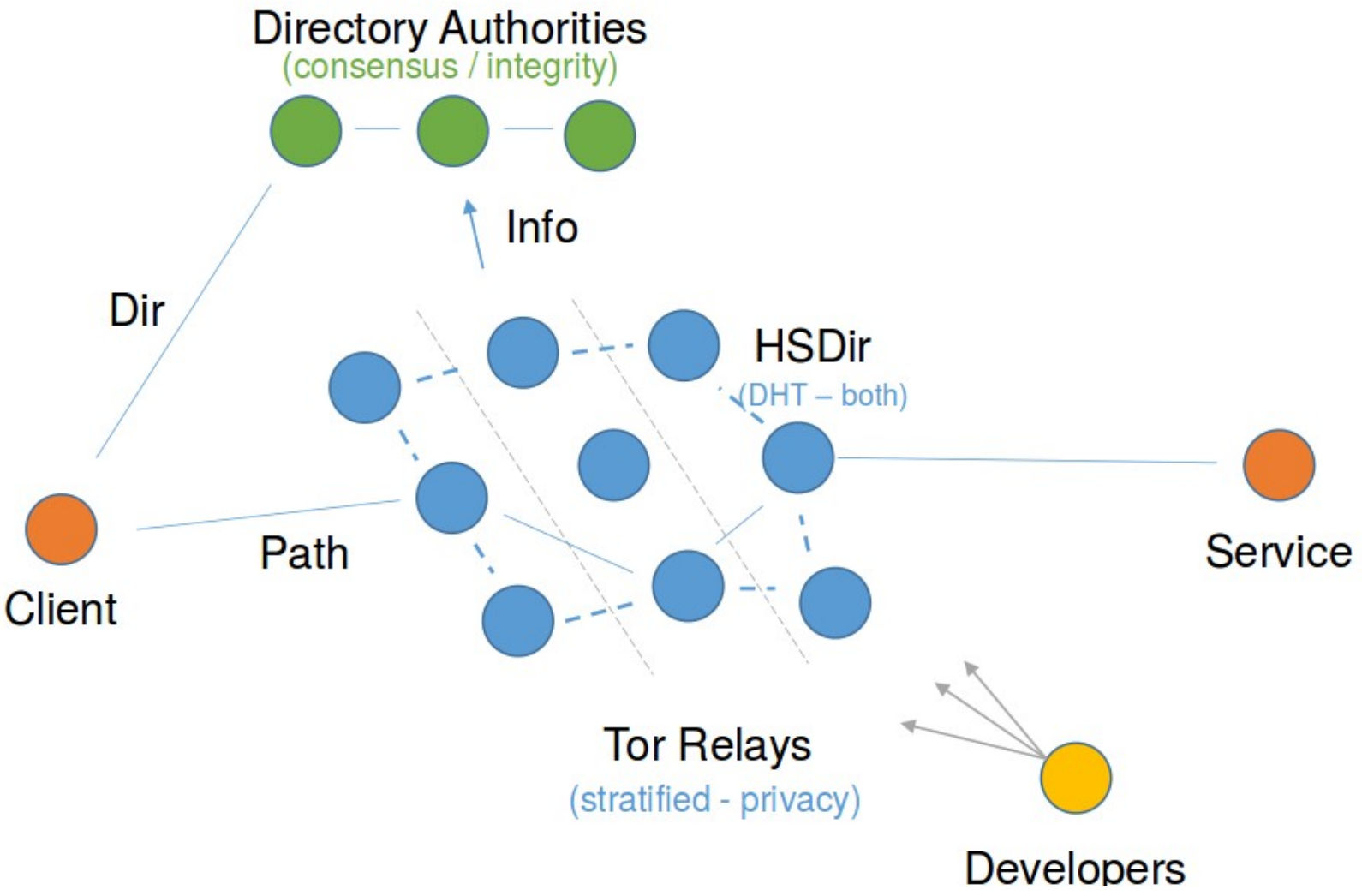
Examples: Darknet mode Freenet; MCON

Hierarchical:

Spanning tree protocols

Examples AS, BGP, SCION

Tor : Stratified Decentralization



What do we gain from decentralization ?

Security

At the heart of traditional cryptography : **Can we realize confidentiality, authentication and integrity without TTPs?**

Threshold encryption: All systems based on threshold assumptions are about distributed architectures. e.g. distributed decryption of ballots in electronic election.

Encrypt blocks and store them (availability). Joint decryption / retrieval. Distributed storage: •Original Eternity Service, Free Haven, Tahoe - LAFS, IPFS • • Private computations

SMPC: “Multi - party” assumes parties do not collude, i.e. distributed authority. Often presented as peers: example 2PC.

Privacy

At the heart of privacy-enhancing technologies : **Anonymity usually requrires a group of other users** e.g . mix network, Tor, crowds, Tarzan, election mix nets.

Hide user actions: Information theoretic Private Information retrieval (PIR) assumes a threshold of honest servers.

Censorship circumvention: Use a decentralized system for escaping censorship , e.g . Eternity, FreeHaven.

Plausible deniability: no block can be ascribed to a specific file, e.g. Tangler, Freehaven

Covertness: Traffic obfuscation against shaping (bittorrent)

Unlinkability of operations : e.g. z.cash – remove link between payer and payee in cryptocurrencies. See also Address book privacy, eg. DP5 – a private presence systems and Xbook : private social networking.

Does decentralization harm privacy and security ?

Internal adversaries: Other nodes may be controlled by the adversary. • It is not trivial to tell whether other nodes are “real”, or a mere multi - instanciation of a single adversary. Even non-adversarial nodes must be incentivised to be truthful (mechanism design)

No clear boundaries : If nodes (servers or clients) may be untrusted. Traditional security architecture with defined 'security boundaries' is not applicable. Examples: routing security in distributed hash tables (DHTs).

Content interception & traffic analysis: Actions mediated though others leads to more opportunities for content interception and metadata collection. e.g . Tor exit node and Bitcoin miners viewing all transactions.

Attacks using inconsistent views : No single authority may mean no authoritative state. A lot of work (Bitcoin mining) has to be done to ensure consistency. Example attack: different views of relays in an anonymity system.

Is there a trade-off ?

Vulnerability to one or many authorities :

Unsafe design pattern for one security property, is a good solution for the others.

High–integrity/low privacy : Bitcoin's at the cost of a public ledger, ie . little privacy. Availability in theory could be low (gossip/broadcast).

High-privacy/low integrity : Tor routes at the cost of no ability to trust nodes (and also available or correct collective statistics). High availability via directory authorities (centralization)

Maybe both ? Zerocash combines high - privacy & high - integrity “efficiently” – uses cryptographic assumptions (SNARKS)

Are minimal points of centralization needed ?

Node / peer finding / indexing seems to be centralized:

Napster : files are on user machines but information routing, indexing and search done centrally.

Tor: Distributed directory infrastructure lists all relays & attributes. Centralized enough to allow blacklisting by firewalls.

Bitcoin : Everyone gets a copy of a high-integrity log but completeness may be difficult to check.

Reputation systems ?

Question: is a lottery a decentralized state decision system?

Decentralization is a design space

How to make decentralized systems scale up: the more participants the more capacity and value (i.e. not Bitcoin or Ethereum, but Bittorrent)?

What do you need to build secure decentralized systems?

1) Deep knowledge of **distributed systems**

2) Deep knowledge of **cryptography** : necessary to achieve simultaneously privacy, security and availability.

3) **Social Incentives and Mechanism design,** game theory and sociology – otherwise selfish or unmotivated motivated actors will destroy systems. Models are very primitive.