

HS18

DISTRIBUTED SYSTEMS ADVANCED & BLOCKCHAIN

TomP2P Recap

T. Bocek

thomas.bocek@hsr.ch

■ p2p.Peer.java

- ping()
- discover()
- broadcast()
- announceShutdown()
- bootstrap()
- **sendDirect()**
- rawDataReply()
- objectDataReply()

■ Needs Peer.java and active connection

■ dht.PeerDHT.java

- add()
- put()
- get()
- digest()
- remove()
- **send()**

■ Security in TomP2P (best-effort security)

- Signature-based, no data encryption
- Messages are signed using SHA1 with DSA
- Sybil attacks!
 - This attack creates large number of identities, may collude

■ How to prevent Data from being overwritten

- Domain and entry protection, requires cooperation
- `StorageLayer.protectionDomainMode (...)`

For domains and entries		
protectionEnabled	ALL	NONE
protectionMode	NO_MASTER	MASTER_PUBLIC_KEY

■ Domain / entry protection

- Set public key `new PeerBuilder(PublicKey)`
 - Enable=ALL, Mode=NO_MASTER → every peer can protect domains, first come first served
 - Enable=NONE, Mode=NO_MASTER → no peer can protect domains
 - Enable=ALL, Mode=MASTER_PUBLIC_KEY → every peer can protect domains, the owner can claim domain
 - Enable=NONE, Mode=MASTER_PUBLIC_KEY → no peer can protect domains except the owner
- Owner of domain 0x1234 is peer where `0x1234 == hash(public_key)`

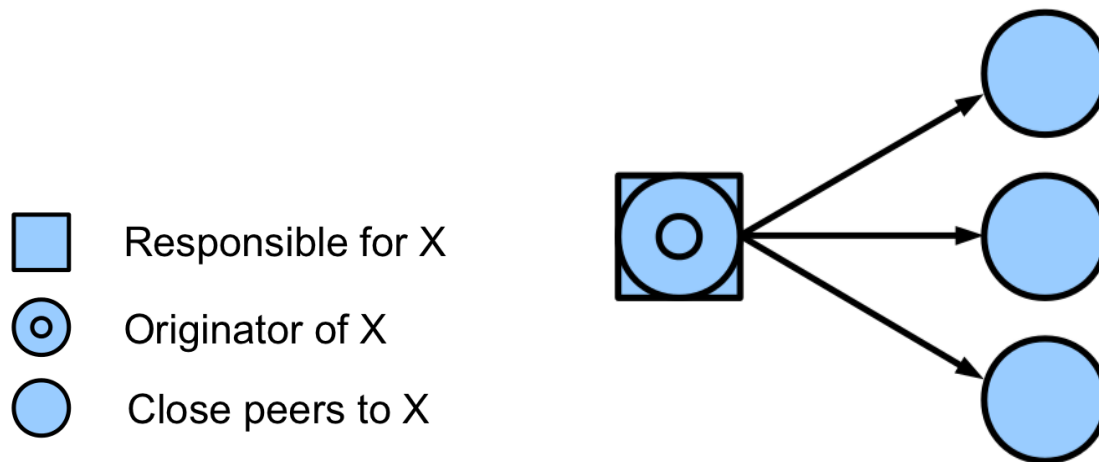
URL: b.socrative.com/student/

Room: HSR

Connectivity, Security, and Robustness

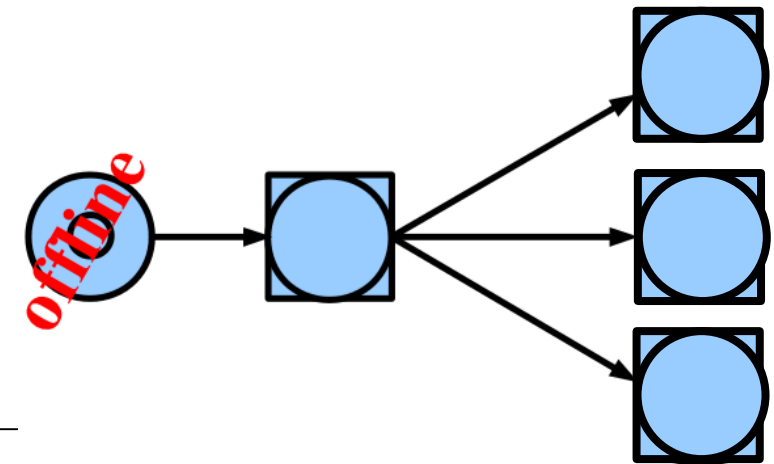
■ Direct Replication

- Replication: enough replicas
 - Originator peer is responsible
 - Periodically refresh replicas
 - Example: tracker that announces its data
- Originator offline → replicas disappear.



■ Indirect Replication

- The closest peer is responsible, originator may go offline
- Periodically checks if enough replicas exist
 - Detects if responsibility changes
 - Requires cooperation between responsible peer and originator
 - Multiple peers may think they are responsible for different versions → eventually solved



nRoot (default is 0Root)

Questions?

Prof. Dr. Thomas Bocek thomas.bocek@hsr.ch
Roman Blum roman.blum@hsr.ch