Peer-to-Peer Systems and Applications

Lecture 7: Peer-to-Peer Economics I

TECHNISCHE UNIVERSITÄT DARMSTADT

Chapter 32:

Part X: Advanced Issues

*Original slides provided by Nicolas Liebau (TU Darmstadt) and David Hausheer (University of Zurich)

0. Lecture Overview

1. Problems and Incentive Patterns

The Free Rider Problem, The Tragedy of the Commons,
The Prisoner's Dilemma, Incentive Patterns

1.1. The Free Rider Problem

- Key idea of P2P systems: Peers share resources with other peers
  - Ideally, each peer contributes as much as it uses from other peers

- However, peers are autonomous entities
  - Act in a selfish and rational way

- Definition of free rider
  - A free rider is a peer, which benefits from the effort of other peers, e.g., by downloading or searching for files, without contributing any resources or performing any tasks itself.

- Observation by Adar and Huberman in Gnutella
  - Almost 70% of the peers share no files at all
  - Nearly 50% of all query responses are returned by only 1% of the peers

1.2. The Tragedy of the Commons

- The Tragedy of the Commons
  - When individuals overuse the public good in order to maximize their own utility, they do not take into account the external costs (negative externality) that have to be borne by everyone
- Public good
  - Non-excludable in supply
    - Individuals cannot be excluded from consuming the resource
  - Non-rival in demand
    - One individual’s consumption does not diminish another user’s value of the good
  - E.g. grassland or clean air
- Externality
  - Occurs when a decision (e.g., to pollute the atmosphere) causes costs (negative externality) or benefits (positive externality) to individuals or groups other than the person making the decision

Hardin (1968): The Tragedy of the Commons, Science.

1.3. The Prisoner’s Dilemma

TECHNISCHE UNIVERSITÄT DARMSTADT

- The Prisoner’s Dilemma
  - Two peers independently select to either cooperate (e.g., share files) or defect (not share files)
  - A peer’s utility u_F for a downloading a file is higher than the cost c_F for providing the file (e.g., $u_F = 2$, $c_F = 1$)
- Payoff matrix (peers’ outcomes)

Peer 2 / *Peer 1*	*Cooperate*	*Defect*
Cooperate	R=1 / R=1	T=2 / S=-1
Defect	S=-1 / T=2	P=0 / P=0

- R: reward if both cooperate
- P: punishment if both defect
- T: temptation if one defects
- S: sucker if one cooperates

$T > R > P > S$

$R = u_F - c_F$

A rational peer will always defect, as the outcome is always higher ($T > R$, $P > S$)

Defection is the dominant strategy, although the total outcome would be higher if both cooperate ($2R > T + S$)

Flood, Drescher (1950): The Prisoner’s Dilemma (unpublished)

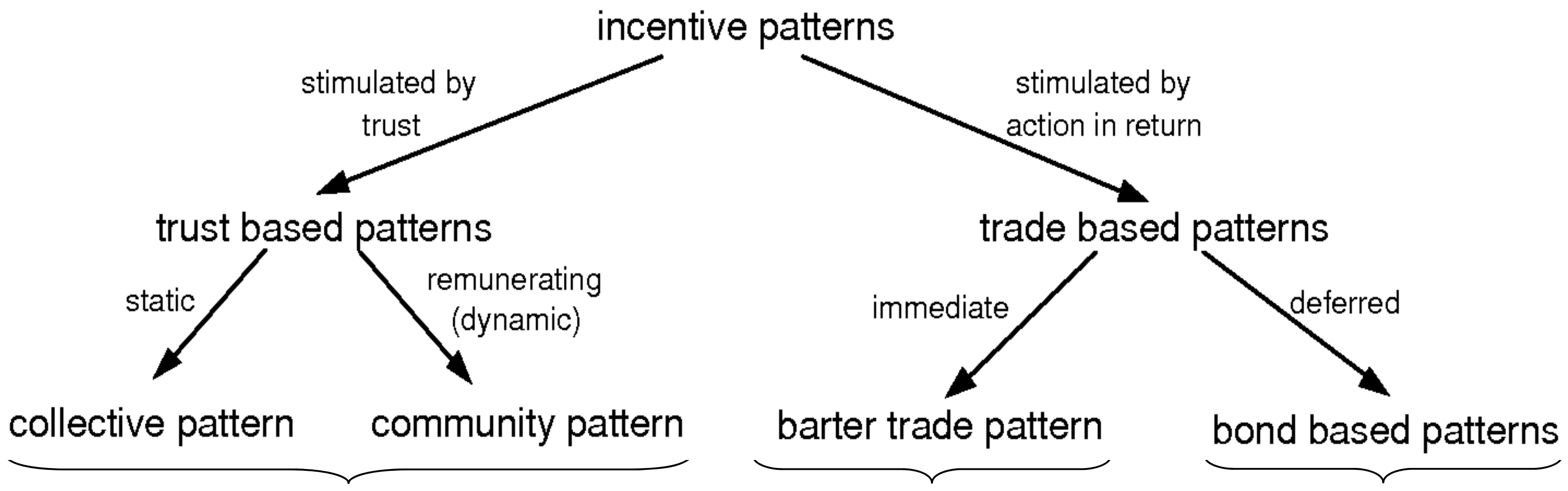
1.4. The Need for Incentives

- Cooperation is unlikely to happen without appropriate incentives for peers to share their resources
  - Can lead to a major degradation of overall performance

- Existing solutions have weaknesses
  - BitTorrent: Tit-for-tat mechanism, eMule: Credit system, KaZaA: Peer points
    - Mainly file sharing-specific
    - Weak security measures (white washing attacks etc.)

- Goal: counter weaknesses, while keeping P2P benefits (scalability, etc.)

1.4. Incentive Patterns

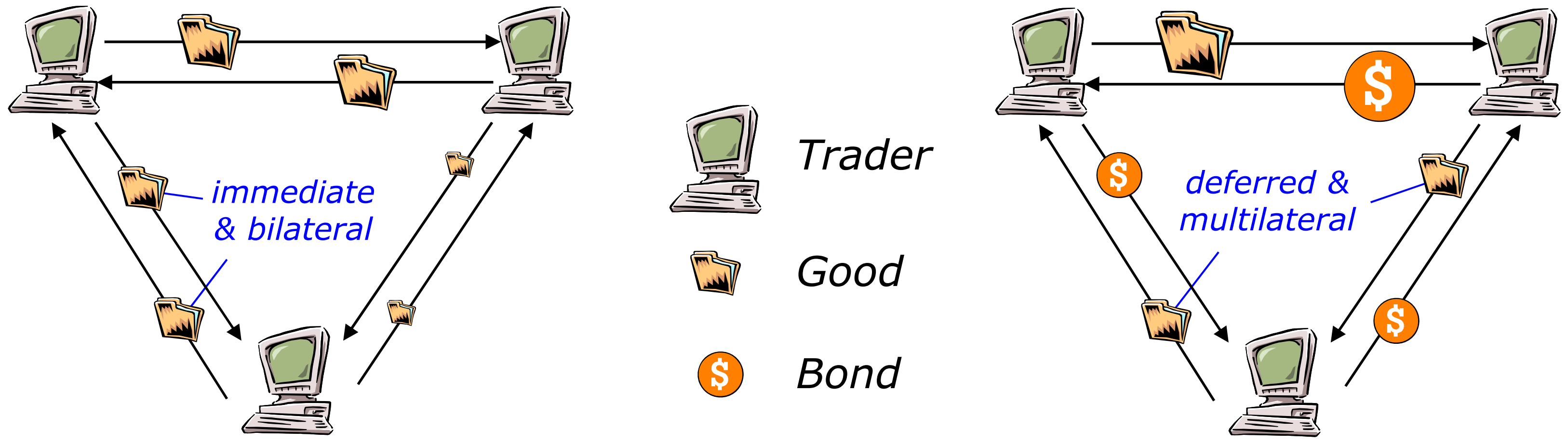


Examples	EigenTrust, Nice, EPFL Trust	BitTorrent, eMule	KaZaA, Mojo Nation
Scalability	+/-	+	+/-
Persistence	+	-	+
Anonymity	-	+	+/-
No Trust	-	+	+/-
Flexibility	+/-	-	+
Acceptance	+	+	-

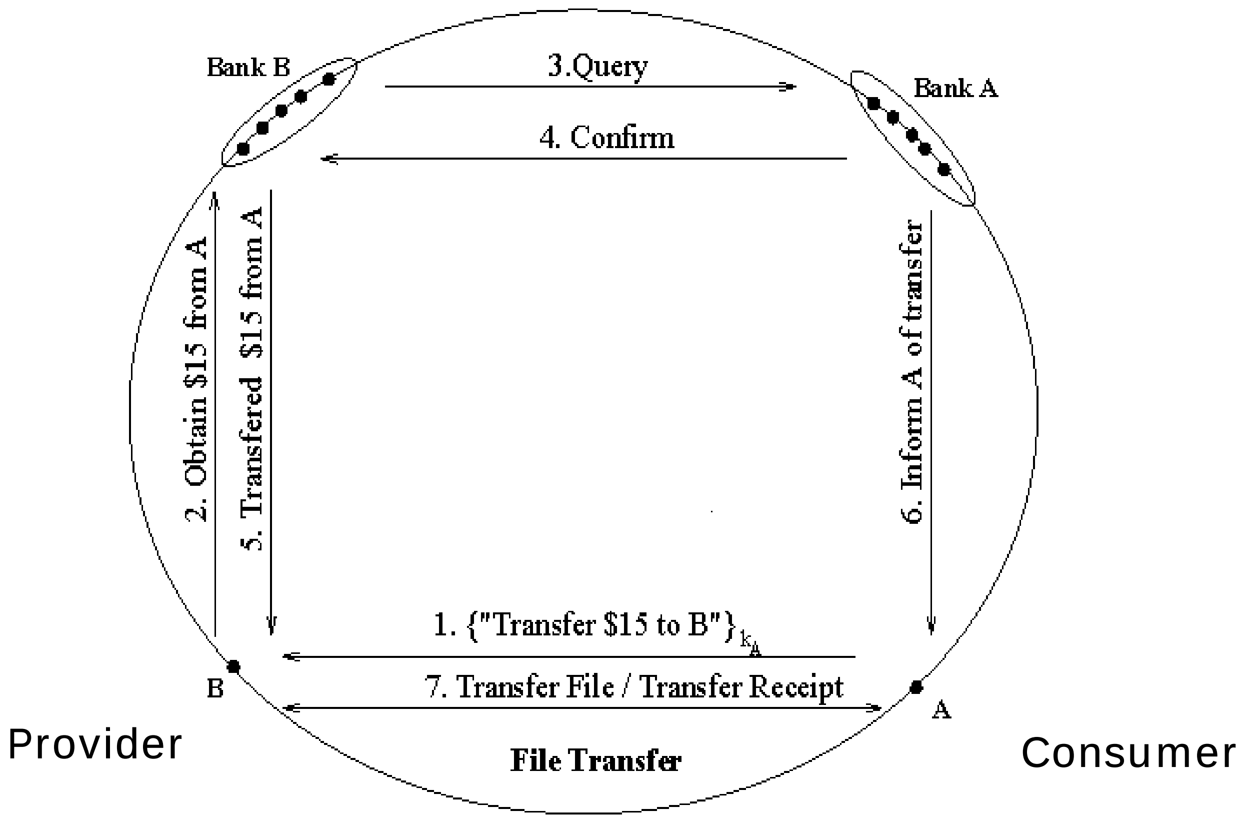
Source: P. Obreiter, J. Nimis: *A Taxonomy of Incentive Patterns*

1.4. Barter-Trade versus Bond-based Patterns

- Barter-Trade-based
  - – Only immediate & bilateral trading
  - – Exchanged goods must be of equal value
  - + Low transaction costs
- Bond-based
  - + Flexibility: deferred & multilateral trading
  - – Forgery
  - – Double-spending
  - – High transaction costs
  - – User acceptance?



1.4. Example for P2P Bond-based Pattern: KARMA



Source: V. Vishnumurthy, S. Chandrakumar, E. Sirer: KARMA: A Secure Economic Framework for Peer-to-Peer Resource; P2P Economics, June 2003.

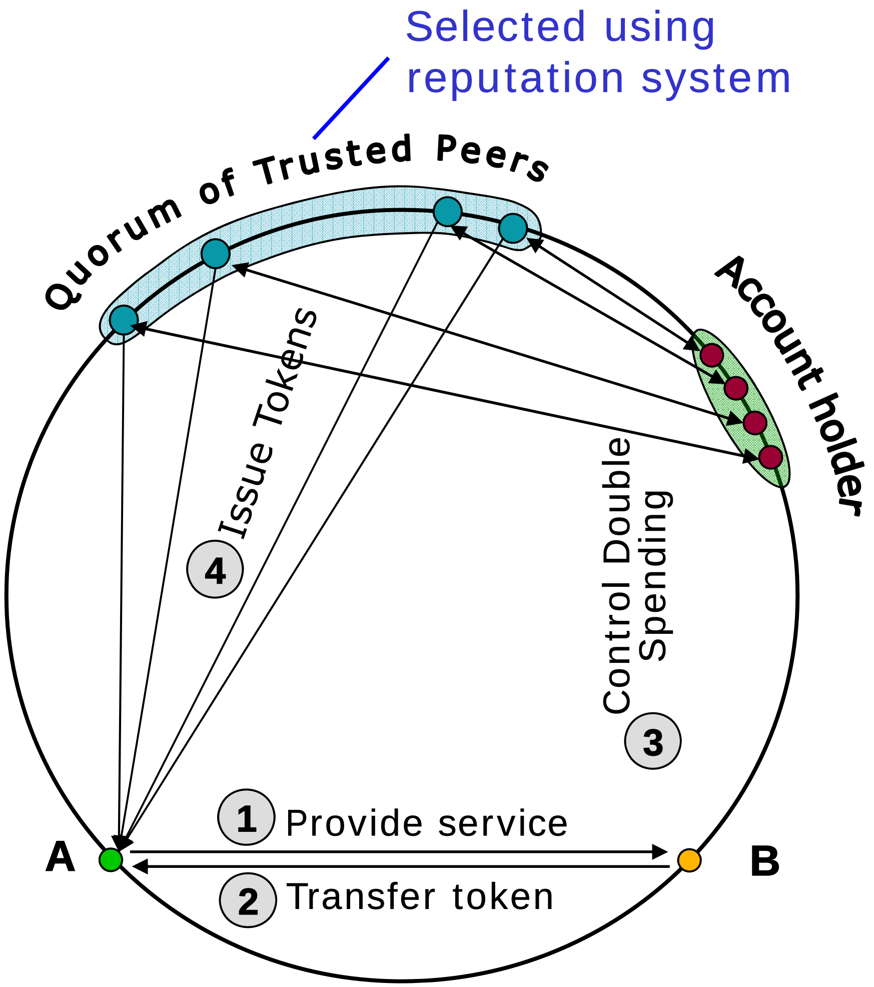
2. Token-based Accounting

Token Structure, Token Transaction, Token Exchange, Protection of Double Spending, Conclusions

N. Liebau et al.: *Token-Based Accounting for P2P-Systems*; KiVS 2005.

2. Token-based Accounting

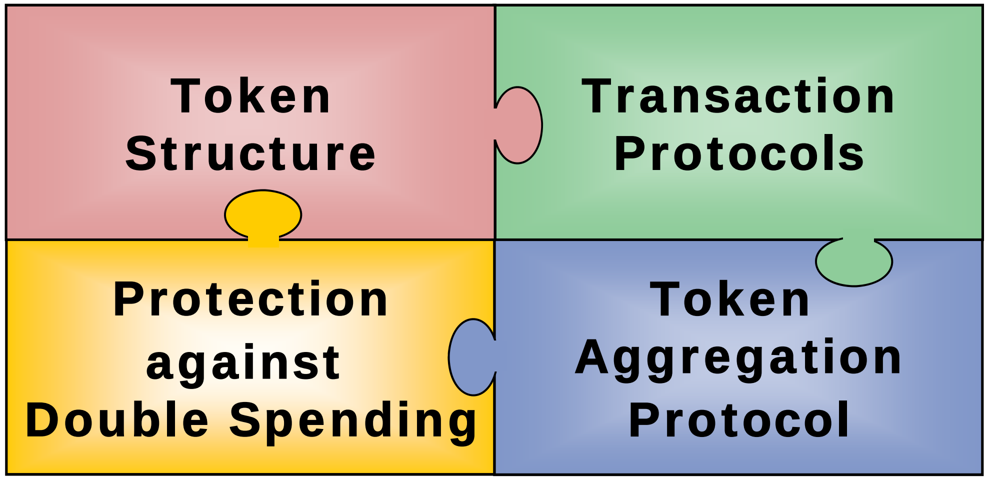
- Requirement
  - Create transaction receipts
- Design decision
  - Do not involve a 3rd peer in transaction
- Goal
  - Enforce behavior
- Strategy
  - Receipts are scarce
- Result
  - Receipts must be issued
- Tokens = Issued Receipts



2. Building Blocks

- Transaction receipts
- General exchange medium
- Prevent forgery, prevent robbery

- Peers exchange services
  - Peers cannot re-spend tokens

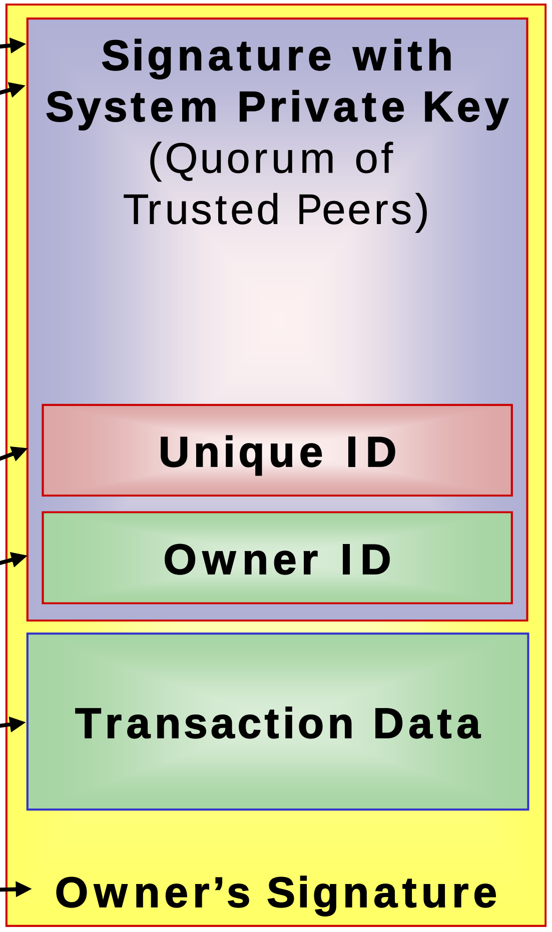

Token Structure
Transaction Protocols
Protection against Double Spending
Token Aggregation Protocol

- Detect & trace double spending

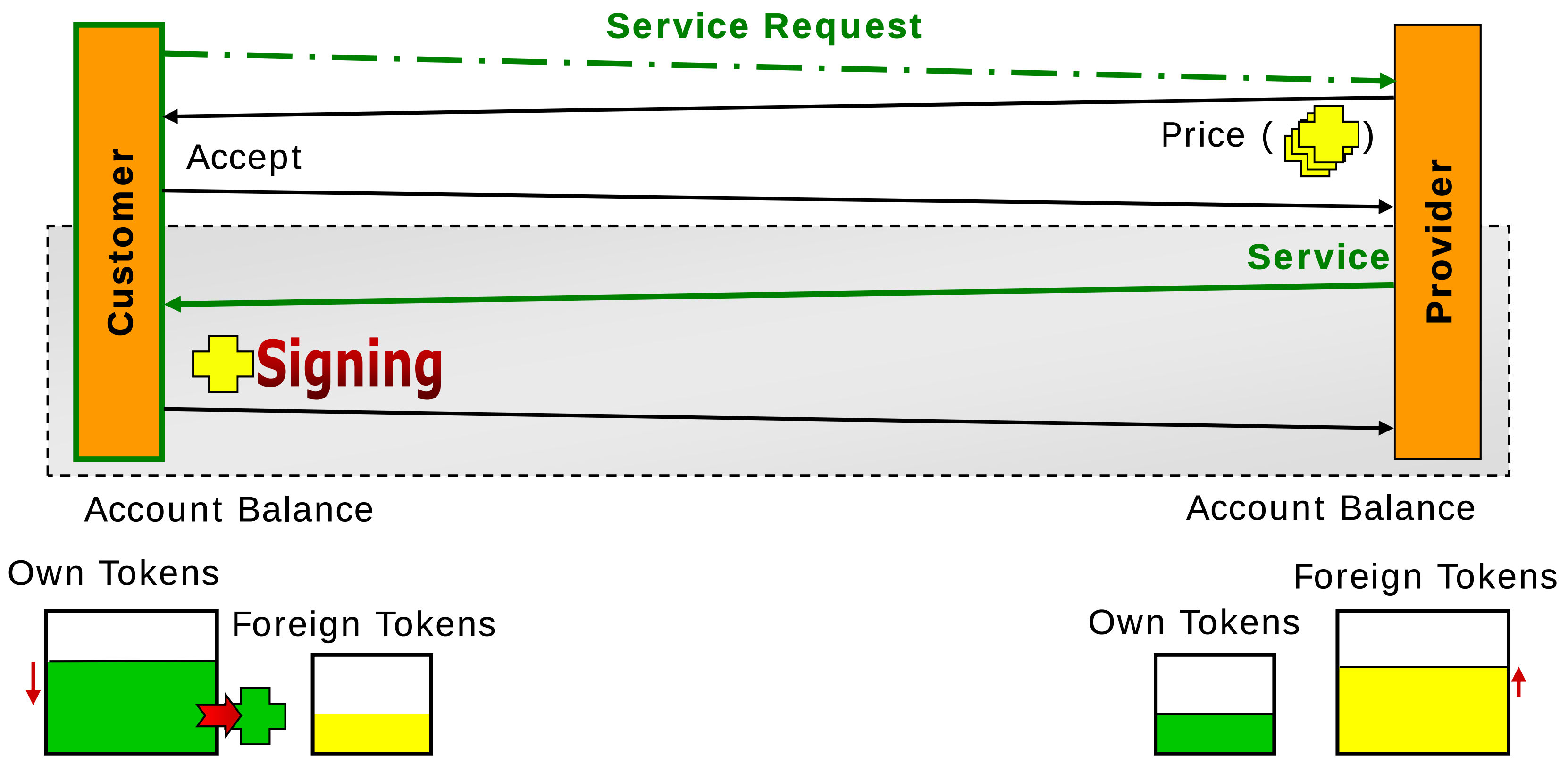
- Issue new tokens
- Exchange of foreign tokens against new own tokens

2.1. Token Structure

- Functions
  - Issue token
    - Sign for validation
  - Prevent forgery
    - Sign tokens
      - With system key
      - In compliance with P2P
  - Prevent double spending
    - Unique token identification
    - Trace to double spender
      - No anonymity required
  - Receipt
    - Avoid falsification



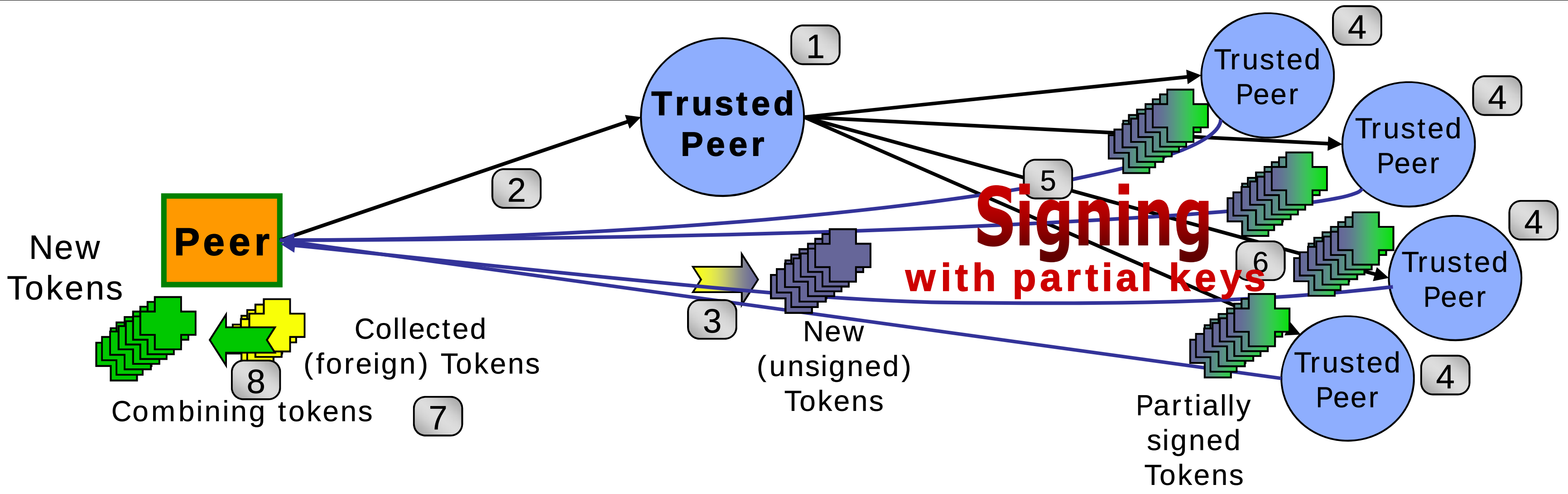
2.2. Using a Token in a Transaction



Exchange-Options

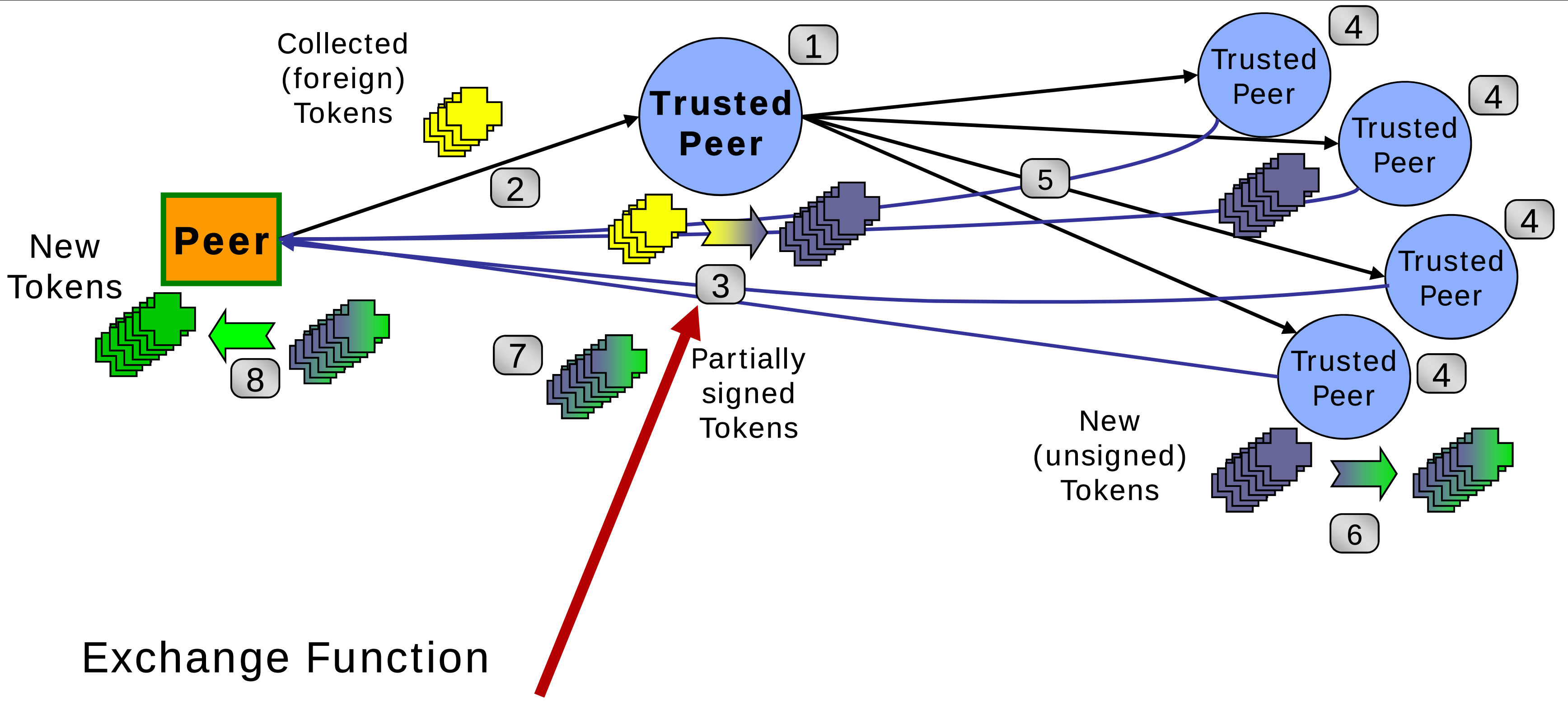
- Post
- Pre
- During transaction (after 1 min., 1 MB, etc.)
- Trustworthy exchange

2.3. Token Exchange (1)



- Sign Token to Issue It
  - A system-wide private/public key pair is needed
- Signing in Peer-to-Peer Way
  - Threshold Cryptography
    - Enables distributed signing/encryption of documents
    - Every trusted peer owns a part of system-wide private key
    - Signing without compromising system private key

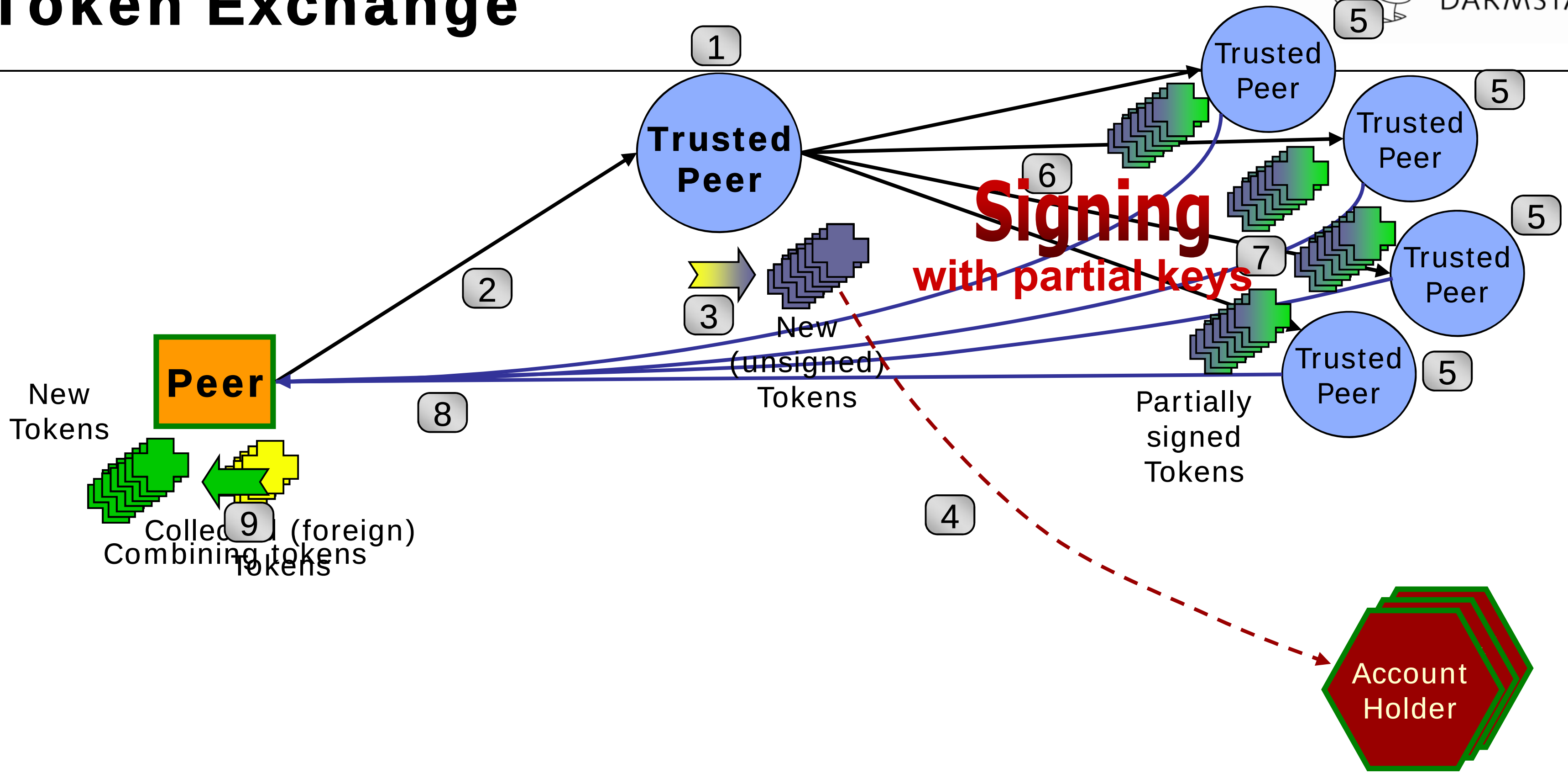
2.3. Token Exchange (2)



Exchange Function

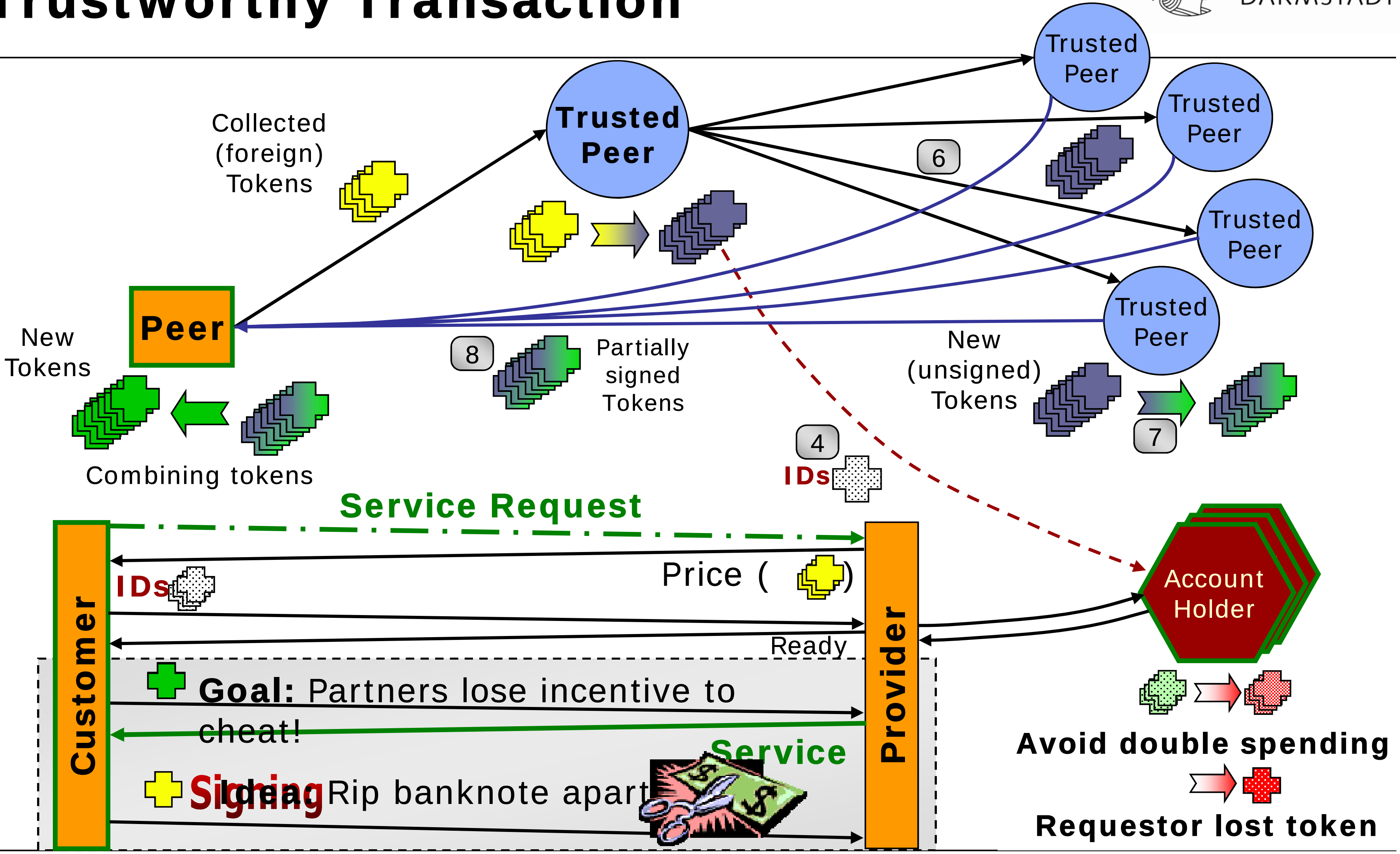
- Multiple options reflecting
  - Different usage policies
  - Different economic systems

2.4. Protection of Double Spending: Token Exchange



Avoid double spending

2.4. Protection of Double Spending: Trustworthy Transaction



Goal: Partners lose incentive to cheat!

Idea: Rip banknote apart

2.5. Token-based Accounting: Conclusions

- Implementation
  - Based on JXTA
- Advantages
  - Flexible
  - Trustworthy
- Evaluation
  - Low traffic overhead
  - Practical values for system variables
    - Quorum size 7 – 10
    - Account holder set 4 - 16
- Current work
  - Does an asymmetric exchange function lead to inflation?

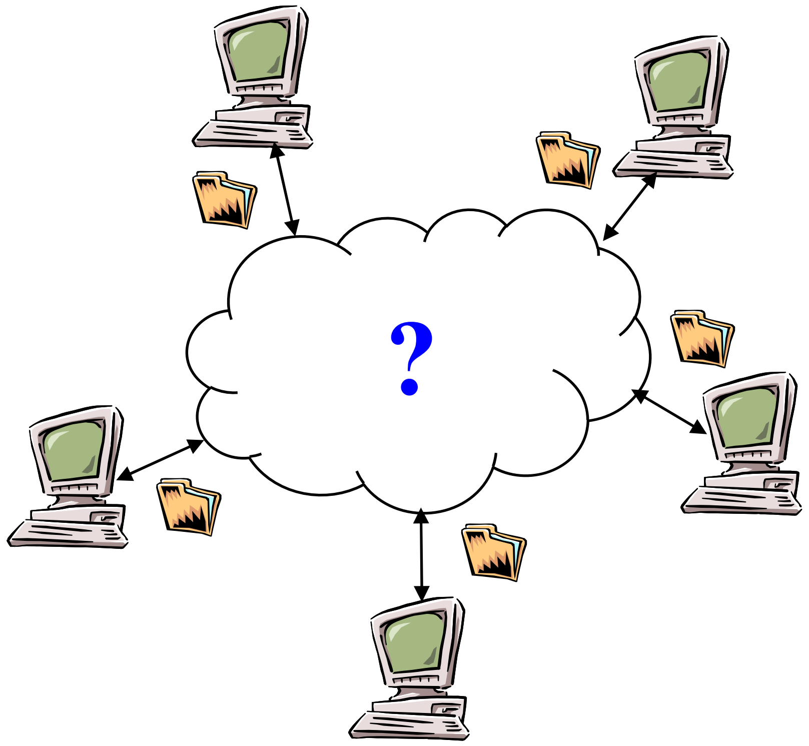
3. PeerMart: Peer-to-Peer Auctions

Architecture, Design and Concept, Implementation and Evaluation

D. Hausheer, B. Stiller: *Decentralized Auction-based Pricing with PeerMart*; IFIP/IEEE IM 2005.

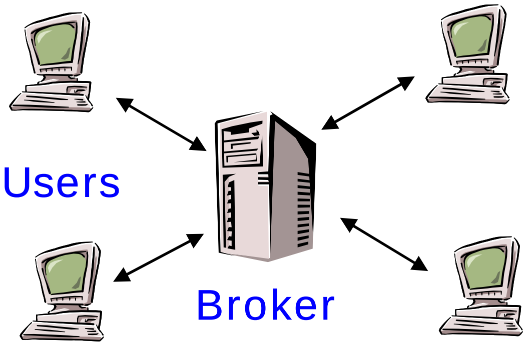
3. PeerMart: Peer-to-Peer Auctions

- Key Question
  - How to build a technically and economically feasible marketplace for trading services over the Internet?

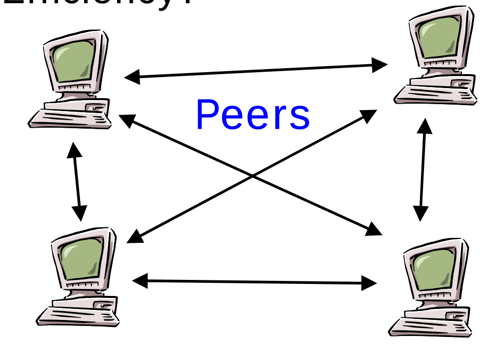


3. PeerMart: Motivation and Goal of Work

- Centralized Marketplace
  - + Efficiency
  - - Single Point of Failure
  - - Vulnerable against attacks
  - - Scalability?



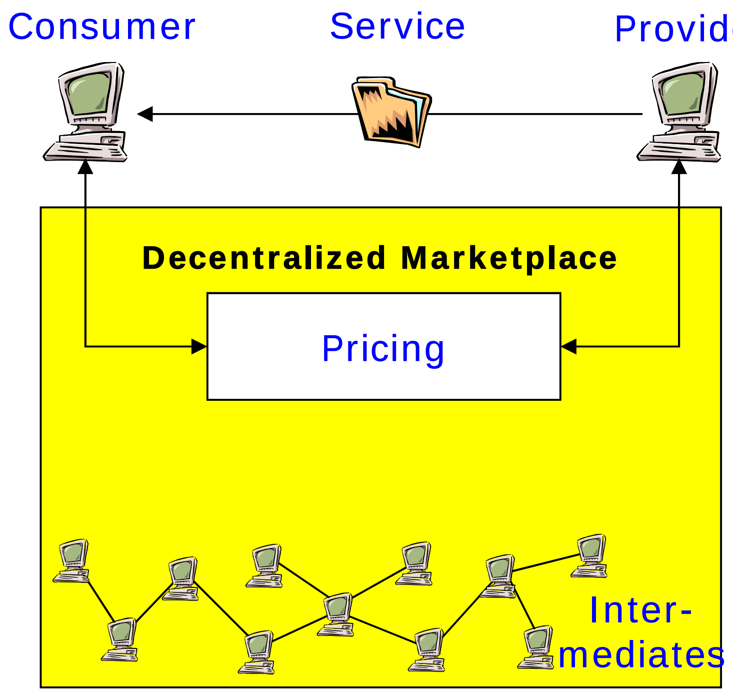
- Decentralized Marketplace
  - + Extensibility
  - + Fault-tolerance
  - - Vulnerable against selfish and malicious behavior of peers
  - - Efficiency?



Goal of Work

Design and implement efficient and scalable trading mechanisms in support of a decentralized and reliable marketplace

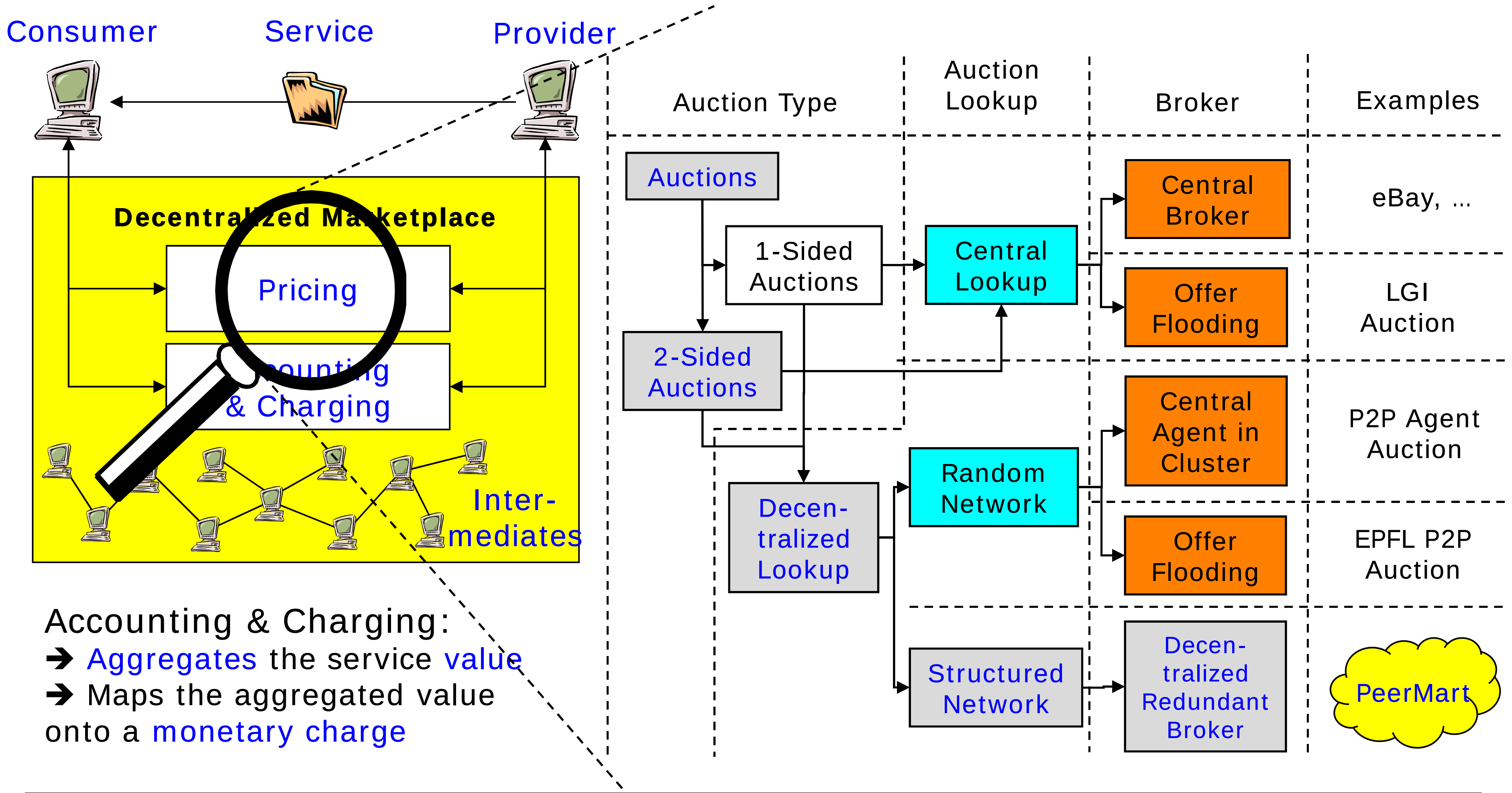
3.1. PeerMart: Architecture and Design Space



Pricing:

- Price *determination*:
  ➔ Maps a service onto a value (price)
- Price *dissemination*:
  ➔ Communicates price to other peers

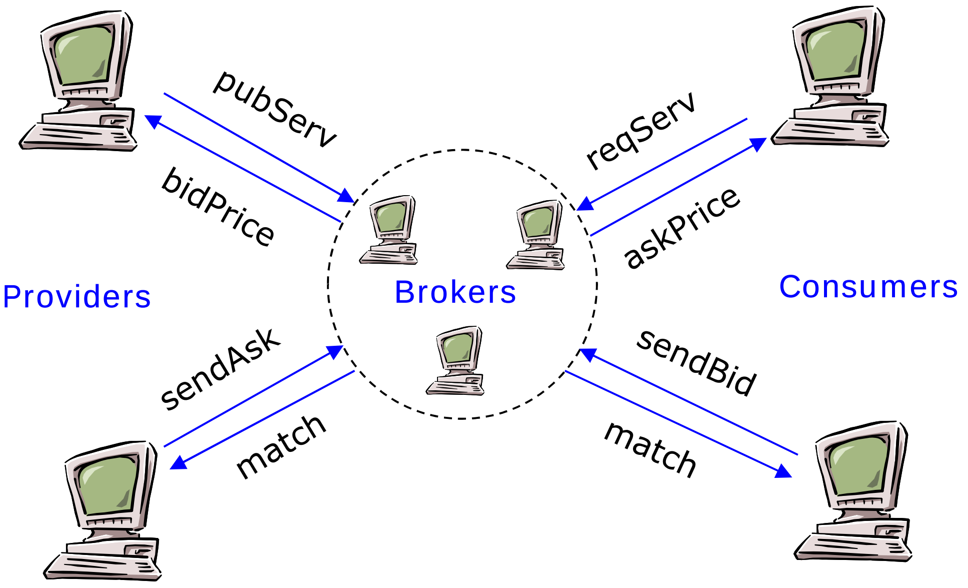
3.1. PeerMart: Architecture and Design Space

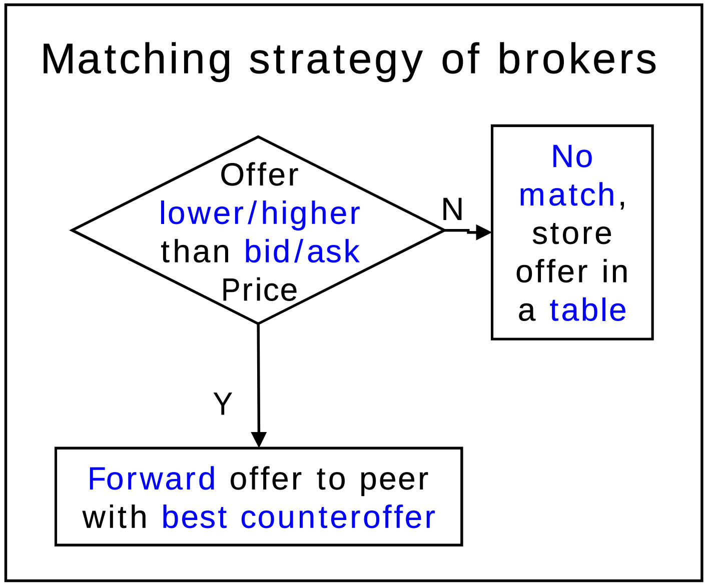


Accounting & Charging:
➔ Aggregates the service value
➔ Maps the aggregated value onto a monetary charge

3.2. PeerMart Concept: Decentralized Double Auctions

- Basic Concept
  - Each service is traded in a Double Auction
  - Each auction is mapped onto a cluster of broker peers



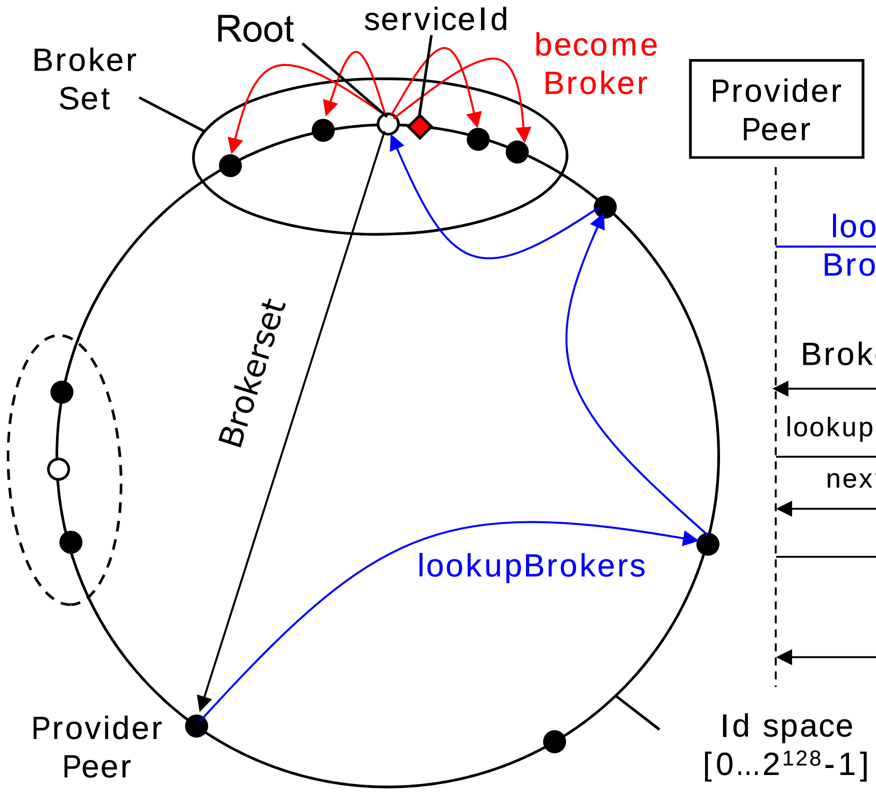


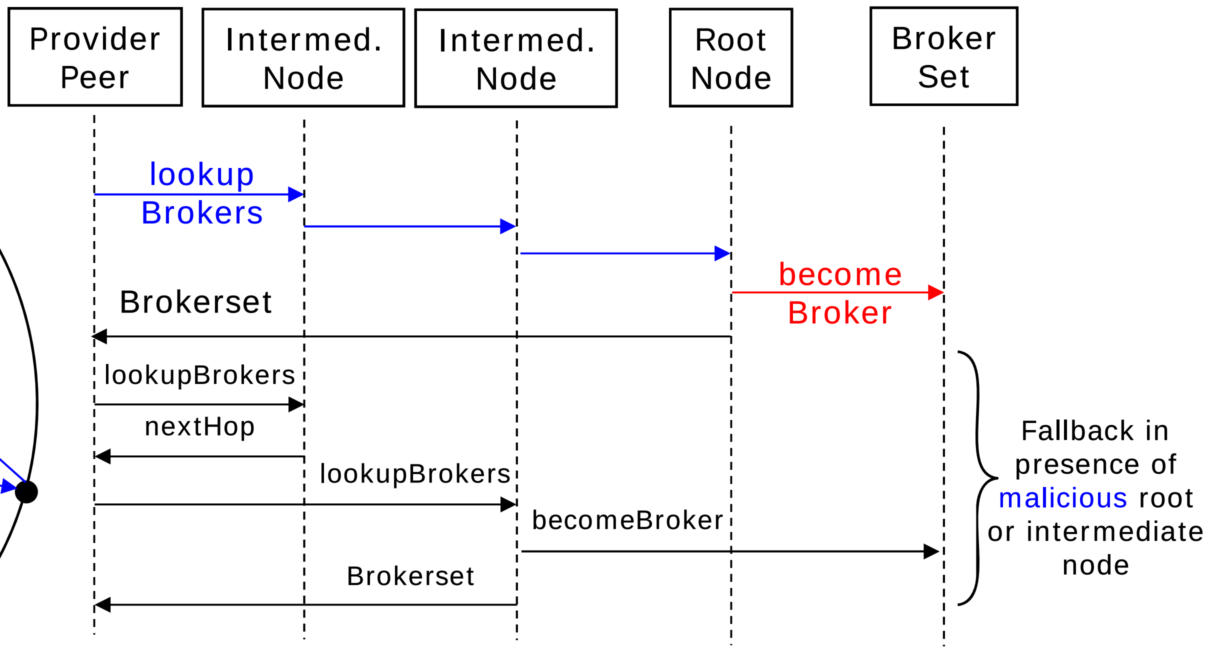
Matching strategy of brokers
Offer lower/higher than bid/ask Price
N
No match, store offer in a table
Y
Forward offer to peer with best counteroffer

3.2. PeerMart Design: Broker Set Lookup

TECHNISCHE UNIVERSITÄT DARMSTADT

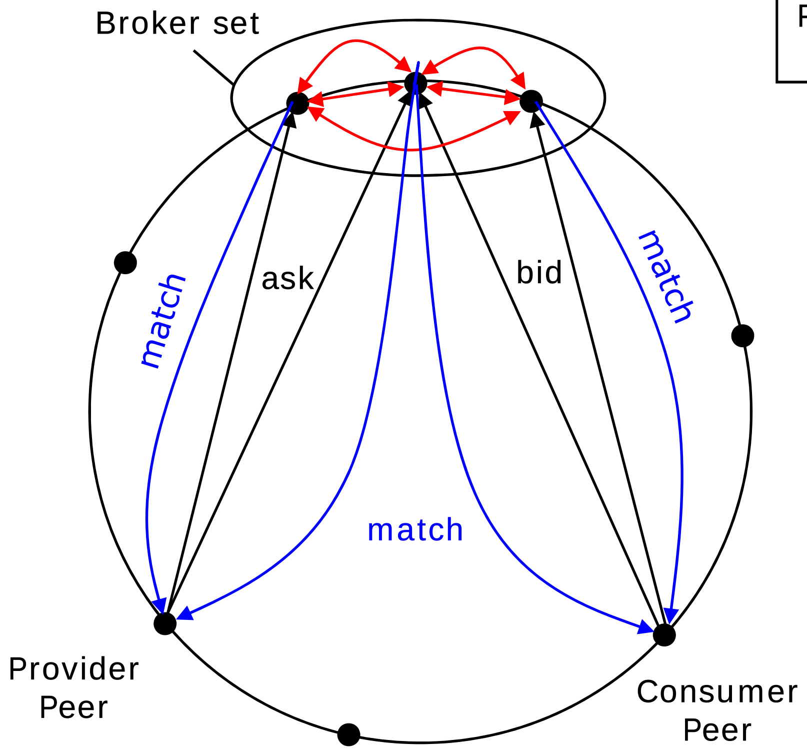
- Each peer has a unique nodeId, peers form a structured P2P overlay network
  - Each peer has public/private key pair, certified offline and bound to nodeId
  - Services have unique serviceId, *n* peers numerically closest to serviceId form brokerset

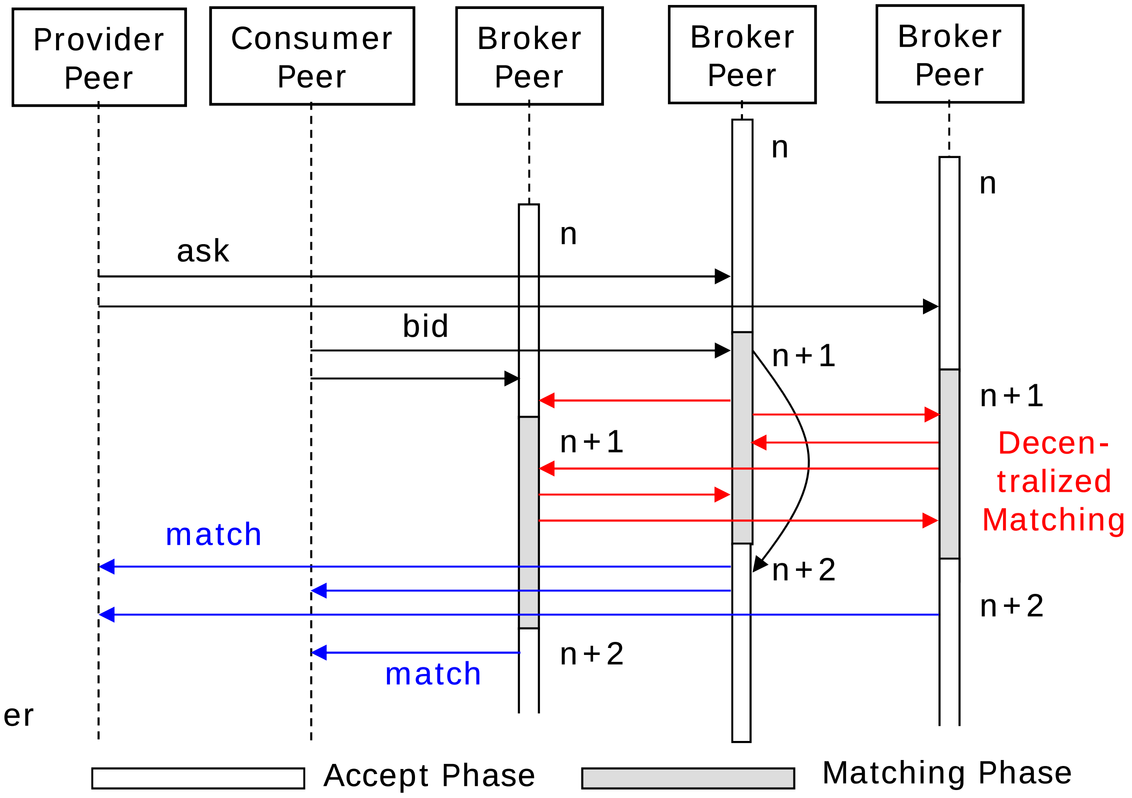




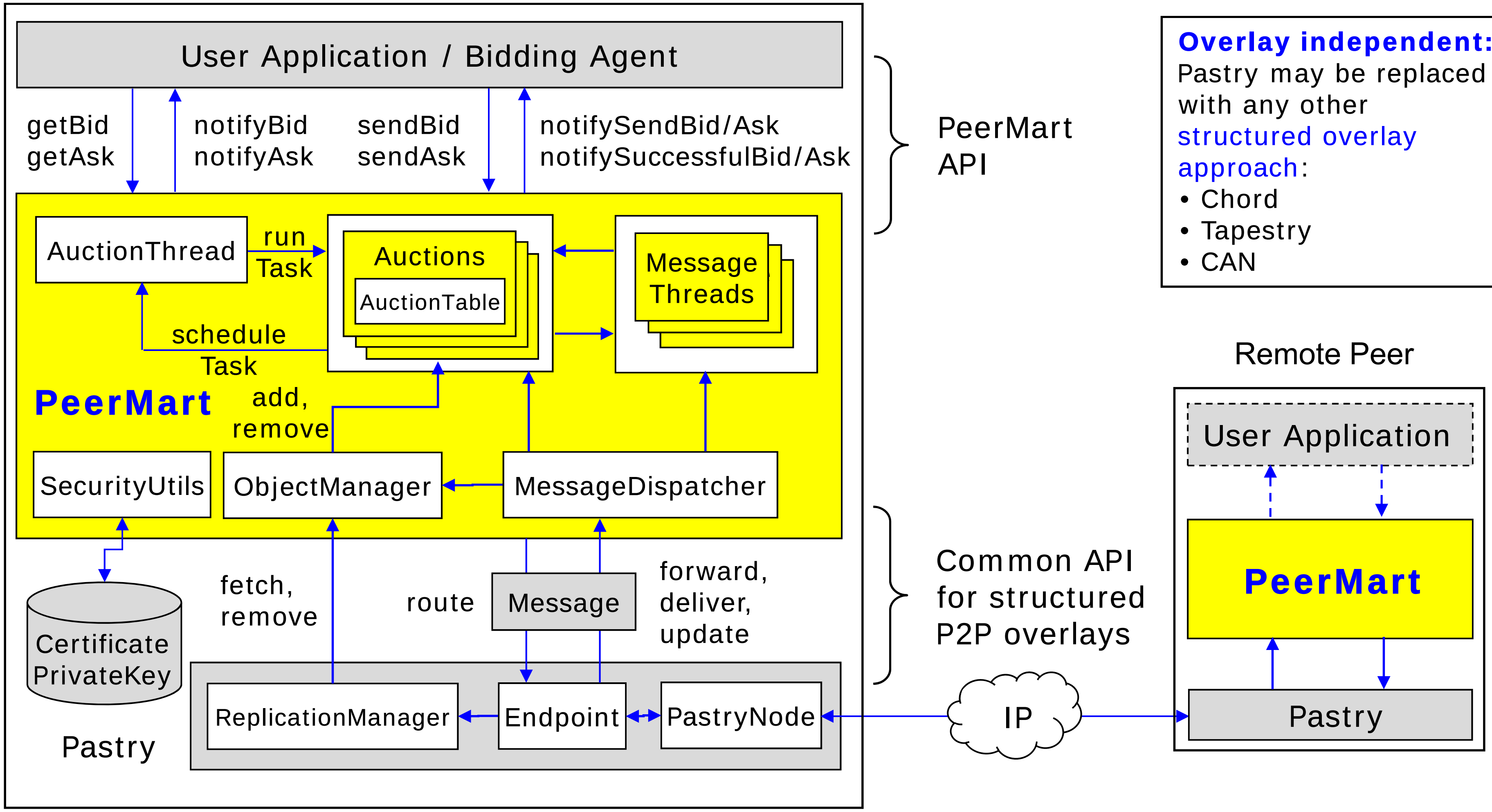
3.2. PeerMart Design: Matching Process

- Peers send price offers to a random subset of *f* broker peers
  - A slotted time is used to tackle message delays between peers
  - Brokers forward candidates for a match, matches determined by majority decisions





3.3. PeerMart: Prototype Implementation in Java on top of Pastry



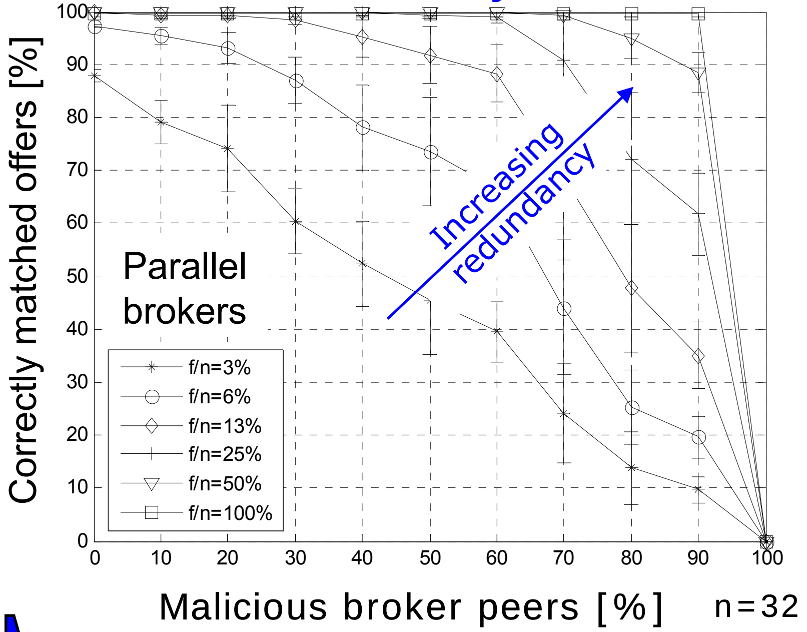
Overlay independent:
Pastry may be replaced with any other structured overlay approach:
- Chord
- Tapestry
- CAN

3.3. PeerMart Evaluation

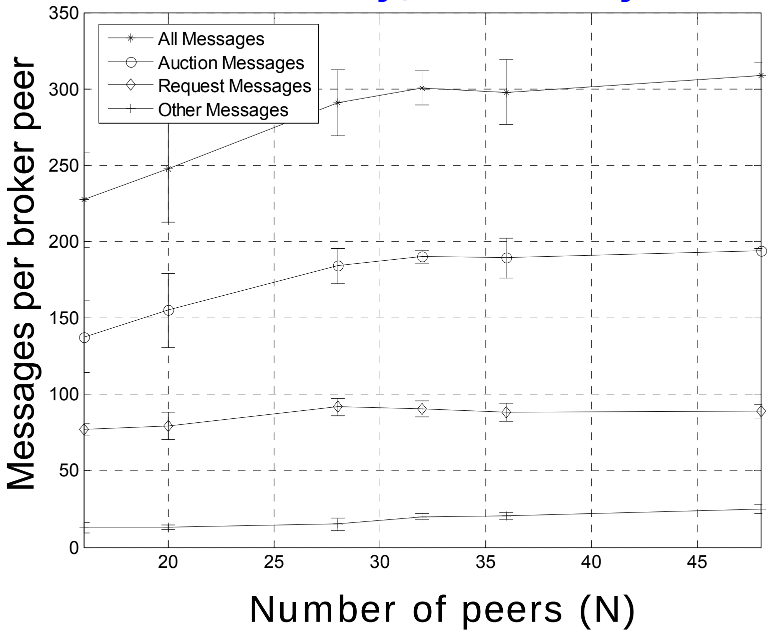
Assumptions:

- Each peer randomly assigned to fixed set of services, # services increased linearly to # peers
- Price offers normally distributed, all offers randomly delayed
- Malicious peers modeled as brokers which do not forward offers to other brokers

Reliability



Efficiency, Scalability



Results:

- PeerMart correctly matches offer pairs for < 70% malicious peers using 50% parallel brokers
- If fraction of malicious peers is < 25%, even 12.5% parallel brokers provide good reliability
- Scales well (moderate overhead)

3.4. Potential Attacks: Corrupt Leaf-sets

- Assumption
  - Attacker has compromised 10% of a 10^6 node network
  - Compromised nodes are uniformly distributed

- Number of nodes controlled in a given 64-member leaf-set
  - X is the number of nodes controlled in a given leaf-set
  - $E[X] = 6.4$
  - Probability of acquiring the majority of a leaf-set:
    - $P(X > 32) = P(X > (1 + 4)6.4) < (e^4 / 5^5)^{6.4} = 5.6 * 10^{-12}$
  - Probability of controlling the majority in some leaf-set:
    - $5.6 * 10^{-12} *$ (total number of leaf-sets) $= 5.6 * 10^{-6}$

From: V. Vishnumurthy, S. Chandrakumar, E. Sirer: KARMA: A Secure Economic Framework for Peer-to-Peer Resource; P2P Economics, June 2003.

3.4. Potential Attacks: Wrong Leaf-sets

TECHNISCHE UNIVERSITÄT DARMSTADT

- Routing Failure Test
  - All nodeIds in set *rn* have a valid nodeId certificate
  - The closest nodeId to the key is the middle one
  - The nodeIds satisfy the definition of a neighbor set
  - The average numerical distance, μ_{rn}, between consecutive nodeIds in *rn* satisfies: $\mu_{rn} < \mu_p * \gamma$ (μ_p is the average distance in the local set)
- Result
  - Test returns negative if *rn* satisfies both conditions, otherwise positive
  - False positives: correct set is incorrect with probability α
  - False negatives: incorrect set is correct with probability β
    - γ controls the tradeoff between α and β
    - Increasing γ decreases α but also increases β

M. Castro, P. Druschel, A. Ganesh, A. Rowstron, D. Wallach: Security for structured peer-to-peer overlay networks; OSDI, December 2002.

4. PSH Incentive Mechanism

Design, Evaluation

T. Bocek, Y. El-khatib, F. Victora Hecht, D. Hausheer, B. Stiller: *CompactPSH: An Efficient Transitive TFT Incentive Scheme for Peer-to-Peer Networks*; IEEE LCN 2009.

4. Private Shared History (PSH) Incentive Mechanism

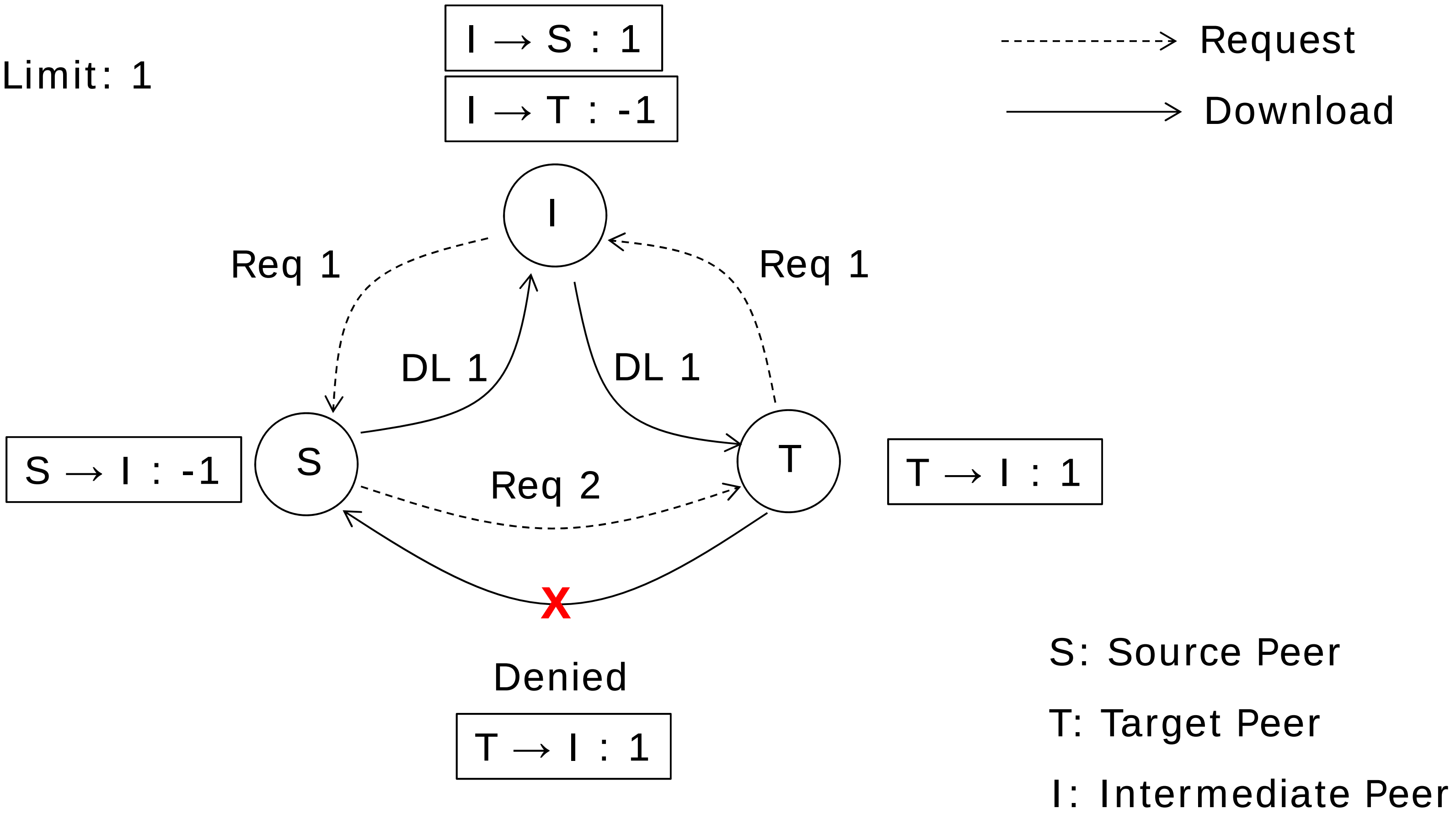
- Motivation

Properties	Shared histories	Tit-for-Tat	Virtual currency
Support asymmetry	+	-	+
Collusion resistance	-	+	+/-
Scalability	+/-	+	+/-

- Main idea of PSH: Combine benefits of private history (TFT) and shared histories
  - Support of asymmetric interest
  - Resistant against collusion
  - Highly scalable

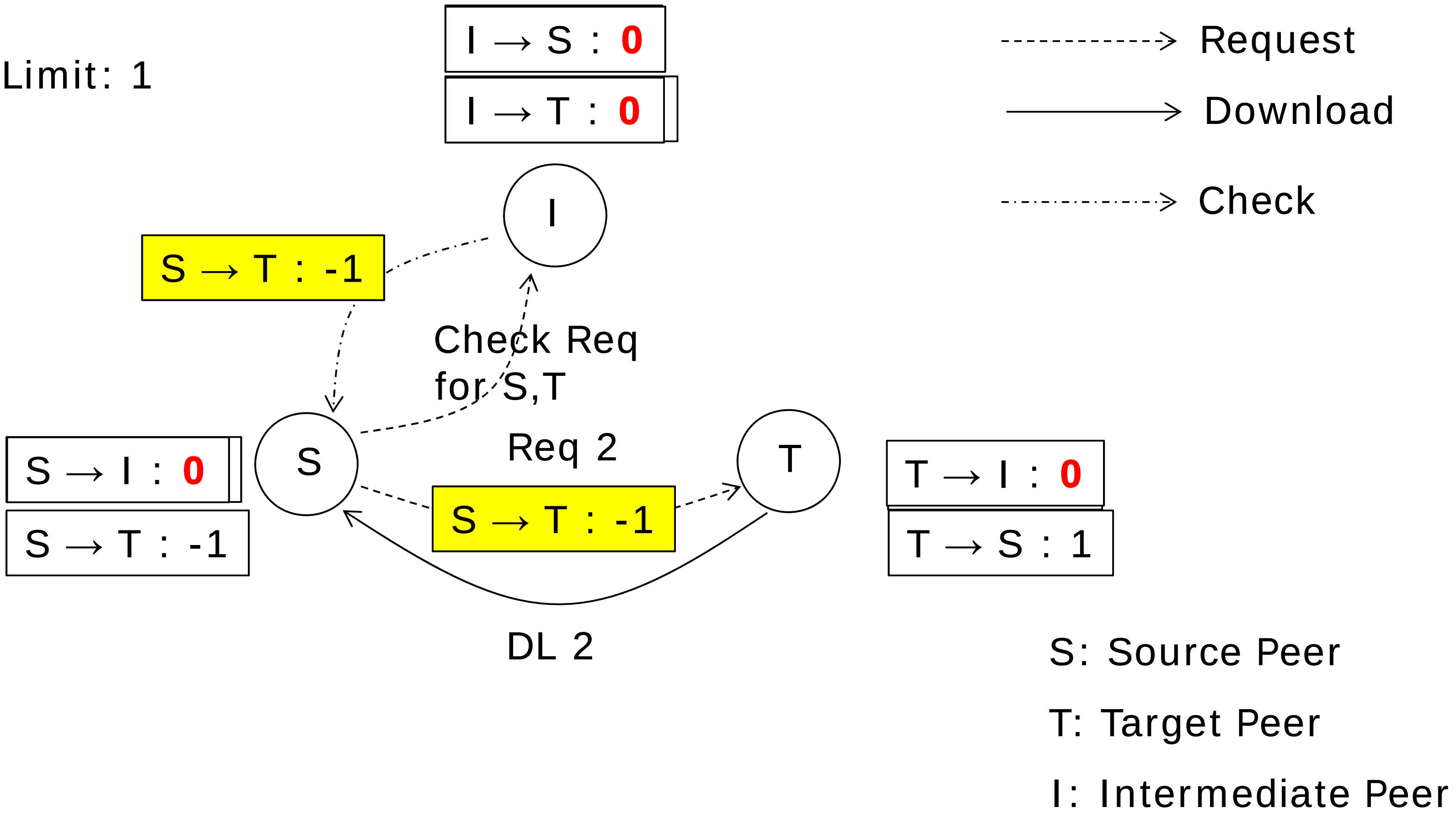
4.1. Support Asymmetry: Find Transitive Path using Shared History

Credit Limit: 1



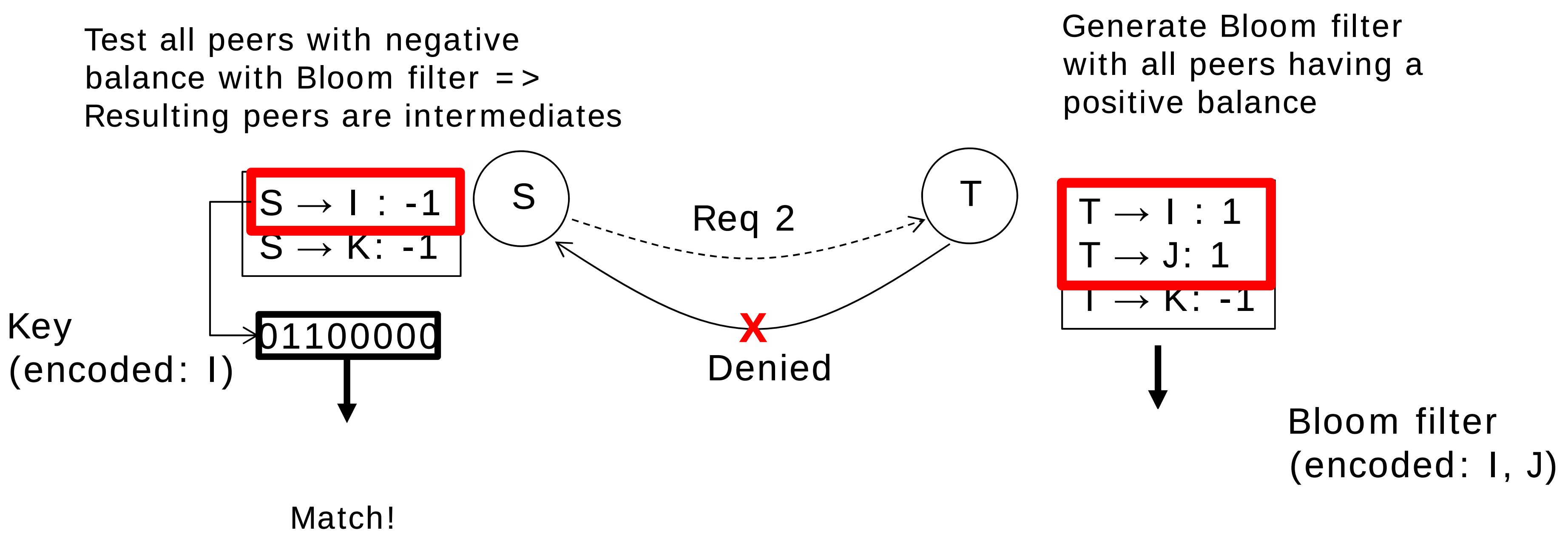
4.1. Collusion Resistance: Verify Information using Private History

Credit Limit: 1



4.1. Scalability: Reduce Overhead of Finding Intermediate Peers

- CompactPSH: Send history as a Bloom filter (space-efficient)


Test all peers with negative balance with Bloom filter => Resulting peers are intermediates
S → I : -1
S → K: -1
Key (encoded: I)
01100000
Match!
S
Req 2
T
X
Denied
Generate Bloom filter with all peers having a positive balance
T → I : 1
T → J: 1
T → K: -1
Bloom filter (encoded: I, J)

4.2. CompactPSH - Evaluation

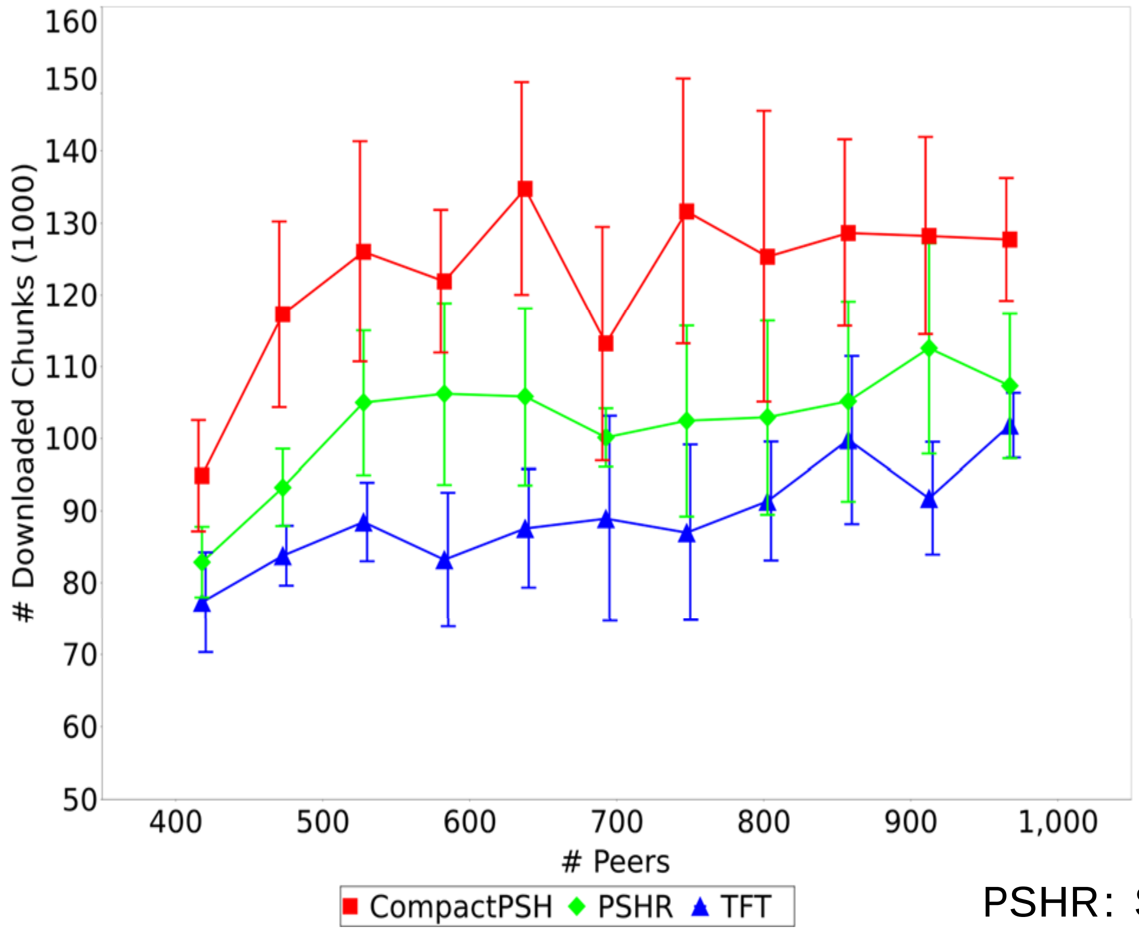
- CompactPSH implemented as a prototype

- Evaluated with input data from The Pirate Bay
  - File popularity / file sizes
  - Scaled down for the experiments

- Distributed experiments on EmanicsLab (~1000 peers on 11 hosts)
  - Parameters: Chunk size (2000 Bytes), credit (8000 Bytes)
  - 5 runs, one run 30 min, 3 downloads per peer according to file popularity, averaged results

http://www.emanicslab.org/ [F. Hecht et al: The Pirate Bay 2008-12 Dataset]

4.2. CompactPSH – Exploitation of Asymmetric Interest

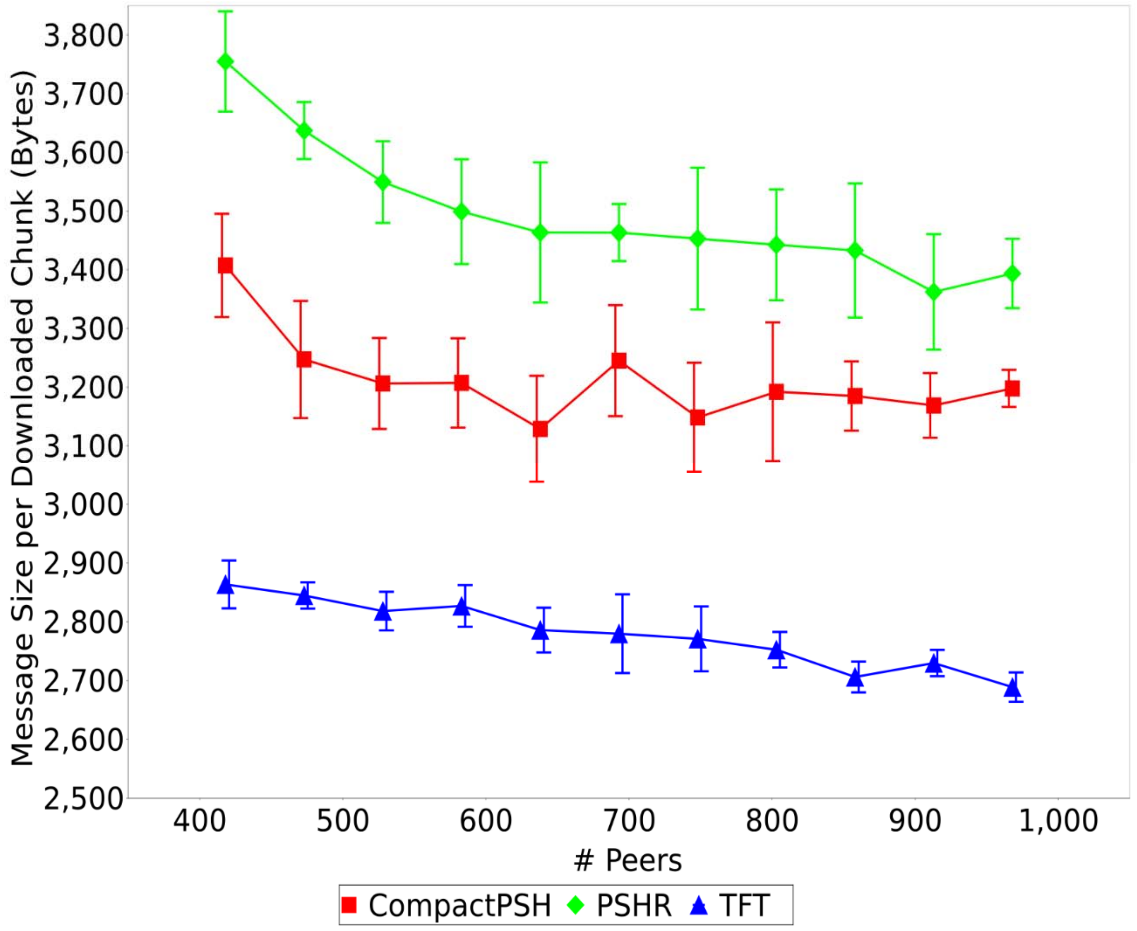


The more peers, the more chunks are exchanged

CompactPSH downloads ~25% more chunks than TFT

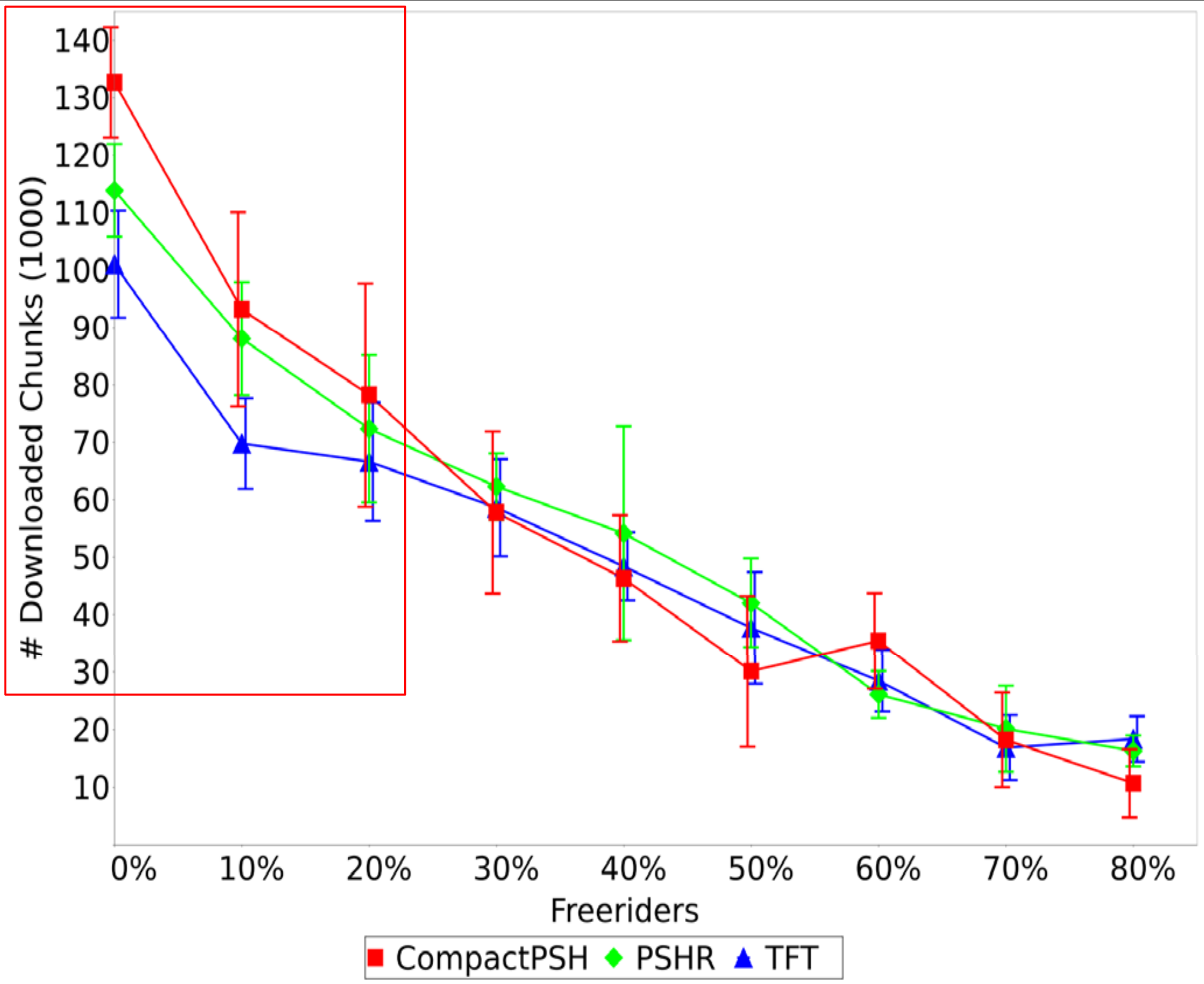
PSHR: Send sample of history

4.2. CompactPSH – Overhead



CompactPSH has lower overhead than PSHR due to less retries

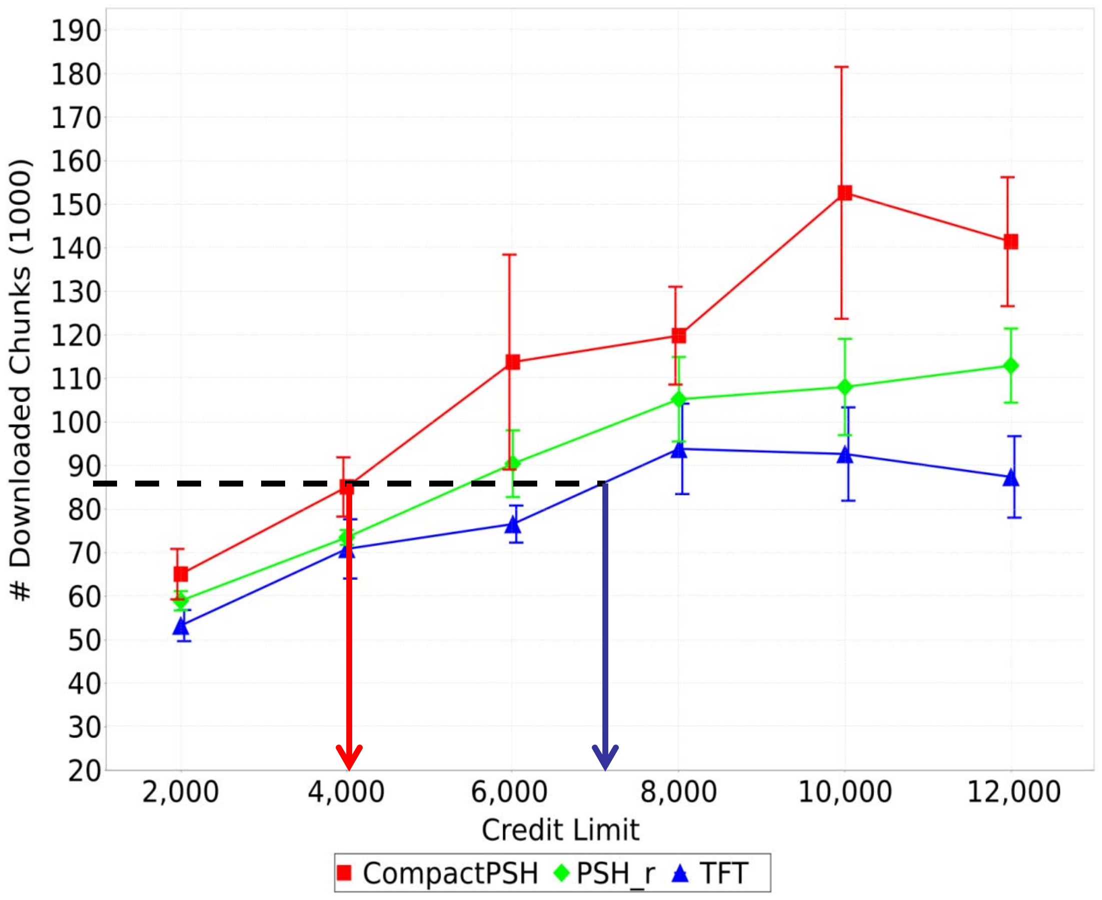
4.2. CompactPSH – Robustness against Freeriders



The more freeriders, the less chunks are exchanged

Up to 20% freeriders CompactPSH achieves highest performance, afterwards no clear winner

4.2. CompactPSH – Credit Limit



More chunks exchanged with increasing credit limit

For same number of exchanged chunks, credit limit needs to be higher in TFT than CompactPSH

=> white-washing more expensive in CompactPSH, if identity creation not free