

# Enabling Collaborative Self-healing Software Systems (5-24218)



Angelos D. Keromytis (PI)  
 Gail Kaiser, Salvatore J. Stolfo (co-PIs)  
 Kangkook Jee, Stelios Sidiroglou (GRAs)

<http://appcomm.cs.columbia.edu>

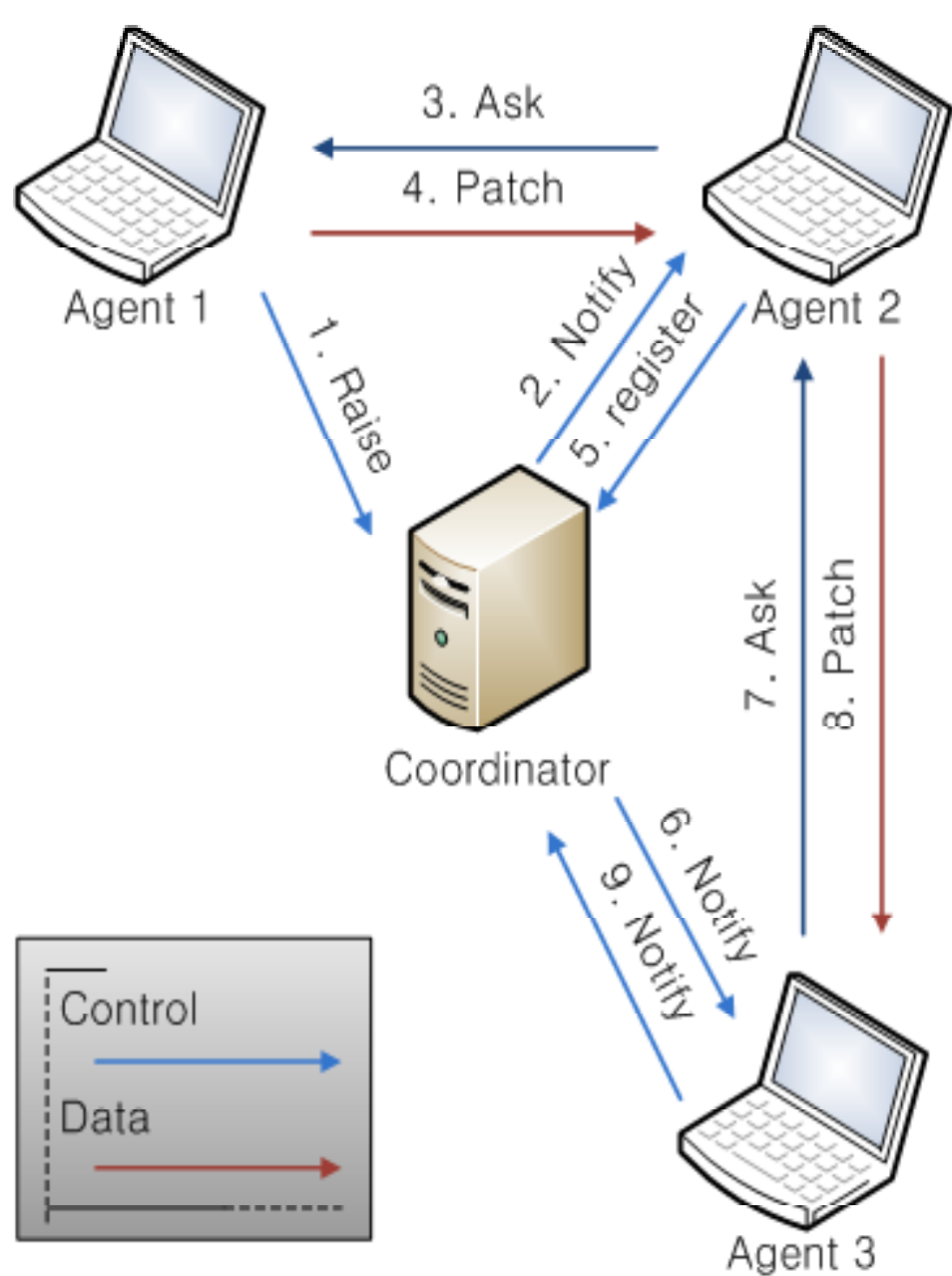
## Abstract

- The propensity for widespread destruction has made software monocultures synonymous with “bad idea” in the software vernacular.
- We attempt to redefine the term by exploiting the homogeneity and scale that define large software monocultures to improve overall security and reliability.
- We introduce and explore the concept of Application Communities: collections of large numbers of independent instances of the same application.
- Members of an application community share the burden of monitoring for flaws and attacks, and notify the rest of the community when such are detected.

## Approach

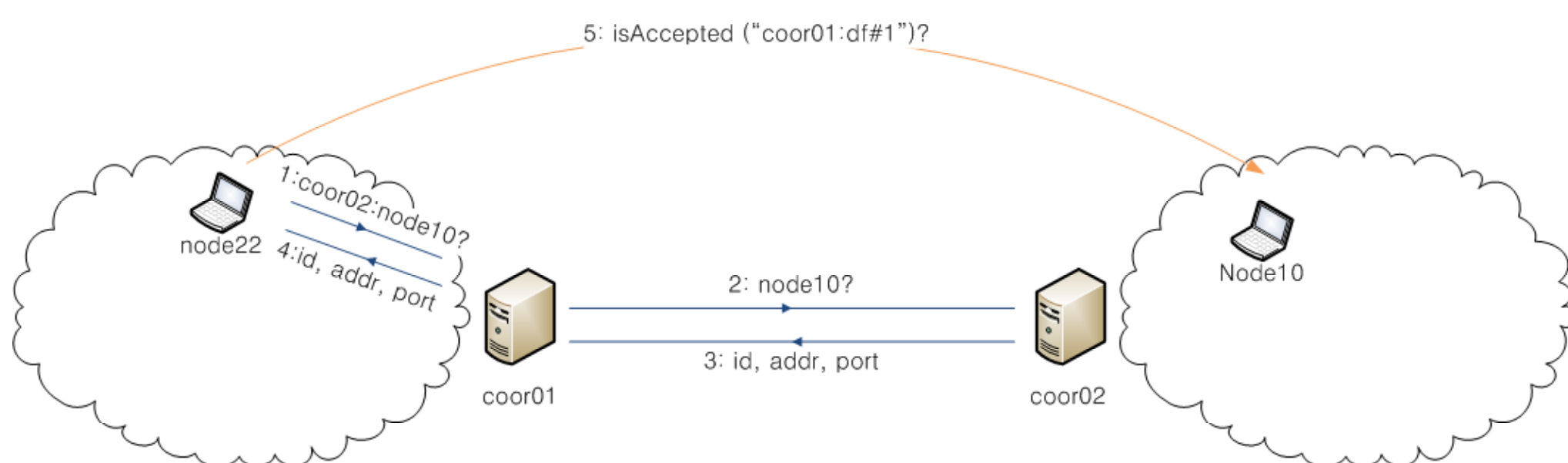
- Vulnerability/Alert dissemination using P2P (Peer-to-Peer) framework
- Nodes can individually verify the validity of alerts/vulnerabilities
- To enable an AC to scale to thousands of nodes, we introduce a efficient hierarchical dissemination infrastructure.
- Support for a variety of host-based monitoring tools
- Automatic selection of transport mechanism (direct, bit-torrent, etc.,) based on size of alert/vulnerability.

## Information sharing in P2P (Peer-to-Peer) framework



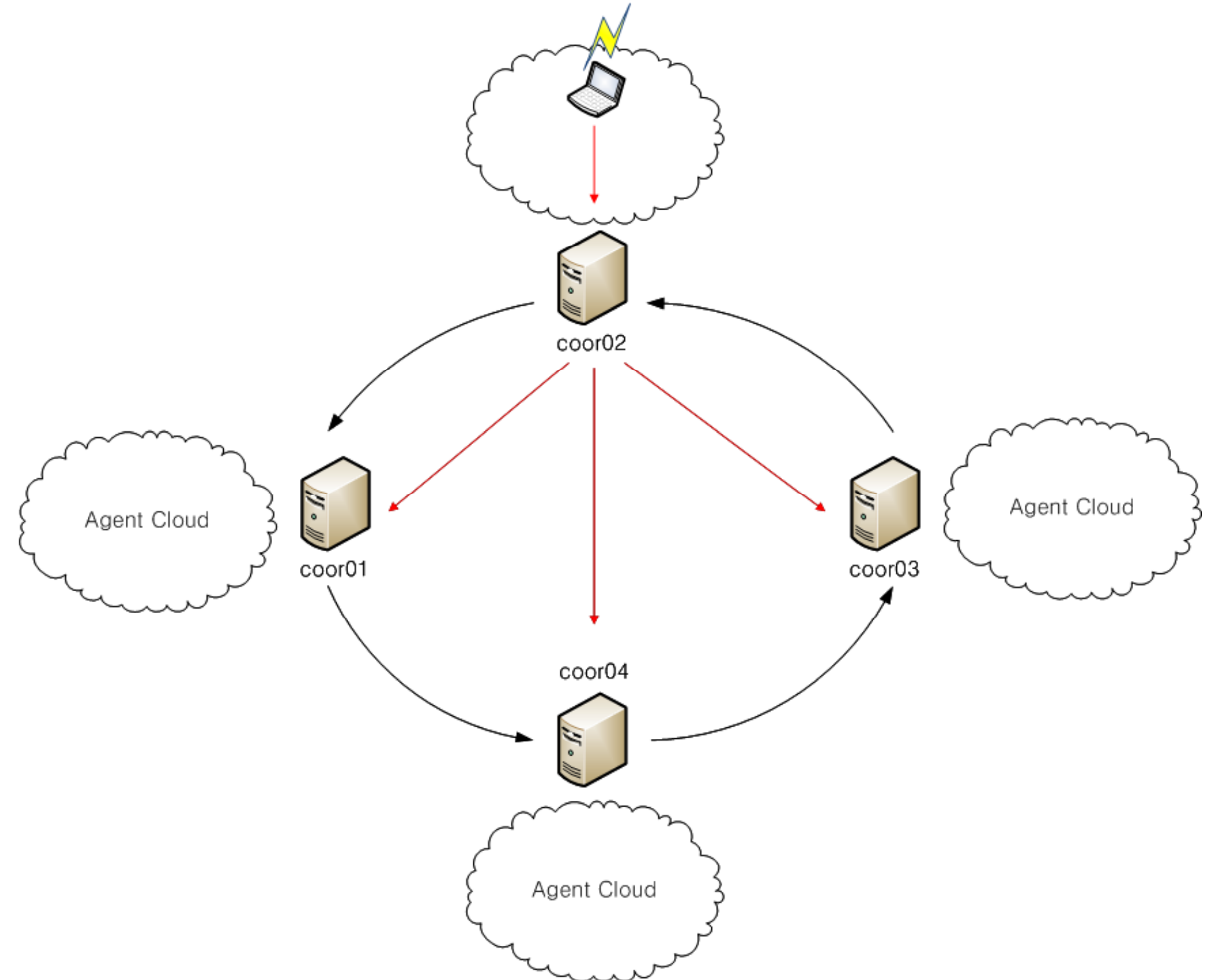
- Example protocol exchange: Publish/subscribe mechanism

## Patch certification procedure



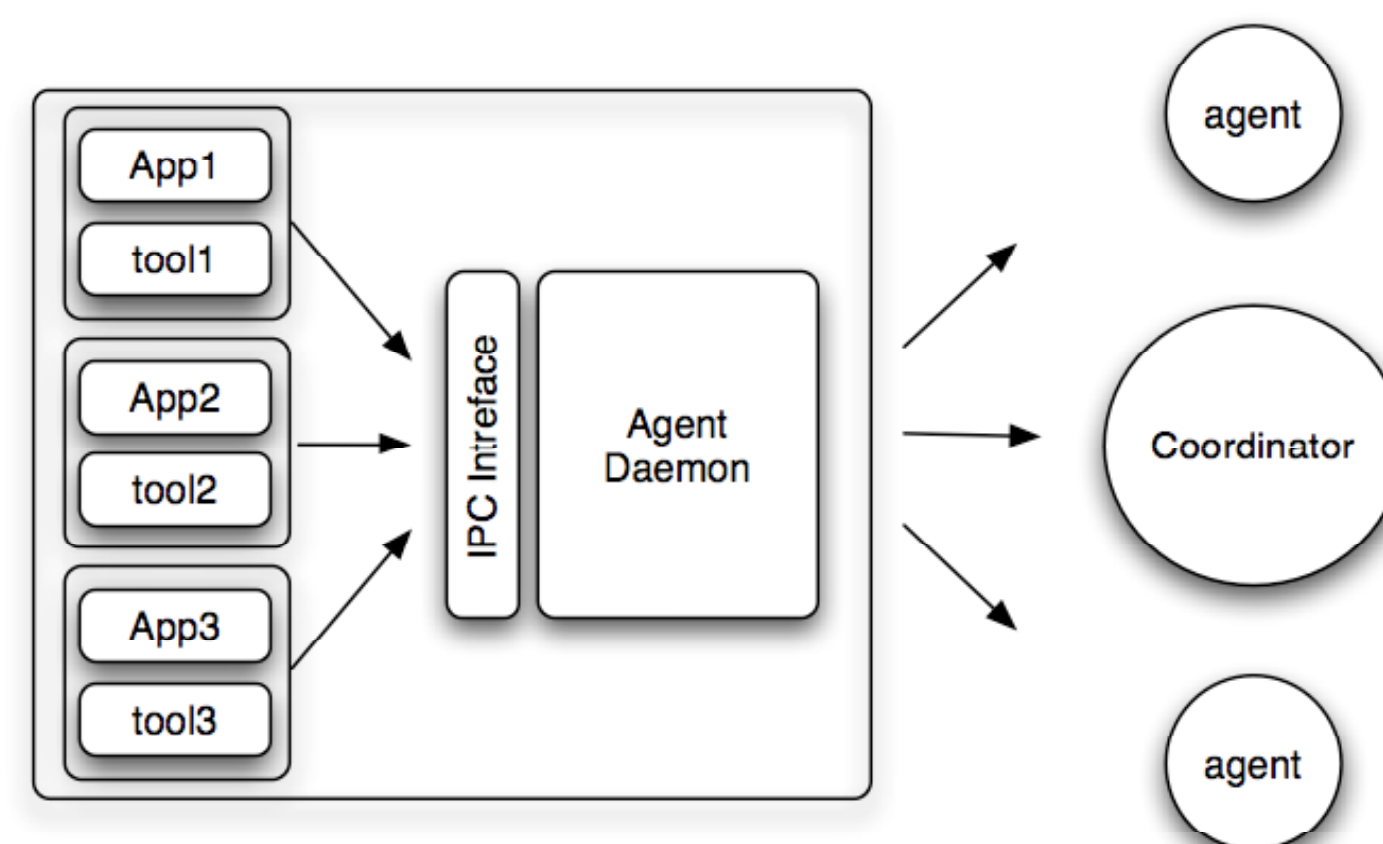
- AC supports the following certification mechanisms:
- Delegation of certification to trusted entity
- Peer rating
- Voting for the credibility

## Distributed Coordinator network



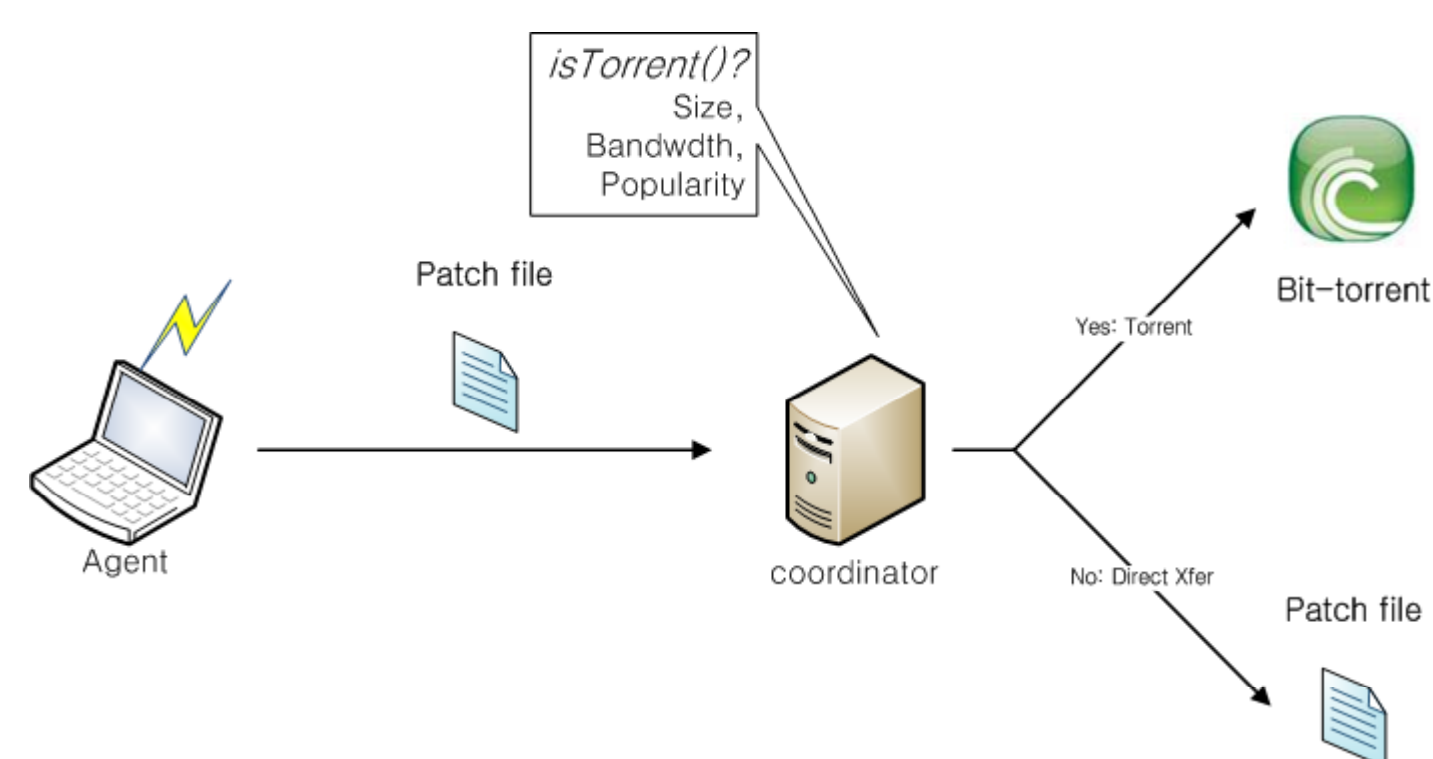
- To scale to tens of thousands of agents, AC relies on a hierarchical network structure

## Agent architecture



- The agent is embedded to each host and provides generic API which can be used by various vulnerability detection tools

## Support for Bit-Torrent Transfer



- Information can be shared by either direct P2P transfer or Bit-torrent transfer
- Decision based on size of the information, number of agents interested (popularity) and the bandwidth of underlying network