Peer-to-peer Architecture for Collaborative Intrusion and Malware Detection on a Large Scale

Mirco Marchetti, **Michele Messori** and Michele Colajanni
WebLab, University of Modena and Reggio Emilia, Italy

Pisa - 9 September 2009
Information Security Conference 2009

Defense scenarios

- Complex information systems
  - Heterogenous networks (wired-wireless)
  - Networks consisting of multiple segments
- Cooperation among multiple organizations

=> **Centralized defensive solutions do not work**

=> Our focus: distributed architectures where cooperation is carried out through p2p schemes

Goals

Building *high-level activity reports* from low-level alerts (related to one peer) about:

- Malware behavior
- Malware diffusion
- Network-based attacks
- Diffusion of intrusions
- Identification of suspicious IP addresses
- Identification of the servers from which the malware is downloaded
- ...

Distributed IDS model
Main components

(Heterogeneous) **analyzers**:

- Watch of host activities
- Sniffing of network traffic
- Interaction with (probable) attacker
- Stateful analysis of gathered data

Collectors:

- Collection of low-level alerts from the analyzers

Collaborator module:

- Aggregation and correlation of the alerts

Distributed IDS
Existing architectures

Centralized processing:

- Distributed analyzers
- Centralized aggregation

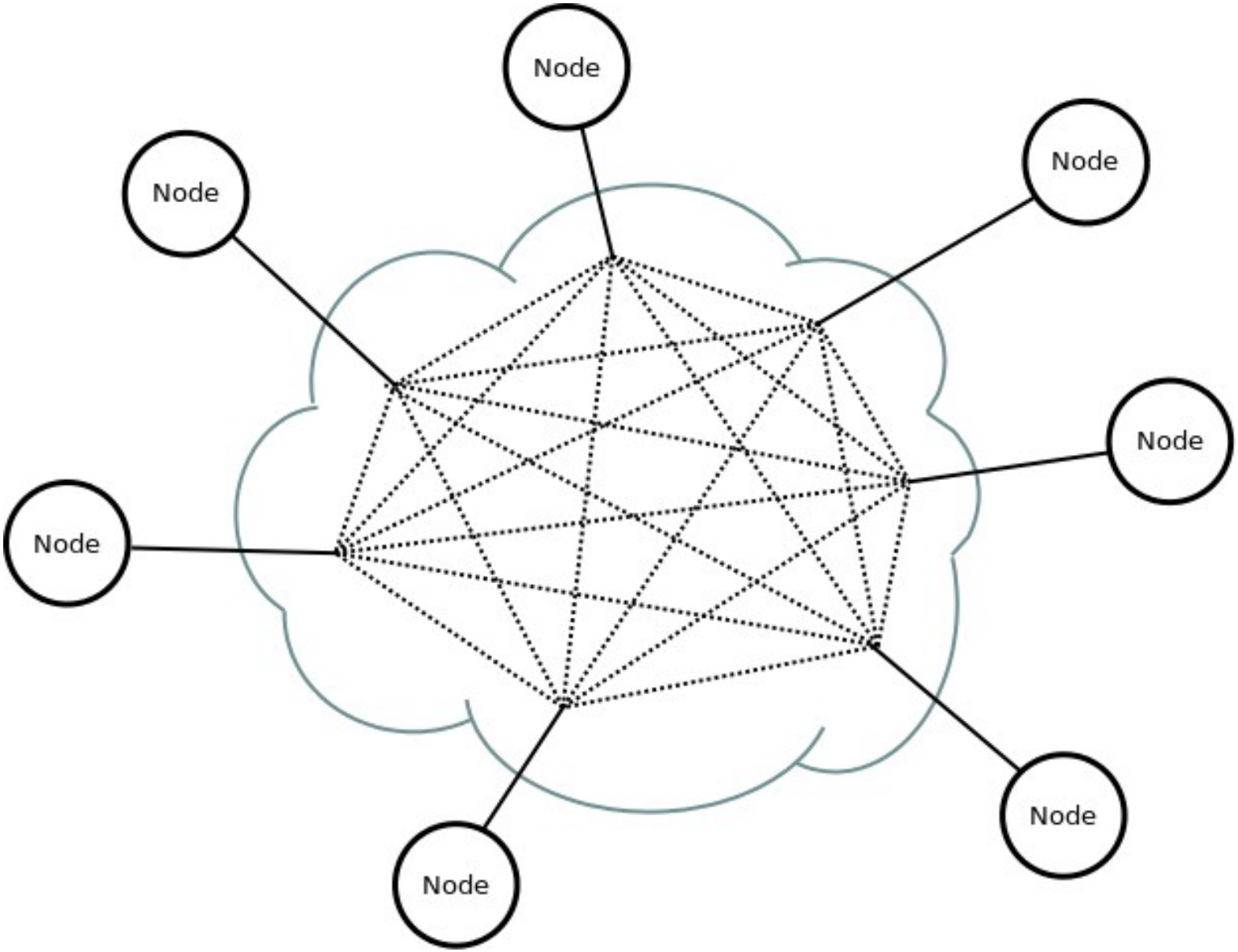
Hierarchical processing:

- Distributed analyzers
- Multi-level hierarchical aggregation

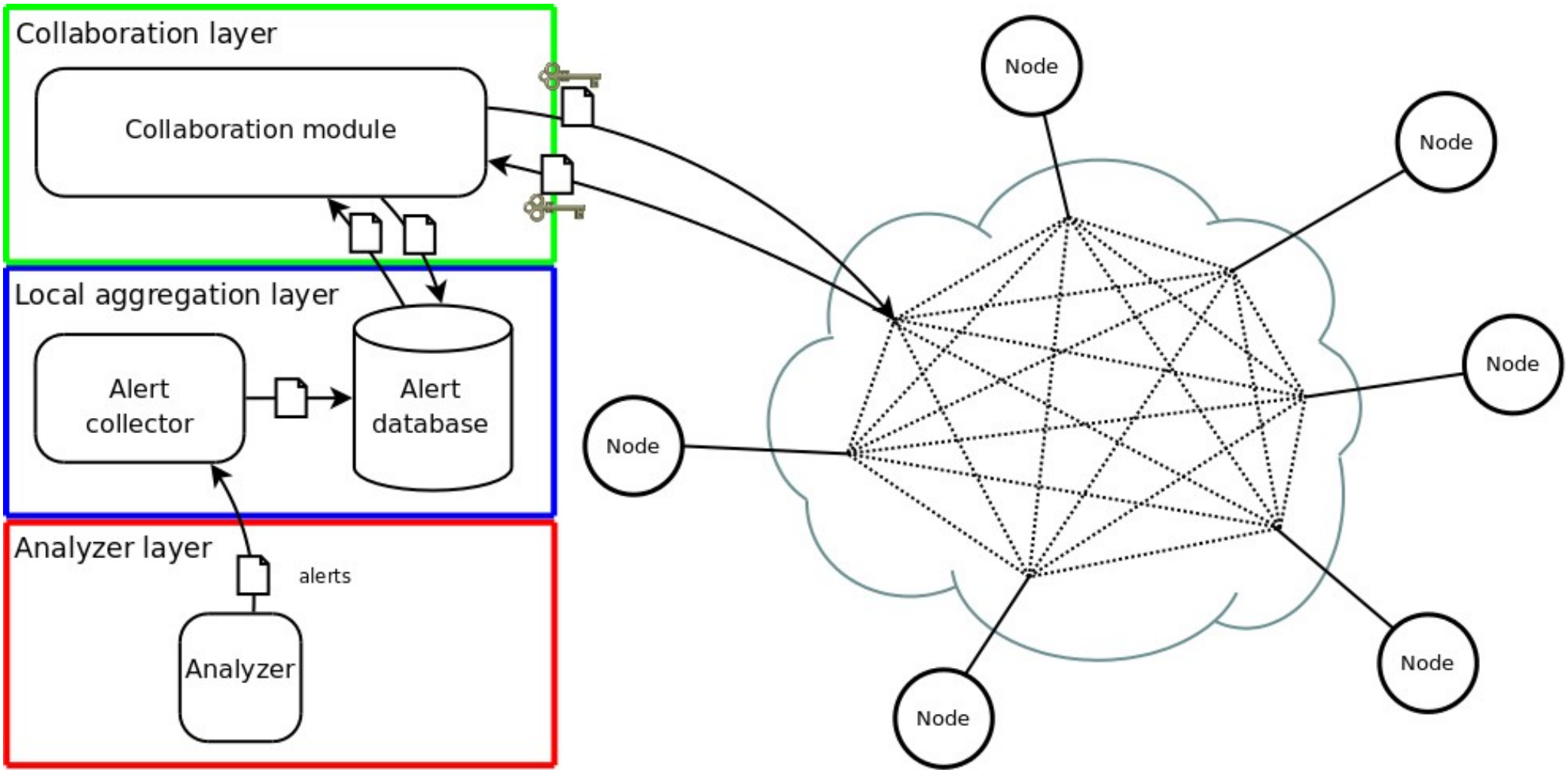
Common drawbacks:

- Single point(s) of failure
- Load unbalance
- Poor or fair scalability

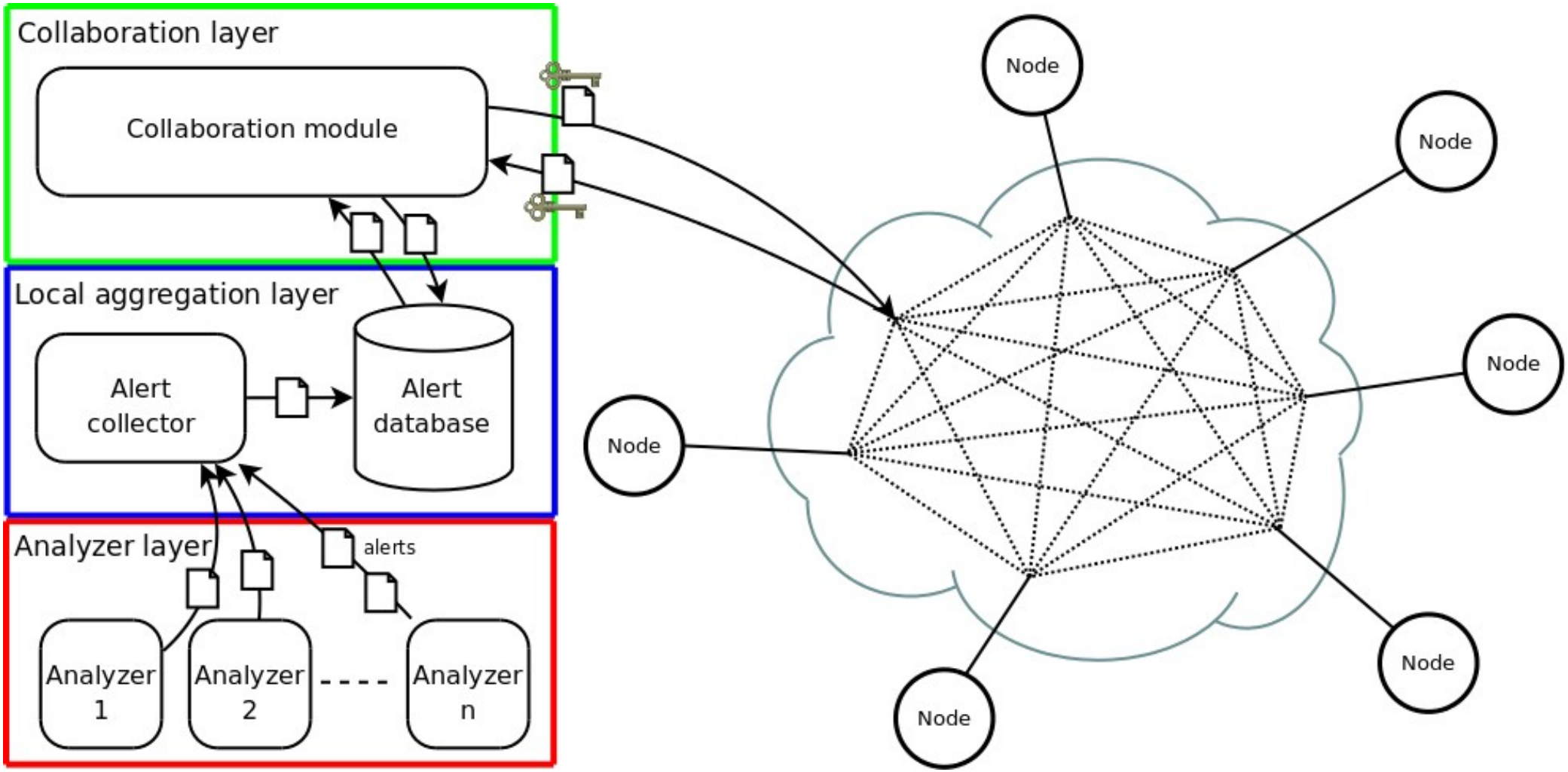
Distributed IDS architecture based on P2P



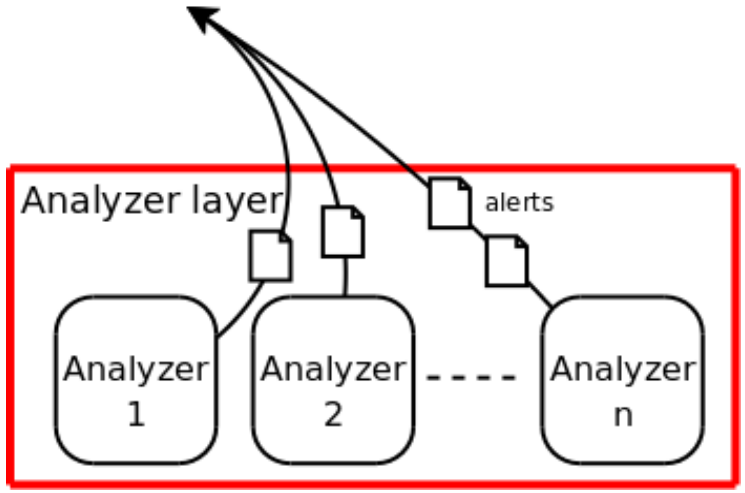
Structure of a single node



Structure of a single node



Analyzer layer



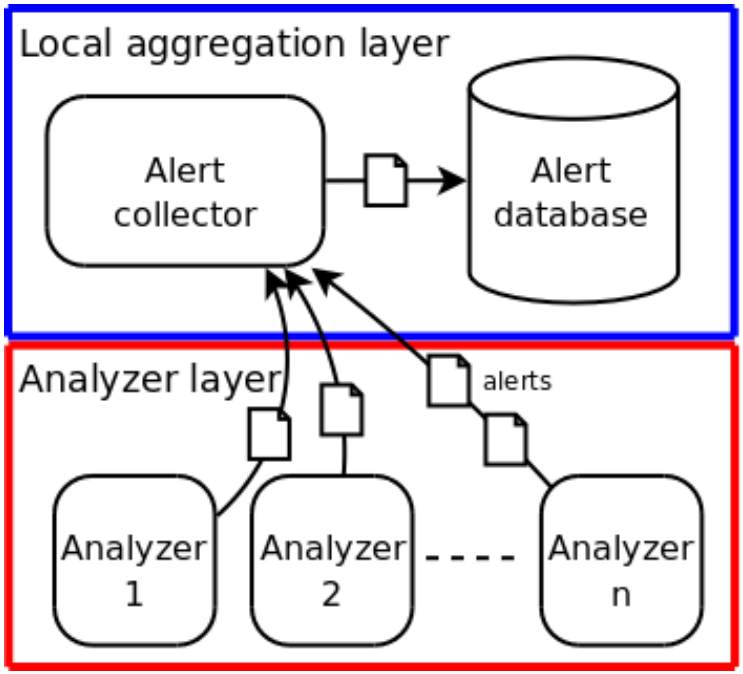
Main features:

- Intrusion detection
- Malware collection

Analyzer types:

- Host IDS
- Network IDS
- Honeypot
- Sensor Manager

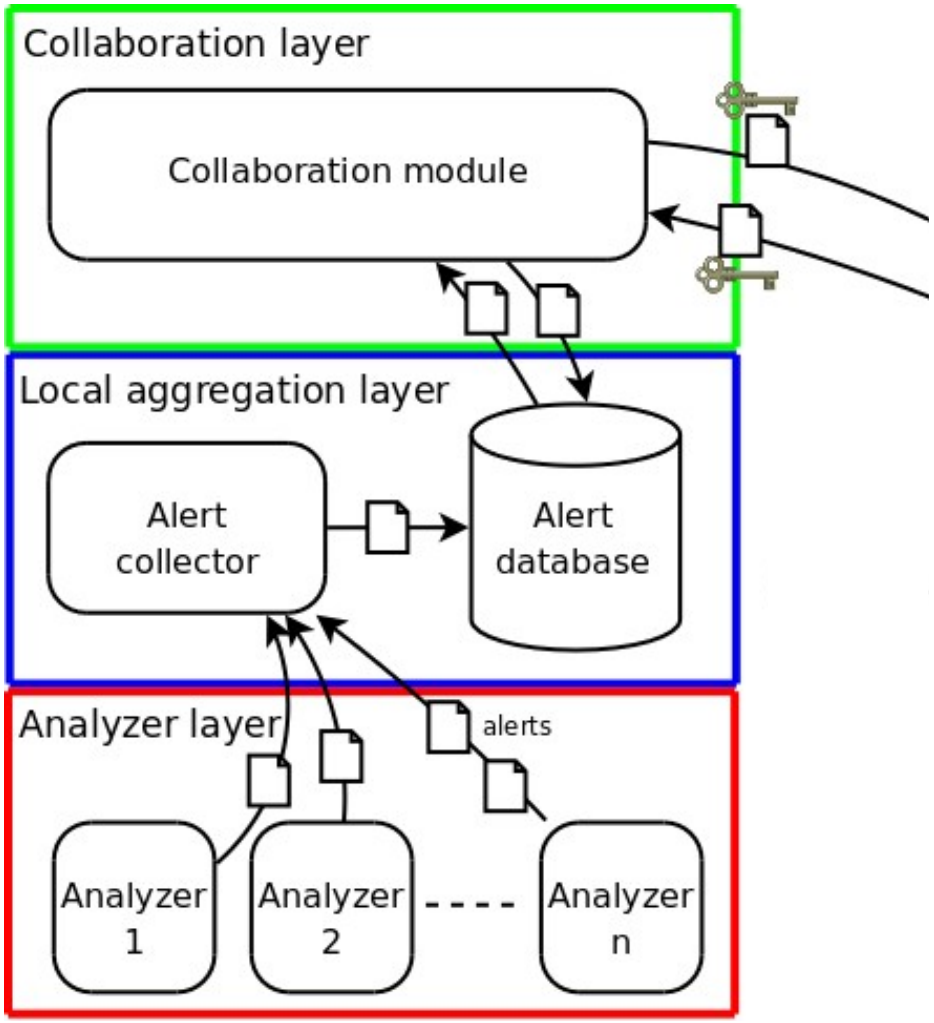
Local aggregation layer



Main features:

- Pre-processing of all collected data for homogeneous storage
- Classification and storage of all alerts in the local *alert database*

Collaboration layer

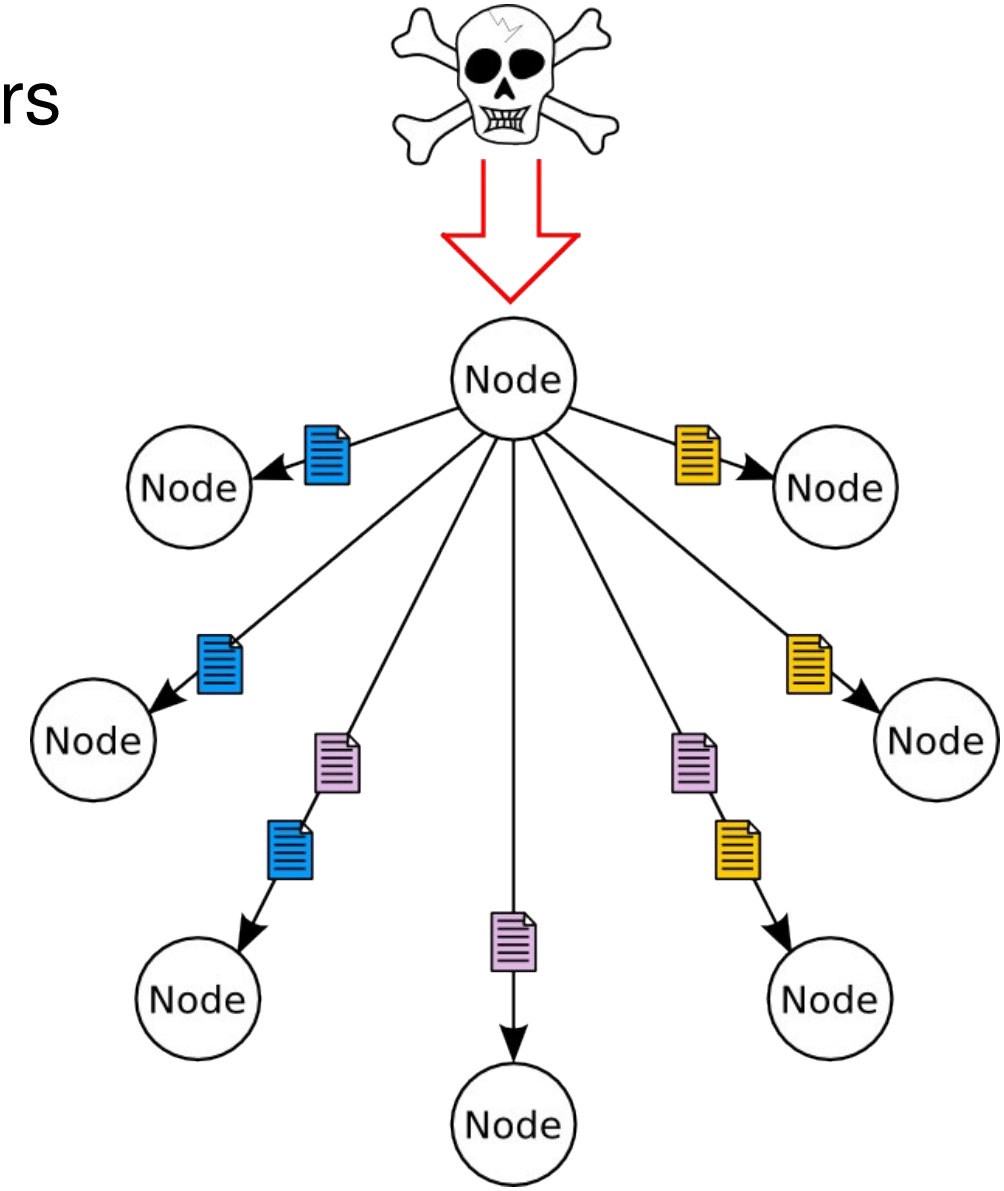


Main features:

- Submits collected alerts to the DHT-based overlay network
- Manages a portion of the hash space
- Disseminates the analysis results

Operations: *Early detection*

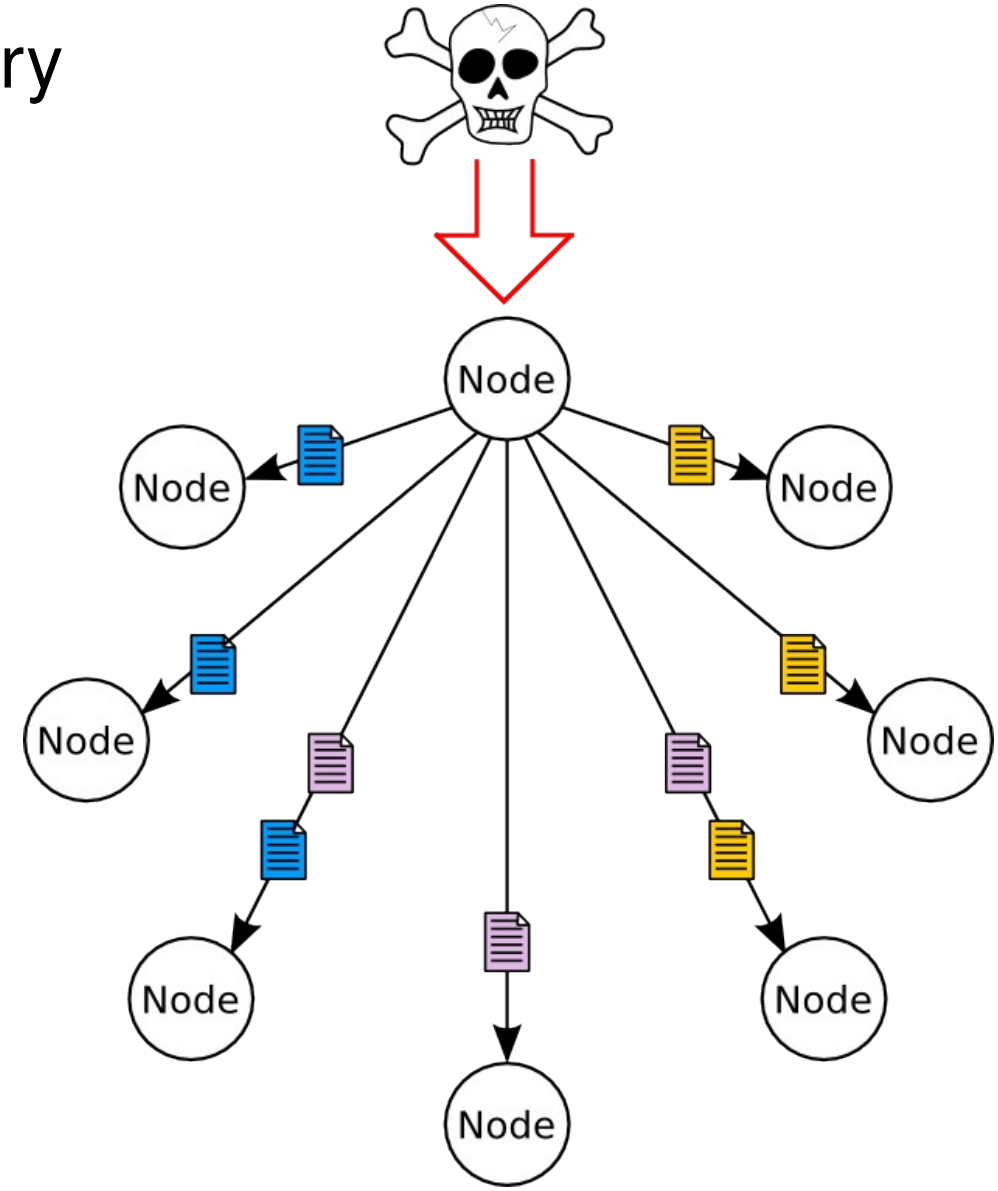
1) Collecting alerts from analyzers
2) Extrapolating information
3) Keys selection
4) Messages distribution based on hash key (, ,)
5) Real-time creation of the replicas
6) Automatic management of the replicas



Operations: *Key selection*

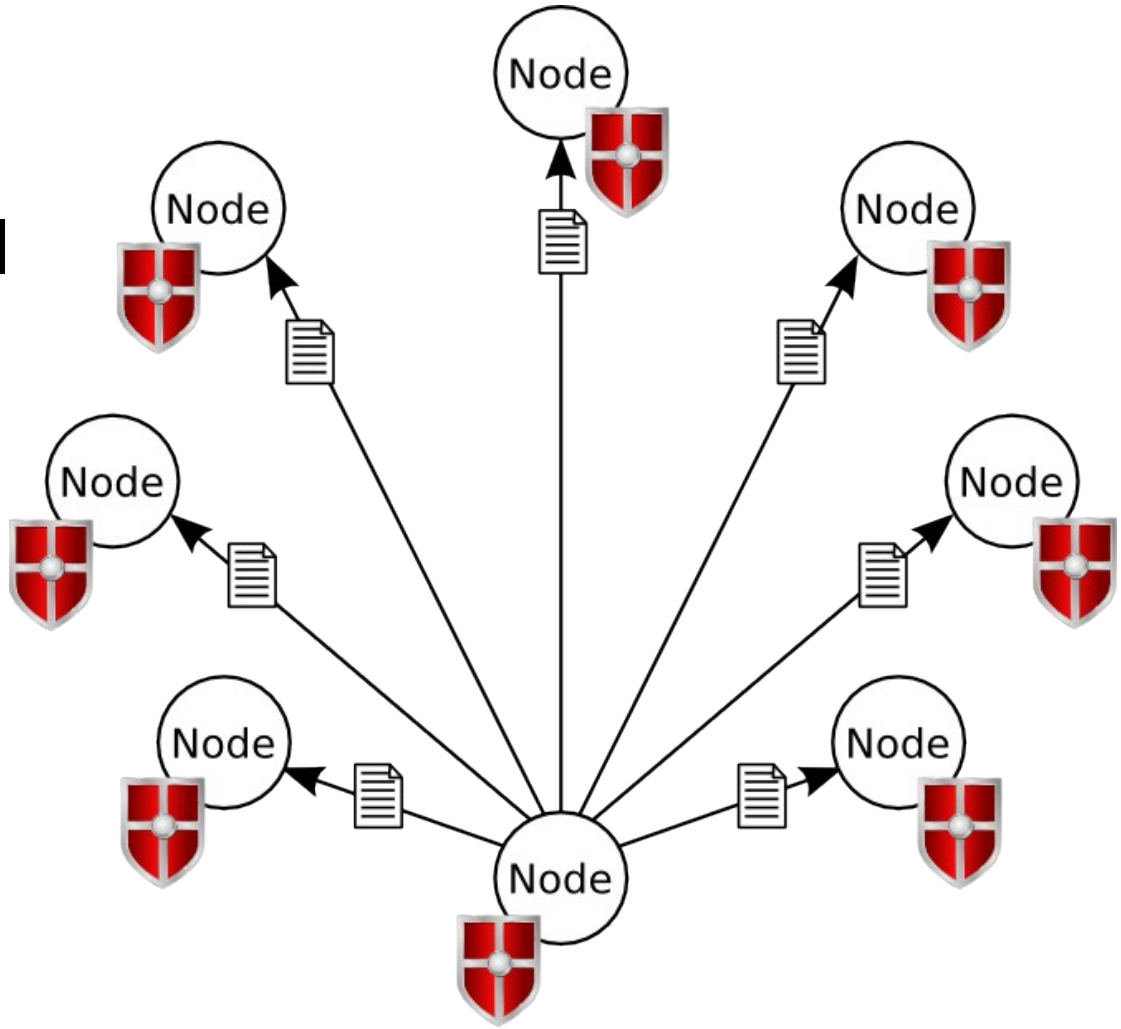
One message submitted for every interesting field, such as:

- Malware's binary code ( )
- IP address of the server from which the malware has been downloaded ( )
- IDS signature ID
- IP address of the attacker ( )



Operations: *Early warning*

7) Processing received messages
8) Broadcast communication of analysis results (high level *activity reports*)( )
9) Early warning threats
10) All peers are protected



Prototype
Components

Analyzers:
- OSSEC (HostIDS)
- Snort (NetworkIDS)
- Nepenthes (Honeypot)

Collector:
- Prelude (Hybrid IDS) – useful for IDMEF
- MySql (DBMS)

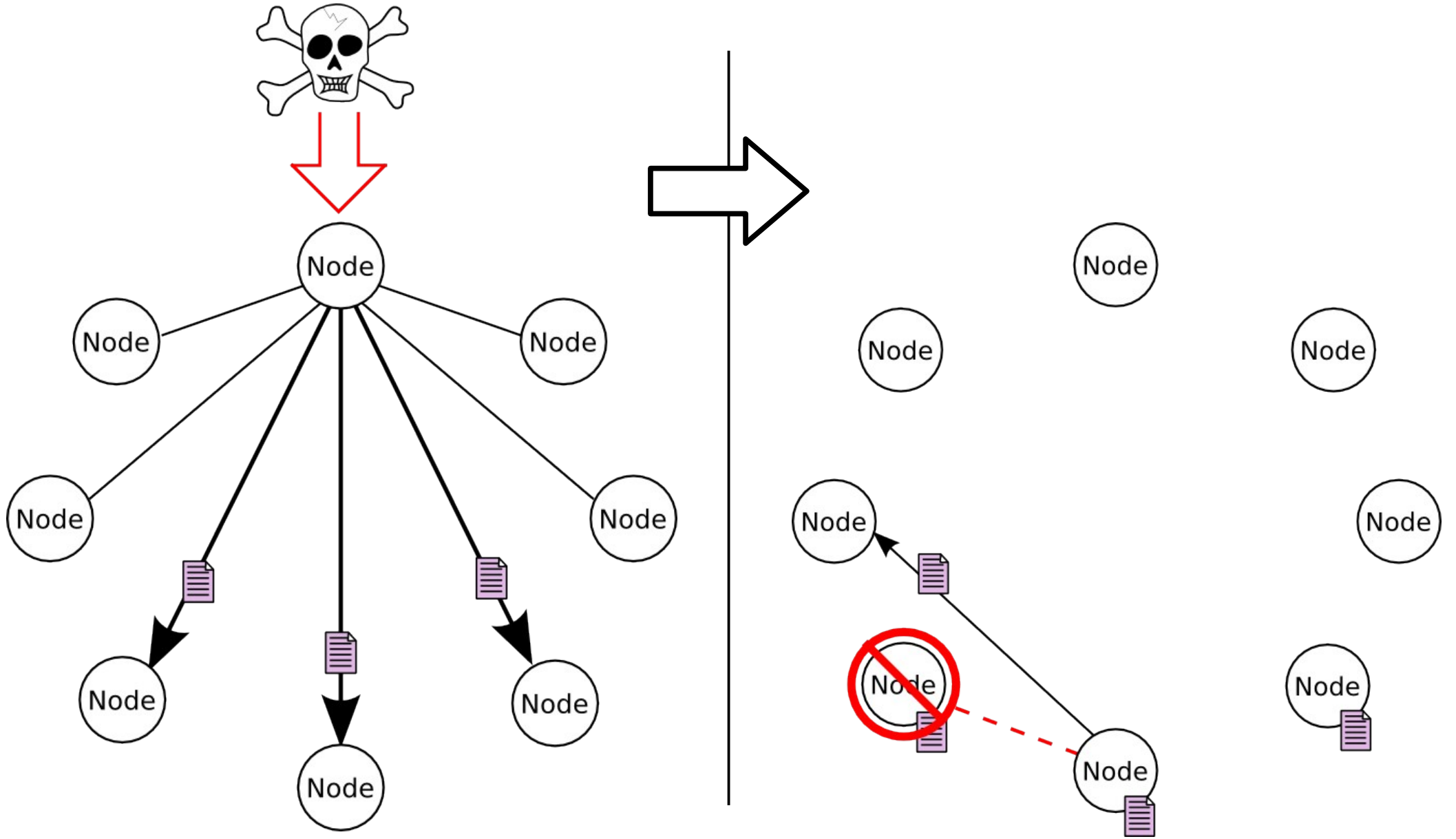
Collaboration module:
- Freepastry libraries (DHT overlay)

Prototype
Present features

- Key generation for different alert fields
- Management of malware bincode
- Anomaly detection threshold-based

- Emulation of thousands of nodes through Freepastry libraries

Fault Tolerance – *graceful degradation*



Simulation results
Fault tolerance

Concurrent faults (%)	k=4	k=5	k=6
1	0.009	0	0
2	0.16	0.003	0
3	0.735	0.019	0.001
4	2.117	0.075	0.002
5	5.022	0.219	0.015
6	9.732	0.542	0.037
7	16.64	1.186	0.081
8	25.859	2.172	0.159
9	36.682	3.774	0.315
10	48.685	5.904	0.529

Message loss probability for a network of 10,000 nodes and for different replica factor k. Values in percentage (%)

Conclusions and future work

Conclusions

- Load balancing (as in Indra)
- Graceful degradation (as in Domino)
- **Interoperability with heterogeneous sensors/analyzers**
- **Malware payload management**
- Prototype

Future works

- To enhance anomaly detection (e.g., new aggregation algorithms)
- Malware analysis (through internal engine or external services)