Unità di Ricerca
Modena e Reggio Emilia

Progetto PRIN
“Autonomic Security”

Il progetto Autonomic Security

- **Obiettivi del progetto**
  - Integrazione della “self-protection” nei sistemi geograficamente distribuiti
  - Modelli ed algoritmi per la rappresentazione ed il self-monitoring dello stato interno di un sensore
  - Modelli ed algoritmi per l'identificazione dei cambi di stato e delle anomalie
  - Strategie di protezione e di contrasto agli attacchi
    - Contesto specifico: Self-protection contro le botnet

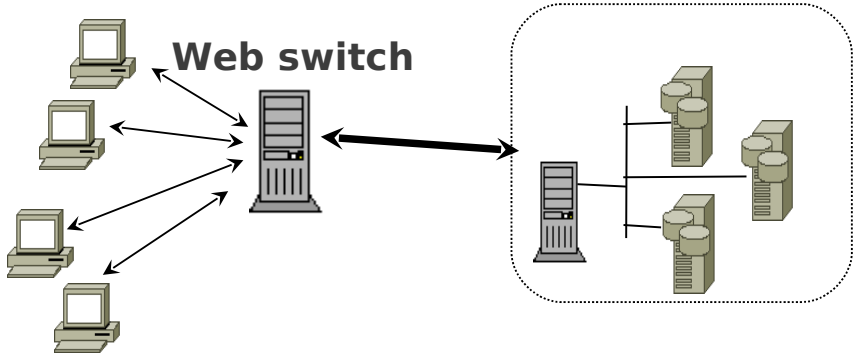
L'unità di ricerca UniMO

- **Partecipanti al progetto**
  - **Mauro Andreolini (ricercatore)**
  - Claudia Canali (ricercatrice)
  - **Sara Casolari (assegnista di ricerca)**
  - Michele Colajanni (professore ordinario)
  - Mirco Marchetti (assegnista di ricerca)
- **Competenze**
  - Gestione di sistemi distribuiti su larga scala
  - Monitoraggio di sistemi distribuiti
  - Virtualizzazione
  - Sicurezza

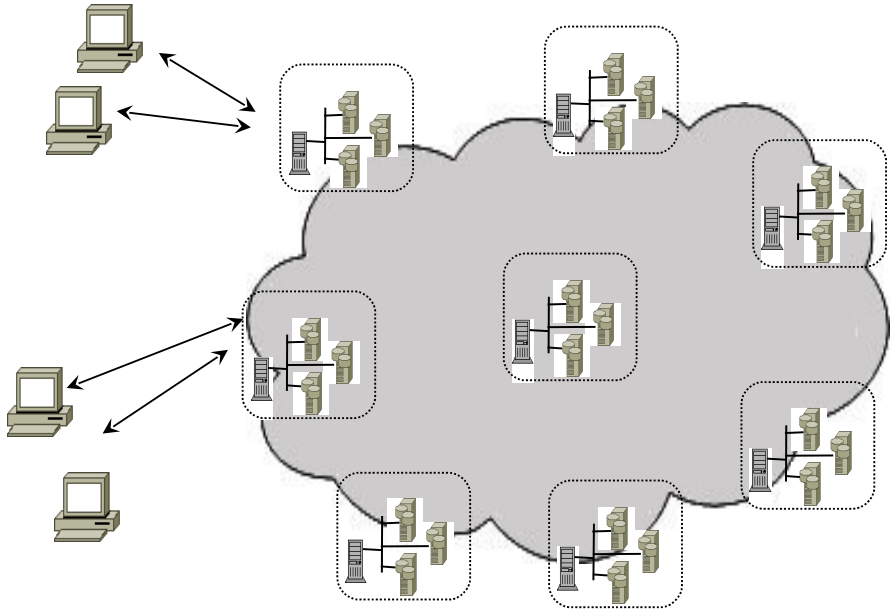
Architetture di riferimento

- **Sistemi distribuiti**

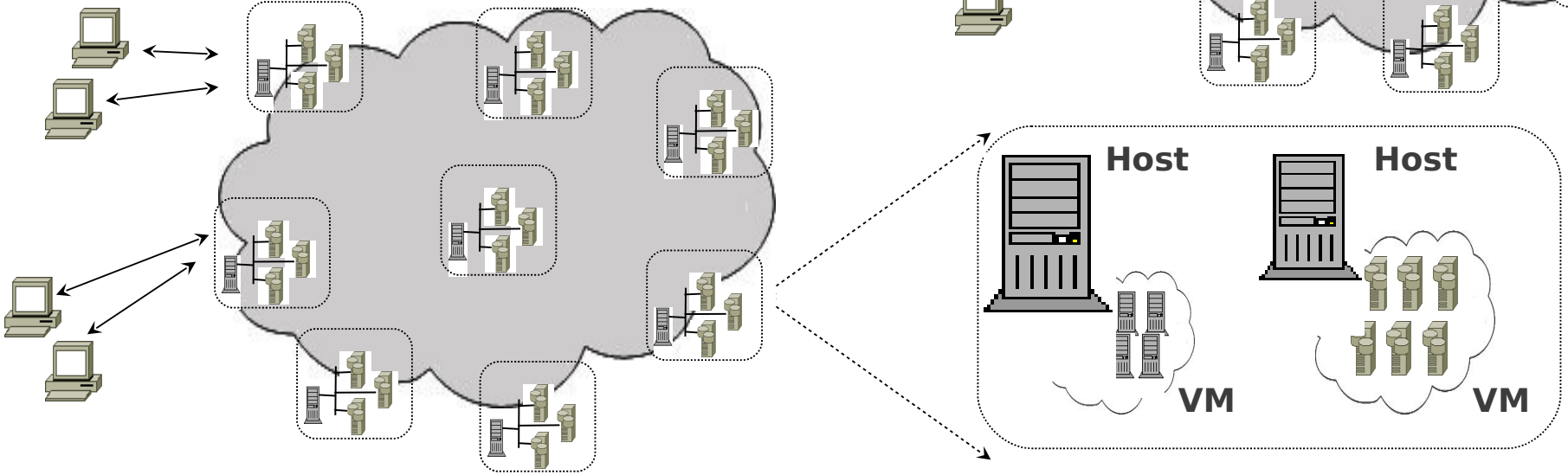
  su scala locale



  su scala geografica



- **Sistemi virtualizzati**



Competenze specifiche

- **Monitoraggio**
  - Filtraggio misure grezze
  - Trasformazione misure → **informazione**
  - Algoritmi decisionali (cambi di stato, periodicità ed anomalie)
  - Modelli predittivi (breve/lungo termine, off/online)
- **Virtualizzazione**
  - Architetture virtualizzate (Xen, KVM, OpenVZ)
  - Monitoraggio di sistemi guest e host
  - Migrazione a caldo di sistemi guest
- **Sicurezza**
  - Sviluppo di architetture ed algoritmi per l'intrusion detection parallela
  - Bilanciamento del carico in ingresso

Stato di avanzamento del progetto

1. **Algoritmi e meccanismi per la detection distribuita di minacce e anomalie**
   - Particolare enfasi sulla **early detection** (scoperta di una possibile minaccia)
   - Particolare enfasi sugli **early warnings** (comunicazione di potenziali minacce)
   - Avanzamento: 80%
2. **Strategie coordinate e distribuite per la reazione agli attacchi**
   - Protezione dei servizi
   - Contrasto alle minacce
   - Avanzamento: 20%

1. Detection distribuita: Raccolta dei dati

- **Raccolta distribuita di grandi quantità di dati legate ad un elevato numero di eventi**
  - Classico schema
    più sonde → acquisitore → singolo DB
    non scala con il numero di host fisici e di macchine virtuali
  - Il DB diventa rapidamente il collo di bottiglia dell'infrastruttura di monitoraggio
    - Utilizzazione di CPU e di memoria particolarmente critiche
  - Necessità di investigare architetture di monitoraggio alternative

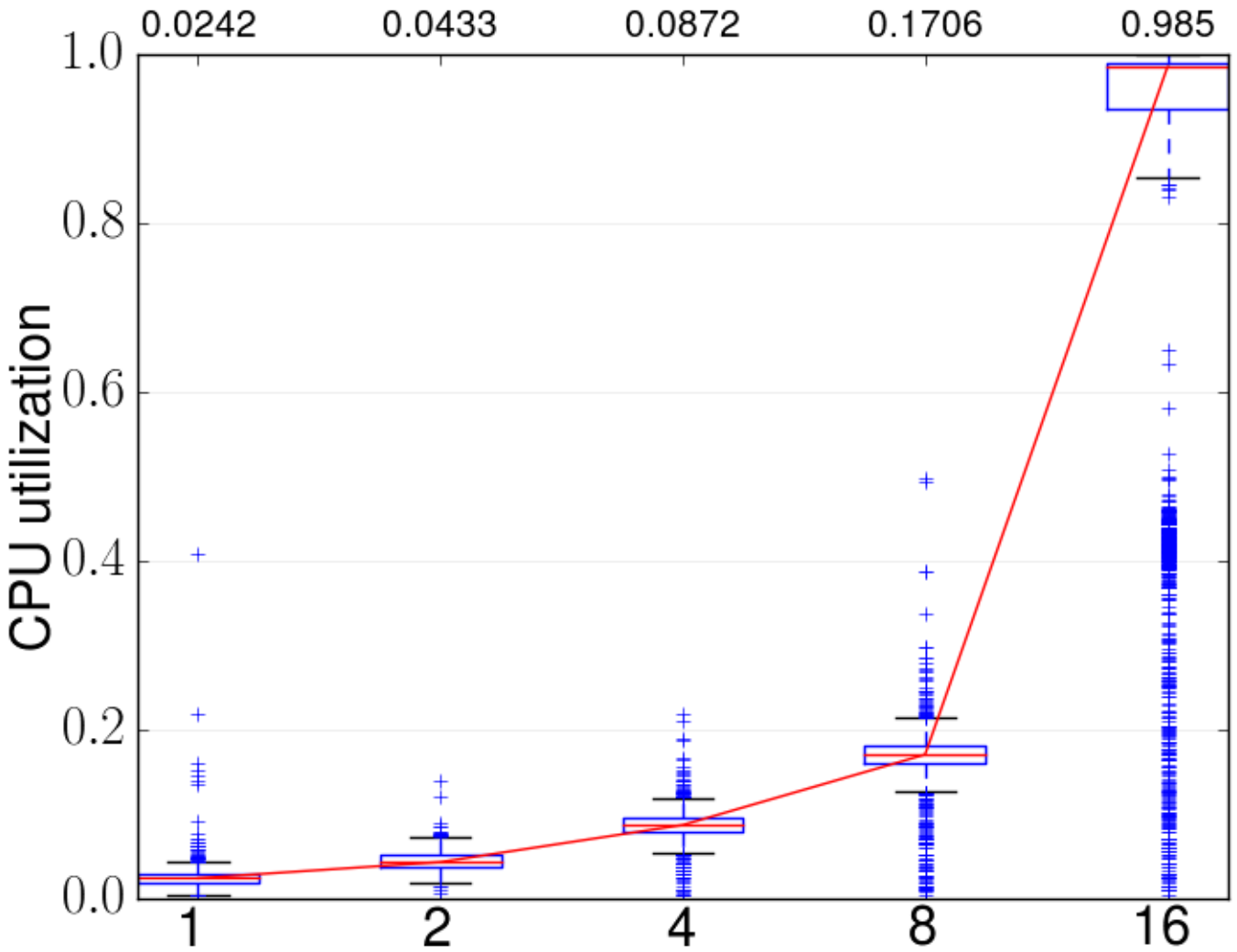
1. Detection distribuita: Raccolta dei dati

- **Raccolta distribuita di grandi quantità di dati legate ad un elevato numero di eventi**
  - Infrastruttura di sistema
    - Hardware: Emulab (20-50 nodi fisici, ognuno con 2 CPU, 2.4GHz, 2GB RAM)
    - OS: Ubuntu GNU/Linux 10.04 (Lucid Lynx)
  - Infrastruttura di monitoraggio
    - Monitor: vmstat, sar, snort
    - Monitor: Python
    - Intervallo di campionamento: 10s
  - Tecnologia di storage
    - MySQL v5.1, RAM-based (engine MEMORY)

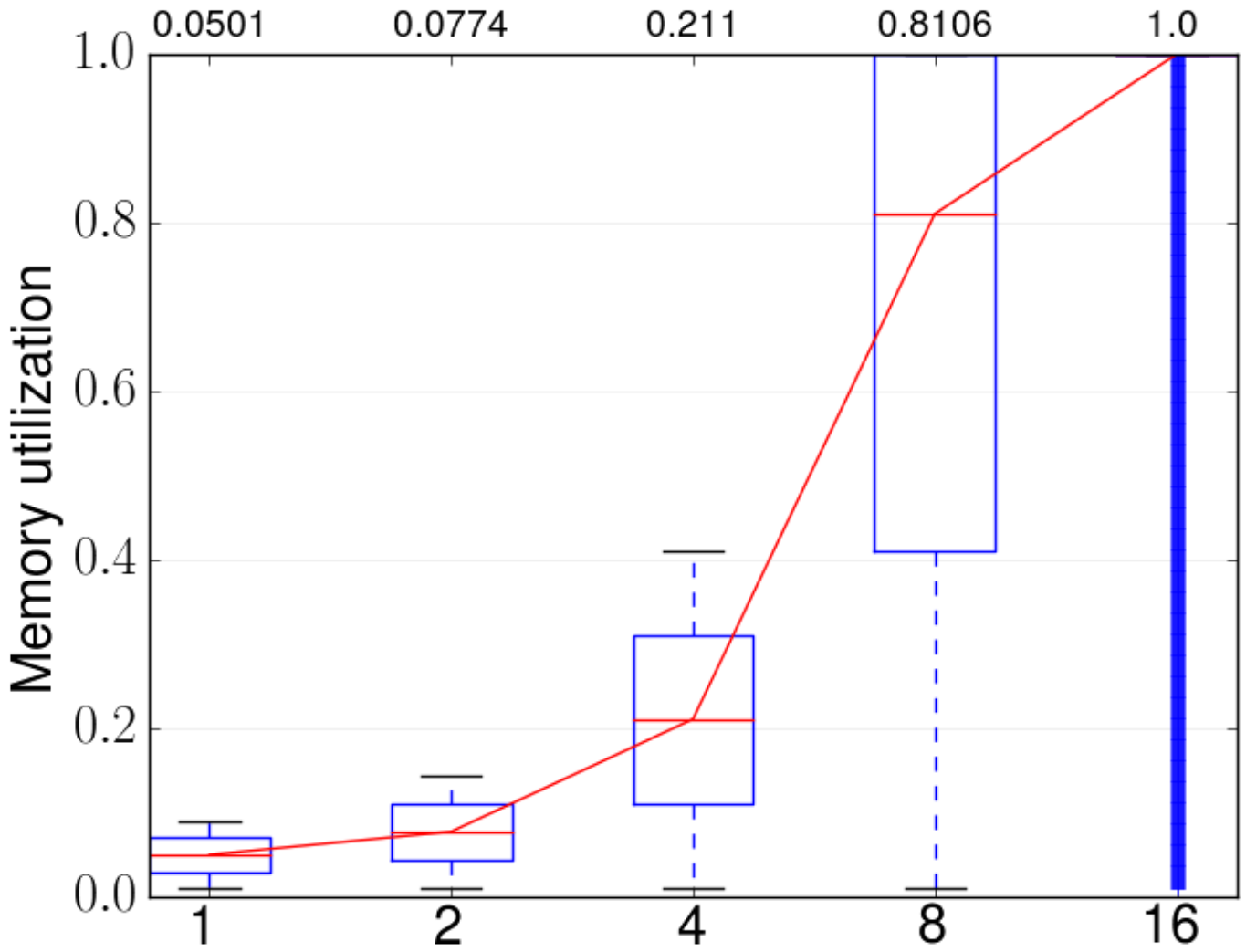
1. Detection distribuita: Raccolta dei dati

- **Raccolta distribuita di grandi quantità di dati legate ad un elevato numero di eventi**
  - Ciascun host fisico esegue 32 macchine virtuali
  - Ciascuna macchina virtuale è equipaggiata con sonde di monitoraggio
  - Valutiamo l'utilizzazione di CPU e di memoria del DB back end all'aumentare del numero di host fisici

1. Detection distribuita: Raccolta dei dati



1. Detection distribuita: Raccolta dei dati



1. Detection distribuita: Raccolta dei dati

- **Raccolta distribuita di grandi quantità di dati legate ad un elevato numero di eventi**
  - Adozione di una architettura di monitoraggio distribuita, gerarchica e ridondante per
    - Collezionamento dei dati
    - Filtraggio preliminare
    - Memorizzazione serie temporali
    - Correlazione, predizione, anomaly detection
  - “Gerarchica”
    - Uno storage per il monitoraggio short-term
    - Uno storage per il monitoraggio longer-term

1. Detection distribuita: Raccolta dei dati

- **Raccolta distribuita di grandi quantità di dati legate ad un elevato numero di eventi**
  - Architettura basata sul paradigma MapReduce
  - Implementazione tramite lo stack Hadoop
  - Short-term:
    - Monitoraggio: monitor Chukwa
    - Filtraggio preliminare: script Map Reduce ad hoc
    - Storage: cluster HDFS
  - Long-term:
    - Analisi: Pig Latin
    - Storage: cluster HBase

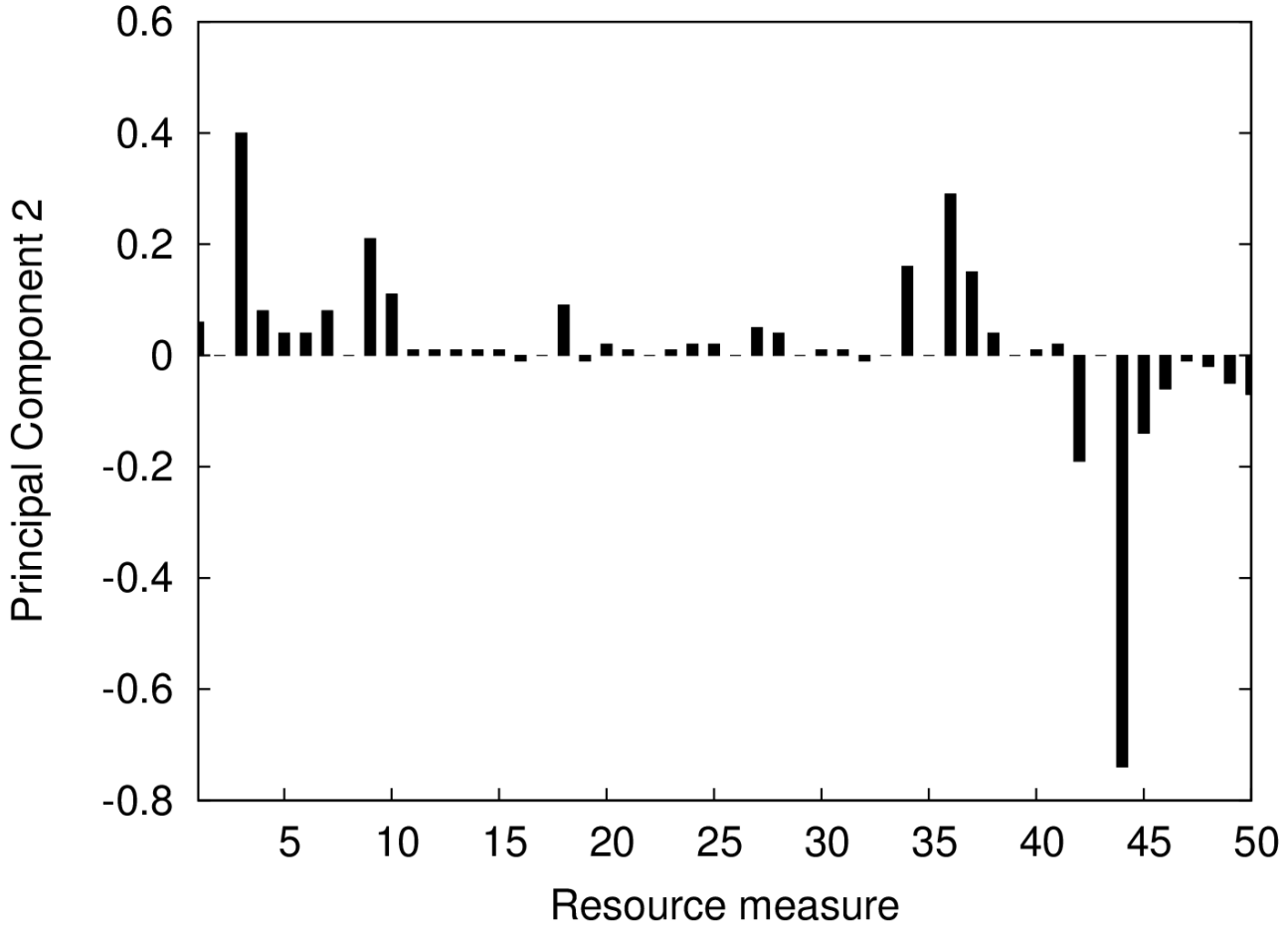
1. Detection distribuita: Raccolta dei dati

- **Applicazione di tecniche di “pulizia”**
  - Controlli short-term di significatività dei dati
    - Presenza
    - Validità (range check, null check)
  - Filtraggio short-term statistico
    - Eliminazione di outlier evidenti
- **Applicazione di algoritmi stocastici per l'individuazione di un andamento pulito del carico**
  - Modelli lineari: medie mobili, AR, ARMA
  - Modelli non lineari: cubic spline, wavelet

1. Detection distribuita: Elaborazione dei dati

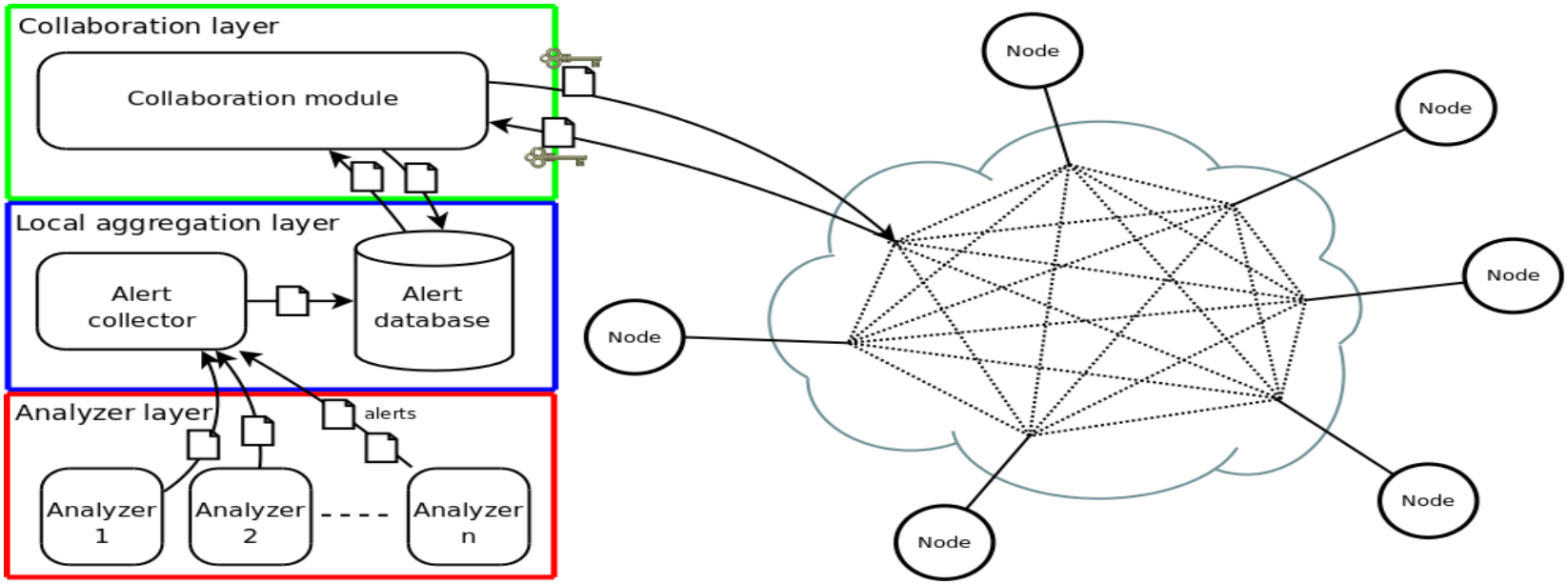
- **Applicazione di tecniche statistiche per il riconoscimento precoce di potenziali minacce (early detection)**
  - Implementate nei moduli di analisi longer term
    - Anomaly detection
    - State change detection: CUSUM algorithm
    - Correlazione fra dati appartenenti a sorgenti eterogenee ed individuazione dei flussi dati rilevanti (PCA analysis)

1. Detection distribuita: Elaborazione dei dati



1. Detection distribuita: Elaborazione dei dati

- **Disseminazione precoce di allarmi legati a potenziali minacce (*early warnings*)**
  - Da fare
  - Strategia di elaborazione: P2P
    - Condivisione degli allarmi e delle risorse di calcolo attraverso un modulo di collaborazione



2. Reaction distribuita: Protezione dei servizi

- **Implementazione dei servizi su architettura virtualizzata**
  - Supporto alla virtualizzazione integrato nell'architettura di monitoraggio
    - Tramite script ad-hoc
  - In alternativa
    - Gestione delle macchine virtuali in OpenStack

2. Reaction distribuita: Protezione dei servizi

- **Migrazione live dei servizi attaccati su reti più sicure**
  - Supporto alla migrazione live/offline integrato nell'architettura di monitoraggio
  - DA FARE: identificazione delle reti “più sicure”
- **Trasformazione del servizio attaccato in un Honeypot**
  - DA FARE: attivazione di un Honeypot
    - Alternativa meno costosa: sostituzione del livello back-end associato al servizio

2. Reaction distribuita: Contrasto alle minacce

- **Deviazione del traffico in ingresso basata su DNS**
  - DA FARE
- **Applicazione di filtri di traffico**
  - DA FARE: attivazione di regole di firewall
- **Contrasto attivo (laddove possibile) sul canale di comunicazione delle botnet**
  - DA FARE: comandi IRC