

Network Events Correlation for Federated Networks Protection System

Michał Choras

Rafał Piotrowski

Juliusz Brzostek

Witold Holubowicz

Rafał Kozik

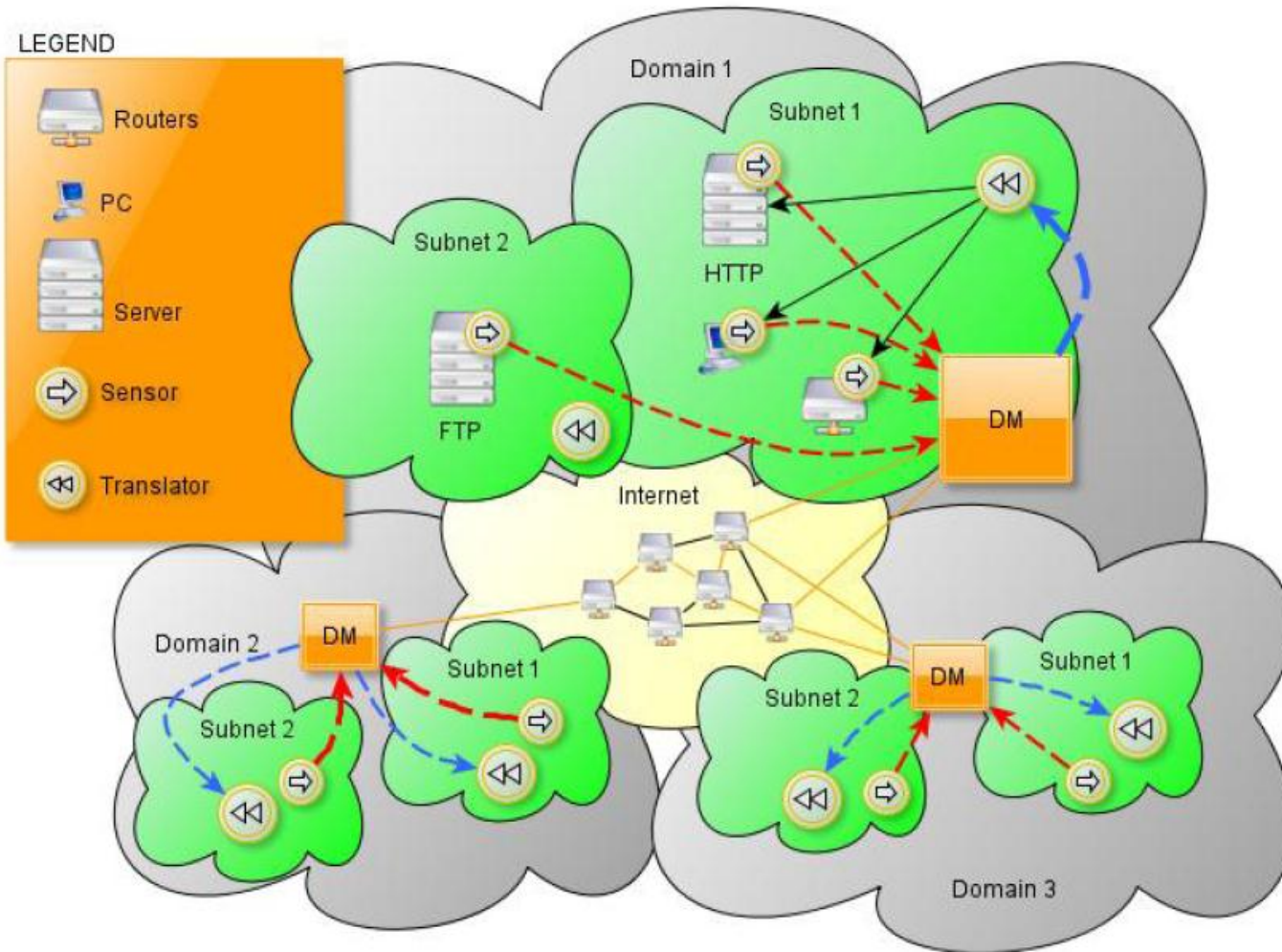
rafal.kozik@itti.com.pl

Introduction

- **The concept of Federated Networks Protection System is being developed in the SOPAS national research project.**
- **The goal of the Federated Networks Protection System developed in the SOPAS project is to protect public administration and military networks which are often connected into Federations of Systems (FoS).**
- **While adopting the concept of federation of networks, the synergy effect for security can be achieved.**

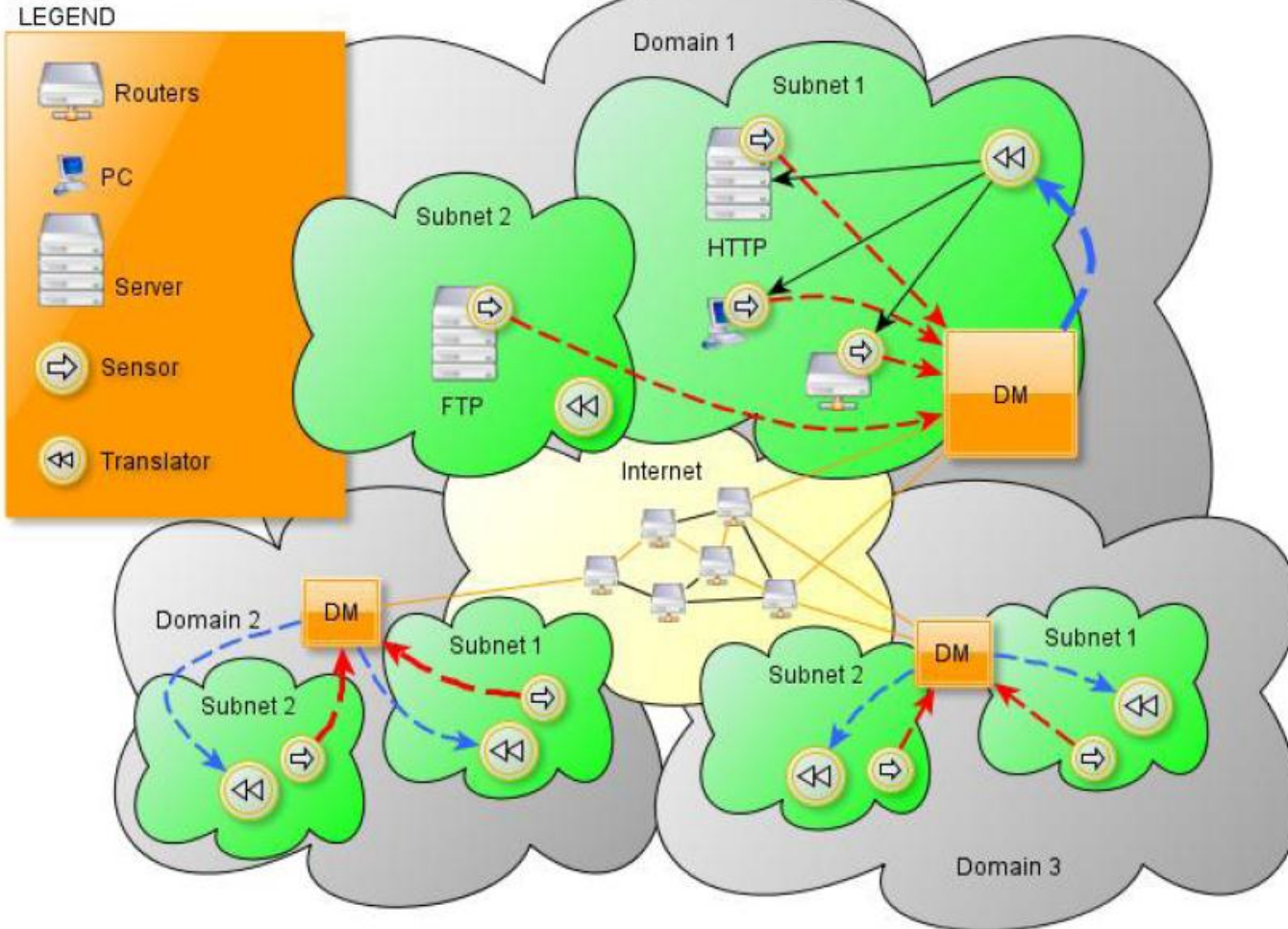
- **This presentation is focused on the:**
 - ✓ **SOPAS Architecture**
 - ✓ **The Decision Module**
 - ✓ **The correlation mechanism**
 - ✓ **The semantic reasoning based on the ontology**
 - ✓ **Communication between the distributed elements of the system**

SOPAS - general architecture



- Distributed topology
- Assets (e.g. WWW, FTP or SQL servers)
- Installed sensors:
 - ✓ Application layer sensors
 - ✓ IDS and IPS systems
 - ✓ Anomaly Detection Systems
 - ✓ ARAKIS
 - ✓ HoneySpider Network (HSN) system
- Decision Modules (FNPS-DM)
- Translator (reaction)

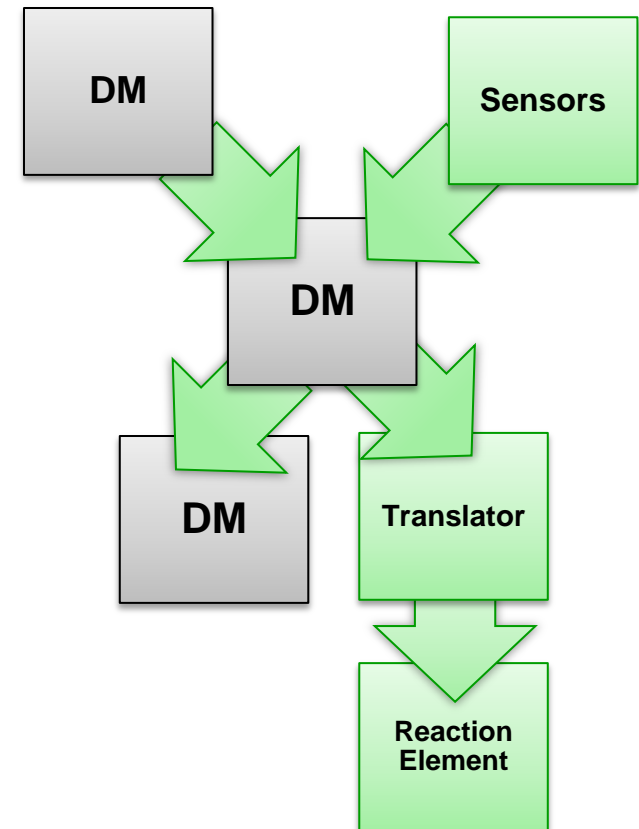
SOPAS - general architecture



- **ARAKIS** is early warning system detecting novel network threats. **ARAKIS-GOV**, a version dedicated for protecting public administration networks, is widely installed in polish public networks.
- **The HoneySpider Network (HSN)** is a system focused primarily on attacks against, or involving the use of, web browsers.

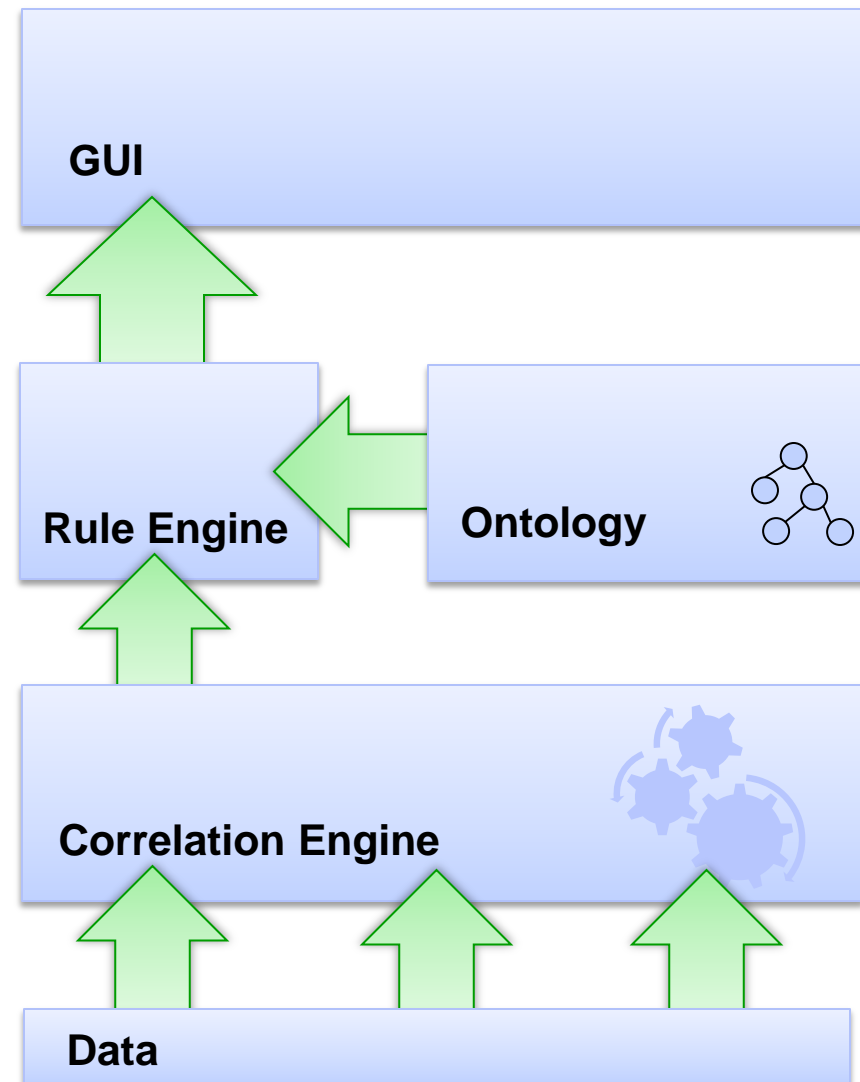
Decision Module (FNPS-DM)

- **Located in each protected domain**
- **Responsible for:**
 - ✓ acquiring network events,
 - ✓ processing network events,
 - ✓ correlating network events,
 - ✓ applying reactions via Translator.
- **Human-in the loop (GUI dedicated for the operator)**



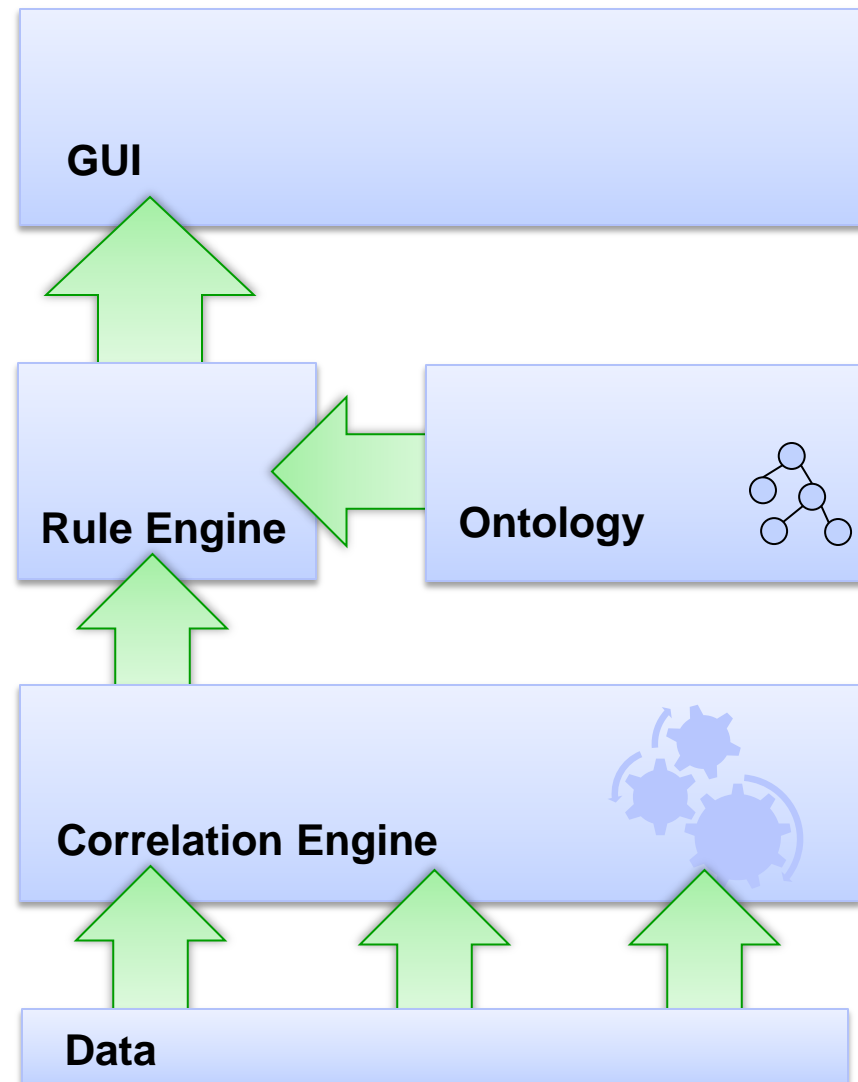
MD components

- **Consists of the following components:**
 - ✓ **Correlation Engine (e.g. based on the Borealis system),**
 - ✓ **CLIPS rule engine,**
 - ✓ **Ontology (in OWL format),**
 - ✓ **Graphical User Interface.**



MD components

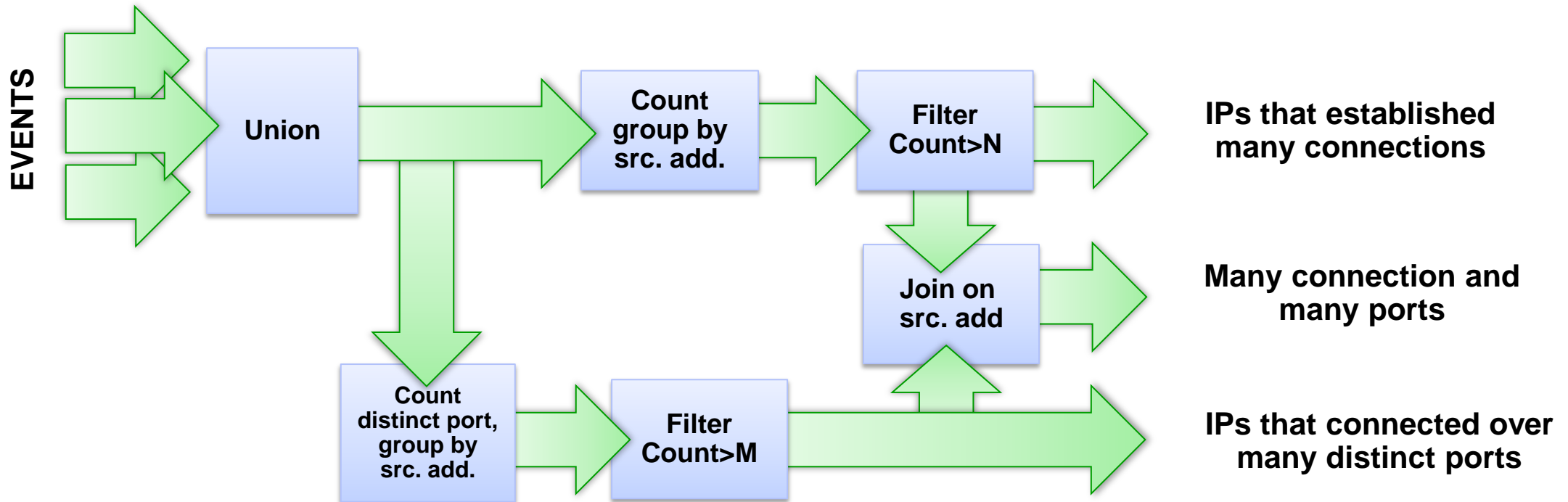
- **Borealis is a distributed stream processing engine and is responsible for gathering information generated by the network sensors.**
- **CLIPS uses ontology that describes broad range of network security aspects.**



Event correlation

- **Data received from network sensors is arranged in streams**
- **Each stream is built of multiple tuples (events)**
- **Borealis (correlation engine) processes streams (using queries) in order to correlate information coming from different sources**
- **Each query executed over the multiple streams consists of operators**

Event correlation – query example



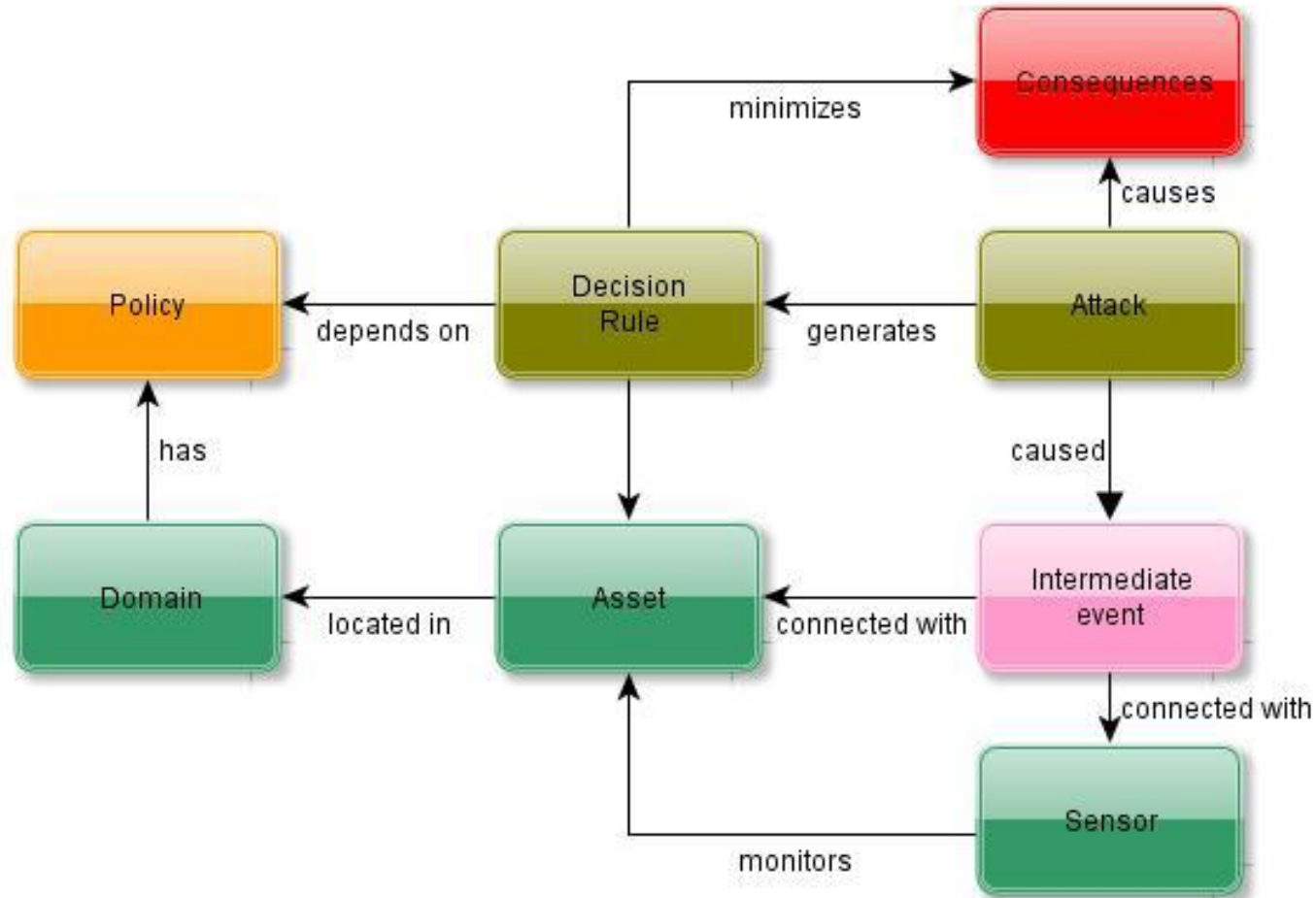
Ontology

- **It is a form of representing a data model of a specific domain**
- **Defines basic terms and relations (security domain)**
- **Security aspects are modeled and formalized in the OWL format**
- **Semantic rules are developed in SWRL language**
- **It provides common language that increases interoperability between domains**

Ontology

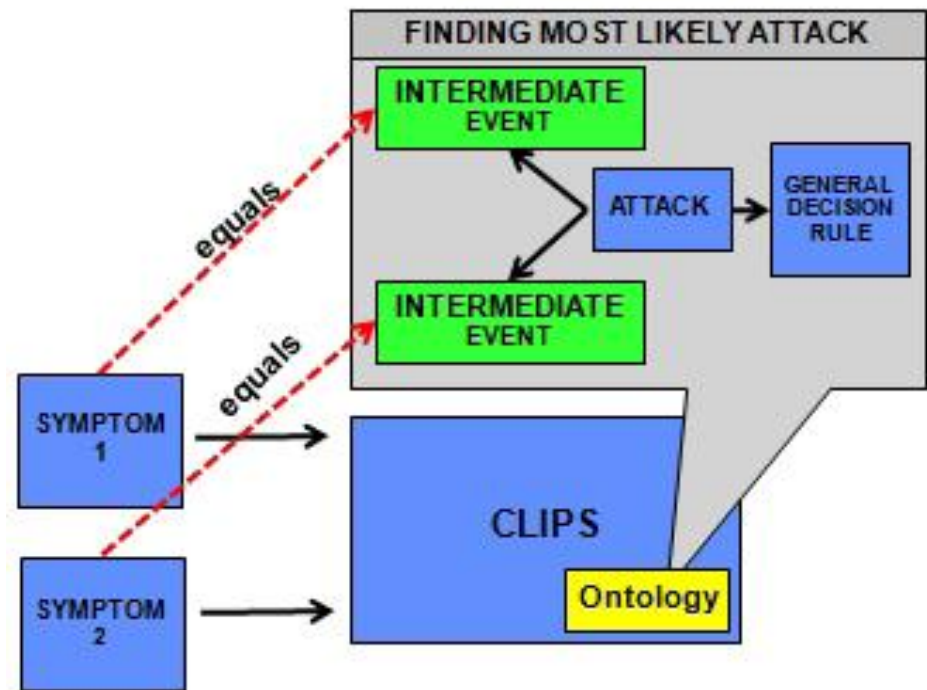
- **It allows the DMs to communicate using common abstract layers and to reason about security facts using common language**
- **It describes following aspects:**
 - ✓ **Attacks, symptoms, attack impact and reactions to attack**
 - ✓ **Asset description and their relations**
 - ✓ **Policy (what reactions are recommended/allowed in the particular domain)**
 - ✓ **Decision Rules (how to react to attack)**

Ontology – main classes



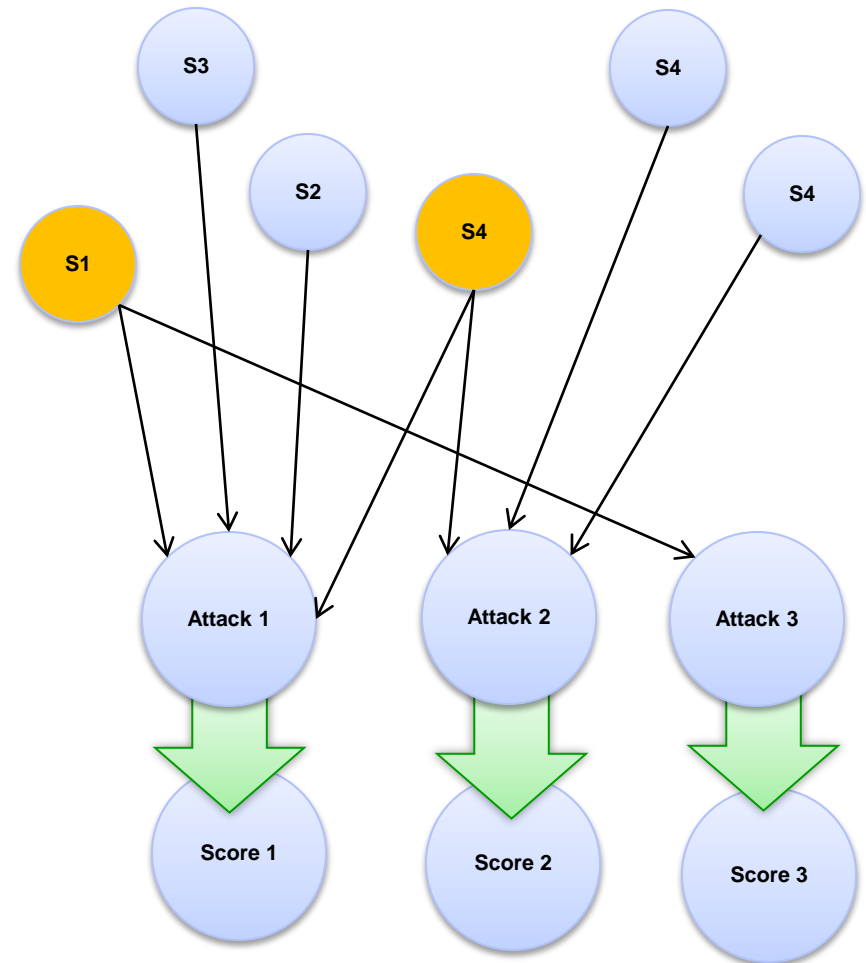
Decisions enhanced with semantic reasoning

- Each intermediate event received by CLIPS rule engine is considered as attack symptom
- Symptom is matched with knowledge in the ontology in order infer the most probable attack



Decisions enhanced with semantic reasoning

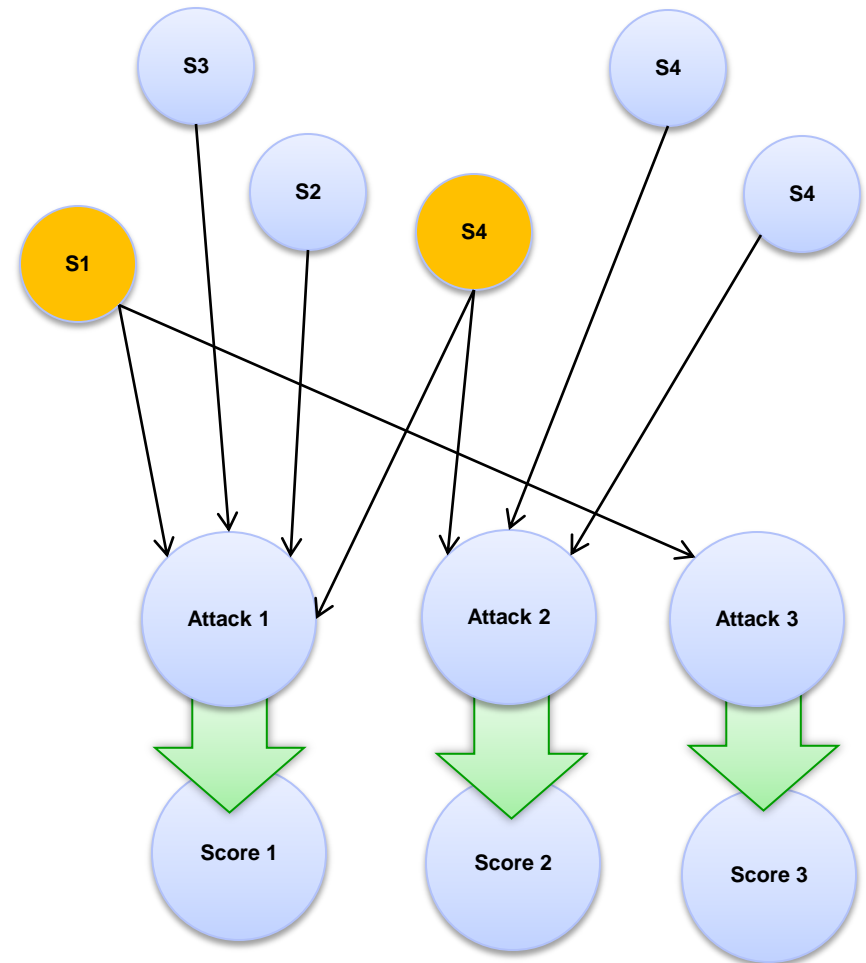
- However, one symptom could match several attacks
- CLIPS is responsible for computing the probability score



Decisions enhanced with semantic reasoning

- The score is computed as probability $p(A|o_1, o_2, \dots, o_n)$
- Finding the most probable attack is modelled as MAP (Maximum A-Posteriori) problem

$$A^* = \arg \max_{AP} (A|o_1, o_2, \dots, o_n)$$

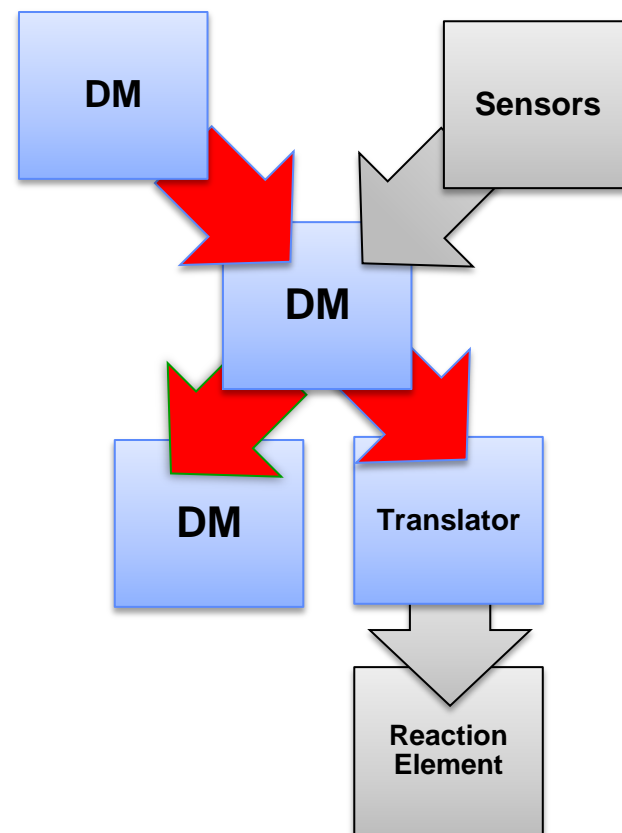


Decisions enhanced with semantic reasoning

- We have assumed that observations (intermediate events) are mutually independent.
- Such approach allows us to apply Bayes theorem to estimate the probability of the attack

$$p(A|o_1, o_2, \dots, o_n) = \frac{1}{Z} p(A) \prod_{i=1}^N p(o_i|A)$$

- **DM communicates with:**
 - ✓ Other DMs
 - ✓ Translator
- **DM- Translator is SOAP-based communication**
- **DM-DM uses P2P protocol**

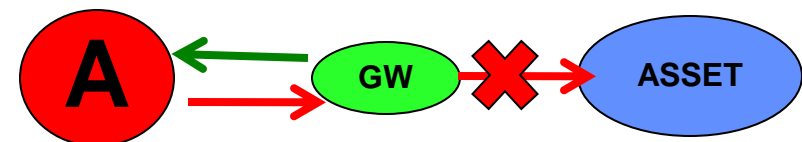
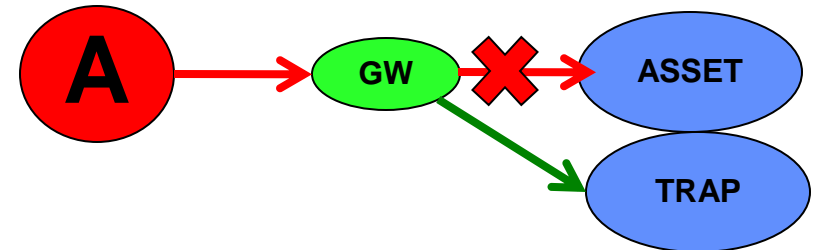
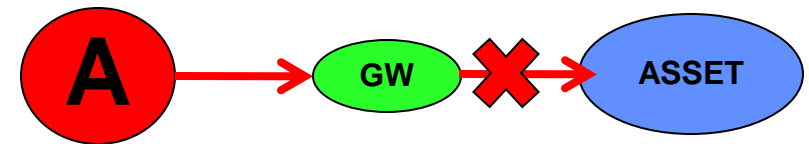


SOAP method:

➤ trExecute(ord_type, ord_param)

✓ ord_type – defines reaction type

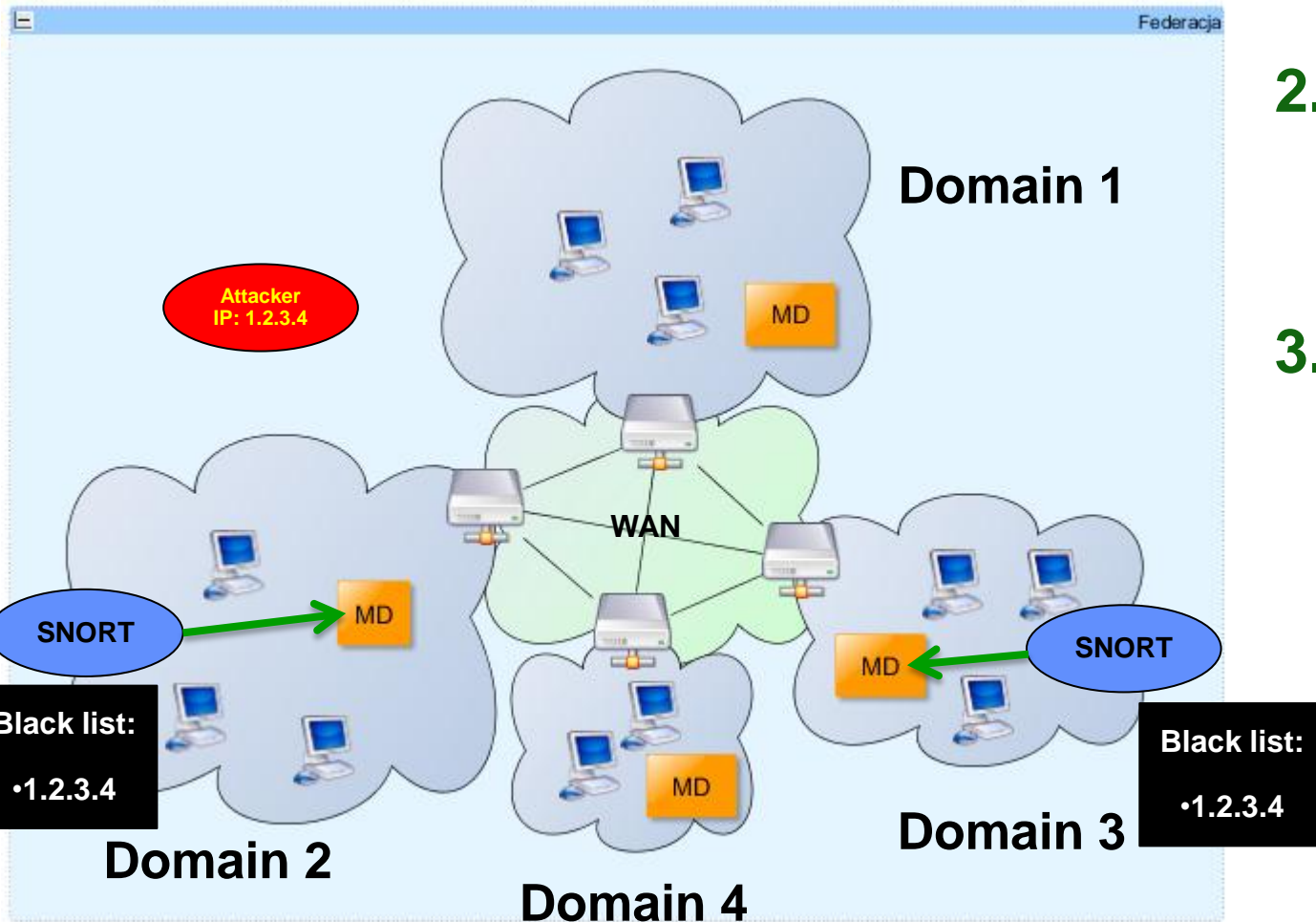
- 1=BLOCK/DROP
- 2=FORWARD_TO_TRAP
- 3=ECHO



Communication: DM-DM

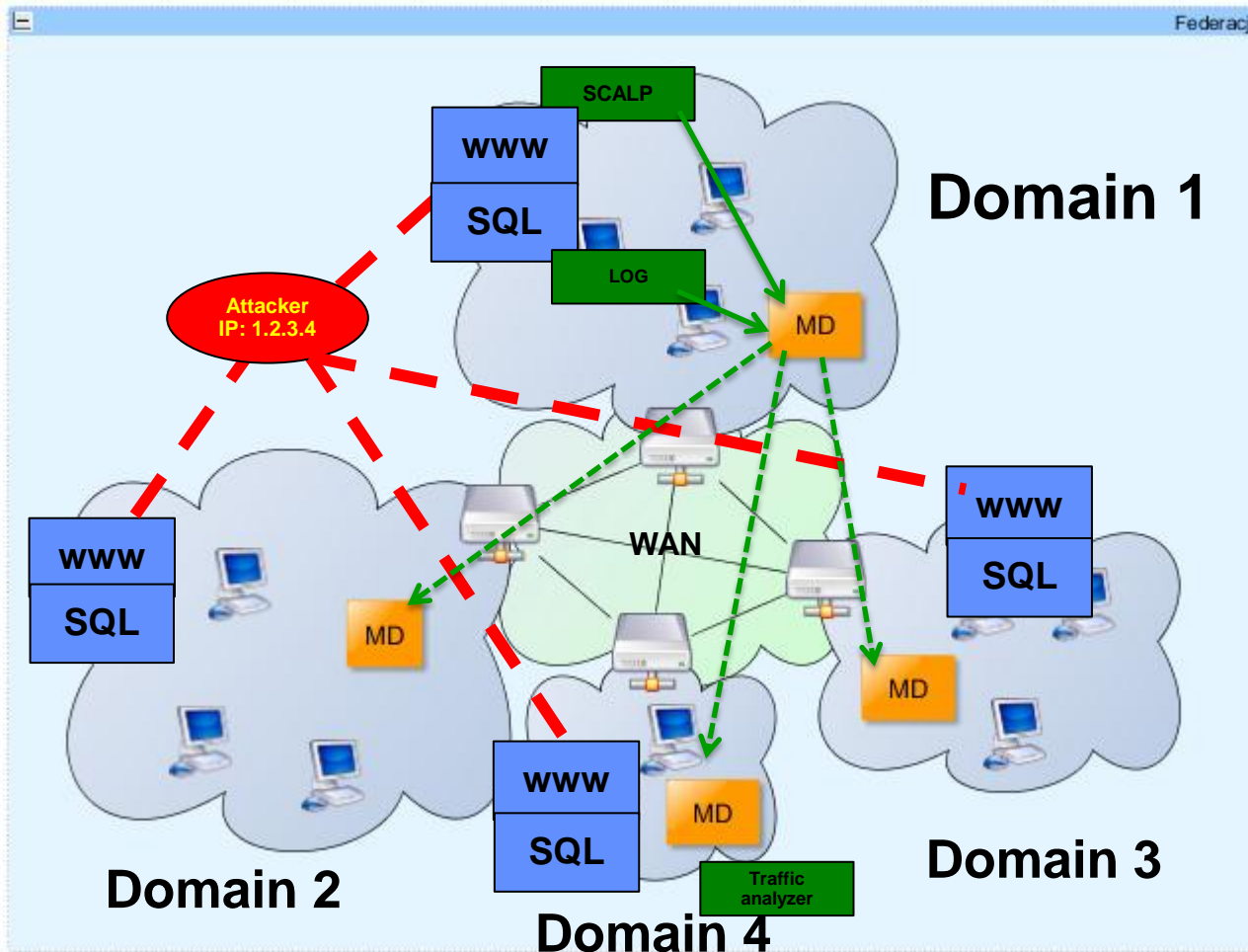
- **The proposed P2P overlay network is dedicated only for a DMs**
- **Each Decision Module is a peer hosting and requesting data concerning federation security aspects.**
- **It is possible that other machines (not only DMs) may also act as peers in FNPS overlay network.**
- **This approach allows the proposed system to overcome IP addressing issues and minimize the configuration cost.**

Correlation: SQLIA detection example



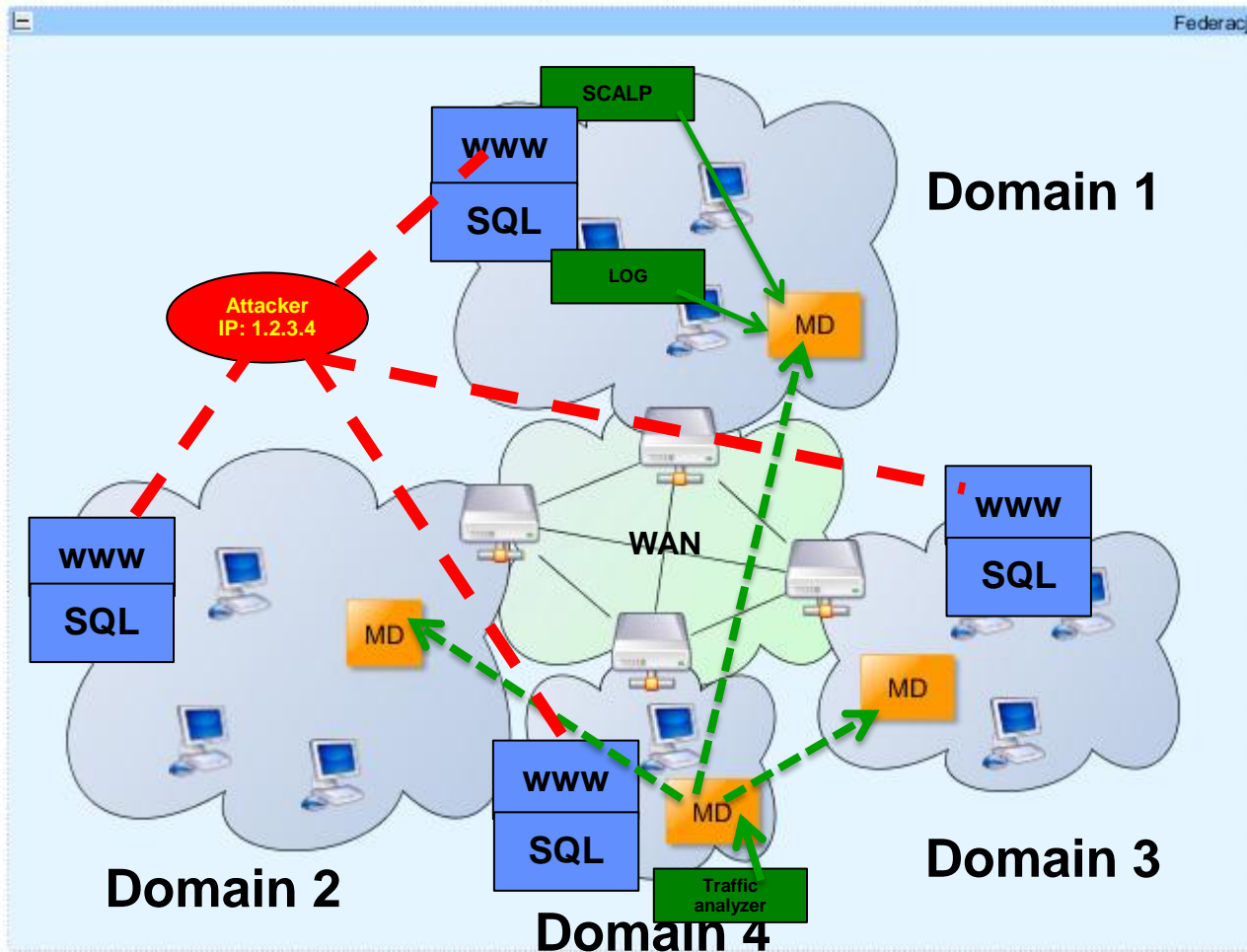
1. IP scanning.
2. Sensors detect port scanning in domain 2 and 3.
3. Decision modules MD 2 and MD3 store IP in the blacklist.

Correlation: SQLIA detection example



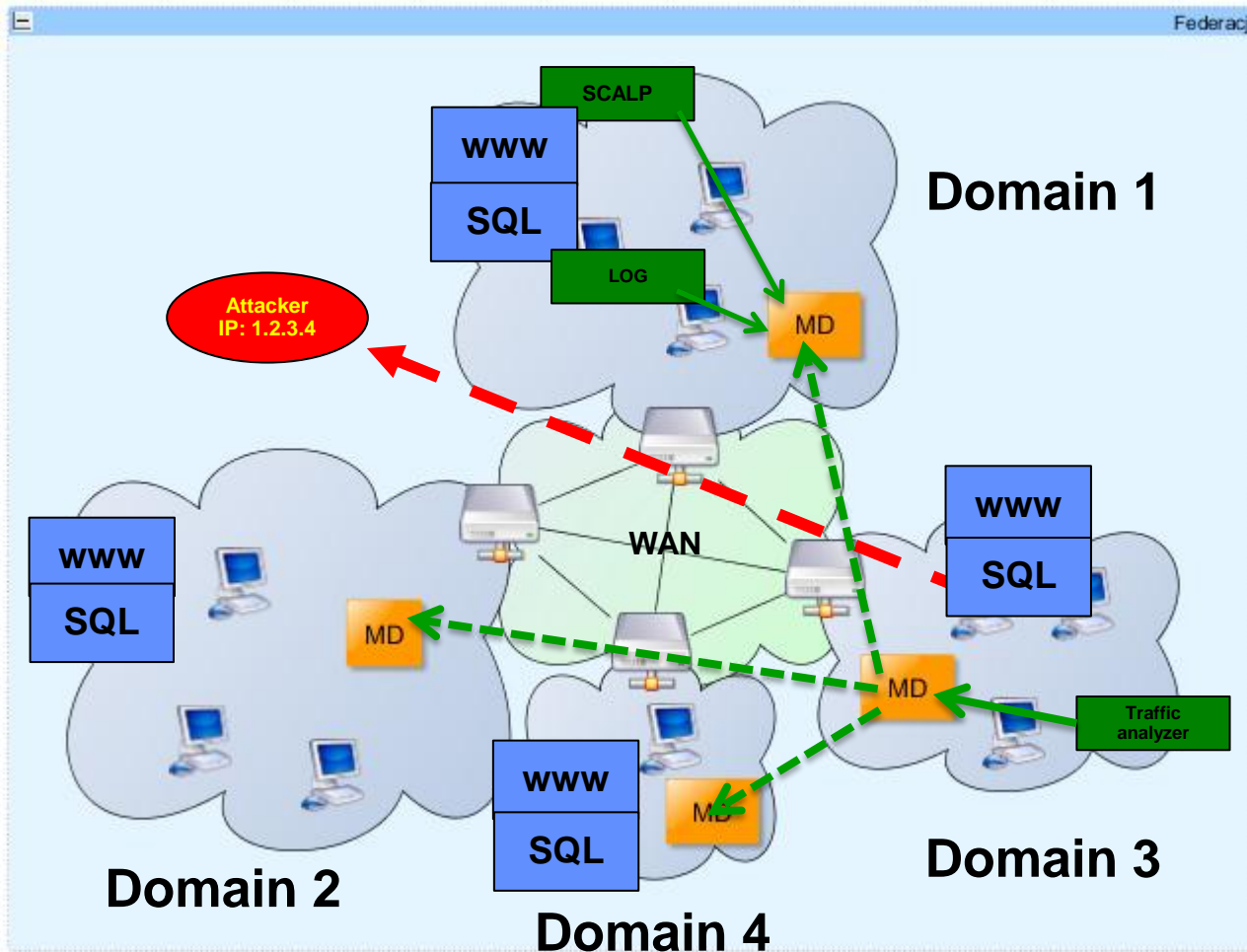
4. Attacker performs penetration tests of many services to find problems with their security.
5. Sensors in domain 1 detect increased traffic coming from one IP in http logs and send this information to MD1. MD1 informs other MDs.
6. Sensors in domain 1 detect many unsuccessful queries in SQL logs. MD1 informs other MDs.

Correlation: SQLIA detection example



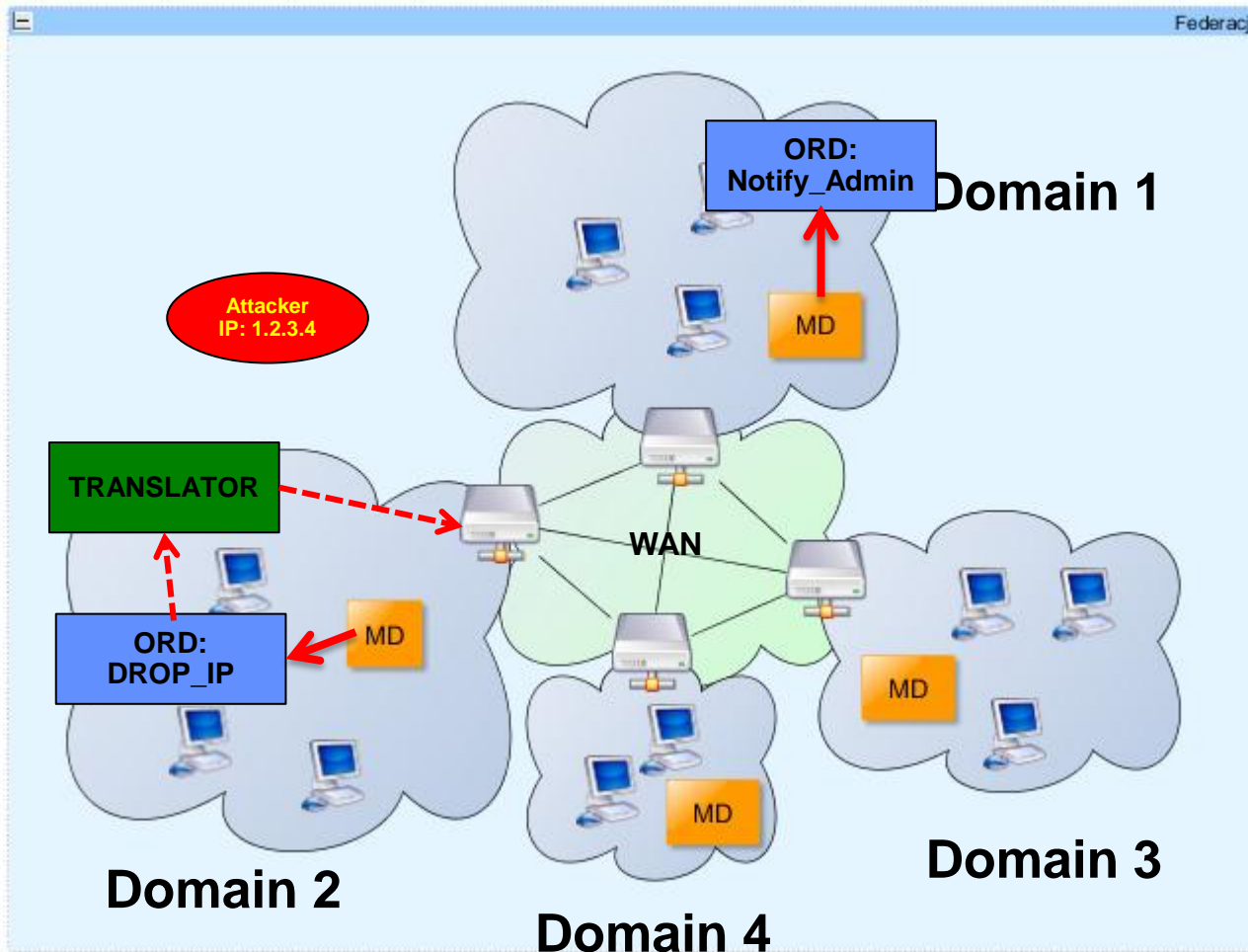
- Domain 4 has no http and SQL logs sensors installed. However, transport layer sensors (ADS) in domain 4 detect suspicious traffic load. MD4 sends this information to other MDs.

Correlation: SQLIA detection example



8. In domain 3, the attack was successful. Transport layer sensors detect large amount of traffic uploaded from domain 3. MD3 sends this information to other domains.

Correlation: SQLIA detection example



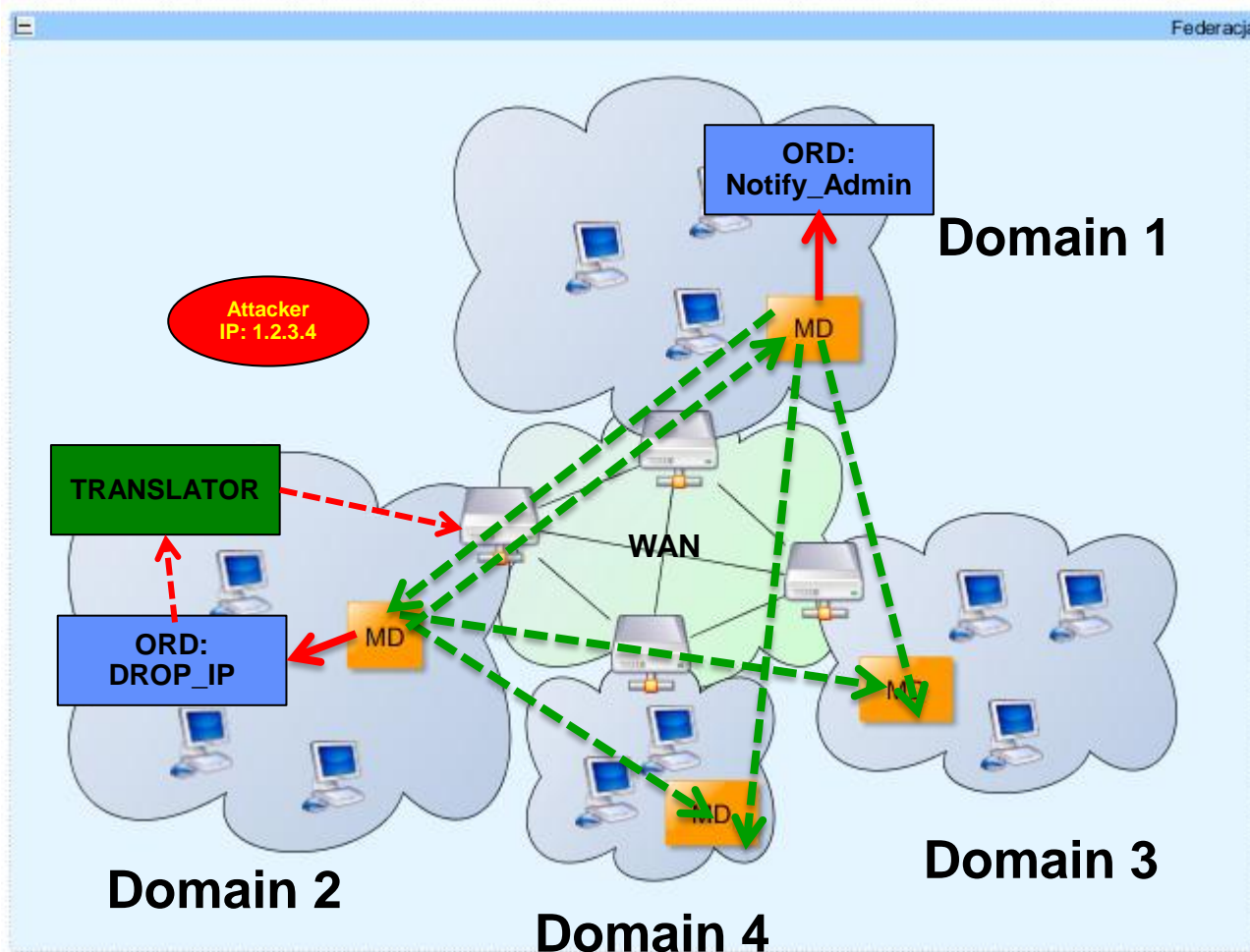
9. After information sharing process, MD know about:

- ✓ Increased flow in transport layer
- ✓ Increased flow in application layer
- ✓ Scanning

10. MD1 generates its decision rule and reaction (inform administrator)

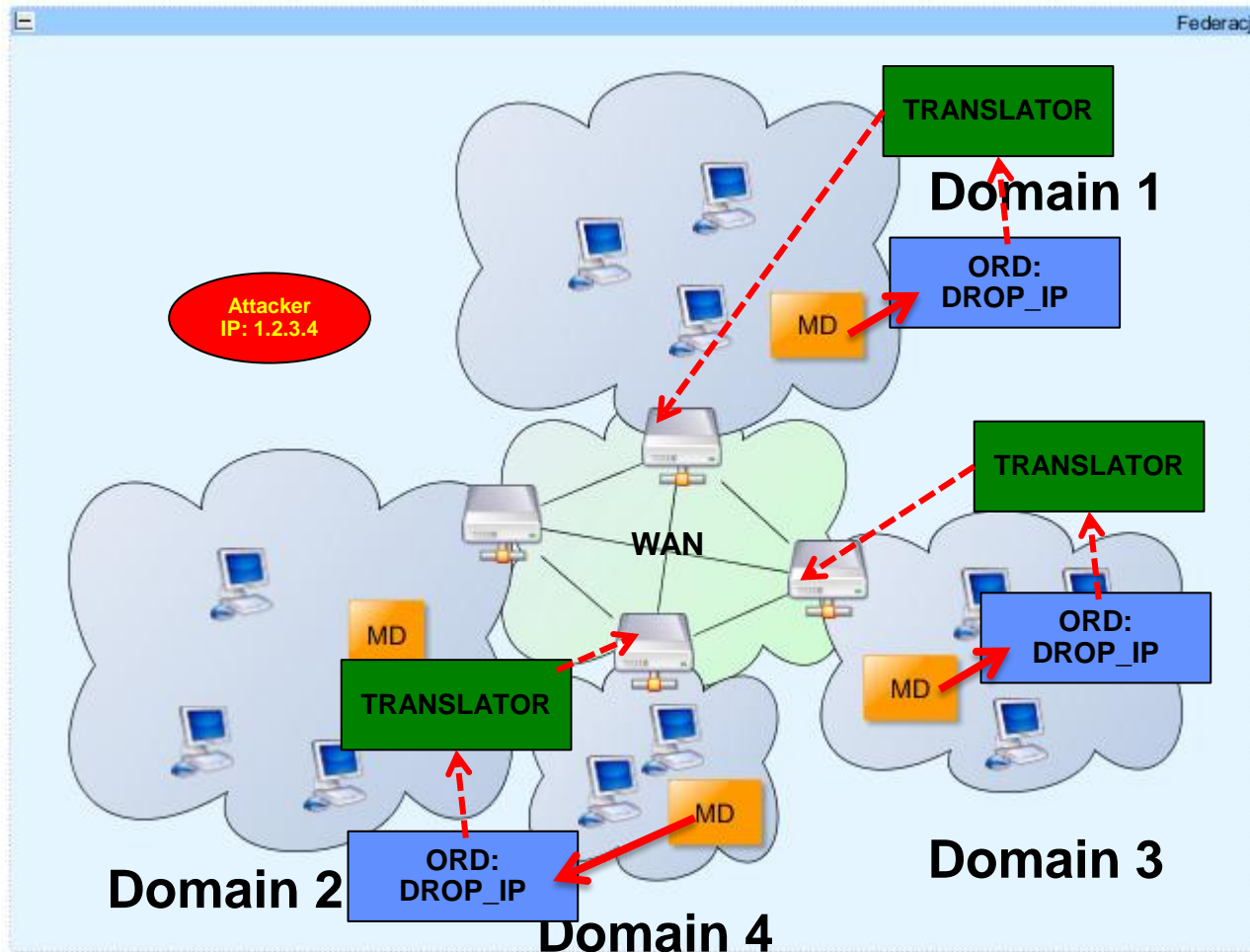
11. MD2 generates its decision rule and reaction block traffic from IP.

Correlation: SQLIA detection example



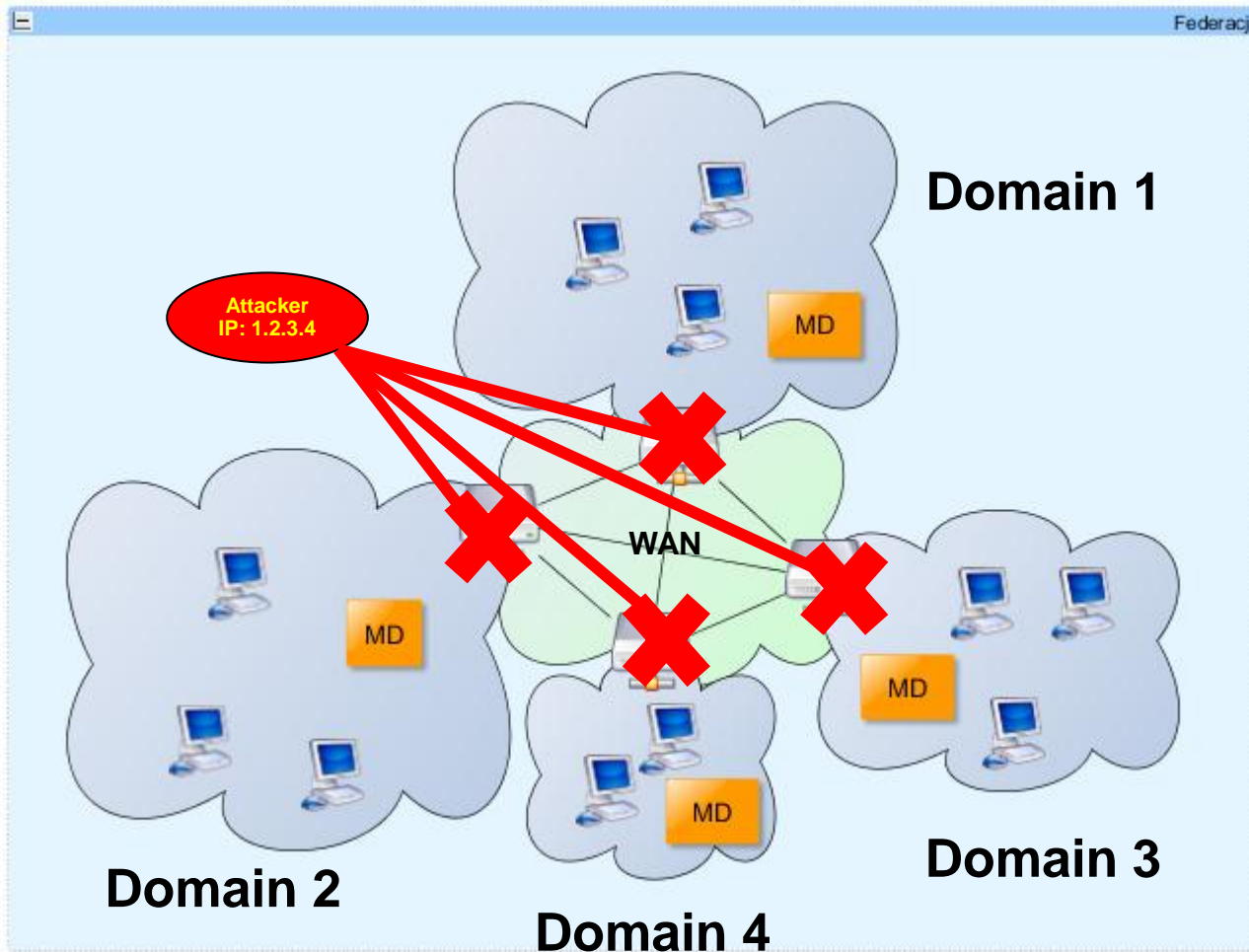
13. MD1 and MD2 send information about attack to MD3 and MD4.

Correlation: SQLIA detection example



14. Reaction rules are distributed and applied.

Correlation: SQLIA detection example



Conclusions

- **Presented paper describes preliminary results of the national project SOPAS funded by Ministry of Science and Higher Education of Poland in the theme of homeland security.**
- **The major contribution of this paper is the concept of Federated Networks Protection System that is being developed in the SOPAS project.**
- **The presented system is dedicated for federated networks and systems used by the public administration and military sector.**