

# Komponenten für kooperative Intrusion Detection in dynamischen Koalitionsumgebungen

**Marko Jahnke**  
*FGAN/FKIE*  
Abt. Kommunikation  
Neuenahrer Str. 20  
D-53343 Wachtberg  
jahnke@fgan.de

**Unter Mitarbeit von**  
Sven Henkel,  
Michael Bussmann  
*FGAN/FKIE*

Jens Tölle  
*Universität Bonn*

**06. Juli 2004**

Frank Ausserlechner  
*FH Koblenz*

# Übersicht

---

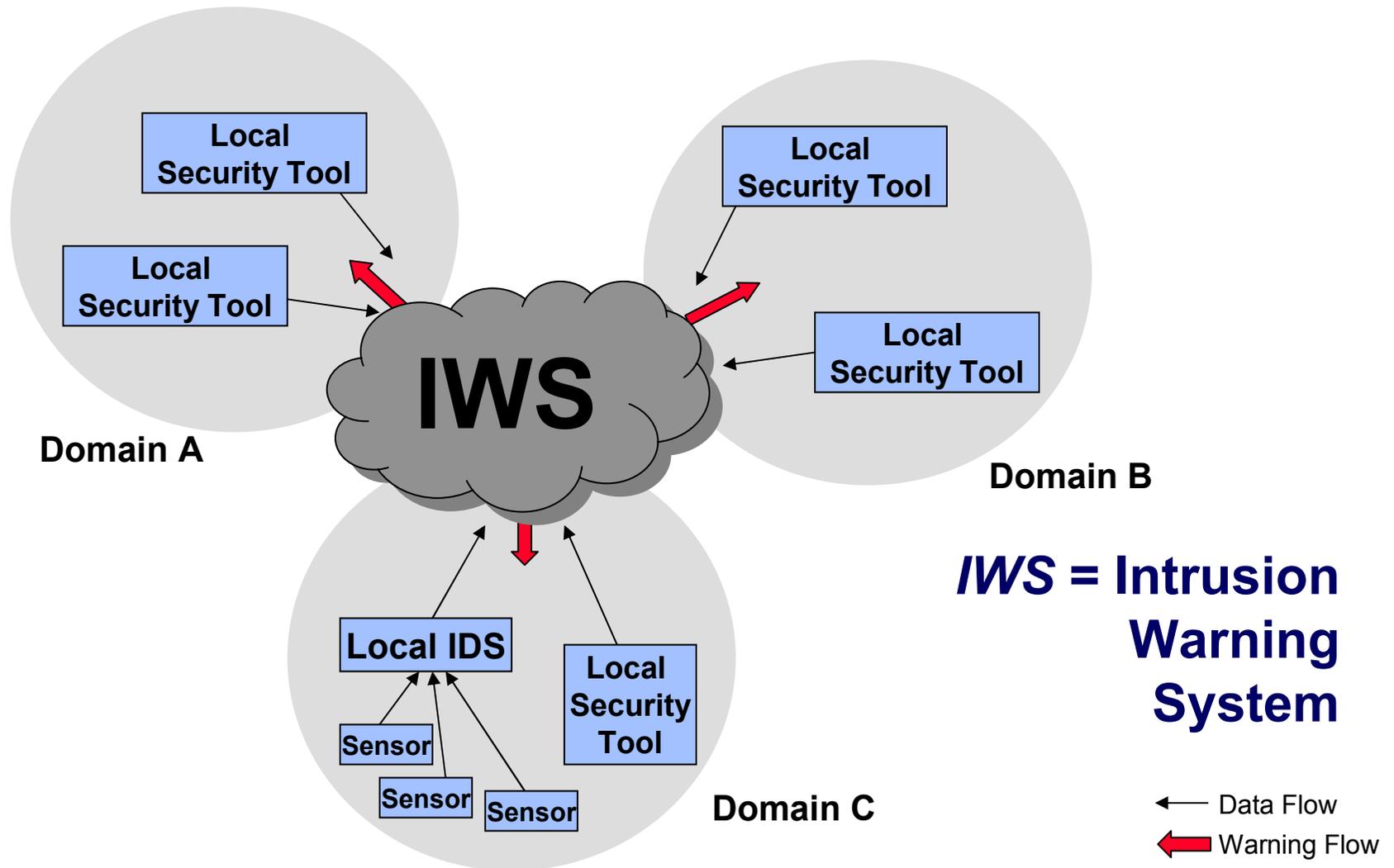
- Herausforderungen für IDS in Koalitionsumgebungen
- Architekturen für kooperative Intrusion Detection
- Vorverarbeitung von Ereignismeldungen
- Anomalieerkennung im Ereignismeldungs-Datenmodell
- *Implementation & bisherige Ergebnisse*

# Dynamische Koalitionsumgebungen

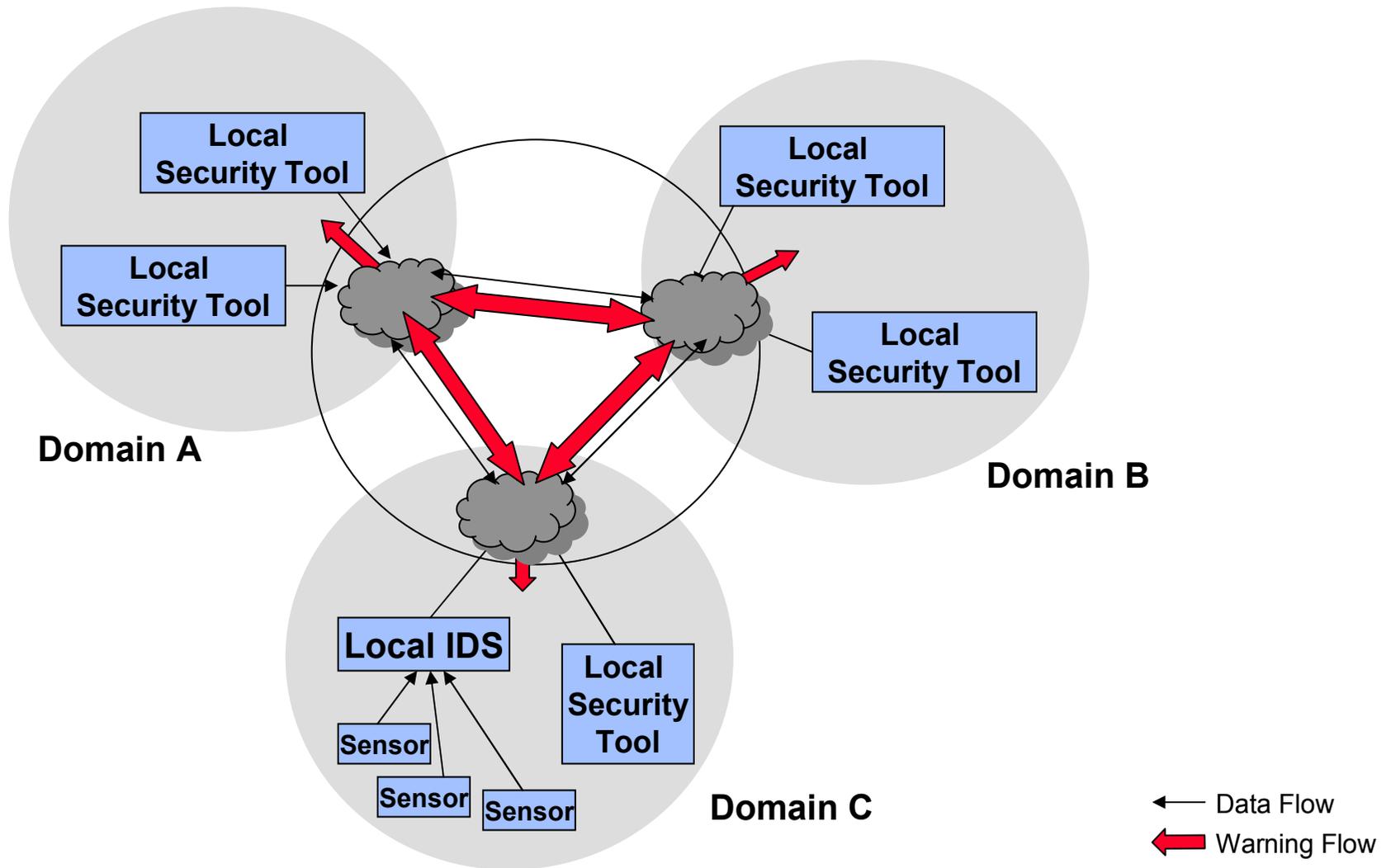
---

- Veränderliche Menge kooperierender, gleichberechtigter Domänen
- Gemeinsame und konträre Ziele und Ansichten
- Beispiele für Koalitionsumgebungen
  - Allianzen von Wirtschaftsunternehmen
  - Kooperierende Strafverfolgungsbehörden
  - Militärische Netzwerke (NATO, SFOR, KFOR, ...)
- Beispiele für Kooperation in Koalitionsumgebungen
  - Logistik & konventionelle Infrastrukturen
  - Elektronische Infrastrukturen

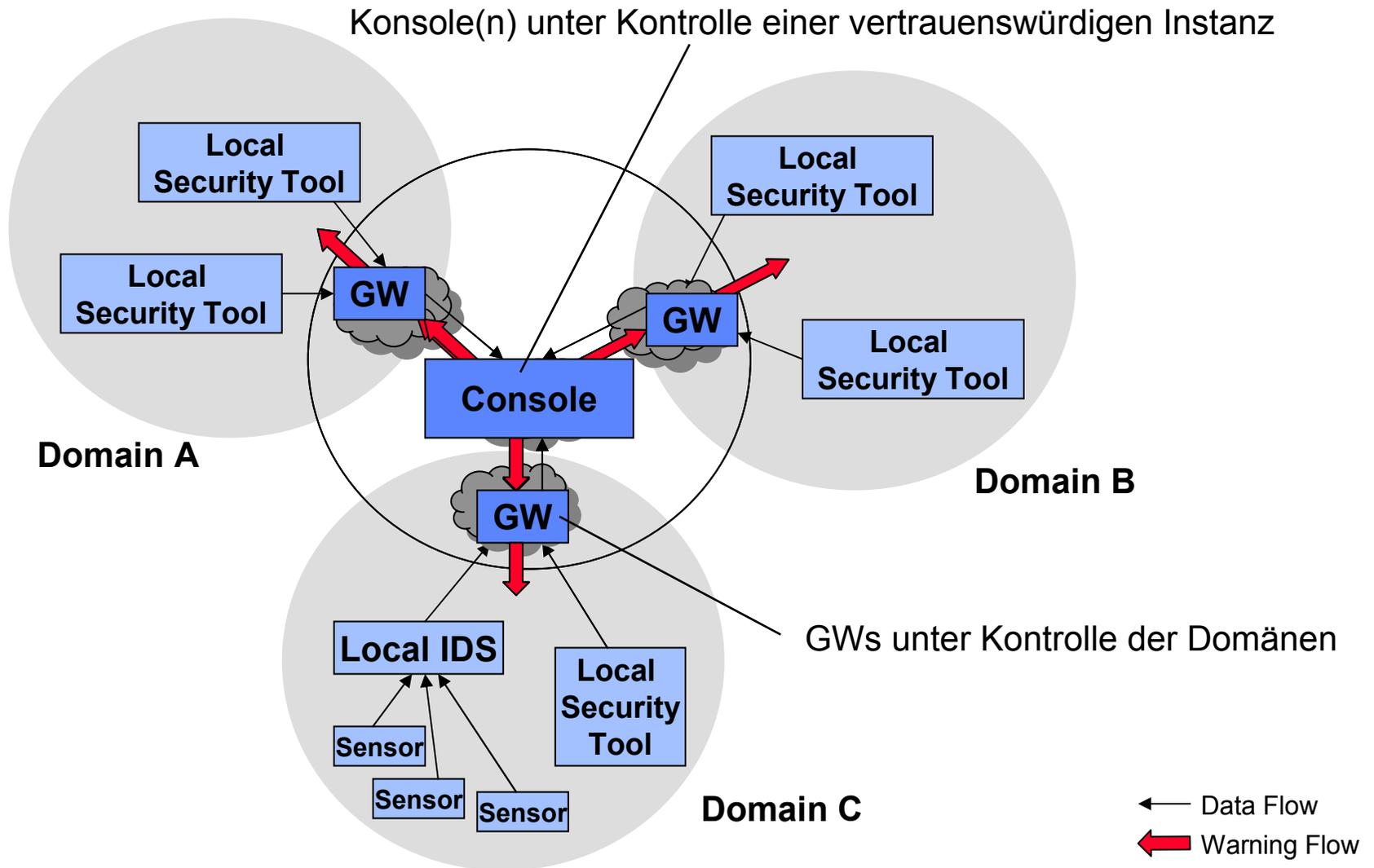
# Kooperative IDS in Koalitionsumgebungen



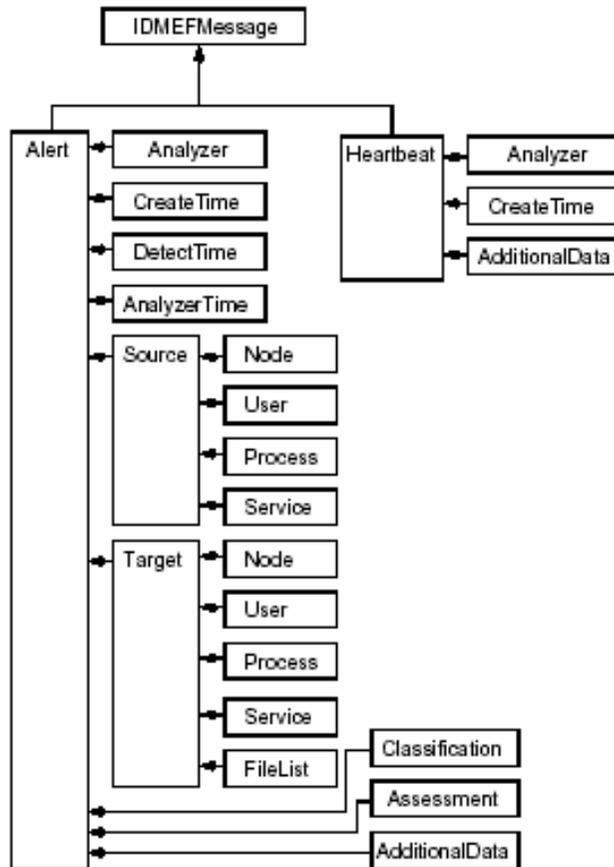
# IWS-Architekturen (1): Rein verteilte Kooperation



# IWS-Architekturen (2): Meta-IDS



# Interoperabilität: Datenmodell & -format



## Intrusion Detection Message Exchange Format (IDMEF)

```

<IDMEF-Message version="1.0">
  <Alert id="abc123456789">
    <Analyzer analyzerid="hq-dmz-analyzer01">
      <Node category="dns">
        <location>Headquarters DMZ Network</location>
        <name>analyzer01.example.com</name>
      </Node>
    </Analyzer>
    <CreateTime ntpstamp="0xbc723b45.0xef449129">
      2000-03-09T10:01:25.93464-05:00
    </CreateTime>
    <Source id="a1b2c3d4">
      <Node id="a1b2c3d4-001" category="dns">
        <name>badguy.example.net</name>
        <Address id="a1b2c3d4-002" category="ipv4-net-mask">
          <address>192.0.2.50</address>
          <netmask>255.255.255.255</netmask>
        </Address>
      </Node>
    </Source>
    <Target id="d1c2b3a4">
      <Node id="d1c2b3a4-001" category="dns">
        <Address category="ipv4-addr-hex">
          <address>0xde796f70</address>
        </Address>
      </Node>
    </Target>
    <Classification origin="bugtraqid">
      <name>124</name>
      <url>http://www.securityfocus.com</url>
    </Classification>
  </Alert>
</IDMEF-Message>
  
```

**Wer berichtet?**

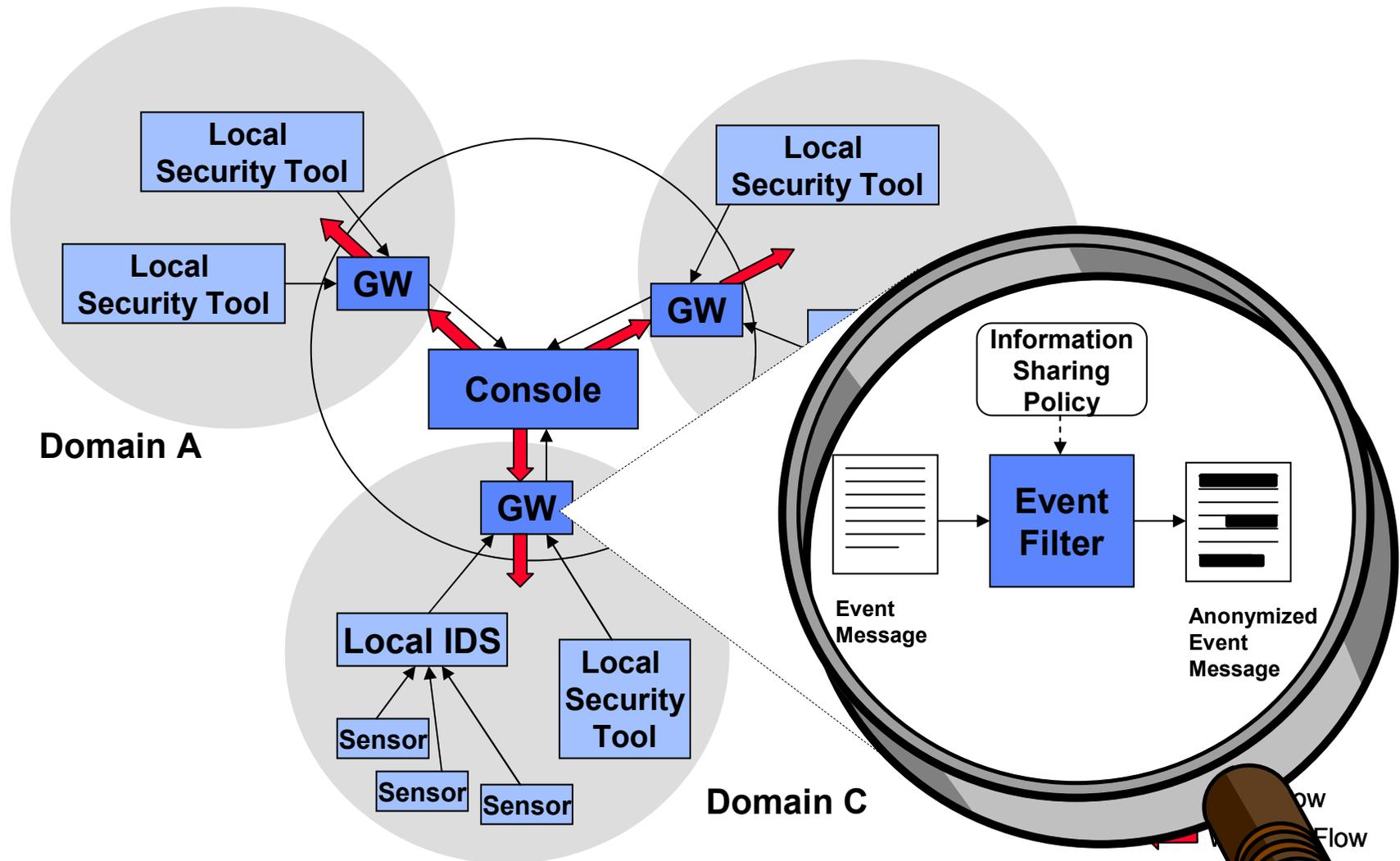
**Wann?**

**Wer?**

**Wo?**

**Was ist passiert?**

# Informationsbereinigung (1): Szenario



# Informationsbereinigung (2): Implementierung

- Spezieller XSLT-Prozessor mit Erweiterungen für *Matching-sensitive* 1:1-Transformationen

```
<IDMEF-Message ...>
```

```
...
```

```
<address>
```

```
192\.\.22\.\.([0-9]{1,3})$v1$\.\.([0-9]{1,3})$v2$
```

Matching-Ausdruck m. SMs

```
<condition>  
  $v2<255)  
</condition>
```

Zusätzliche Bedingung

Matching-Template

```
<transform>
```

```
<address> <xsl::copy>
```

```
191.72.<xsl::value-of select="$v1"/>.<xsl::value-of select="$v2"/>
```

```
</xsl::copy> </address>
```

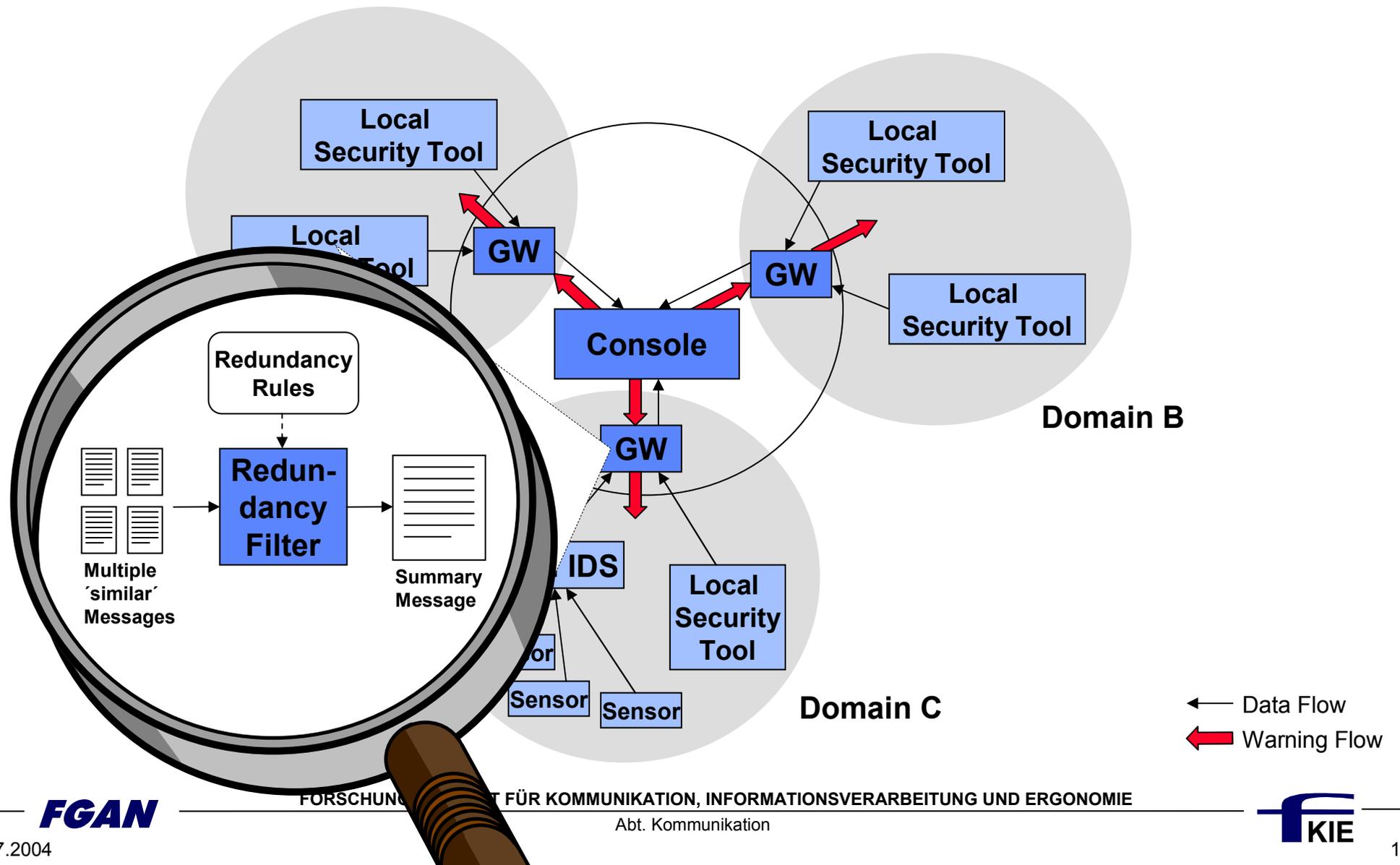
```
</transform>
```

Transformations-Template

```
</address>
```

```
...
```

# Redundanzfilterung (1): Szenario



# Redundanzfilterung (2): Implementierung

- Erweiterung des Informationsbereinigers um *relative Matchings* für Ähnlichkeitseigenschaft bei  $n:1$  -Transformationen

<IDMEF-Message ...>

```
...
<address>(.*?)$SRCADDRESS$ </address>
...
<relMatch deltaT="20000" maxMatchings="40">
  <xsl:copy>
    ...
    <address> <xsl:value-of select="$SRCADDRESS"/> </address>
    ...
    <summary do="create">
      ..
    </summary>
    <summary do="update">
      ...
    </summary>
  </xsl:copy>
</relMatch>
```

**Absolutes Matching-Template**

**Matching-Ausdruck**

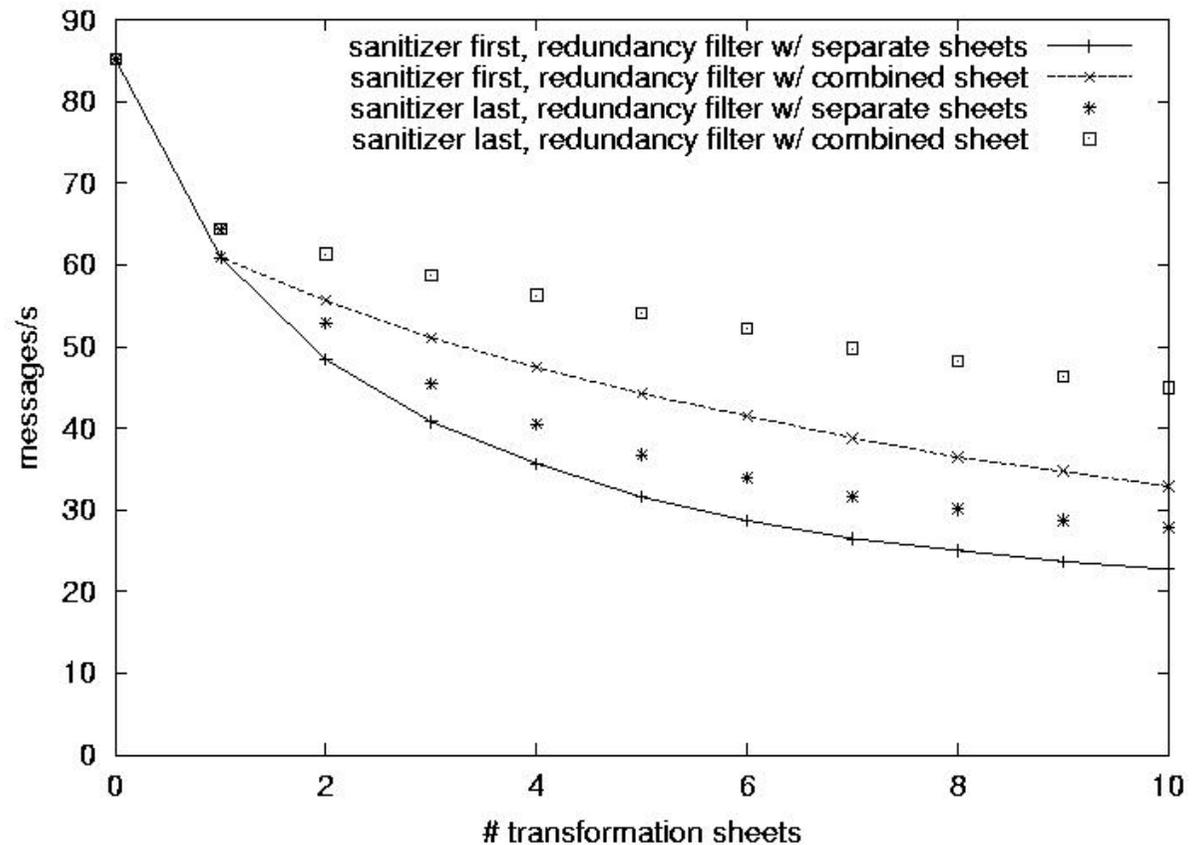
**Summary-Erzeugung**

**Summary-Update**

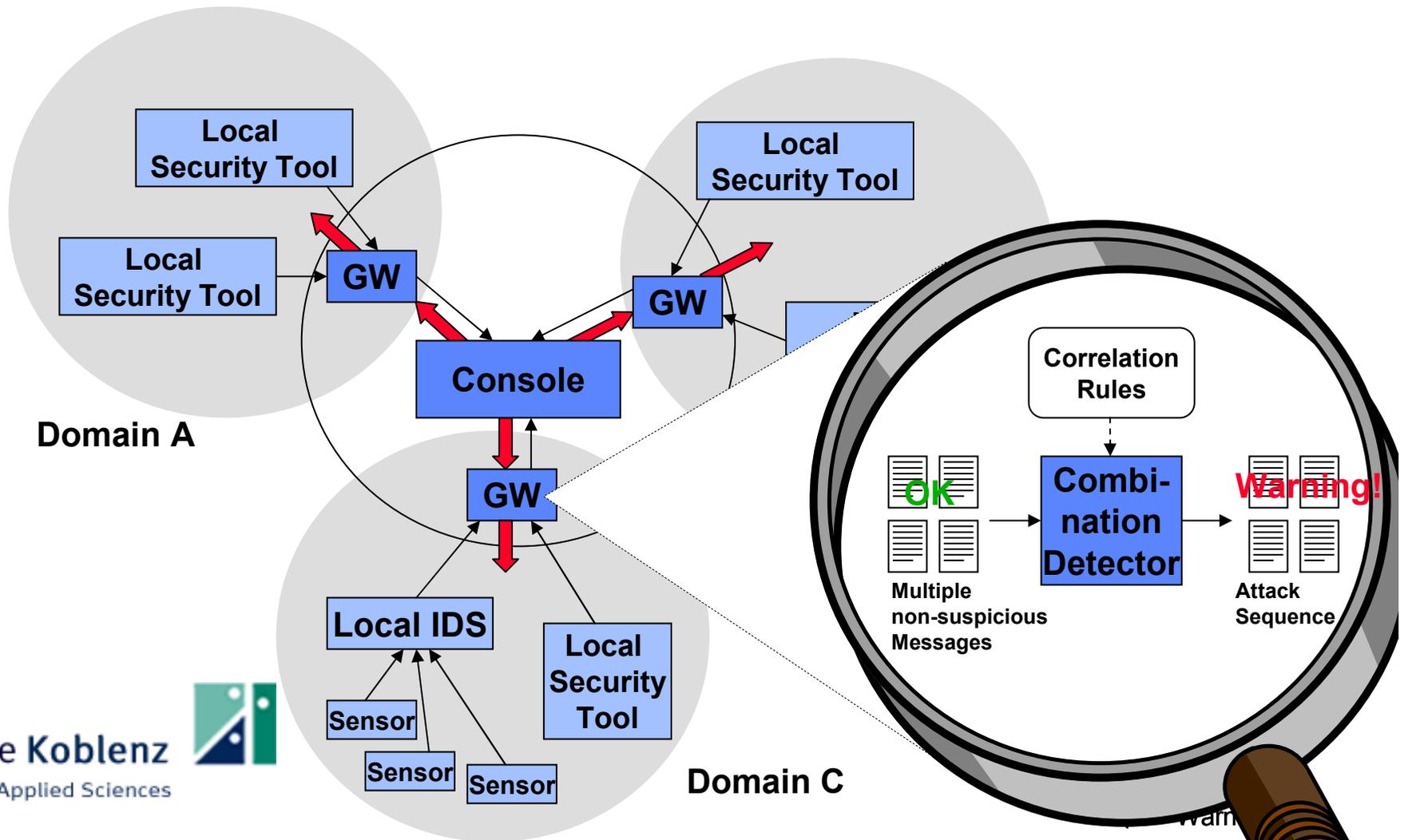
**Relatives Matching-Template**

# XSLT-Verarbeitung: Durchsatzmessungen

- Kombinierte Anwendung von Informationsbereiniger und Redundanzfilter im IWS-Gateway
- Hardware:  
PIII/1GHz/128MB
- $f(x) \approx 1 / (a + bx)$
- Anwendungsreihenfolge ist entscheidend
- Separate vs. kombinierte Transformationen



# Kombinationserkennung (1): Szenario

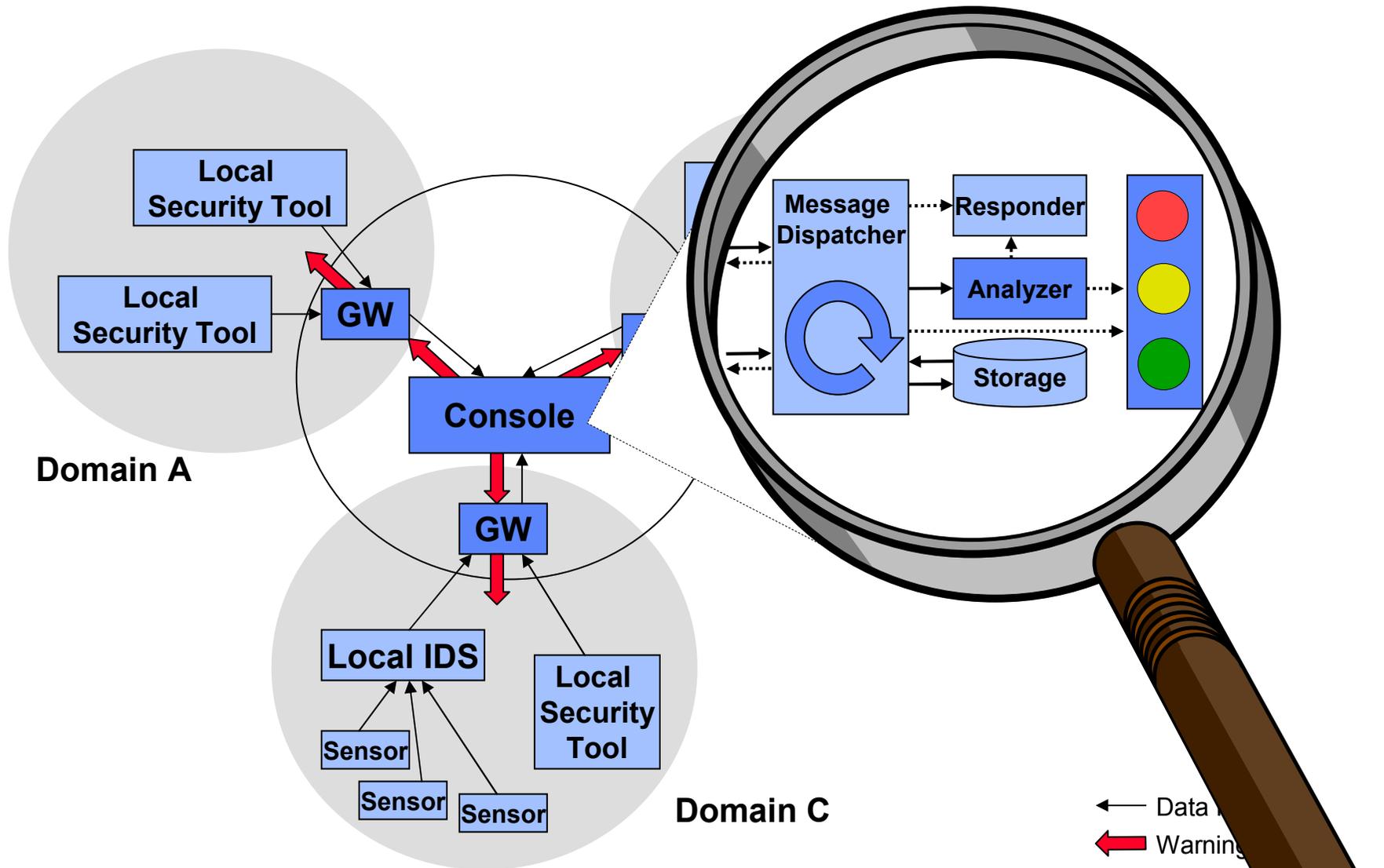


## Kombinationserkennung (2): Implementierung

---

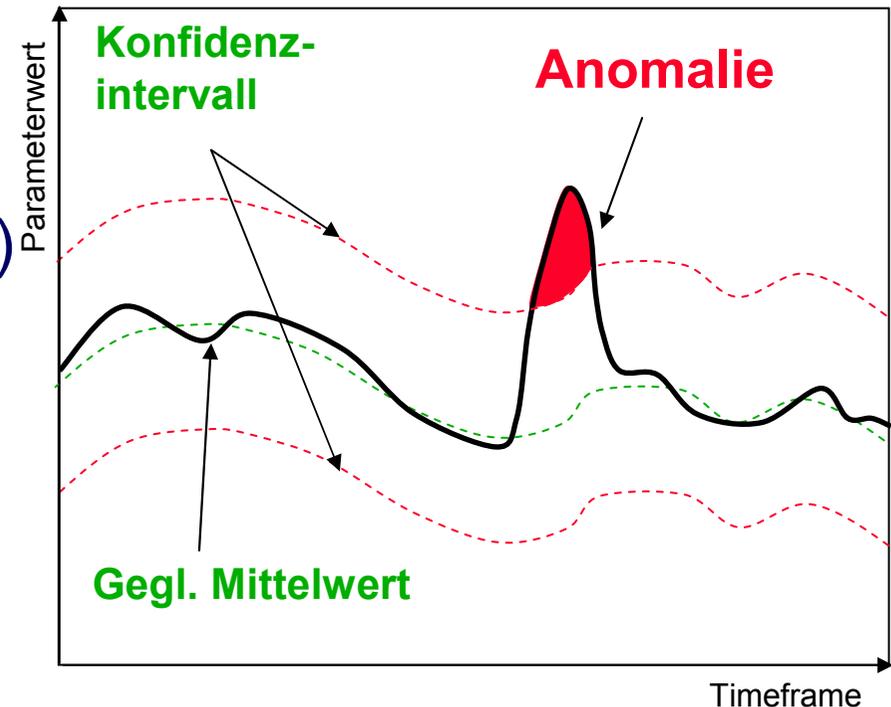
- $n:(n+1)$  -Transformation in der Online-Variante
- Extremer Speicheraufwand für naive Implementation
- Erweiterung des Redundanzfilters um Just-In-Time-  
Erzeugung von Templates (Matching-Stufen)
- Verschachtelte Spezifikationen mit  
Transformationsvorschriften für die  
Kombinationsmeldung am Ende
- Integration und Evaluation dauern noch an

# Anomalieerkennung (1): Szenario



# Anomalieerkennung (2): Flussparameteranalyse

- Beispiele für Messparameter pro  $\Delta t$ :
  - # Meldungen
  - # Bytes / Meldung
  - # unterschiedlicher Klassifikationen (Signaturen)
  - # unterschiedlicher Quell-/Zieladressen
  - # unterschiedlicher Analyseknotten
- Alarm, wenn Parameter Konfidenzintervall um Mittelwert verlässt
- **Nachteil:** Keine Betrachtung semantischer Information



# Anomalieerkennung (3): Quelle-Ziel-Graph

- Aufbau eines Quelle-Ziel-Graphen pro Timeframe



<IDMEF-Message ...

...

<Source>

<address> 192.22.2.44 </address>

...

</Source>

...

<Target>

<address> 191.72.30.2 </address>

...

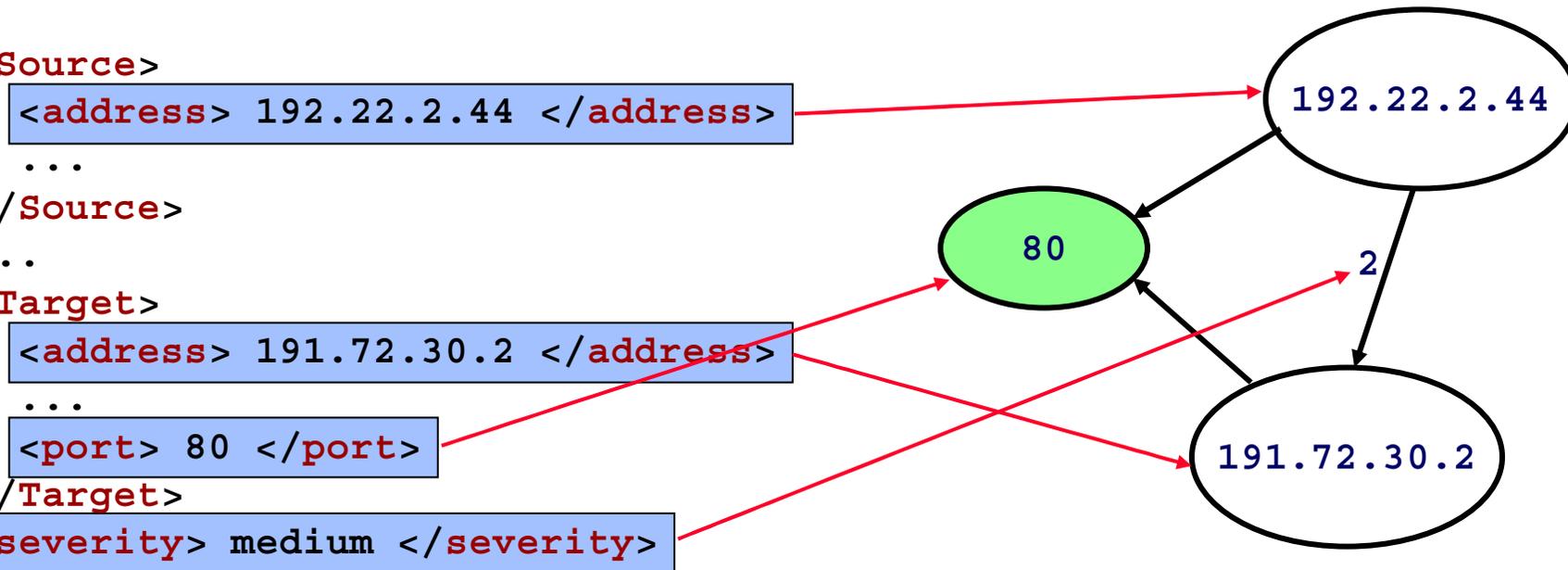
<port> 80 </port>

</Target>

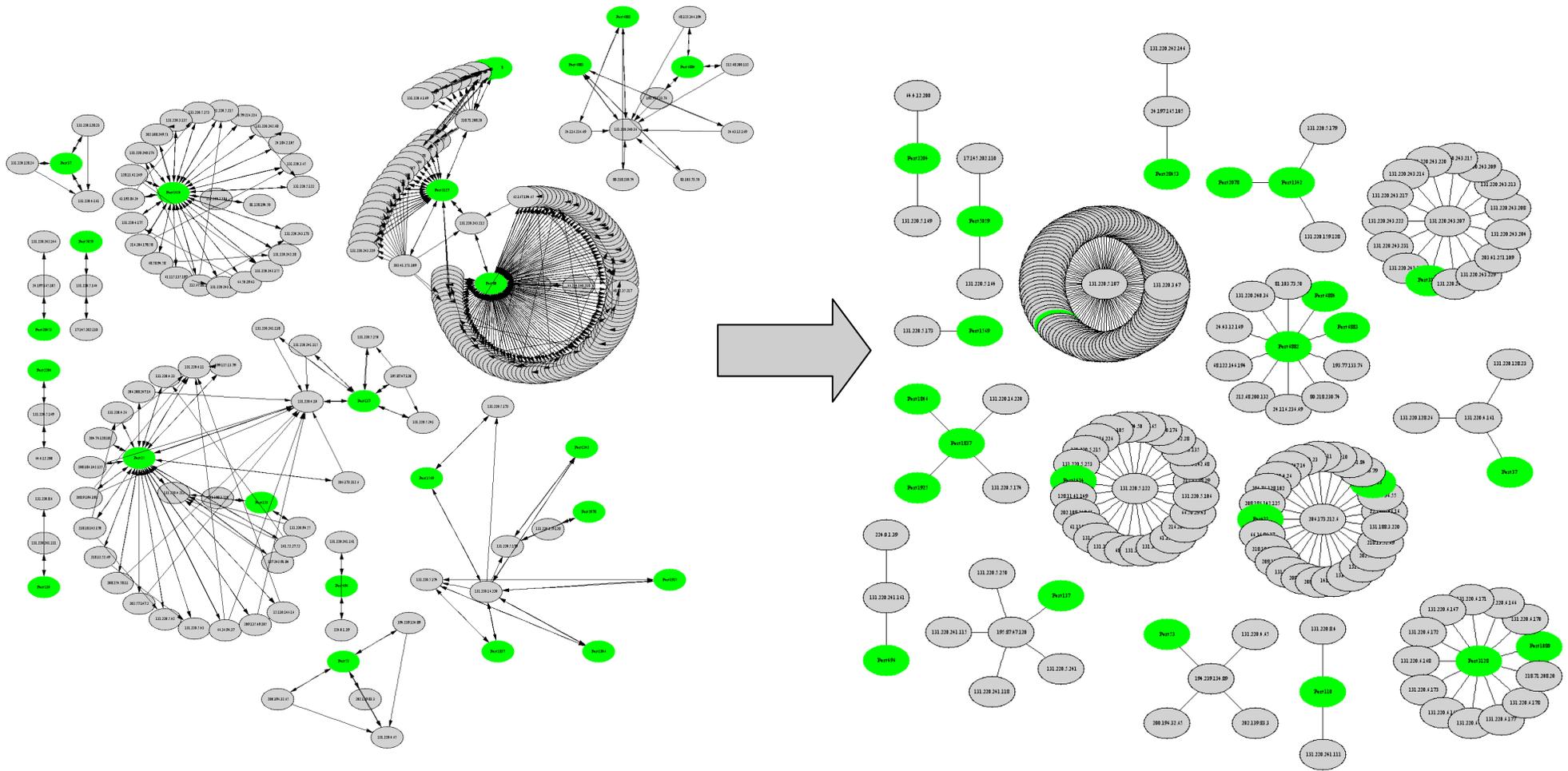
<severity> medium </severity>

...

</IDMEF-Message>



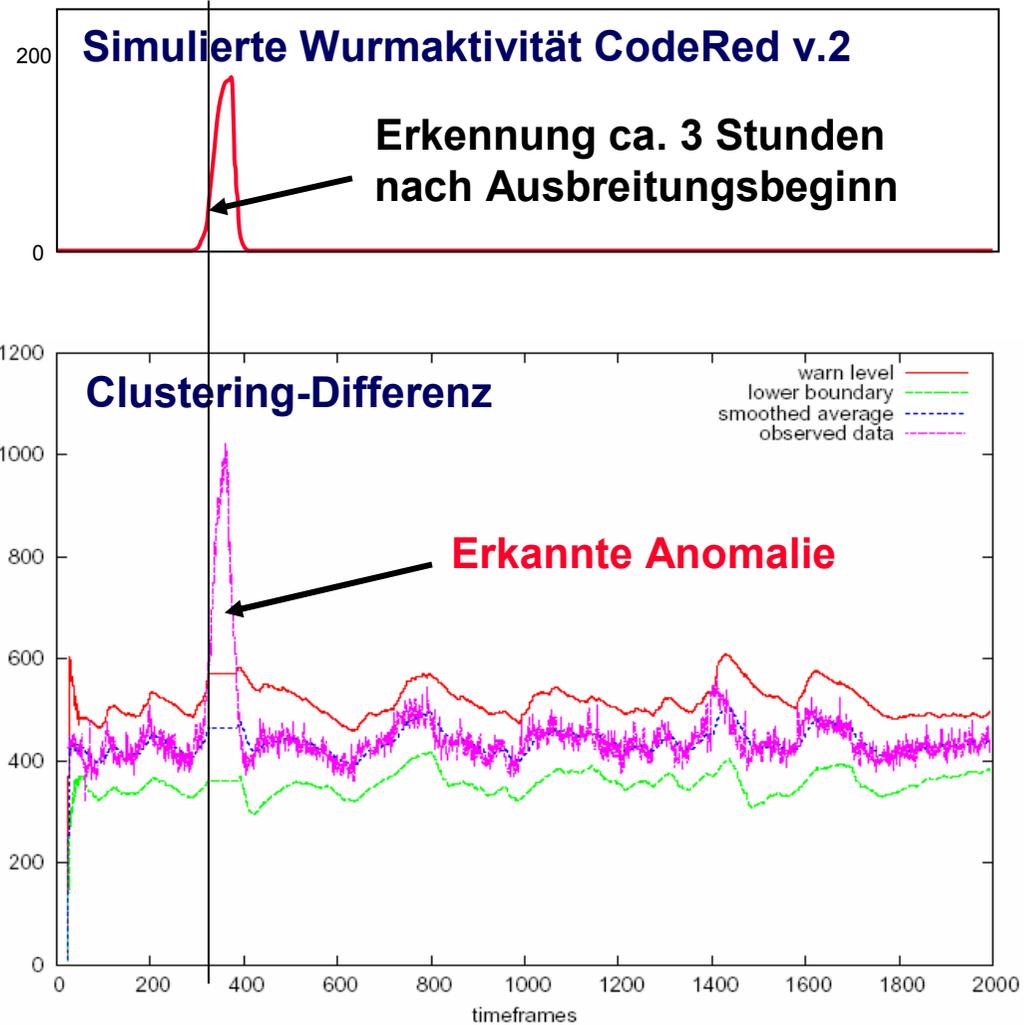
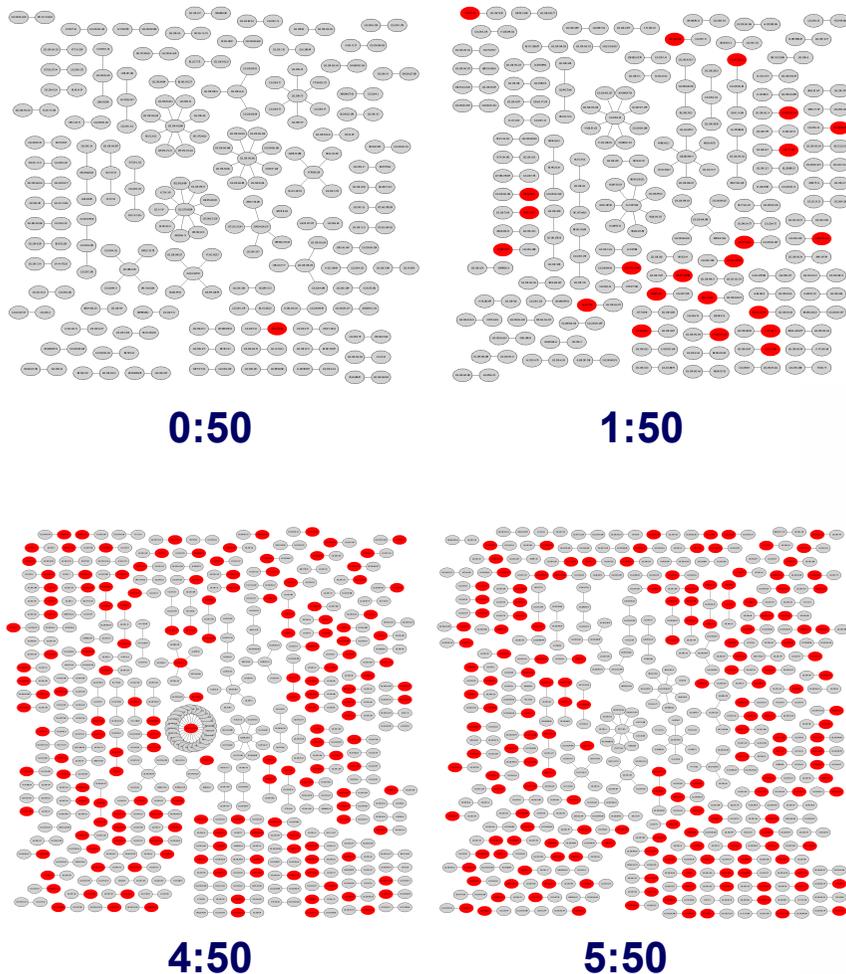
# Anomalieerkennung (4): Graph-Clustering



Quelle-Ziel-Graph mit Diensten/Ports

Cluster

# Anomalieerkennung (5): Clustering-Abweichung



# Zusammenfassung

---

- Meta-IDS als Architektur für IWS in Koalitionsumgebungen
- Vorverarbeitung von Ereignismeldungen durch XSLT
  - Informationsbereinigung
  - Redundanzfilterung
  - Online-Kombinationserkennung
- Anomalieerkennung im Ereignismeldungs-Datenmodell
  - Einfache Flussparameteranalyse
  - Clustering von Quelle-Ziel-Graphen
- XML/XSLT-basierte Ansätze leistungs- und ausbaufähig
- Anomalieerkennung erfolgreich in Simulation und Praxis

# Ausblick

---

- Vervollständigung der Integration, Stabilisierung
- Multi-Domain-Demonstrator
  
- Extraktion von Informationen über Anomalien
- Auswirkungen der Informationsbereinigung auf die Anomalieerkennung
- Formalisierung & Simulationen
  - Grenzen der Verfahren
  - Skalierbarkeit

# Kontakt

---

## Marko Jahnke

Post-  
anschrift: FGAN/FKIE  
Abt. Kommunikation  
Neuenahrer Str. 20  
D-53343 Wachtberg

Email: [jahnke@fgan.de](mailto:jahnke@fgan.de)

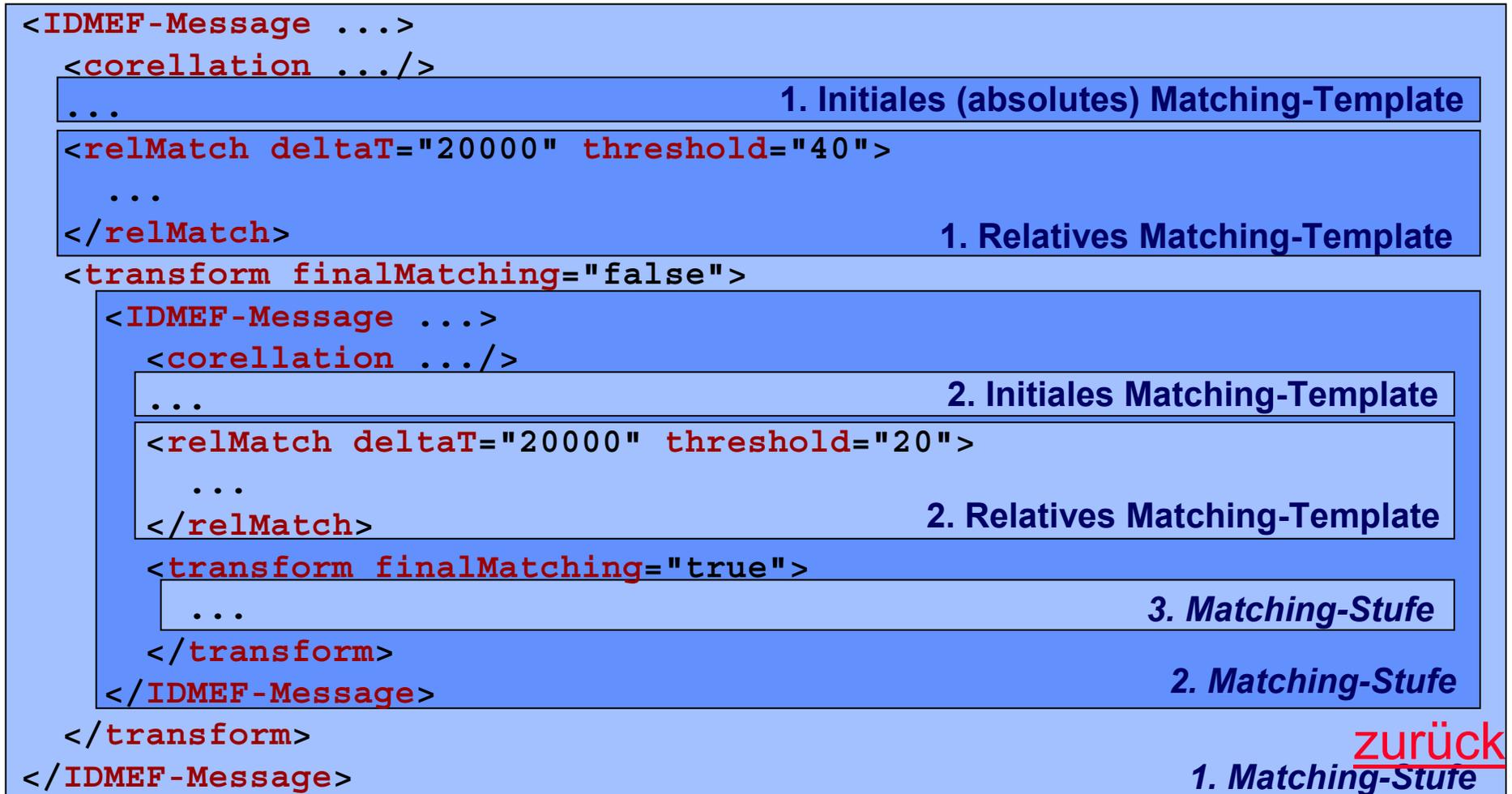
## Jens Tölle

Universität Bonn  
Institut für Informatik, Abt. IV  
Römerstr. 164  
D-53117 Bonn

[toelle@cs.bonn.edu](mailto:toelle@cs.bonn.edu)

# Kombinationserkennung (3): Implementierung

- *Just-In-Time*-Erzeugung von Templates durch Matching-Stufen



# GUI Screenshots (1): IWS Console

- Console GUI visualization
  - Messages w/ anonymized addresses (prefix 999.999)
  - Summary messages from redundancy filter (blue rows)

The screenshot displays the IWS Console GUI. At the top, there are menu items: GUI, Client, Events, EventList, Help. Below the menu is a status bar with fields for Console host (tick), Fallback host (none), IDMEFport (4441), Ulport (4443), Status (connected), and Threat level (3). A tree view on the left shows two clients: donald and tick, each with sensor and channel information. The main area is a table of events with columns: Time, Node, Info, Priority, Protocol, Sourceaddress, and Targetaddress. The table contains 15 rows of data, with some rows highlighted in blue (summary messages) and others in yellow (ICMP PING speedera). Below the table are controls for Events (50), max events (50), min priority (0), min warning (5), and buttons for Details, Delete, and Delete All. At the bottom, there is a log window showing messages from Mar 22, 2004, and a User command input field with a send button. The FGAN and KIE logos are visible in the bottom left and right corners of the GUI window.

Time >>	Node	Info	Priority	Protocol	Sourceaddress	Targetaddress
2004-03-16 15:56:45	track:LogsurferAdapter	Summary: CISCO access list hit	1	udp	148.223.35.119	999.999.7.1
2004-03-16 15:56:47	track:LogsurferAdapter	Summary: CISCO access list hit	1	tcp	4.61.252.56	999.999.91.75
2004-03-16 15:56:54	track:LogsurferAdapter	Summary: CISCO access list hit	1	udp	200.65.9.14	999.999.110.219
2004-03-16 15:56:56	track:LogsurferAdapter	Summary: CISCO access list hit	1	tcp	129.21.109.195	999.999.72.54
2004-03-16 15:57:03	track:LogsurferAdapter	Summary: CISCO access list hit	1	tcp	220.217.133.199	999.999.80.177
2004-03-16 15:57:07	track:LogsurferAdapter	Summary: CISCO access list hit	1	udp	68.237.198.175	999.999.91.188
2004-03-16 15:57:09	track:LogsurferAdapter	Summary: CISCO access list hit	1	tcp	68.23.39.182	999.999.32.252
2004-03-16 15:57:14	track:SnortAdapter	ICMP PING speedera	1	icmp	208.184.39.130	999.999.2.5
2004-03-16 15:57:14	track:SnortAdapter	ICMP PING speedera	1	icmp	202.160.241.130	999.999.2.5
2004-03-16 15:57:14	track:SnortAdapter	ICMP PING speedera	1	icmp	64.15.251.198	999.999.2.5
2004-03-16 15:57:14	track:SnortAdapter	ICMP PING speedera	1	icmp	64.37.246.2	999.999.2.5
2004-03-16 15:57:15	track:LogsurferAdapter	Summary: CISCO access list hit	1	udp	62.210.251.108	999.999.59.218
2004-03-16 15:57:19	track:LogsurferAdapter	Summary: CISCO access list hit	1	tcp	61.115.163.81	999.999.23.242

## GUI Screenshots (2): IWS Gateway

- Gateway GUI controls information sanitizing & filtering

