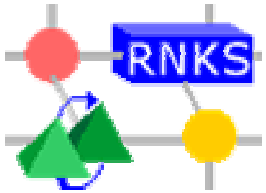
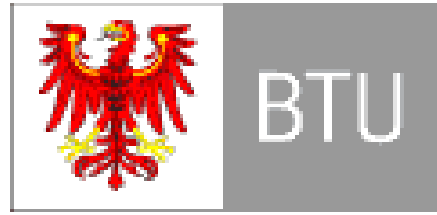



**Brandenburgische Technische Universität Cottbus**

**Lehrstuhl Rechnernetze und Kommunikationssysteme**




**Brandenburgische Technische Universität Cottbus**

# Intrusion Detection

## Möglichkeiten und Probleme einer wirksamen Erkennung sicherheitsgefährdender Aktionen im Internet

**Hartmut König**

{koenig@informatik.tu-cottbus.de}
http://www-rnks.informatik.tu-cottbus.de/~forschung

# Intrusion Detection

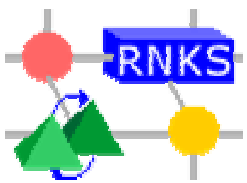
I. Aufgaben und Ziele

II. Wie funktioniert ein Intrusion Detection System ?

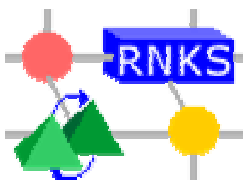
III. Probleme des Einsatzes von Intrusion Detection Systemen

IV. Eigene Forschungsarbeiten

V. Ausblick



# I.
# Aufgaben und Ziele

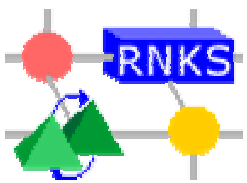


# Sicherheit in Rechnernetzen

Die zunehmende Verlagerung wirtschaftlicher und privater Prozesse auf das Internet erhöht die Abhängigkeit dieser Prozesse von IT-Systemen. Deren wachsende technologische Komplexität bildet die Grundlage für ein stetig steigendes **Bedrohungspotential**.

- Mechanismen erforderlich für
  - Schutz der Rechner und Netzinfrastrukturen
  - Erkennung und Abwehr von Einbrüchen
  - Sicherung der Überlebensfähigkeit von Netzen

☞ nach Möglichkeit automatische und echtzeitfähige Schutzmechanismen

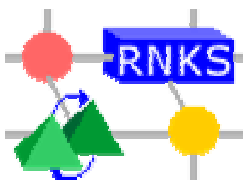


# Allgemeines Einbruchsproblem

Nach dem erfolgreichen Passieren der Authentifizierung durch ein abgefangenes oder erratenes Passwort ist der Angreifer zweifelsfrei als Nutzer identifiziert, d. h., das Betriebssystem akzeptiert seine Manipulationen.

- Der Angreifer hat Zugriff zu allen Unterverzeichnissen, Dateien und Programmen, die zu dem Account gehören.
- Der Angreifer hat Zugang zum Netz.

☞ Die Zugriffskontrolle kann dies nicht mehr verhindern !!!

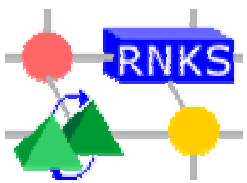


# Allgemeines Einbruchsproblem

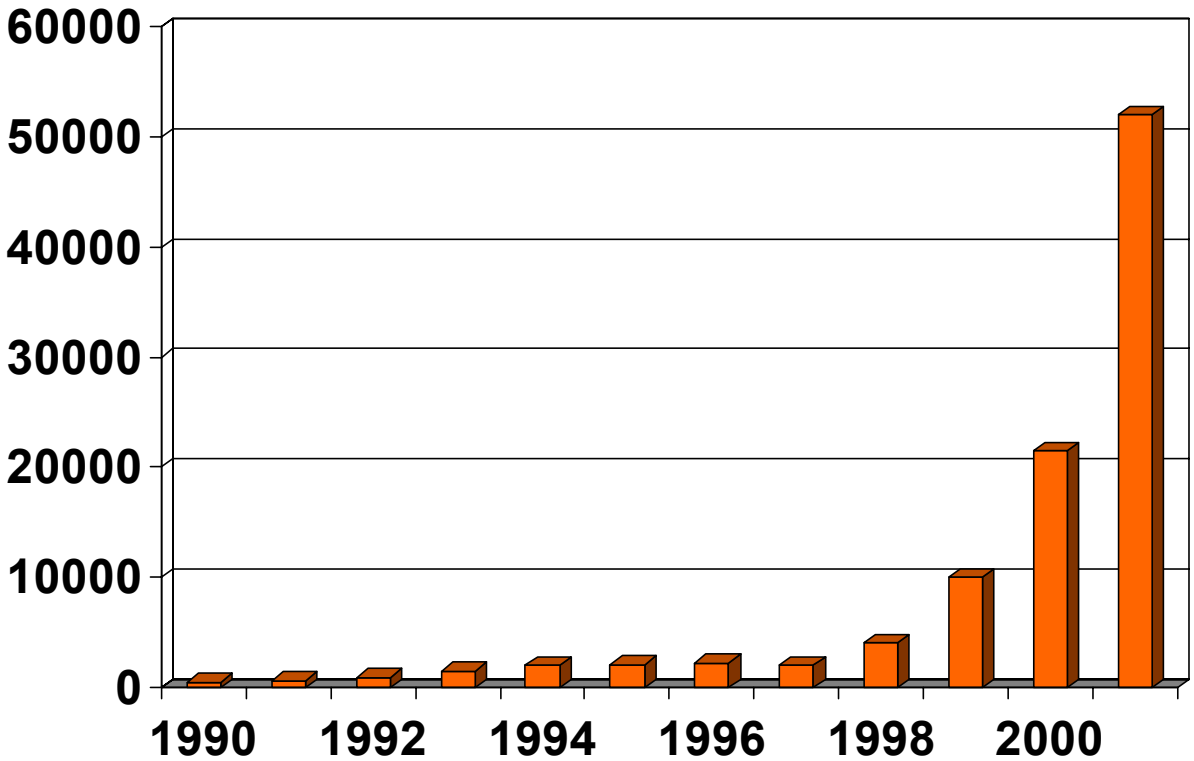
Nach dem erfolgreichen Passieren der Authentifizierung durch ein abgefangenes oder erratenes Passwort ist der Angreifer zweifelsfrei als Nutzer identifiziert, d. h., das Betriebssystem akzeptiert seine Manipulationen.

- Der Angreifer hat Zugriff zu allen Unterverzeichnissen, Dateien und Programmen, die zu dem Account gehören.
- Der Angreifer hat Zugang zum Netz.

☞ Die Zugriffskontrolle kann dies nicht mehr verhindern !!!

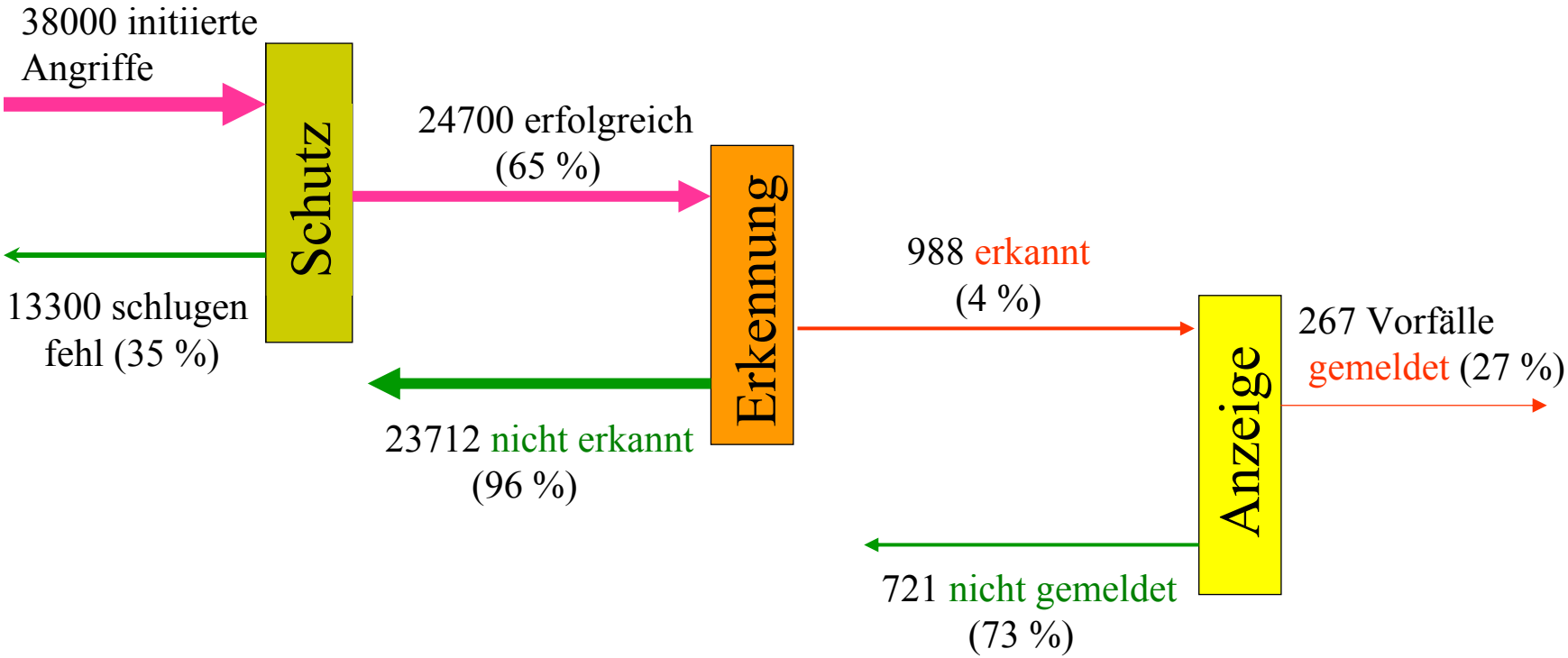


# Gemeldete sicherheitsrelevante Vorfälle beim CERT/CC




Amerikanisches Computer Emergency Response Team/Coordination Center:
http://www.cert.org/stats/cert_stats.html

# Testangriffe auf Rechner des US-Militärs durch die DISA




Quelle: Defense Information Systems Agency, GAO/AIMD-96-84, May 1996

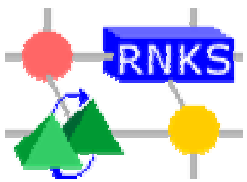
# Ursachen für die Verletzlichkeit von Rechnersystemen und Netzinfrastrukturen

Die Hauptursachen für die Verletzlichkeit von Rechnersystemen und Netzinfrastrukturen liegt in Unzulängigkeiten von Betriebssystemen, Datenbanken, Kommunikations-protokollen, Netzdiensten, und anderer Komponenten.

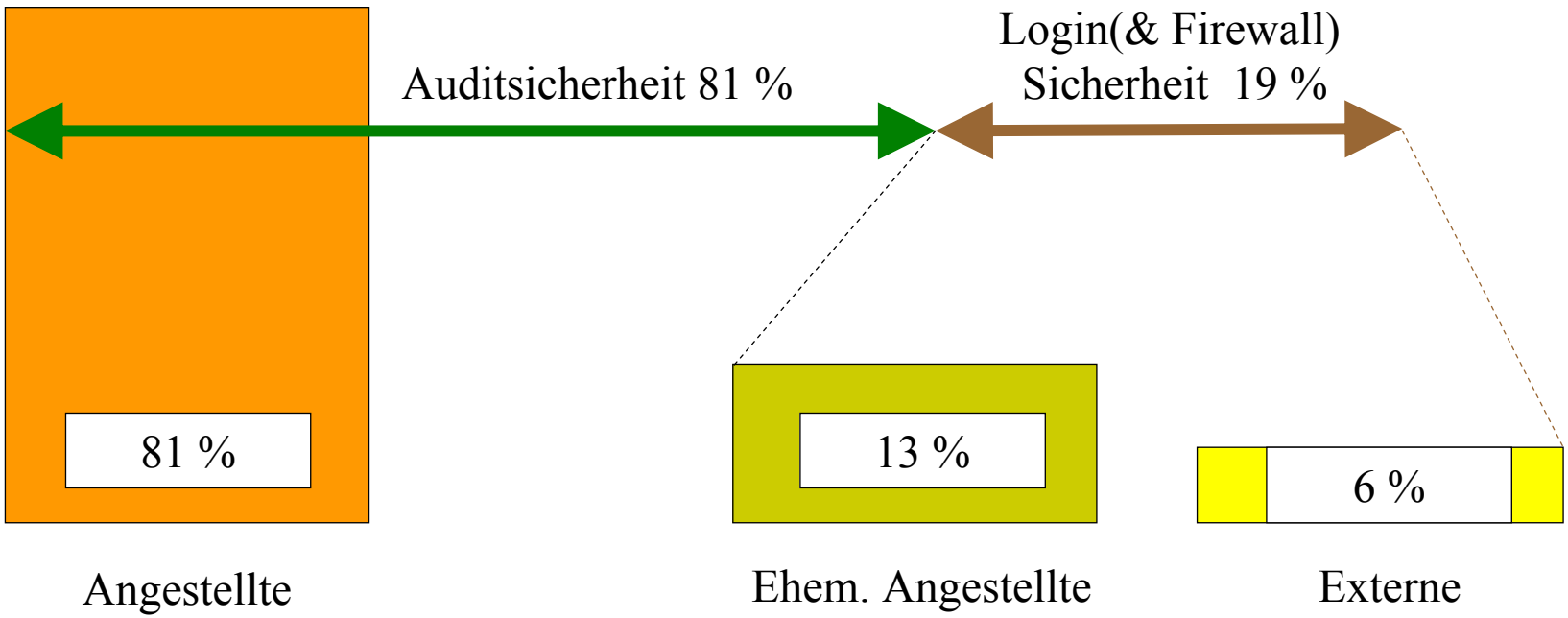
- **Gründe:** - Entwurfs- und Implementierungsschwachstellen
  - Spezifikationslücken
  - Implementierungsfehler
  - Konfigurationsfehler
  - Feature-Orientierung

  - Mißbrauch durch Menschen
    - Attacken von außen (Hacker, Spione, Terroristen)
    - Attacken von innen (Mitarbeiter)

  - Sorgloses Verhalten von Systemnutzern



# Insider-Problem



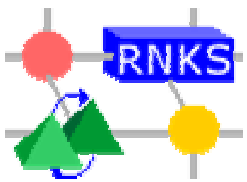

Source: Data Processing Management Assoc 1992

# Ziel des Intrusion Detection

Überwachung von Rechnersystemen und Netzinfrastrukturen zur Erkennung von Einbrüchen und Systemmissbrauch.

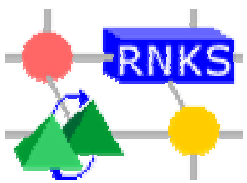
- Unterschied zu anderen Sicherheitsmechanismen
  - Die meistens Sicherheitsmechanismen sind präventiv.
  - Intrusion Detection setzt eine Sicherheitsverletzung voraus !!!

☞ **Primär ein Sicherheitsmechanismus zur Erkennung und Bekämpfung von erfolgten Sicherheitsverletzungen !!!**



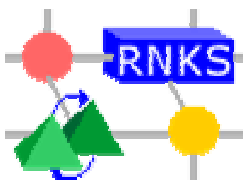
# Was kann man mit Intrusion Detection erreichen ?

- Überwachung der Einhaltung von Sicherheitspolitiken
  - Erkennen von Eindringlingen
  - Erkennen von System-Missbrauch
- Schadenserkennung und -begrenzung
  - *Response*-Mechanismen
- Erfahrungsgewinn
  - Verbesserung der Erkennungs- und Abwehrstrategien
- Abschreckungseffekt



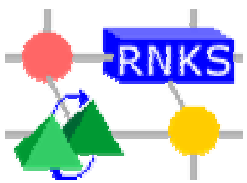
# Einsatzformen des Intrusion Detection (1)

- **Überwachter Bereich**
  - Host
    - Host-basierte IDS
    - Auswertung der protokollierten Auditdaten
  - Netz
    - Netz-basierte IDS
    - Analyse des Netzverkehrs
  - Hybride Systeme

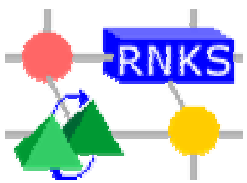


# Einsatzformen des Intrusion Detection (2)

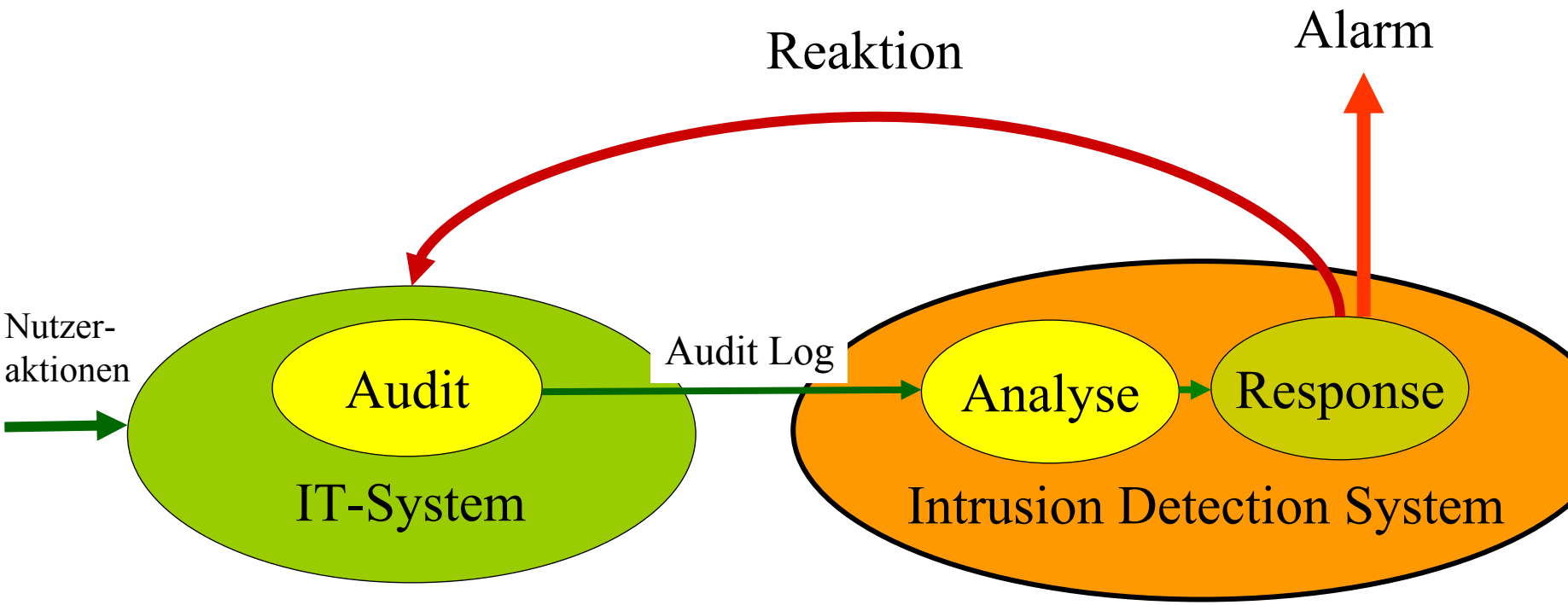
- **Auswertungszeitpunkt**
  - offline
    - *post mortem*-Analyse
    - Präventiv
  - online
    - Echtzeitanalyse
    - reaktiv



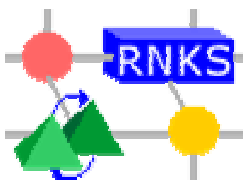
# II
# Wie funktioniert ein Intrusion Detection System ?



# Prinzip des Intrusion Detection



# II.1
# Audit



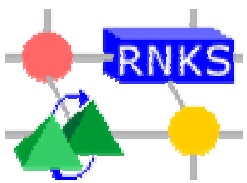
# Audit

- elementare Sicherheitsfunktion, die sicherheitsrelevante system- und netzinterne Aktionen aufzeichnet.
- Grundlage des Intrusion Detection
  - Aufzeichnung in Audit Records / Files
  - „Überwachungskamera“ der Rechnersysteme und –netze
- Audit-Daten geben detailiert Auskunft darüber

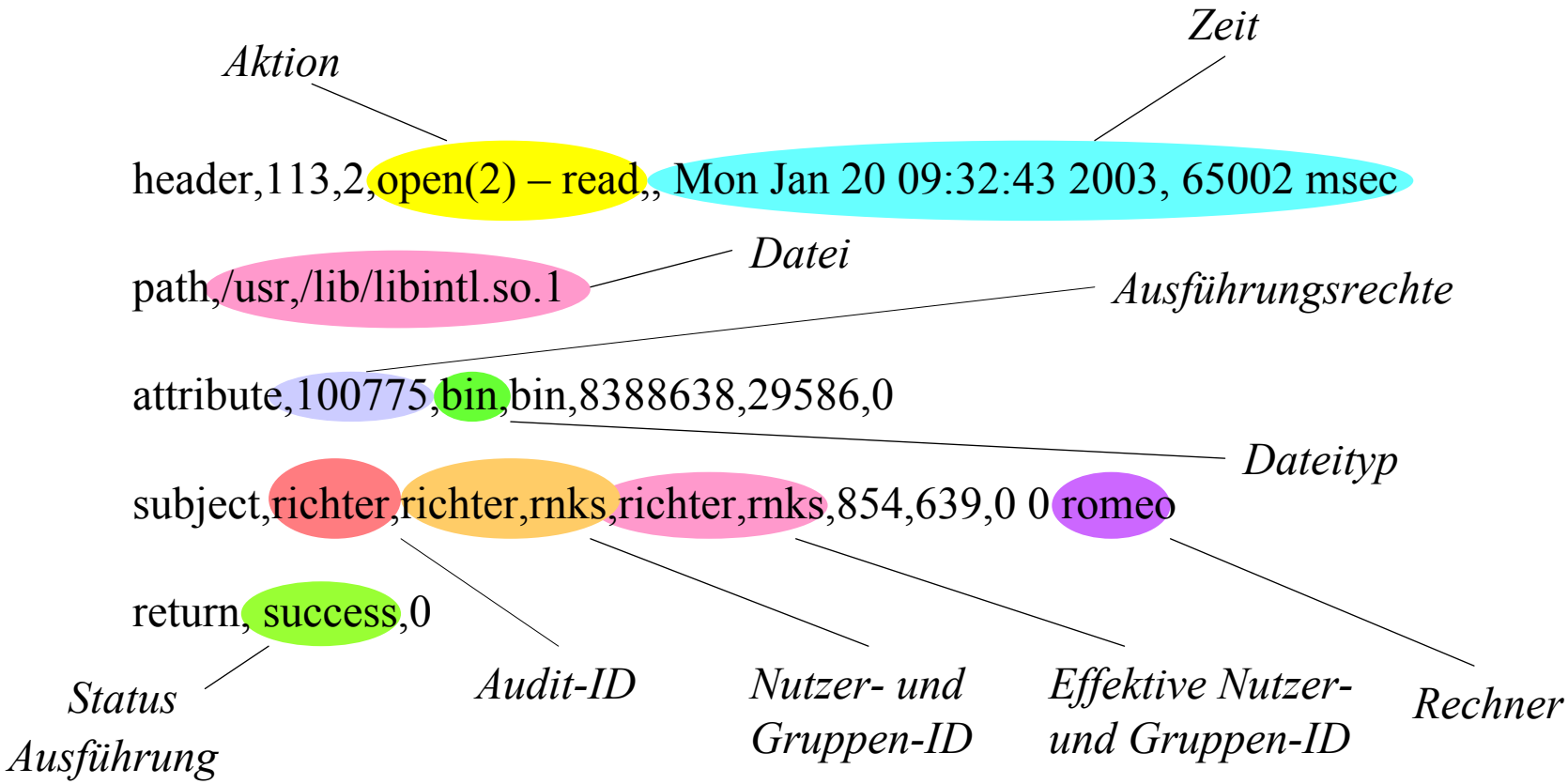
  **WER** hat

  **WANN**, **WO** und **WIE**

  auf **WELCHE** und **WESSEN** Ressourcen zugegriffen.

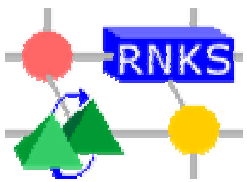


# Beispiel: Solaris Auditdaten

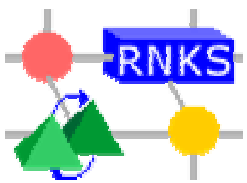


# Audit-Arten

- **Betriebssystem-Audit**
  - Funktion des Betriebssystems
  - *Ereignisse*: Systemaktivitäten
- **Netz-Audit**
  - Netzmonitore
  - *Ereignisse*: Protokolldateneinheiten
- **Applikations-Audit**
  - für die Anwendung zu programmieren
  - *Ereignisse*: applikationsspezifische, sicherheitsrelevante Aktivitäten



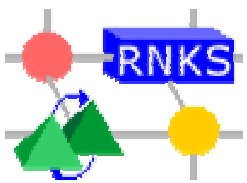
# II.2
# Intrusion Detection Systeme



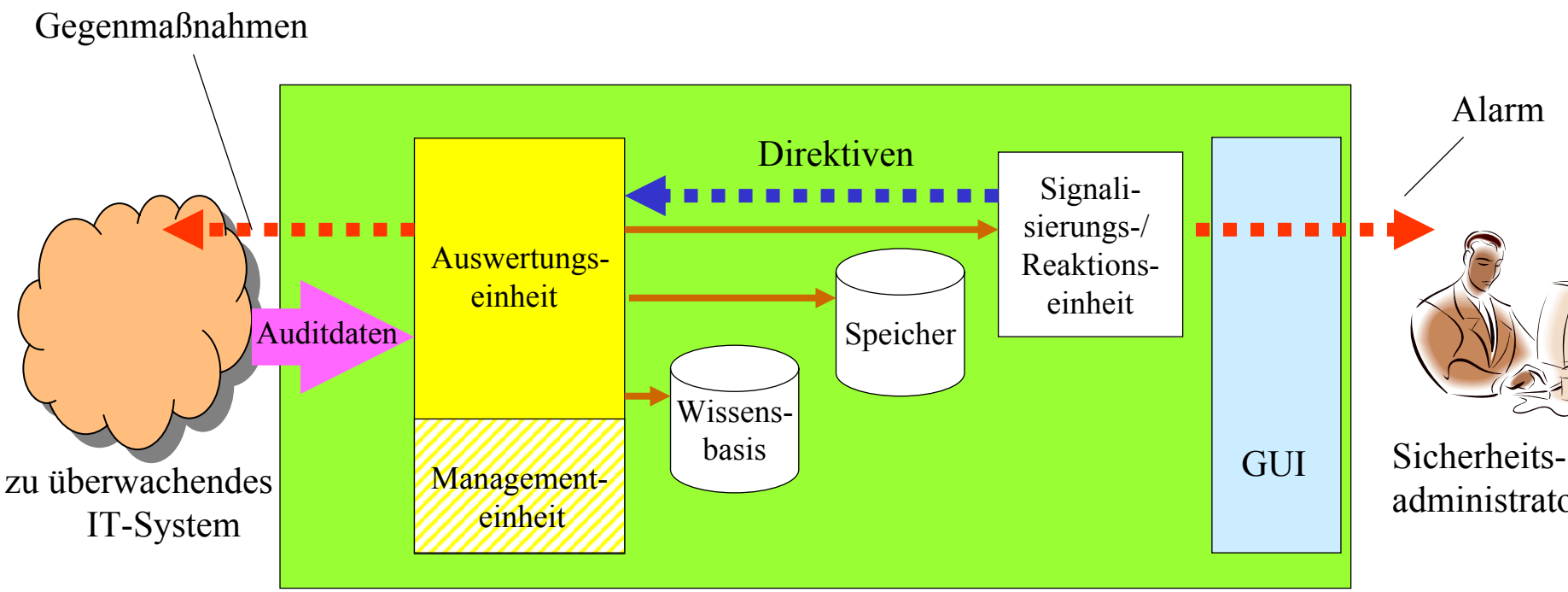
# Intrusion Detection-Systeme (IDS)

Intrusion Detection Systeme dienen der Erkennung von Angriffen und System-Missbrauch.

- **Aufgaben:**
  - Aufzeichnung aller sicherheitsrelevanten Aktionen des überwachten Systems (Audit)
  - Vorverarbeitung und Management der aufgezeichneten Daten
  - Automatisierte Analyse dieser Daten
  - Anzeige von Attacken und eventuell Einleitung von Gegenmaßnahmen zur Abwehr erkannter Angriff

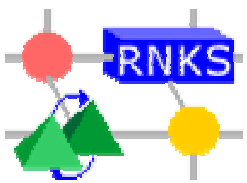


# Prinzipieller Aufbau eines Intrusion Detection Systems



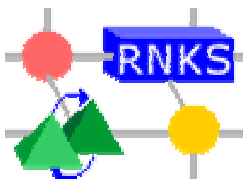
# Analyse-Ansätze

- Anomalie-Erkennung (*anomaly detection*)
- Signaturanalyse (*misuse detection*)
- Policy-basierte Erkennung (*policy based detection*)



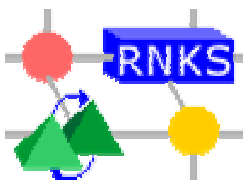
# Anomalie-Erkennung (1)

- Erkennen von signifikant abweichendem Nutzerverhalten
- **Zugrundeliegende Annahmen**:
  - Nutzer haben Gewohnheiten in der Systembenutzung
    - bestimmte Nutzungszeit
    - Häufigkeit der Nutzung
    - Umfang des Dateizugangs
    - Aufruf bestimmter Programme/Dateien
  - “Normales“ nutzertypisches Verhalten ist statistisch beschreibba
    - verhaltensbasierte Authentifikation
- **Auswertungsbasis**: Referenzprofile des Nutzers
  - ein Angriffsszenarium muss nicht explizit vorgegeben werden



# Anomalie-Erkennung (2)

- **Auswertungsverfahren**: z. B. Neuronale Netze
  - lernfähig
  - k*ompliziert*: Korrelation zwischen Parametern
- **Probleme**:
  - Erfassung nutzerspezifischer Verhaltensmerkmale
    - berufsspezifische und persönliche Eigenheiten
  - ständige Aktualisierung erforderlich
  - hohe Zahl von Fehlalarmen
  - bei zufälliger Übereinstimmung Angreifer ↔ Referenzprofil keine Identifikation möglich



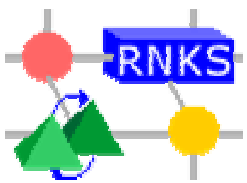
# Signaturanalyse (1)

- Erkennen von Einbrüchen auf der Grundlage bekannter und hypothetischer Angriffsszenarien
  - Angriffssignaturen

- **Voraussetzung**:
  - Kenntnis der Schritte eines Angriffs und der notwendigen Kontextbedingungen

☞ Das Leistungsvermögen der Signaturanalyse hängt vom Umfang, der Qualität und der Aktualität der zugrundeliegenden Angriffsmodelle ab !!!

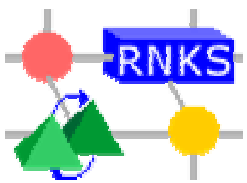
↳ “Hase-Igel-Spiel“



# Signaturanalyse (2)

- **Probleme**:
  - Erfassen von Attacken
    - Schwachstellenanalysen
    - Auswertung von Einbrüchen
  - Erfassen der Signaturen
    - Identifikation der Manifestation einer Attacke in den Audit-Records
    - Abspeicherung in Form von Regeln in einem Expertensystem
  - geeignete Beschreibung der Attacken

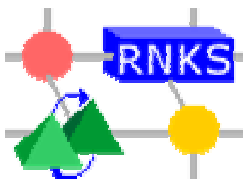
☞ **empirisches Vorgehen !!!**



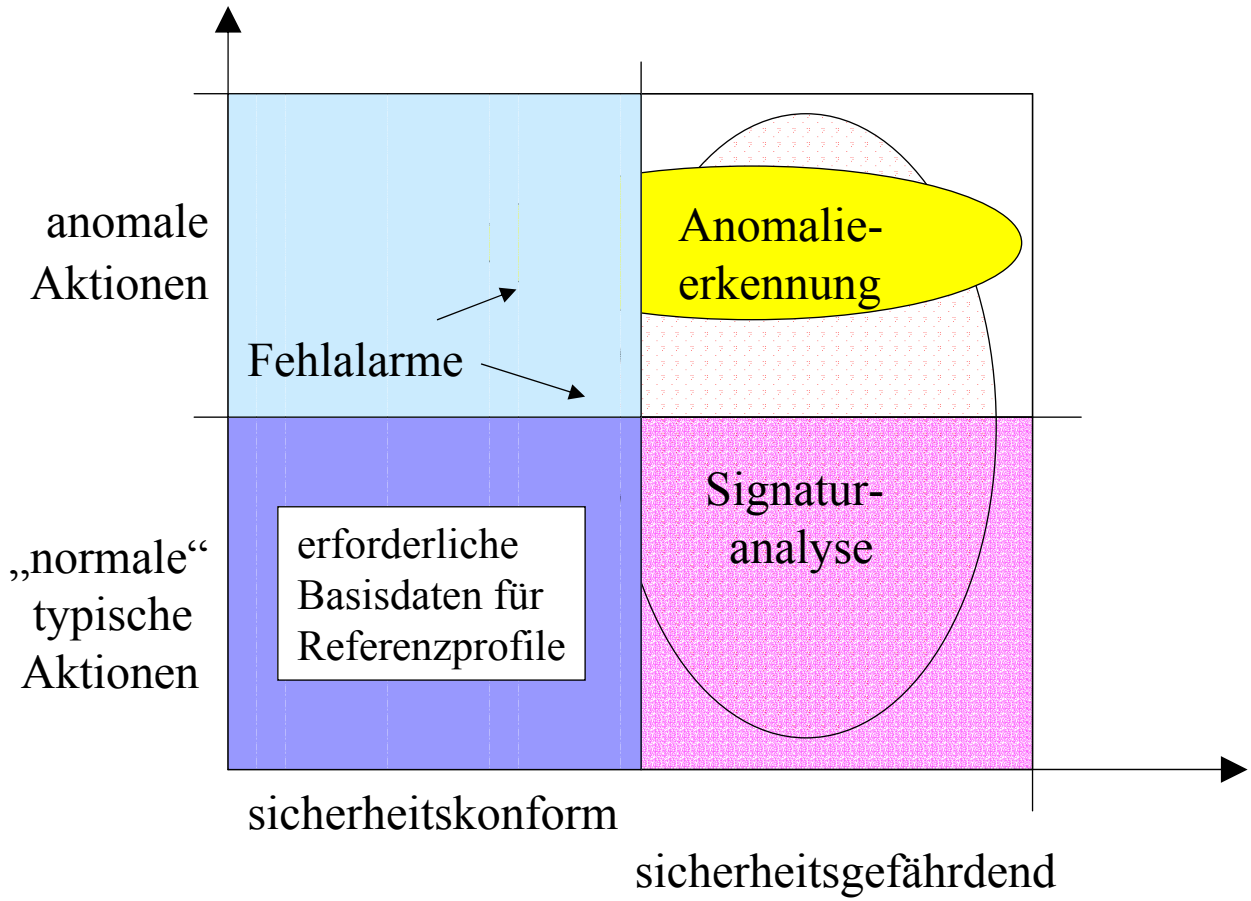
# Anomalie-Erkennung vs. Signaturanalyse

- Anomalie-Erkennung
  - theoretisch anspruchsvoll
  - aufwendig in der Umsetzung
  - begrenzte Wirksamkeit
- Signaturanalyse
  - vergleichsweise geringerer Aufwand
  - wirkungsvoller in der Erkennung
  - keine Erkennung anomalen Verhaltens

☞ **komplementäre Techniken !!!**



# Verhaltensebenen des Intrusion Detection

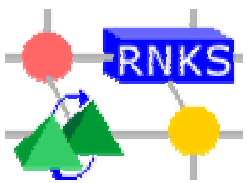



entnommen aus: Sobirey, M.: Datenschutzorientiertes Intrusion Detection. Vieweg-Verlag, 1999.

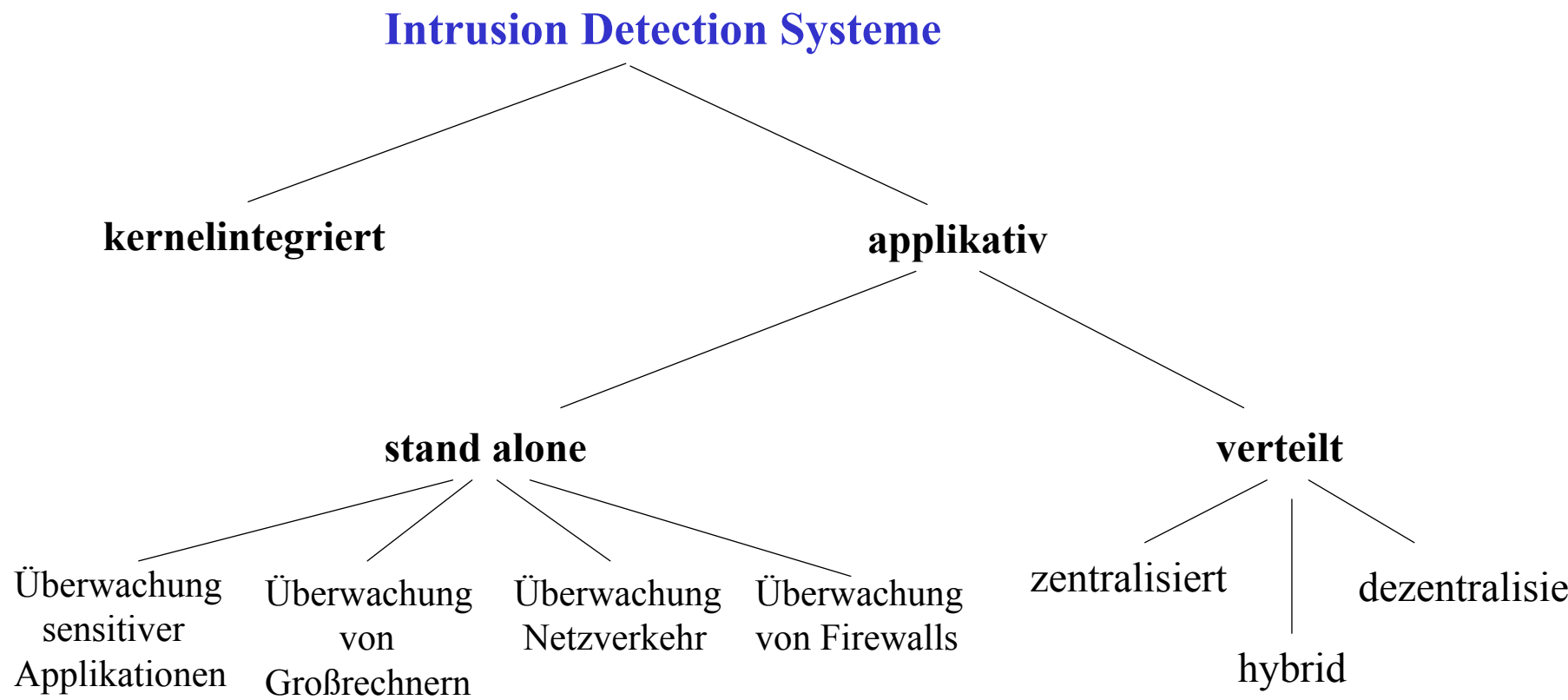
# Policy-basierte Erkennung

- Erkennen von Einbrüchen auf der Grundlage der Spezifikation erlaubten Verhaltens (*default permit*)
  - Vorgabe durch Sicherheitspolitik für das System
  - spezifikations-basierte Erkennung
  - Erkennungsprinzip der Signaturerkennung (*default deny*)

☞ bislang weniger genutzt



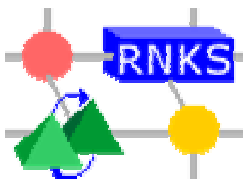
# Klassifikation der Intrusion Detection Systeme



# Beispiele für IDS

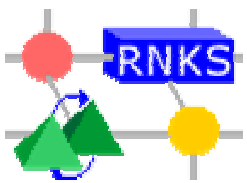
| Name of system | Publ. year | Detection principle | Time of detection | Granularity | Audit source | Type of response | Data-processing | Data-collection | Security | Inter-oper. |
|---|---|---|---|---|---|---|---|---|---|---|
| Haystack [51] | 1988 | hybrid | non-real | batch | host | passive | centralised | centralised | low | low |
| MIDAS [50] | 1988 | hybrid | real | continuous | host | passive | centralised | centralised | low | low |
| IDES [41] | 1988 | anomaly[a] | real | continuous | host | passive | centralised | distributed | low | low |
| W&S [54] | 1989 | anomaly | real | continuous | host | passive | centralised | centralised | low | low[b] |
| Comp-Watch [11] | 1990 | anomaly[c] | non-real | batch | host | passive | centralised | centralised | low | low |
| NSM [21] | 1990 | hybrid[d] | real | continuous | network[e] | passive | centralised | centralised[f] | low | low[g] |
| NADIR [25] | 1991 | policy | non-real | continuous | host[h] | passive | centralised | distributed | low | low |
| Hyperview [7] | 1992 | hybrid | real | continuous | host | passive | centralised | centralised | low | low |
| DIDS [52] | 1992 | hybrid | real | continuous | both[i] | passive | distributed | distributed | low | low[j] |
| ASAX [18] | 1992 | policy | real[k] | continuous[l] | host | passive | centralised | centralised | low | higher[m] |
| USTAT [24] | 1993 | policy | real | continuous | host | passive | centralised | centralised | low | low[n] |
| DPEM [30] | 1994 | policy[o] | real | batch | host | passive | distributed | distributed | low | low |
| IDIOT [34] | 1994 | policy | real[p] | continuous | host | passive | centralised | centralised | low | higher |
| NIDES [1] | 1995 | hybrid | real[q] | continuous | host[r] | passive | centralised | distributed | low[s] | higher[t] |
| GrIDS [53] | 1996 | hybrid[u] | non-real | batch | both[v] | passive | distributed | distributed | low | low |
| CSM [58] | 1996 | policy | real | continuous | host | active[w] | distributed | distributed | low | low |
| Janus [17] | 1996 | policy | real | continous | host | active[x] | centralised | centralised | low | low |
| JiNao [15] | 1997 | hybrid | real | batch | "host"[y] | passive | distributed | distributed | low | low |
| EMERALD [47] | 1997 | hybrid | real | continuous | both | active | distributed | distributed | moderate | high |
| Bro [46] | 1998 | policy | real | continuous | network | passive | centralised | centralised[z] | higher | low |

# III.
# Probleme des Einsatzes von Intrusion Detection Systemen



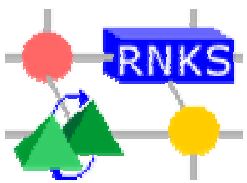
# Anforderungen an Intrusion Detection Systeme

- leichte Integrierbarkeit in IT-Systeme
- einfache Konfigurierbarkeit und Wartung
- autonome und fehlertolerante Arbeitsweise
- Geringe Inanspruchnahme von Systemressourcen
- Minimierung von Fehlalarmen und nicht erkannten Sicherheitsverletzungen
- Inhärente Selbstschutzmechanismen



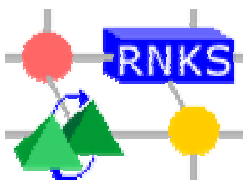
# Probleme des Einsatzes von Intrusion Detection Systemen

- Hohes Aufkommen an Auditdaten
- Datenschutz
- Beschränkte Auswertungseffizienz
- Fehlalarme
- *Response*-Mechanismen
- Selbstschutz
- Kooperation von Intrusion Detection Systemen
- Hoher Wartungsaufwand



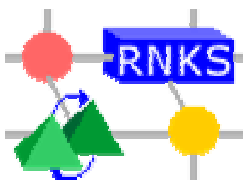
# Problem: Auditdaten

- **Umfang**
  - sehr hohes Datenaufkommen
  - Ort und Dauer der Aufbewahrung
  - automatische Auswertung anzustreben
- **Schutz**
  - Auditdaten können Ziel von Angreifern sein
  - Vernichtung / Verfälschung
- **Informationsgehalt**
  - welche Informationen sind relevant
  - oftmals unzureichend
  - keine Attackenerkennung  Fehlalarme



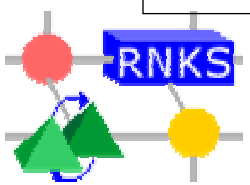
# Experiment: Aufkommen an Auditdaten (1)

- **Ziel**: Beobachtung von 4 unterschiedlich konfigurierten SUN-Workstations
- **Zeitdauer**: 4 Tage
- **Protokollierte Aktionen***:* - Systemanmeldungen (*logins*)
  - Lese- und Schreibzugriffe
  - Erzeugen und Löschen von Dateien
  - Modifikation von File-Attributen
  - prozessrelevante/administrative Aktionen
  - Starten von Prozessen



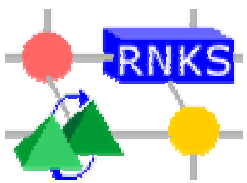
# Experiment: Aufkommen an Auditdaten (2)

| | Rechner 1 | Rechner 2 | Rechner 3 | Rechner 4 |
|---|---|---|---|---|
| SparcStation | 20 | 20 | 5 | 5 |
| Konfiguration | stand alone<br>NFS-, Mail-, WWW-, Print- Install.- u. Backup-Server | dataless<br>NFS-Client | stand alone<br>NFS-Client | dataless<br>NFS-Client |
| aktive Nutzer | 2-3 | 4-5 | 1 | 1 |
| Auditdaten (Mbyte) in 4 Tagen | 150 | 108 | 18 | 19 |
| in 24 h | 37,5 | 27 | 4,5 | 4,75 |
| in 1 h | 1,56 | 1,12 | 0,19 | 0,20 |



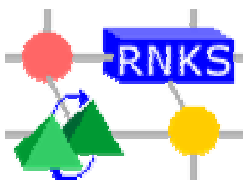
# Lösungsansätze

- Reduzierung des Datenaufkommens
  - Beschränkung auf die Protokollierung sicherheitsrelevanter Aktionen
- Beschränkung der Überwachungsdomäne
  - HONA-Ansatz
- Verteilte Erfassung und Bearbeitung
  - Vorverarbeitung in den systemnahen Überwachungseinheiten

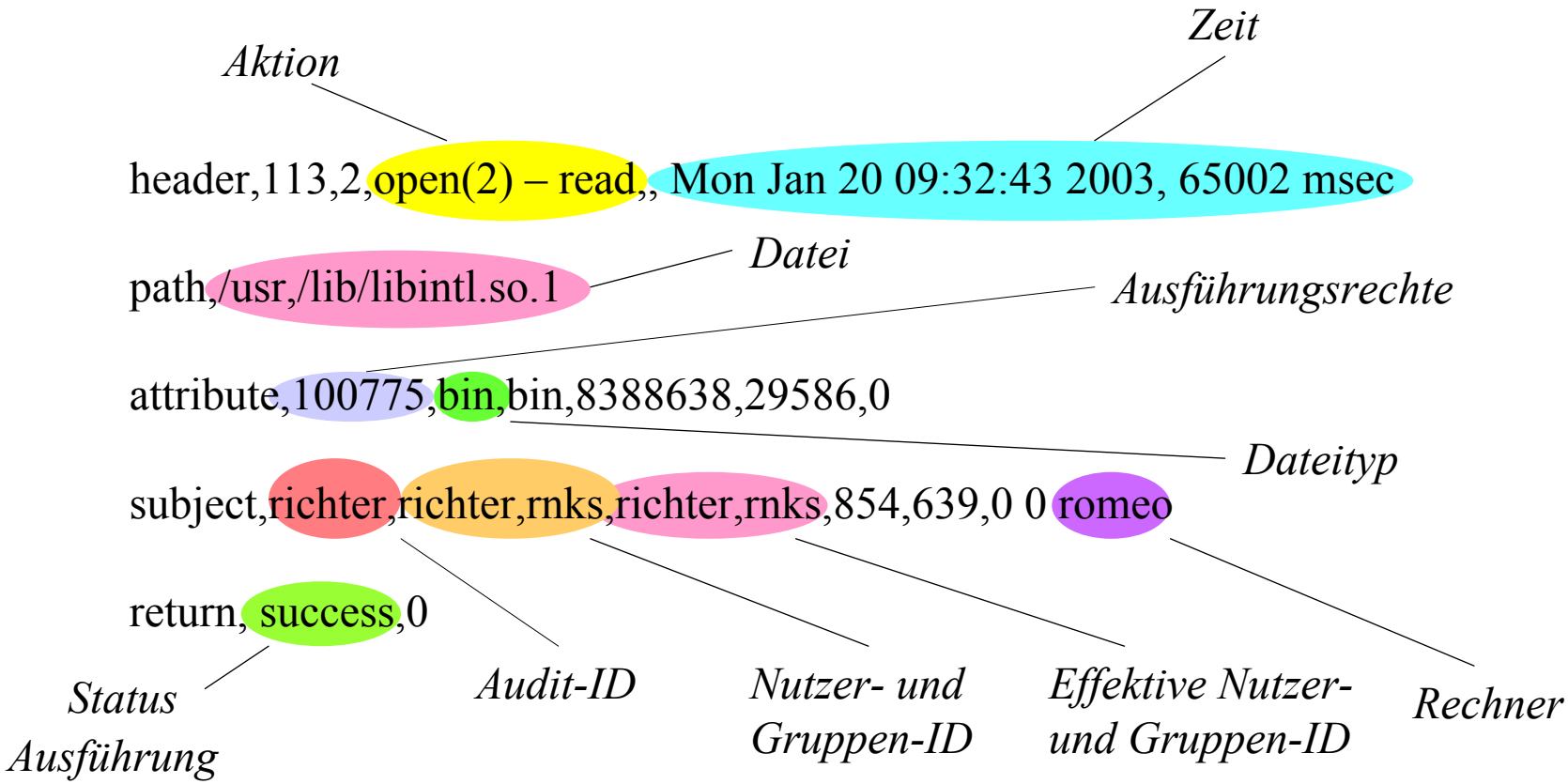


# Probleme des Einsatzes von Intrusion Detection Systemen

- Hohes Aufkommen an Auditdaten
- Datenschutz
- Beschränkte Auswertungseffizienz
- Fehlalarme
- *Response*-Mechanismen
- Selbstschutz
- Kooperation von Intrusion Detection Systemen
- Hoher Wartungsaufwand



# Beispiel: Solaris Auditdaten

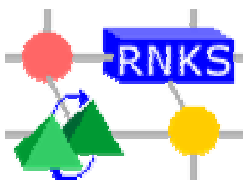


# Nutzeridentifizierende Einträge in Auditdaten

- **Konkrete Nutzer-IDs:**
  Audit-ID, reale Nutzer- und Gruppen-IDs,
  effektive Nutzer- und Gruppen-IDs

- **Bedingt nutzeridentifizierende Daten:**
  Namen der Home- und Unterverzeichnisse,
  Datei- und Programmnamen

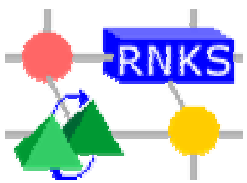
- **Minimal nutzeridentifizierende Daten:**
  Host-Type, Aktion + Zeit + Status,
  Zugriffsrechte + Aktion + Status



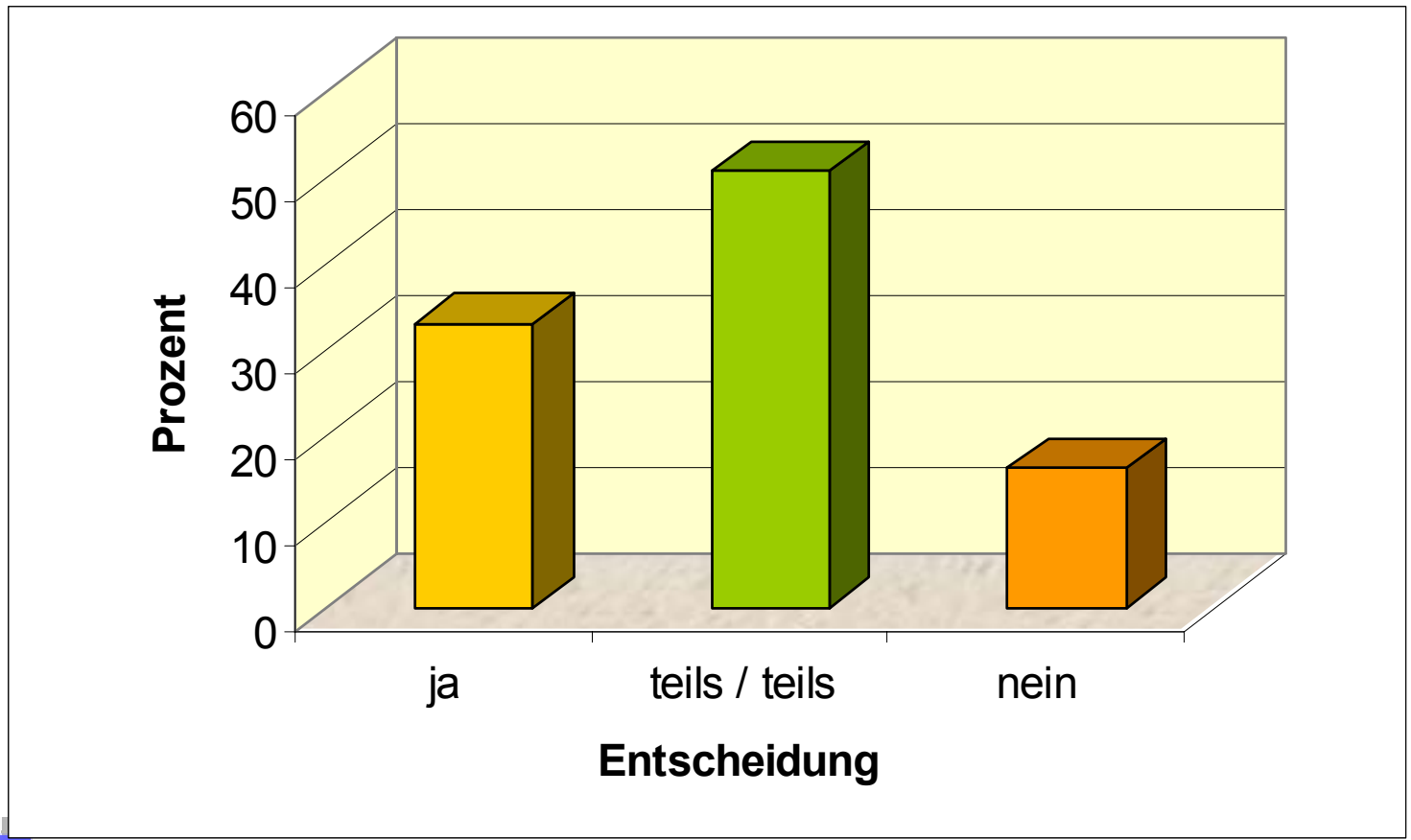
# Befindlichkeiten überwachter Nutzer

- **Umfrage von Dr. Michael Sobirey**
- **Befragte Organisationen**:

  Universität Hamburg, Universität Hildesheim, TU Berlin, TU Magdeburg, FAW Ulm, debis AG Bremen, Siemens-Nixdorf München
  - 155 Fragenbögen / 148 Antworten
- **Fragen:** - Ist der Einsatz von IDS prinizipiell notwendig ?
  - Unter welchen Bedingungen können Sie dem Einsatz von IDS zustimmen ?
  - In welchem Umfang ist der Einsatz von IDS notwendig ?
  - Wie würden Sie persönlich auf den Einsatz von IDS in Ihrem Arbeitsumfeld reagieren ?

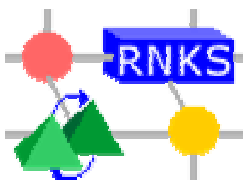


# Umfrageergebnisse: Notwendigkeit IDS

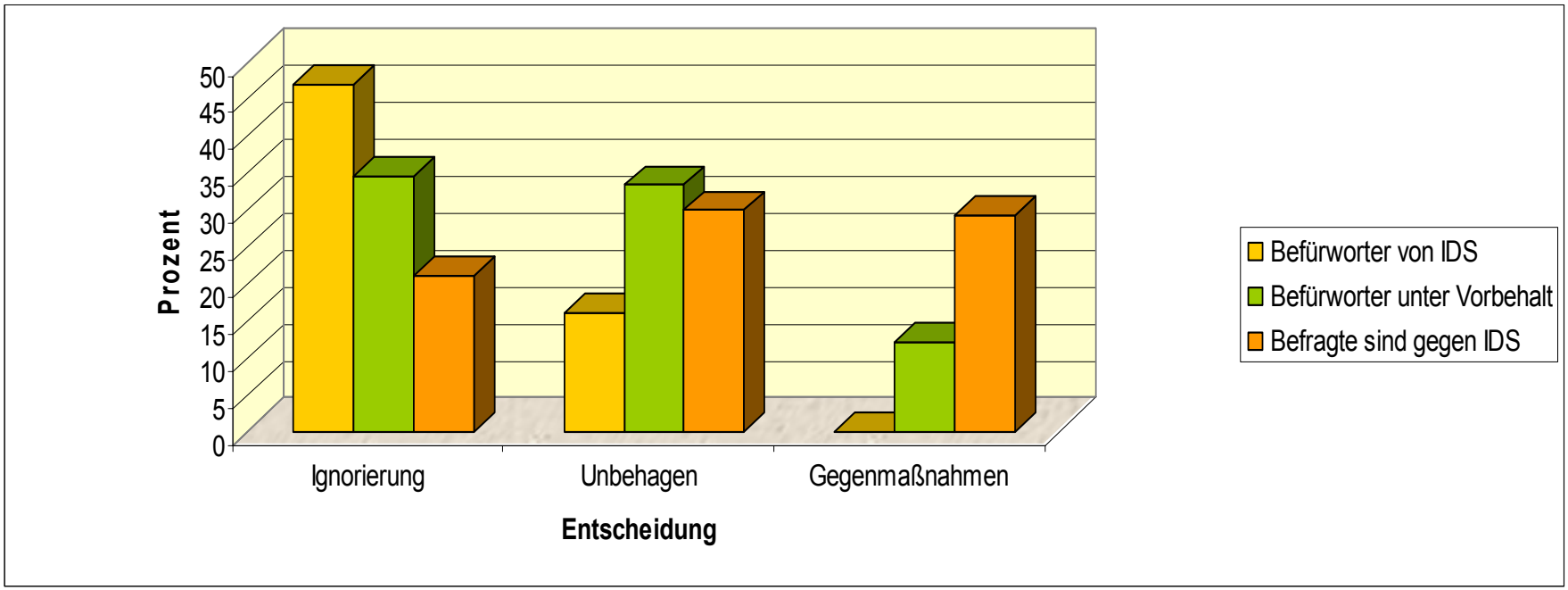


# Umfrageergebnisse: Bedingungen / Umfang der IDS-Überwachung

- **Frage 2**:
  - überwiegend Forderung nach restriktivem Einsatz
- **Frage 3**:
  - mehrheitlich Beschränkung auf sensitive Bereiche
  - knapp 50% Unterstützung bei Befragten, die Einsatz von IDS prinzipiell ablehnen



# Umfrageergebnisse: Reaktionen auf IDS-Überwachung

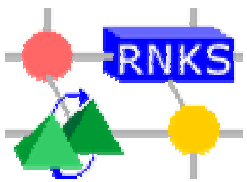


# Fazit

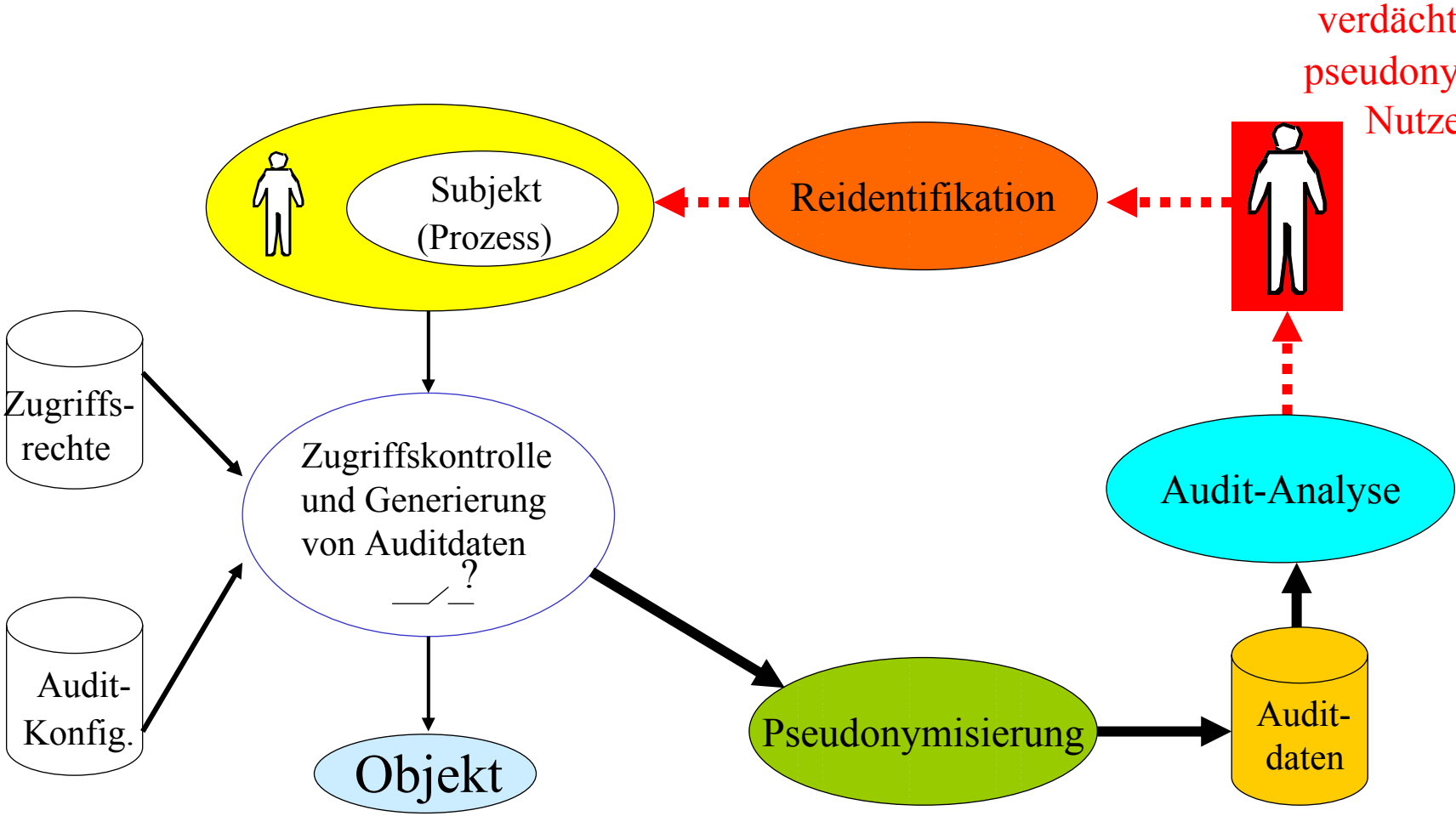
- Intrusion Detection Systeme ermöglichen die Erhebung personenbezogener Daten.
  - Verletzung des Rechts auf informationelle Selbstbestimmung
  - Missbrauch möglich
  - Psychische Belastung

☞ Der Einsatz von IDS muss sensibel geplant werden und den Anforderungen des Datenschutzes angepasst werden !!!

- **Lösungsansätze**: - 4-Augen-Prinzip (*offline*)
  - Pseudonymes Audit (*online*, *offline*)



# Pseudonymes Audit

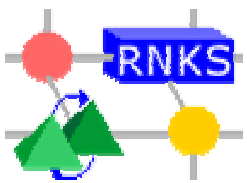


# Vorteile des pseudonymisierten Audit

- Reduzierte Beobachtbarkeit des Nutzerverhaltens
- Reidentifikation nur bei begründeten Verdacht
- Gewährleistung von Nachweisbarkeit und Datenschutz

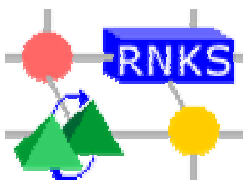
☞ Verfahrenstechnischer Schutz gegen mißbräuchliche Entschlüsselung:

- ➢ DFG-Projekt Uni Dortmund (LS Prof. Biskup)



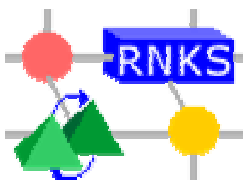
# Probleme des Einsatzes von Intrusion Detection Systemen

- Hohes Aufkommen an Auditdaten
- Datenschutz
- Beschränkte Auswertungseffizienz
- Fehlalarme
- *Response*-Mechanismen
- Selbstschutz
- Kooperation von Intrusion Detection Systemen
- Hoher Wartungsaufwand



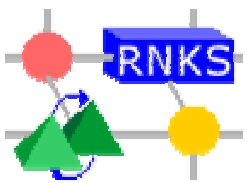
# Beschränkte Auswertungseffizienz

- Festgefügte Analysestrukturen
  - zentrale Auswerteeinheit / Passive Agenten
  - keine Anpassung an spezifische Überwachungstopologien
  - keine (Teil-) Auswertung in den Agenten
  - keine Behandlung von Überlastsituationen
- Ungenügende Effizienz
  - regelbasierte Auswertung / Verwendung eines Standard-Expertensystems
  - schwerfällig bzgl. nebenläufigen Attackenabläufen und Attackenvarianten



# Probleme des Einsatzes von Intrusion Detection Systemen

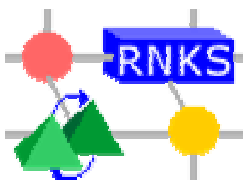
- Hohes Aufkommen an Auditdaten
- Datenschutz
- Beschränkte Auswertungseffizienz
- Fehlalarme
- *Response*-Mechanismen
- Selbstschutz
- Kooperation von Intrusion Detection Systemen
- Hoher Wartungsaufwand



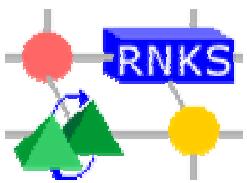
# Fehlalarme

Die Suche nach den in Signaturen kodierten Mustern schließt eigentlich Fehlalarme per Definition aus.

- Praxis zeichnet ein anderes Bild.
  - hohe Zahl von Fehlalarmen (bis zu 10000 pro Monat berichtet)
- **Ursachen**:
  - qualitative Einschränkungen der Audit-Funktionen
  - vor allem: empirische Signaturentwicklung
- **Lösungsansätze:**
  - Alarmkorrelationen
  - systematische Ableitung von Signaturen

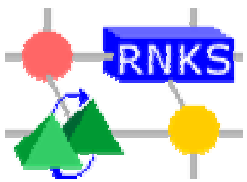


# IV.
# Eigene Forschungsarbeiten

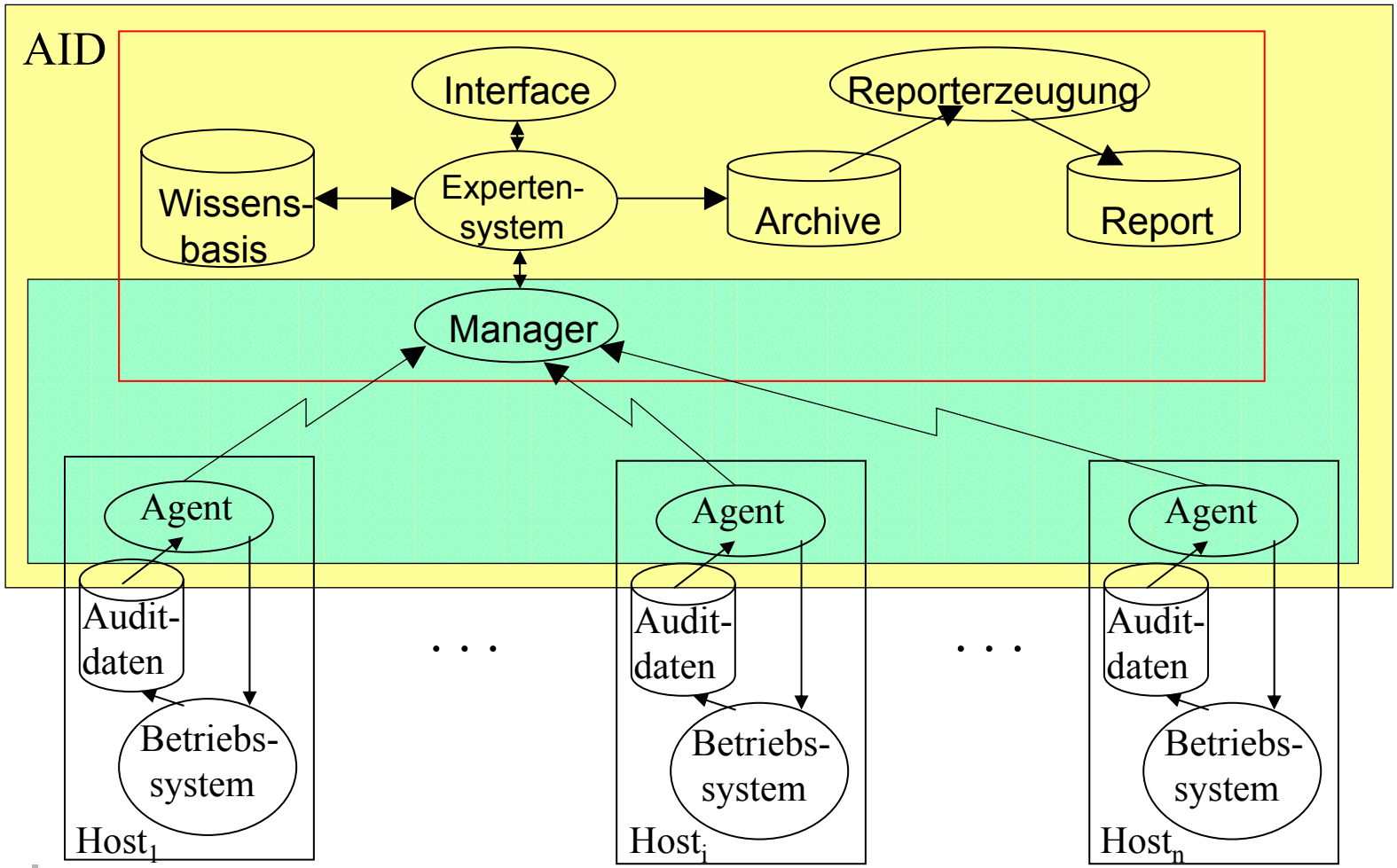


# Startpunkt AID

- **Entwickler:** BTU Cottbus / LS RNKS
  - Dr. Michael Sobirey, Birk Richter
- **Ziel**: Überwachung lokaler Netze
- **Auswertungsprinzip:** zentralistisch
  - keine Auswertung in den Agenten
  - Auswertungseinheit: *RTworks* Expertensystem
  - Abruf der Auditdaten von den Agenten durch gesteuertes Polling
  - Betriebssystem-unabhängiges Auditdaten-Format
  - Analyseleistung: - 2,5 Mbyte/min

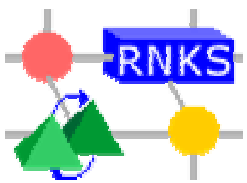


# AID-Architektur



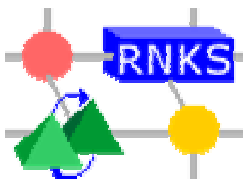
# Innovative Elemente von AID

- Echtzeit-Überwachung eines lokalen Netzes
  - Signaturanalyse
- Datenschutz-orientiertes Audit
  - Datenschutz-orientiertes pseudonymes Audit
- Verbesserung des Netz-Audit
  - Host-orientiertes Netz-Audit (HONA)



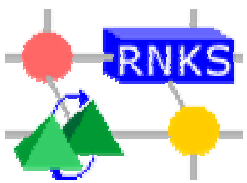
# Defizite von AID

- **Feste Auswertungsstruktur**
  - Zentrale Auswertungseinheit / Passive Agenten
  - keine Anpassung an Überwachungserfordernisse
  - keine (Vor-) Auswertung in den Agenten
  - keine Behandlung von Überlastsituationen
- **Aufwendige Signatur-Updates**
  - Fehlen geeigneter Beschreibungsmittel
- **Ungenügende Effizienz**
  - regelbasierte Analyse
  - Nutzung kommerzieller Expertensysteme

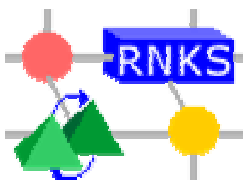


# Verfolgte Ansätze

- Flexible, verteilte Systemarchitekturen
  - ☞ HEIDI-Ansatz
- Nutzerfreundliche Signaturbeschreibung und –Updates
  - ☞ Signaturbeschreibungssprache SHEDEL
- Integration effizienter Auswertungsmethoden
  - ☞ Analysealgorithmus STRAFER



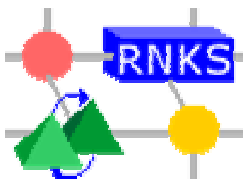
# IV.1.
# Der HEIDI-Ansatz
(*High-Efficient Intrusion Detection Infrastructure*)



# HEIDI-Ansatz

- **Ziel**: Entwicklung flexibler IDS-Architekturen, die den Erfordernissen der Einsatzumgebung und der Arbeitslast angepasst werden können.
  - Generierung maßgeschneiderter IDS
  - Baukastensystem

- **Basiskomponenten**
  - Sensoren
  - Agenten
  - Nutzerinterfaces



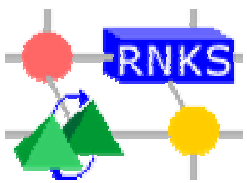
# HEIDI-Sensoren

Module zum Erfassen und Vorverarbeiten von Auditdaten.

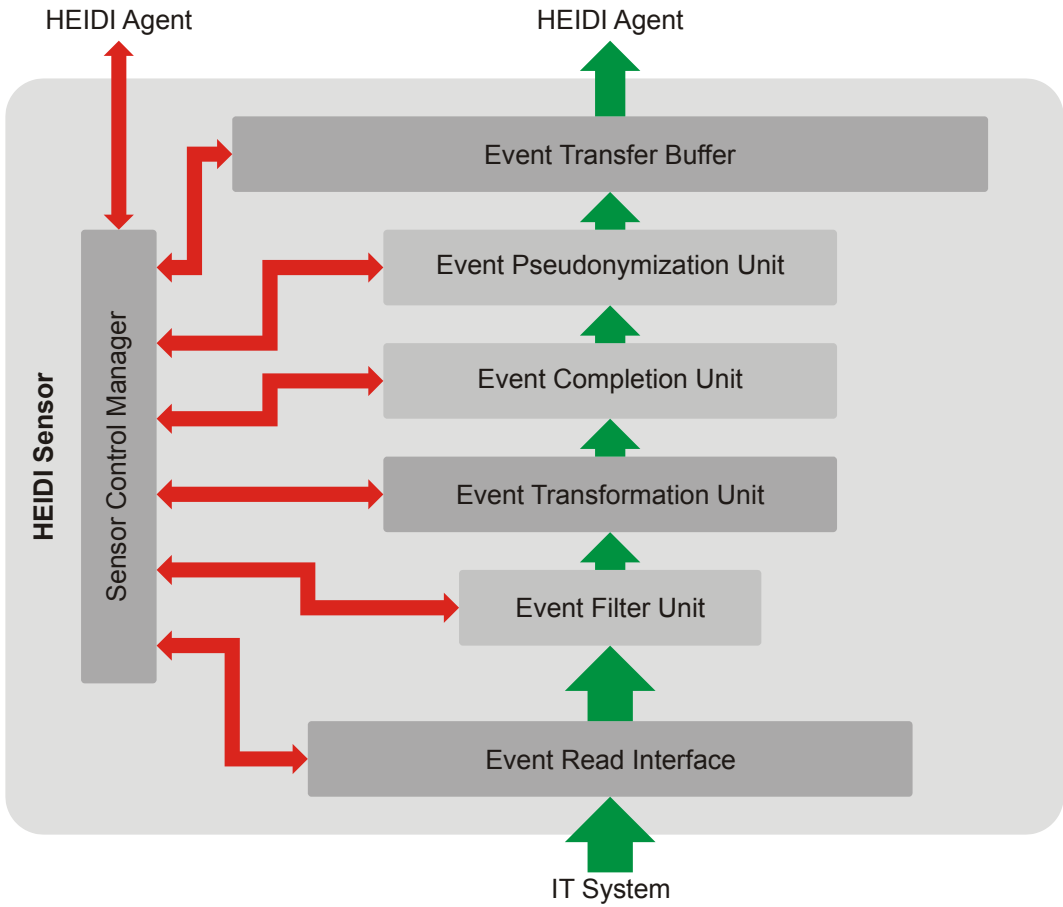
- **Ziel**: schnelles Erfassen, Verarbeiten und Weiterleiten der Daten
- **Operationsmodus**: autonom
- **Lokation**: wo erforderlich

  → in Abhängigkeit von der verwendeten Sicherheits-strategie
- **Komponenten**: permanent und optional

☞ Verschiedene Sensoren auf einem Host werden durch den Agenten koordiniert.



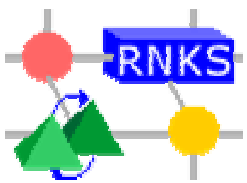
# HEIDI-Sensor



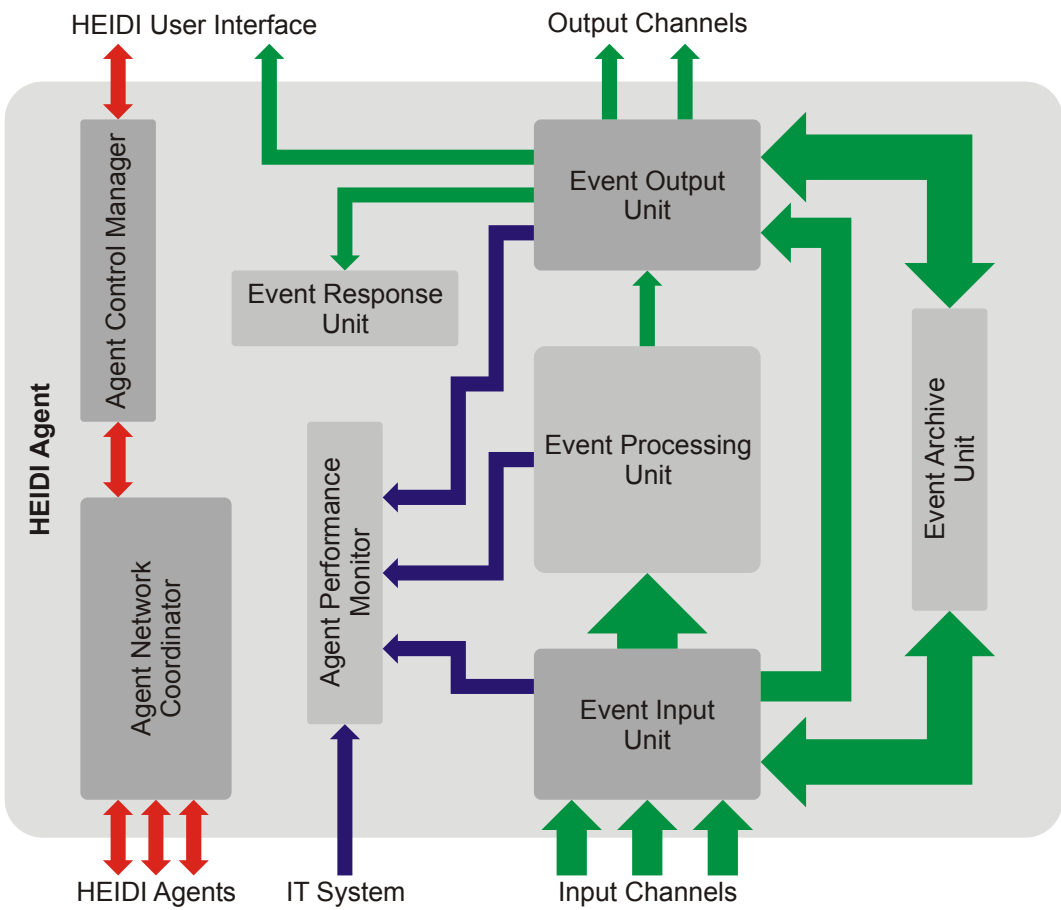
# HEIDI-Agenten

Modul für die Auswertung lokal und entfernt erfasster Auditdaten

- **Operationsmodus**: autonom
- **Lokation**: mindestens 1 Agent pro Host
- **Arbeitsweise**:
  - vorrangig lokale Auswertung
  - Weiterleitung von Daten und erkannten Teilsignaturen
  - gleichzeitige Ausführung mehrerer Analyseprozesse
  - Delegierung bei Überlast
- **Komponenten**: permanent und optional



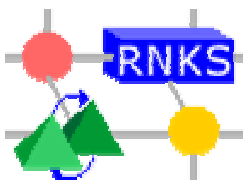
# HEIDI-Agent



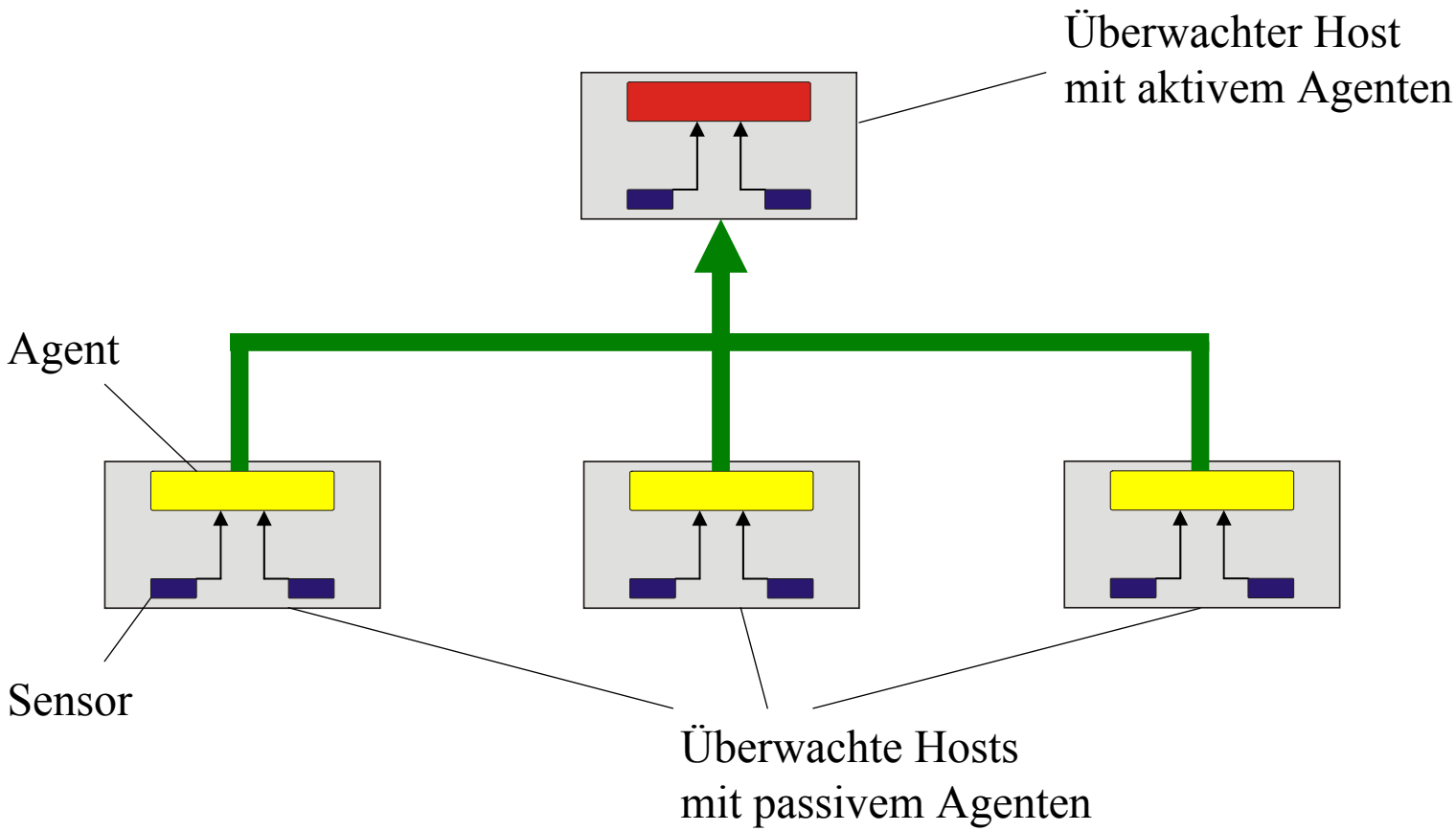
# HEIDI-Nutzerinterface

Graphisches Interface für den Sicherheitsadministrator für die Verwaltung des Intrusion Detection Systems.

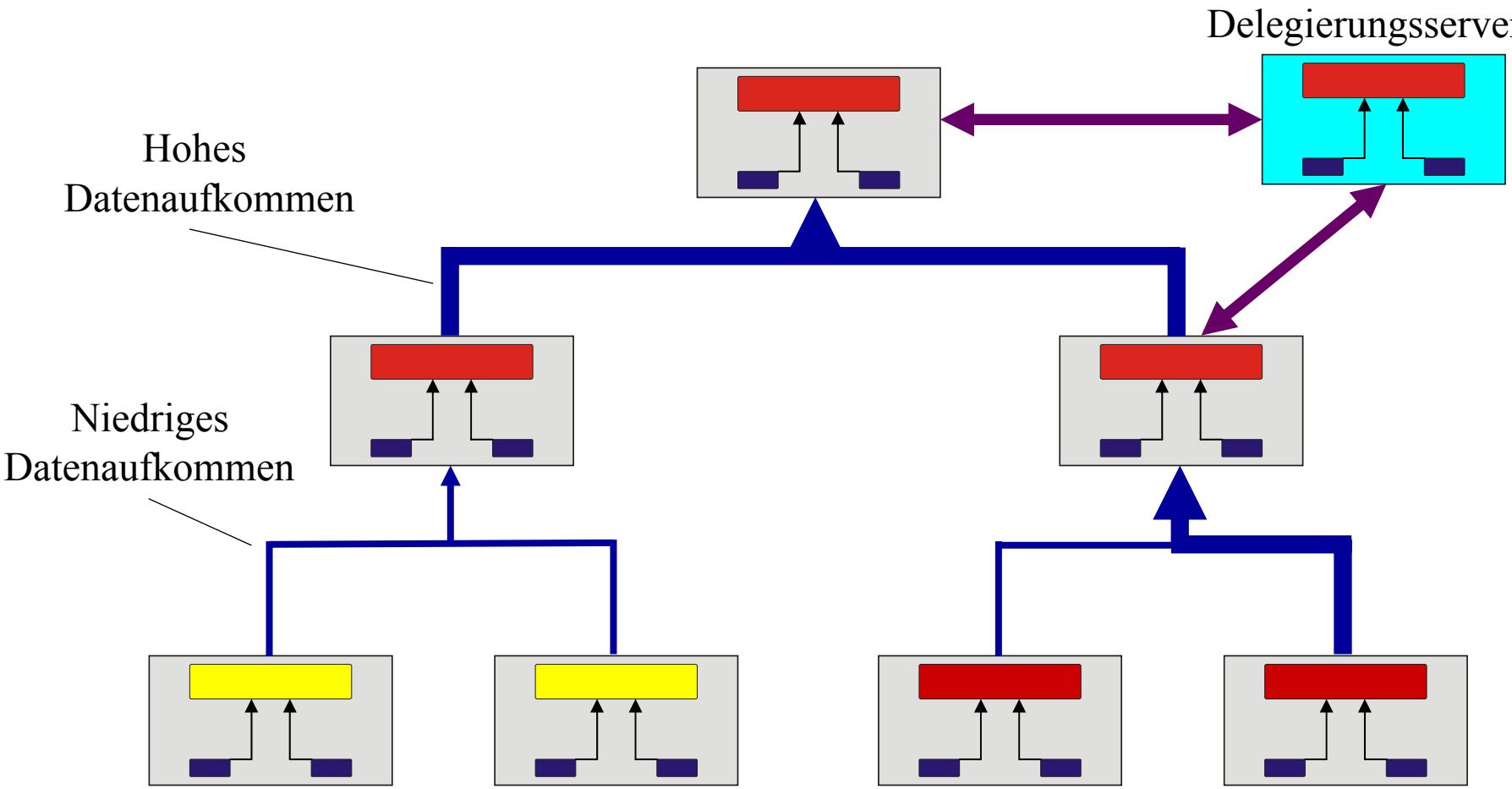
- Systemkonfiguration
- Systemmanagement
  - Anbindung verschiedener Agenten
- Visualisierung der Analyseergebnisse



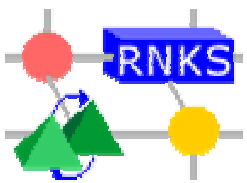
# AID-Struktur mittels HEIDI



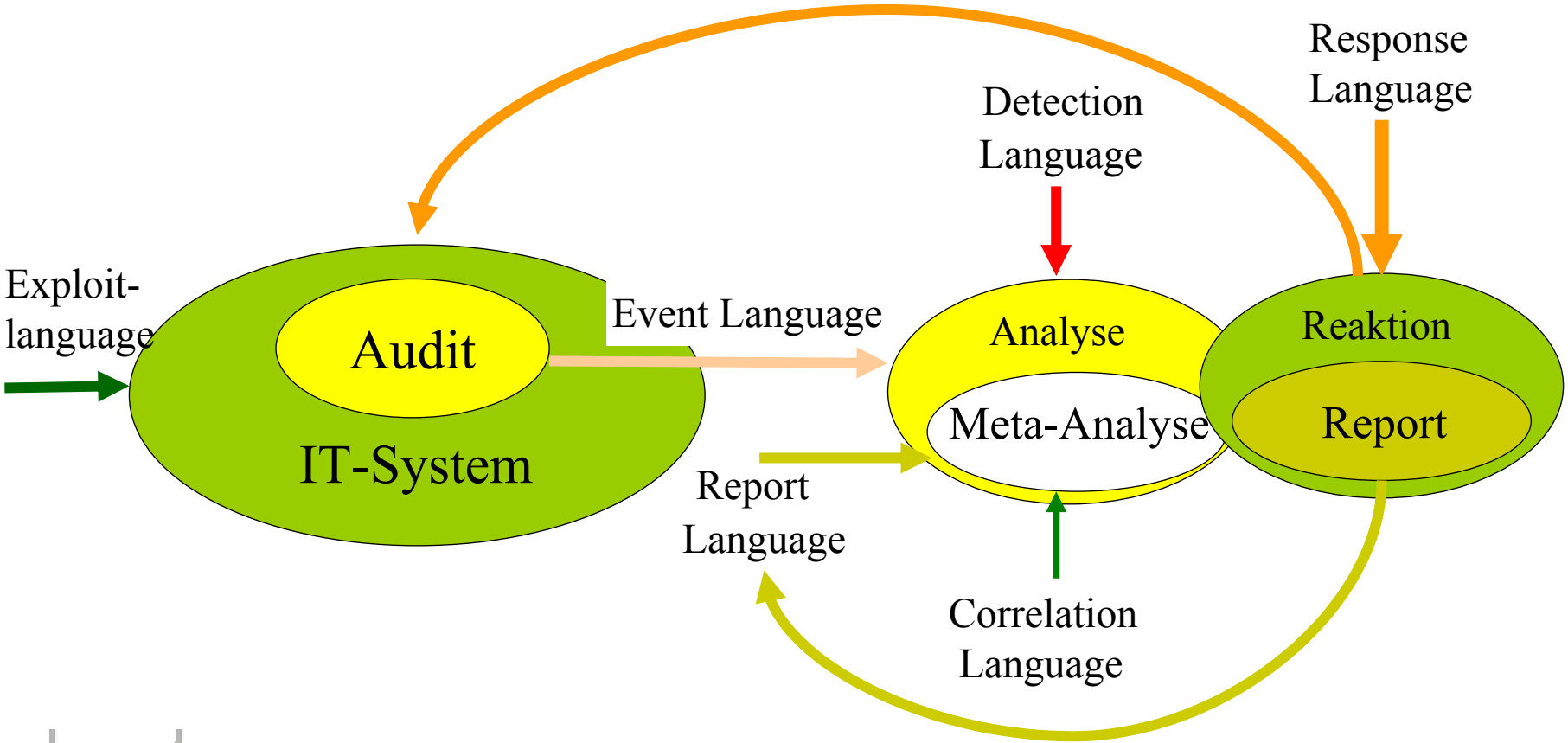
# Verteiltes IDS mit HEIDI



# IV.2.
# SHEDEL
(*Simple Hierarchical Event Description Language*)



# Attackensprachen im Intrusion Detection

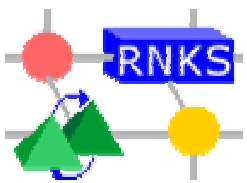


# Detection Languages

- Detection Languages dienen der Kodierung von Signaturen

☞ **Typischerweise verwendet jedes IDS eine eigene Sprache !!!**

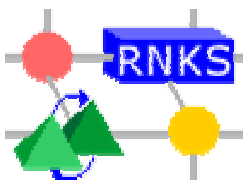
- **Beispiele:**
  - P-Best-Sprache in Emerald
  - RUSSEL für ASAX
    - regel-basierte Sprachen
  - STATL für STAT-Tool-Suite
    - zustandsorientiert
  - IDIOT-IDS-Sprache
    - gefärbte Petri Netze



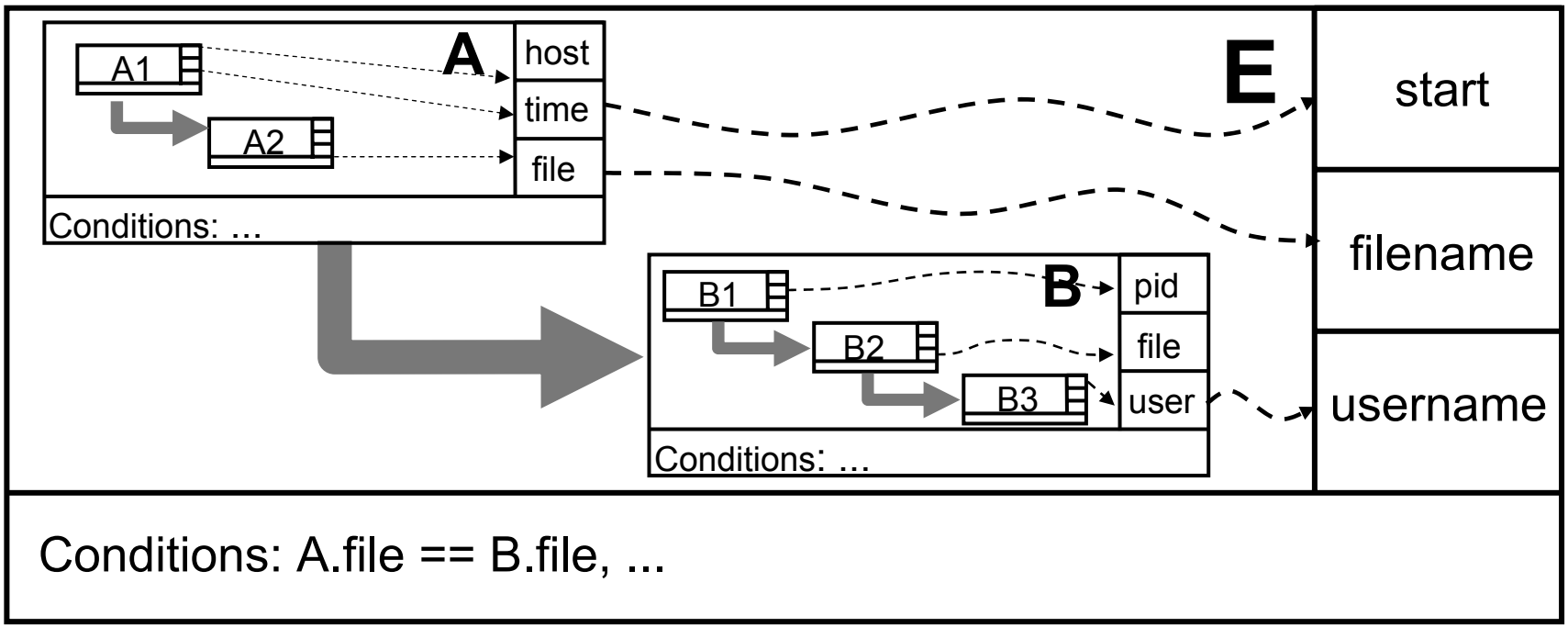
# SHEDEL

- **Ziel**: Ereignis-basierte Beschreibung von Signaturen unabhängig von der Erkennungsprozedur
  - → Einfache Aktualisierung von Signaturbasen
  - → Grundlage für Analysealgorithmus
- **Basiskonzept**: Ereignis
  - → Typ + Merkmalen
  - → besteht aus einer Folge von Ereignisse (Schritte)
  - → Beschreibung der temporalen Beziehungen
  - → Bedingungen
  - → Aktionen

  Ereignishierarchien



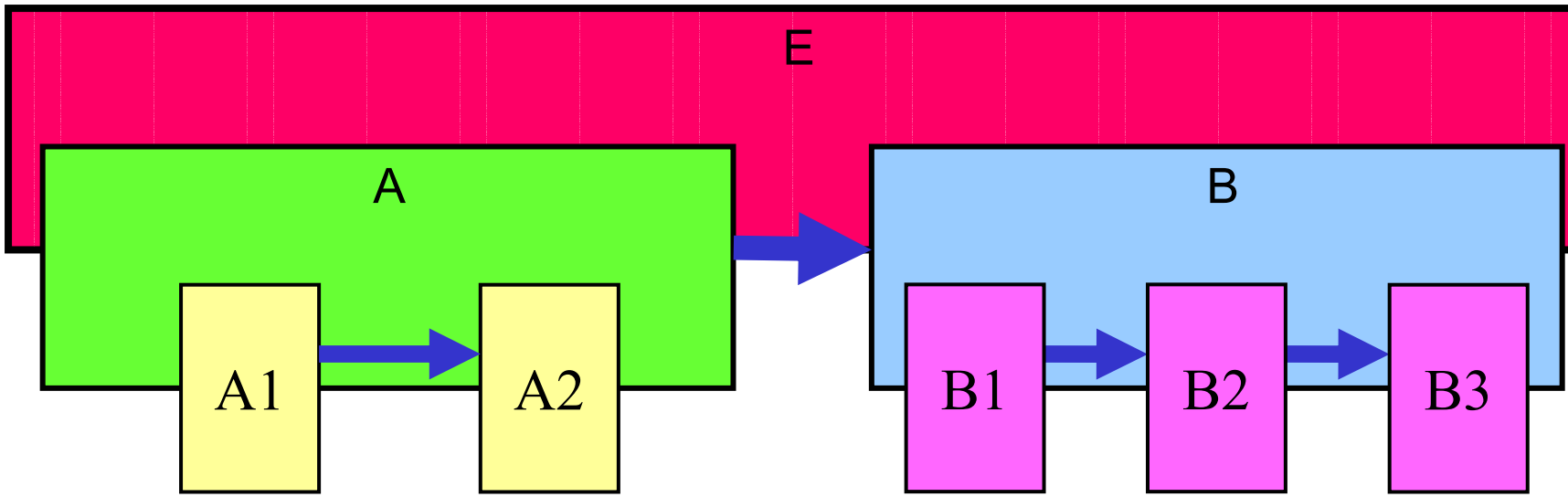
# Ereignisbeschreibung in SHEDEL (1)




Zeitliche Folge

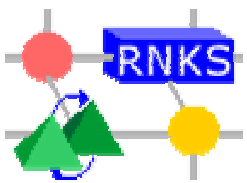
Merkmals-Definition

# Ereignisbeschreibung in SHEDEL (2)




☞ Ereignishierarchien für Teilsignaturen

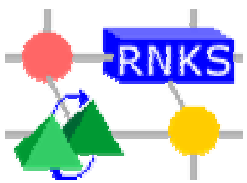
# IV.2.2
# STRAFER
(*STRAight Forward Event Recognition*)



# STRAFER-Algorithmus (1)

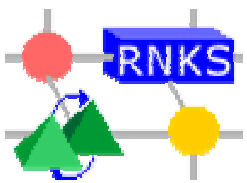
- **Ziel**: Reduzierung der Analyse-/Entdeckungszeit

- **Ansatz**: - Ausnutzung von Wissen über Struktur und Inhalt der Auditdaten
  - Verringerung der Anzahl zu überprüfender Bedingungen
  → Grundlage: SHEDEL Signaturbeschreibungen



# Herkömmliche Ansätze

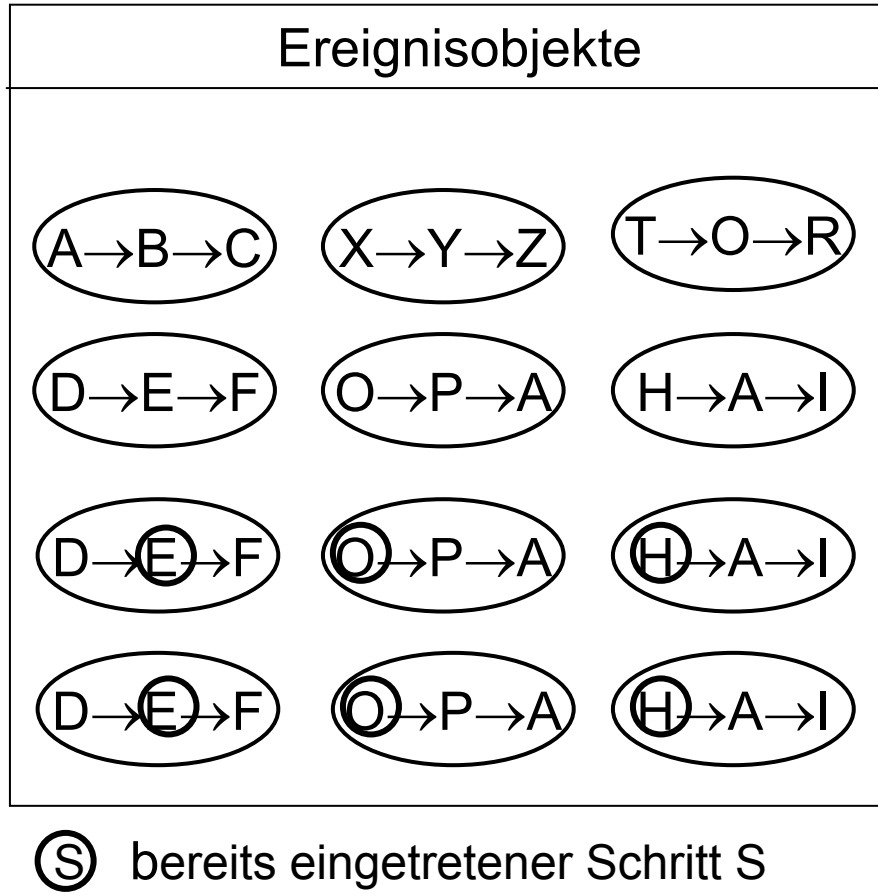
- Nutzung der Interferenzalgorithmen der Expertensysteme
  - EMERALD (P-Best), AID (RTworks), CMDS (Clips)
  - Nutzung von Standard-Match-Algorithmen
- Code-Erzeugung (STAT, IDIOT)
  - Abbildung der Signaturen in C++-Module
  - verarbeiten unabhängig voneinander Auditdaten
- **Vermutung**: General-purpose Analyseverfahren implizieren Mehraufwand



# Grundidee von STRAFER (1)

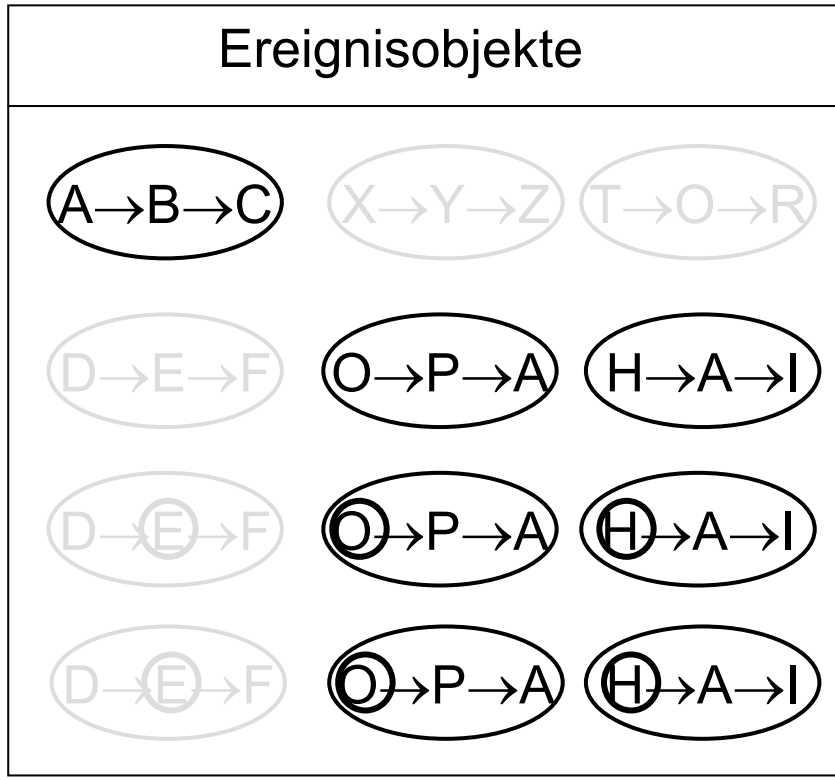
| Eingangsereignis |
| --- |
| Typ: A |

- Welche Ereignisobjekte (partielle Matches) können mit einem Eingangsereignis vom Typ A erweitert werden?
- Nur Ereignisobjekte, die Schritte vom Typ A enthalten!



# Grundidee von STRAFER (2)

| Eingangsereignis |
| --- |
| Typ: A |

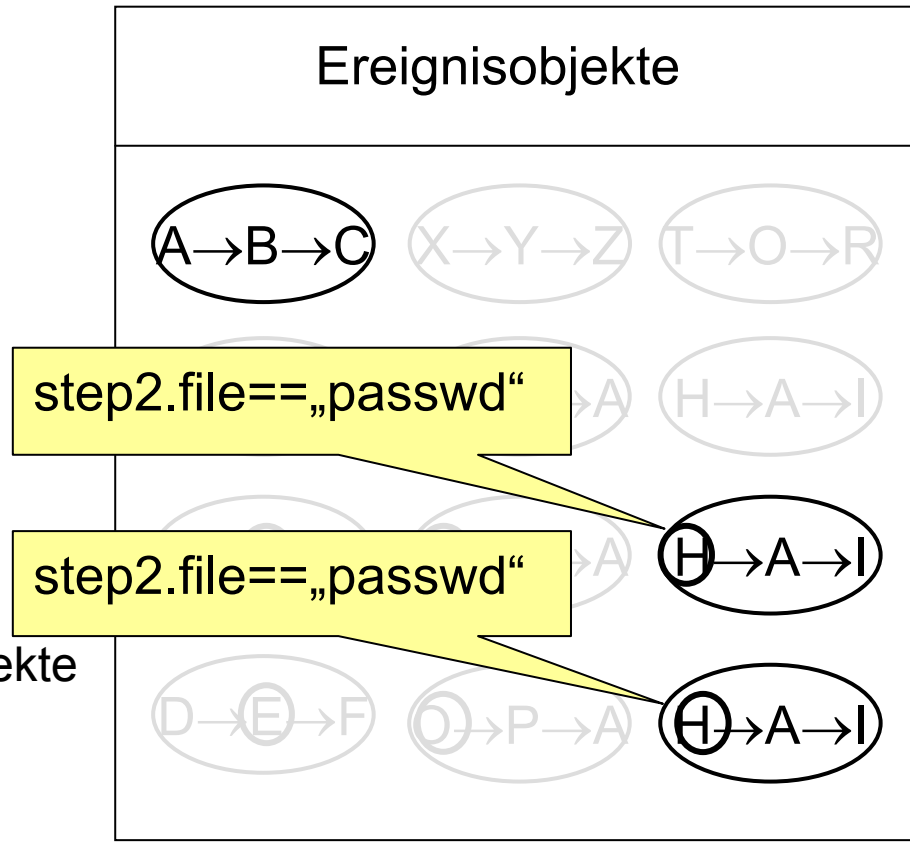



- Nur Schritte vom Typ A, die keinen oder bereits eingetretene Vorgängerschritte besitzen, können feuern

Ⓢ bereits eingetretener Schritt S

# Grundidee von STRAFER (3)

| Eingangsereignis |
| --- |
| Typ: A |

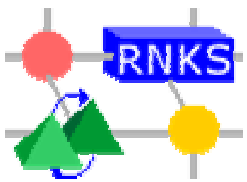



- Intra-Event-Bedingungen an einem Signaturschritt liefern bei gleichem Eingangsereignis für alle Ereignisobjekte dieser Signatur das gleiche Ergebnis und werden nur einmal geprüft.

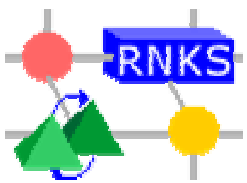
Ⓢ bereits eingetretener Schritt S

# Stand der Arbeiten

- **HEIDI**
  - Prototyp-Implentierung Sensoren und Agent (WIN32)
    - fast fertiggestellt
    - Konfiguration verschiedener Architekturen
    - Leistungsmessungen
- **SHEDEL**
  - Compiler
  - Graphisches Interface
  - SHEDEL II
- **STRAFER**
  - Prototyp-Implementierung
  - Vergleichsmessungen



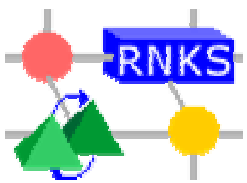
# IV.
# Ausblick



# Zukunft der Intrusion Detection (1)

Intrusion Detection Systeme sind ein hochwirksames Mittel für die Erkennung, Analyse und Verhinderung von Einbrüchen in Rechnersystemen und Netzinfrastrukturen.

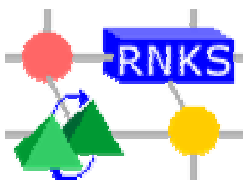
- zunehmende Bedeutung
- momentan einzige Sicherheitstechnik, die nach erfolgten Sicherheits-verletzungen greift
- notwendige komplementäre Ergänzung anderer Sicherheitstechniken



# Zukunft des Intrusion Detection (2)

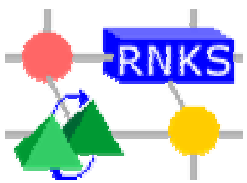
Der wirksame Einsatz von Intrusion Detection Systemen erfordert jedoc noch einen weiteren Technologiefortschritt !!!

- **Hauptprobleme:**
  - Komplexität der Technologie und des Einsatzgebietes
  - Schnelle und effiziente Signaturerfassung
  - Rela-Time-Fähigkeit der Einbruchserkennung
  - Intelligente Integration in das Netz-/Systemmanagement



# Literatur (1)


- **Escamilla, Terry: Intrusion Detection**.
  John Wiley & Son, 1998
- **Amoroso, Edward G.: Intrusion Detection**.
  Intrusion.Net Books, New Jersey, 1999
- **Sobirey, Michael: Datenschutzorientiertes Intrusion Detection.**
  Vieweg-Verlag, DuD-Fachbeiträge, 1999

# Literatur (2)


- **McHugh, J.: Intrusion and intrusion detection.** In: International Journal of Information Security. Heidelberg, Springer Verlag, 2001, Nr. 1, S. 14-35.

- **Axelsson, Stefan: Research in Intrusion Detection Systems: A Survey.** Goeteborg, Chalmers University of Technology, Technical Report No. 98-17, 1998.

- **Holz, T.; Meier, M.; König, H.: Bausteine für effiziente Intrusion Detection Systeme.** PIK 25 (2002) 3, S. 144-157.

  (zu letztem Abschnitt)