



# PIPES<sup>med</sup> : Ein flexibles Werkzeug zur Verarbeitung kontinuierlicher Datenströme in der Medizin

Bernhard Seeger, Jürgen Krämer, Thomas Penzel, Richard Lenz  
Philipps-Universität Marburg

Gefördert durch die DFG (Se 553/4-3)



## Motivation und Problemdefinition

## Datenstrommanagement

- Anforderungen
- Architekturübersicht

## PIPES

- Funktionalität
- Projekte und Ergebnisse

## Zusammenfassung

# Motivation

---

- Explosion der Kosten im Gesundheitswesen
  - Alternde Gesellschaft
  - Viele chronisch Kranke
- Lösungsansatz
  - Durchgehender Einsatz von Sensoren zum Monitoring
    - Stärkung der Eigenverantwortlichkeit
    - Selbstständiges Wohnen
  - Attraktivität
    - Niedrige Kosten für die Datengewinnung durch Sensoren
    - Steigende Akzeptanz auf Grund der Miniaturisierung
- Problem
  - Verarbeitung der kontinuierlich anfallenden Datenmassen
  - Semantische Lücke
    - Low-Level Sensordaten  $\leftrightarrow$  Bedarf in der Anwendung

# Praxis des Monitorings

- Firmen
  - Dräger/Siemens
  - Philipps
  - ...
- Produkte
  - Datenerfassung
  - Direkter Plot der erfassten Daten in Echtzeit auf einem Bildschirm
- Produkte je nach Aufgabe (z. B. Produktkatalog von Philipps):
  - Gas Monitoring
  - Blood Pressure
  - Capnography
  - ECG
  - Hemodynamic
  - Neurological
  - Pulse Oximetry
  - Temperature
  - Transcutaneous Gases
  - ...



## Multiple Datenströme

### ■ IST:

- Prinzip: „Anzeigen und Vergessen“
- Niedrige Schwellenwerte: Falsch positive Alarme
- Desensibilisierung der Mitarbeiter

### ■ SOLL:

- Zeitnahe Interpretation –  
Erkennen komplexer Risikosituationen

# Überblick

---

Motivation und Problemdefinition

**Datenstrommanagement**

- Anforderungen
- Architekturübersicht

PIPES

- Funktionalität
- Projekte und Ergebnisse

Zusammenfassung

# Anforderungen

---

- Entkopplung der Sensoren von spezifischen Anwendungen
- Fusion und Transformation der Sensordaten
  - auf Basis einer einfachen Anfragesprache
- Flexible Definition von Alarmen
  - Multivariate Alarme
  - Ausnutzung verschiedener Sensoren
  - Berücksichtigung von Patientendaten
- Leistungsgarantien
  - Echtzeitverhalten
  - Skalierbar in der Anzahl der Nutzer

# Architektur

**Präsentationsebene:** z.B. Visualisierung

*High-level Datenströme*

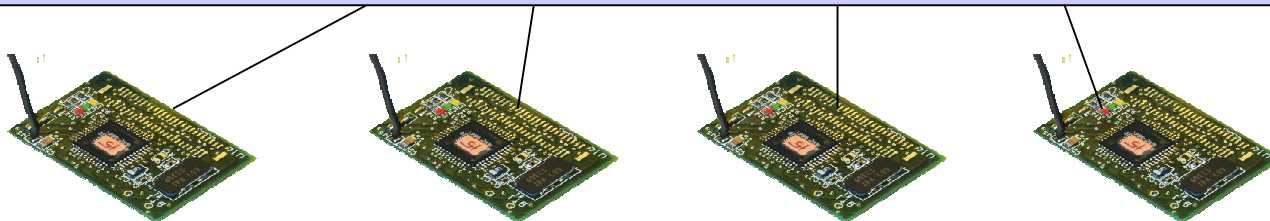
**Applikationsebene:**

- Fusion von Datenströmen – Verknüpfung mit Datenbanken
- Filtern – Erkennen ungewöhnlicher Ereignisse - Veredlung

*Formatierte und annotierte Datenpakete*

**Directory-Service:**

- Transformation in hardwareunabhängiges Format(z.B. XML)
- Publish/Subscribe Mechanismen



Rohdaten

# Fokus von PIPES

**Präsentationsebene:** z.B. Visualisierung

*High-level Datenströme*

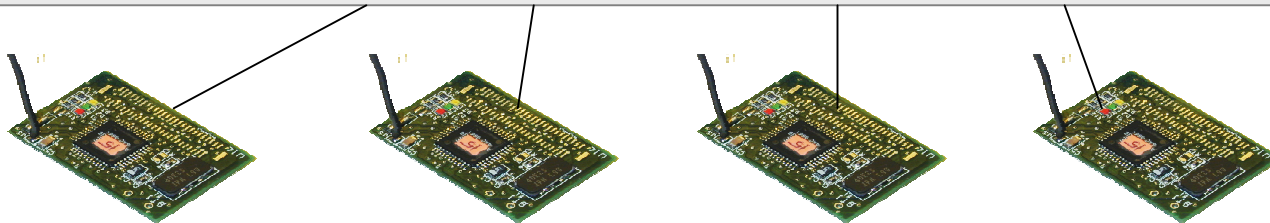
**Applikationsebene:**

- Fusion von Datenströmen – Verknüpfung mit Datenbanken
- Filtern – Erkennen ungewöhnlicher Ereignisse

*Formatierte und annotierte Datenpakete*

**Directory-Service:**

- Transformation in hardwareunabhängiges Format(z.B. XML)
- Publish/Subscribe Mechanismen



*Rohdaten*



# Überblick

---

Motivation und Problemdefinition

Datenstrommanagement

- Anforderungen
- Architekturübersicht

**PIPES**

- Funktionalität
- Projekte und Ergebnisse

Zusammenfassung

# Fokus von PIPES

---

- PIPES:
  - Public Infrastructure for Processing and Exploring Data Streams
  - Infrastruktur für DSMS (DatenstromManagementSysteme)
- Unterschiede zu DBMS
  - Menge von Datensätzen ist potentiell unendlich und zeitbehaftet
  - Anfragesemantik entspricht der von **temporalen Datenbanken**
- Benutzeranforderungen
  - Kein Einfluss veralteter Daten auf aktuelle Ergebnisse
  - Integration der Daten verschiedener Ströme bzgl. zeitlicher Nähe
    - Z. B. zur Definition eines Alarms
- Gleitende Zeitfenster
  - Endlicher Ausschnitt des Datenstroms
  - Einschränkung der Verarbeitung auf das aktuelle Zeitfenster

# Gleitende Zeitfenster in Anfragesprachen

---

- CQL (Continuous Query Language)  
Anfragesprache für kontinuierliche Datenströme
  - siehe Streams-Projekt der Stanford University (Widom et al)
- Erweiterung von SQL um Zeitfenster
  - PressureStream (PatientenID, Pressure, TimeStamp)

“Bei welchen Patienten war der Blutdruck  
**über einen Zeitraum von 5 Minuten** höher als 190?”

```
SELECT PatientenID
FROM PressureStream [Range 5 minutes]
GROUP BY PatientenID
HAVING (Min(Pressure) > 190);
```

# Fusion von Datenströmen

---

- Sinnvolle Kombination verschiedener Datenströme
  - ➔ Definition multivariater Alarme auf Datenströmen
- Beispiel
  - Alarmbedingung
    - HeartRate > 150 AND Pressure > 180  
für ein Zeitraum von 2 Minuten

```
SELECT PatientenID
FROM  PreasureStream [Range 2 minutes],
      RateStream [Range 2 minutes]
WHERE PreasureStream.PatientenID = RateStream.PatientenID
GROUP BY PatientenID
HAVING (Min(Pressure) > 190 AND Min(HeartRate) > 150);
```

# Verteilungs- und Dichteschätzer

---

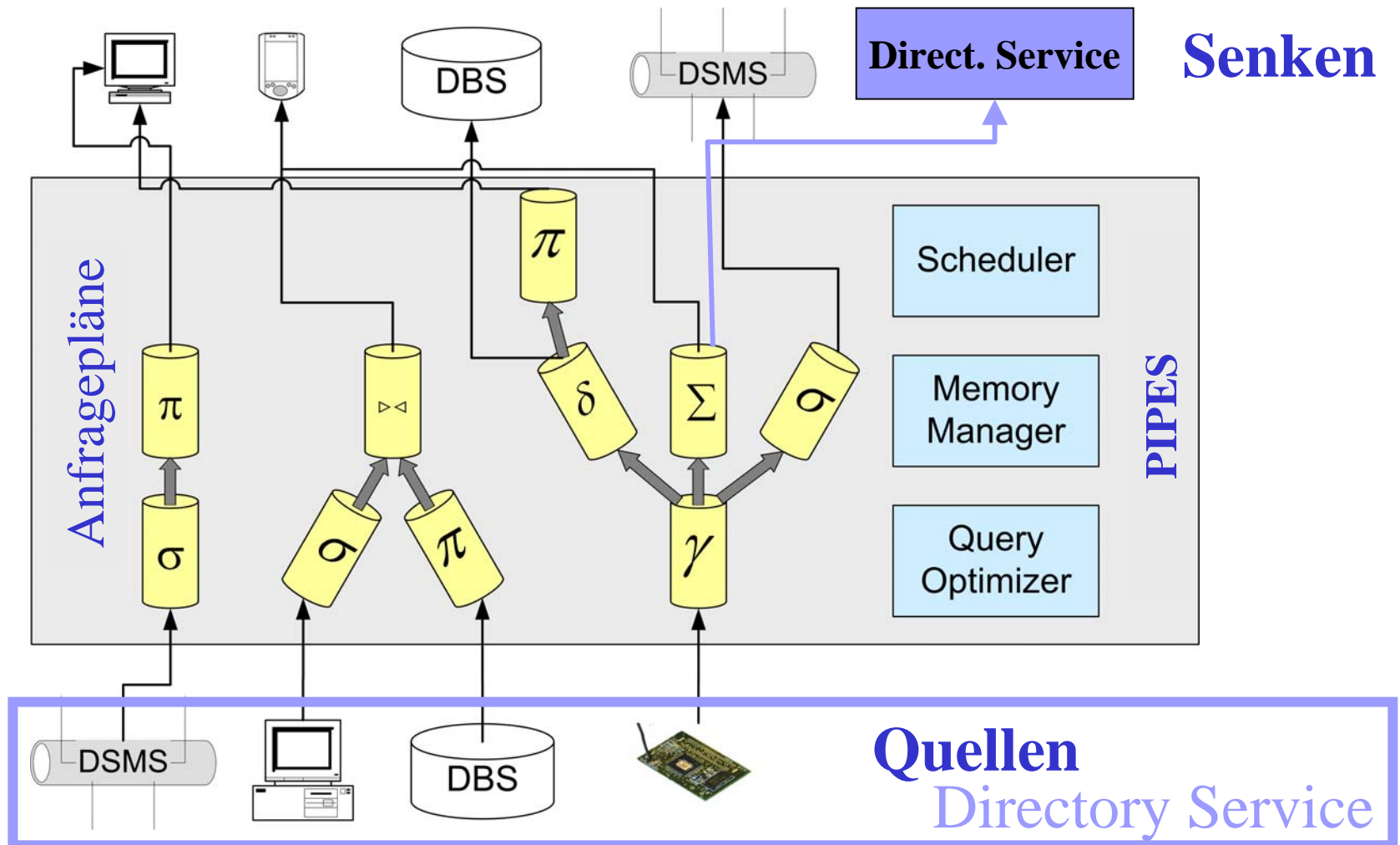
- Ziele
  - Kontinuierliche Analyse von Datenströmen
  - Schätzung der aktuellen **Verteilung** des Datenstroms
  - Vergleich der Verteilung mit den zuletzt beobachteten Daten
- Warum Verteilungs- und Dichteschätzer?
  - Schätzung wichtiger Kenngrößen des Datenstroms wie z. B. der Momente und Extrema
- Methoden der nichtparametrischen Statistik
  - Keine Modellannahmen
  - Schätzung einer unbekanntem Verteilung

# Anwendung

---

- Ausreißerentdeckung in Datenströmen
  - Hat der neue Datensatz einen ungewöhnlichen Wert?
- Gibt es eine Häufung eines Musters in einem Datenstrom?
- Welche Muster treten häufig zeitlich nah beieinander auf?

# Gesamtbild von PIPES



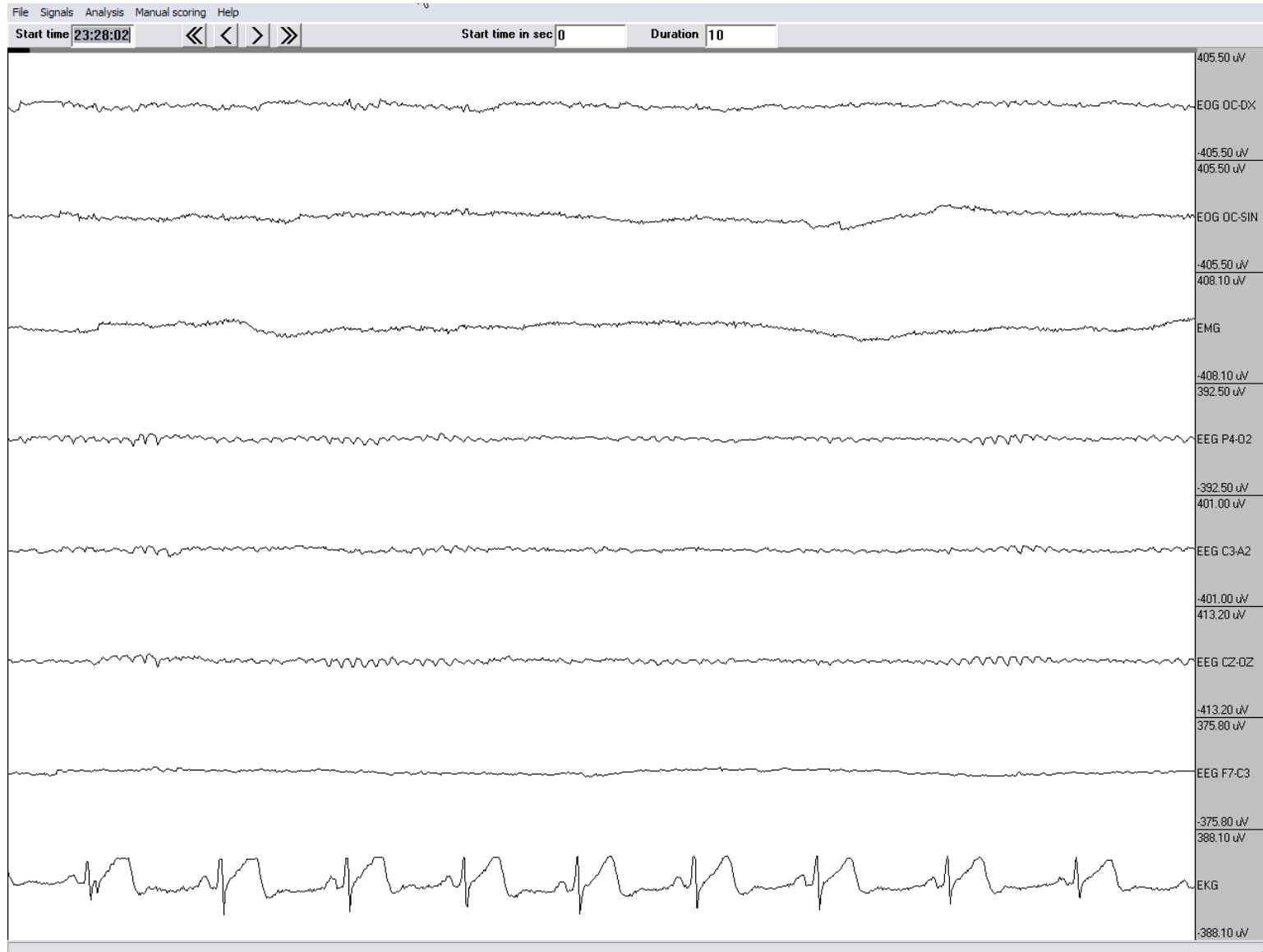
# Anwendungen

---

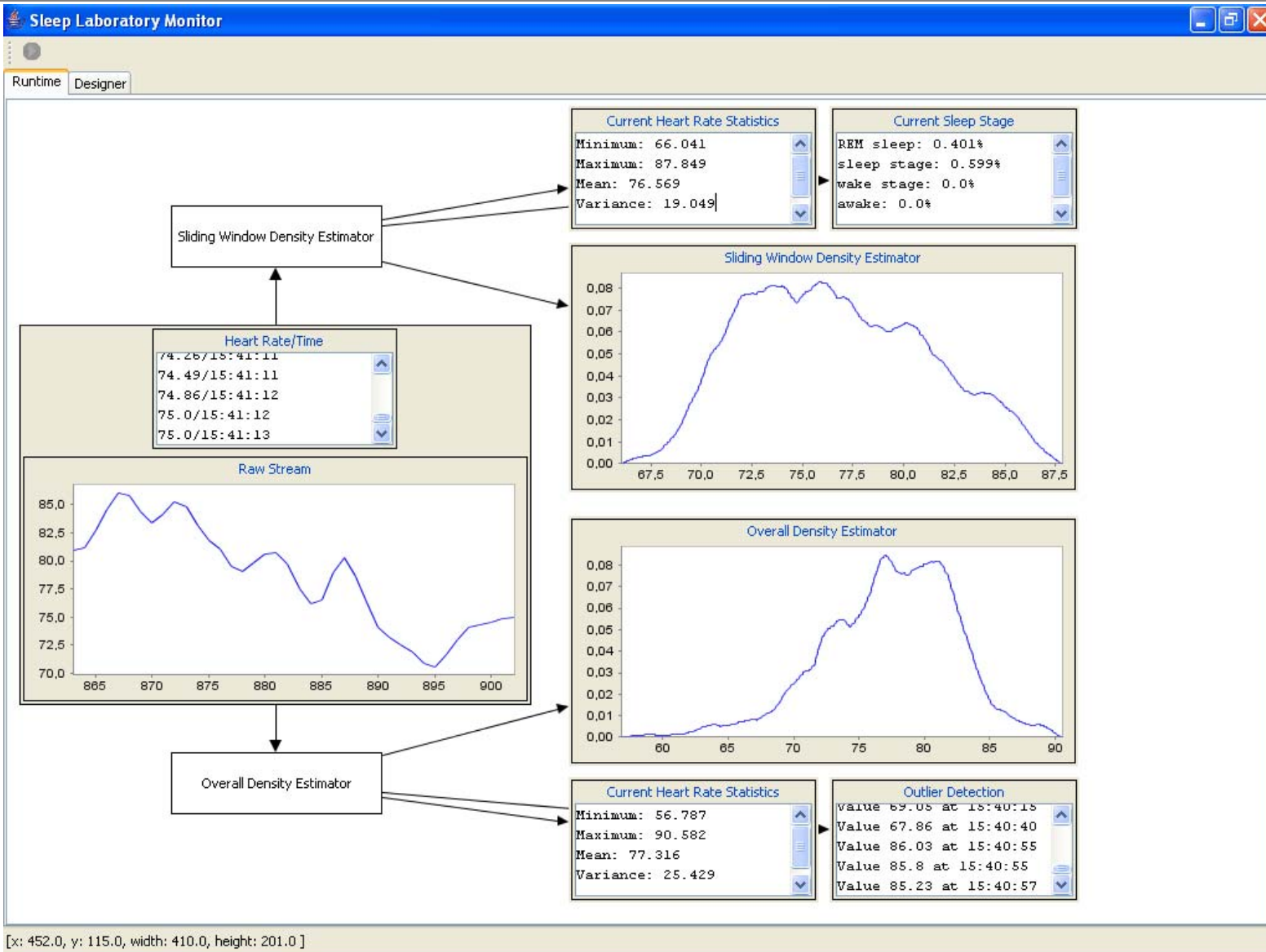
- Bestehende Anwendungen von PIPES
  - Sensordatenverarbeitung bei Produktionsanlagen
    - Kopplung von PIPES mit kommerziellen Systemen
  - Schlaflabor der Uni Marburg
    - Online-Analyse der Datenströme



# Beispiel: Visualisierung



# Beispiel: Veredlung von Datenströmen



[x: 452.0, y: 115.0, width: 410.0, height: 201.0]

# Überblick

---

**Motivation und Problemdefinition**

**Datenstrommanagement**

- Anforderungen
- Architekturübersicht

**PIPES**

- Funktionalität
- Projekte und Ergebnisse

**Zusammenfassung**

# Zusammenfassung

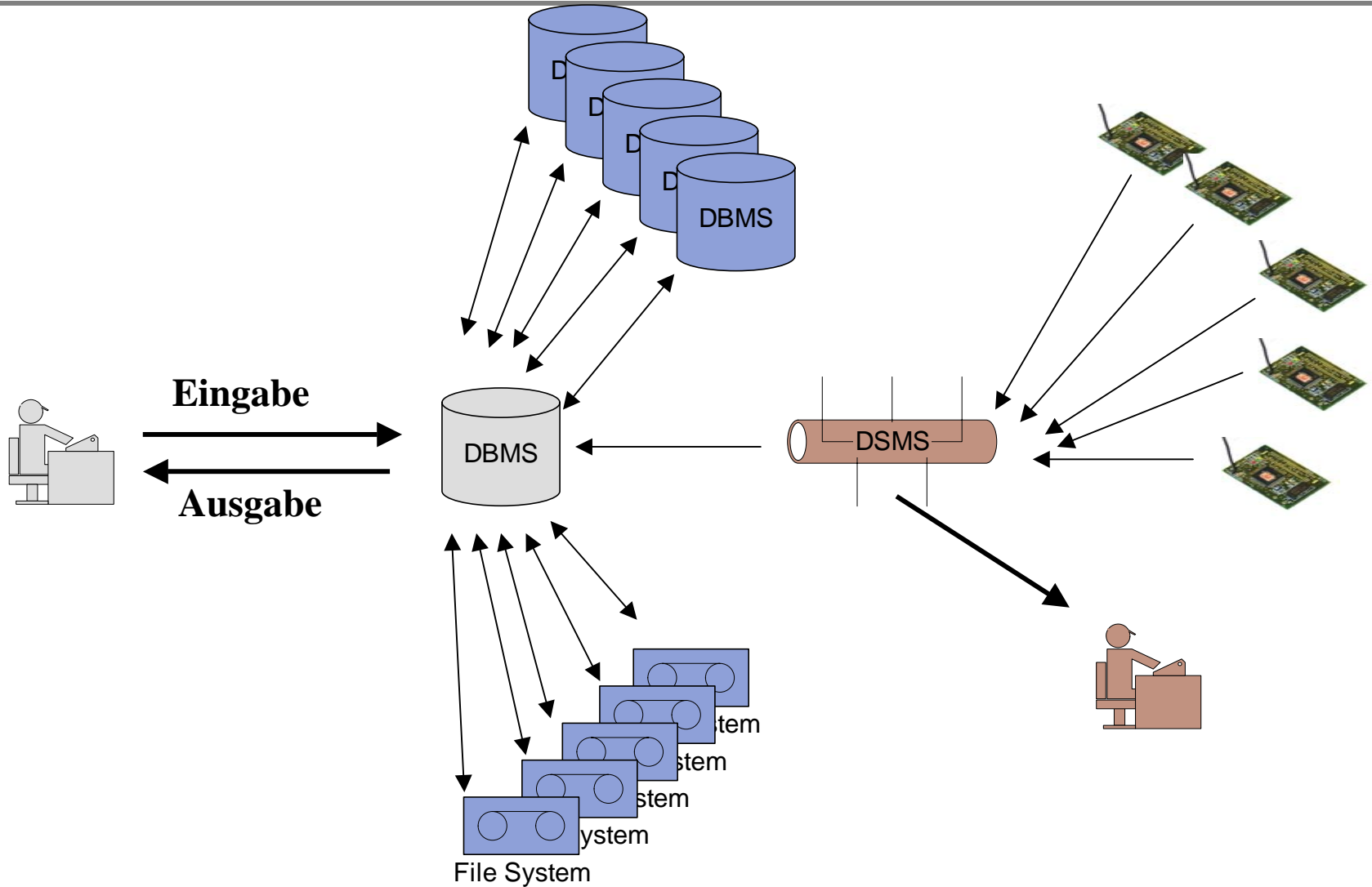
---

- Zunehmende Notwendigkeit zur Online-Verarbeitung von Datenströmen
- Ziel von PIPES:
  - Transformation der low-level Daten in semantisch relevantere Daten
- Ansatz von PIPES
  - Verwendung und Adaption der erfolgreichen Techniken aus dem Bereich Datenbanken
- Visionäre Anwendung
  - Überwachung von Patienten im Intensivbereich

---

**Vielen Dank  
für Ihre Aufmerksamkeit**

# Informationslandschaft



# Praxis des Monitorings

---



## ■ Infinity® Omega

- erstes System zur Patientenüberwachung mit zwei Bildschirmen
  - Weshalb Beschränkung auf zwei Bildschirme?
- Vervielfachung der Anzahl der gleichzeitig angezeigten Monitoring-Parameter und Kurven
  - ➔ Überladene Bildschirm
- Flexible Voreinstellung
  - Anzeige relevanter Optionen, einschließlich Parameter, Kurvenanordnung und **Alarmgrenzen**
  - ➔ Abhängigkeit von der Hardware

# Alarme

---

- Definition von Schwellenwerten auf einem Parameter → Alarmgrenzen
    - Auslösung eines Alarms
  - Problem
    - Kritische Situationen i. A. abhängig von mehreren Parametern
- 
- Konservative Alarmgrenzen
    - Viele falsche Alarme
      - Desensibilisierung des Pflegepersonals



# Directory-Service

---

- Zweck
  - Beschreibung und Bereitstellung in einem hardwareunabhängigen Format
- Aufgaben
  - Transformation der hardwareabhängigen Sensordaten in eine standardisierte Form
    - z. B. XML
  - Beschreibung der Sensordaten durch Metadaten
  - Zugriff auf die Daten durch Standardprotokolle
- Lösung
  - Methoden der Signalverarbeitung
  - Publish-Subscribe-Mechanismus
    - Anwendungen subscribieren Sensordaten
    - Kontinuierliche Weiterleitung der Daten eines Sensors an die beim Sensor subscribierten Applikationen

# Applikationsebene

---

- Zweck
  - Transformation von low-level in high-level Daten
  
- Aufgaben
  - Fusion von Sensordaten
    - Kombination verschiedener Sensoren
    - Bereitstellung als **logischer** Sensor
  - Anreicherung von Sensordaten mit persistenten Daten
  - Komplexe Analysen auf Sensordaten
    - Erkennen von ungewöhnlichen Daten
  
- Lösung
  - Anfragesprache ähnlich zu SQL
  - Publish/Subscribe-Mechanismus für Anfragen
    - Registrierung von Anfragen durch Anwender
    - Kontinuierliches Senden der Antworten

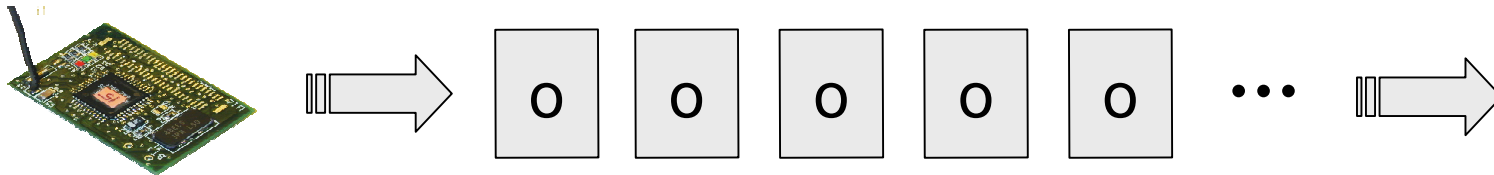
# Präsentationsebene

---

- Zweck
  - Bereitstellung der Anfrageergebnisse auf verschiedenen Hardwareplattformen
  
- Aufgaben
  - Verwaltung von Metadaten zur Beschreibung von Ausgabegeräten (und Benutzerkontexte)
    - Bandbreite
  
- Lösung
  - Bereitstellung kontext-sensitiver Methoden zur Transformation und Kompression der Daten

# Sensoren und Datenströme

- Kontinuierliche Folge von Datensätzen



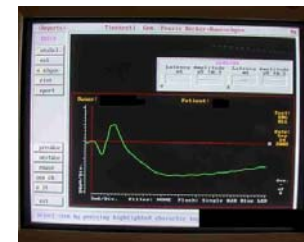
- Zeit als integraler Bestandteil

- Autonome Datenquellen

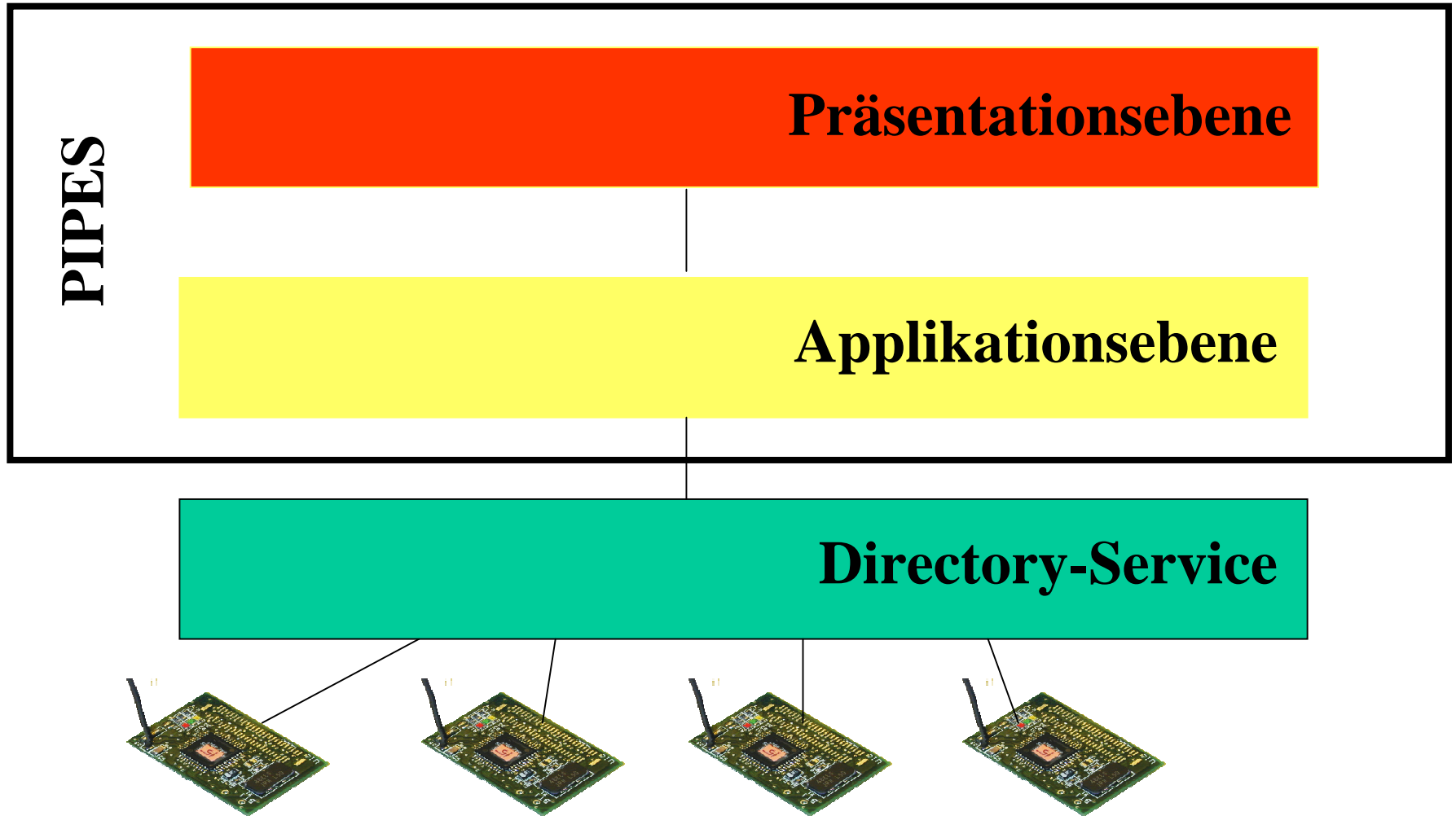
- Sensoren, mobile Geräte, Softwareapplikationen, ...

- Technologien

- Lokale und globale Vernetzung



# 3-Ebenen-Architektur



# Interner Ablauf der Anfrage

---

- Fensteroperator (in der from-Klausel)
  - Jedes Tupel der Datenströme mit Zeitpunkt  $t$  erhält zusätzlich einen Endzeitpunkt  $t + 1 + 2$  Minuten.
  - Ergebnistupel bekommt ebenfalls ein Zeitintervall
    - Schnitt der Eingabeintervalle

- Beispiel:

- $w = 2$  Minuten

- EingabeTupel

- $(123, 195, 12:00) \rightarrow (123, 195, 12:00, 12:02)$

- $(123, 150, 12:01) \rightarrow (123, 150, 12:01, 12:03)$

- Ergebnistupel

- $(123, 195, 150, 12:01, 12:02)$

# Kumulierter Schätzer

- Idee des Frameworks
  - Blockweise Verarbeitung des Datenstroms
  - Separater Schätzer für jeden Block
  - Iterative Berechnung der konvexen Linearkombination der Blockschätzer
- Formale Darstellung für endlichen Datenstrom

$$\hat{g}_j(x) = \sum_{i=1}^j \omega_i \hat{f}_i(x) \quad \text{mit} \quad \sum_{i=1}^j \omega_i = 1, \omega_i \geq 0 \quad \forall i = 1, \dots, j$$

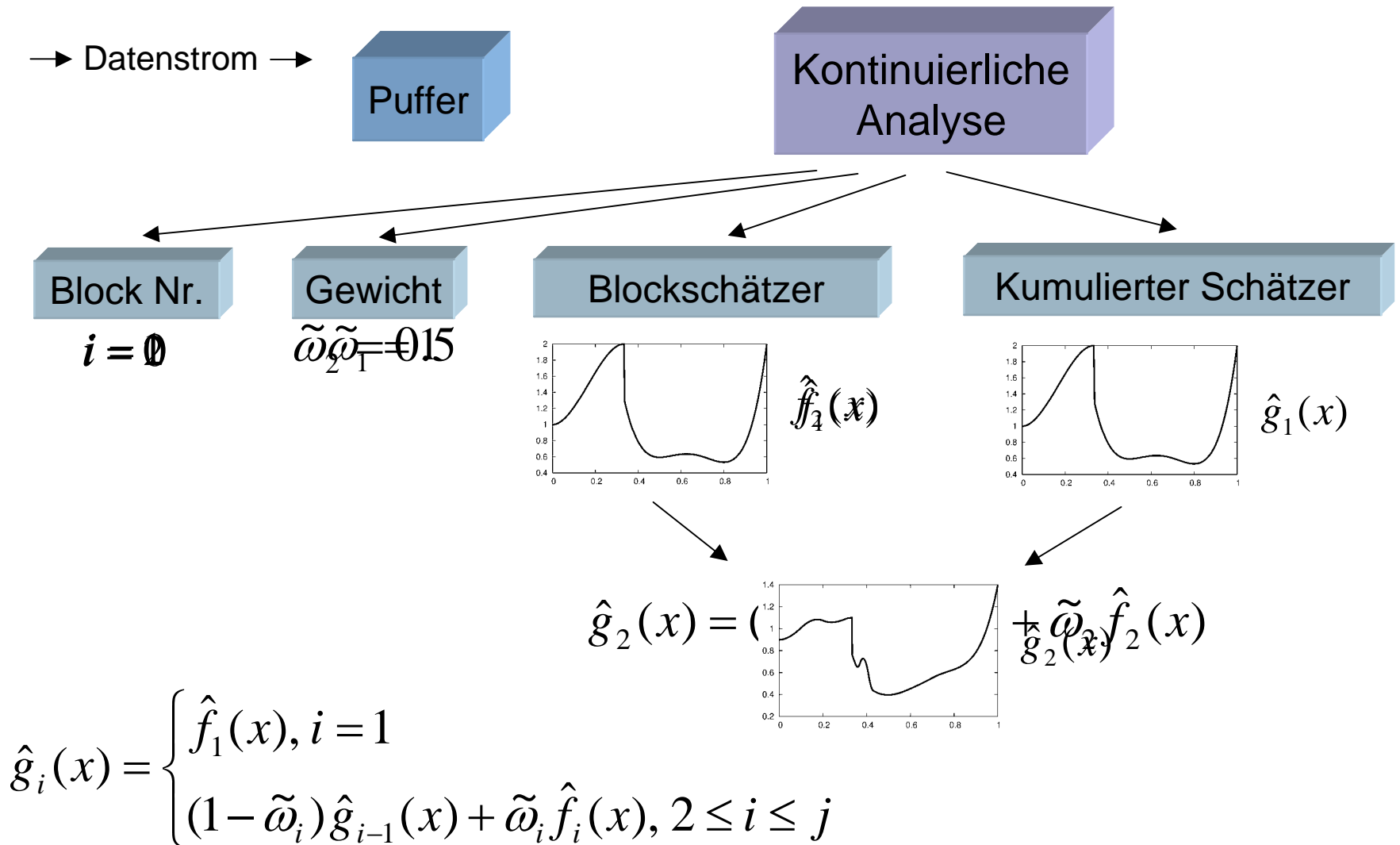
- Online-Berechnung des **kumulierten Schätzers**

## Kumulierter Schätzer

$$g_i(x) = \begin{cases} \hat{f}_1(x), & i=1 \\ (1 - \tilde{\omega}_i) \hat{g}_{i-1}(x) + \tilde{\omega}_i \hat{f}_i(x), & 2 \leq i \leq j \end{cases}$$

$$\text{mit } \tilde{\omega}_i, 0 \leq \tilde{\omega}_i \leq 1, 2 \leq i \leq j, \tilde{\omega}_1 = 1$$

# Kumulierter Schätzer (2)



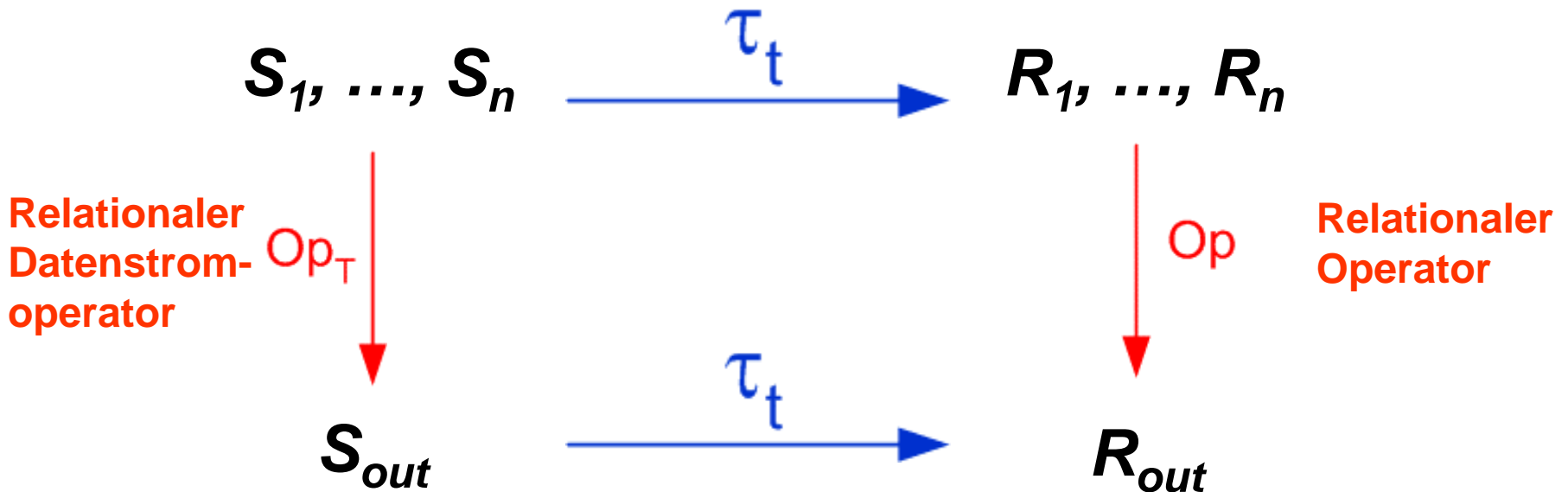


# Relationale Stromoperatoren

---

## Schnappschuss-Reduzierbarkeit

- Schnappschuss  $\tau_t$ 
  - Abbildung eines physischen Datenstroms auf eine nicht-temporale Relation.
  - Relation enthält genau die zum Zeitpunkt  $t$  **gültigen Datensätze**



# Framework für Analyseverfahren

---

- Berechnung nichtparametrischer Schätzer auf Datenströmen
- Erfüllung der Anforderungen für Analyseverfahren
  - Online berechenbar
  - Limitierter Speicher
- Eigenschaften des Frameworks
  - Kumulierte Schätzer
  - Parameter
    - Kompressionsschritt
    - Konvexer Vereinigungsschritt
    - Schätztechnik
  - Speicheradaptivität

# Aktuelle Arbeiten

---

- Fragestellungen
  - Laufzeitumgebung
    - Realzeit-Anforderungen
    - Scheduling der Operatoren
  - Kostenmodelle
    - Entwicklung neuer Schätztechniken
  - Entwicklung von Strategien für Adaptivität
    - Speicheradaptivität
    - CPU-Ablaufsteuerung
  - Analyseoperatoren
    - Medizintypische Operatoren

# Anfrage in CQL

---

```
SELECT PatientenID
FROM PreasureStream [Range 2 minutes],
     RateStream [Range 2 minutes]
WHERE PreasureStream.PatientenID = RateStream.PatientenID
GROUP BY PatientenID
HAVING (Min(Pressure) > 190 AND Min(HeartRate) > 150);
```

Sinnvolle Ausgaben erfordern i. A. noch die Verknüpfung mit Daten aus persistenten Datenbanken

*(z.B. eine strukturierte elektronische Patientenakte).*