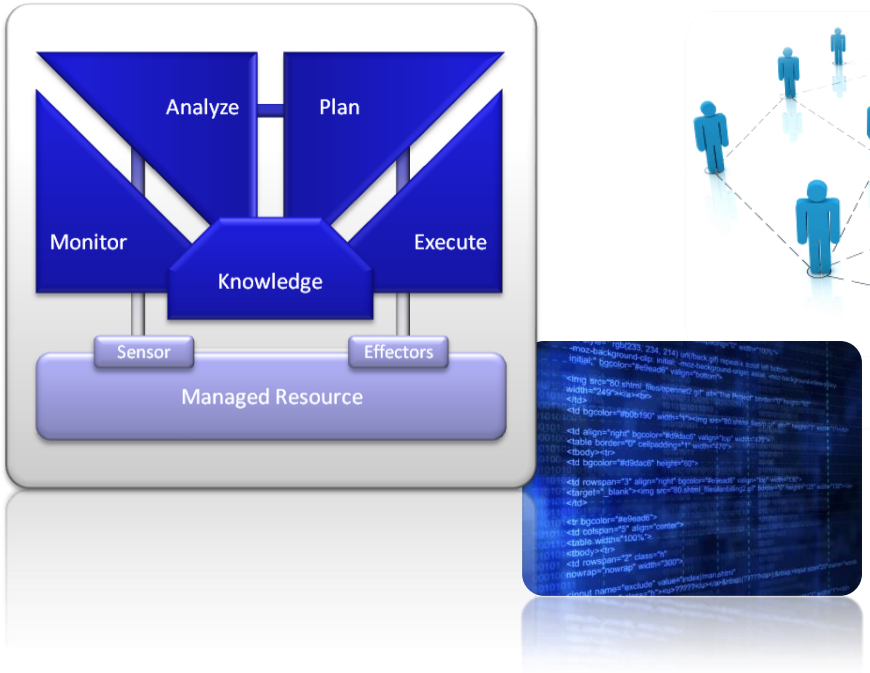
Autonomic Computing in Peer-to-Peer Systems

TECHNISCHE UNIVERSITÄT DARMSTADT



Michael Behrisch

Tutor: Kalman Graffi

KOM - Multimedia Communications Lab
Prof. Dr.-Ing. Ralf Steinmetz (director)
Dept. of Electrical Engineering and Information Technology
Dept. of Computer Science (adjunct professor)
TUD – Technische Universität Darmstadt
Merckstr. 25, 64283 Darmstadt, Germany
Tel.+49 6151 166150, Fax. +49 6151 166152
www.KOM.tu-darmstadt.de

Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

- Introduction to Autonomic Computing
- Contribution
- Autonomic Element
- Summary

Motivation for Autonomic Computing

TECHNISCHE UNIVERSITÄT DARMSTADT

Complexity in Distributed Computing

Administration is a highly complex task:

- Lack of a centralized computing instance.
- Uncertainty and Unreliability (link-, node-failure).
- Highly complex decentralized data replication-, recovery-, and control flow mechanisms.

Complexity in Software Systems

Software programs and their environments are reaching tens of millions of lines of code:

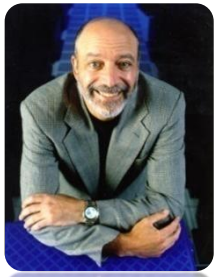
- Microsoft's Windows Vista approx. 50 million lines of code (LOC).
- Microsoft Windows XP approx. 40M LOC.
- Open-source projects, e.g. Eclipse “Europa” more than 17M LOC.

The Autonomic Computing Vision

TECHNISCHE UNIVERSITÄT DARMSTADT

Vision

Paul Horn (Senior Vice President IBM Research) envisions autonomic elements as…



"…individual system constituents that contain resources and deliver services to human and other autonomic elements. Autonomic elements will manage their internal behavior and their relationships with other autonomic elements in accordance with policies that humans or other elements have established."

Goals

Self-managing software and hardware systems.

Behavior in accordance with high-level policies from human administrators.

Reduced management complexity without shifting efforts from one point to another.

Shift from a system-centric towards a human-centric computing approach.

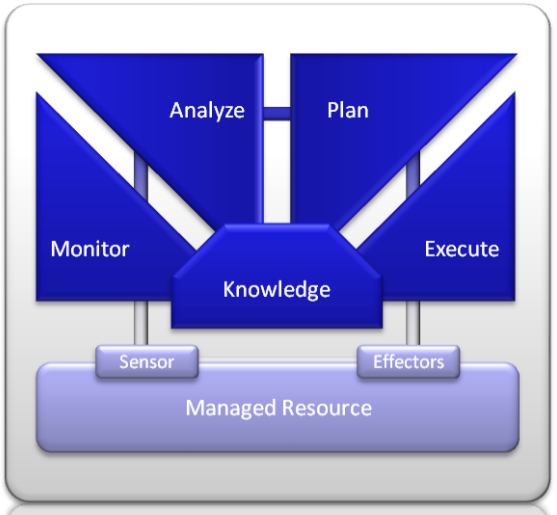
Characteristics of Autonomic Systems

TECHNISCHE UNIVERSITÄT DARMSTADT

Characteristics

- Self-Optimizing
- Self-Configuring
- Self-Awareness
- Self-Protecting
- Self-Healing
- Context-Aware
- Open
- Anticipatory

Autonomic Element



Mandatory functional unit

Implements the “MAPE” loop

Autonomic Manager and Managed Resource

Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

- Introduction to Autonomic Computing
- **Contribution**
- Autonomic Element
- Summary

Contribution

TECHNISCHE UNIVERSITÄT DARMSTADT

Bachelor Thesis

Title: “Autonomic Computing in Peer-to-Peer Systems”

Goals

Evaluates current state of the technology (maturity, challenges, obstacles, etc.).
Evaluates current state of research (university and industry).

Differences between all autonomic computing approaches.

Motivates all necessary functional units (MAPE loop).
Information about research in the specific fields.

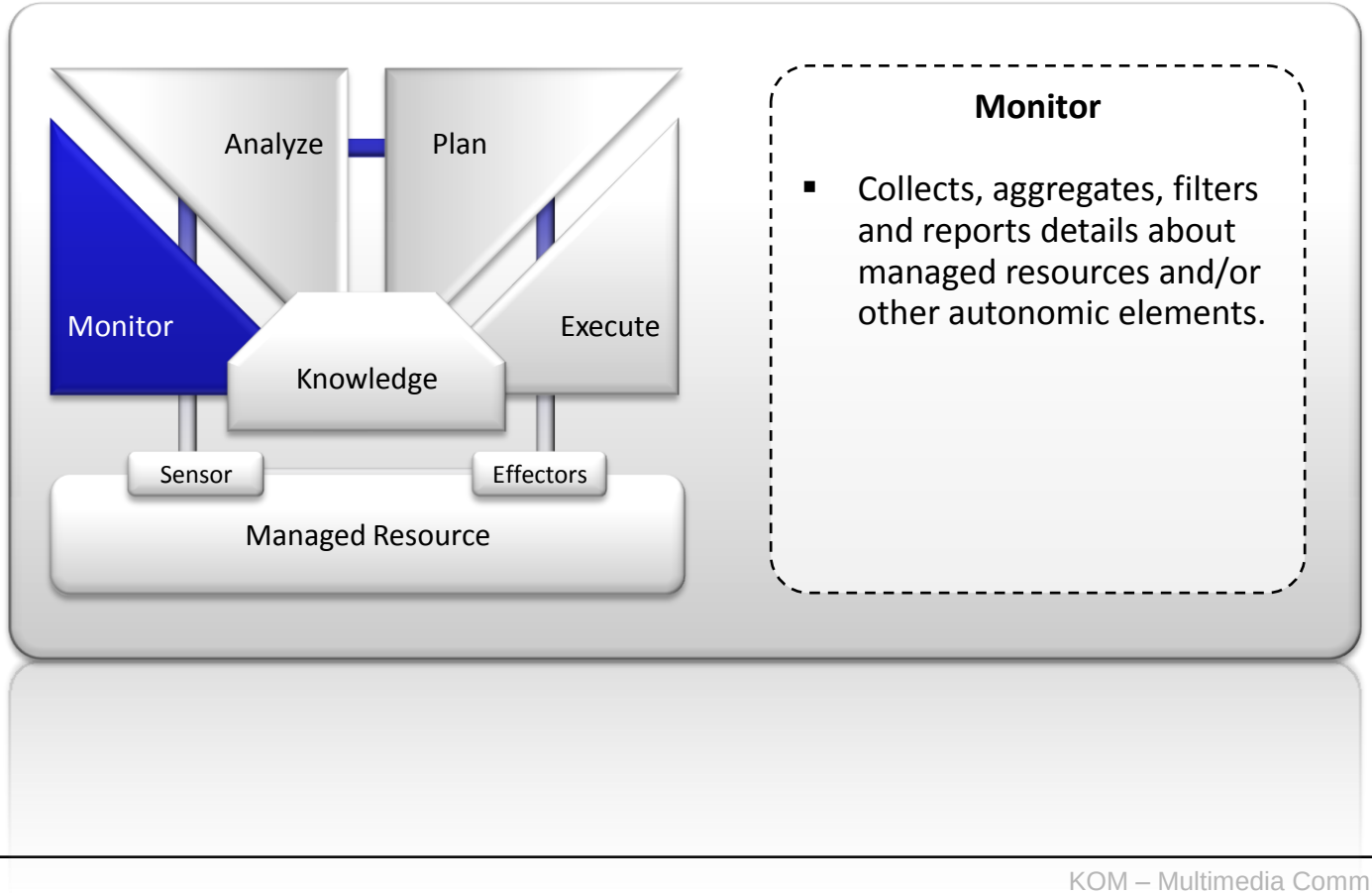
Shows and evaluates possible implementation approaches.

Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

- Introduction to Autonomic Computing
- Contribution
- **Autonomic Element**
- Summary

Autonomic Element - Monitoring



Monitor

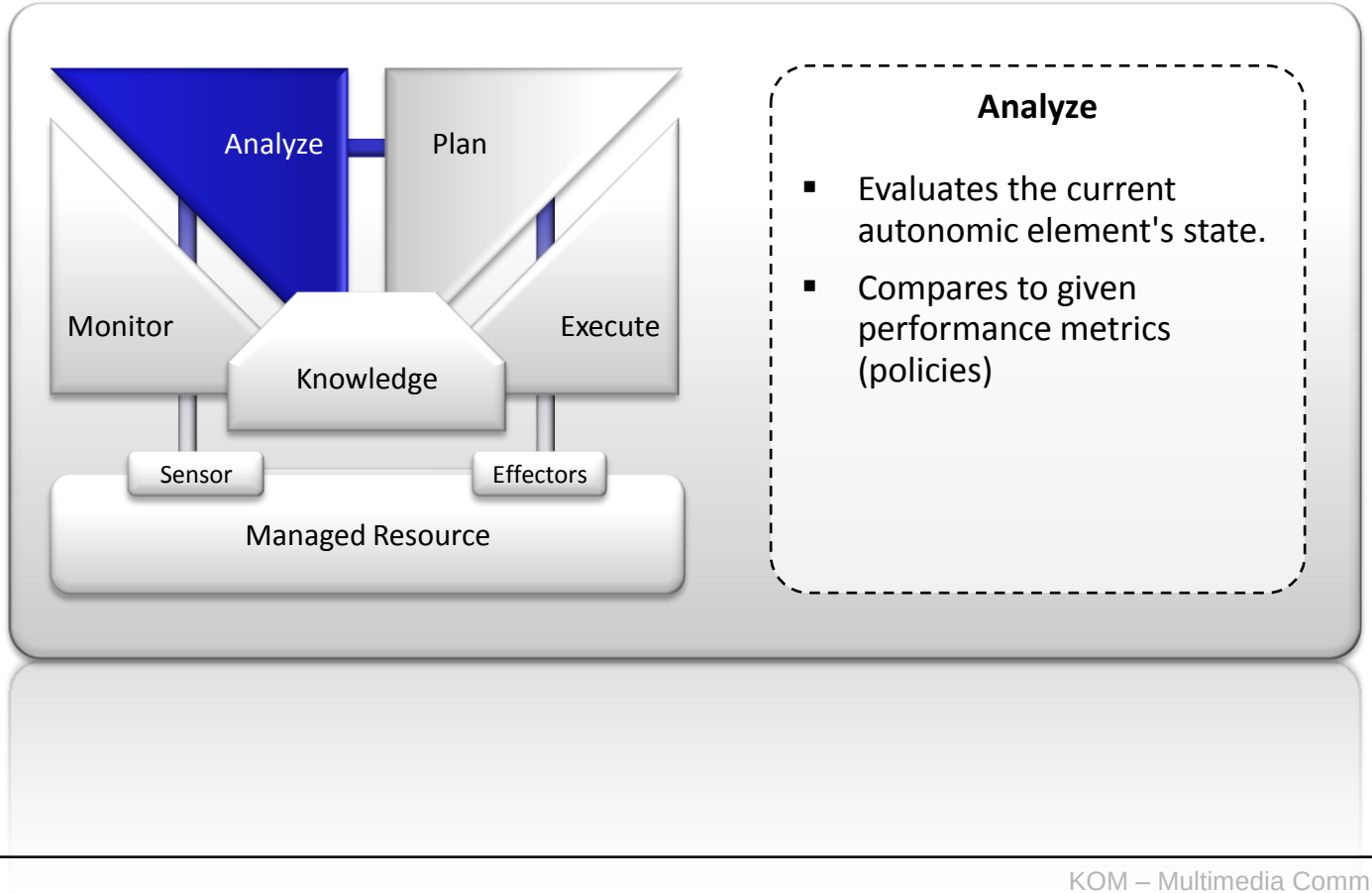
- Collects, aggregates, filters and reports details about managed resources and/or other autonomic elements.

Autonomic Element - Monitoring

TECHNISCHE UNIVERSITÄT DARMSTADT

Description	• Collecting, aggregating, filtering and reporting. • Information about managed resources and/or other autonomic elements.
Goals	• Gives an insight into the running system. • Knowledge-building tool for the future decisions.
Examples	• P2P-DIET, SkyEye.KOM, DASIS, T-MAN, • Intel Active Management Technology (AMT) • IBM System z10 Active Resource Monitoring (ARM)

Autonomic Element - Analyze



Analyze

- Evaluates the current autonomic element's state.
- Compares to given performance metrics (policies)

Autonomic Element - Analyze

TECHNISCHE UNIVERSITÄT DARMSTADT

Description

- Evaluates the current state against given metrics.
- Implements a policy engine.

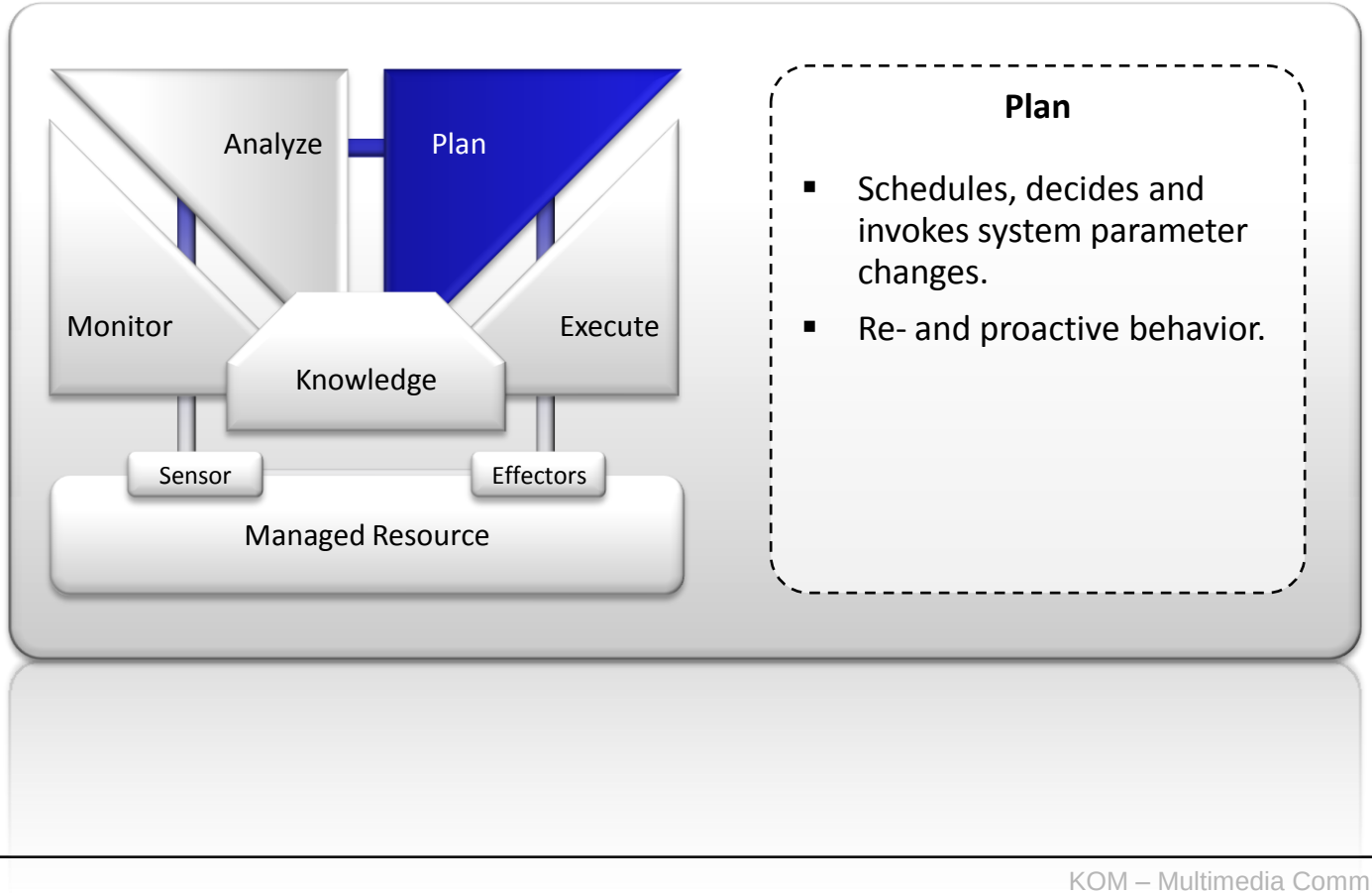
Goals

- Analyze if a component works according to the given quality expectations.
- Issues counteractions if service works not insufficiently.

Examples

- Policy Definition Languages: Autonomic Computing Policy Language (ACPL), XACML, WS-Policy, WS-Agreement
- Policy deployment, -evaluation processes
- Parameter and metric correlations.

Autonomic Element - Plan

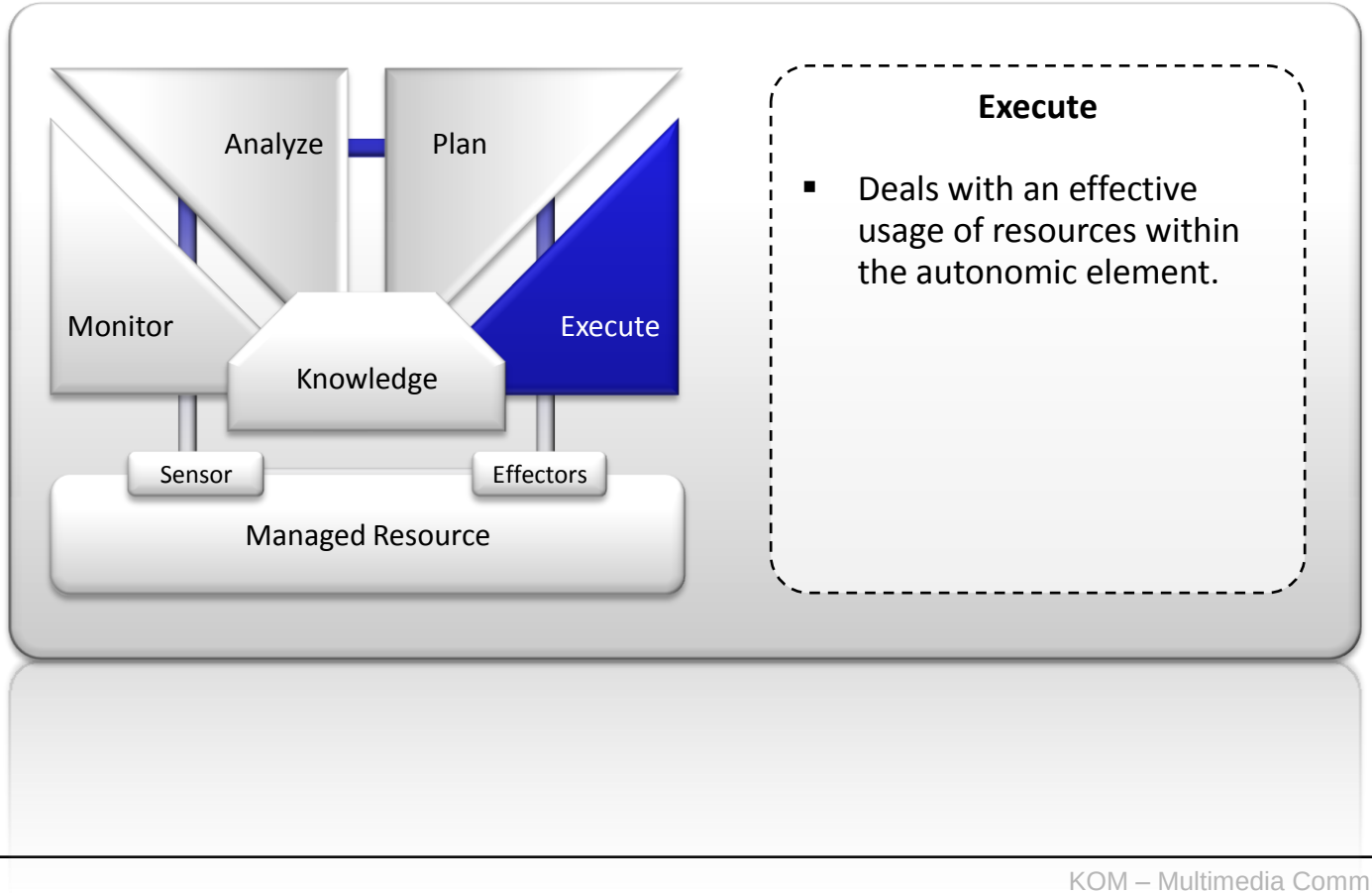


Autonomic Element - Plan

TECHNISCHE UNIVERSITÄT DARMSTADT

Description	• Schedules, decides, and invokes all necessary changes • reactively, or -even more desirable- proactively.
Goals	• Reach a performance function's global optimum.
Examples	• Genetic algorithms and genetic programming • Simulated Annealing (VFSA, S^3 Networks) • Tabu Search (Fixed Cellular Network Coverage) • Multi-Agent algorithms (Ant Colony Optimization)

Autonomic Element - Execute



Execute

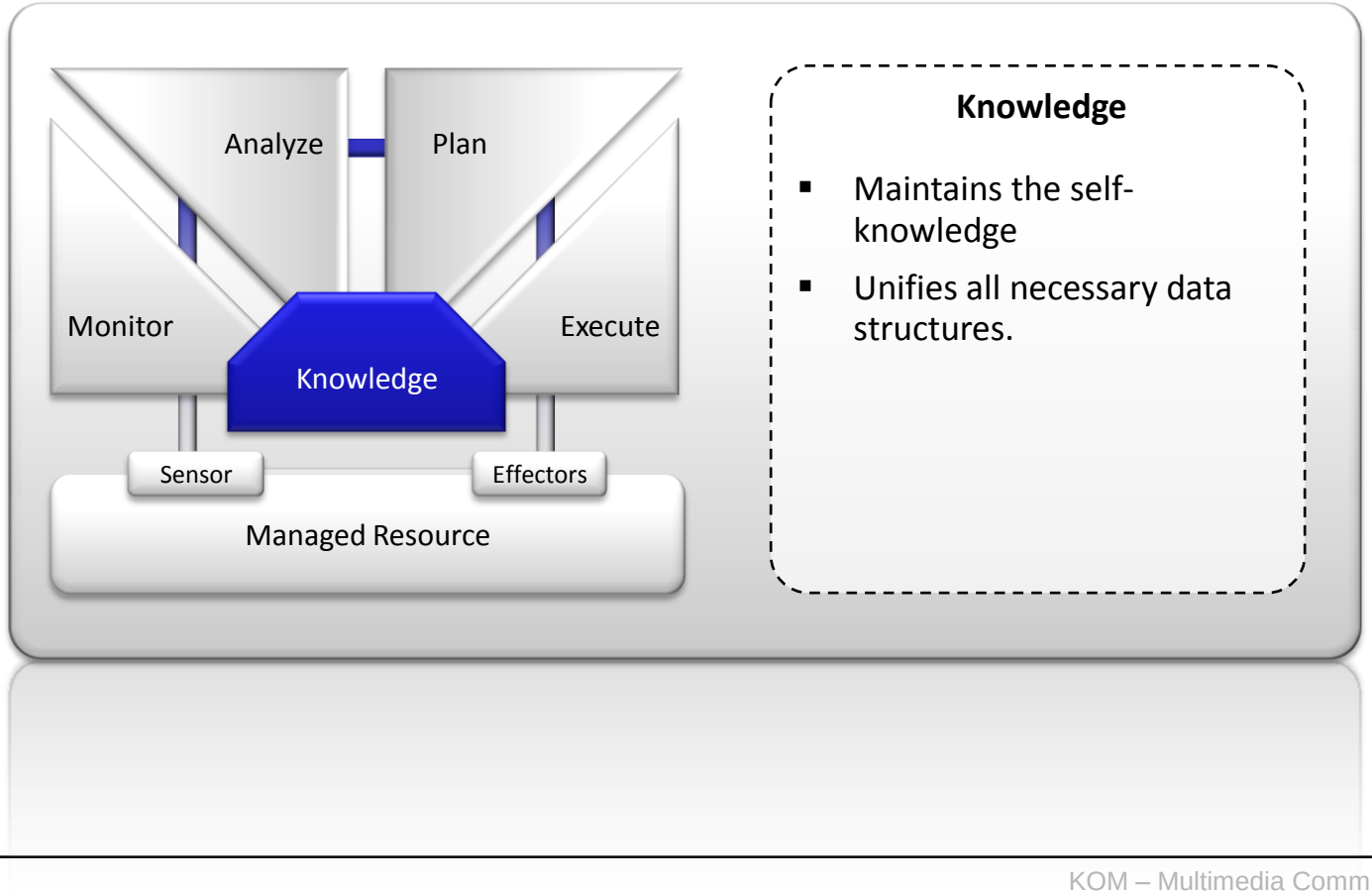
- Deals with an effective usage of resources within the autonomic element.

Autonomic Element - Execute

TECHNISCHE UNIVERSITÄT DARMSTADT

Description	• Deals with an effective usage of resources. • Controls external executions (subcontracting).
Goals	• AEs must act selfishly, while supporting to the entire Autonomic System with vital services. • Controlling on various management layers
Examples	• Peer Resource Layer (Wave Scheduler, HiPNOS) • Peer-to-Peer Network Layer (Popularity-aware prefetch caching, load balancing for multimedia streaming) • Internet Layer (GTap, biased neighbor selection,)

Autonomic Element - Knowledge



Knowledge

- Maintains the self-knowledge
- Unifies all necessary data structures.

Autonomic Element - Knowledge

TECHNISCHE UNIVERSITÄT DARMSTADT

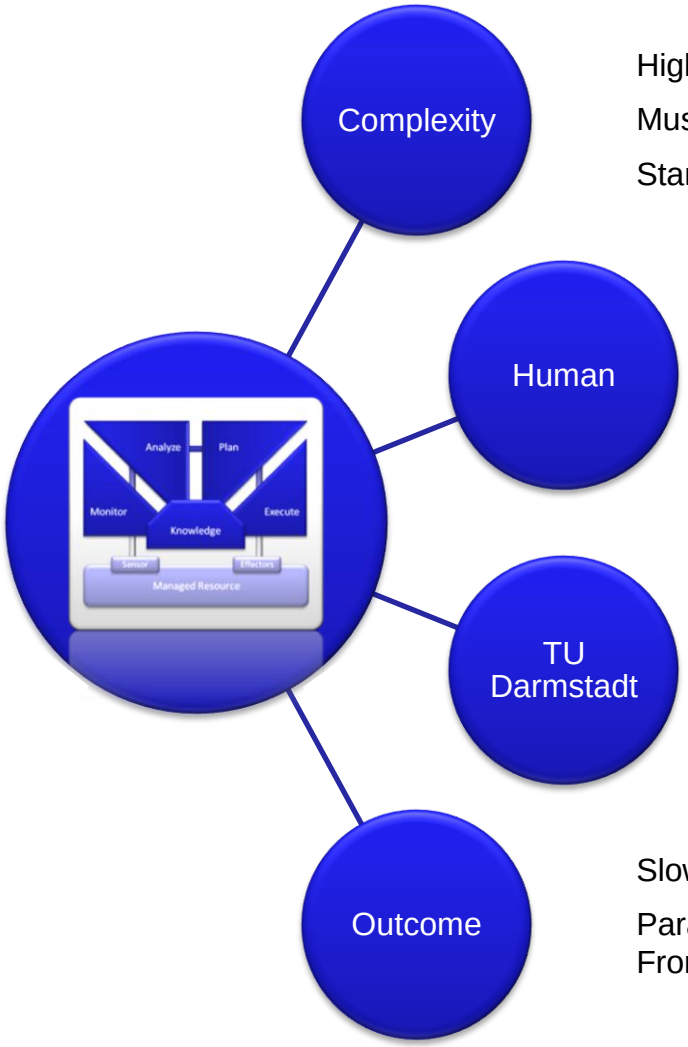
Description	• Unifies policies, restrictions, preferences, problem symptoms and -solving strategies, operating logs. • A central information repository.
Goals	• Improve all decisions by giving facts for sustainable and substantiated decisions.
Examples	• Self-Monitoring Information (System Task Library (STL)) • Problem Detection and Self-Diagnosis Knowledge Flow • Problem Resolution Knowledge (IBM DataCase System)

Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

- Introduction to Autonomic Computing
- Contribution
- Autonomic Element
- **Summary**

Summary



Highly complex vision.

Must reduce the management complexity in all fields.

Standards are inevitable.

Human-intervention in extraordinary situations.

Support for routine administration tasks.

Minimum hands-on intervention of administrators.

Iterative Autonomic Computing approach.

Focus on Peer-to-Peer Problems (Monitoring).

Leading in Autonomic Computing in P2P Systems.

Slow, but steady development process.

Paradigm shift:
From system-centric towards a human-centric approach.

Questions?

TECHNISCHE UNIVERSITÄT DARMSTADT

Thank you for your attention!