EMERGENT SOFTWARE SYSTEMS

Summer School

Barry Porter & Roberto Rodrigues Filho
School of Computing and Communications
Lancaster University

Funded by The Royal Society Newton Fund

THE ROYAL SOCIETY

Autonomic Computing and Emergent Software Systems

- **Autonomic Computing (AC)**
  - Motivation
  - IBM Concept
  - Autonomic Computing Architecture
- **Emergent Software Systems (ESS)**
  - Motivation
  - Defining “Emergent”
  - ESS Concept

Autonomic Computing

Autonomic Computing

- **Motivation**
  - **Software complexity**
    - Tens of thousands line of code systems (very large systems)
    - High number of interacting distributed modules
    - Highly dynamic operanting environments (uncertainty)

Autonomic Computing

- **Motivation**
  - **Human-led Management**
    - In case of failures, or implementing changes to system involves:
      - Monitoring the system;
      - Analysing systems data performance
      - Detecting anomalies or impacts changes might cause;
      - Isolating parts of the system that needs to be changed; and
      - Implementing the changes are increasingly cumbersome;

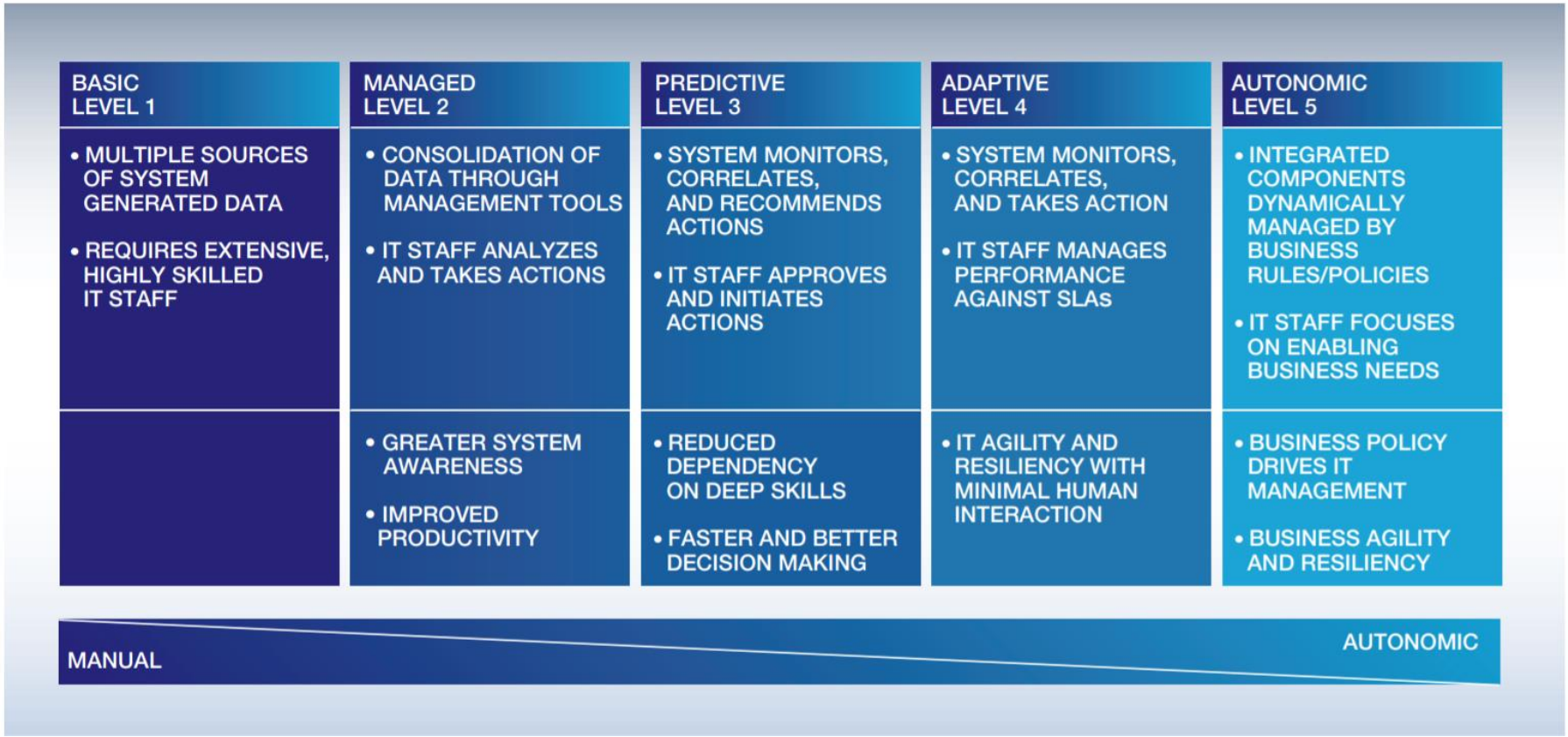
Autonomic Computing

- **Motivation**
  - **Human-led Management**
    - “Contemporary systems have reached a level of complexity that is beyond the human capacity to understand”

Autonomic Computing

- **IBM Concept**
  - Autonomic Computing (AC) is a term coined by IBM in the early 2000s.
  - AC has the goal to create self-managing systems;
    - Inspired by human body systems, AC systems may be divided into:
      - complete autonomous/self-managing modules with no human actions required;
      - semi-autonomous/self-managing modules that can operate without human actions, but a human operator could take control over the software actions;
      - complete human-dependent modules that work only guided by human operators;
  - Given high level goals, the self-managing parts of the system should take the necessary actions, with minimum human intervention, to satisfy the provided goals;

Autonomic Computing



Autonomic Computing

- **IBM Concept**
  - **Self-* properties**
    - Self-healing
    - Self-protecting
    - Self-optimising
    - Self-configuring

Autonomic Computing

- **IBM Concept**
  - **Self-* properties**
    - Self-healing
    - Self-protecting
    - Self-optimising
    - Self-configuring

Self-adaptive

Autonomic Computing

- **IBM Concept**
  - **Self-adaptive Systems**

Self-adaptive Systems are capable of adapting their internal behaviour to accommodate changes caused by systems stakeholders in the functional/non-functional requirements, or by the operating environment (e.g. fluctuaction of the incoming workload), whilst maintaining a high level provided goal.

Autonomic Computing

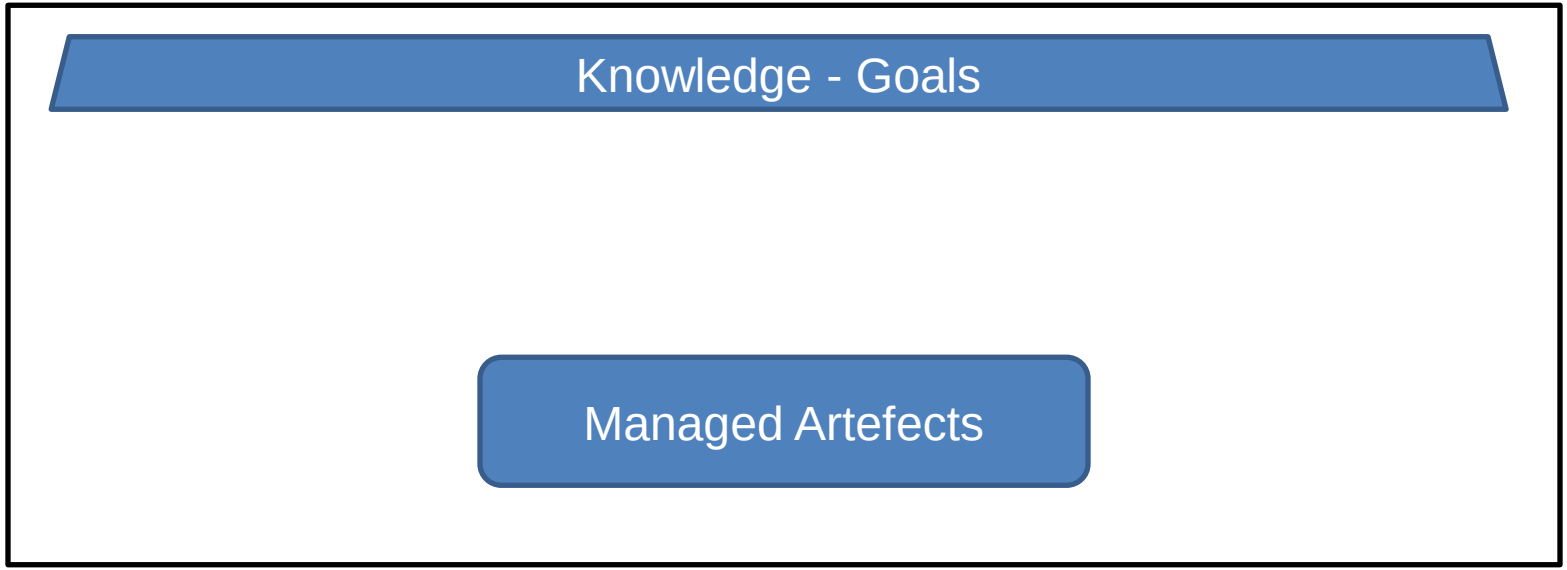
- Autonomic Computing Architecture

  MAPE-K – Monitoring, Analysing, Planning, Executing -- Knowledge

Autonomic Computing

- Autonomic Computing Architecture

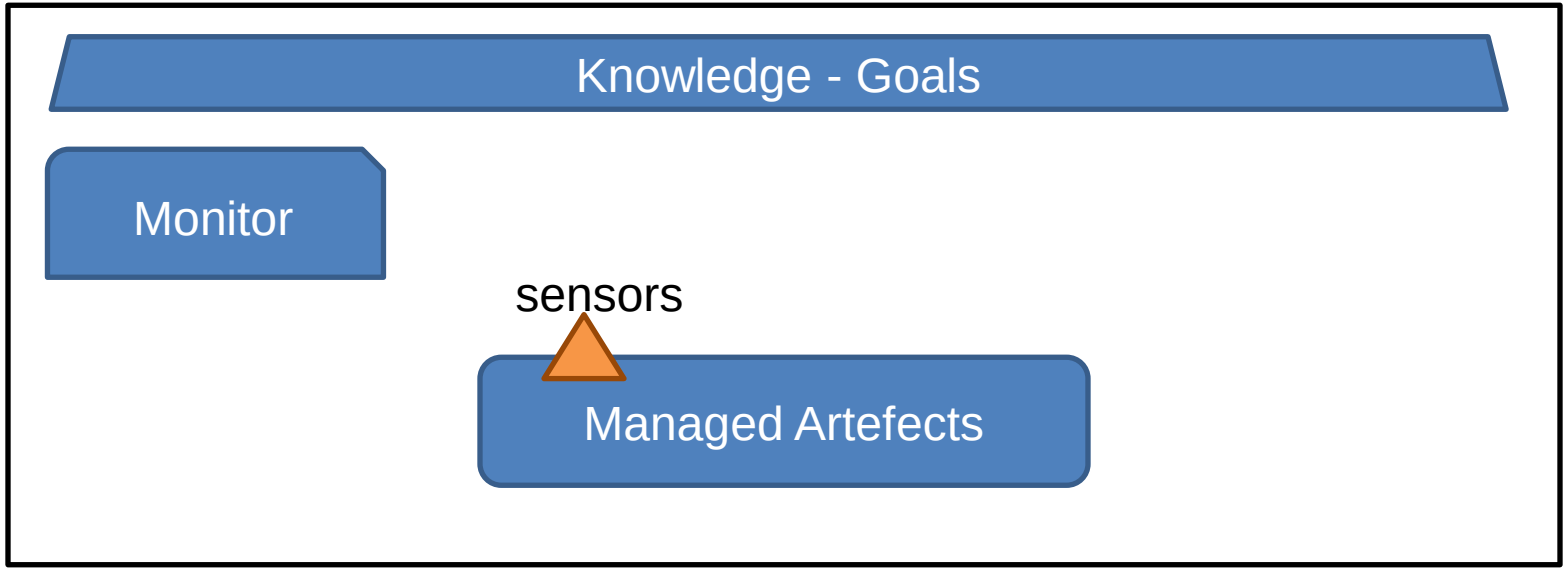
  MAPE-K – Monitoring, Analysing, Planning, Executing -- Knowledge



Autonomic Computing

- Autonomic Computing Architecture

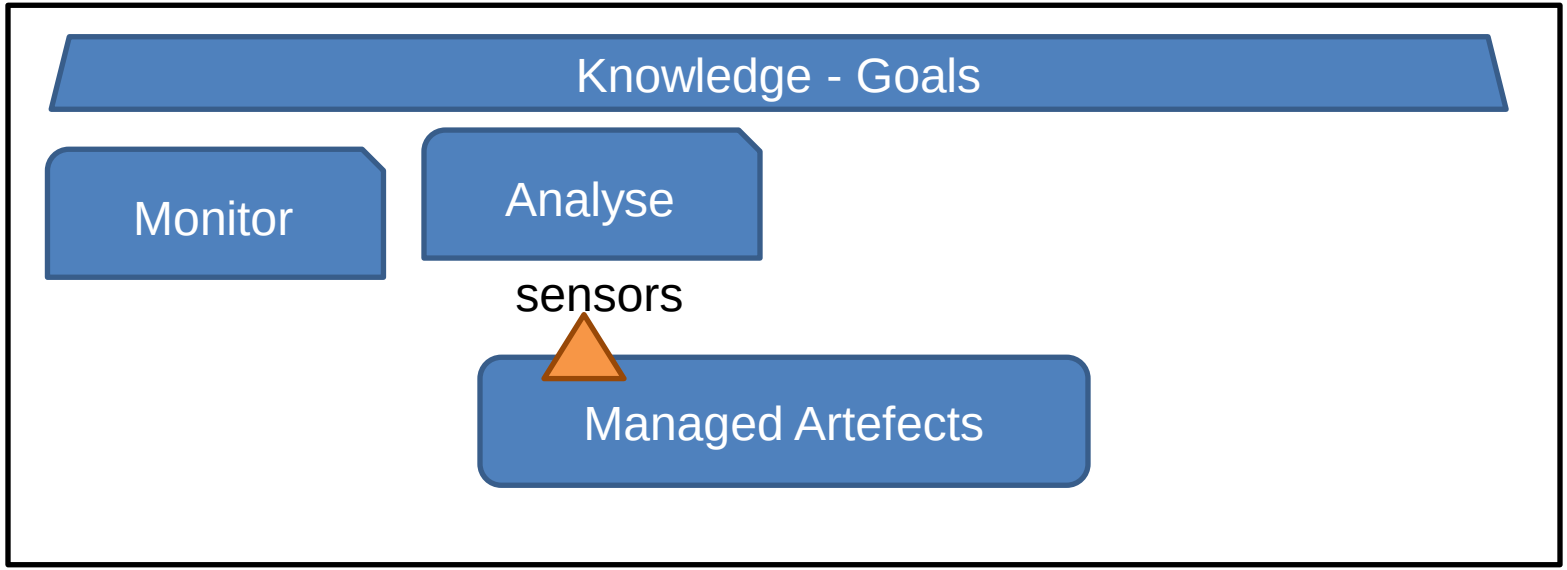
  MAPE-K – Monitoring, Analysing, Planning, Executing -- Knowledge



Autonomic Computing

- Autonomic Computing Architecture

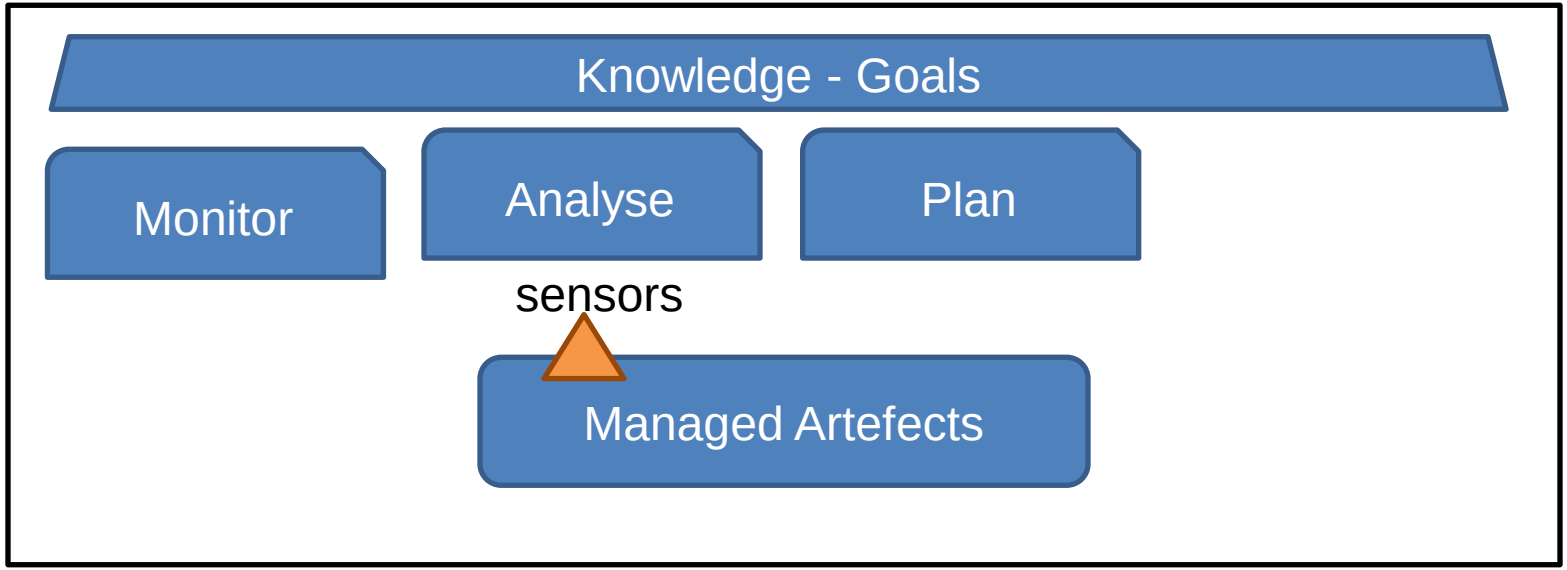
  MAPE-K – Monitoring, Analysing, Planning, Executing -- Knowledge



Autonomic Computing

- Autonomic Computing Architecture

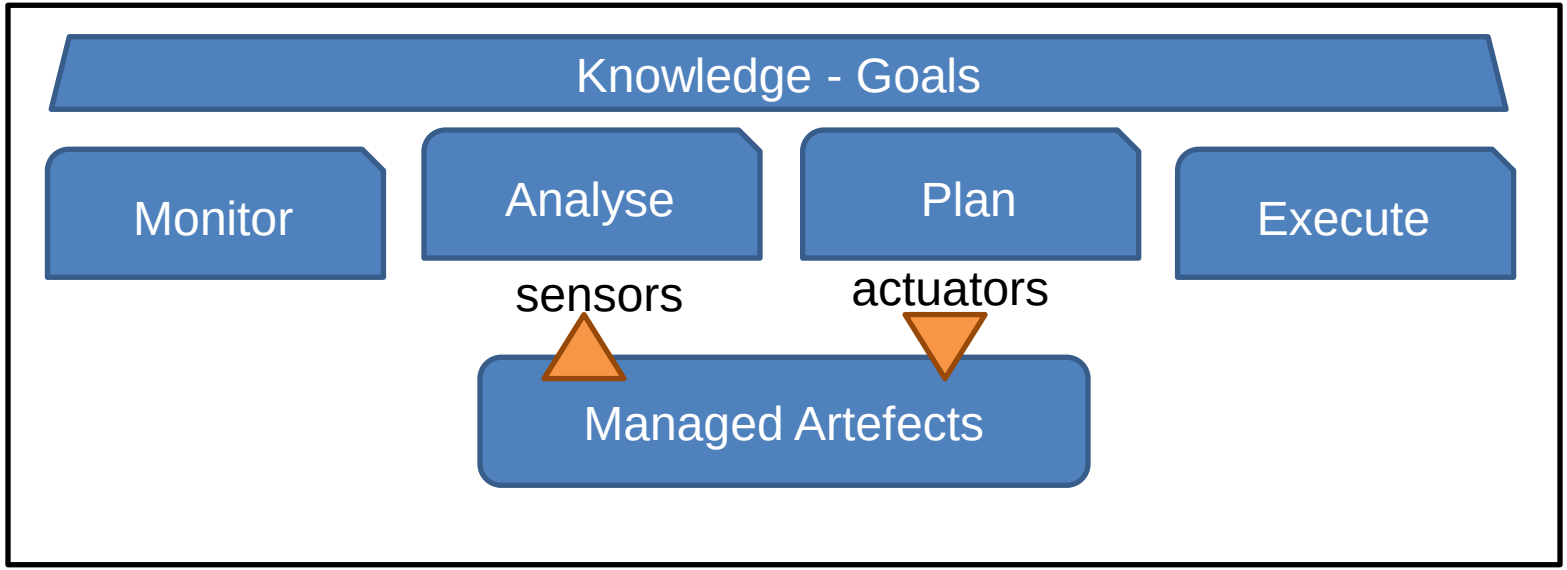
  MAPE-K – Monitoring, Analysing, Planning, Executing -- Knowledge



Autonomic Computing

- Autonomic Computing Architecture

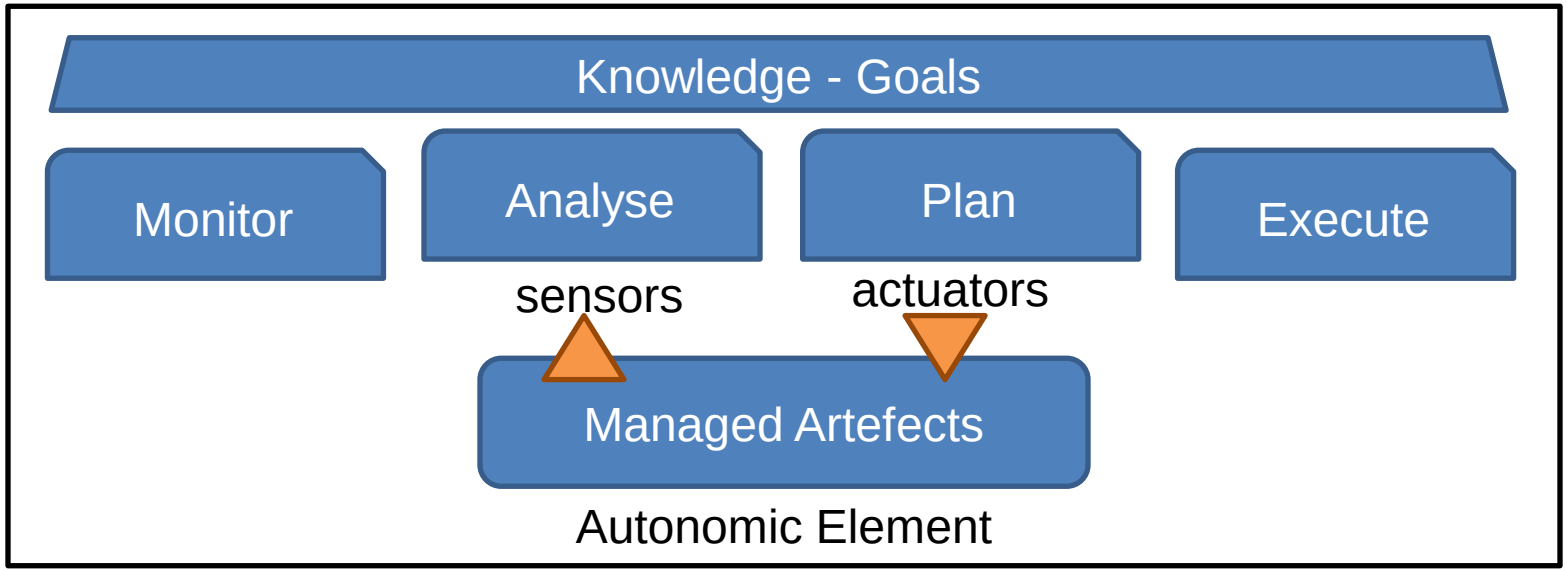
  MAPE-K – Monitoring, Analysing, Planning, Executing -- Knowledge



Autonomic Computing

- Autonomic Computing Architecture

  MAPE-K – Monitoring, Analysing, Planning, Executing -- Knowledge



Emergent Software Systems

Emergent Software Systems

- **Motivation**
  - **Developing AC systems is very cumbersome and laborious;**
    - Develop the system;
    - Identify Autonomic Elements;
    - Creating adaptation points (e.g.: feature models);
    - Defining the adaptation logic:
      - Manually describing when/to what configuration the system must adapt;
      - Manually expressing through logical expressions when and to what configuration the system must adapt;
      - Training the system to create a mathematical model or machine learning model that captures when and to what configuration to adapt;

Emergent System

- **Defining Emergent**
  - Emergent System is an approach to develop Autonomic Systems.
  - Emergent system:
    - starts with no domain-specific information;
    - builds its own understanding of its own architecture and operating environment with minimum human involvment;
    - does not require predefined models/adaptation rules that guide software adaptation at runtime;
    - does not require feature-models or any extra technology (other than the system code itself) to support seamless online adaptation.

Emergent System

- **Defining Emergent**
  - Emergent means:

Emergent System

- **Defining Emergent**
  - Emergent means:
    1. The composed system is bigger than the sum of its smaller parts;

Emergent System

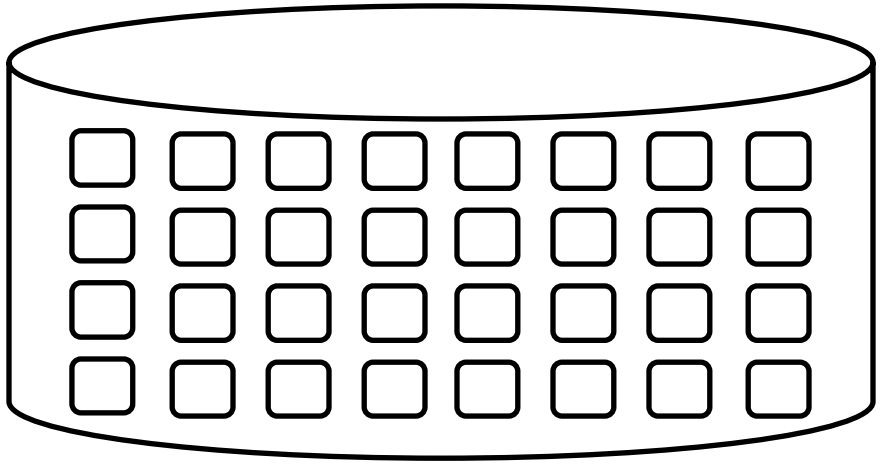
- **Defining Emergent**
  - Emergent means:
    1. The composed system is bigger than the sum of its smaller parts;
    2. The system is able to find unexpected optimal compositions;

Emergent System

- **Defining Emergent**
  - Emergent means:
    1. The composed system is bigger than the sum of its smaller parts;
    2. The system is able to find unexpected optimal compositions;
    3. The system is a result of its operating conditions.

Emergent System

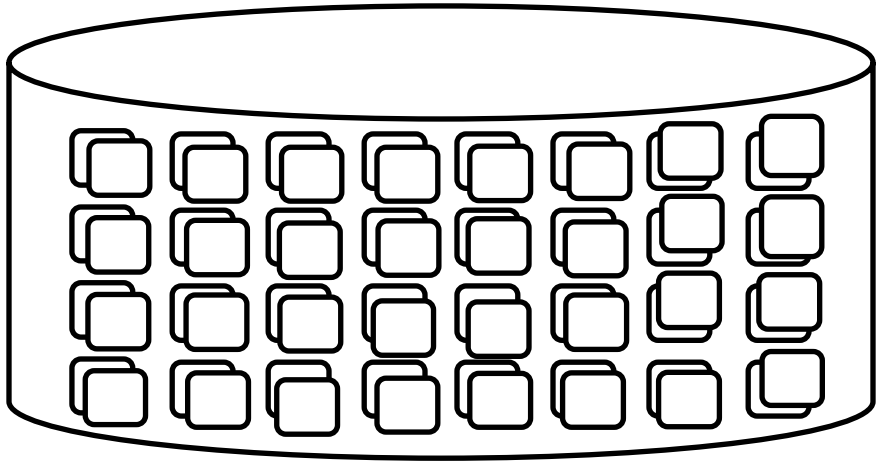
- **Emergent System Concept**
  - We consider a pool of small units of software behaviour:



Repository

Emergent System

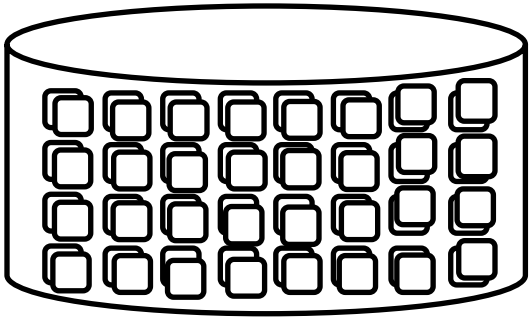
- **Emergent System Concept**
  - We consider a pool of small units of software behaviour (and variantions):



Repository

Emergent System

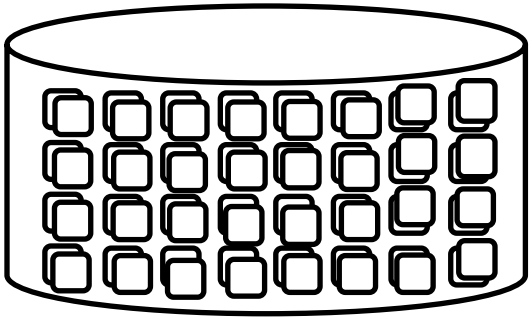
- **Emergent System Concept**
  - We consider a pool of small units of software behaviour (and variantions);
  - We consider the small units have information of their dependencies, etc. (component-based model);



Repository

Emergent System

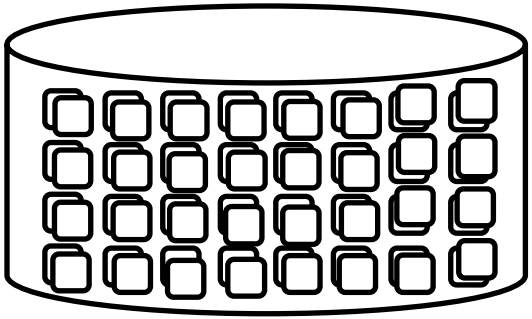
- **Emergent System Concept**
  - We consider a pool of small units of software behaviour (and variantions);
  - We consider the small units have information of their dependencies, etc. (component-based model);



Repository

Emergent System

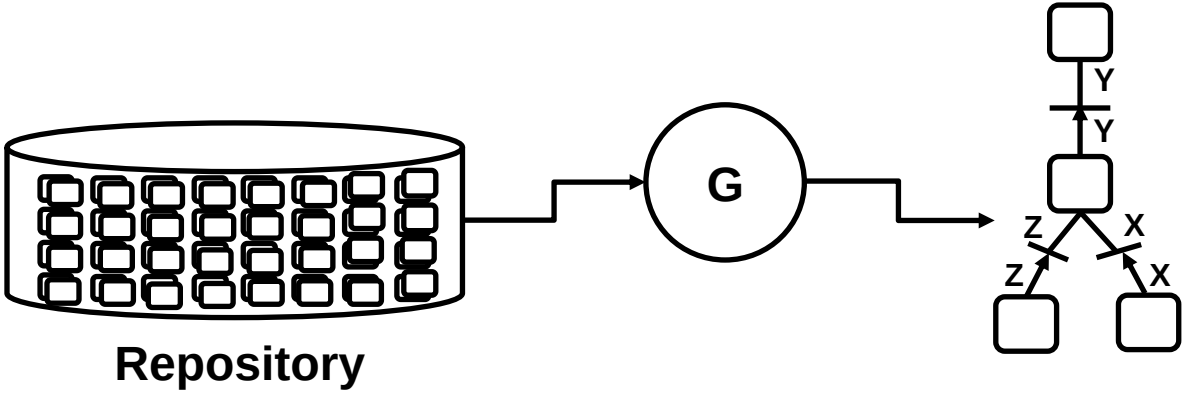
- **Emergent System Concept**
  - We consider a pool of small units of software behaviour (and variantions);
  - We consider the small units have information of their dependencies, etc. (component-based model);



Repository

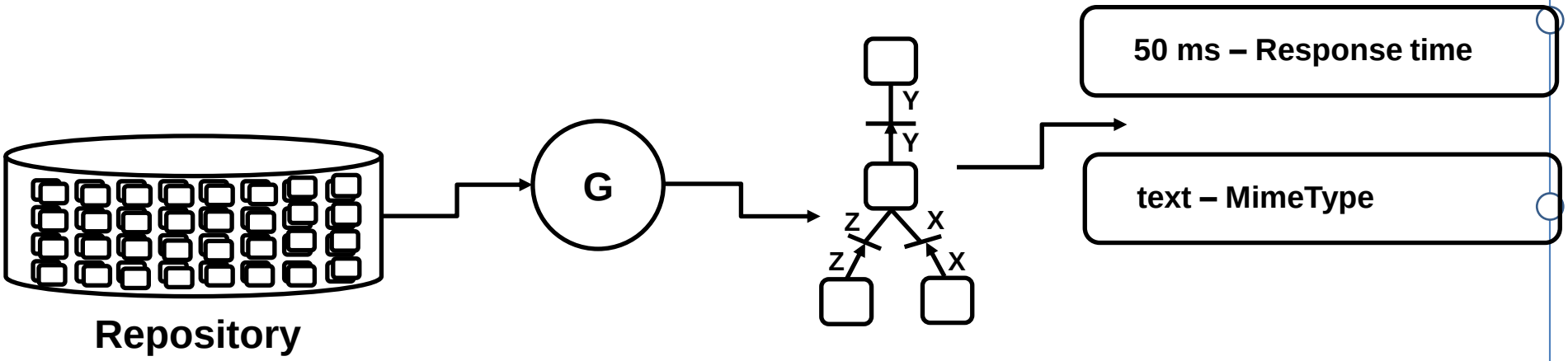
Emergent System

- **Emergent System Concept**
  - We consider a pool of small units of software behaviour (and variantions);
  - We consider the small units have information of their dependencies, etc. (component-based model);
  - We consider a high level goal;



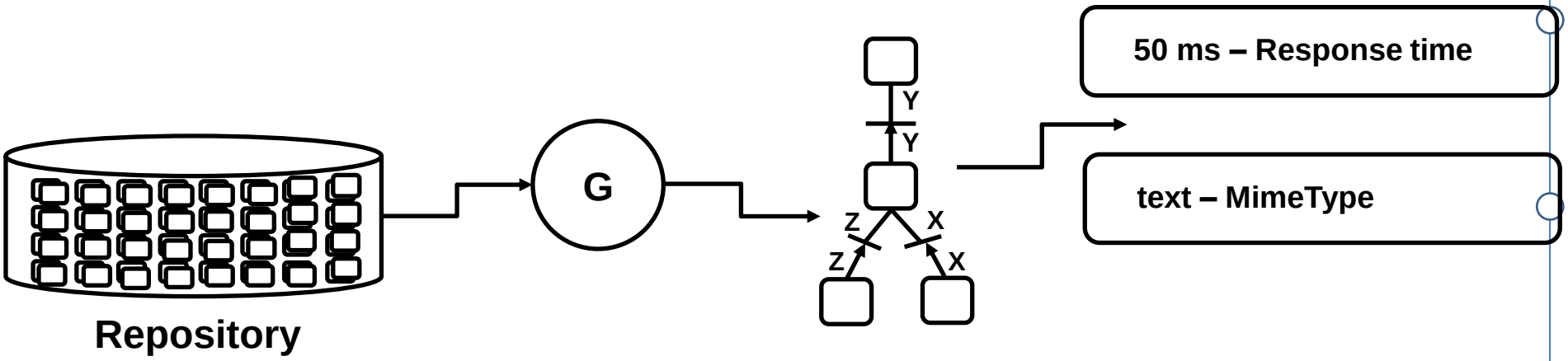
Emergent System

- **Emergent System Concept**
  - ...
  - We consider the components emit **Metrics** and **Events**



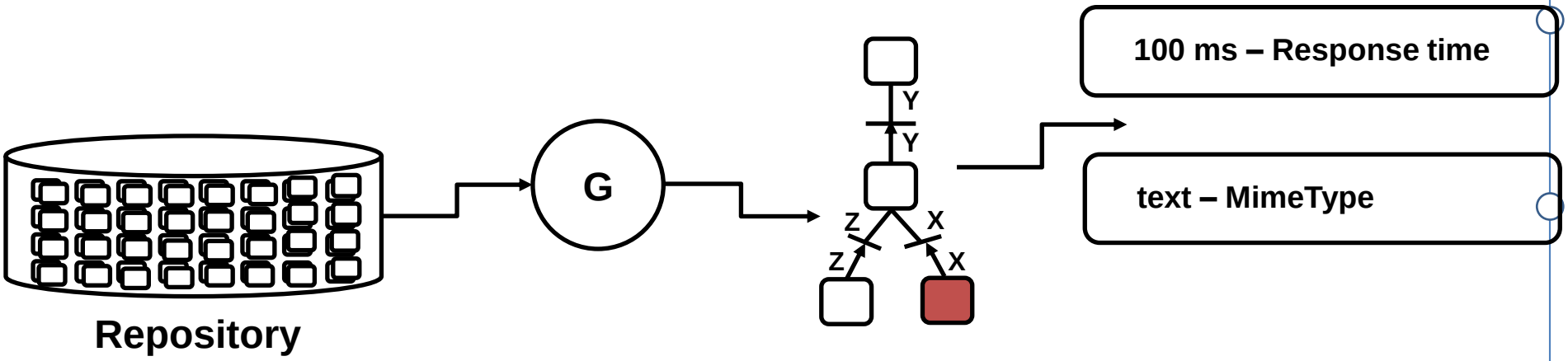
Emergent System

- **Emergent System Concept**
  - ...
  - We consider the components emit **Metrics** and **Events**
  - The goal of Emergent Systems is find the best available composition that maximises the provided goal G



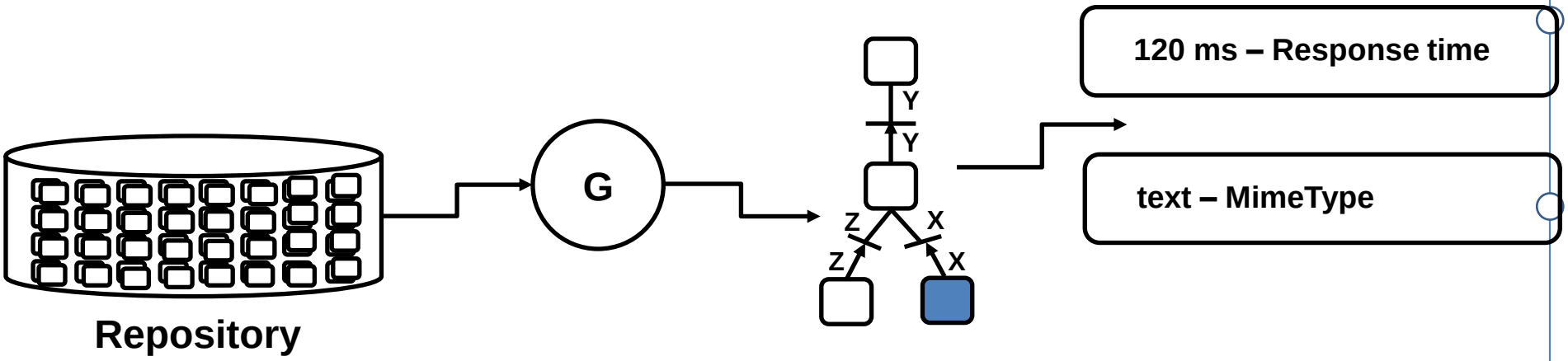
Emergent System

- **Emergent System Concept**
  - ...
  - We consider the components emit **Metrics** and **Events**
  - The goal of Emergent Systems is find the best available composition that maximises the provided goal G



Emergent System

- **Emergent System Concept**
  - ...
  - We consider the components emit **Metrics** and **Events**
  - The goal of Emergent Systems is find the best available composition that maximises the provided goal G



Summary

- **Autonomic Computing (AC)** is a research area that aims at creating self-managing systems (with minimum of human involvement in certain stages of the software management);
- **AC** has gained significant importance considering the increasing levels of complexity in managing systems;
- **Emergent Software Systems (ESS)** is an approach to reduce the effort in creating autonomic systems;
- **ESS** focus on composition rather than adaptation. Adaptation unfolds as a consequence of software composition;
- **ESS** are able to build their own understanding of their software architecture and their executing environment.

Practical Assignment

For this practical assignment, we expect you to use our component that implements the Assembly process described in Lecture 2, to compose and adapt a calculator software we make available. We also expect you to expand the provided calculator software to add new features to it and thus experiencing how the event model work.