



Sofia University "St. Kl. Ohridski"
Department of Software Technologies



Reference Architecture for Self-Adaptive Microservice Systems

KRASIMIR BAYLOV, ALEKSANDAR DIMOV

Outline

Microservices and Self-Adaptive Systems

Problem Statement

Our solution

Summary

Microservices

Small autonomous services that work together¹

Opposite to monolith application

Independent deployments

Effective scalability

Rapid delivery

...

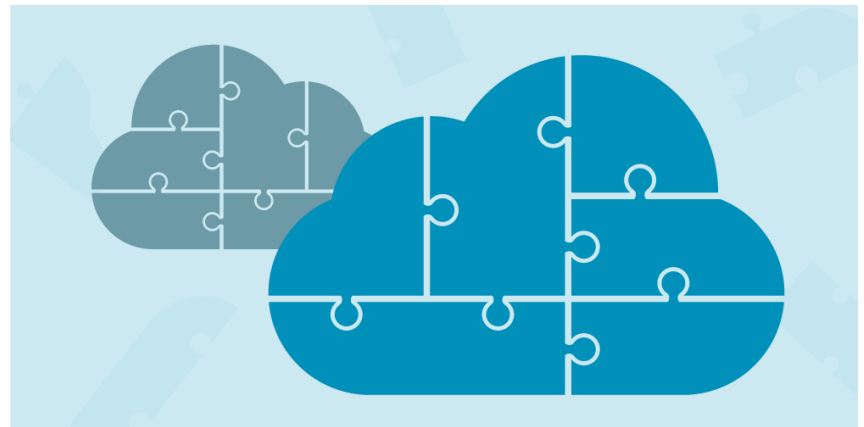


Image: www.bettercloud.com

¹ S. Newman, *Building Microservices*, O'Reilly Media, Inc., 2015.

Microservice pros and cons

Everything comes at a certain cost

Increased complexity

Constant monitoring

Complexity of distributed systems

Microservices complexity

- Distributed monitoring
- Administration
- Detecting and fixing failures

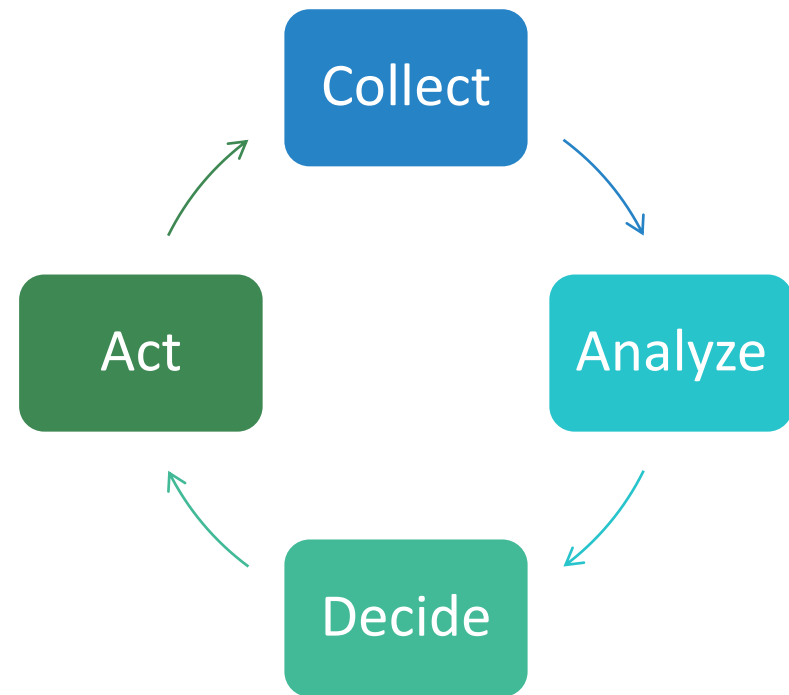
Self-Adaptive Systems

Response to increasing complexity of software systems

Follow high level goals

Evolve with limited or no human supervision

Using control loop to operate



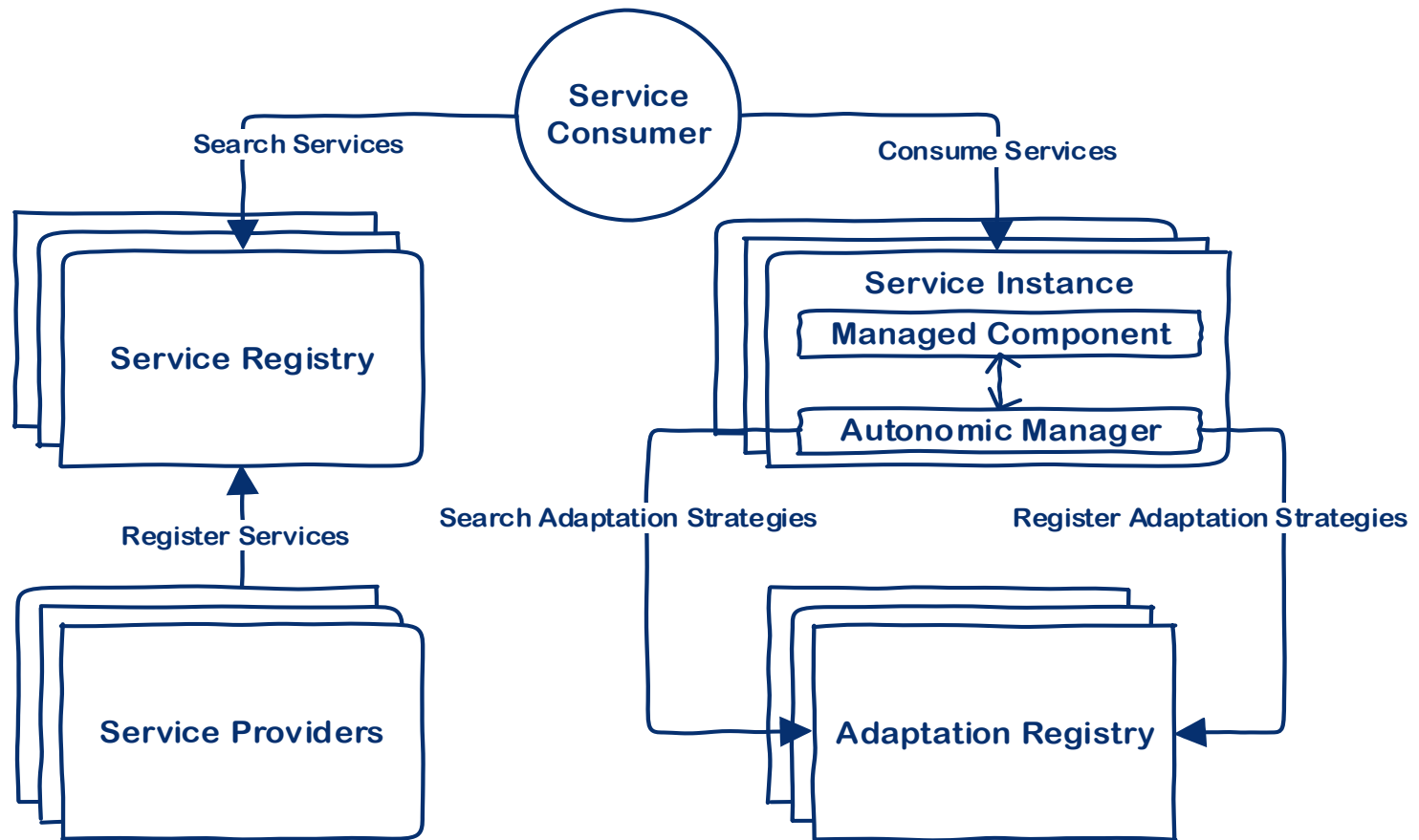
Our Solution

Provide a reference architecture

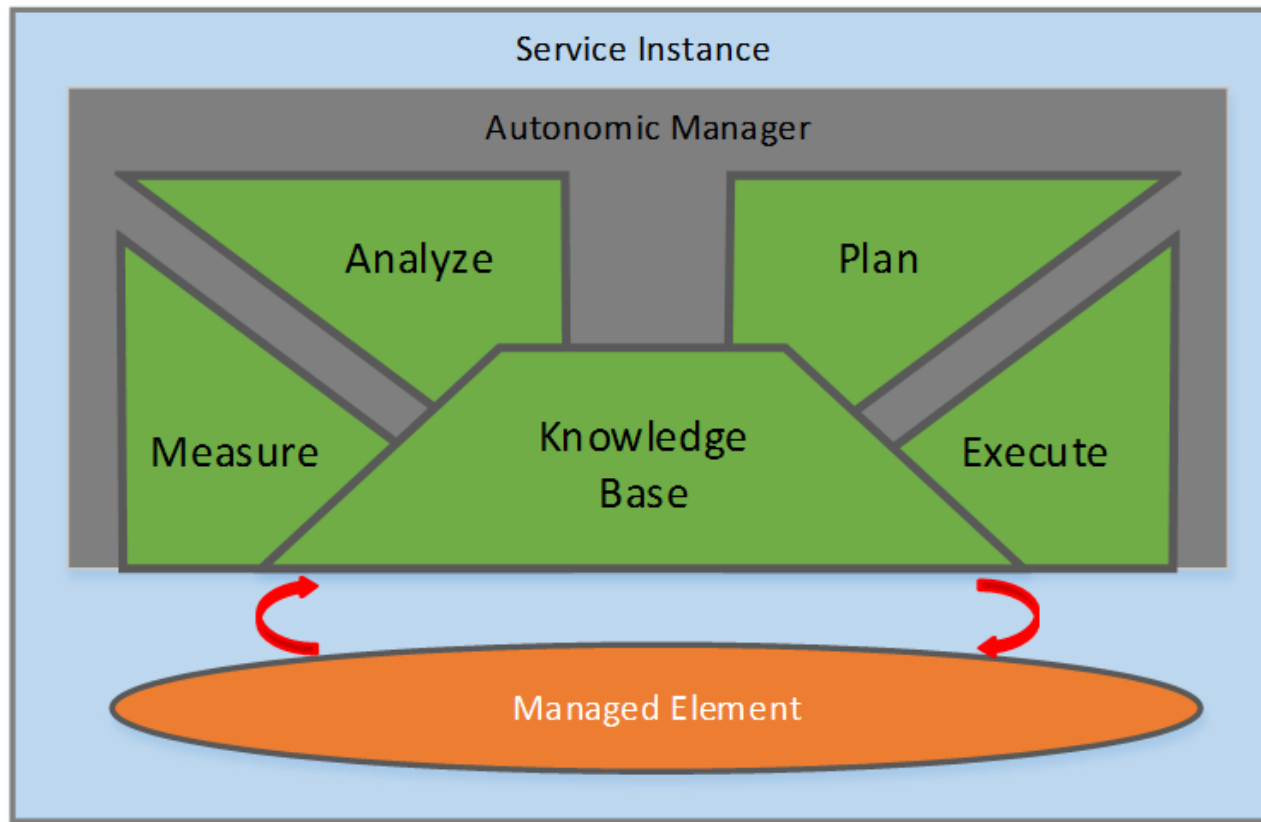
Combining microservices and self-adaptation capabilities

Based on SOA (Service oriented architecture)

Reference Architecture



Service instance implementation



Reference architecture instantiation

Consider a basic e-Trade micro-service use case

Initially it does not apply self-adaptive mechanisms

Lets consider an overload of the shipping module



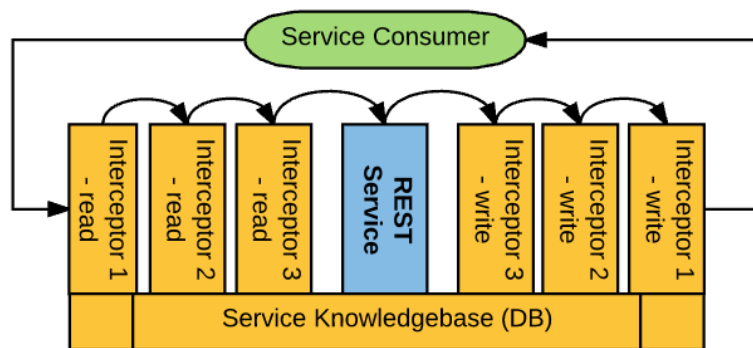
Possible implementation according to the reference architecture

Based on Jersey Java Library

Apache Derby is used for Service Knowledgebase

Interceptors apply changes to services

They have access to a specific object, called the context of the service and can manipulate it at runtime



Summary

Reference architecture for self-adaptive microservice systems

- Flexible
- Doesn't require all services to be self-adaptive
- Supports distributed knowledge sharing for adaptation mechanisms

Based on SOA, but introduces additional components to support self-adaptation

- Autonomic manager
- Adaptation registry

Directions for Future Research

High level goals that include cooperation between multiple services

Protocol for data exchange

Testing against real systems

Questions



- Krasimir Baylov (krasimirb@uni-sofia.bg)
- Aleksandar Dimov (aldi@fmi.uni-sofia.bg)