08.01.2024

Warszawa

Pracownia Dyplomowa Magisterska

Autonomiczne pętle sterowania i wykorzystanie operatorów Kubernetes na potrzeby zarządzania cyklem życia usług skonteneryzowanych (CNF).

Andrzej Gawor

Opiekun: dr. Inż. Dariusz Bursztynowski

# Cele pracowni

1. Próba określenia zasad modelowania pętli autonomicznej opartej o wzorzec CRD (Custom Resource Definition) oraz operatory Kubernetes, wzorowanej na ogólnym podejściu ETSI ENI wg dokuemntu z punktu 1.
2. Zapoznanie się ze sposobami komunikacji pętli z zewnętrznymi komponentami
3. Zapoznanie się ze sposobem modelowania polityk w zewnętrznej aplikacji.
4. Przygotowanie architektury platformy.
5. Przygotowanie przykładu na podstawie, którego można zaprezentować demo zamodelowanych pętli.
6. Iteracyjna faza developmentu platformy (opracowanie konwencji ogólnych, konwencji nazewnictwa, zdefiniowanie granic co jest zakresem kodu operatora, a co polityką w OPA itp..)
7. Próba dojścia do data-driven operatorów.
8. Udokumentowanie platformy i stworzenie instrukcji jej użytkowania.

# Rezultaty

## Zapoznanie się ze sposobami komunikacji pętli z zewnętrznymi komponentami

A diagram of a function

Description automatically generated

Zbadano pod tym kątek komunikacje HTTP oraz gRPC.

A screen shot of a computer screen

Description automatically generated

Komunikacja gRPC opisana w <https://github.com/0x41gawor/grpc-demo>.

## 2.2 Zapoznanie się ze sposobem modelowania polityk w zewnętrznej aplikacji.

Tutaj zapoznano się z rozwiązaniem Open Policy Agent: <https://github.com/0x41gawor/pcrf-opa>, które można wykorzystać w implementacji platformy do definiowania polityk poza kodem operatora.

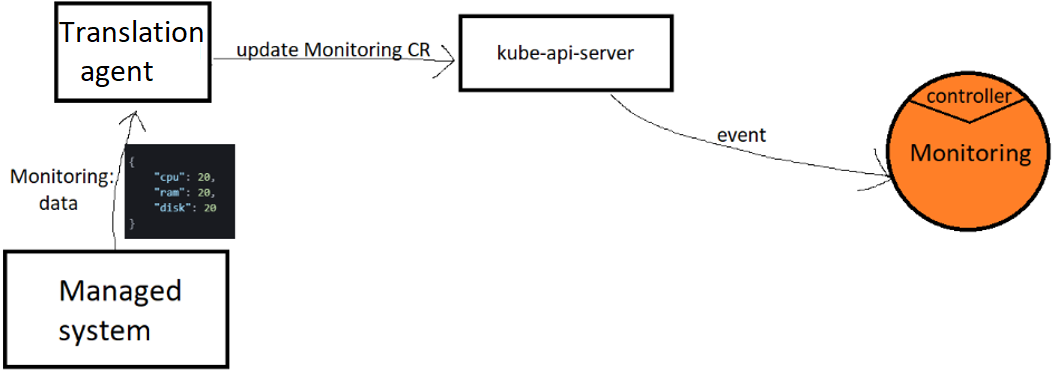
A screenshot of a computer program

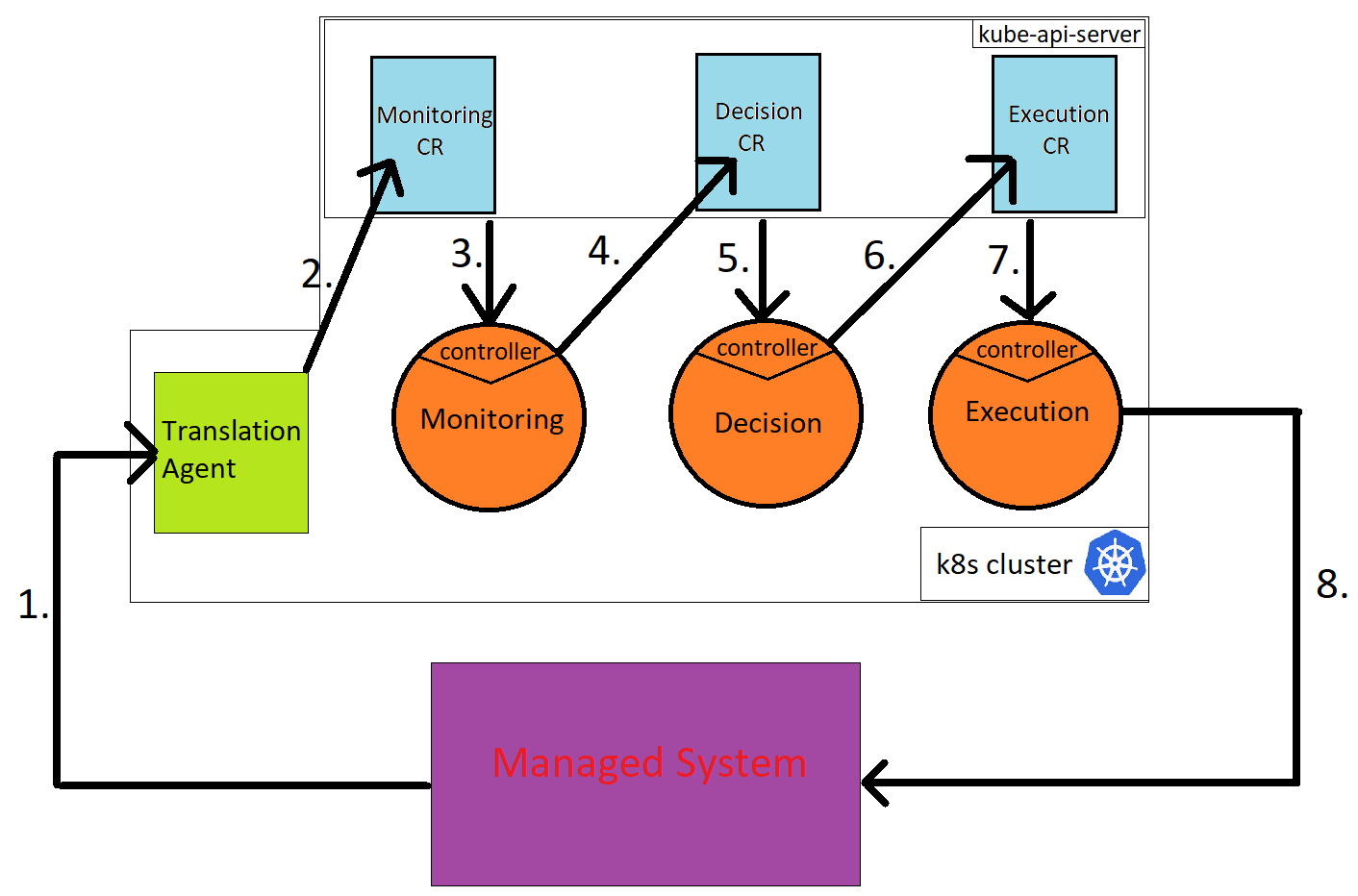
Description automatically generated

## 2.3 Przygotowanie architektury platformy (od strony implementacyjnej)

A screenshot of a computer

Description automatically generated





1. **Managed System** sends some externally defined data to **Translation-Agent**. **Translation-Agent** translates it to our ecosystem understandable form and..
2. **Proxy-Pod** sends Update of Monitoring CR to the **kube-api-server**. It triggers that ...
3. **kube-api-server** sends update event to the **Monitoring Controller**. **Monitoring Controller** performs monitoring logic (e.g. data translation, normalization) and ...
4. **Monitoring Controller** sends Update of Decision CR to the **kube-api-server**. It triggers that ...
5. **kube-api-server** sends update event to the **Decision Controller**. **Decision Controller** performs decision logic.
6. **Decision Controller** sends Update of Execution CR to the **kube-api-server**. It triggers that ....
7. **kube-api-server** sends update event to the **Execution Controller**. **Execution Controller** performs execution logic which are ..
8. The action is performed on the **Managed System**.

## 2.4 Przygotowanie przykładu na podstawie, którego można zaprezentować demo zamodelowanych pętli.

Przykład opisano tu: <https://github.com/0x41gawor/pdmgr/tree/master/mmet>

A map of poland with black text and black text

Description automatically generated

# Przyszłe kroki

1. Iteracyjna faza developmentu platformy (opracowanie konwencji ogólnych, konwencji nazewnictwa, zdefiniowanie granic co jest zakresem kodu operatora, a co polityką w OPA itp..)
2. Próba dojścia do data-driven operatorów.
3. Udokumentowanie platformy i stworzenie instrukcji jej użytkowania.