

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH



Instytut Instytut telekomunikacji

Praca dyplomowa magisterska

na kierunku Telekomunikacja
w specjalności Teleinformatyka i Zarządzanie w Telekomunikacji

Stworzenie platformy do modelowania i uruchamiania zamkniętych pętli
sterowania w Kubernetes

Andrzej Gawor

Numer albumu 300528

promotor

dr inż. Dariusz Bursztynowski

WARSZAWA 2025

Sworzenie platformy do modelowania i uruchamiania zamkniętych pętli sterowania w Kubernetes

Streszczenie. Projekt opisany w niniejszej pracy skupia się na zaproponowaniu oraz przedstawieniu implementacji architektury platformy, która pozwala na modelowanie oraz uruchamianie zamkniętych pętli sterowania w Kubernetes. Genezą projektu jest praca jednego z komitetów ETSI o nazwie "ENI - Experiential Networked Intelligence", która skupia się na ułatwieniu pracy operatora sieci telekomunikacyjnych wykorzystując mechanizmy sztucznej inteligencji w zamkniętych pętlach sterowania. ENI w jednym ze swoich dokumentów dokonuje przeglądu zamkniętych pętli sterowania znanych ludzkości z innych dziedzin.

Naturalnym następnym krokiem jest zaproponowanie platformy, na której operator mógłby takowe pętle projektować oraz uruchamiać. W tym celu zdefiniowano zestaw wymagań oraz założeń dla takiego systemu. Jako środowisko uruchomieniowe wybrano Kubernetes z racji, że jest to system dobrze znany w społeczności oraz sam natywnie używa zamkniętych pętli sterowania. Następnie przeprowadzono obszerną analizę jak za pomocą mechanizmów rozszerzania Kubernetes takich jak "Custom Resources" oraz "Operator" pattern można stworzyć framework umożliwiający modelowanie zamkniętych pętli sterowania. Praca opisuje powstałą platformę, jej architekturę, semantykę składni w definiowanych obiektach, zasady działania, integracje z zewnętrznymi systemami oraz instrukcję jej użytkowania. Omówiona została również implementacja platformy, technologie za nią stojące oraz decyzje podjęte podczas jej powstawania. Finalnie przedstawiono również test działania platformy w praktyce wykorzystując do tego emulator systemu 5G jakim jest Open5GS w połączeniu z UERANSIM. Pracę podsumuje lista wniosków oraz potencjalnych dróg rozwoju platformy.

Słowa kluczowe: Zamknięte pętle sterowania, Kubernetes, Zarządzanie sieciami telekomunikacyjnymi, Automatyzacja, Go, Open5GS, Mikroserwisy

Creation of a platform for designing and running closed control loops in Kubernetes

Abstract. The project described in this thesis focuses on proposing and implementing a reusable architecture (hereafter referred to as "the Platform") that enables the modeling and execution of closed control loops in Kubernetes. The genesis of the project lies in the work of one of the ETSI committees called "ENI - Experiential Networked Intelligence," which aims to simplify the work of telecommunications network operators by leveraging artificial intelligence mechanisms in closed control loops based on metadata-driven and context-aware policies. In one of its documents, ENI reviews closed control loops known to humanity from other fields. A natural next step is to propose a platform on which operators could design and execute such loops.

To achieve this, a set of requirements and assumptions for such a system was defined. Kubernetes was chosen as the runtime environment due to its widespread adoption in the community and its inherent use of closed control loops. An extensive analysis was conducted on how Kubernetes extension mechanisms, such as "Custom Resources" and the "Operator" pattern, could be used to create a framework enabling the modeling of closed control loops.

This thesis describes the developed platform, its architecture, the semantics of syntax in the defined objects, operational principles, integrations with external systems, and a user guide. It also discusses the platform's implementation, the technologies behind it, and the decisions made during its development. Finally, the thesis presents a practical test of the platform's functionality using the Open5GS 5G system emulator in combination with UERANSIM. The work concludes with a list of findings and potential extensions or improvements to the platform.

Keywords: Closed Control Loops, Kubernetes, Managing Telco-Networks, Automation, Go, Open5GS, Microservices

Spis treści

1. Wstęp	7
1.1. Przedmowa	7
1.2. Cel i zakres pracy	7
1.3. Struktura dokumentu	8
2. Stan wiedzy	8
2.1. Wstęp	8
2.2. Historia	8
2.3. ENI	10
2.4. Przegląd obiecujących pętli sterowania	12
2.4.1. OODA	12
2.4.2. MAPE-K	12
2.4.3. FOCALÉ	12
2.4.4. COMPA	14
Bibliografia	15
Wykaz symboli i skrótów	16
Spis rysunków	16
Spis tabel	16
Spis załączników	16

1. Wstęp

1.1. Przedmowa

Wraz z rozwojem telekomunikacji stopień jej skomplikowania jak i mnogość podłączonych urządzeń stale rośnie. Sieci 5G zwiastują obsługę miliardów urządzeń, co sprawia, że tradycyjne podejście do zarządzania sieciami staje się niewystarczające. W pewnym momencie manualne operowanie sieciami (ang. *human-driven networks*) stanie się wręcz niemożliwe. Dlatego obserwujemy obecnie zwrot w stronę wirtualizacji oraz automatyzacji sieci. Jednocześnie dynamiczny rozwój sztucznej inteligencji otwiera nowe możliwości. Te dwa czynniki stanowią wspólnie świetny fundament do tego, aby branża sieci telekomunikacyjnych postawiła sobie za cel budowę "inteligentnych" sieci - takich, które są w pełni autonomiczne, samowystarczalne oraz niewymagają nadzoru ludzkiego.

W tym celu ETSI (European Telecommunications Standards Institute) powołało komitet o nazwie ENI - Experiential Networked Intelligence, który ma na celu wypracowanie specyfikacji dla Kognitywnych Systemów Zarządzania Siecią (Cognitive Network Management system). Kognitywny system oznacza taki, który jest w stanie uczyć się i podejmować decyzje bazujące na zebranej wiedzy w sposób przypominający ludzki umysł. ENI opiera swoją architekturę na zamkniętych pętlach sterowania.

Pętlą sterowania, ENI nazywa mechanizm, który monitoruje wydajność systemu lub procesu poddawanego kontroli w celu osiągnięcia pożądanego zachowania. Innymi słowy, pętla sterowania reguluje działanie zarządzanego obiektu. Pętle sterowania można podzielić na zamknięte lub otwarte w zależności od tego czy działanie sterujące zależy od sprzężenia zwrotnego z kontrolowanym systemem lub obiektem. Jeśli tak, pętle nazywamy zamkniętą, jeśli nie - otwartą.

W przypadku architektury ENI, zamknięta pętla sterowania służy jako model organizacji pracy (ang. *workflow*) elementów odpowiedzialnych za sztuczną inteligencję. W dokumencie [1], ENI dokonało przeglądu obiecujących architektur zamkniętych pętli sterowania znanych ludzkości. Naturalnym następnym krokiem jest zaproponowanie platformy, na której można takowe pętle zamodelować oraz uruchomić.

1.2. Cel i zakres pracy

Celem pracy jest kontynuacja badań ENI, polegająca na zaproponowaniu architektury platformy, na której możliwe będzie modelowanie oraz uruchamianie zamkniętych pętli sterowania. Niniejsza praca nie skupia się na aspektach związanych ze sztuczną inteligencją. Platforma ma jedynie służyć do zamodelowania oraz uruchomienia *workflow* pętli, ale sama nie stanowi środowiska wykonawczego dla jej komponentów.

W zakres pracy wchodzi: sformułowanie wymagań na platformę, opracowanie proponowanej architektury wsparte licznymi badaniami podczas jej powstawania, implementacja PoC (ang. *Proof of Concept*) według proponowanej architektury, przeprowadzenie testów platformy oraz analiza jej potencjału w kontekście dalszego rozwoju. Po publikacji praca

może stanowić podstawy do konkretnych implementacji Kognitywnych Systemów Zarządzania Siecią, specyfikowanych przez ENI.

1.3. Struktura dokumentu

Dokument został podzielony na 6 sekcji. Druga sekcja przedstawia bardziej szczegółowo niż we wstępie badania podjęte przez ENI. Stanowi ona nieco wprowadzenie teoretyczne oraz pojęciowe, aby ułatwić przekaz w dalszej części pracy. Sekcja trzecia zawiera opis proponowanej architektury. Sekcja czwarta opisuje implementację PoC platformy oraz napotkane wyzwania podczas formułowania architektury, które celowo zostały zebrane w jedno miejsce i umieszczone oddzielnie w celu łatwiejszej lektury. Sekcja numer pięć przedstawia użycie platformy w praktyce przy okazji stanowiąc jej test. Na koniec, w sekcji szóstej przeprowadzono analizę zaproponowanej architektury, przedstawiono jej możliwości oraz ograniczenia oraz wskazano kierunki potencjalnego rozwoju.

2. Stan wiedzy

2.1. Wstęp

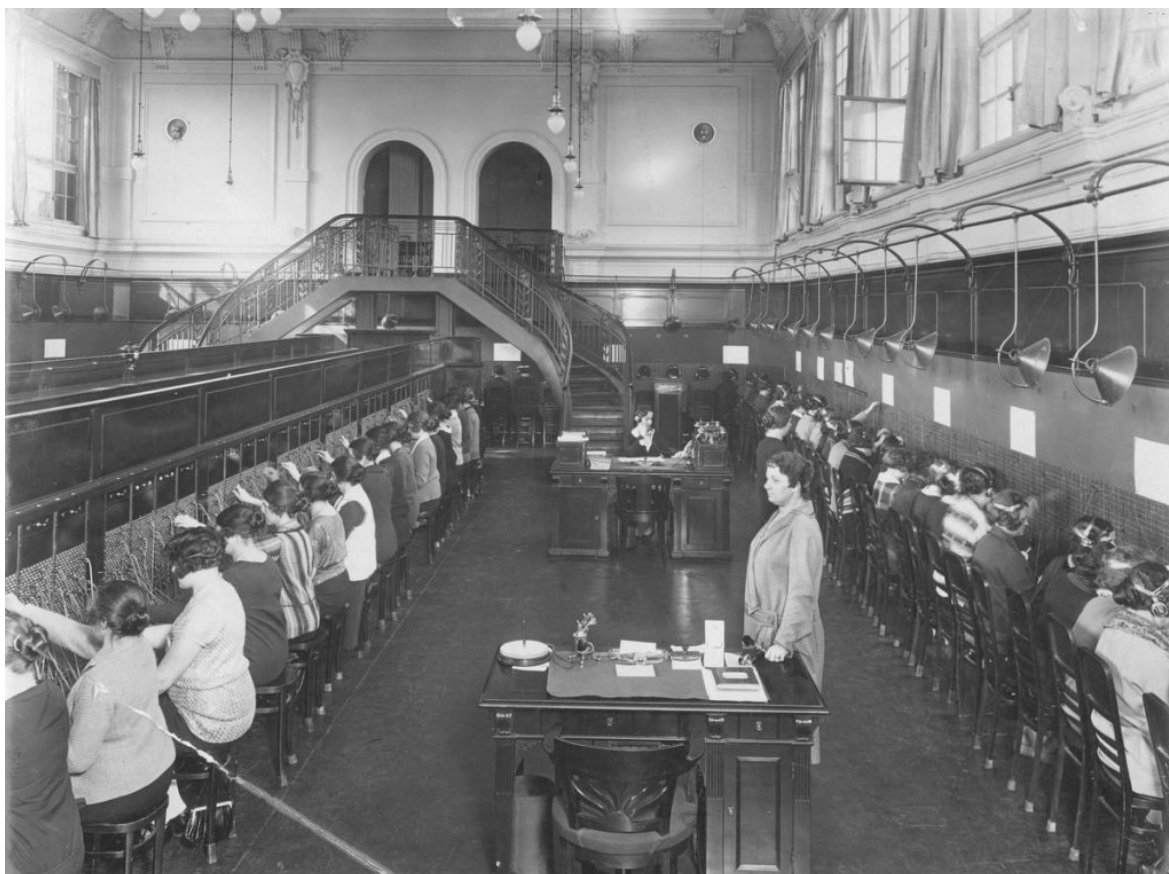
Elo elo 320.

2.2. Historia

Na początku XX wieku, gdy sieci telekomunikacyjne dopiero się rozwijały, wszystkie połączenia zestawiane były ręcznie. W momencie, gdy abonent podnosił słuchawkę telefonu, jego aparat wysyłał sygnał do lokalnej centrali telefonicznej. Na tablicy świetlnej zapalała się lampka informująca telefonistkę o próbie połączenia. Telefonistka odbierała, pytając, z kim abonent chce się połączyć. Następnie wprowadzała odpowiednią wtyczkę do odpowiedniego gniazda na tablicy rozdzielczej, zestawiając fizyczne połączenie między dwoma liniami. Jeśli rozmowa miała się odbyć na większą odległość (np. między miastami) połączenie przekazywane było przez kolejne centrale. Każda centrala po drodze wymagała ręcznej obsługi przez pracujące w nich telefonistki.

Dziś taki scenariusz wydaje się wręcz absurdalny, a oferty pracy dla telefonistek dawno już zniknęły z tablic ogłoszeń. Praca wykonywana przez telefonistki (czyli komutacja łączy) nadal jest potrzebna do prawidłowego funkcjonowania sieci telekomunikacyjnych, lecz wykonywana jest przez programy komputerowe w sposób w pełni zautomatyzowany. Ręczna komutacja była pierwszym krokiem w kierunku rozwoju globalnych sieci telekomunikacyjnych i choć z dzisiejszej perspektywy wydaje się być bardzo pracochłonna i ograniczająca, bez niej nie powstałyby fundamenty, na których oparto późniejsze systemy automatyczne. Jest to przykład tego, jak technologia stopniowo uwalniała człowieka od bezpośredniej obsługi różnych systemów (dając mu przestrzeń na rozwój w innych obszarach).

Na zasadzie indukcji możemy przyjąć, iż dziś znajdujemy się w podobnym położeniu - sieci telekomunikacyjne wciąż wymagają bezpośredniego zarządzania przez człowieka. Po prostu granica tego styku - system-człowiek - jest mocno przesunięta. Dziś człowiek



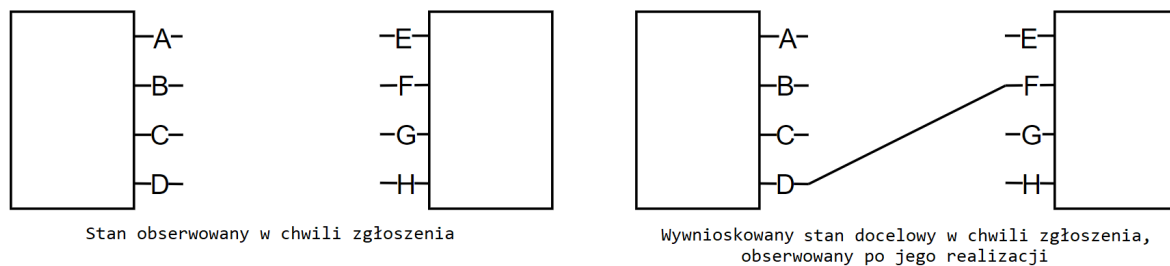
Rysunek 2.1. Pracowniczki warszawskiej centrali telefonicznej na zdjęciu z końca lat 20. (domena publiczna).

spotyka się z systemami dużo bardziej złożonymi, a zarazem dającymi dużo więcej możliwości. Możliwe, że nie istnieje ostateczny punkt styku i systemy telekomunikacyjne zawsze będą wymagały nadzoru ludzkiego. Niezależnie od tej kwestii przesunięcie tej granicy w czasach telefonistek, a dziś wymaga automatyzacji innego rodzaju.

O telefonistkach możemy powiedzieć, że zarządzały one pracą systemu telekomunikacyjnego¹. Regulowały one jego działanie. W momencie zgłoszenia zastawały one system w pewnym stanie i w ramach realizacji zgłoszenia musiały doprowadzić system do nowego (docelowego) stanu. Cała ich praca tak naprawdę polegała na "wnioskowaniu" (ang. *reason*) jak ma wyglądać stan docelowy, a następnie na wykonaniu *akcji*, która doprowadzała system do tego stanu.

Pomówmy trochę o *wnioskowaniu*. Jak ono przebiega? Po pierwsze telefonistka otrzymuje *dane*. Dane są to fakty lub statystyki zebrane razem w celu analizy. Telefonistka zbiera dane w formie faktów jakie miały miejsce. Zadzwoił aparat telefoniczny, telefonistka podniosła słuchawkę i usłyszała w niej, że abonent chce się połączyć ze swoim szwagrem mieszkającym w kamienicy pod numerem 35 na ulicy polnej. Dostępne dane, telefonistka przekuła na *informację*. Informacja to dane pozbawione formy przekazu. Równie dobrze, rozmowa z abonentem mogła przebiec zupełnie inaczej lub telefonistka mogła by otrzymać listowną prośbę o realizację połączenia. Zupełnie inne dane dostarczyły by tej samej

¹ t.j. siecią PSTN



Rysunek 2.2. Dwa stany systemu: sprzed, oraz po realizacji zgłoszenia o treści: "D" chce zadzwonić do "F"

informacji, czyli to, że abonent spod konkretnego łącza, chce się dodzwonić do abonenta spod innego, konkretnego łącza. Gdy telefonistka otrzyma informacje na temat zgłoszenia, łączy ją z dotychczasowo posiadanymi informacjami, jak rozmieszczenie łączy na tablicy komutacyjnej. Następnie używa swojej *wiedzy*, czyli zestawu wzorców, które pozwalają jej wytłumaczyć oraz przewidzieć co się wydarzyło, dzieje w tej chwili lub może wydarzyć w przyszłości. Wiedza opiera się na informacji oraz umiejętnościach pozyskanych dzięki doświadczeniu lub edukacji. Obserwuje ona aktualny stan systemu i *rozumując*, na podstawie wiedzy wyobraża sobie stan docelowy (pożądany) oraz definiuje *akcje*, jakie należy przeprowadzić na systemie, aby osiągnął on ten stan.

W przypadku komutacji proces wnioskowania jest względnie prosty, dający się do zapisania w języku programowania. Dlatego też wraz z rozwojem technologicznym dokonano przejścia na automatyczne centrale telefoniczne. Jednakże opis wyżej nadal pozostaje ważny. Do centrali trafiają dane w postaci sygnału na jednym z łącz. Sygnał przedstawia numer MSISDN abonenta końcowego. Centrala przekuwa te dane na informacje, a następnie program w jej pamięci dokonuje wnioskowania. Dzięki programowalnym przełącznikom komutacyjnym również wykonanie *akcji* (czyli połączenie ze sobą dwóch łącz) jest możliwe bez udziału człowieka. W dzisiejszych czasach proces wnioskowania wykonywany przez ludzi nadzorujących system telekomunikacyjny już nie jest tak prosty. Dzięki wirtualizacji i programowalności funkcji sieciowych każdy rodzaj rozumowania jaki można skondensować do programu komputerowego został już poddany temu procesowi. Teraz akcje wykonywane przez człowieka rzeczywiście wymagają jego inteligencji². Dlatego więc następnym krokiem w rozwoju telekomunikacji jest stworzenie inteligentnych sieci.

2.3. ENI

ENI rozwija specyfikacje w kierunku Kognitywnego Systemu Zarządzania, który będzie regulował działanie sieci poprzez wykorzystanie technik sztucznej inteligencji, takich jak uczenie maszynowe (ang. *machine learning*) oraz wnioskowanie (ang. *reasoning*).

Z poprzedniej sekcji wiemy, że aby *wnioskować* potrzebna jest wiedza. Skąd taki system pozyskiwałby wiedzę? Otóż, dzieje się to poprzez proces *machine learning*'u. Pierwsza faza czyli tzw. *trening*, odbywa się jeszcze przed wdrożeniem systemu. Ale system również może uczyć się "z doświadczenia" (ang. *experience*) po wdrożeniu, kiedy już jest w pełni

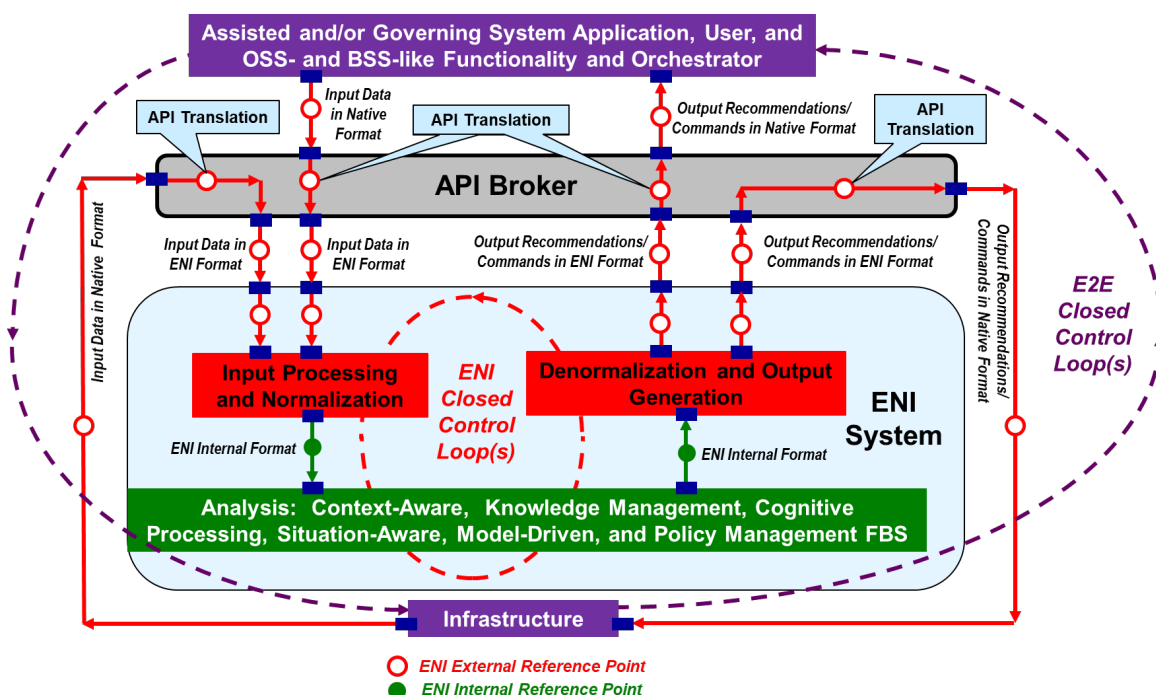
² lub sprawczości w świecie fizycznym, której komputerom brakuje

operacyjny. Stąd mówimy o empirycznej inteligencji sieciowej (ang. *experiential networked intelligence*).

Czyli z jednej strony wiedza potrzebna do wnioskowania, a w ostateczności podejmowania decyzji płynęłaby z samej sieci, co w przypadku telefonistki równoważne jest zapalającej się lampce na tablicy świetlnej. Z drugiej strony telefonistka ma również niejako wbudowaną w siebie wiedzę o tym, jak zorganizowane są łącza na przełącznicy komutacyjnej. Podobnie sieć musi mieć pewną części wiedzy nieco narzuconą z góry. Ta część wiedzy stanowi jako zestaw reguł używanych do zarządzania i kontrolowania stanu sieci. Takie reguły nazywamy *politykami*.

Polityki mogą płynąć od: aplikacji zarządzających siecią, użytkowników sieci, systemów OSS/BSS lub Orkiestratora. Ważne jest to, aby polityki były *świadome kontekstu*. Pozwoli to na stworzenie systemu kognitywnego, czyli takiego, który uczy się, wnioskuje oraz podejmuje decyzje w sposób przypominający ludzki umysł. Taki system z kolei jest w stanie w dużym stopniu odciążyć operatora i zautomatyzować zadania jak:

- dynamiczne przydzielanie zasobów (ang. *dynamic resource allocation*),
- równoważenie obciążenia (ang. *load balancing*),
- zarządzanie wydajnością energetyczną (ang. *energy efficiency management*),
- samo naprawiające się sieci (ang. *self-healing networks*),
- optymalizacja jakości doświadczeń użytkowników (ang. *QoE optimization*),
- egzekwowanie polityk (ang. *policy enforcement*),
- zapewnienie zgodności z regulacjami (ang. *regulatory compliance*)
- i wiele innych.



Rysunek 2.3. Wysokopoziomowa architektura funkcjonalna ENI

Wysokopoziomą architekturę funkcjonalną ENI pokazano na rycinie 2.3. Jej działanie

można opisać jako dwie *zamknięte pętle sterowania*. Za pomocą zewnętrznej, system zarządzania (użytkownik, OSS/BSS, orkiestrator) reguluje pracę sieci (infrastruktury). Dokonuje tego nie bezpośrednio, lecz za pomocą ENI System, który sam wewnętrznie również posiada zamkniętą pętlę sterowania. Wewnętrzna pętla podczas wnioskowania bierze pod uwagę nie tylko wiedzę płynącą bezpośrednio z sieci, ale również tę od orkiestratora. Wszystkie komponenty odpowiedzialne za sztuczną inteligencję są umieszczone na rycinie w zielonym prostokącie i stanowią *workflow* wewnętrznej pętli.

2.4. Przegląd obiecujących pętli sterowania

Przedstawione w poprzedniej sekcji architektura jest jedynie funkcjonalna i zawiera w sobie dużo abstrakcji. Ostateczna postać wewnętrznej zamkniętej pętli sterowania w ENI system jest jeszcze w trakcie specyfikacji. ENI w dokumencie [1] dokonało przeglądu istniejących, możliwych do zaaplikowania architektur zamkniętych pętli sterowania. Niniejsza praca dyplomowa chce zaproponować architekturę do modelowania oraz uruchamiania pętli w ENI System, dlatego ważne jest, aby poznać jej składowe elementy. Nie mamy na celu rozważać dokładnej funkcjonalności tych w środku tych elementów, jedynie poznać ich interfejsy, abyśmy wiedzieli co jako platforma uruchomieniowa powinniśmy być w stanie im zapewnić.

// Opis wyżej powinien być we wstępie tego rozdziału.

// A tu napisz, że każda pętla ma jakieś elementy i fajnie, bo my potem musimy tym elementom dać żyć.

2.4.1. OODA

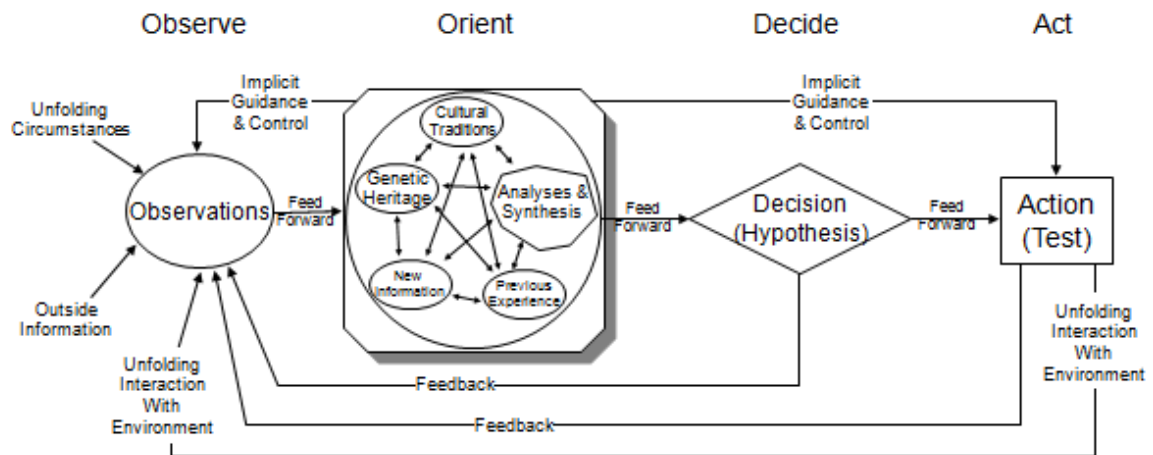
Pętla autorstwa Johna R. Boyda składa się z 4 części: "Observe", "Orient", "Decide", "Act". Stanowi podsumowanie jego myśli na temat strategii, taktyki oraz procesów decyzyjnych. John R. Boyd był pułkownikiem Sił Powietrznych USA i jednym z najbardziej wpływowych myślicieli wojskowych XX wieku. Mimo iż był wojskowym jego koncepcja składająca się z nieustannej obserwacji, orientacji, podejmowania decyzji i działania jest kluczowa w każdej formie walki. Może być również stosowana w polityce, biznesie, życiu codziennym. Główną koncepcją jest ciągle dostosowywanie się do zmieniających się warunków oraz podważanie istniejących założeń. Koncepcja ta świetnie nadaje się do adaptacyjnych oraz kognitywnych systemów zarządzania.

2.4.2. MAPE-K

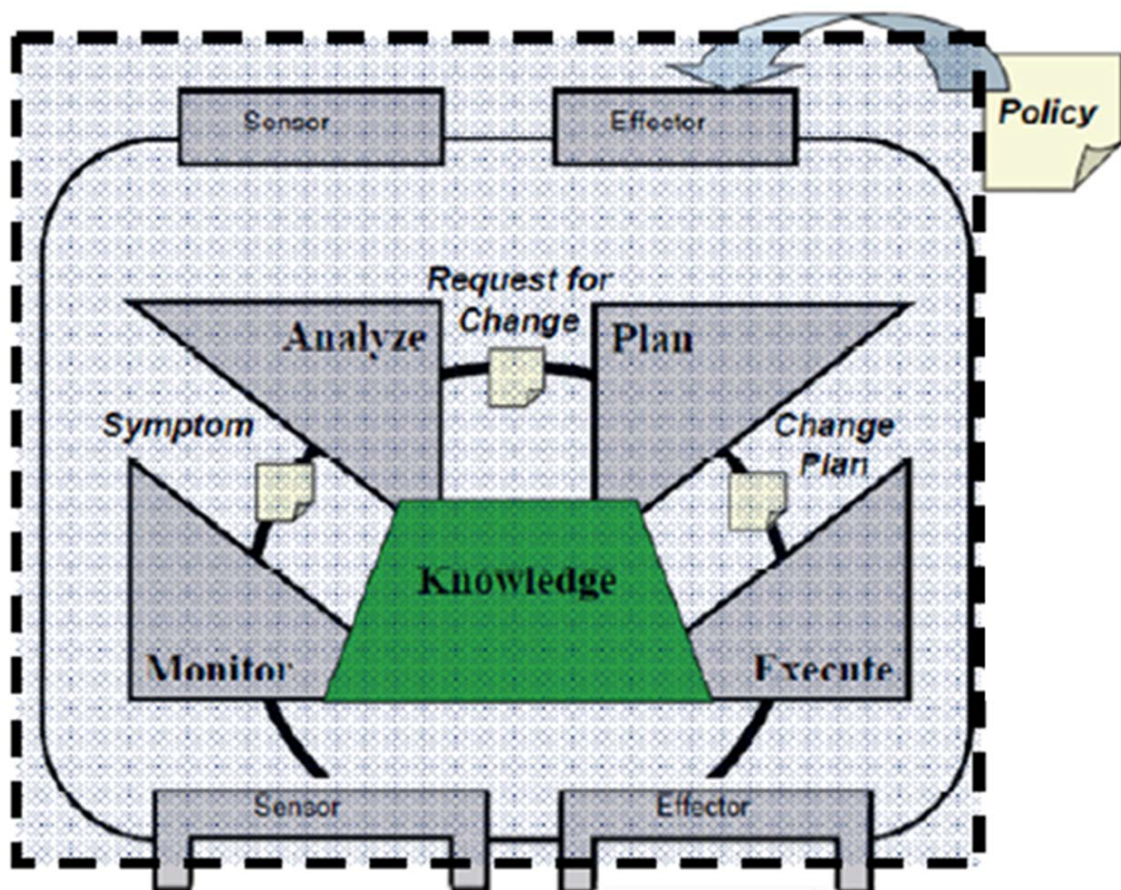
MAPE-K jest to pętla opracowana przez IBM w dziedzinie autonomicznego przetwarzania i systemów samoadaptujących się. Główne jej etapy to Monitorowanie, Analiza, Planowanie, Wykonanie, a każdy z nich ma dostęp do wspólnej bazy wiedzy.

2.4.3. FOCAL

Nazwa tej pętli pochodzi słów: Podstawa (ang. *Foundation*), Obserwacja (ang. *Observation*), Porównanie (ang. *Comparison*), Analiza (ang. *Analysis*), Uczucie (ang. *Learning*), Wnioskowanie (ang. *Reason*). Niespotykanym dotąd elementem może być Podstawa, która

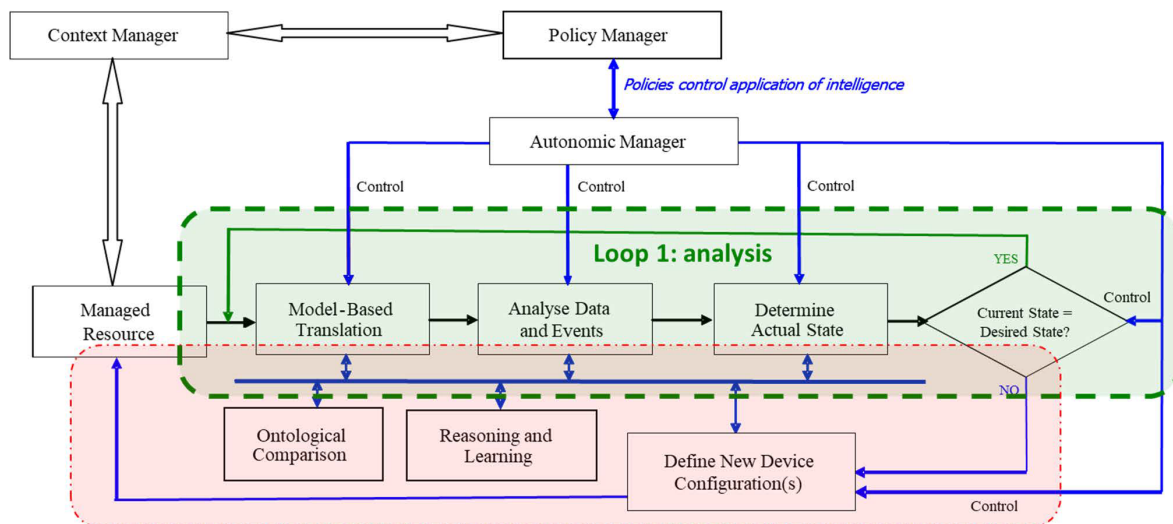


Rysunek 2.4. Architektura pętli OODA



Rysunek 2.5. Architektura pętli MAPE-K

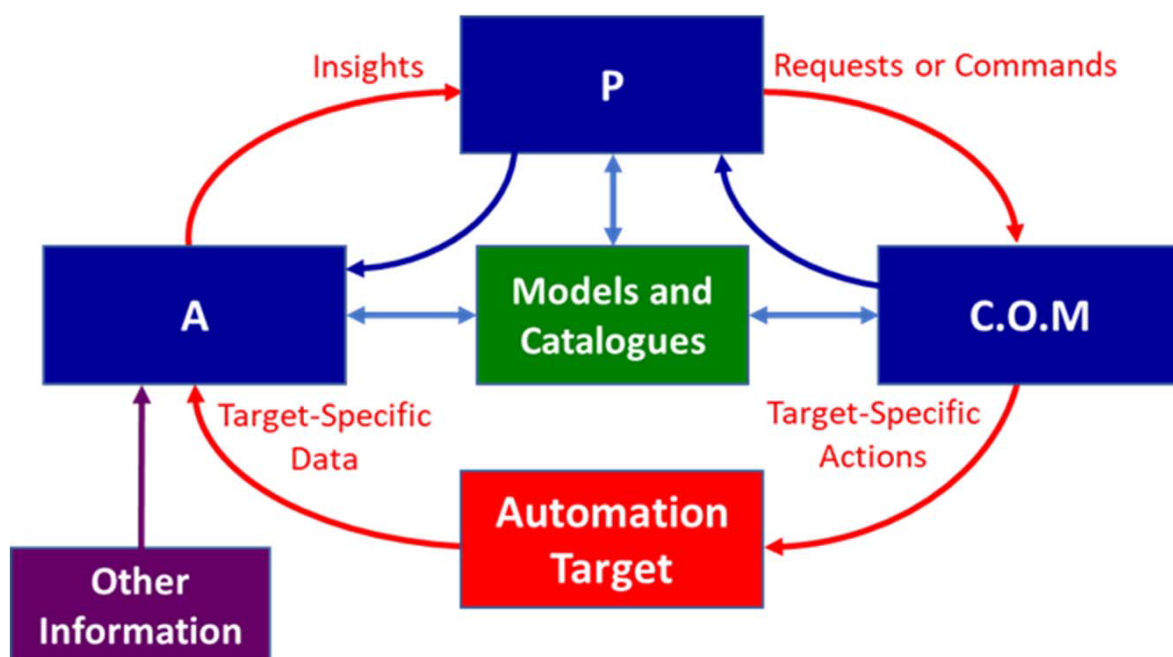
jest statyczną częścią modelu i określa strukturę oraz podstawowe zasady systemu. Ta architektura jako pierwsza proponuje też element odpowiedzialny za uczenie maszynowe (ang. *machine learning*).



Rysunek 2.6. Architektura pętli FOCAL

2.4.4. COMPA

Elementy tej pętli to: Analityka (ang. *Analytics*), Polityki (ang. *Policies*) oraz Sterowanie, Orkiestracja i Zarządzanie (ang. Control, Orchestration, Management)



Rysunek 2.7. Architektura pętli COMPA

Bibliografia

- [1] ETSI GR ENI 017 - *"Experiental Networked Intelligence (ENI); Overview of Prominent Closed Control Loops Architectures"*, Dostęp zdalny (24.01.2025): https://www.etsi.org/deliver/etsi_gr/ENI/001_099/017/02.02.01_60/gr_ENI017v020201p.pdf, 2024.

Wykaz symboli i skrótów

ETSI – European Telecommunications Standards Institute

ENI – Experiential Networked Intelligence

Spis rysunków

2.1 Pracowniczkę warszawskiej centrali telefonicznej na zdjęciu z końca lat 20. (domena publiczna).	9
2.2 Dwa stany systemu: sprzed, oraz po realizacji zgłoszenia o treści: "D" chce zadzwoić do "F"	10
2.3 Wysokopoziomowa architektura funkcjonalna ENI	11
2.4 Architektura pętli OODA	13
2.5 Architektura pętli MAPE-K	13
2.6 Architektura pętli FOCAL	14
2.7 Architektura pętli COMPA	14

Spis tabel

Spis załączników

1. Definicje	16
------------------------	----

Załącznik 1. Definicje

Architektura – Zbiór reguł i metod opisujących funkcjonalność, organizację oraz implementację systemu.

Workflow – Sekwencja połączonych węzłów, czasami zależnych warunkowo, która realizuje określony cel. Zazwyczaj workflow definiuje się w celu organizacji pracy

Wnioskowanie (ang. *reasoning*) – Proces, w którym system wyciąga logiczne wnioski z dostępnych danych i wiedzy

Kognitywność (ang. *cognition*) – Proces rozumienia danych oraz informacji w celu produkcji nowych danych, informacji oraz wiedzy

System kognitywny – System, który uczy się, wnioskuje oraz podejmuje decyzje w sposób przypominający ludzki umysł