KUBETEUS: An Intelligent Network Policy Generation Framework for Containers

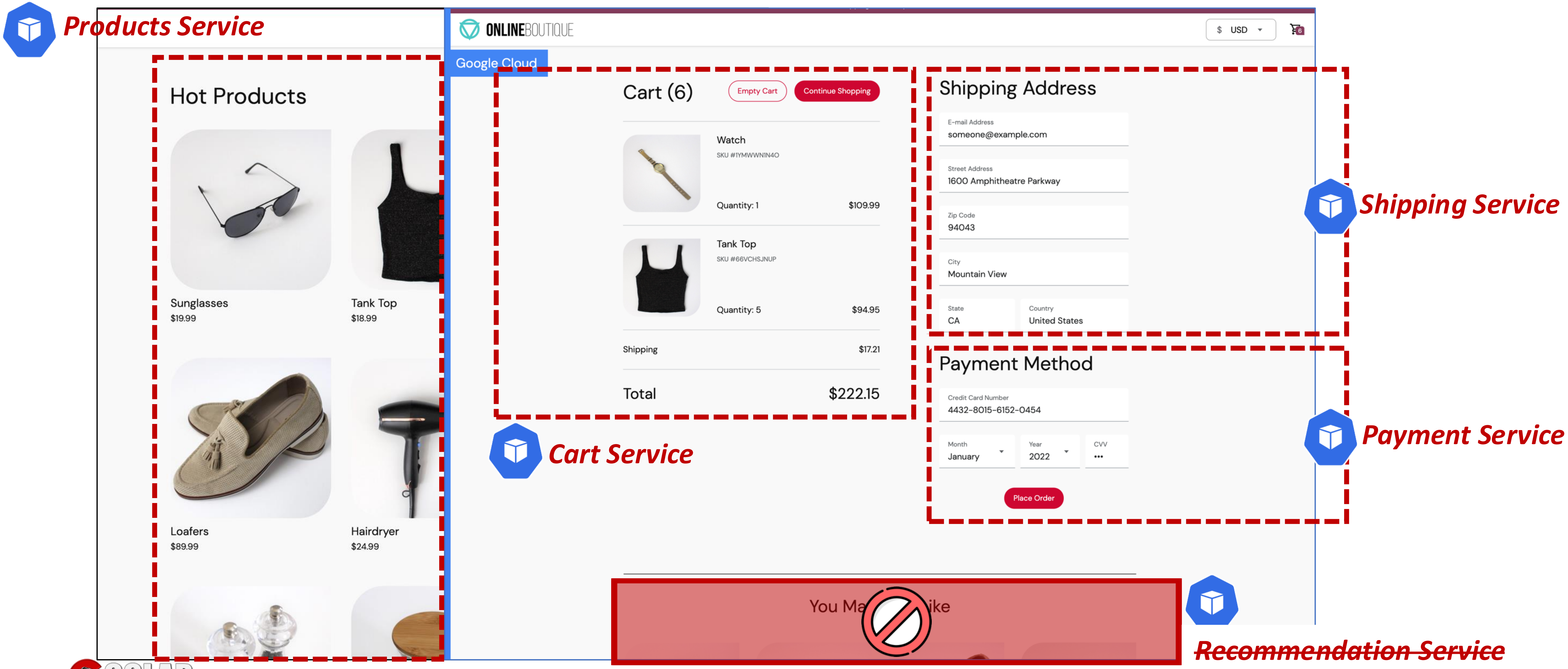
Bom Kim, Hyeonjun Park, Seungsoo Lee

Department of Computer Science & Engineering

Incheon National University, Korea

May 21st, 2025

Microservice Architecture (MSA)



Products Service

Hot Products

Sunglasses $19.99	Tank Top $18.99
Loafers $89.99	Hairdryer $24.99

Cart (6)

Watch SKU #1YMWWN1N4O	Quantity: 1	$109.99
Tank Top SKU #66VCHSJNUP	Quantity: 5	$94.95
Shipping		$17.21
Total		$222.15

Cart Service

Shipping Address

E-mail Address: someone@example.com

Street Address: 1600 Amphitheatre Parkway

Zip Code: 94043

City: Mountain View

State: CA

Country: United States

Shipping Service

Payment Method

Credit Card Number: 4432-8015-6152-0454

Month: January

Year: 2022

CVV: •••

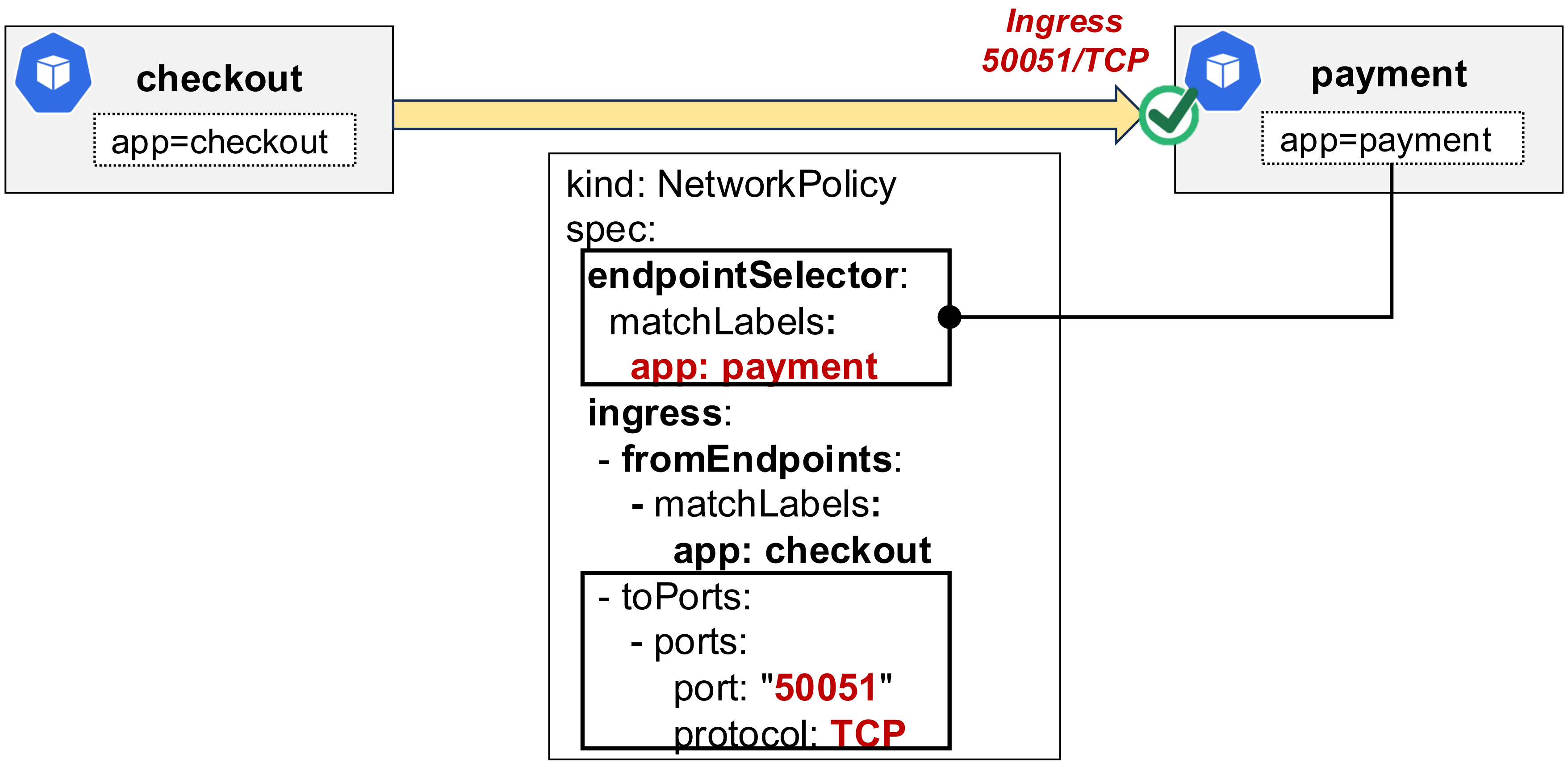
Payment Service

~~Recommendation Service~~

https://github.com/GoogleCloudPlatform/microservices-demo

Network Policy in Cloud-native Environments

- Enables **fine-grained access control** between containers using rule-based policies



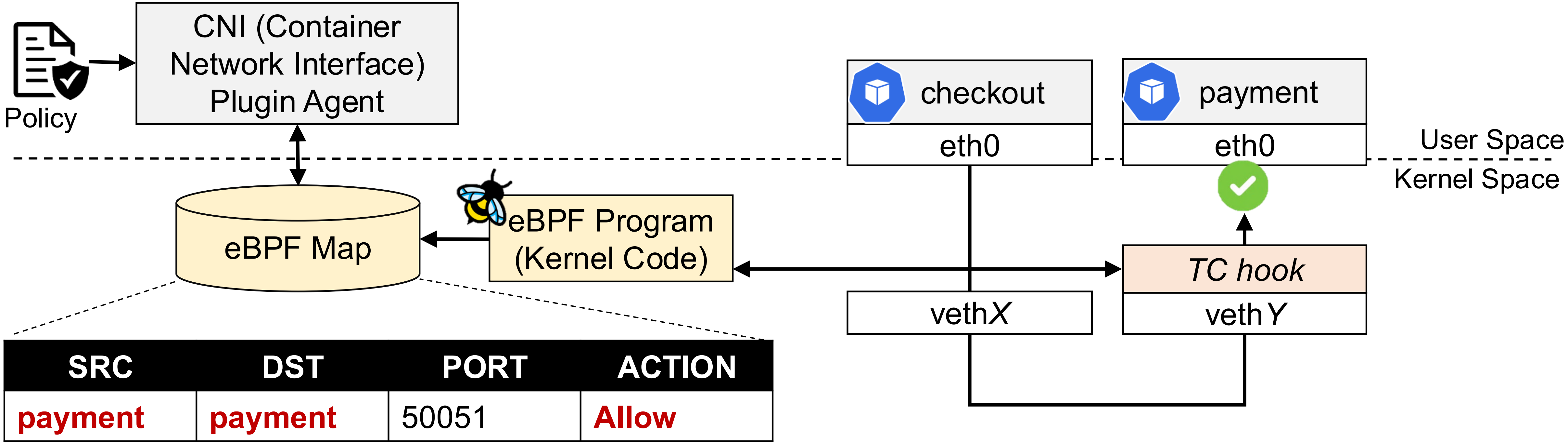
```
kind: NetworkPolicy
spec:
  endpointSelector:
    matchLabels:
      app: payment
  ingress:
   - fromEndpoints:
      - matchLabels:
          app: checkout
   - toPorts:
      - ports:
          port: "50051"
          protocol: TCP
```

Ingress Network Policy

Policy Enforcement via eBPF

*eBPF: extended Berkeley Packet Filter

- Enables real-time policy enforcement at critical network paths with **minimal performance overhead**



SRC	DST	PORT	ACTION
payment	**payment**	50051	**Allow**

Container Network Interface (CNI)

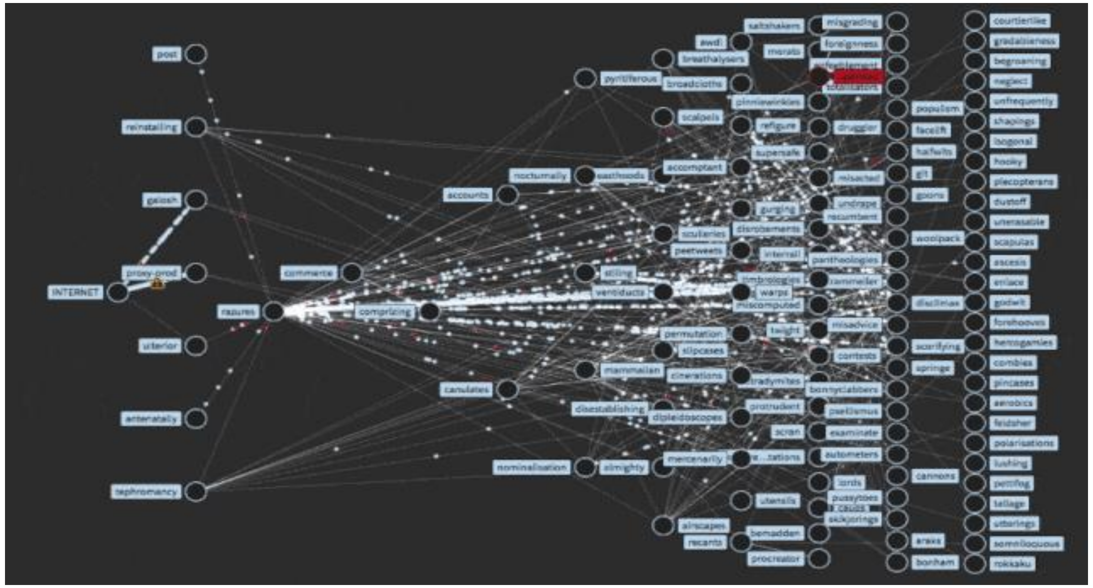
- Configures **container network settings** and enables communication between containers

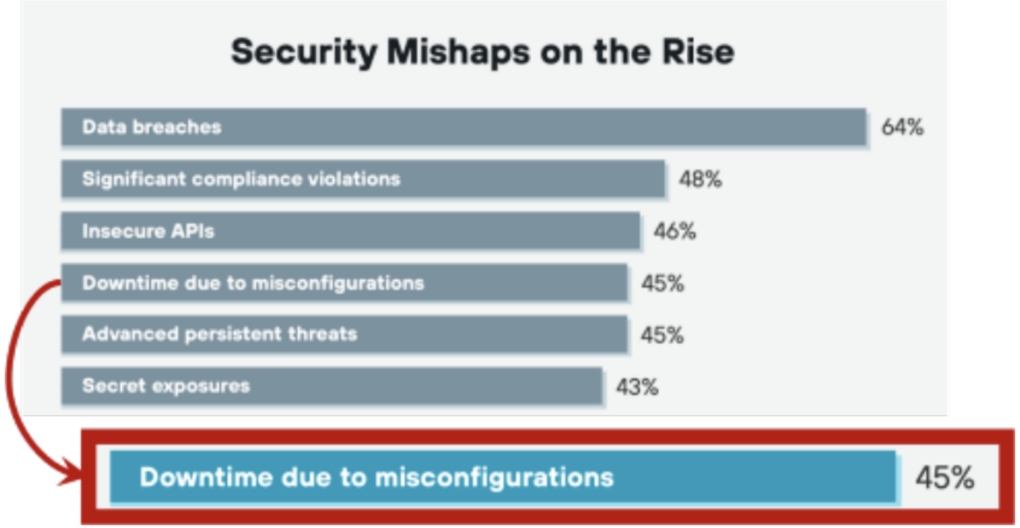
CNI	Network Policy Support	Scope	L3/L4	L7	Action	Priority
Flannel	-	-	-	-	-	-
Weave Net	✓ (Kubernetes Policy)	Namespace	Ingress, Egress	No	Allow, Deny	No
Kube-router			Ingress			
Calico	✓ (Kubernetes / Calico Policy)	Namespace, Cluster-wide	Ingress, Egress	HTTP	Allow, Deny, Log, Pass	✓
Cilium	✓ (Kubernetes / Cilium Policy)	Namespace, Cluster-wide	Ingress, Egress	HTTP, gRPC, Kafka	Allow, Deny	No
Antrea	✓ (Kubernetes / Antrea Policy)	Namespace, Cluster-wide	Ingress, Egress	HTTP	Allow, Drop, Reject, Pass	✓

Problem Statements (1/3)

- Numerous containers + manual management
  → **time-consuming** and error-prone (**misconfiguration**)

+500 Microservices

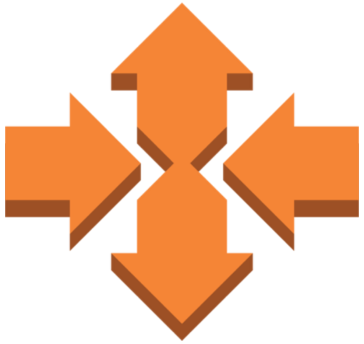




<2024 Palo Alto Networks Cloud-Native Security Report>

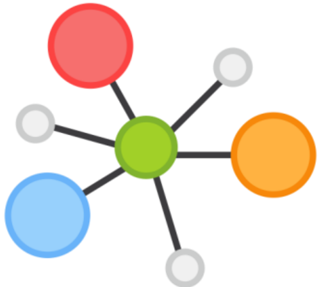
Problem Statements (2/3)

- Insufficient understanding of the **dynamic nature** of container-based microservices

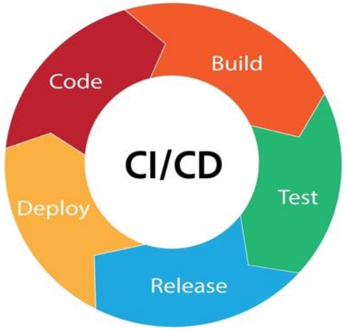


Frequent Scaling

Short-lived Containers



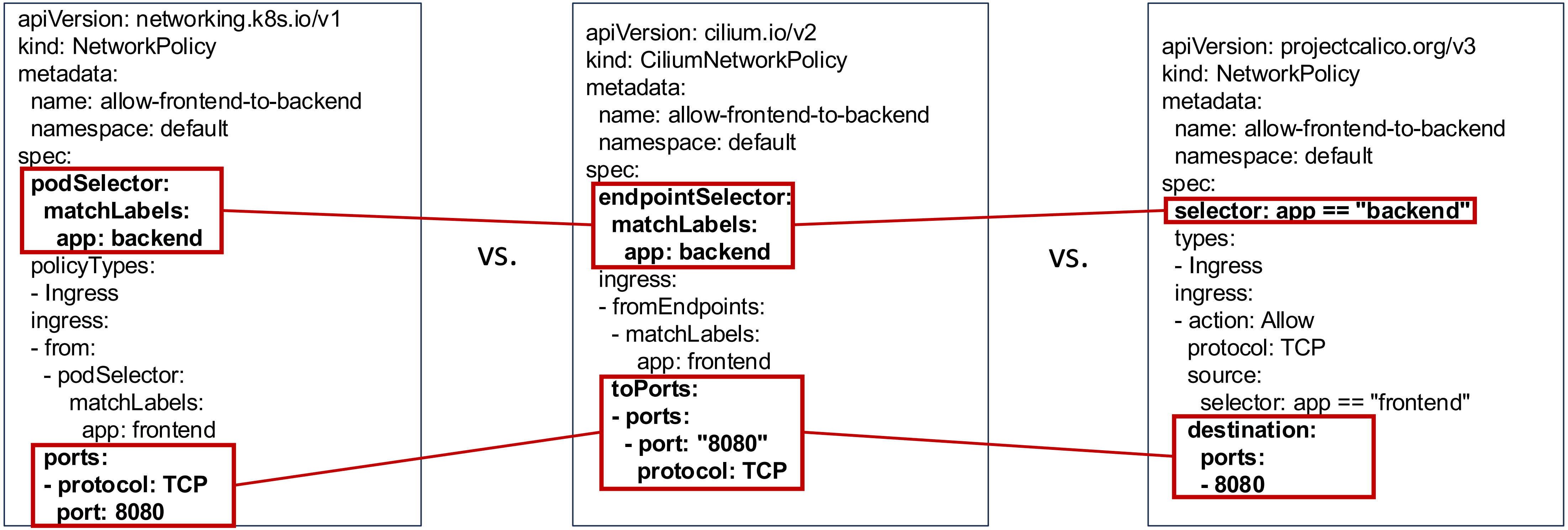
Changing Topology



DevSecOps

Problem Statements (3/3)

- Diverse and incompatible Container Network Interfaces (CNIs)



kubernetes

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-frontend-to-backend
  namespace: default
spec:
  podSelector:
    matchLabels:
      app: backend
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: frontend
    ports:
    - protocol: TCP
      port: 8080
```

cilium

```
apiVersion: cilium.io/v2
kind: CiliumNetworkPolicy
metadata:
  name: allow-frontend-to-backend
  namespace: default
spec:
  endpointSelector:
    matchLabels:
      app: backend
  ingress:
  - fromEndpoints:
    - matchLabels:
        app: frontend
    toPorts:
    - ports:
      - port: "8080"
        protocol: TCP
```

PROJECT CALICO

```
apiVersion: projectcalico.org/v3
kind: NetworkPolicy
metadata:
  name: allow-frontend-to-backend
  namespace: default
spec:
  selector: app == "backend"
  types:
  - Ingress
  ingress:
  - action: Allow
    protocol: TCP
    source:
      selector: app == "frontend"
    destination:
      ports:
      - 8080
```

Related Work

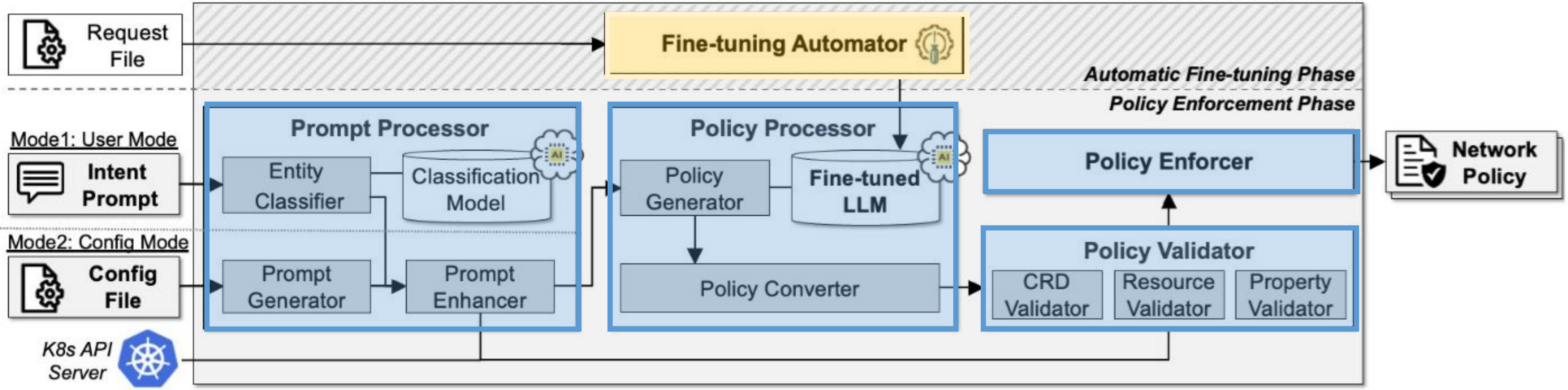
Research	Automation	Validation	Heterogeneous CNI Support	AI Model	Policy Scope
AUTOARMOR (Li et al., USENIX Security ’21)	✓ (static analysis, graph-based)	-	-	-	Microservice network policy
Log2Policy (Xu et al., ACSAC ’23)	✓ (log-based)	-	-	Word2vec, DBSCAN	Microservice access control
Hey, Lumi! (Jacobs et al., USENIX ATC ’21)	✓ (NLP-driven)	Partial (user feedback)	-	NER (Word2vec, Bi-LSTM/CRF)	Campus network management policy
KUBETEUS	✓ (LLM-based, Fully automated)	✓ (multi-step)	✓	Fine-tuned LLM, NER (BERT)	CNI-based Network policy

KUBETEUS Design Considerations

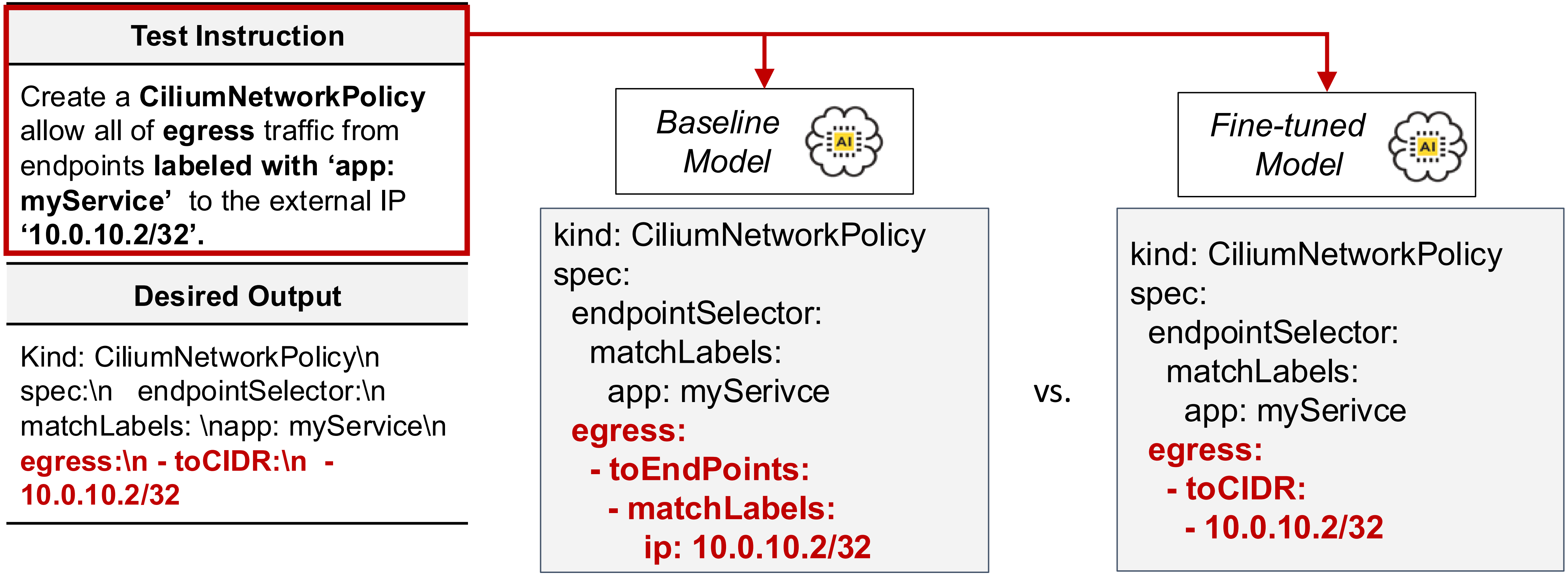
- 1. **Advanced Intelligent Policy Generation**
  - Understands **user intent** and **real-time cluster state** to generate tailored network policies
  - Analyzes **service relationships** via configuration files for advanced policy generation
- 2. **Multi-step Policy Validation**
  - Verifies **syntactic correctness** of generated policies
  - Ensures **existence** and **consistency** of referenced resources within the cluster
- 3. **Support for Diverse Container Network Interfaces**
  - Automatically detects **characteristics of various CNIs**
  - Generates network policies optimized for the specific cluster environment

KUBETEUS Overviews

- (i) Automatic Fine-tuning Phase
- (ii) Policy Enforcement Phase



Why Fine-Tune for LLMs?



Test Instruction

Create a **CiliumNetworkPolicy** allow all of **egress** traffic from endpoints **labeled with ‘app: myService’** to the external IP **‘10.0.10.2/32’.**

Desired Output

Kind: CiliumNetworkPolicy\n
spec:\n   endpointSelector:\n
matchLabels: \napp: myService\n
egress:\n - toCIDR:\n -
10.0.10.2/32

Baseline Model

```
kind: CiliumNetworkPolicy
spec:
  endpointSelector:
    matchLabels:
      app: mySerivce
  egress:
    - toEndPoints:
      - matchLabels:
          ip: 10.0.10.2/32
```

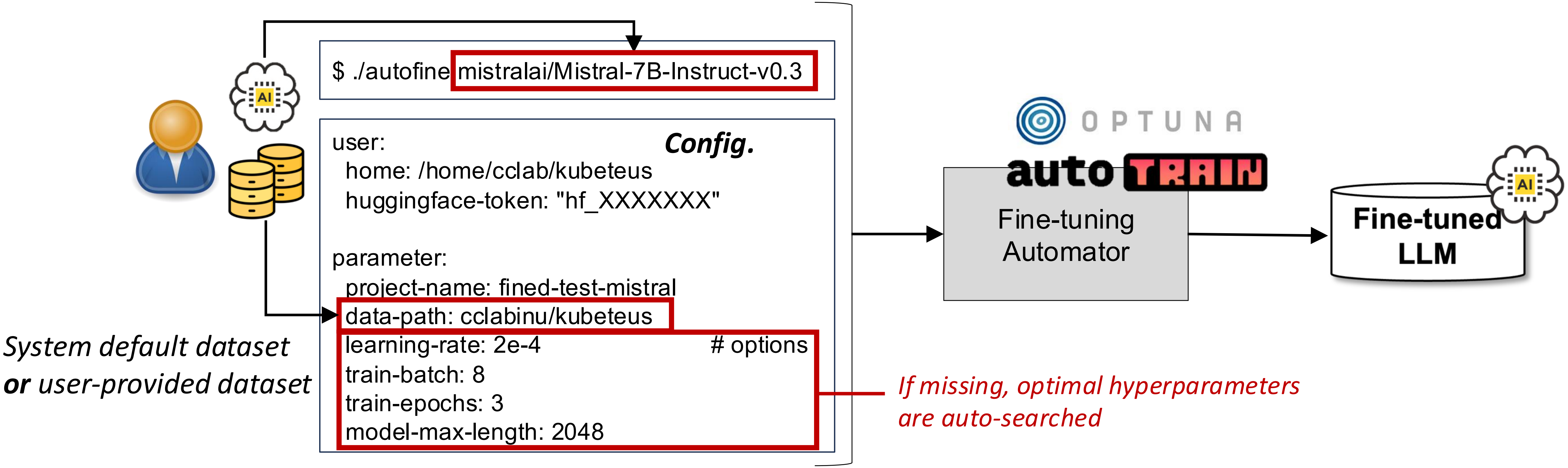
vs.

Fine-tuned Model

```
kind: CiliumNetworkPolicy
spec:
  endpointSelector:
    matchLabels:
      app: mySerivce
  egress:
    - toCIDR:
      - 10.0.10.2/32
```

KUBETEUS Details: Finetuning Automator

- Simplifies complex fine-tuning through **easy-to-use configuration inputs**
- Enables **seamless adaptation to new LLMs** without manual tuning or expert intervention



```
$ ./autofine mistralai/Mistral-7B-Instruct-v0.3
```

Config.

```
user:
  home: /home/cclab/kubeteus
  huggingface-token: "hf_XXXXXXX"

parameter:
  project-name: fined-test-mistral
  data-path: cclabinu/kubeteus
  learning-rate: 2e-4                  # options
  train-batch: 8
  train-epochs: 3
  model-max-length: 2048
```

System default dataset ***or*** *user-provided dataset*

If missing, optimal hyperparameters are auto-searched

KUBETEUS Details: Dataset and Model

Type	Format	#Origin	#Train	#Test	#Total	Size(MB)
Policy	json	857	132,899	33,225	166,124	187.3
Intent	csv	857	40,048	10,012	50,060	93.2

Table 1. Summary of Datasets of network policies and intent prompts.

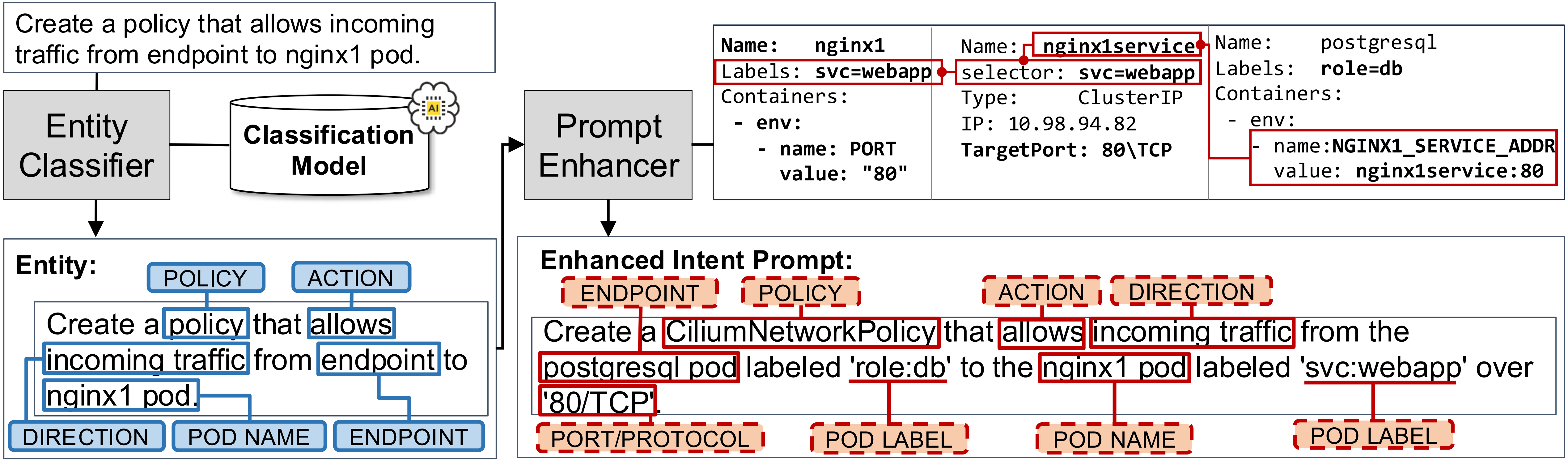
Model	#Size	#Params
Meta/Meta-Llama-3-8B-Instruct	16GB	8.03B
DeepSeek/deepseek-coder-7b-instruct-v1.5	14GB	6.91B
MistralAI/Mistral-7B-Instruct-v0.2	15GB	7.24B
Google/codegemma-7b-it	17GB	8.54B
Meta/codeLlama-7b-Instruct-hf	14GB	6.74B

Table 2. Summary of LLLMs fine-tuned for policy generation.

- Expanded the dataset from 857 to **166,124** samples through structured shuffling
- Selected an **open-source text-to-text model** for policy generation based on:
  - Compact size and low parameter count for efficient, accurate performance
  - Above-average policy generation quality across diverse requirements

KUBETEUS Details: Prompt Processor

- **(i) Entity Classifier**: BERT-based classifier that identifies intents and entities
- **(ii) Prompt Enhancer**: Analyzes resource relationships and generates structured prompts



Create a policy that allows incoming traffic from endpoint to nginx1 pod.

Create a policy that allows incoming traffic from endpoint to nginx1 pod.

```
Name:    nginx1
Labels:  svc=webapp
Containers:
 - env:
   - name: PORT
     value: "80"
```

```
Name:  nginx1service
selector:  svc=webapp
Type:      ClusterIP
IP: 10.98.94.82
TargetPort: 80\TCP
```

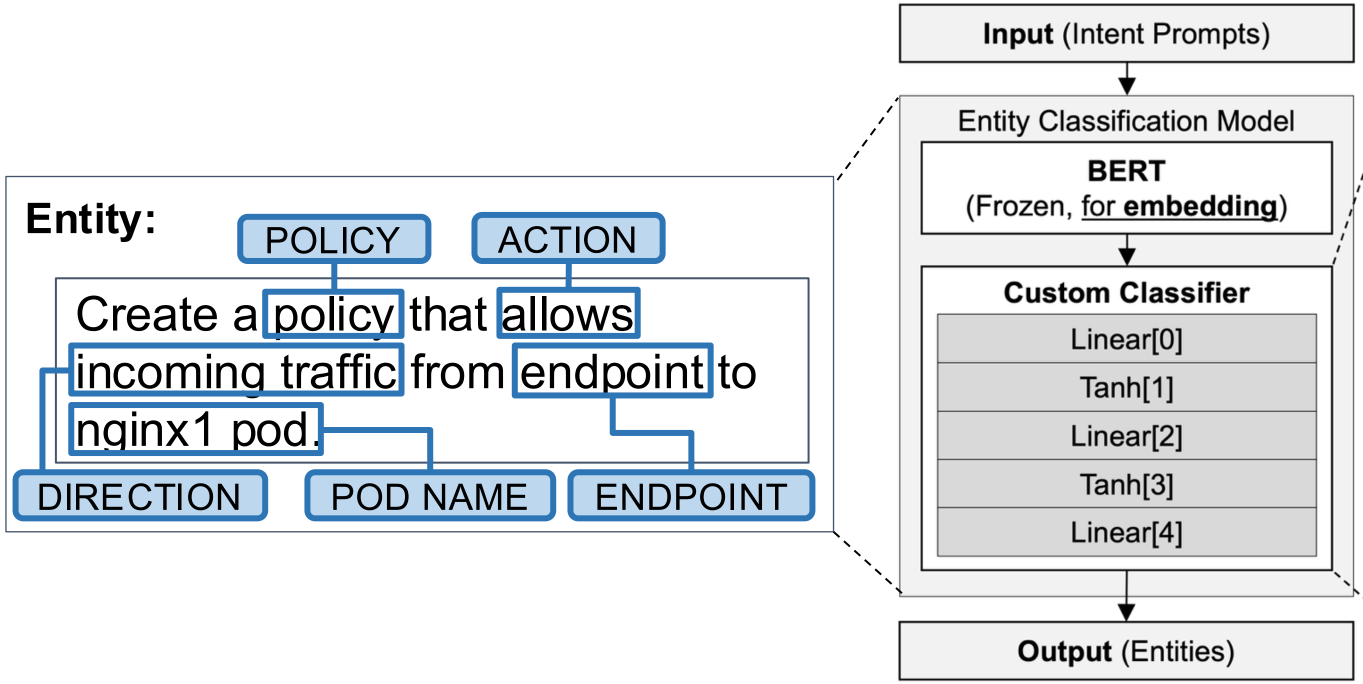
```
Name:     postgresql
Labels:   role=db
Containers:
 - env:
   - name:NGINX1_SERVICE_ADDR
     value: nginx1service:80
```

Enhanced Intent Prompt:

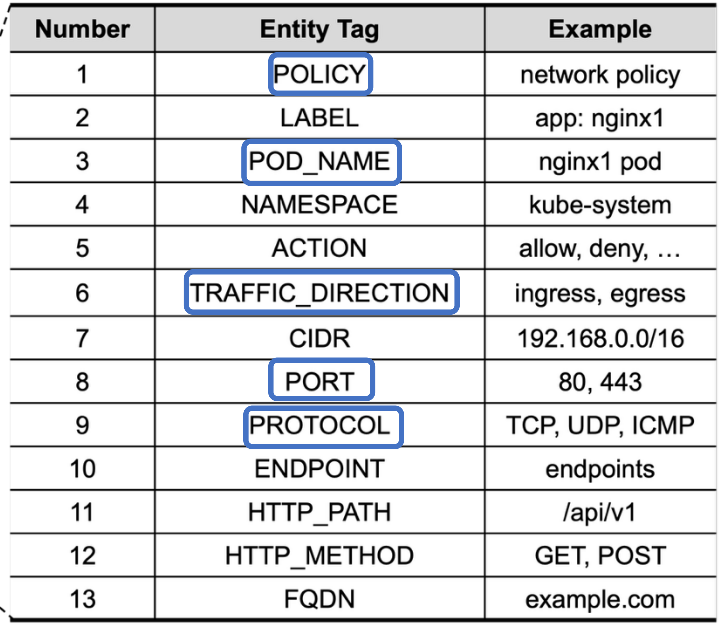
Create a CiliumNetworkPolicy that allows incoming traffic from the postgresql pod labeled 'role:db' to the nginx1 pod labeled 'svc:webapp' over '80/TCP'.

KUBETEUS Details: Entity Classification Model

- **Domain-Specific NER** using BERT
  - Identifies network policy-related entities with a custom classifier
  - Supports **13** entity types (e.g., **POLICY, LABEL, POD NAME, NAMESPACE**)

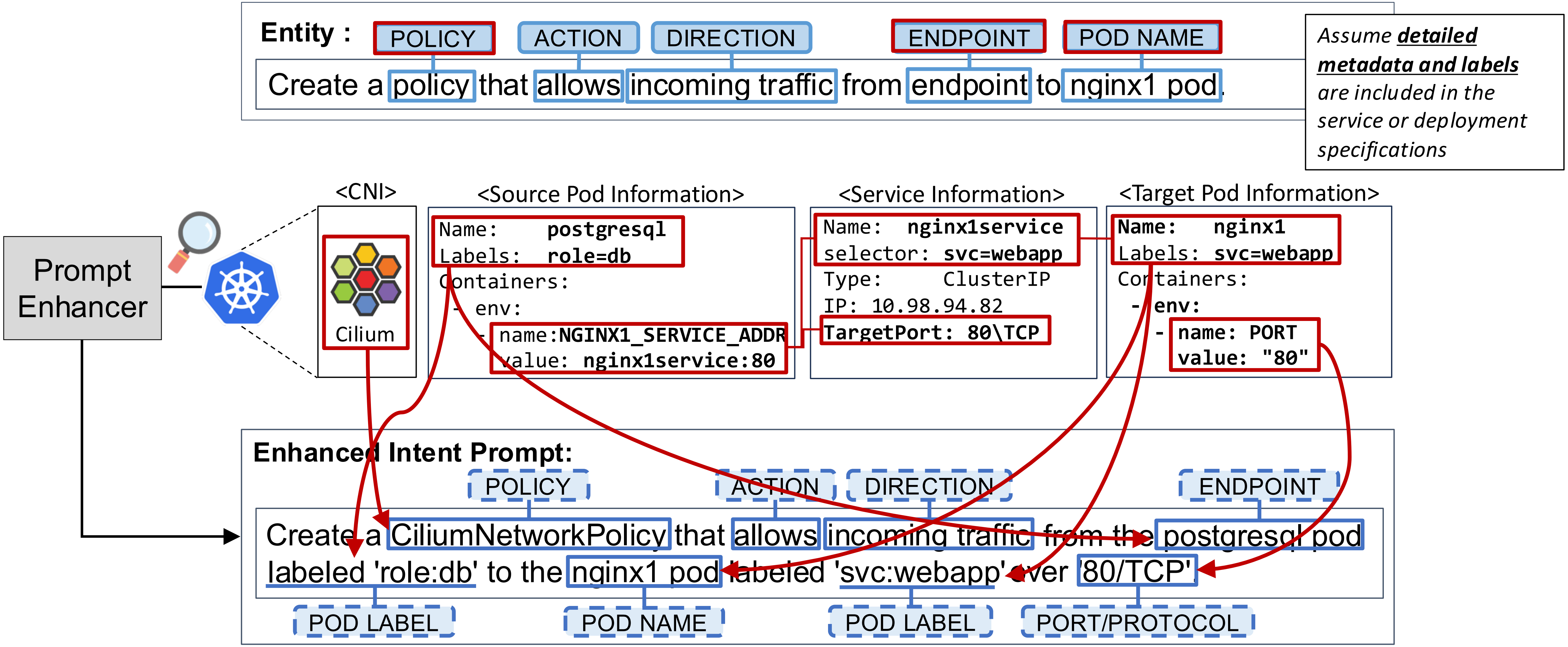


Create a policy that allows incoming traffic from endpoint to nginx1 pod.



Number	Entity Tag	Example
1	POLICY	network policy
2	LABEL	app: nginx1
3	POD_NAME	nginx1 pod
4	NAMESPACE	kube-system
5	ACTION	allow, deny, …
6	TRAFFIC_DIRECTION	ingress, egress
7	CIDR	192.168.0.0/16
8	PORT	80, 443
9	PROTOCOL	TCP, UDP, ICMP
10	ENDPOINT	endpoints
11	HTTP_PATH	/api/v1
12	HTTP_METHOD	GET, POST
13	FQDN	example.com

KUBETEUS Details: Prompt Engineering



Entity : POLICY ACTION DIRECTION ENDPOINT POD NAME

Create a policy that allows incoming traffic from endpoint to nginx1 pod.

Assume ***detailed metadata and labels*** *are included in the service or deployment specifications*

Prompt Enhancer

<CNI>

Cilium

<Source Pod Information>

```
Name:     postgresql
Labels:   role=db
Containers:
  - env:
    - name:NGINX1_SERVICE_ADDR
      value: nginx1service:80
```

<Service Information>

```
Name:    nginx1service
selector:  svc=webapp
Type:      ClusterIP
IP:  10.98.94.82
TargetPort:  80\TCP
```

<Target Pod Information>

```
Name:    nginx1
Labels:  svc=webapp
Containers:
  - env:
    - name: PORT
      value: "80"
```

Enhanced Intent Prompt:

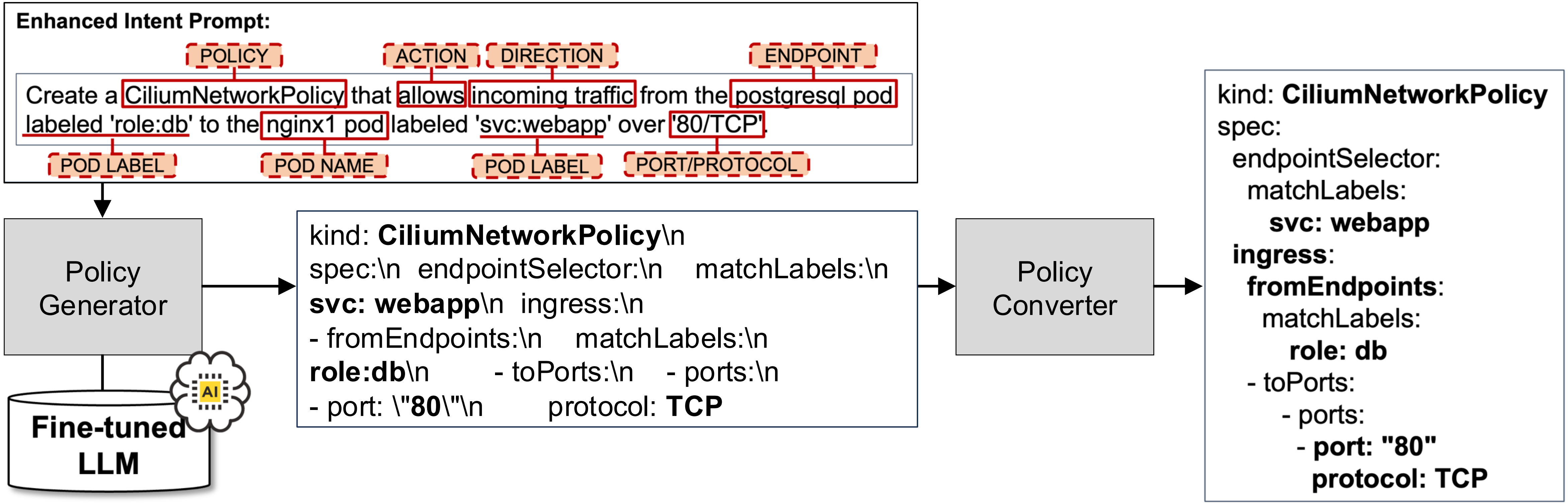
POLICY ACTION DIRECTION ENDPOINT

Create a CiliumNetworkPolicy that allows incoming traffic from the postgresql pod labeled 'role:db' to the nginx1 pod labeled 'svc:webapp' over '80/TCP'.

POD LABEL POD NAME POD LABEL PORT/PROTOCOL

KUBETEUS Details: Policy Processor

- **(iii) Policy Generator**: Generates network policies using an LLM (text-to-text model)
- **(iv) Policy Converter**: Converts generated policies into CNI-specific formats



Enhanced Intent Prompt:

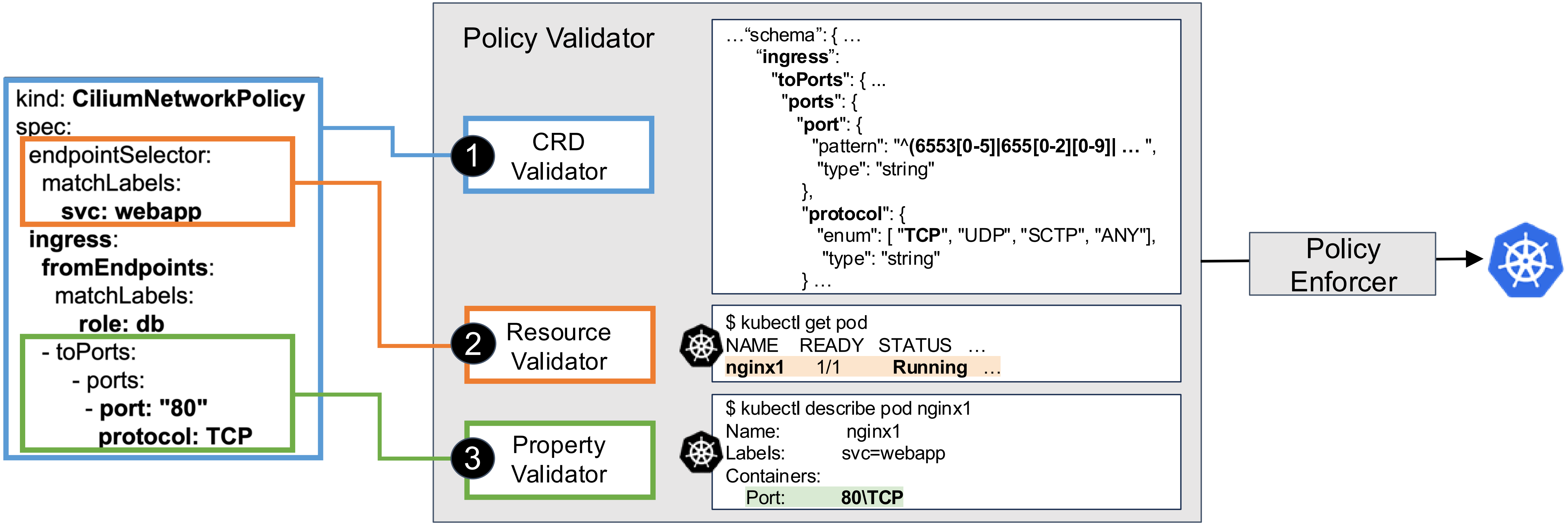
Create a CiliumNetworkPolicy that allows incoming traffic from the postgresql pod labeled 'role:db' to the nginx1 pod labeled 'svc:webapp' over '80/TCP'.

```
kind: CiliumNetworkPolicy\n
spec:\n  endpointSelector:\n    matchLabels:\n
svc: webapp\n  ingress:\n
- fromEndpoints:\n    matchLabels:\n
role:db\n        - toPorts:\n    - ports:\n
- port: \"80\"\n        protocol: TCP
```

```
kind: CiliumNetworkPolicy
spec:
  endpointSelector:
    matchLabels:
      svc: webapp
  ingress:
    fromEndpoints:
      matchLabels:
        role: db
    - toPorts:
        - ports:
          - port: "80"
            protocol: TCP
```

KUBETEUS Details: Policy Validator and Enforcer

- Ensures that policies are correctly configured and applicable to the cluster environment: (1) ***CRD Validator***, (2) ***Resource Validator***, (3) ***Property Validator***



```
kind: CiliumNetworkPolicy
spec:
  endpointSelector:
    matchLabels:
      svc: webapp
  ingress:
    fromEndpoints:
      matchLabels:
        role: db
  - toPorts:
      - ports:
        - port: "80"
          protocol: TCP
```

```
…"schema": { …
    "ingress":
      "toPorts": { ...
        "ports": {
          "port": {
            "pattern": "^(6553[0-5]|655[0-2][0-9]| … ",
             "type": "string"
          },
          "protocol": {
             "enum": [ "TCP", "UDP", "SCTP", "ANY"],
              "type": "string"
          } …
```

```
$ kubectl get pod
NAME     READY   STATUS    …
nginx1     1/1       Running   …
```

```
$ kubectl describe pod nginx1
Name:           nginx1
Labels:         svc=webapp
Containers:
   Port:           80\TCP
```

Evaluation: Performance of Entity Classifier

- Achieves over **97% accuracy (97.3% F1-score)** using a BERT-based classifier
- Both BERT and RoBERTa surpass 97% F1, showing balanced precision and recall
- Remaining ~3% errors are mostly due to ambiguous entities or non-standard input formats
    - Model limitations are complemented with deterministic, rule-based pattern analysis

Model	Accuracy	#Origin	#Train	#Test
BERT	0.971	0.974	0.972	0.973
RoBERTa	0.961	0.968	0.961	0.965

Table 3. The summary of entity classifier performance with metrics.

Evaluation: Performance of Fine-tuned LLM

- Achieves a ***360%*** increase in BLEU after fine-tuning (Mistral-7B-Instruct-v0.2)
- Achieves a ***233%*** increase in ROUGE-2 after fine-tuning (Mistral-7B-Instruct-v0.2)
- Improves METEOR and chrF++ by approximately 26% and 22%,respectively

Type	Model Name	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L	chrF++
Baseline Model	Deepseek-coder-7b-instruct-v1.5	0.52	0.79	0.71	0.57	0.69	0.80
	Meta-Llama-3-8B-Instruct	0.19	0.36	0.53	0.27	0.44	0.44
	codegemma-7b-it	0.49	0.75	0.72	0.59	0.70	0.80
	Mistral-7B-Instruct-v0.2	0.10	0.37	0.34	0.15	0.29	0.47
	CodeLlama-7b-Instruct-hf	0.28	0.53	0.58	0.33	0.47	0.64
Fine-tuning Model	Deepseek-coder-7b-instruct-v1.5	**0.76**	0.81	**0.87**	0.78	0.83	**0.87**
	Meta-Llama-3-8B-Instruct	0.41	0.60	0.66	0.59	0.64	0.68
	codegemma-7b-it	0.72	**0.82**	0.85	**0.80**	**0.84**	**0.87**
	Mistral-7B-Instruct-v0.2	0.46	0.55	0.57	0.50	0.54	0.65
	CodeLlama-7b-Instruct-hf	0.31	0.54	0.59	0.34	0.48	0.65

Conclusion

- ***Automated*** Framework for Network Policy Generation
  - KUBETEUS provides a fully automated solution for generating network policies in real-world cloud-native environments.

- LLM + Prompt Engineering ***Integration***
  - This is the first study to integrate prompt engineering with fine-tuned LLMs for cloud-native environments—moving beyond traditional log-based and static approaches.

- ***Accurate*** & ***Reliable*** Policy Generation
  - A multi-step validation process ensures accuracy, while fine-tuning boosts performance in both entity recognition and policy generation.

THANK YOU

zxx0313@inu.ac.kr