INCHEON NATIONAL UNIVERSITY

한국정보보호학회
Korea Institute of Information Security & Cryptology

CYBERSECURITY CONVERGENCE LAB

AURORA: Automated Unified Resource-Oriented rule Alignment for Cloud-Native Environments

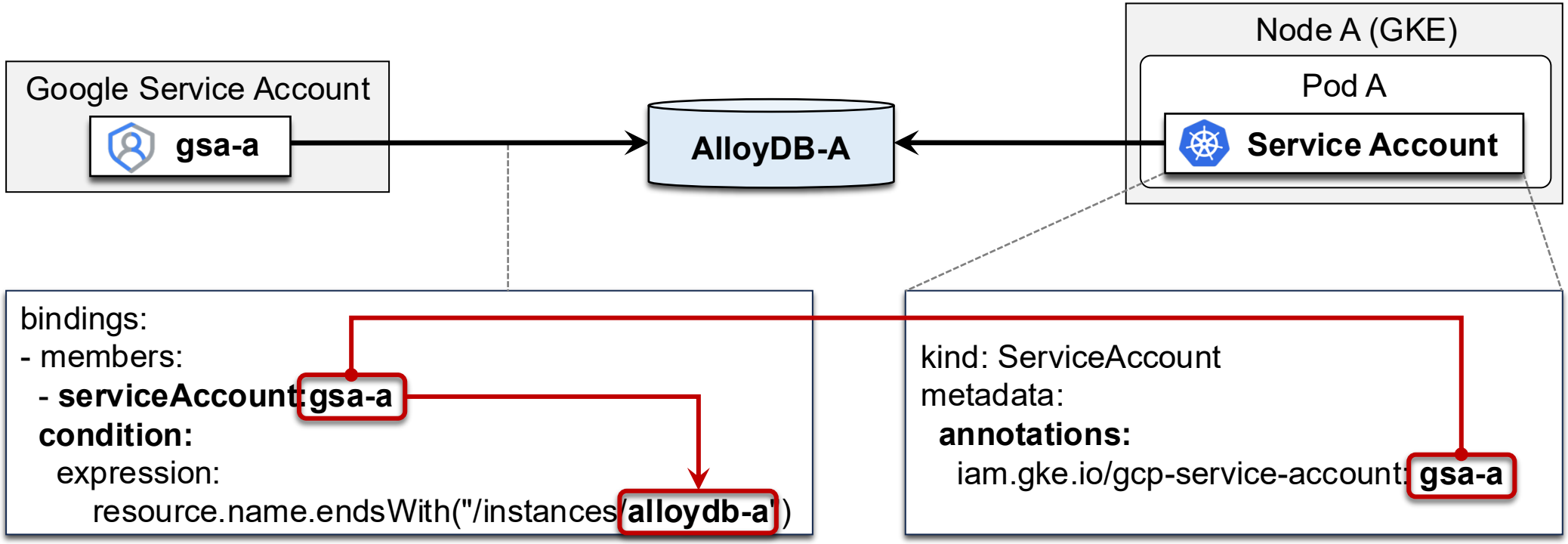
Minkyung Kim

Department of Computer Science & Engineering

Incheon National University

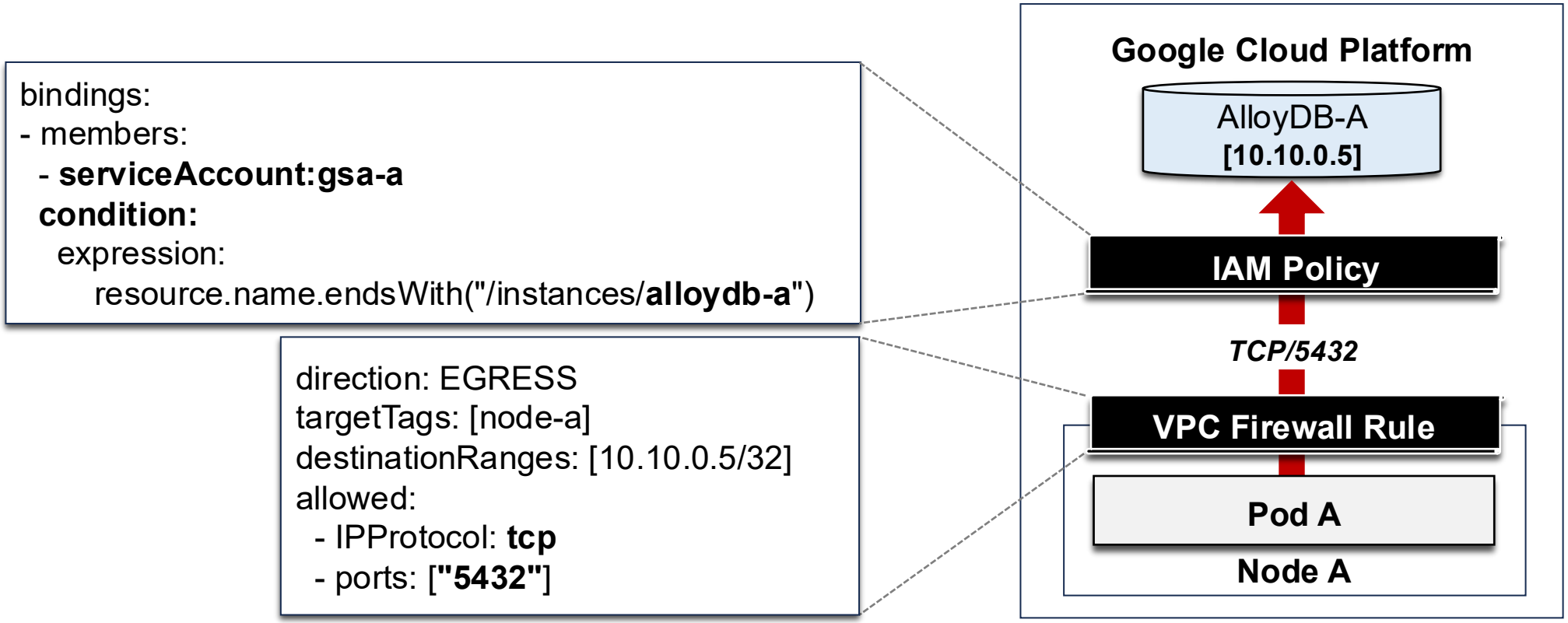
Background [1] Cloud Resource Access Architecture

- In GKE, Pods access **Google Service Account permissions** through Kubernetes **Service Account annotations**.



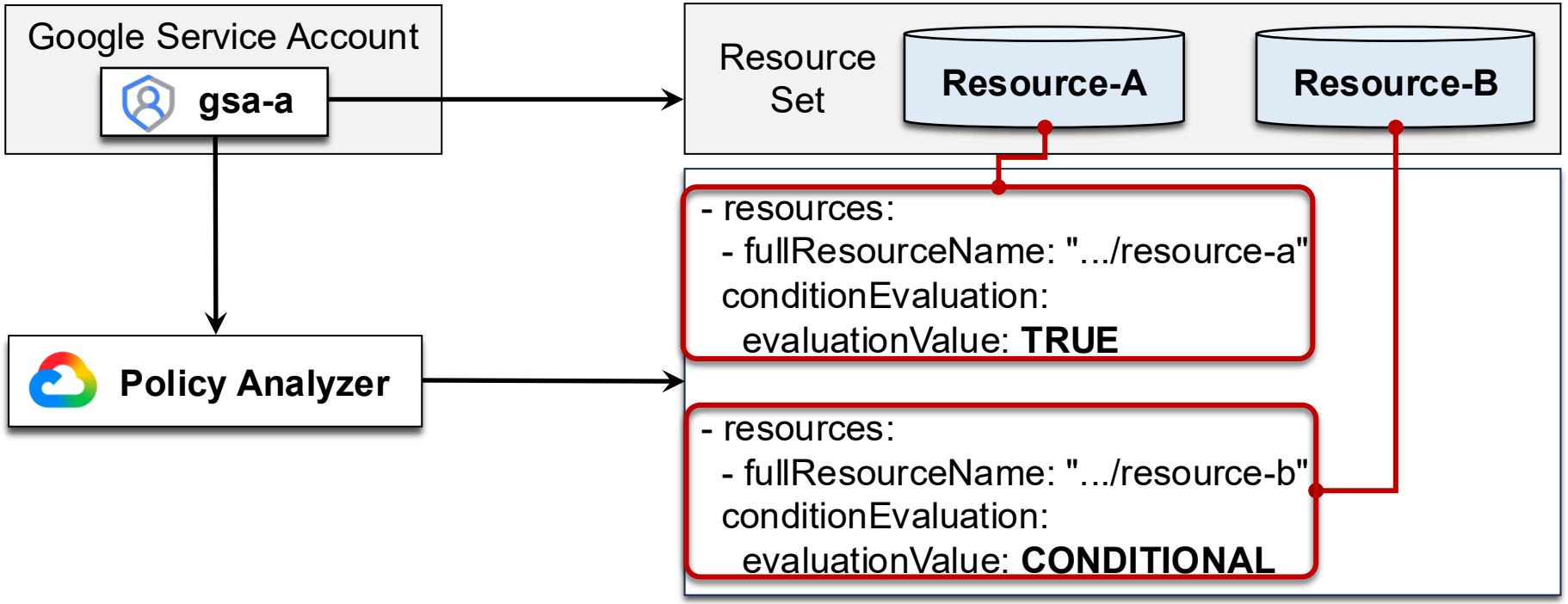
Background [2] IAM and Cloud firewall-Based Access Control

- **IAM**: Controls access permissions to cloud resources.
- **VPC Firewall**: Controls network traffic.



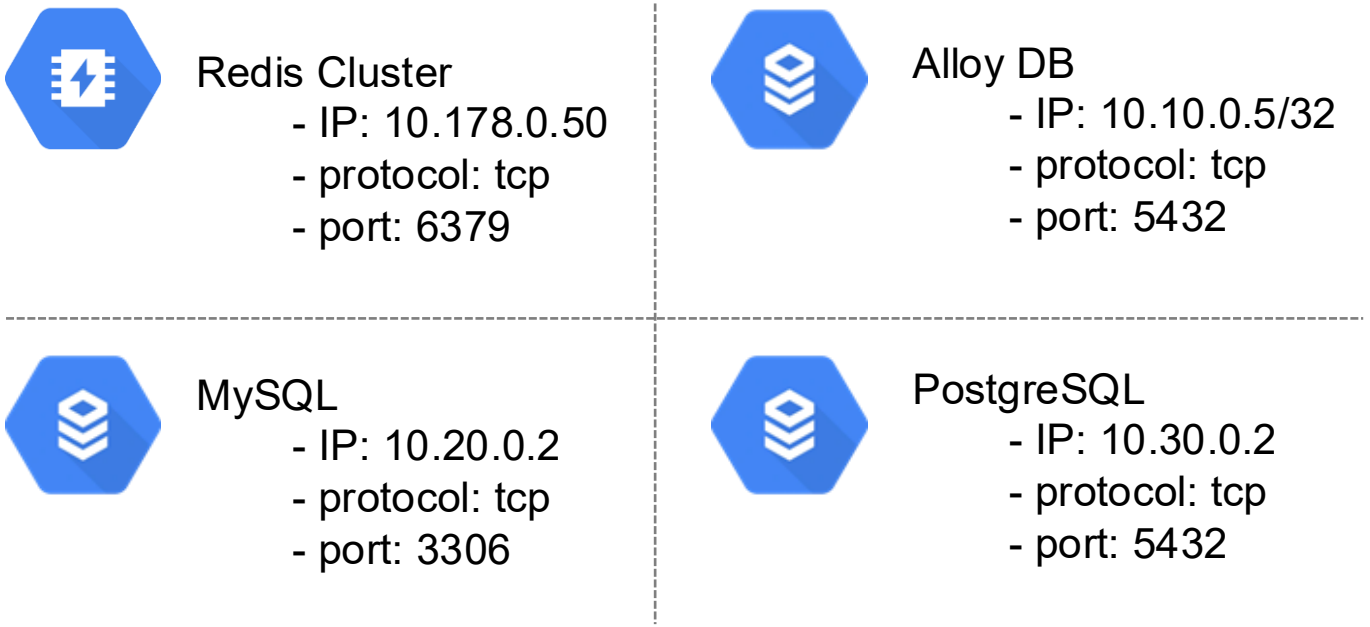
Background [3] Policy Analyzer

- **Policy Analyzer** identifies **Service Account's resource access** from IAM policies.
- **Conditions**: **TRUE** if met, **CONDITIONAL** if unevaluable; excluded if unmet.



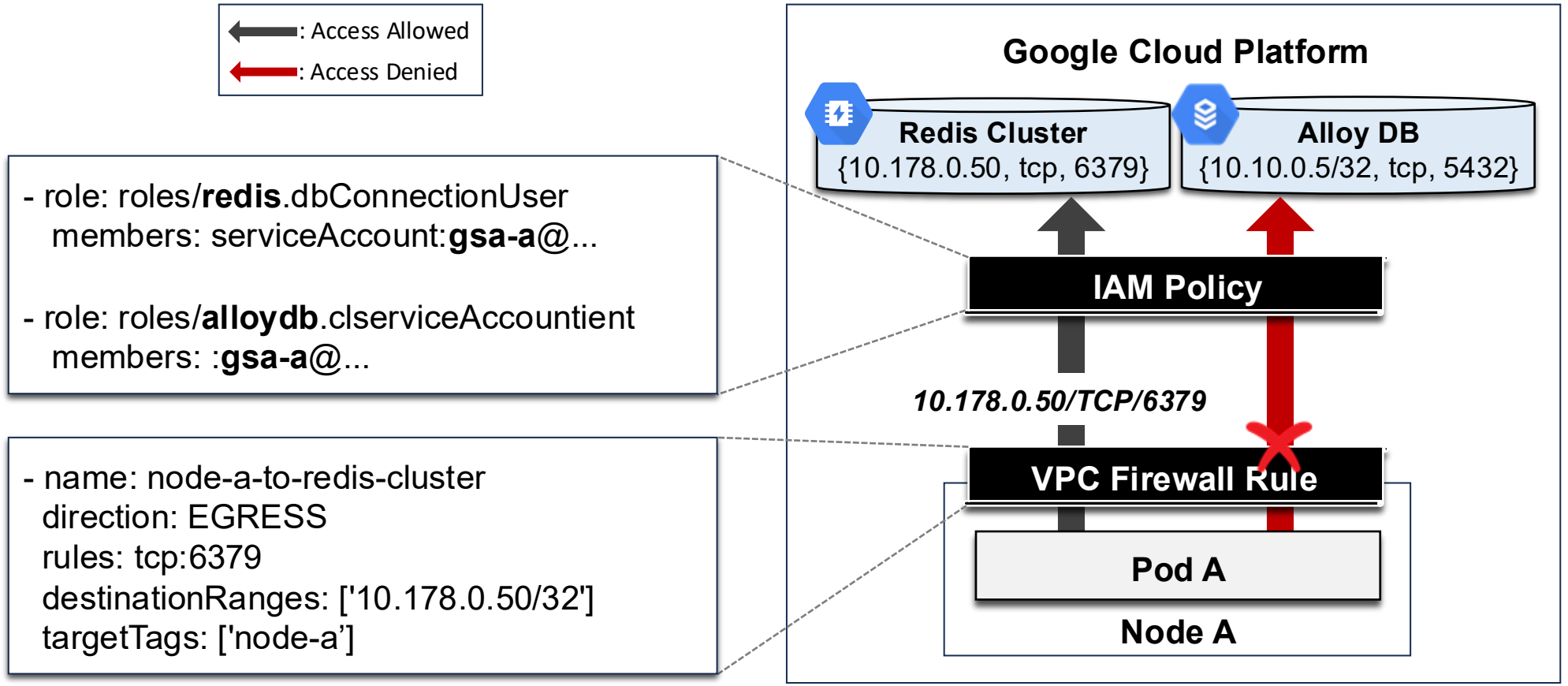
Problem Statement [1] Heterogeneous Network Requirements Across Resources

- **Network access requirements** vary by **cloud resource.**



Problem Statement [2] IAM–VPC Firewall Mismatch

- Firewall rules required for **IAM-authorized access** may be missing.



Problem Statement [3] Dynamic Changes in Cloud-Native Environments

- Unhandled **Pod, resource, or IAM changes** may **overpermit or block legitimate access.**

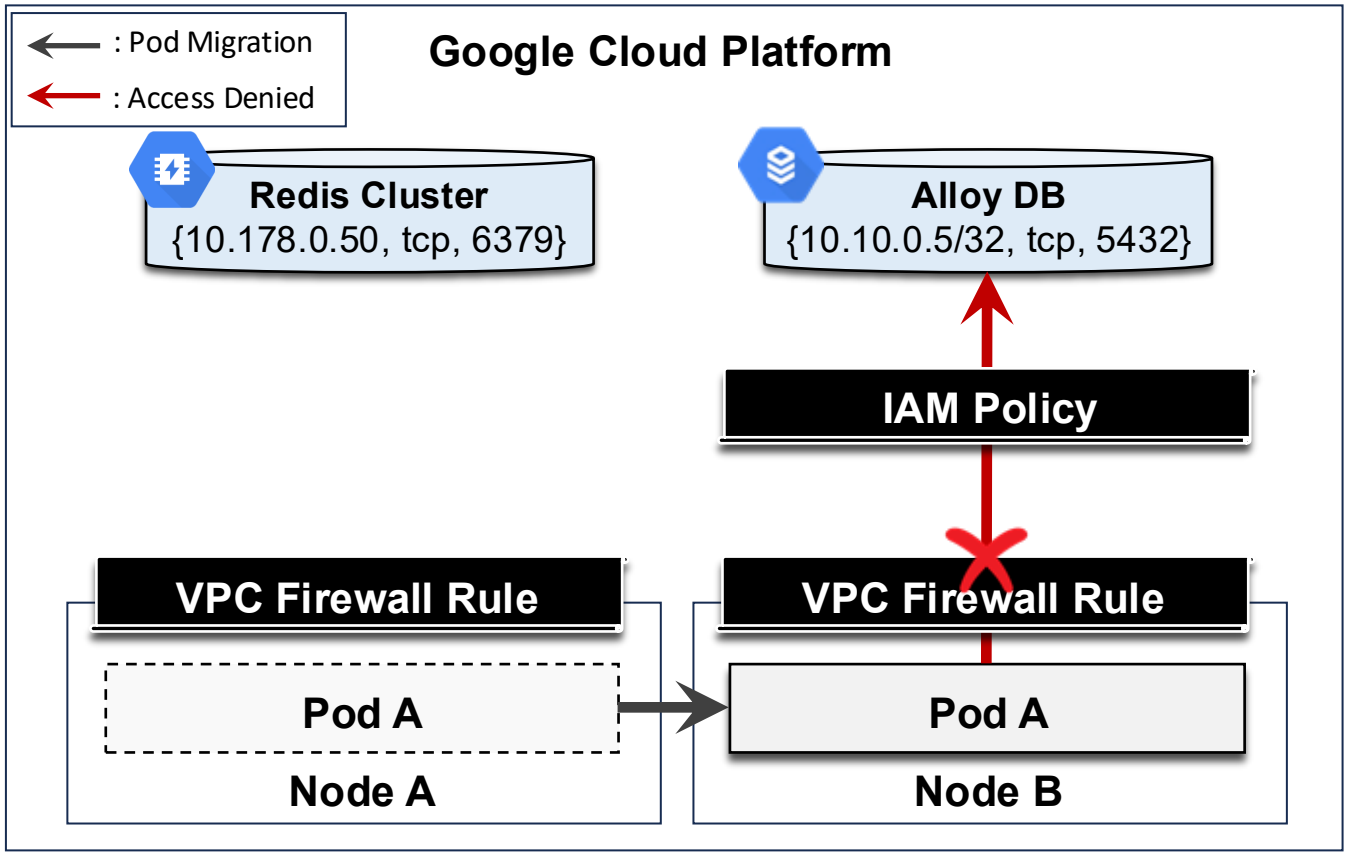
```
- name: node-a-to-redis-cluster
  direction: EGRESS
  rules: tcp:6379
  destinationRanges: ['10.178.0.50/32']
  targetTags: ['node-a']

- name: node-a-to-alloy-db
  direction: EGRESS
  rules: tcp:5432
  destinationRanges: ['10.10.0.5/32']
  targetTags: ['node-a']
```

Firewall(A)

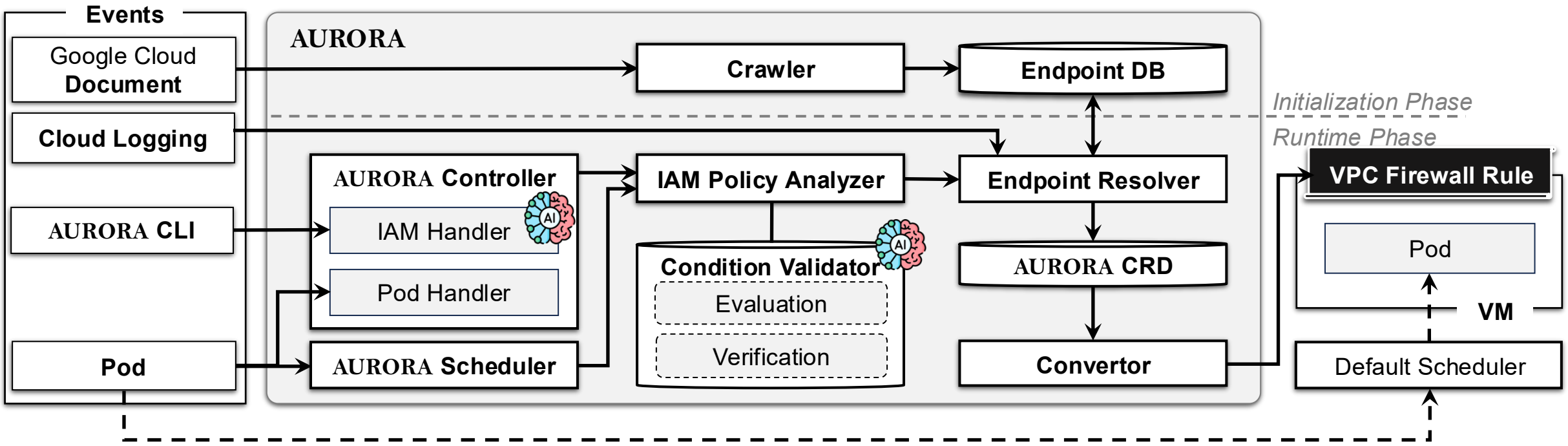
```
- name: node-b-to-redis-cluster
  direction: EGRESS
  rules: tcp:6379
  destinationRanges: ['10.178.0.50/32']
  targetTags: ['node-b']
```

Firewall(B)



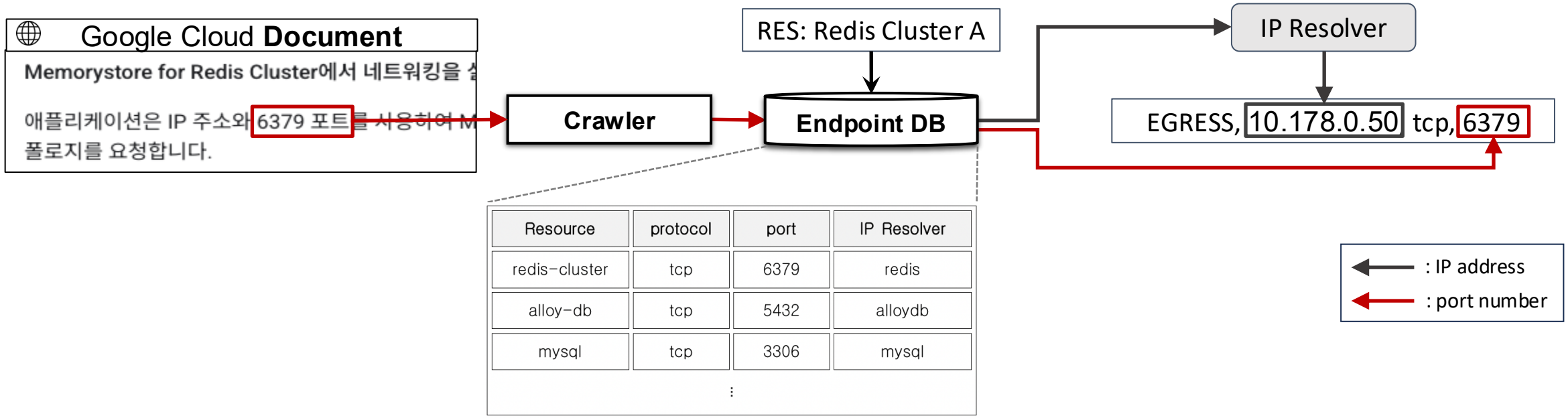
AURORA Overview

- Detect **Dynamic Events** and Reconfigure **VPC Firewall Rules.**



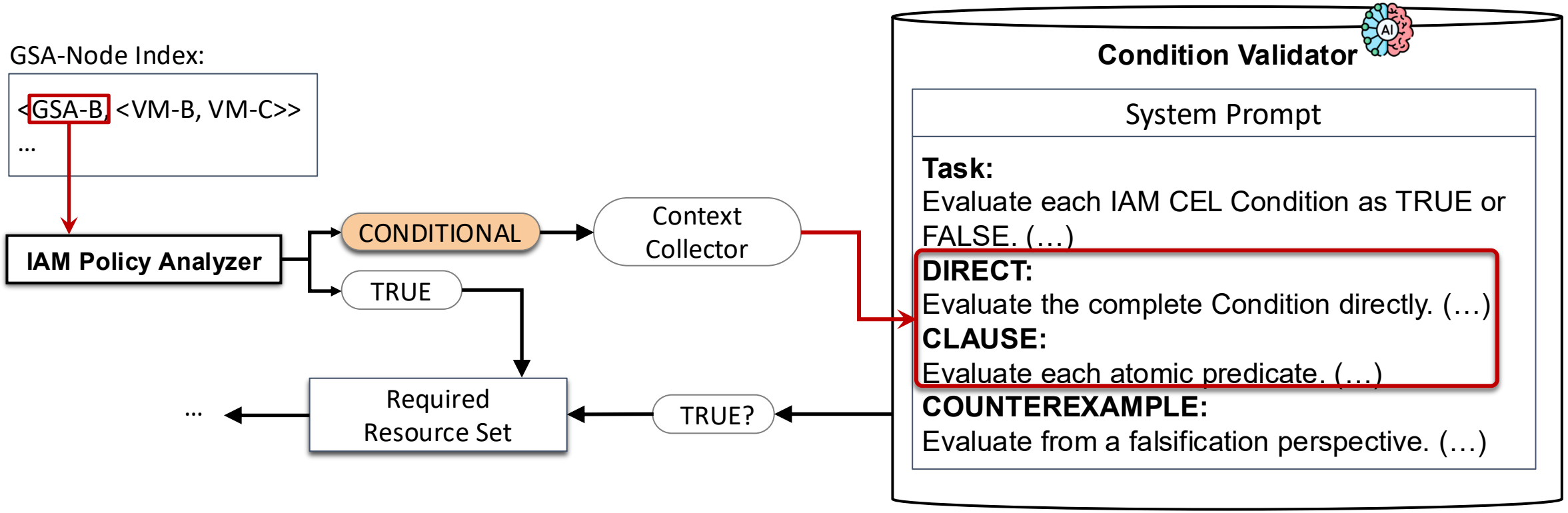
Detail [1] Network Access Requirements Derivation

- Combine the **per-resource port·protocol DB** with actual **endpoint(IP)**.



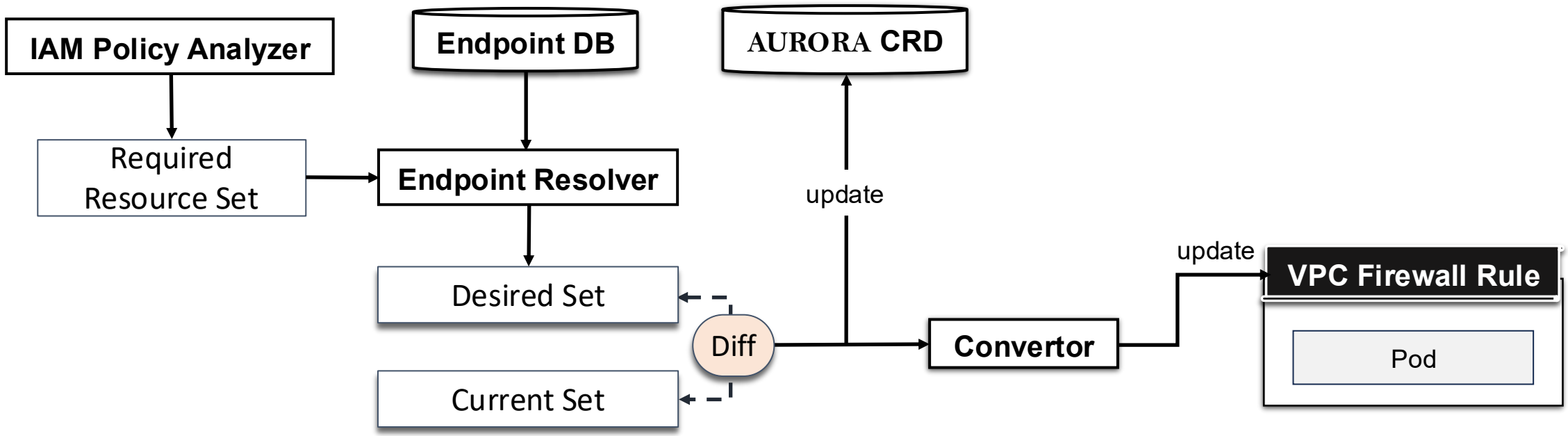
Detail [2] IAM-Based Firewall Rule Configuration

- Extract resources accessible to **Google Service Accounts** via **Policy Analyzer**.
- Evaluate **CONDITIONAL** results using the **Condition Validator**.



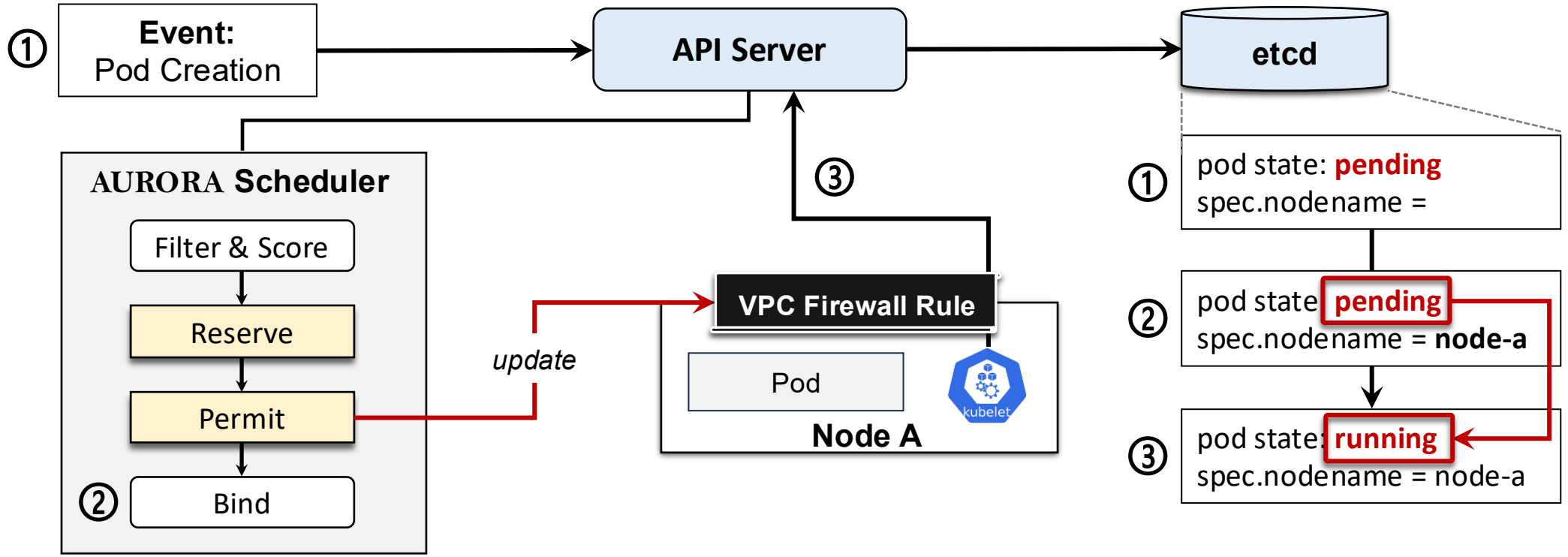
Detail [2] IAM-Based Firewall Rule Configuration

- Convert **authorized resources** into **VPC firewall rules** and reconcile with **current rules**.



Detail [3] Firewall Reconfiguration for Dynamic Changes

- Detect **change events** and ensure **VPC firewall rule ordering** for each change type.
  - Pod Creation



Evaluation [1] Translation Accuracy

- Firewall Reconfiguration upon Pod Relocation
  - w/o AURORA – Legitimate access blocked, excessive access allowed.
  - w/ AURORA – Legitimate access allowed

>_ GSA IAM Role Retrieval

```
$ gcloud projects get-iam-policy
--flatten="bindings[].members"
--filter="bindg\ingmembersings.
:serviceAccount:${WORKLOAD_GSA}"
--format="value(bindings.role)“

roles/redis.dbConnectionUser
```

>_ Firewall rule Retrieval Before Pod Migration

```
$ gcloud ... get-effective-firewalls "$NODE_A"
10.178.0.50/32    tcp    6379
10.178.0.49/32    tcp    11002
10.178.0.49/32    tcp    11003
$ gcloud ... get-effective-firewalls "$NODE_B"

No matching Redis firewall rules
```

>_ Pod Migration Due to Node Drain

```
$ kubectl drain "$NODE_A" --pod-selector="app=cartservice"
```

>_(A.1) Access Denied (w/o AURORA)

```
$ kubectl get pod -A -l cart...

NAME                                READY
STATUS
cartservice-59c9f485b4-sfkmg        0/1
CrashLoopBackOff
```

>_(B.1) Access Allowed(w/ AURORA)

```
$ kubectl get pod -A -l cart...

NAME                                READY
STATUS
cartservice-59c9f485b4-sfkmg        1/1
Running
```

>_(A.2) Firewall rule Retrieval (w/o AURORA)

```
$ gcloud ... get-effective-firewalls
"$NODE_A"
10.178.0.50/32    tcp    6379
10.178.0.49/32    tcp    11002
10.178.0.49/32    tcp    11003

$ gcloud ... get-effective-firewalls
"$NODE_B"
No matching Redis firewall rules
```

>_(B.2) Firewall rule Retrieval (w/ AURORA)

```
$ gcloud ... get-effective-firewalls
"$NODE_A"
No matching Redis firewall rules

$ gcloud ... get-effective-firewalls
"$NODE_B"
10.178.0.49/32    tcp    11003
10.178.0.49/32    tcp    11002
10.178.0.50/32    tcp    6379
```

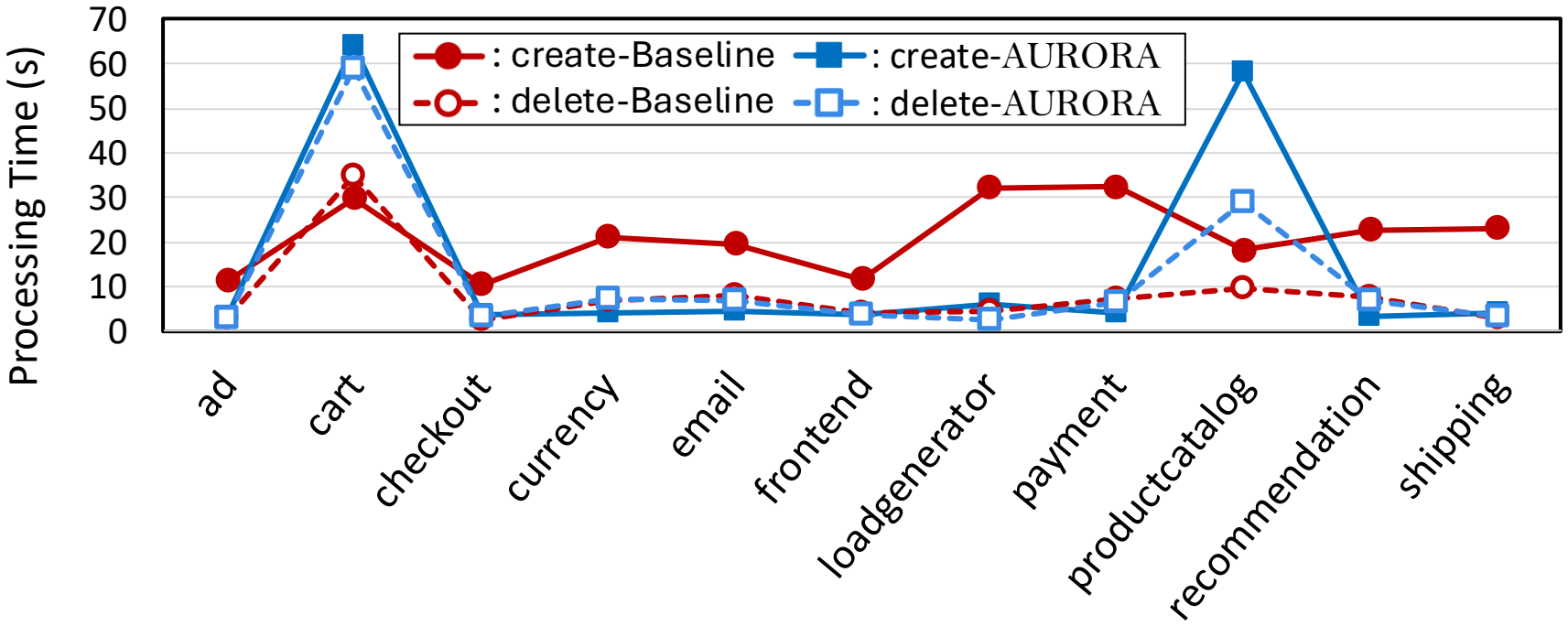
Evaluation [2] LLM Performance

- Evaluate **LLM effectiveness** in handling **conditional bindings**.
  - **Dataset:** 60 **GSA–Resource pairs** returned as **CONDITIONAL** by Policy Analyzer.
  - **Method:** Compare **baseline** and **AURORA** against **ground truth**.

	w/o AURORA	w/ AURORA
TP/TN/FP/FN	23 / 0 / 37 / 0	23 / 37 / 0 / 0
Accuracy(%)	38.33	100
Precision(%)	38.33	100
Recall(%)	100	100
F1(%)	55.42	100

Evaluation [3] overhead

- Processing Overhead for Pod Creation and Deletion
  - **~68% overhead** over baseline for **Pod deletion**.
  - Cost of automating **GSA analysis** and **network requirement derivation**.



Conclusion

- IAM-based firewall automation
- Ordered rule updates for dynamic changes
- Validated condition handling and rule reconfiguration