INCHEON NATIONAL UNIVERSITY

CYBERSECURITY CONVERGENCE LAB

ARES: 서버리스 환경에서의 에이전틱 AI 기반 실시간 권한 우회 방지 시스템

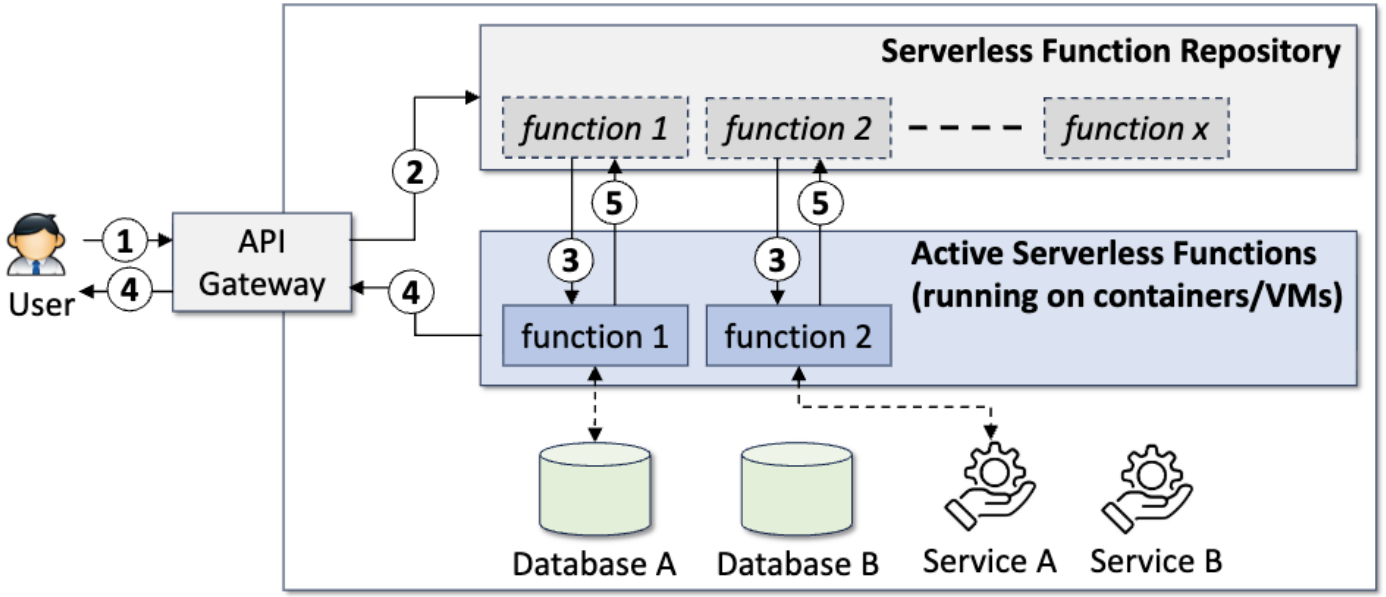
Changhee Shin

Department of Computer Science & Engineering

Incheon National University

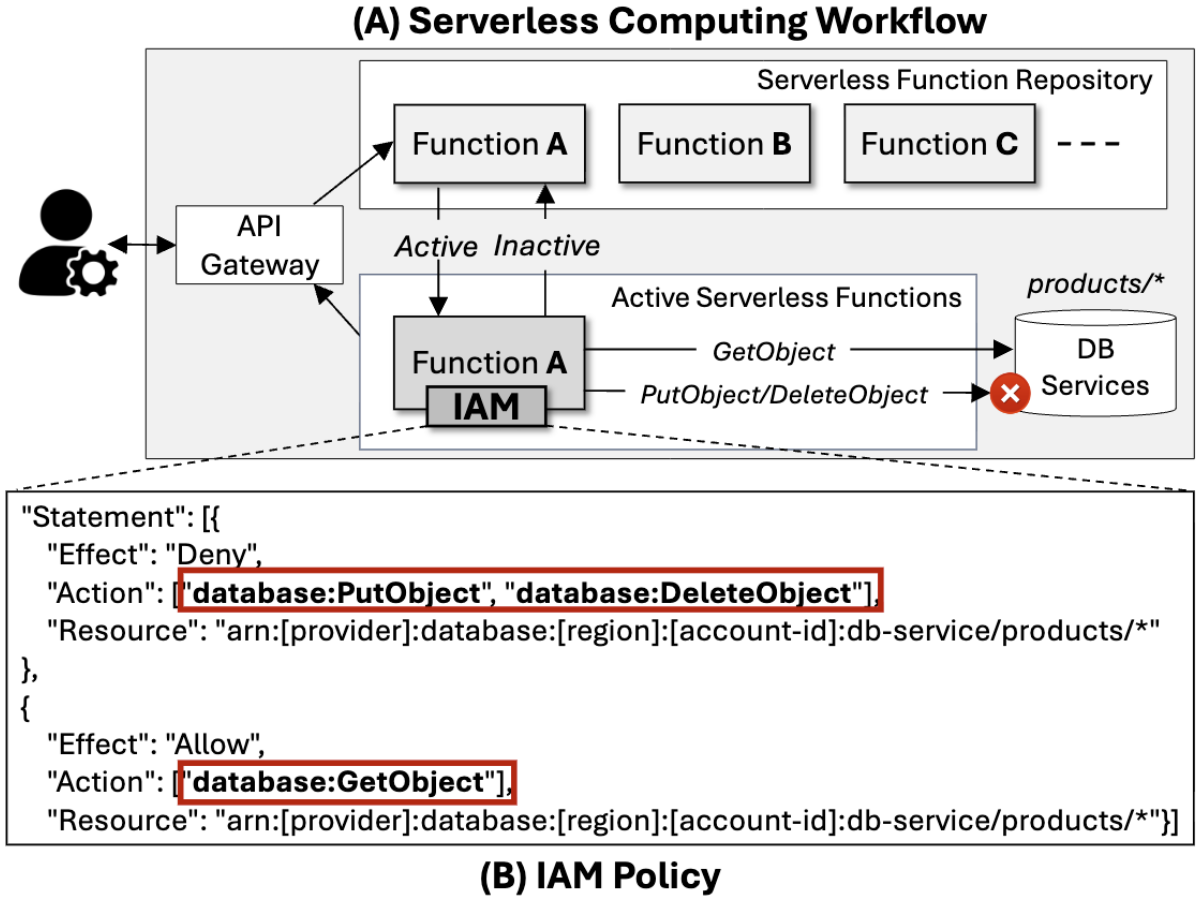
Background – Serverless

- Applications are split into independent functions.
- Functions are activated only by incoming events.
- Execution instances are created and terminated dynamically.



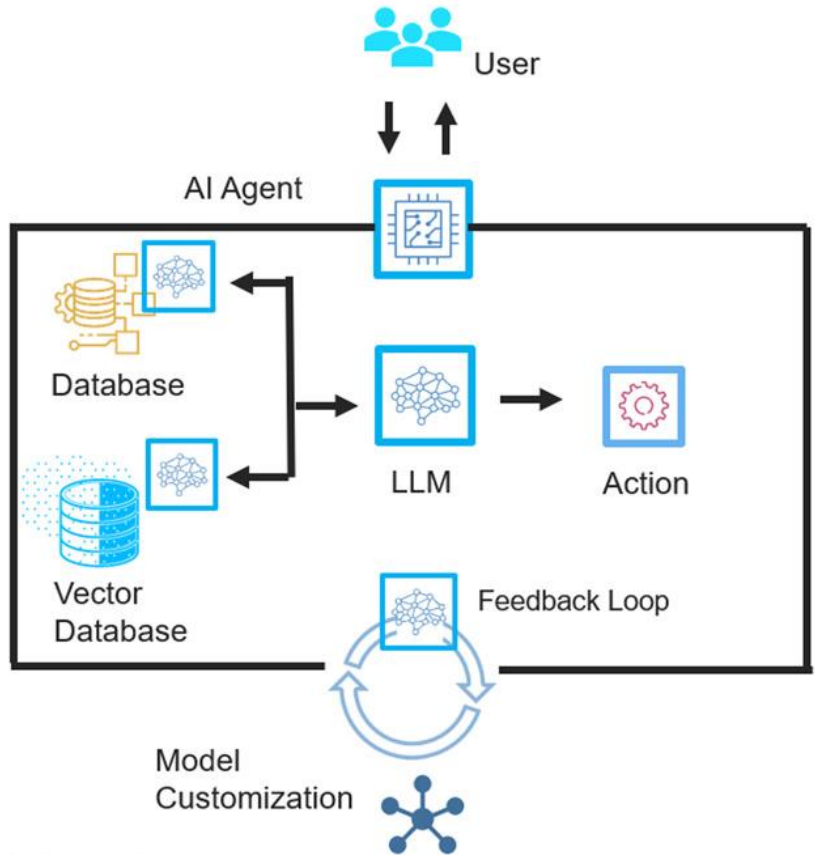
Background – Serverless IAM

- IAM defines allowed actions for each function.
- Policies specify target resources and access conditions.
- Resource access is evaluated using the function's identity.



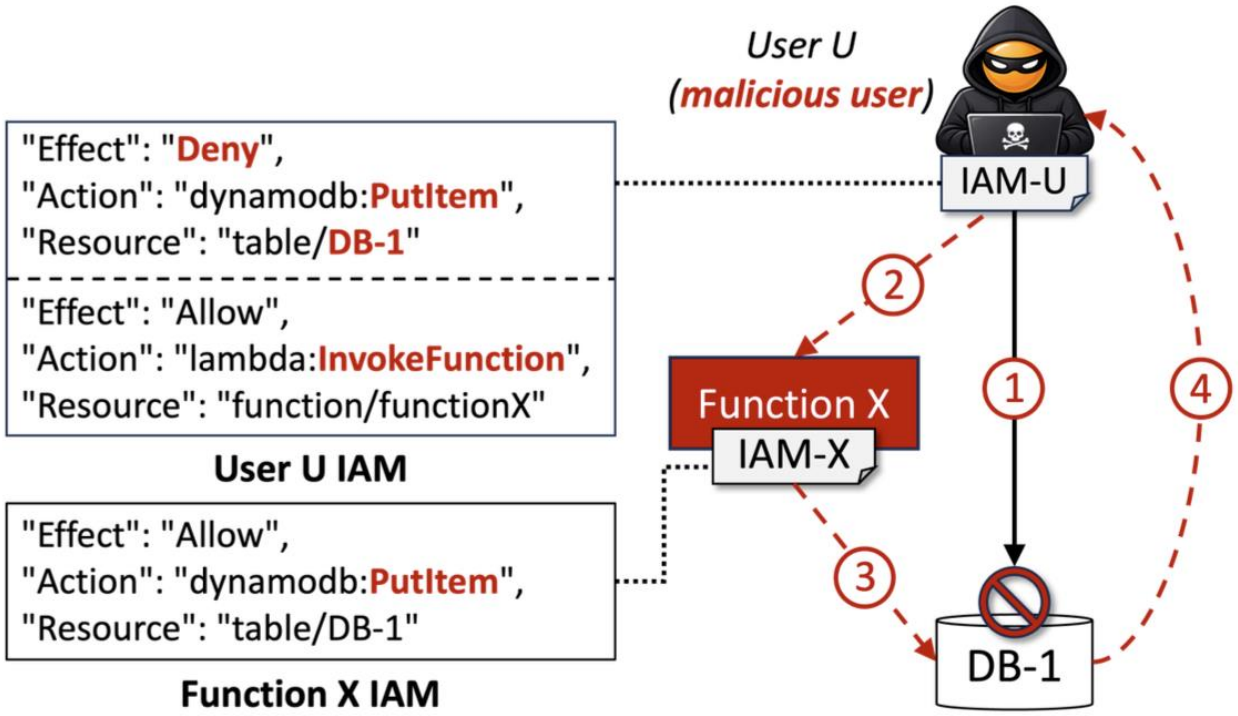
Background – 에이전틱 AI

- Plans actions toward a given objective.
- Uses external tools and knowledge sources.
- Updates decisions from observed results.



Problem Statements

- Caller-unaware IAM allows unauthorized users to reach protected resources through allowed functions.
  - A user may be denied direct access to a resource.
  - The same user may invoke a function that has that permission.
  - The platform checks the function's permission, not the caller's intent.

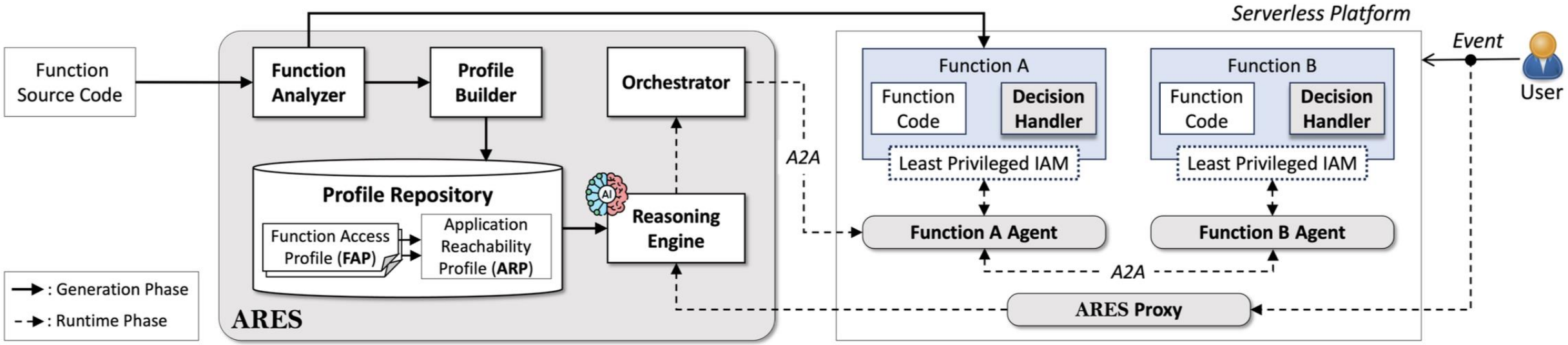


Requirements

- RQ1. How can we identify indirect privilege bypass paths?
- RQ2. How can we generate environment-aware access profiles?
- RQ3. How can we block bypass attempts before execution?

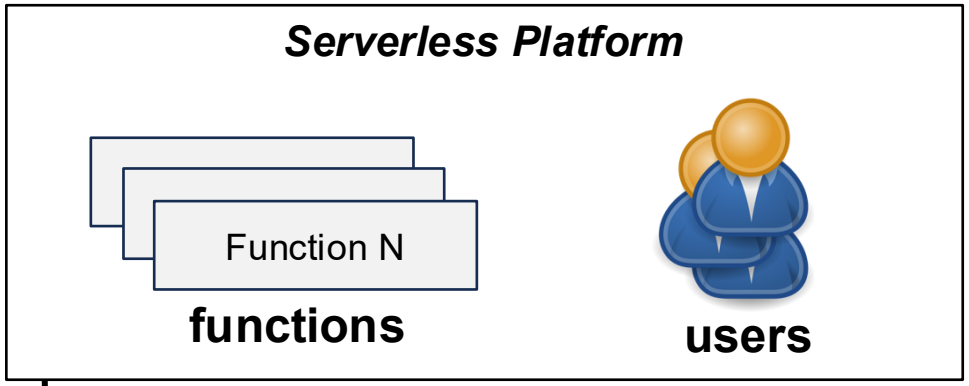
Design

- ARES combines offline profile generation with online runtime enforcement.



Details – Generation Phase

- Extracts Function Access Profiles to capture each function's real access capability.



API

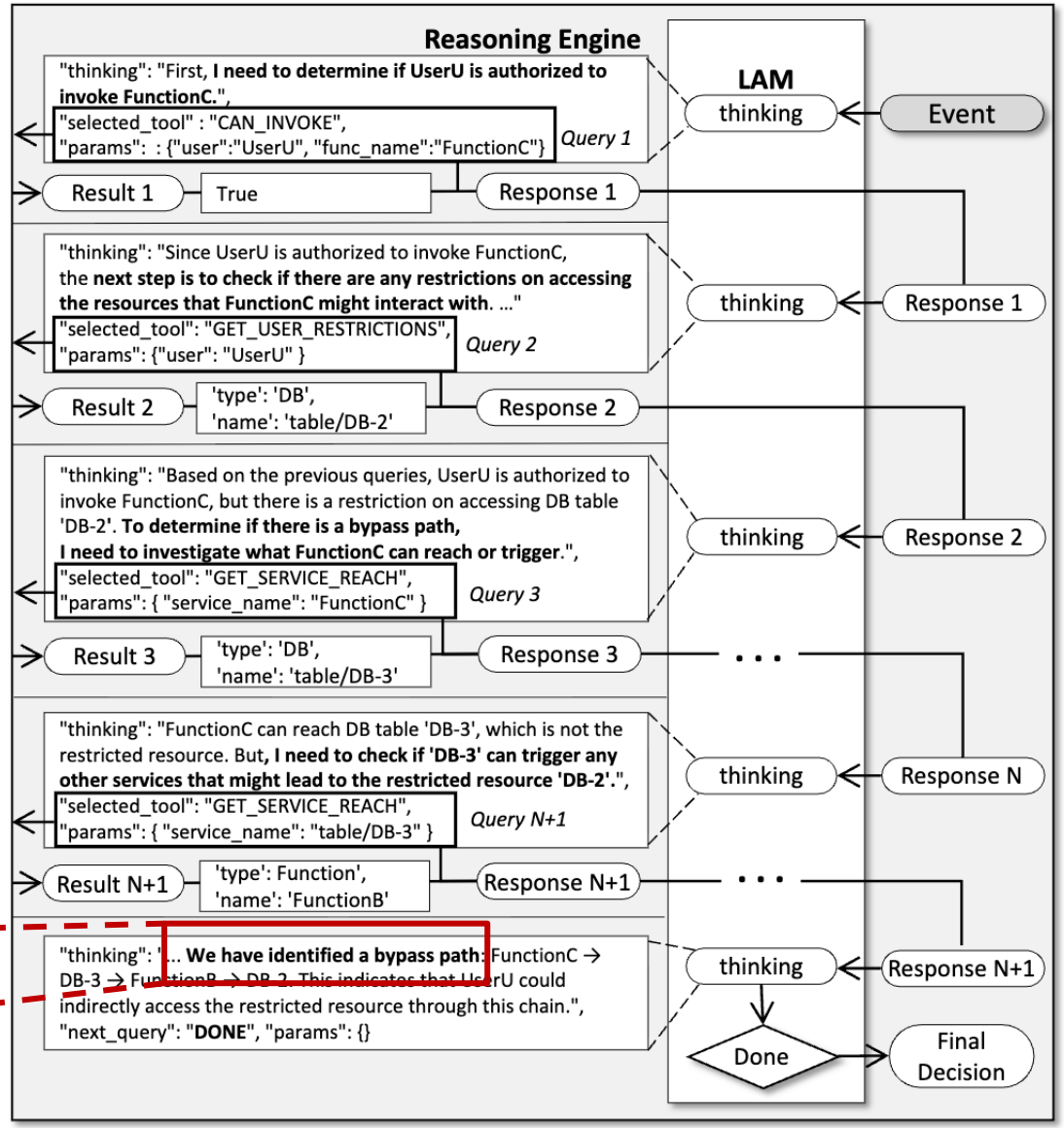
Type		Args
User		*userA*
Function		*GetItemFunc*
Database	DynamoDB	*TestDB*
Database	$DB_THUMBNAILS	*media-prod*

Function Access Profile (FAP)


```
{"function_name": "GetItemFunc",
 "metadata": {"lang": "python3.11", "provider": "gc"},
 "principle": [
  {"type": "User", "name": "userA"},
  {"type": "DynamoDB", "name": "TestDB"},
 "action": ["dynamodb:getItem"],
 "target_resource": [
  {"type": "storage", "name": "DynamoDB/TestDB"}]}
```

Details – Runtime Phase

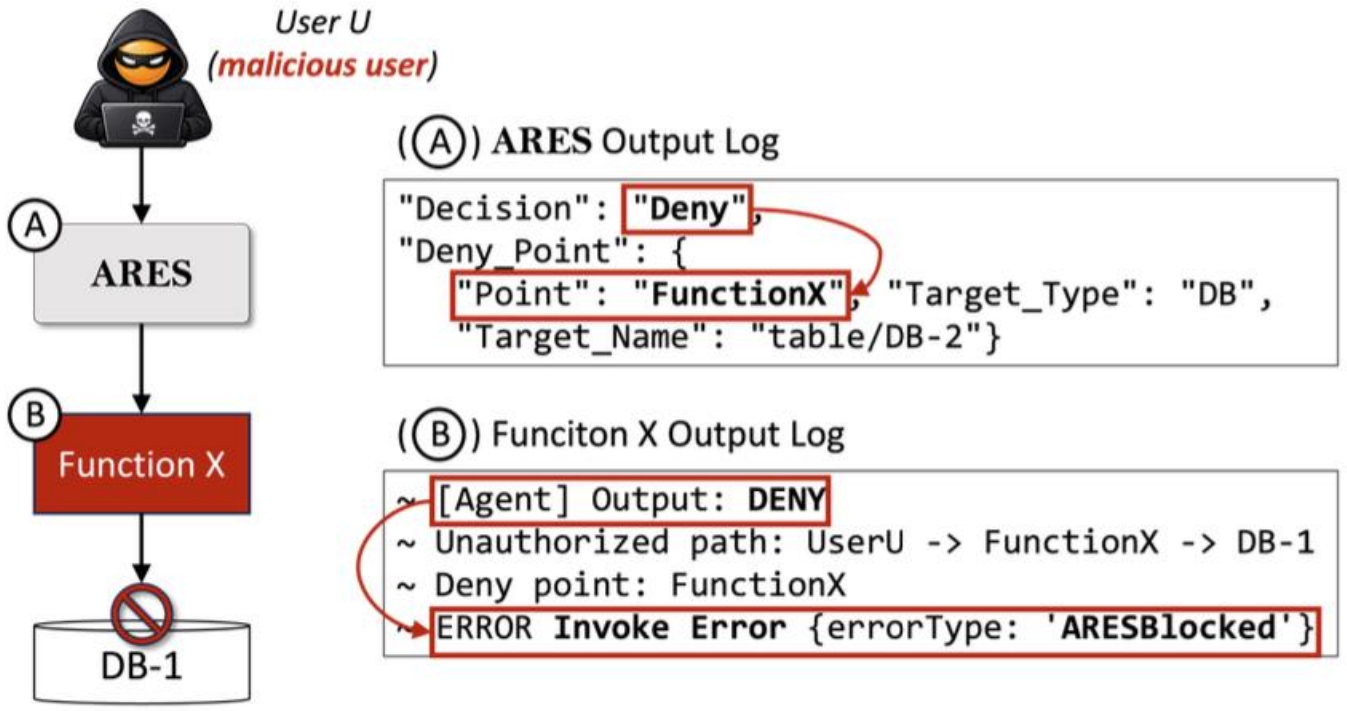
- blocks suspicious invocations by comparing caller intent with function capability.
  - The proxy captures caller, function, and target resource.
  - The reasoning engine checks the invocation against stored profiles.
  - If the path violates the caller's permission, ARES denies execution.



We have Identified a bypass path

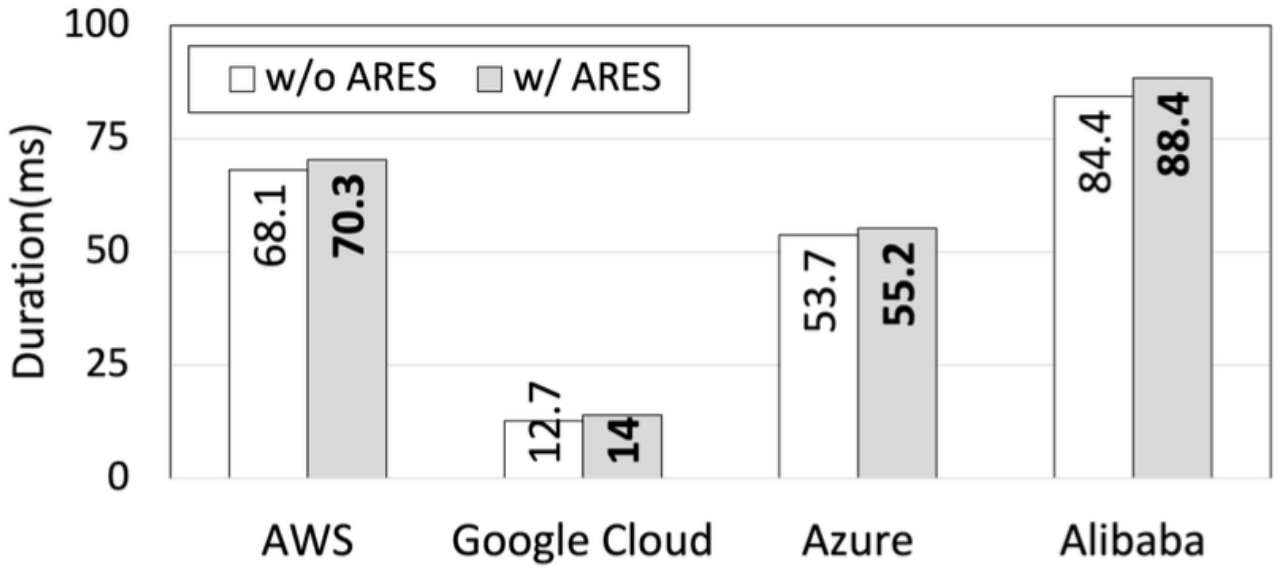
Evaluation

- Successfully blocks privilege bypass attempts across major cloud platforms.



Evaluation

- ARES achieves real-time enforcement with low runtime overhead.



Conclusion

- ARES provides real-time, invocation-path-aware protection against serverless privilege bypasses.
  - **Caller-unaware IAM** creates indirect access risks.
  - ARES profiles **function capabilities and runtime invocation paths**.
  - Agentic AI enables **context-aware detection and blocking**.

Q / A