Security Initiatives at KEK

Computing Research Center, KEK

Jo UETA

Main Tasks of the Security Team

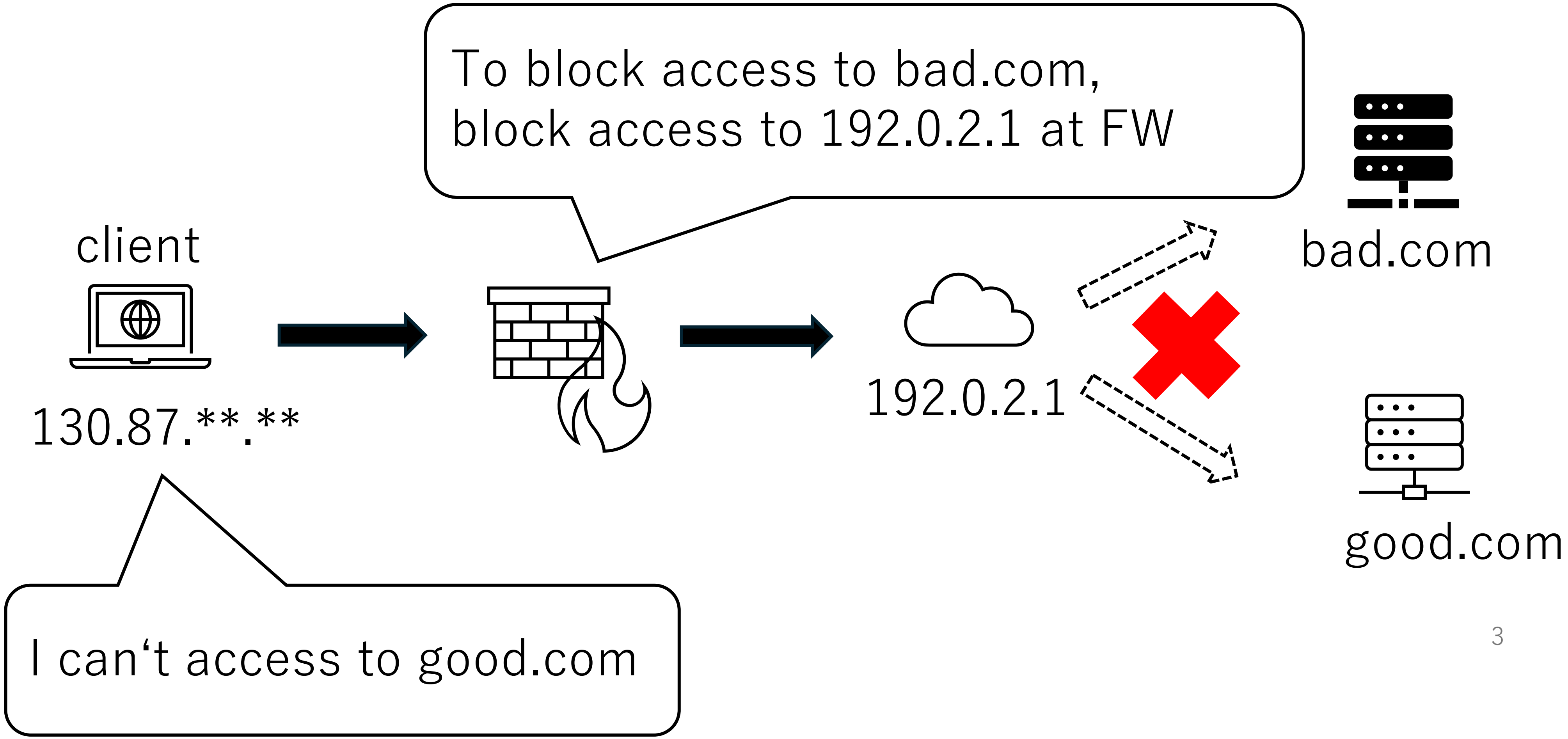
- Managing Security Systems
  - Firewall
  - Vulnerability Assessment System
  - ①DNS Firewall
  - System for Collecting Logs
    - ②DNS Response Log

- Security Operations
  - Incident Response
  - ③Report Incident Response Activities
  - Investigation
  - Collaboration with Other Organizations

Three main topics today!

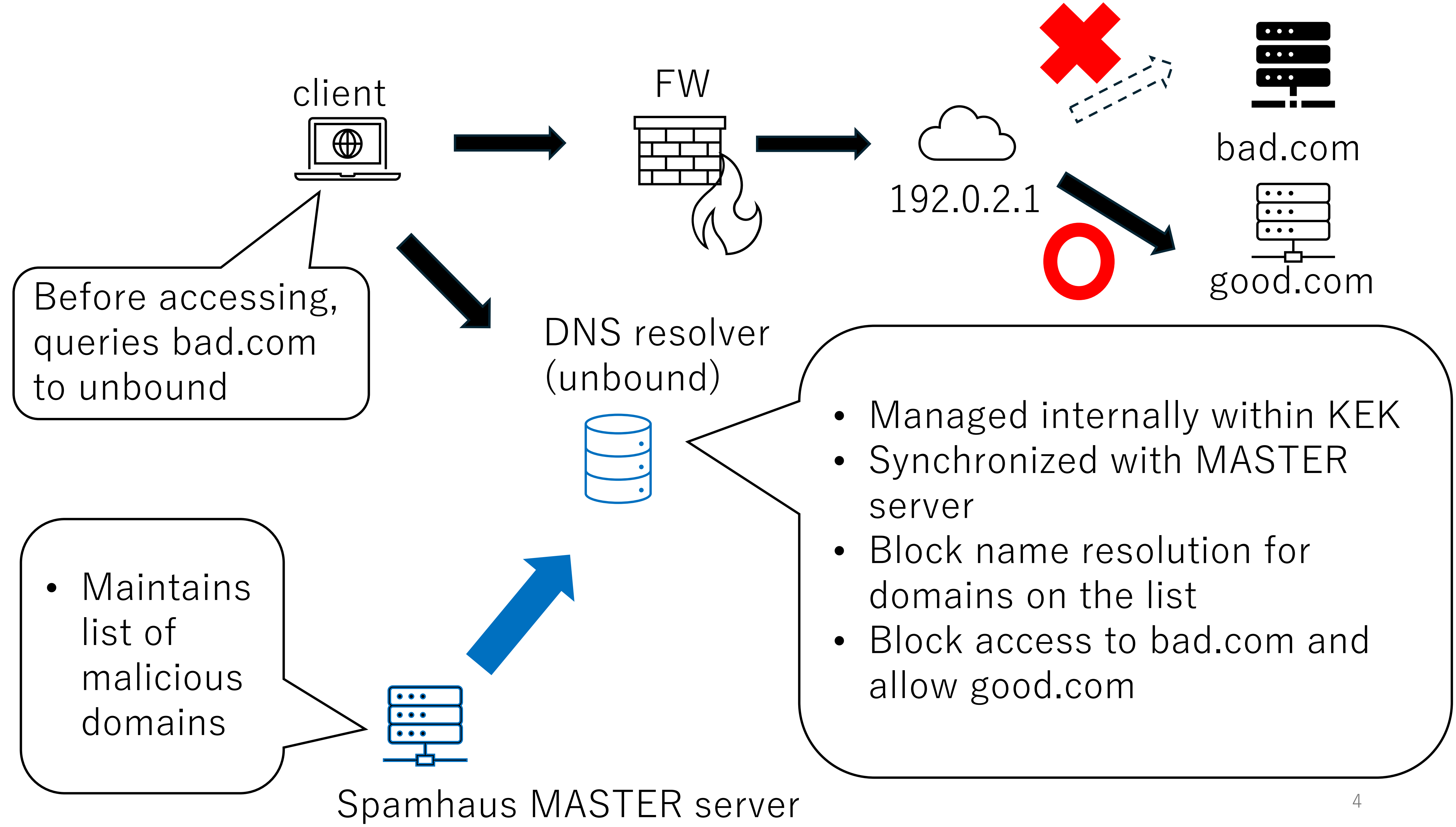
Topic 1 : Why DNS Firewall ?

- Limitation of controlling access using firewall



To block access to bad.com,
block access to 192.0.2.1 at FW

I can‘t access to good.com



client

FW

192.0.2.1

bad.com

good.com

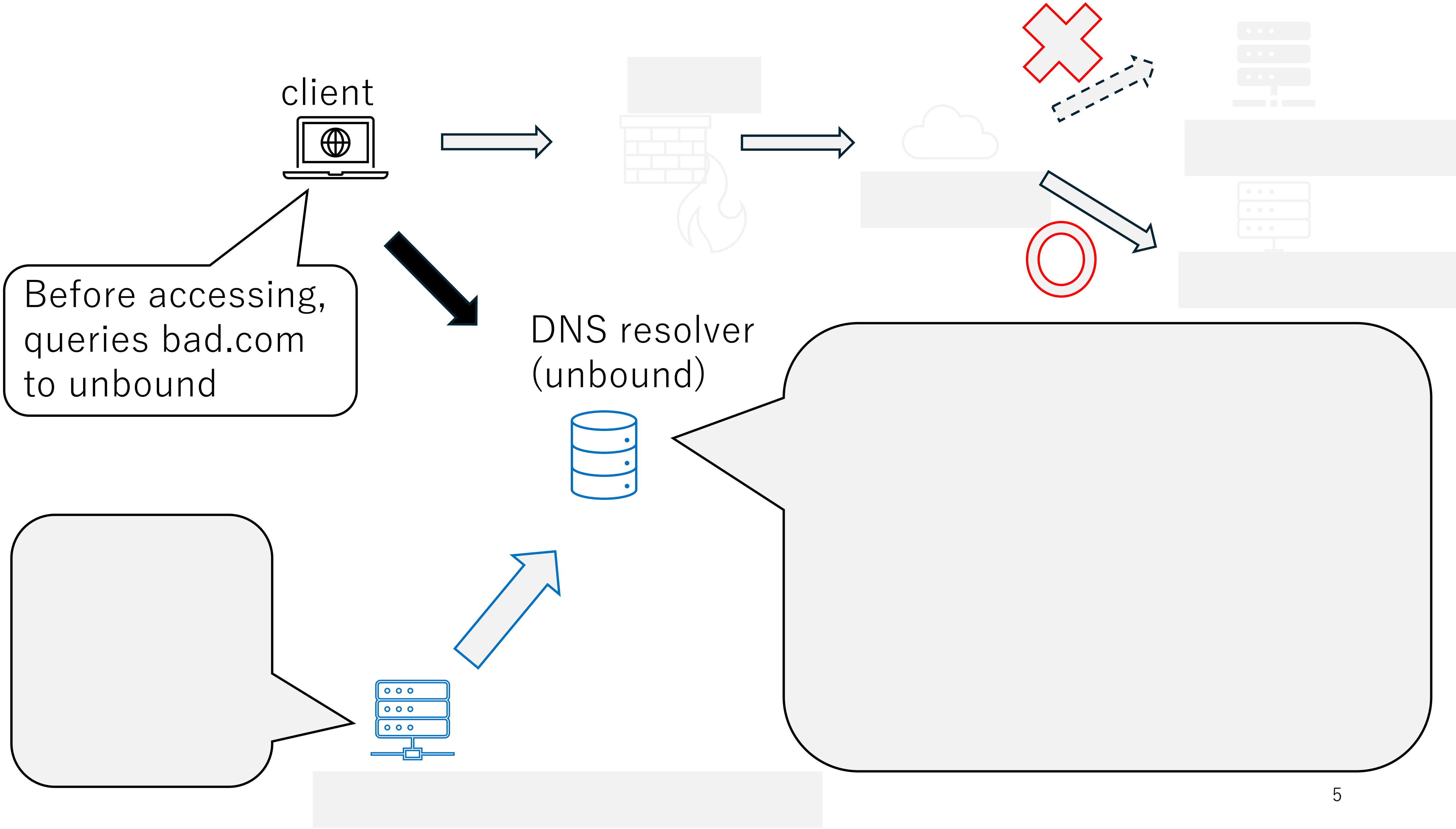
Before accessing, queries bad.com to unbound

DNS resolver (unbound)

- Managed internally within KEK
- Synchronized with MASTER server
- Block name resolution for domains on the list
- Block access to bad.com and allow good.com

- Maintains list of malicious domains

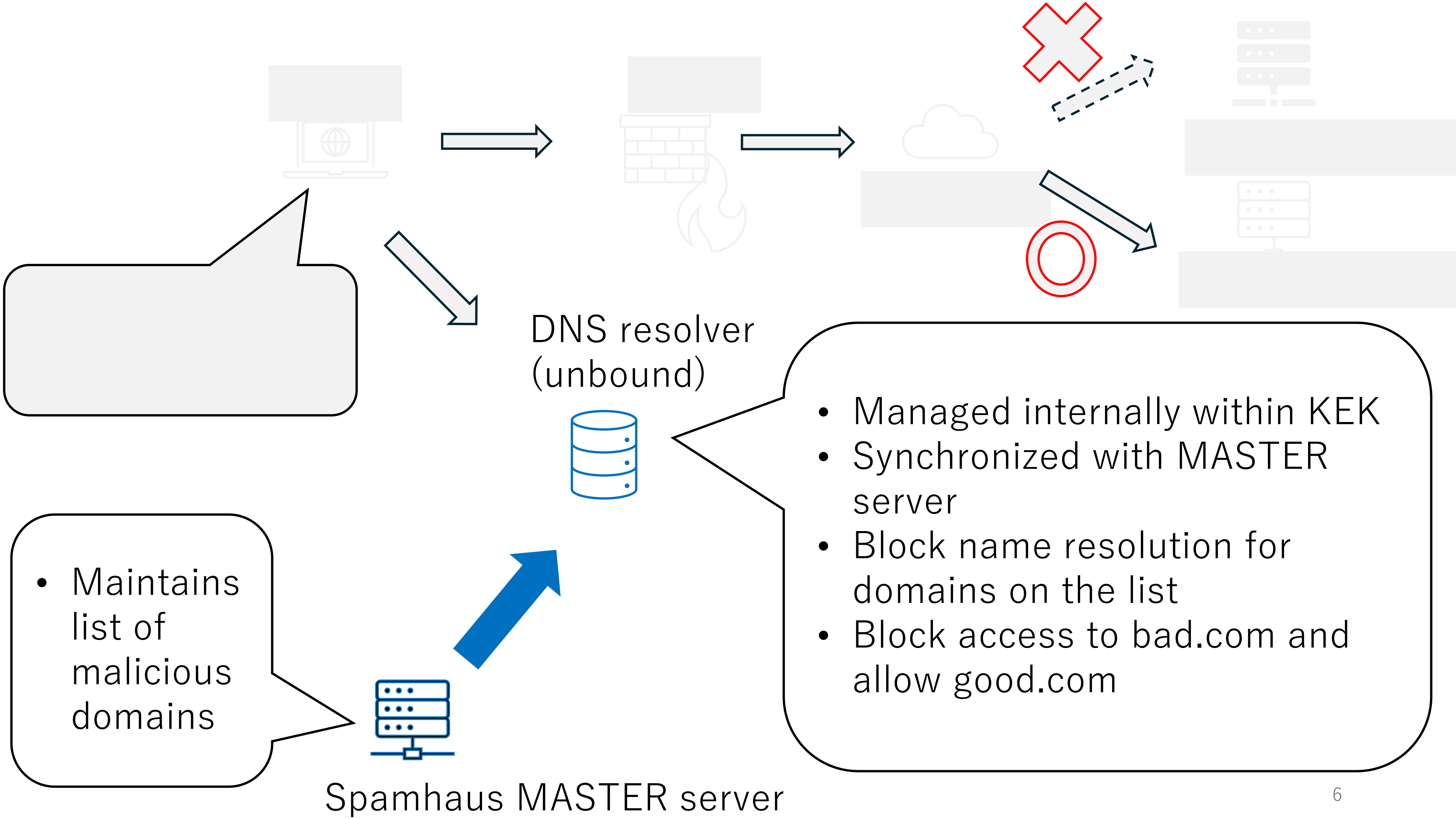
Spamhaus MASTER server



client

Before accessing, queries bad.com to unbound

DNS resolver (unbound)

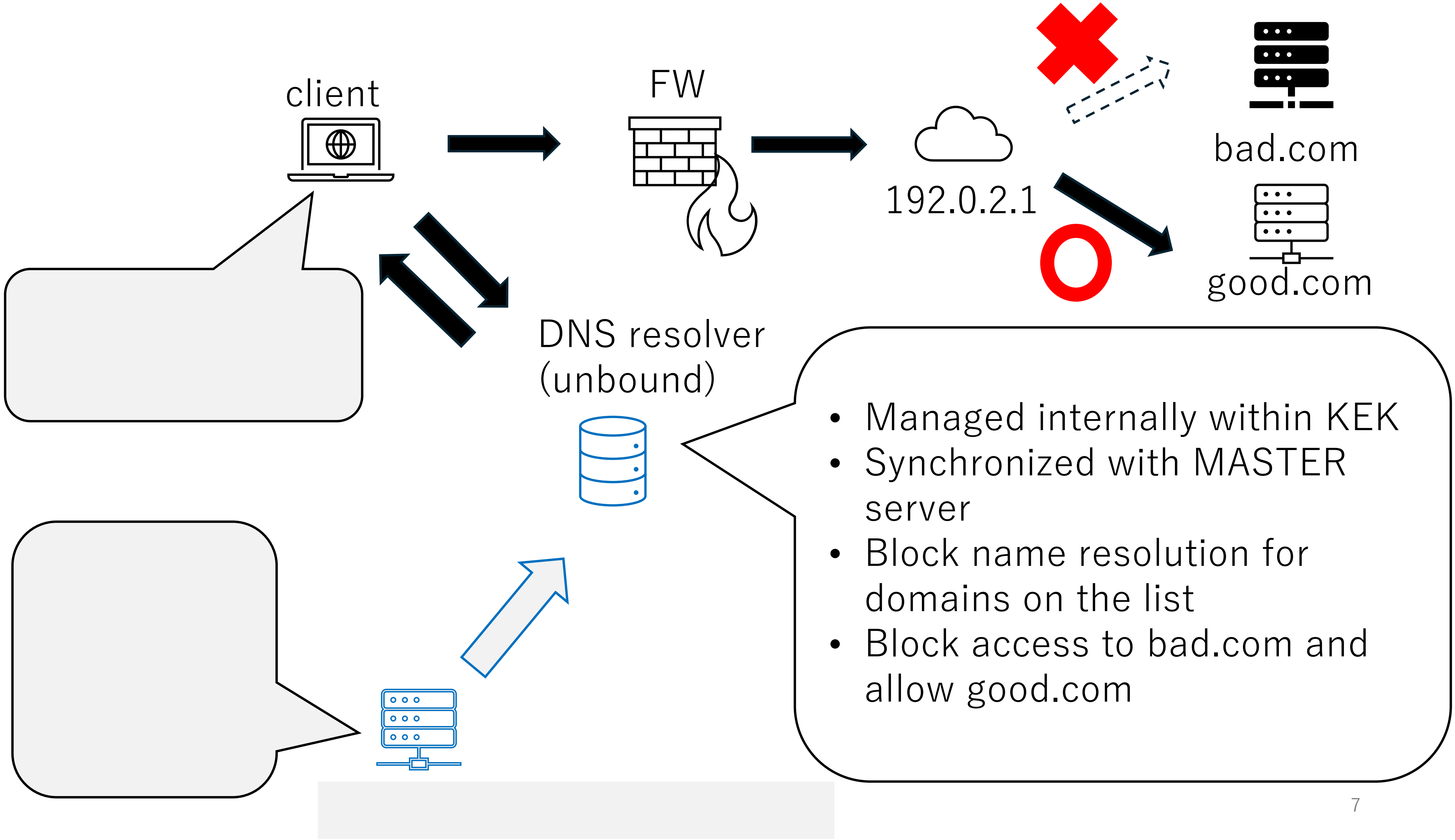


DNS resolver (unbound)

- Managed internally within KEK
- Synchronized with MASTER server
- Block name resolution for domains on the list
- Block access to bad.com and allow good.com

- Maintains list of malicious domains

Spamhaus MASTER server



- Managed internally within KEK
- Synchronized with MASTER server
- Block name resolution for domains on the list
- Block access to bad.com and allow good.com

List of Malicious Domains

- The list consists of 11 categories.
  - Botnet C&C
  - Adware
  - Malware
  - Phishing
  - Crypto mining
  - and others

- Zone files are created for each category.
- The largest file is approximately **270 megabytes** in size.
- The list is updated periodically (exact frequency not yet measured).

Behavior When Blocked

- When a listed domain is queried, the unbound basically returns **NXDOMAIN**.
  - NXDOMAIN means that the domain does not exist.

- **Issue:**
  - When a user accesses a blocked domain via a browser, it is difficult for the user to recognize that the access was filtered by DNSFW.
  - As a result, the user may report it as a general network problem.
- **Countermeasure:**
  - Redirecting users to a notification page indicating that the access was blocked.
  - For HTTPS connections, common browsers display security warnings.

- **We do not have perfect solution, NXDOMAIN is better option**

Current Status on DNS Firewall Deployment

- The technical preparation is complete
  - Run Unbound with DNS Firewall enabled
  - Confirm that queries for FQDNs on the list are actually blocked.

- Start by covering the Computing Research Center
  - Configuring DNS settings centrally via DHCP

- **Items to confirm before operation**
  - Which FQDNs are actually blocked
  - Why those FQDNs are resolved
  - The blocking does not interfere with our work

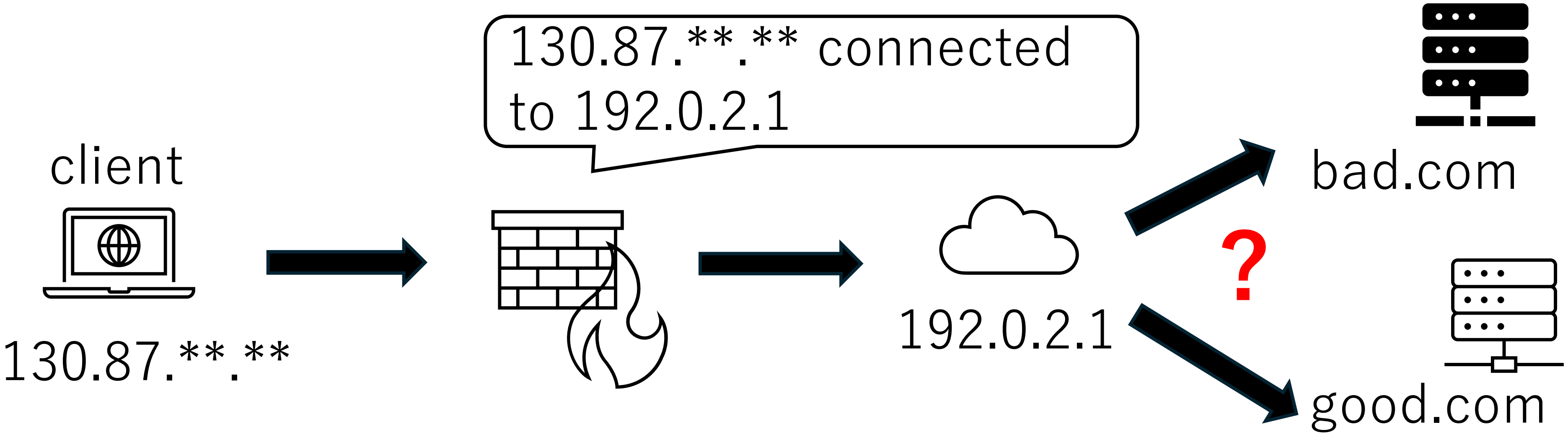
An Example of Blocked Domains

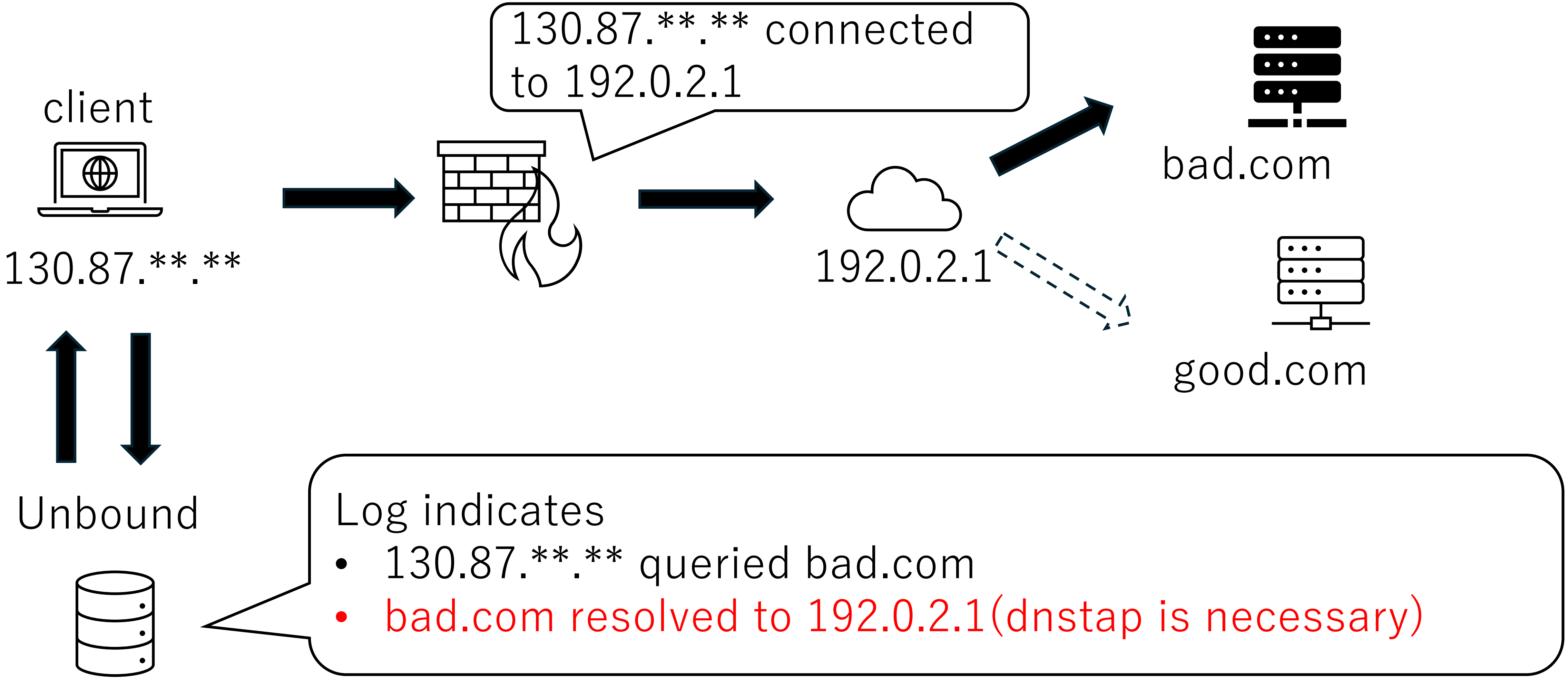
- polyfill.io
  - A service that provides JavaScript code to normalize browser functionality.
  - Since 2024, behavior involving redirection to malicious sites has been observed.
  - Currently, resolver used within KEK and public resolvers (1.1.1.1 and 8.8.8.8) return NXDOMAIN
  - I analyzed internal DNS resolution logs and identified polyfill scripts included in one of our websites.

  - **This domain can be blocked without any issues**

Topic 2 : Logging DNS Response IPs

- **Motivation**
  - To Identify the domains a client connected to
- **Challenge**
  - An IP (e.g., 192.0.2.1) hosts both of bad.com and good.com
  - FW session log “only” indicates the client connected to 192.0.2.1

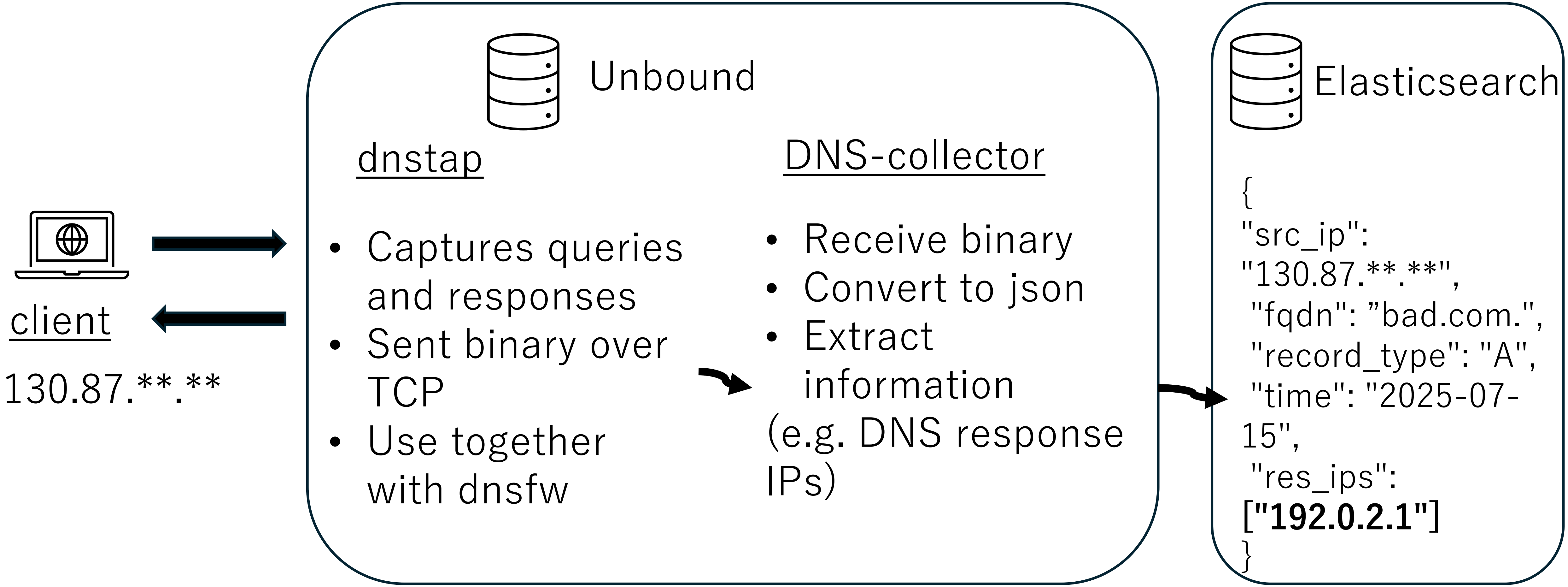




130.87.**.** connected to 192.0.2.1

Log indicates

- 130.87.**.** queried bad.com
- bad.com resolved to 192.0.2.1(dnstap is necessary)



dnstap

- Captures queries and responses
- Sent binary over TCP
- Use together with dnsfw

DNS-collector

- Receive binary
- Convert to json
- Extract information

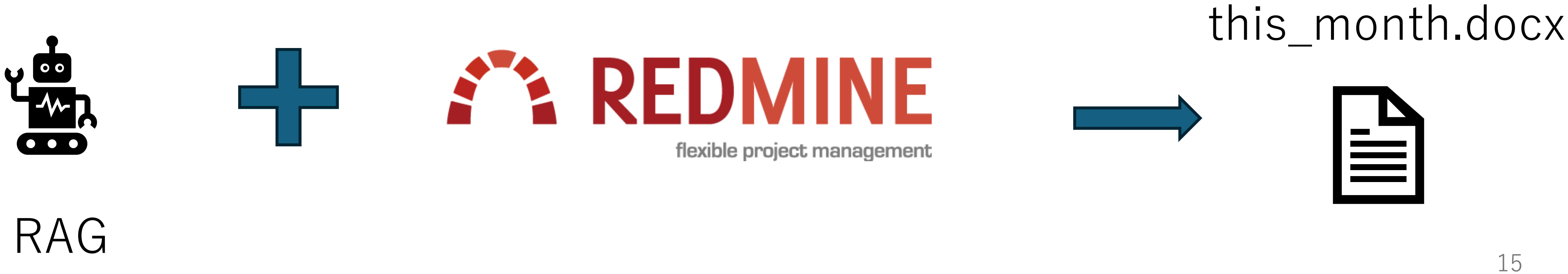
(e.g. DNS response IPs)

```
{
"src_ip":
"130.87.**.**",
 "fqdn": ”bad.com.",
 "record_type": "A",
 "time": "2025-07-
15",
 "res_ips":
["192.0.2.1"]
}
```

- Data was sent to a test Elasticsearch
- Currently, Testing transferring data to the production Elasticsearch

Topic 3 : Generating CSIRT Monthly Report

- Use Redmine for incident response/consultation management
- I write monthly report by checking Redmine tickets for each incidents
- The report is very standardized, but preparation is time-consuming.
  - Try to Generate monthly report by using LLM
- Information related to CSIRT activities cannot be sent to external services.
  - Use the RAG framework in a local environment.



this_month.docx

RAG

Run the LLM Locally

- Hardware information
  - Asus Ascent GX10 x 2
  - Memory:128 GB
    - CPU:ARM v9.2-A
    - GPU:NVIDIA Blackwell
  - 150 x 150 x 51 mm
  - 180W

  - These two machines can be connected with a dedicated cable to enable distributed processing.

Comparison of Execution Environment

1. Ollama
   - Run inference with simple step
   - Models are typically 4-bit precision

2. vLLM : Libraries needed for distributed processing on two machines
   - We can choose 16-bit precision

- Key Metrics for comparison
  - Model Size : The number of trainable parameters
  - Precision : The numerical bit-width used for model computation

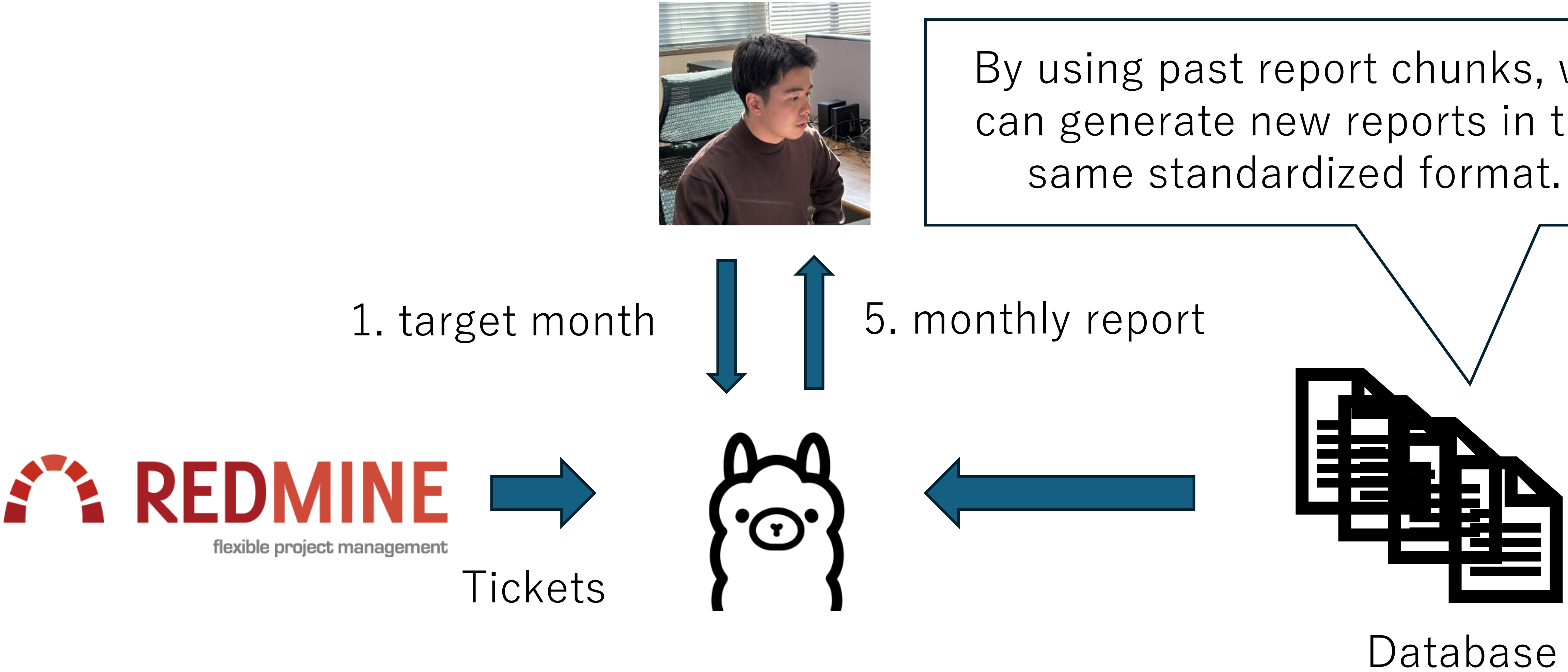
- Experiments on a single machine

  - [Environment 1] Ollama (4bit-precision for all models)
    - gemma3:27b
    - llama3.1:70b
    - gpt-oss:120b
    - Generate acceptable Japanese text quickly.

  - [Environment 2] vLLM
    - google/gemma-3-27b-it(16bit-precision)
      - Takes over 1 minute to produce 500 tokens
      - Reducing output to 200 tokens slightly speeds up

- Running models at a 16-bit precision takes a long time
- 16-bit inference is not necessary for RAG framework
- Ollama is easy to use for 4-bit models, I use Ollama for now

By using past report chunks, we can generate new reports in the same standardized format.



1. target month

5. monthly report

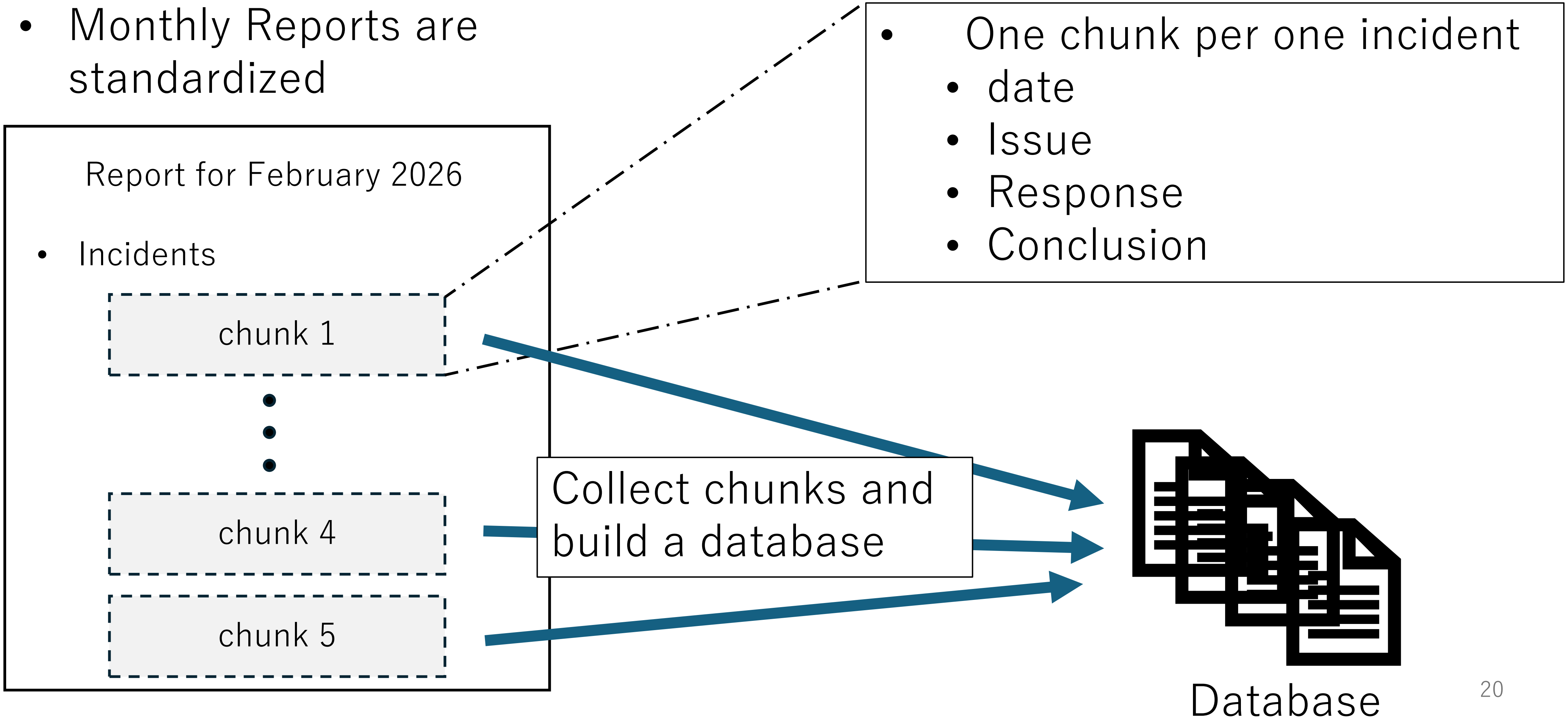
Tickets

Database

2. Retrieve tickets from Redmine API
3. Retrieve “Past Report Chunks” from database
4. Generating monthly report

What Are Past Report Chunks?

- Monthly Reports are standardized



Report for February 2026

- Incidents

chunk 1

chunk 4

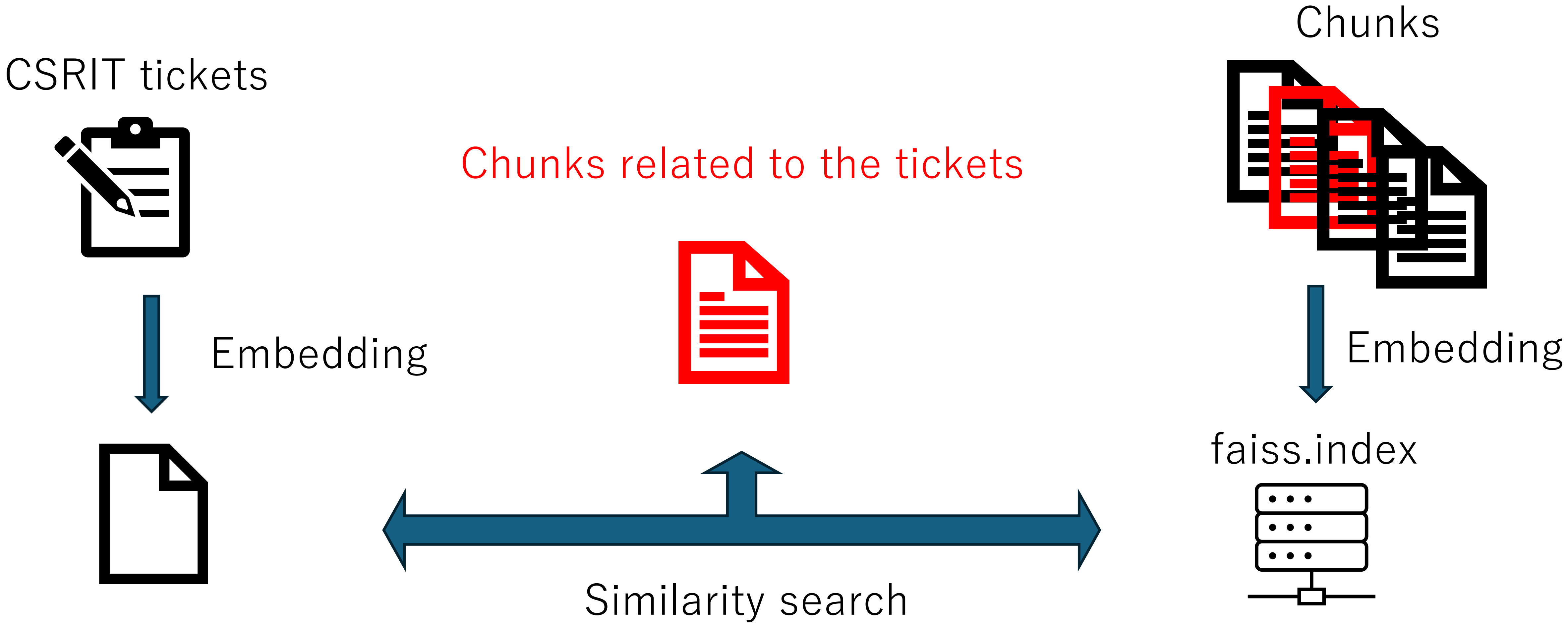
chunk 5

- One chunk per one incident
  - date
  - Issue
  - Response
  - Conclusion

Collect chunks and build a database

Database

How to Retrieve Past Report Chunks



CSRIT tickets

Embedding

Chunks related to the tickets

Chunks

Embedding

faiss.index

Similarity search

- Embedding is a way to represent text as a vector
- Finds items with similar meaning

Future Work: Report Generation

- Retrieved data (tickets and past report chunks) will be used as input for the LLM
- Prompt:
  - Based on the information in {**ticket**}, generate a report by referring to {**chunks**}.
- Generated reports will be reviewed to verify accuracy and consistency

Summary

- Today’s Topics
  1. DNS Firewall: How It Works and Deployment Challenges
  2. System for Collecting Logs: Purpose and Design
  3. Incident Report Generation: Current Status

- This talk was an overview of three ongoing projects that I’m involved in.
- We will continue to improve them step by step