

# Semantic router using data stream to enrich services

Authors: Koichi Inoue, Dai Akashi, Michihiro Koibuchi, Hideyuki Kawashima and Hiroaki Nishi



Keio University, Japan  
Koichi Inoue

1



## Two different things ...

### Evolutions on Service/Application layer

- Server-based evolution is going on
  - Search engines and Web services are provided by End-hosts, and users can enjoy it on their web browsers
  - Data from many sources becomes contents and enriches web services
  - Mashup is enabled by providing API, which enhances data interaction

### Evolutions on Network/Infrastructure layer

- Routers and Switches take a responsibility for packet transmission, converting protocol, firewall, and load balancing
- Eventually, it is evaluated at throughput and traffic control

This gap causes,

- P2P overlay network, Net neutrality issue between service providers and carriers, ...



# To fill the gap

## Related work

- Cisco ISR (Integrated Service Router) AXP (Application eXtension Platform)
  - Extension module for edge routers
  - System integrators and user companies can develop application with the provided API
  - The target applications are security functions and integration with mission critical system

The goal is to create a platform which enhances third parties' network application development.

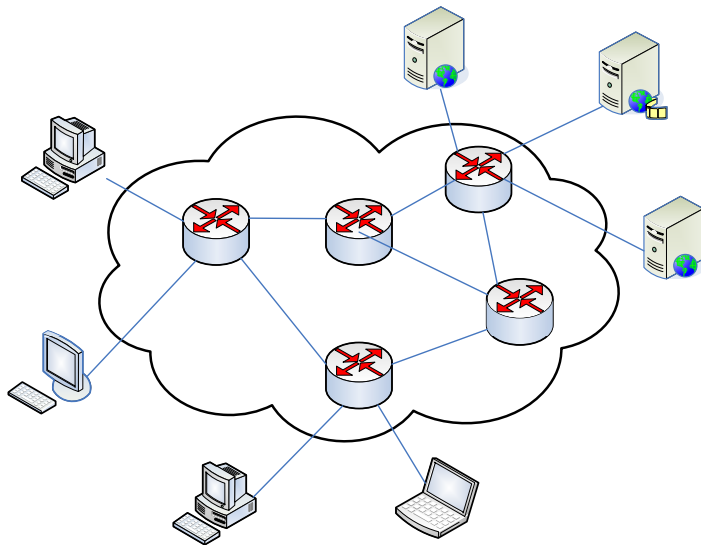
- Content base routing
  - Proposed by Moscola, Cho and Lockwood
  - XML router inspects XML messages and route XML packets based on the pre-defined table
  - Reconfigurable hardware inspect XML packets at multi-gigabit per seconds

Its routing table has to be manually configured. It is XML protocol friendly, but is not service-friendly yet.

Why don't you take advantage of data?



# Collecting data in the router

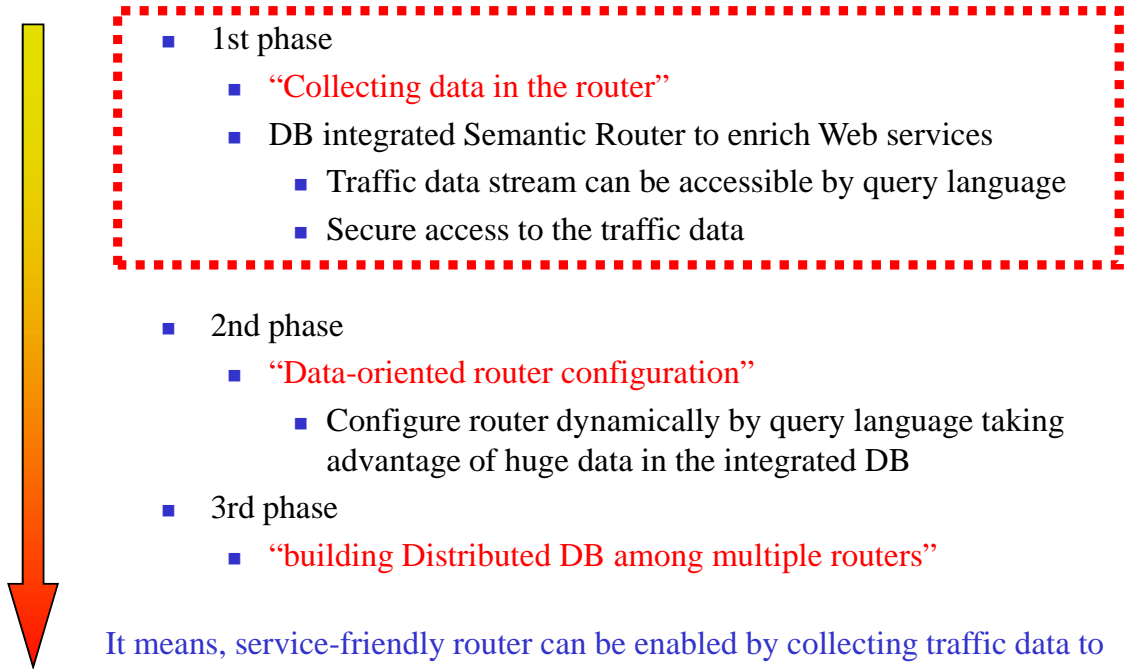


- Routers in the Semantic Router research includes,
- Core routers
  - Metro routers
  - Edge routers

End-hosts	Switches/Routers
Located at network edge	Located at network branch
Data actively collectable	Data passively collectable



# Proposed Semantic Router



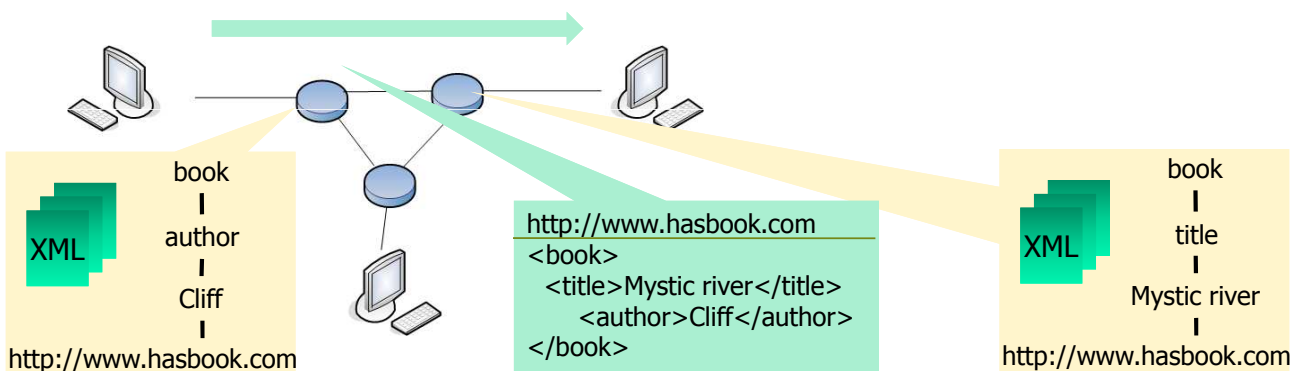
- 1st phase
  - “Collecting data in the router”
  - DB integrated Semantic Router to enrich Web services
    - Traffic data stream can be accessible by query language
    - Secure access to the traffic data
- 2nd phase
  - “Data-oriented router configuration”
    - Configure router dynamically by query language taking advantage of huge data in the integrated DB
- 3rd phase
  - “building Distributed DB among multiple routers”

It means, service-friendly router can be enabled by collecting traffic data to start interacting with Web services. Then, collected data will help to determine routing table, and its configuration can be made by SQL-like query language.



# Concept of Semantic Router Ph1

1. Snoop traffic data stream
2. Store traffic data stream at the integrated DB
  - SSRQL (Semantic Switch Router Query Language)





# Key challenges

---

- Massive data processing
  - Define Query Language SSRQL to access traffic data stream (packet, content)
  - Task allocation between hardware and software to enable line speed processing
- Access Control
  - Manage accessible information for each client
    - LAC (Layer based Access Control)
  - AHP (Authority Hand-over Protection)
    - AHP-sa (semantic analysis): use semantic analysis to protect the leakage
    - AHP-pf (post fetch): “double-check” the contents to identify the privacy level



# Query language SSRQL

---

- SSRQL (Semantic Switch Router Query Language) definition
  - Default SSRQL ⇒ Statement issued by router administrator. Besides data collection/insertion, define privacy protection policy, and take care of hardware resource management.
  - User SSRQL ⇒ Statement for application programmers to collect specific data that may enrich their services or research activities. Query is restricted due to privacy protection policy defined by administrator.
  - Compiled SSRQL ⇒ Statement for hardware processing. Generated and translated from Default SSRQL and User SSRQL.
  - Active SSRQL ⇒ Statement exchanged among routers to maintain distributed database and to define task allocation.



## Example of SSRQL

- Store HTML file which includes “foo”

```
INSERT TABLE store_data WHERE src port = 80 AND  
REGEXP (<html>.*foo.*</html>) limit 1GB
```

- Get access time list of users who visited foo.com

```
SELECT access_time FROM store_data WHERE  
REGEXP "GET foo.com"
```

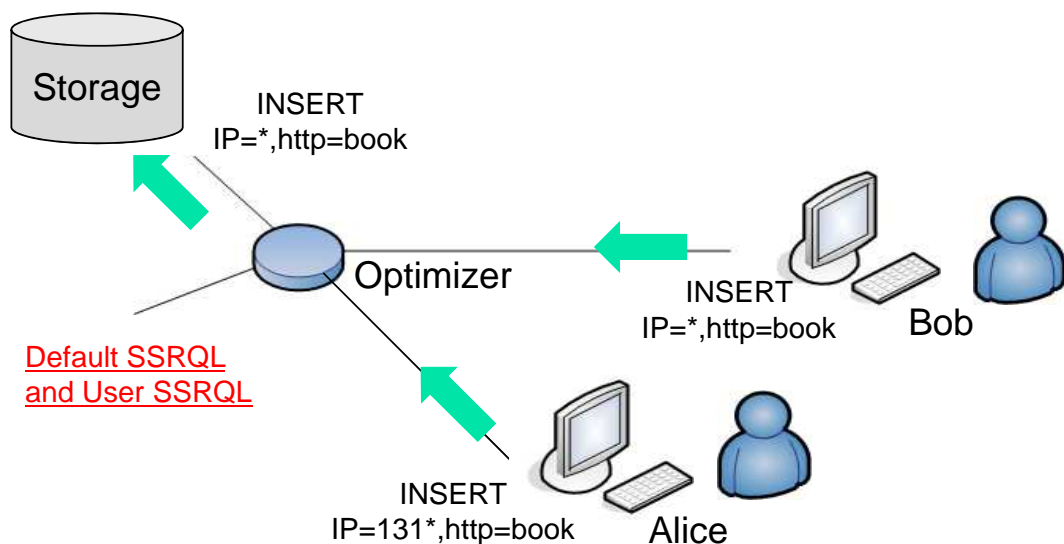
- Change routing table

```
INSERT TABLE routing_table values (src, dst, next_hop)
```



## Multiple Request Optimization

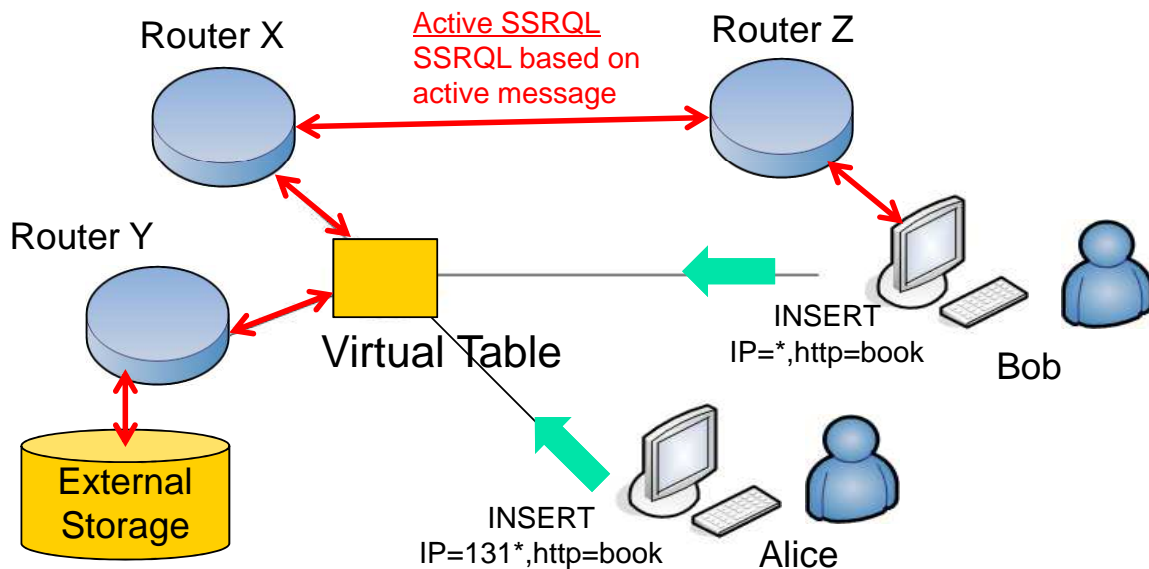
- Eliminate duplicated queries





# Distributed Database

## Virtual Table and Active SSRQL



11

Koichi Inoue / Keio Univ.



## Key challenges

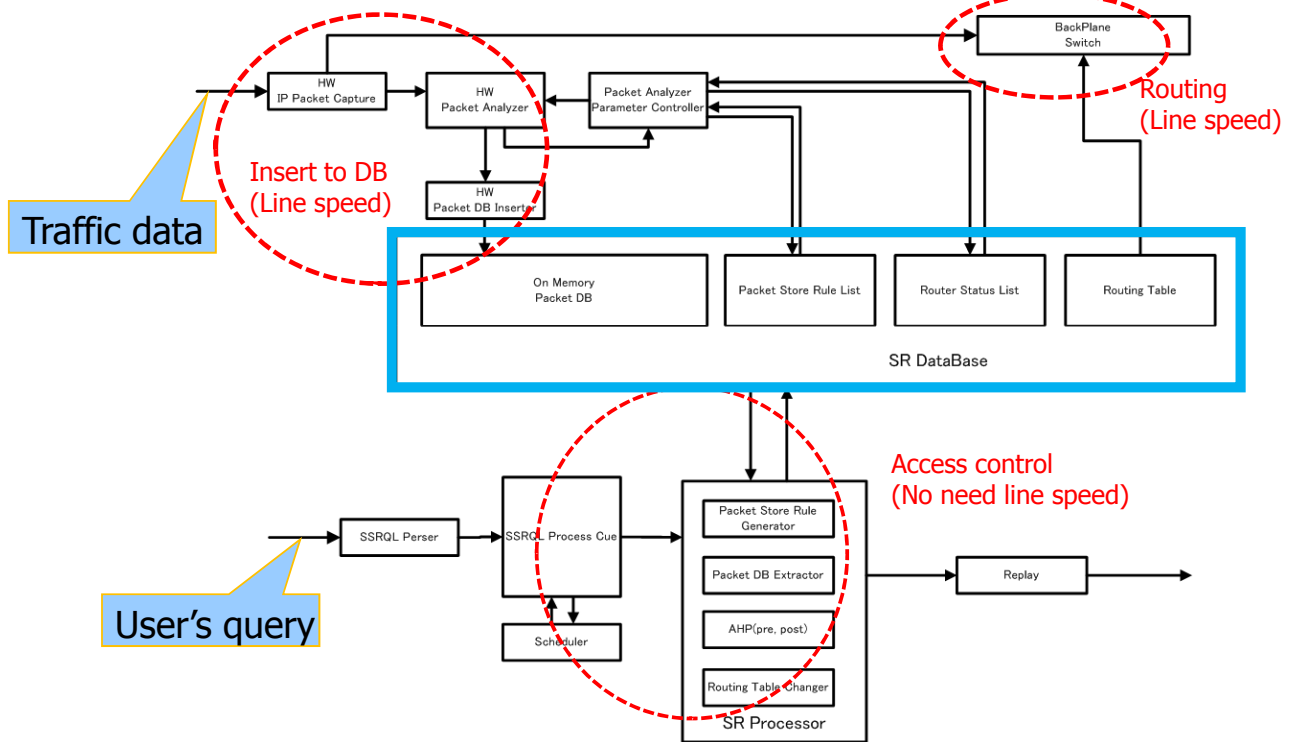
- Massive data processing
  - Define Query Language SSRQL to access traffic data stream (packet, content)
  - Task allocation between hardware and software to enable line speed processing
- Access Control
  - Manage accessible information for each client
    - LAC (Layer based Access Control)
  - AHP (Authority Hand-over Protection)
    - AHP-sa (semantic analysis): use semantic analysis to protect the leakage
    - AHP-pf (post fetch): “double-check” the contents to identify the privacy level

12

Koichi Inoue / Keio Univ.



# Task allocation between HW / SW



13

Koichi Inoue / Keio Univ.



# Key challenges

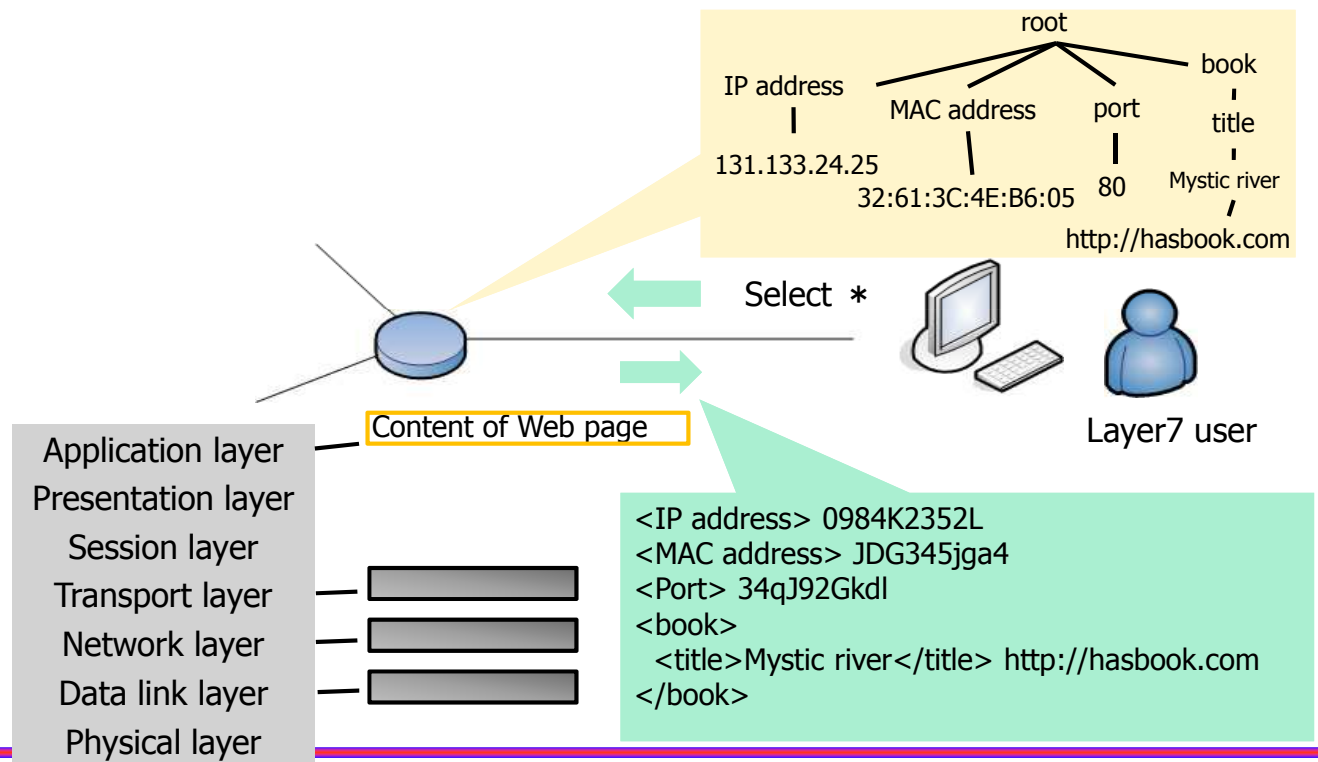
- **Massive data processing**
  - Define Query Language SSRQL to access traffic data stream (packet, content)
  - Task allocation between hardware and software to enable line speed processing
- **Access Control**
  - Manage accessible information for each client
    - LAC (Layer based Access Control)
  - AHP (Authority Hand-over Protection)
    - AHP-sa (semantic analysis): use semantic analysis to protect the leakage
    - AHP-pf (post fetch): "double-check" the contents to identify the privacy level

14

Koichi Inoue / Keio Univ.



# LAC (Layer based Access Control)

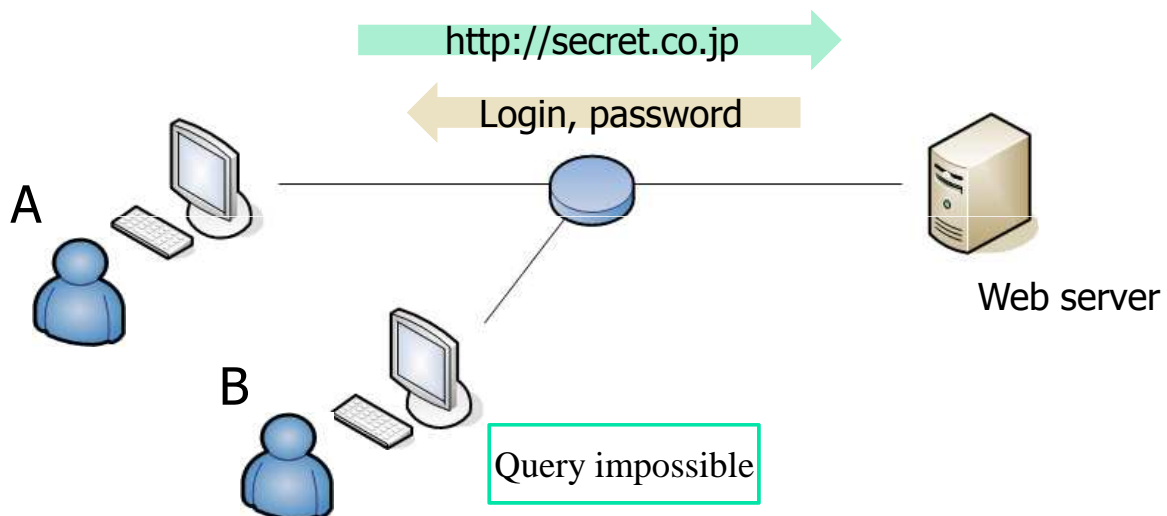


15

Koichi Inoue / Keio Univ.



# AHP-sa (semantic analysis)



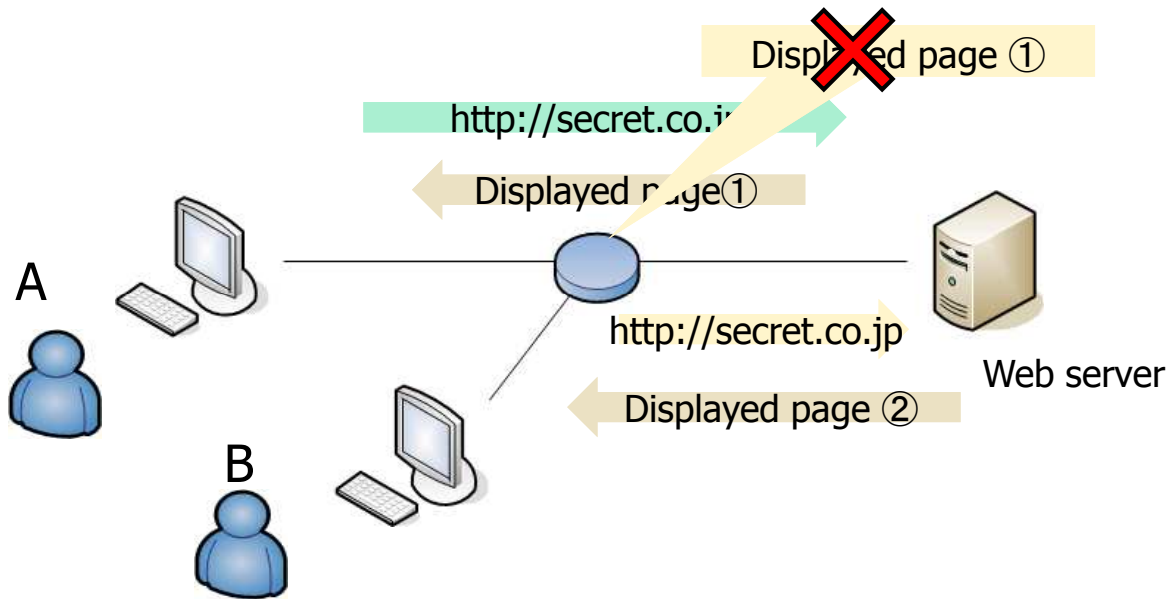
- If there is privacy information, data from user A won't be stored.

16

Koichi Inoue / Keio Univ.



## AHP-pf (post fetch)



- If the double-checked contents are different from the originally received, data won't be stored.



## Application

- Users' behavioral history
- Real-time Real-popularity search
- Long tail contents search
- Creating API for non-API contents
- DDoS attack detect and blocker
- Layer 7 firewall

All up to your query.



## Conclusion

---

- Future Internet must be more service-friendly based on the data-oriented architecture
- Semantic Router is proposed
  - Semantic Router enables to use traffic data stream to enrich web services
  - Semantic Router has associated in-memory database
  - Semantic Router provides traffic data to service providers with extended SQL-based query language and strict access control to maintain privacy.
- Eventually, Routers won't be a cost for telecom carriers any longer, but Semantic Routers will generate additional revenue for them. Network Neutrality issue could be resolved, and telecom carriers and service providers can build win-win relationship finally.



## Thank you

---

For more information,  
Koichi Inoue  
Kinoue@west.sd.keio.ac.jp