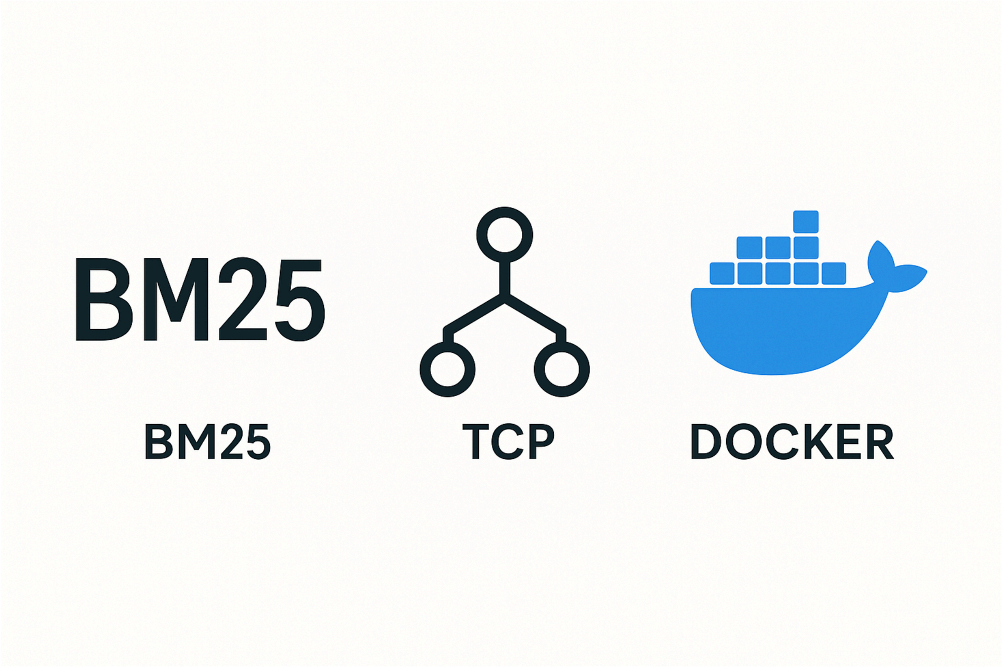
Retrivium

BM25 over the network

Authors: Liao Pei-Wen, Makovskyi Maksym, Wu Guo Yu

What problem we are trying to solve ?



Our solution to the problem (1)

- Real-time document indexing
- Multi-user access via TCP
- Relevance-based search (`QUERY`)
- Document browsing (`SHOW`)
- Dynamic file uploads (`UPLOAD`)

How to use ? (1)

1. **Pull the Image**

```
docker pull ghcr.io/maxmakovskiy/retrivium:main
```

How to use ? (2)

2. **Create the network**

```
docker network create dai-retrivium
```

How to use ? (3)

3. **Prepare data**

   Create a directory with text files that will be indexed and searchable.

   For example:

```
$ mkdir data && cd data
$ mkdir docs
$ echo "a cat is a feline and likes to eat bird" > docs/file1.txt
$ echo "a dog is the human's best friend and likes to play" > docs/file2.txt
$ echo "a bird is a beautiful animal that can fly" > docs/file3.txt
$ mkdir uploads
$ echo "April always begins on the same day of the week as July" > uploads/file10.txt
```

How to use ? (4)

4. **Start server**

```
$ docker run --rm -it --network dai-retrivium \
$ -v $(pwd)/documents:/app/documents --name retrivium-server \
$ ghcr.io/maxmakovskiy/retrivium:main server --port 6433 -D documents
```

How to use ? (5)

5. **Start client**

```
$ docker run --rm -it --network dai-retrivium \
$ -v $(pwd)/to_upload:/app/uploads \
$ ghcr.io/maxmakovskiy/retrivium:main client --port 6433 --host retrivium-server
```

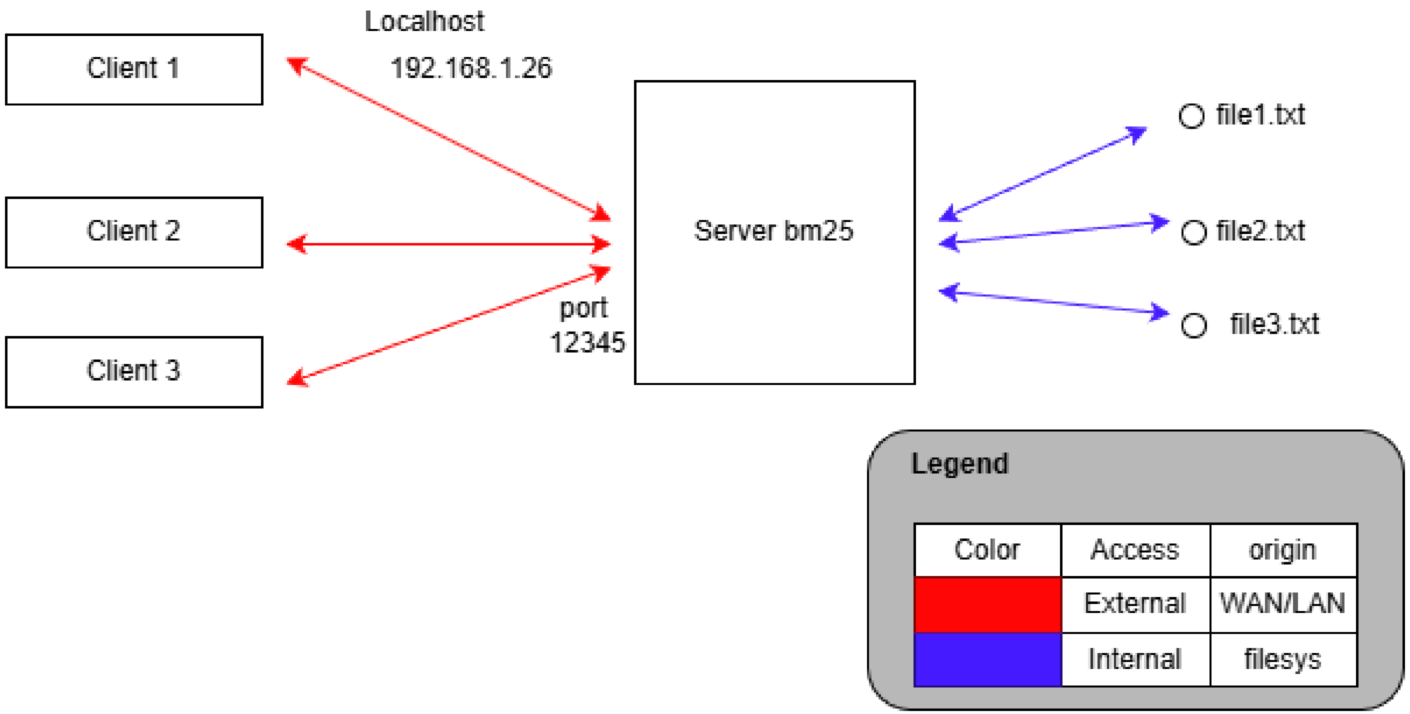
How to use ? (6)

6. **Test commands**

   Try searching and listing documents in the client prompt.

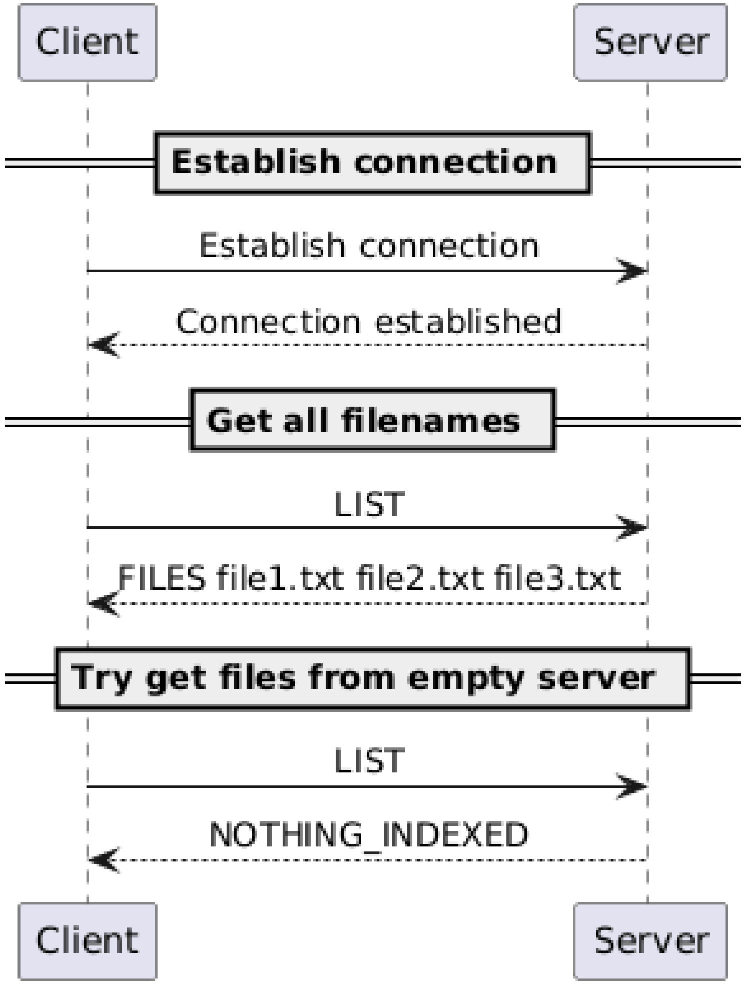
```
# Type in the client terminal
> LIST
> QUIT
```

How it works ?

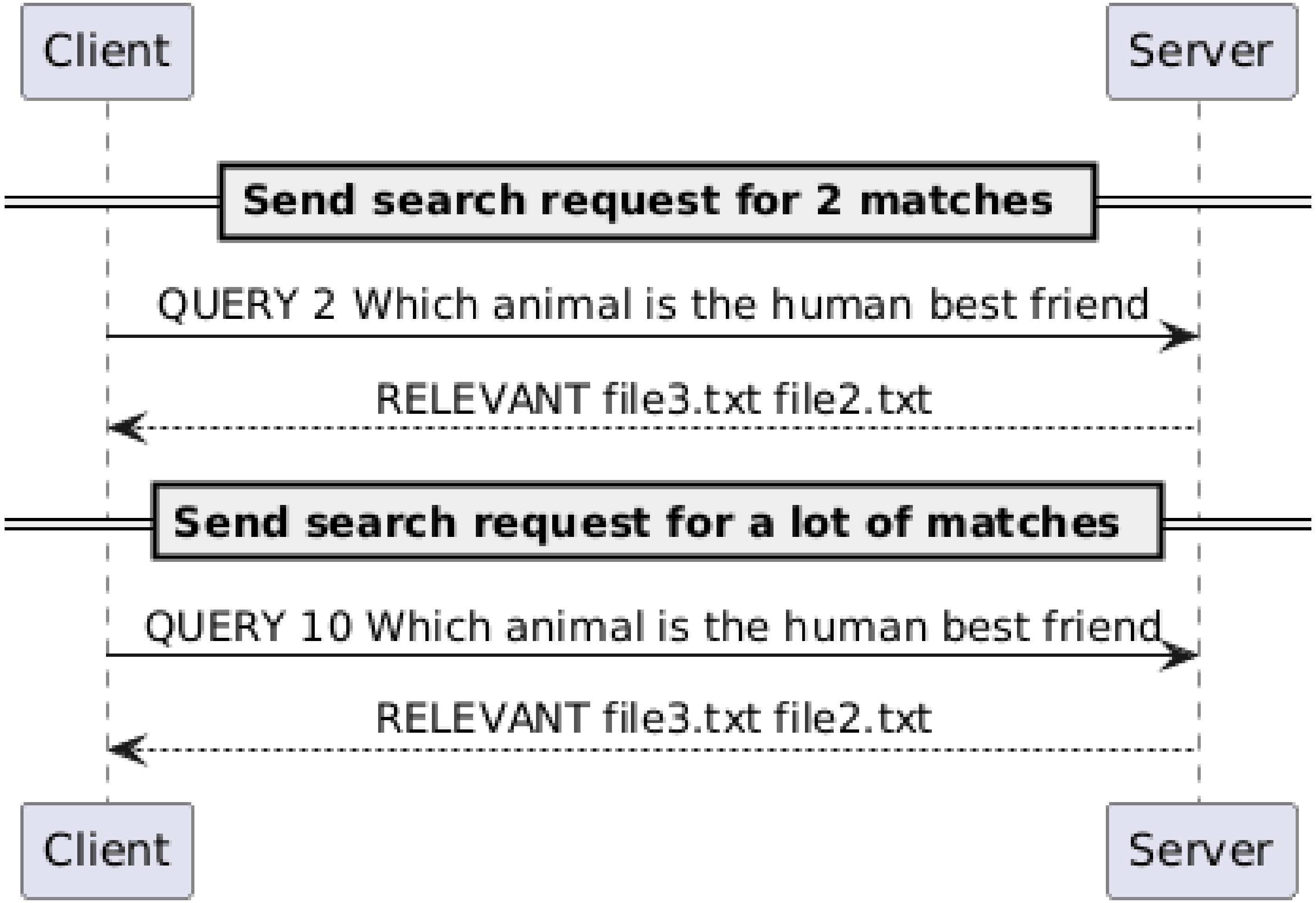


Color	Access	origin
🟥	External	WAN/LAN
🟦	Internal	filesys

Example (1) : connect and list

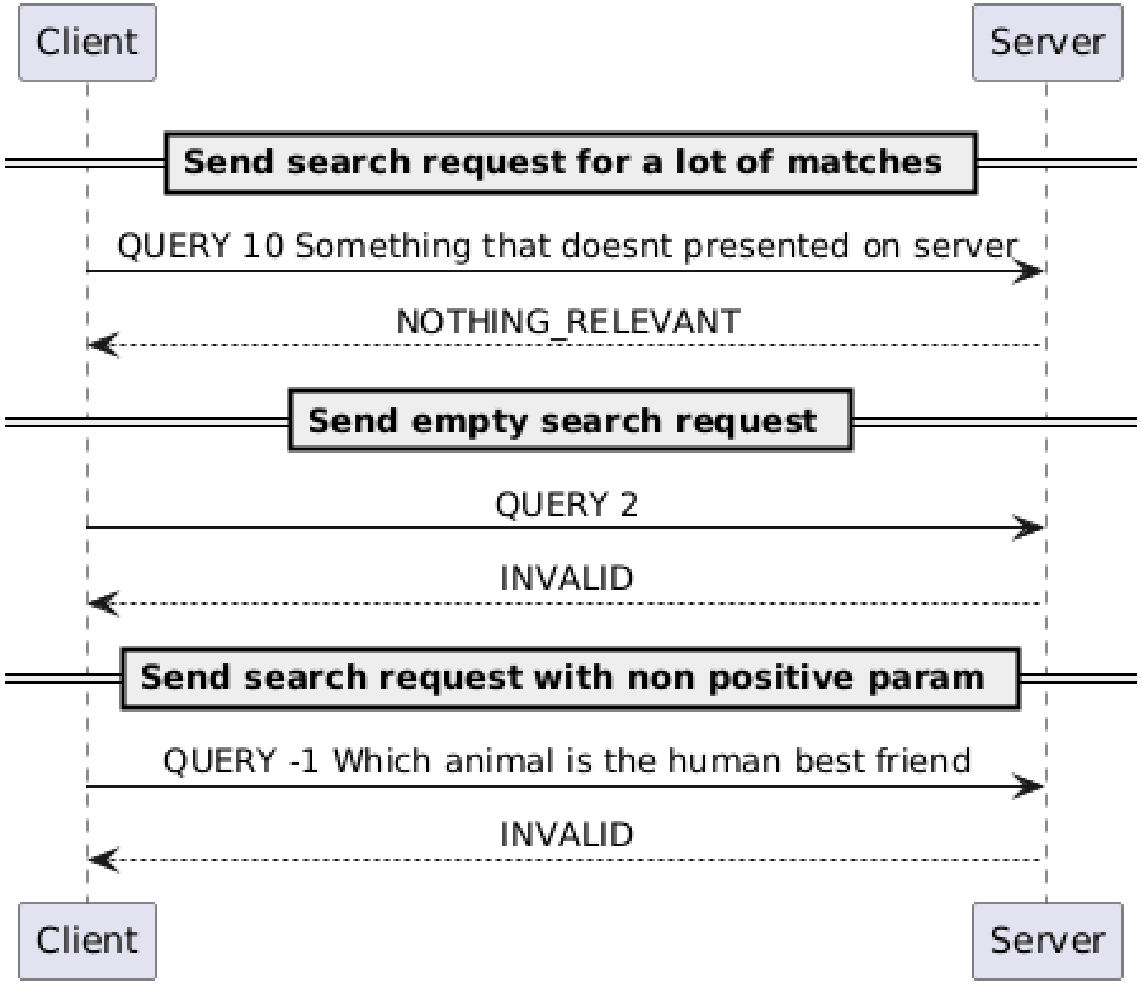


Example (2) : query success



Client
Server
Send search request for 2 matches
QUERY 2 Which animal is the human best friend
RELEVANT file3.txt file2.txt
Send search request for a lot of matches
QUERY 10 Which animal is the human best friend
RELEVANT file3.txt file2.txt
Client
Server

Example (3) : query fail



Client

Server

Send search request for a lot of matches

QUERY 10 Something that doesnt presented on server

NOTHING_RELEVANT

Send empty search request

QUERY 2

INVALID

Send search request with non positive param

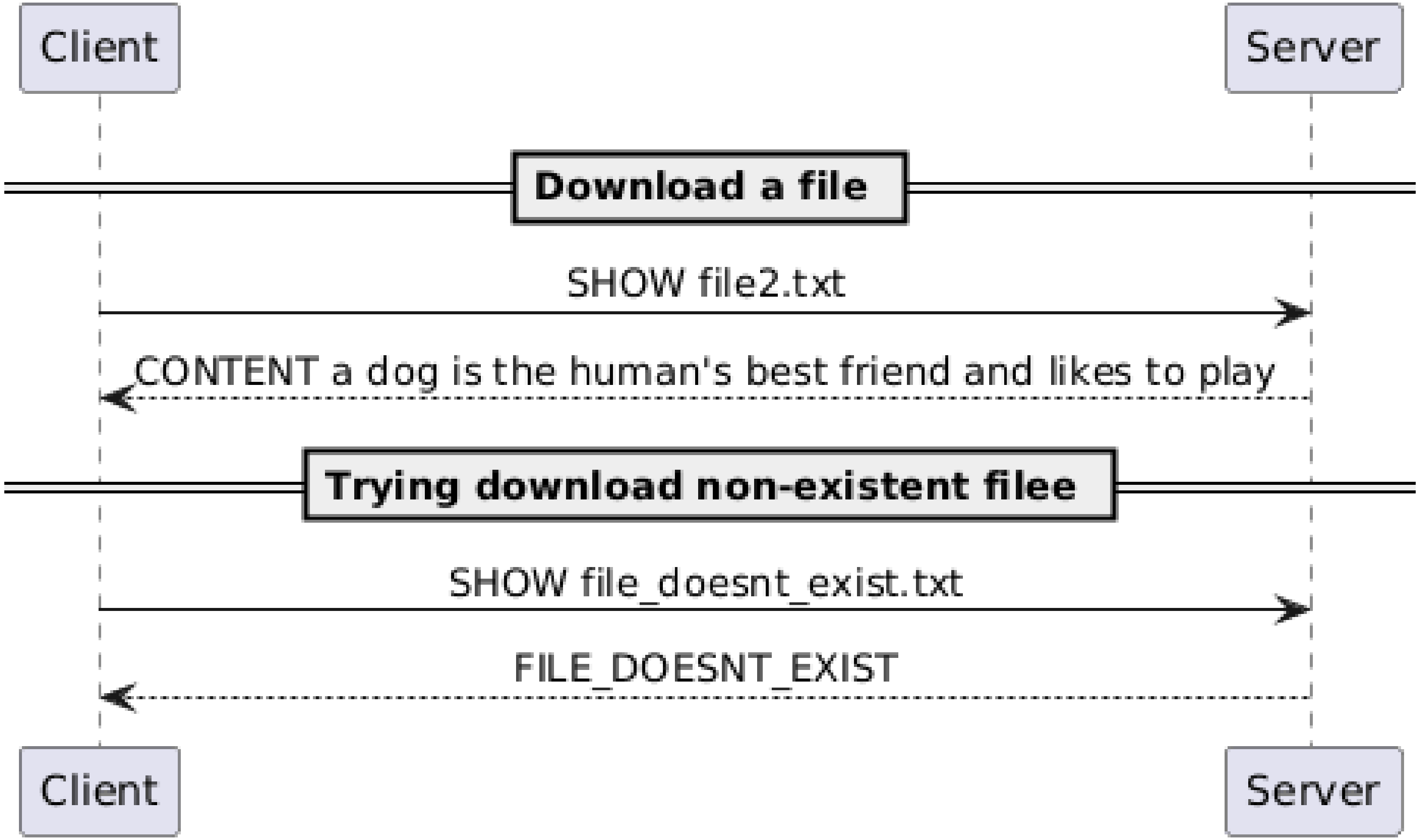
QUERY -1 Which animal is the human best friend

INVALID

Client

Server

Example (4) : download



Client

Server

Download a file

SHOW file2.txt

CONTENT a dog is the human's best friend and likes to play

Trying download non-existent filee

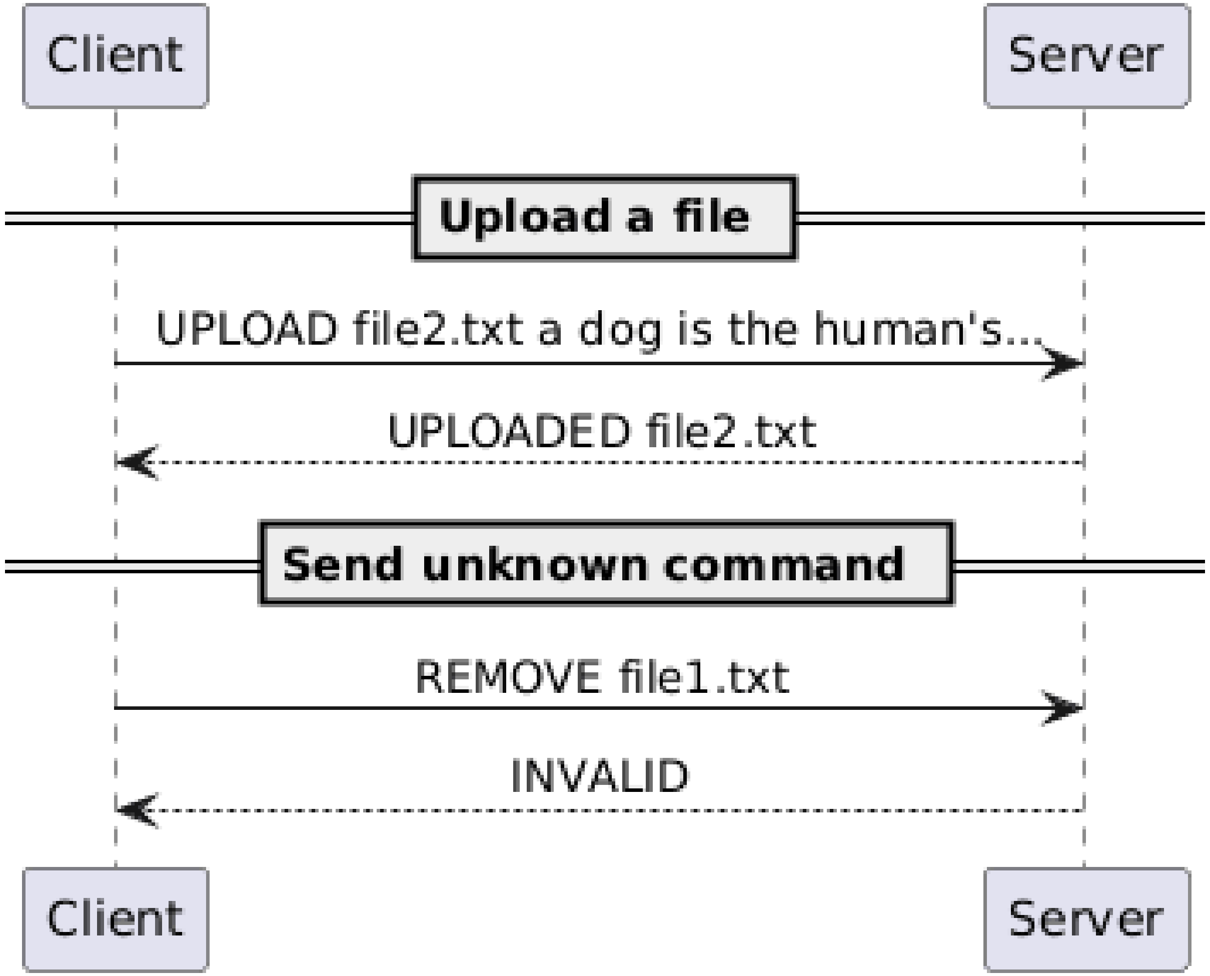
SHOW file_doesnt_exist.txt

FILE_DOESNT_EXIST

Client

Server

Example (5) : download and ill-formed cmd



Use Cases (1)

- Code Snippet Repository

```
> Query 5 binary search tree implementation
```

- Configuration File Finder

```
Find which service has connection timeout set too low
> QUERY 5 timeout connection 30 seconds
> SHOW <suspicious-config.conf>
```

Use Cases (2)

- Log File Analysis

```
Folder contains daily server logs
> QUERY 5 error 404 authentication failed
```

- Legal document search

```
> QUERY 10 intellectual property patent infringement
```

Roadmap - Current Limitations & Planned Improvements (1)

1. **Java locks**

[Limitation]

Race conditions

Roadmap - Current Limitations & Planned Improvements (2)

2. **Authentication & Security**

[Limitation]

- No encryption - data transmitted in plain text (TCP)
- No authentication - anyone can connect
- No access control - all users see all documents
- No audit logs - can't track who searched what

Roadmap - Current Limitations & Planned Improvements (3)

3. **Authentication & Security (continue)**

[Improvement]

- User authentication with username/password or API keys
- TLS/SSL encryption for secure transmission
- Role-based access control

Roadmap - Current Limitations & Planned Improvements (4)

4 **Persistent Index Storage**

[Limitation]

- Index stored only in memory
- Re-indexes ALL documents on every server restart
- Loses index when server stops
- Slow startup with large document collections

Roadmap - Current Limitations & Planned Improvements (5)

5 **Persistent Index Storage (continue)**

[Improvement]

- Save index to disk (serialization or database)
- Incremental indexing - only index changed files
- Fast startup by loading pre-built index
- Support for very large datasets

Thank you for your attention !