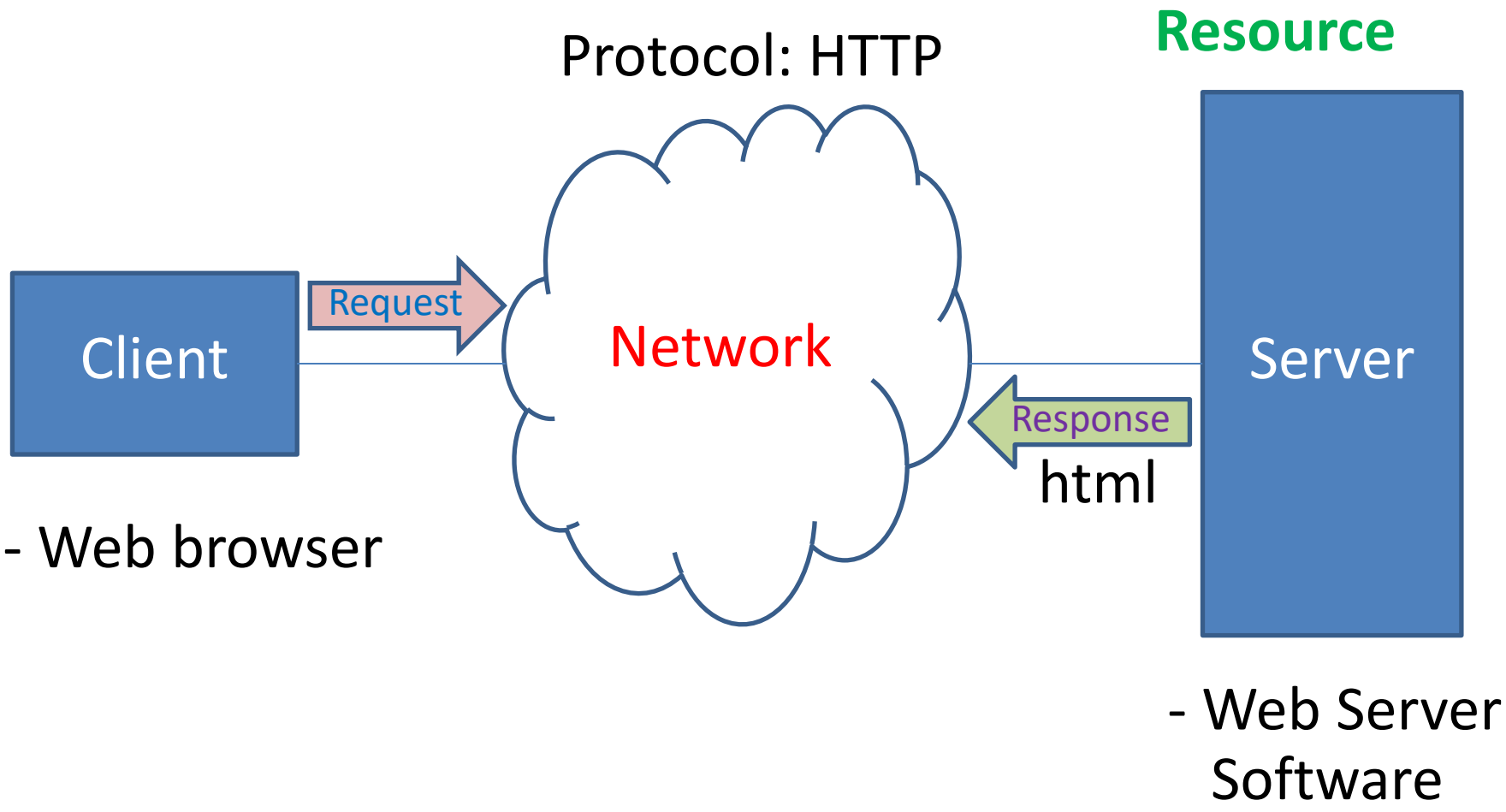
Introduction

2566

Web Concept


Protocol: HTTP
Resource
Client
Request
Network
Server
Response
html
- Web browser
- Web Server Software

Hypertext Transfer Protocol

- Request – Reply protocol (RR)
- HTTP Resources are identify by URI (or URL)
- 2 types of message
  - Request message

  - Response message

Example of HTTP Exchange

- Request message:

```
GET /index.html HTTP/1.1
Host: www.example.com
[Blank Line]
```

Request line

header line

- Response message:

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 1999 23:59:59 GMT
Content-Type: text/html
Content-Length: 1354
[Blank Line]
<html>
…
```

Status line

Header lines

Body

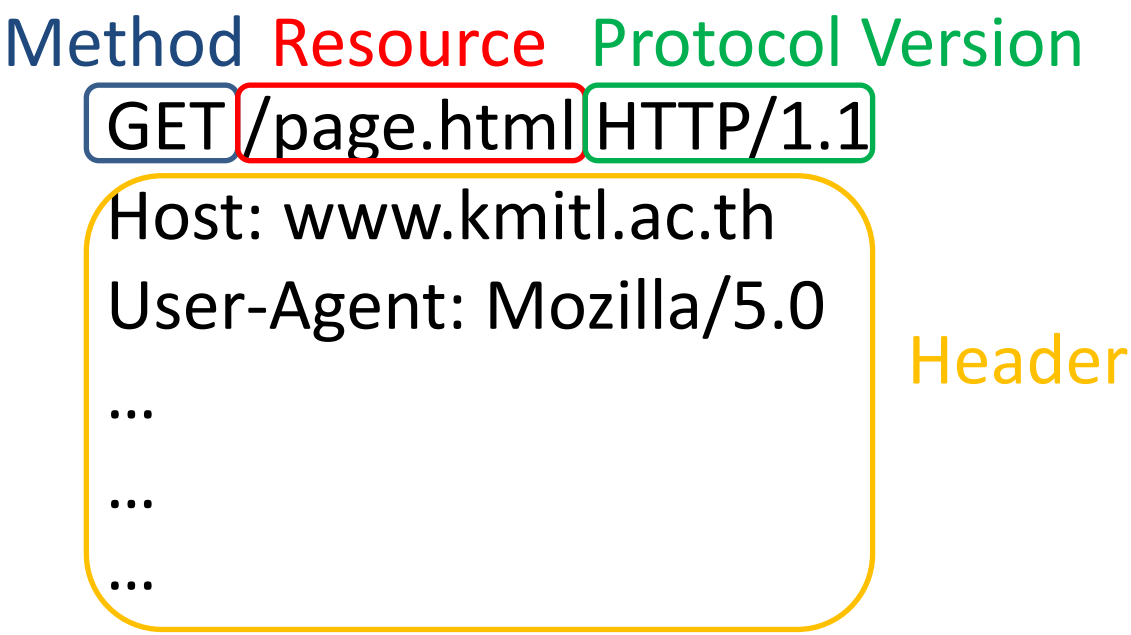
HTTP Protocol

- 2 common methods
  - Get
  - Post

HTTP GET Method

URL: http://www.kmitl.ac.th/page.html

- Requesting resource



HTTP GET Method

URL:

Query string

http://www.kmitl.ac.th/q.php?id=123&name=John

Method Resource Protocol Version

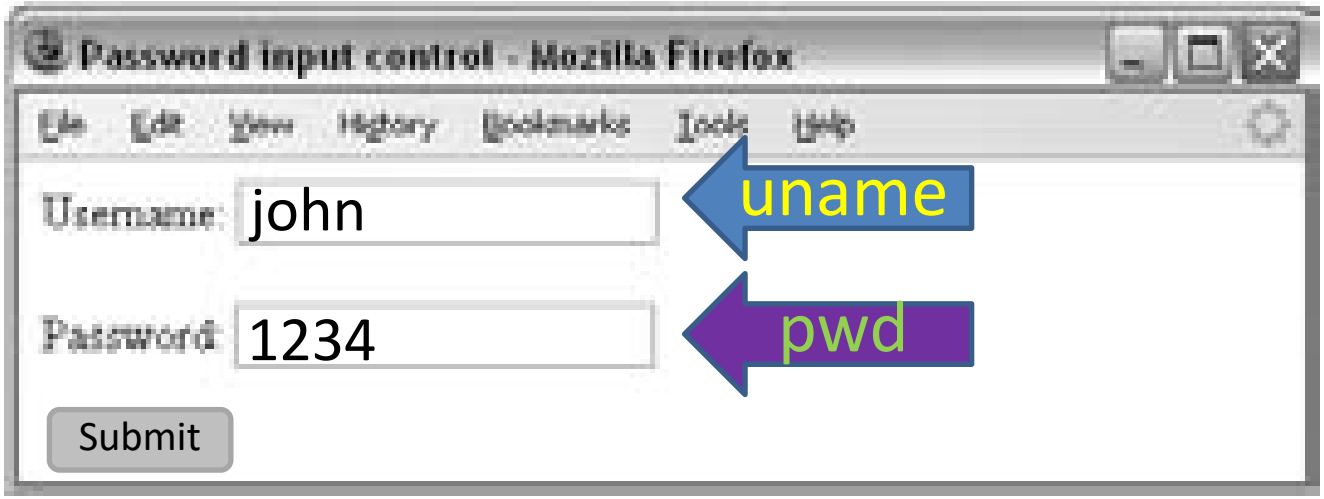
GET /q.php?id=123&name=John HTTP/1.1

Host: www.kmitl.ac.th
User-Agent: Mozilla/5.0
…
…
…

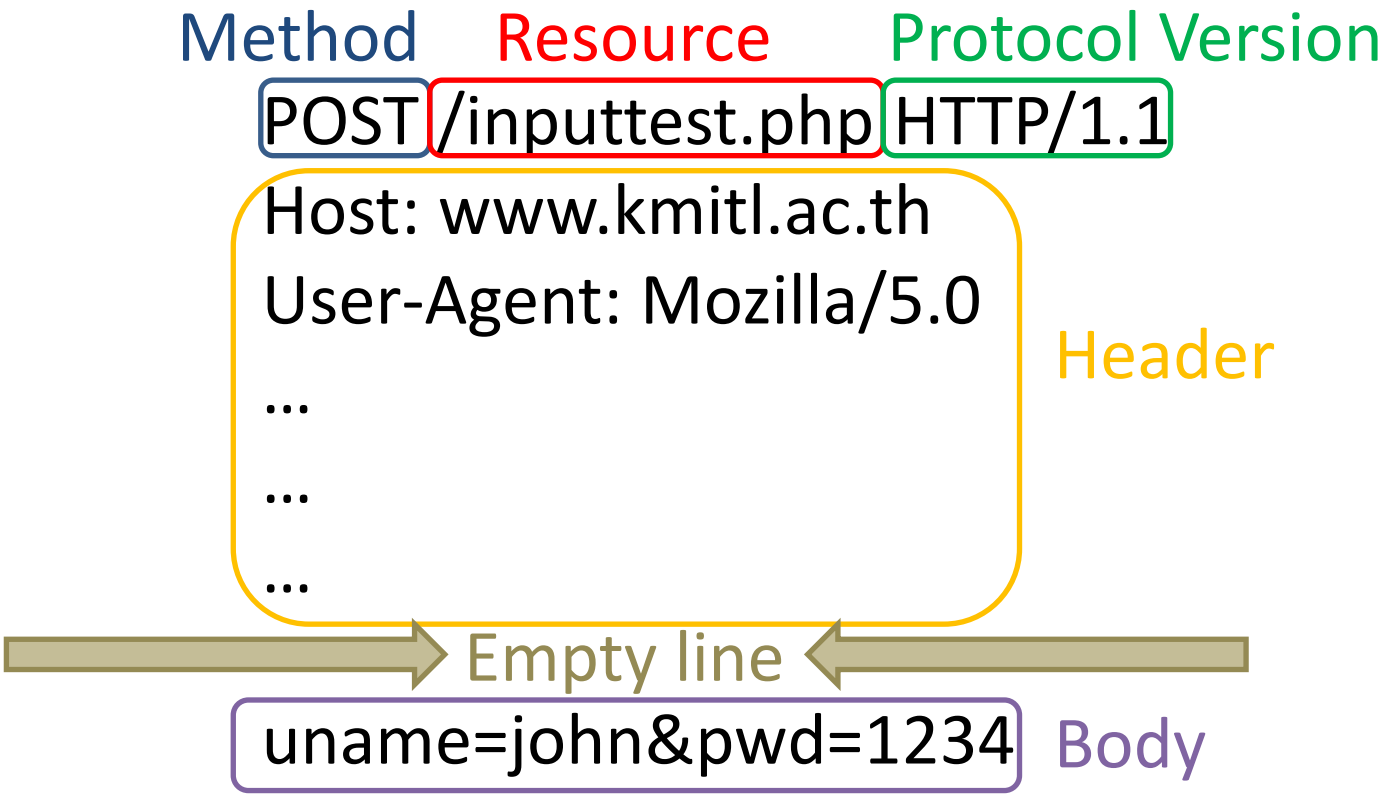
Header

HTTP POST Method

< form action=”http://www.kmitl.ac.th/inputtest.php” **method=”post”** >
Username: < input type=”text” **name=”uname”** value=”” size=”20” maxlength=”20” >
< br >
Password: < input type=”password” **name=”pwd”** value=”” size=”20” maxlength=”20” >
< input type=”submit” value=”Submit” >
< /form >



HTTP POST Method



Get VS Post

	Get	Post
Data length to be sent	~4000	unlimited
Can send a chunk of data	No	Yes
Location of data in message	URL	Body
Can access CGI without using Form	Yes	No
Can retrieve file or other resource	Yes	No

How does it work?

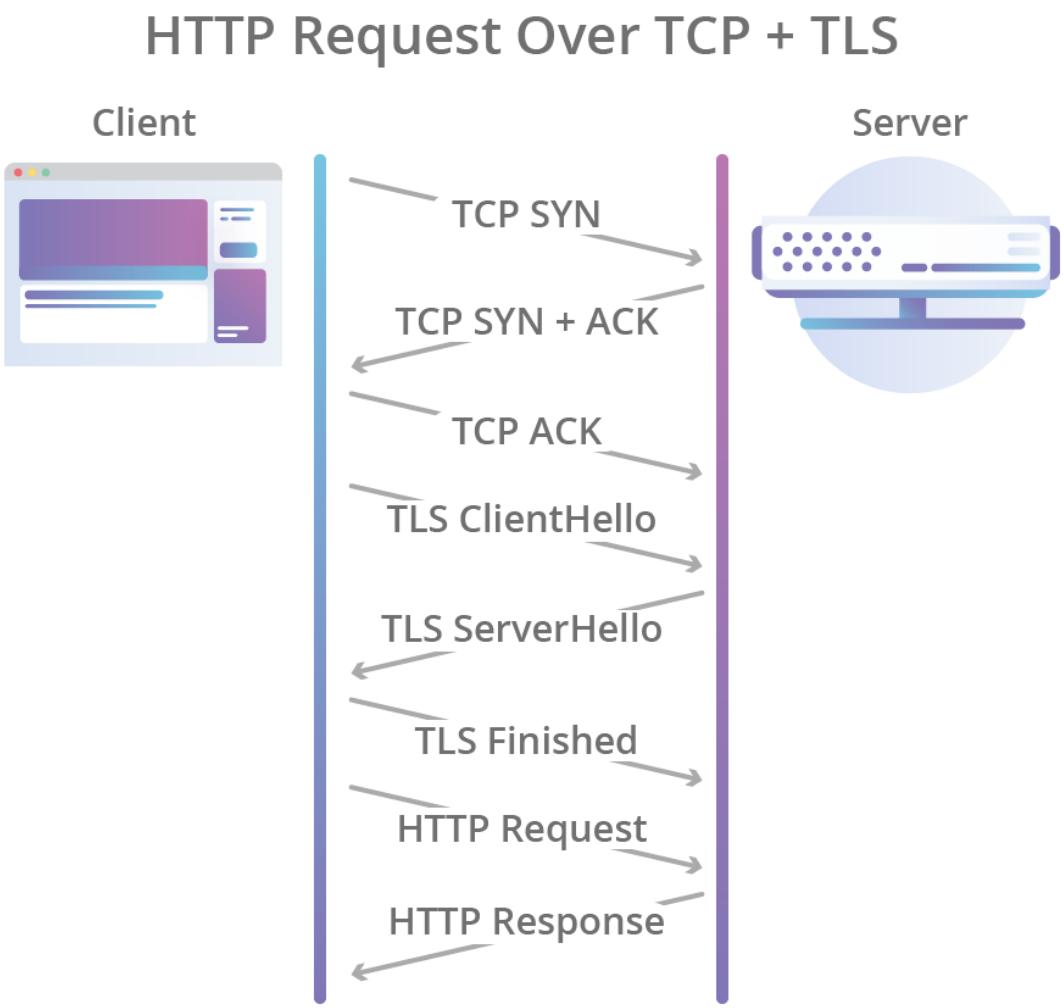


Image source: Cloudflare

How does it work?

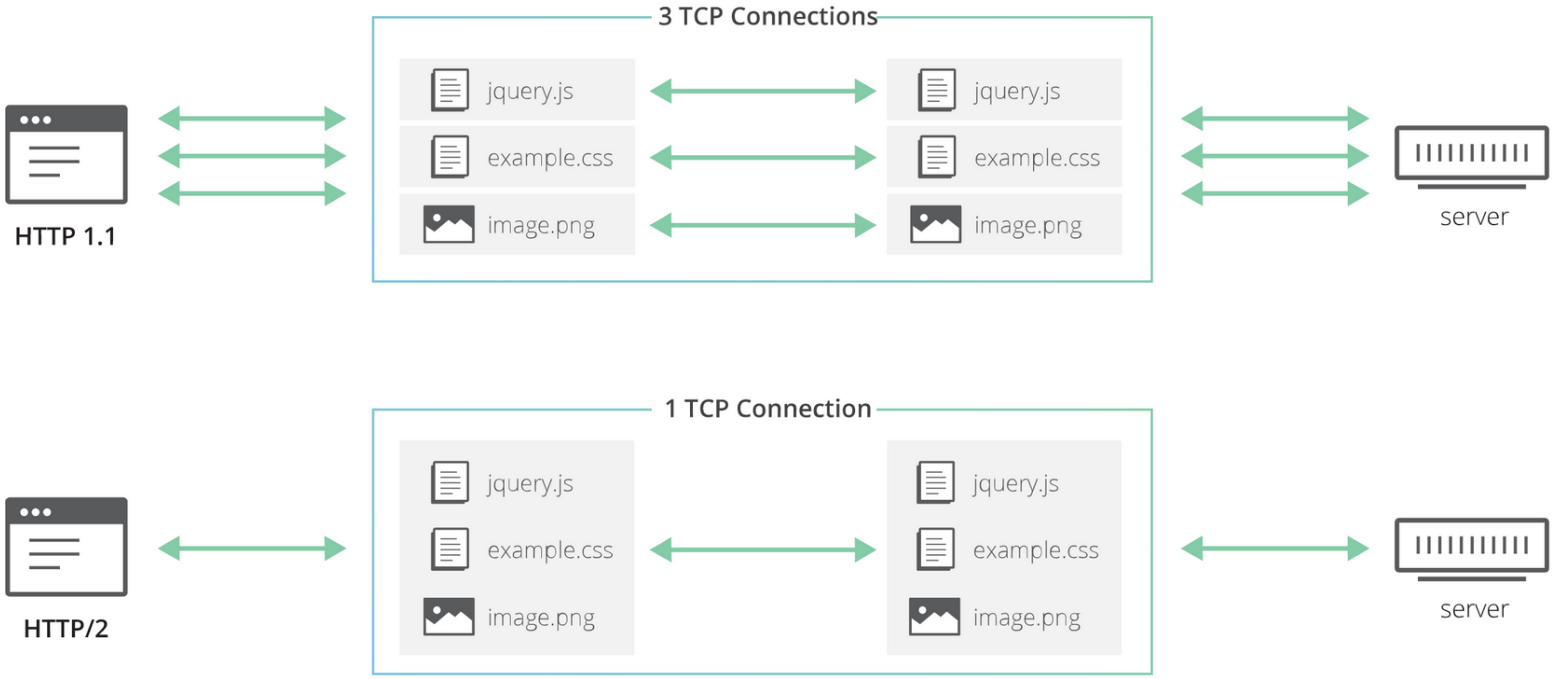


Image source: Cloudflare

How does it work?

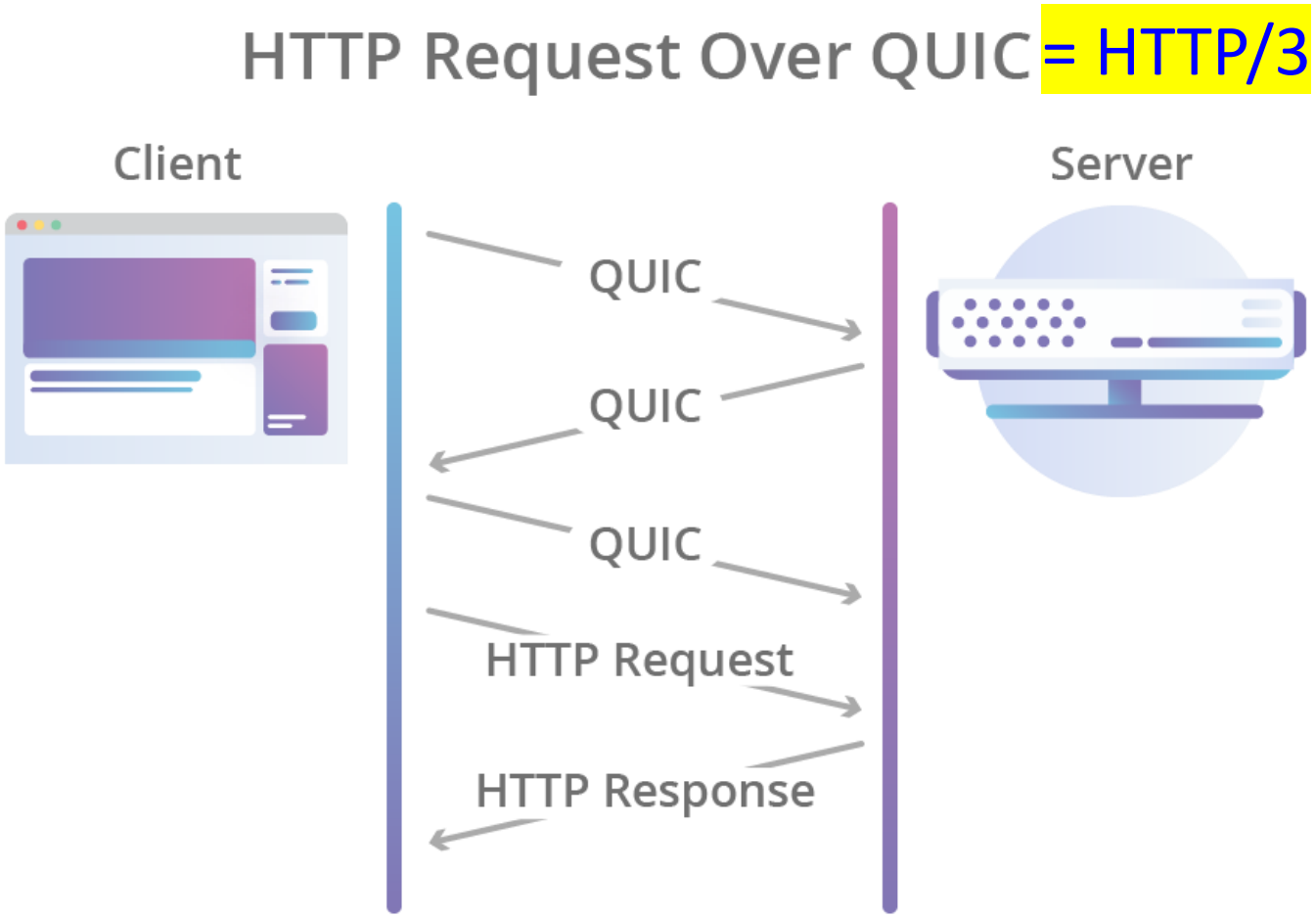


Image source: Cloudflare

Introducing HTML and xHTML

- HTML standard is overseen by W3C
- HTML 4.01 released Dec. 1999
  - Added stricter rules to HTML 4.01 in Jan. 2000 creating what is known as xHTML
  - xHTML = Extensible Hypertext Markup Language

HTML5

- HTML5 released as a living-standard by WHATWG in 2012 and is continuously updated
  - WHATWG = Web Hypertext Application Technology Working Group
    - Formed 2004
    - Major web browser vendors are member of WHATWG
    - W3C supported WHATWG in 2006
  - Living-standard = new feature can be added but old features cannot be removed

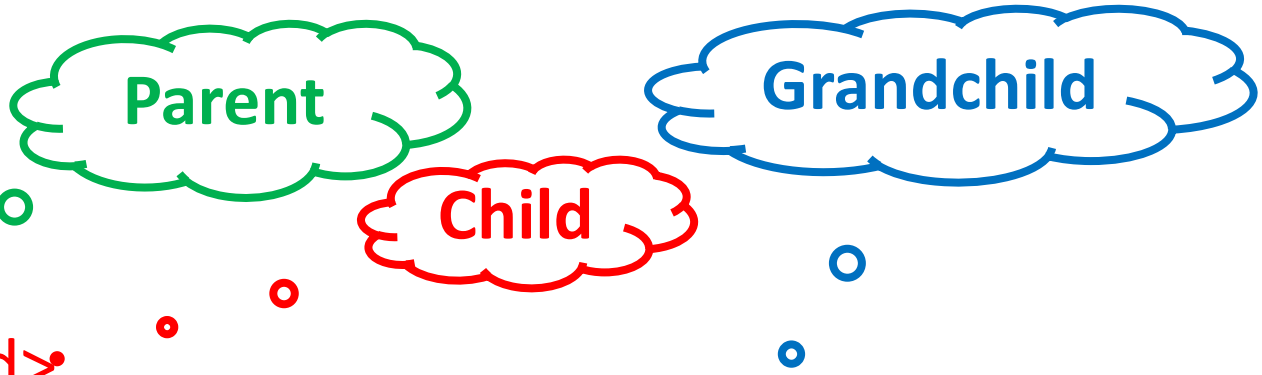
HTML5

- W3C developed a definitive HTML5
  - W3C and WHATWG decided on separation in 2012
  - W3C HTML5 released 2014
  - W3C HTML5.1 2nd edition released 2017
  - W3C HTML5.2 released 2017

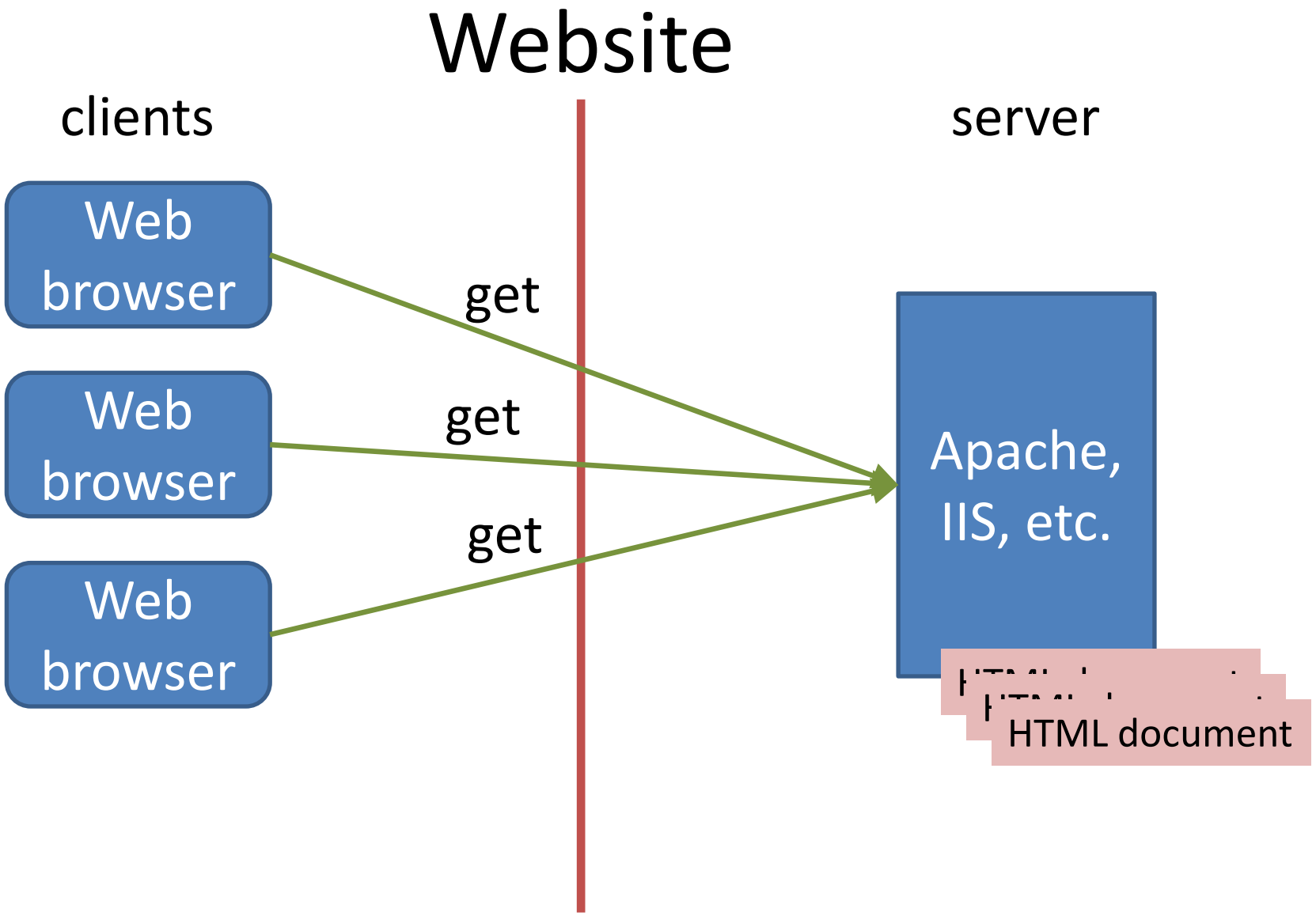
Hypertext Markup Language

- Tags, Elements and Attributes
  - Tags = tokens enclosed by angle brackets - < >
  - Elements define the structure of document and lay the foundation for its presentation and manipulation, contained within one or two tags
  - Attributes = Tag modifiers compose of 2 parts: name and value
- Mostly case insensitive and not necessary to quote the value part of the attributes

- HTML example




```
<html>
    <head>
        <title>Page's title</title>
    </head>
    <body>
        <h1>headline 1</h1>
        <p>Paragraph of text</p>
    </body>
</html>
```


Website
clients
server
Web browser
Web browser
Web browser
get
get
get
Apache, IIS, etc.
HTML document

Other Web Technologies

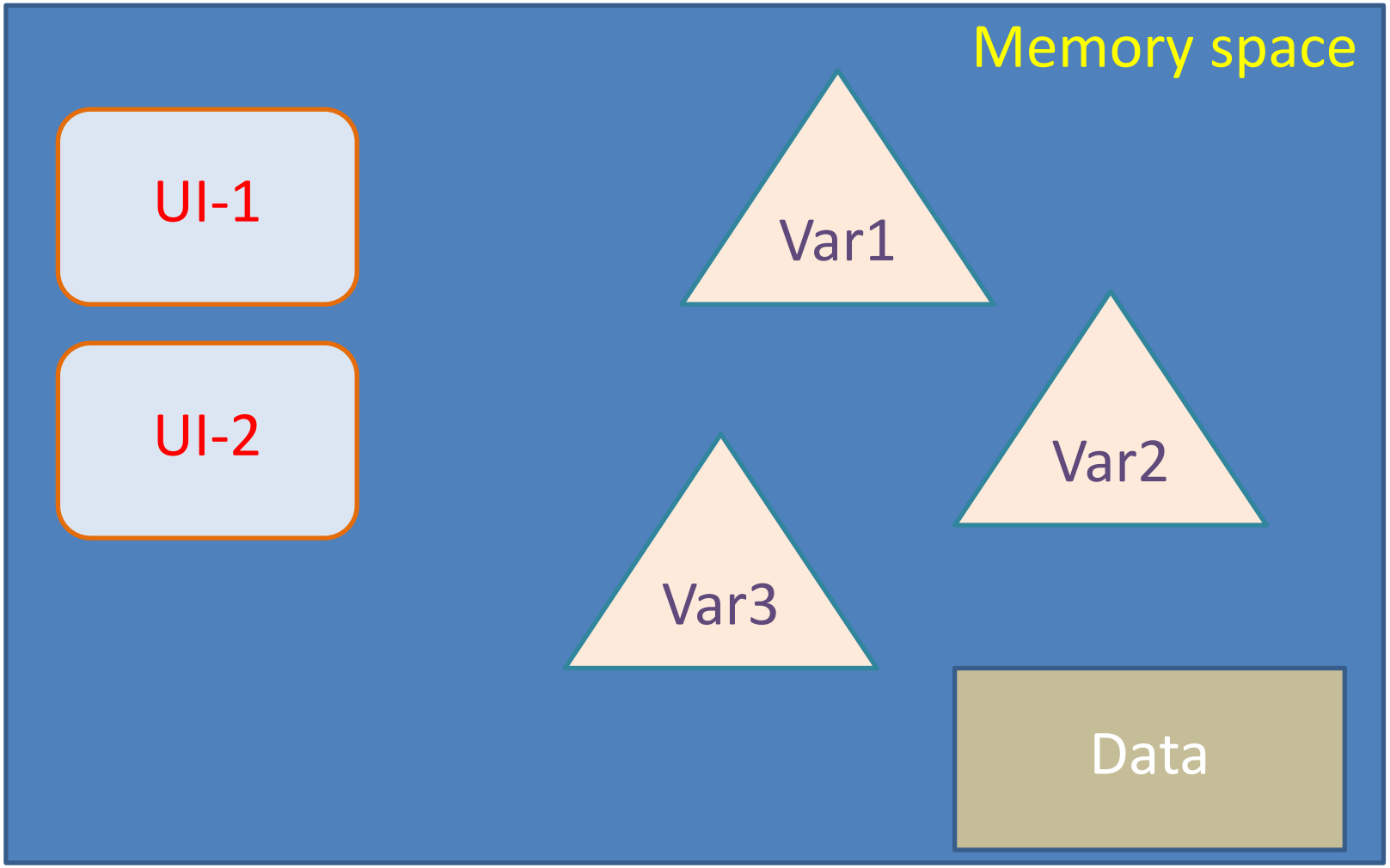
- CSS
  - Cascading Style Sheets
  - Controls visual aspects of web pages
  - Uses in conjunction with DOM
- DOM
  - Document Object Model
  - Defines the structure of html document
- ECMAScript
  - Client-side scripts
  - JavaScript is a dialect of the ECMAScript standard

Improving the Web Experience

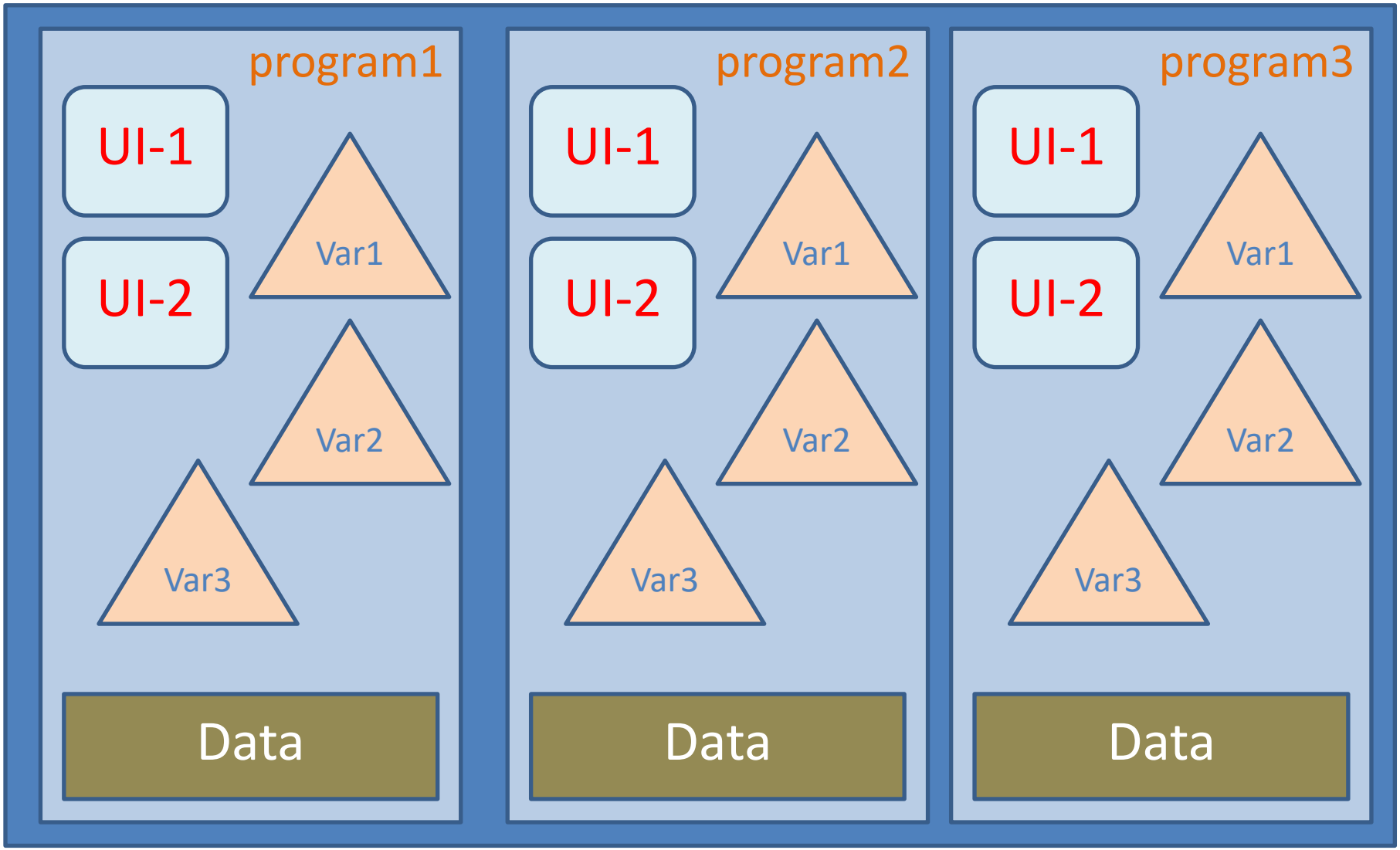
- Interaction
  - Client-side scripting
- Refreshing an entire page is not efficient
  - AJAX

Web Dev

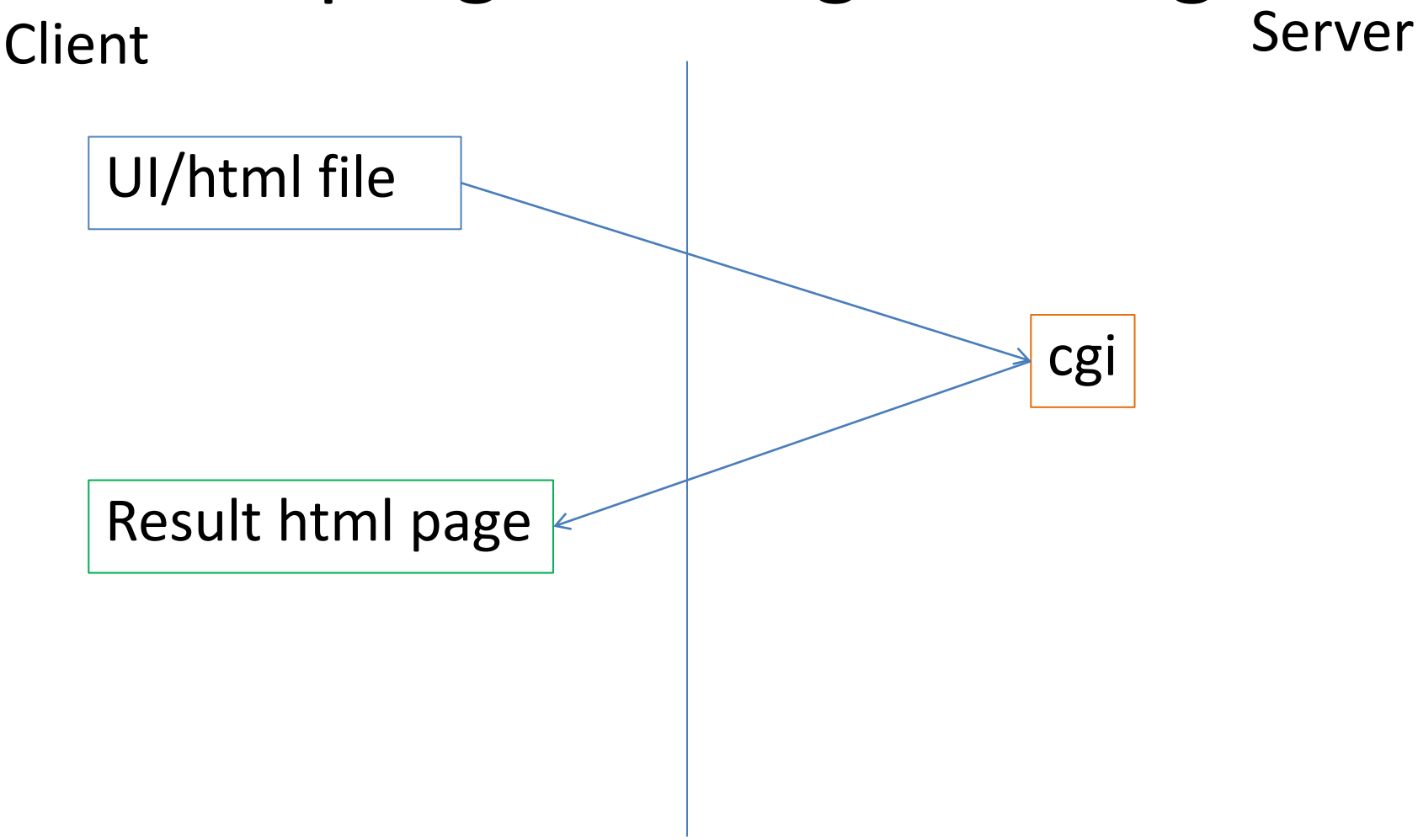
Common Program



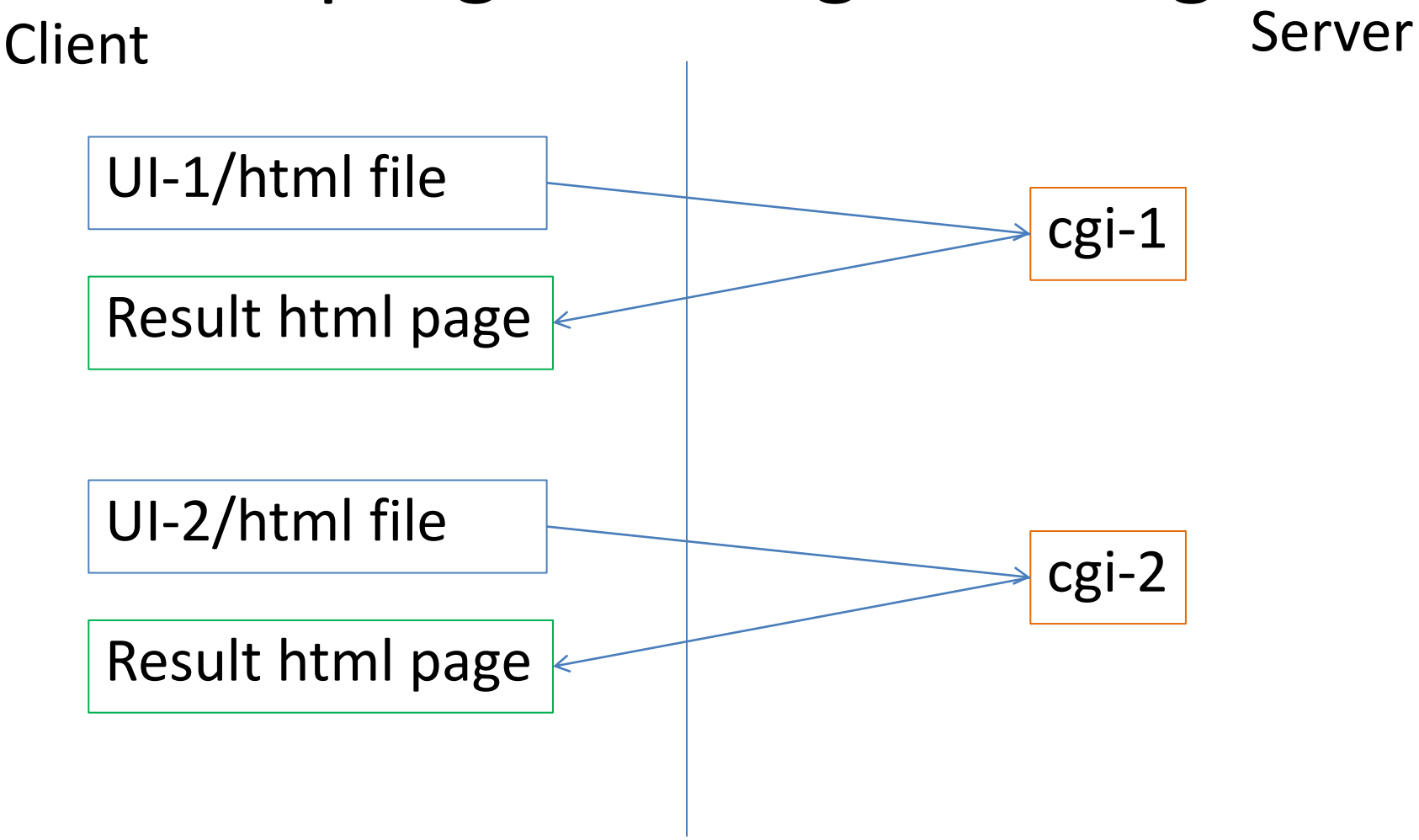
Common Program



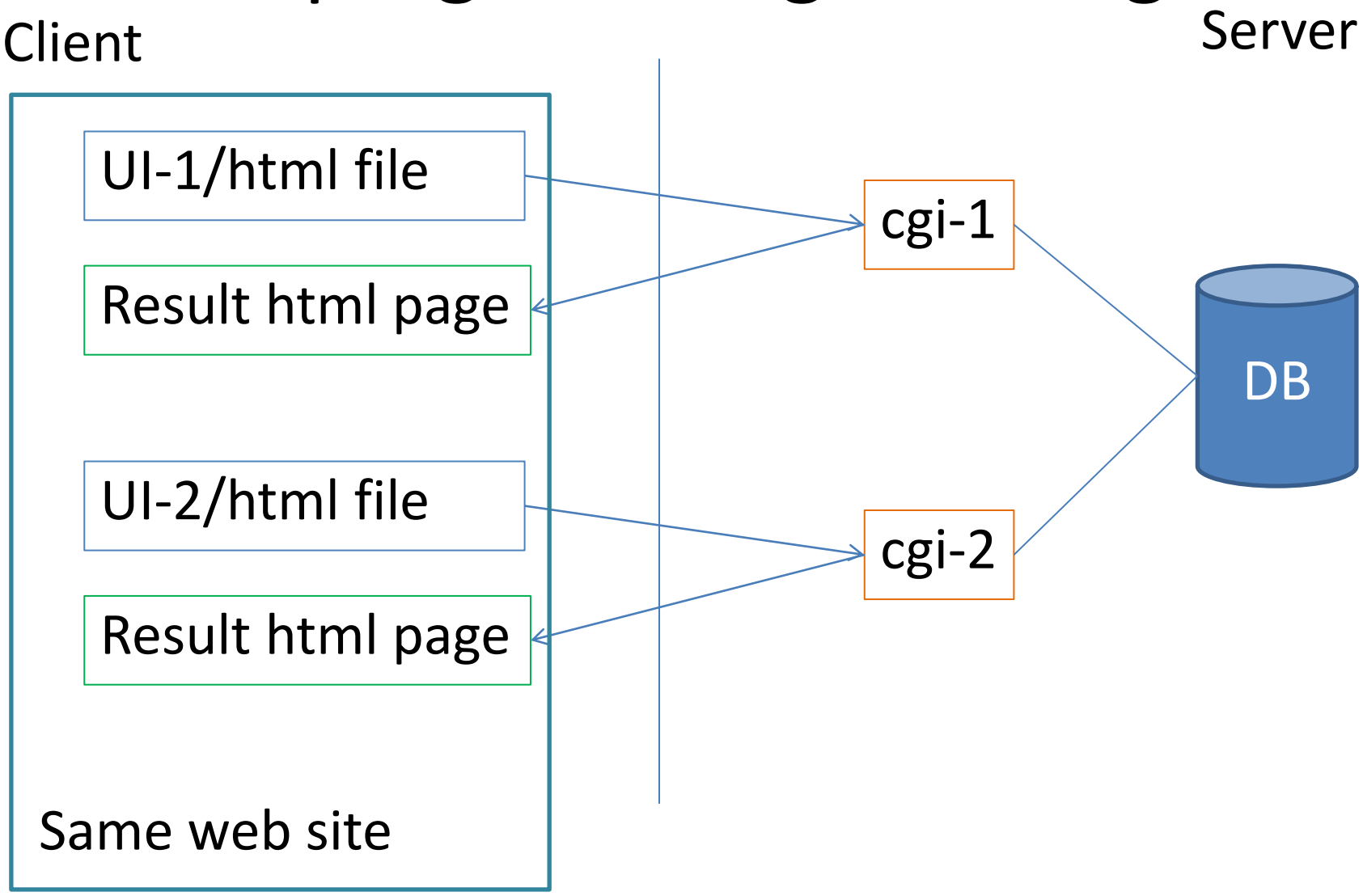
Web programming challenge



Web programming challenge



Web programming challenge

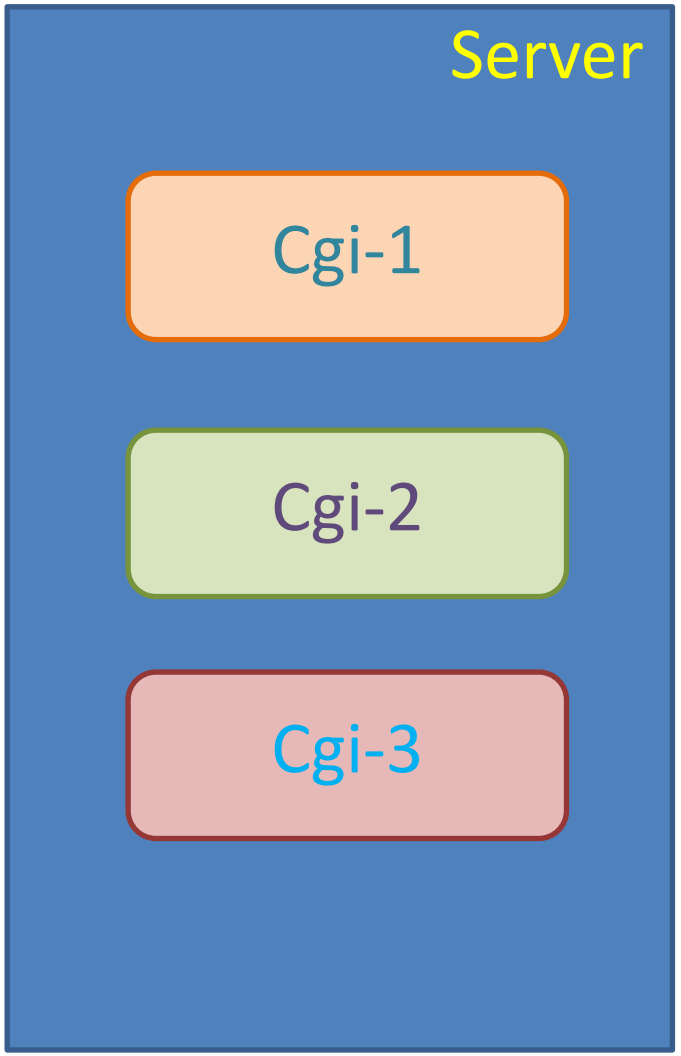


Difficulties

Service existence

Service info.
-Parameters
-Address
-Return type
-Etc.

Data format



Example of data sending with GET

www.sample.com/chk.cgi**?**fn=John**&**ln=Doe

URI

Data

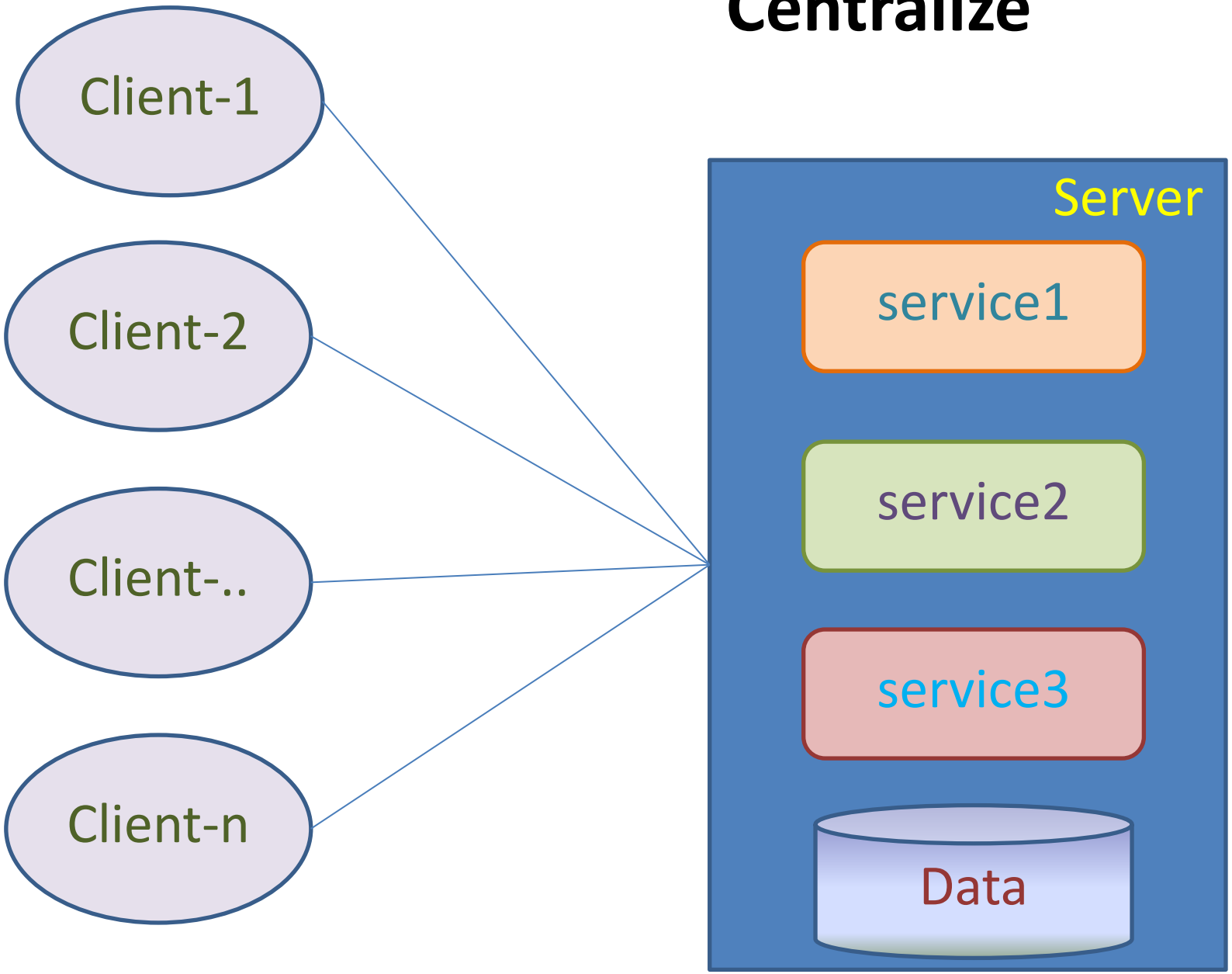
Evolution of WWW

- From User Point of View
  - Web 1.0 (1991 – 2003)
    - Static in term that user can not change any data
    - Example: www.kmitl.ac.th
  - Web 2.0 (2003 – present)
    - Users can do more than just retrieve information
    - Provide user with UIs, software, storage
    - User customizable
    - Examples: www.facebook.com, www.wikipedia.org

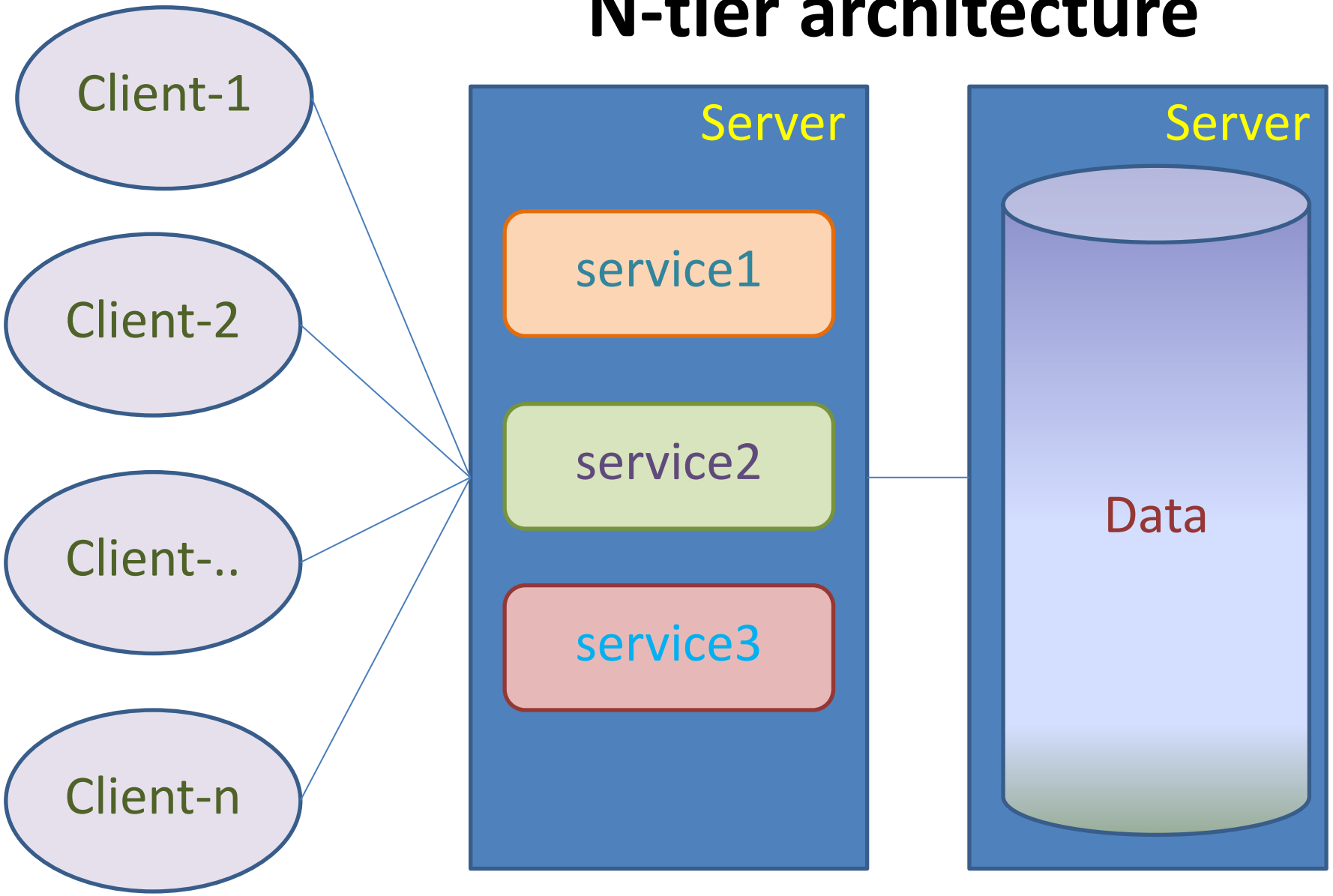
Evolution of WWW (cont.)

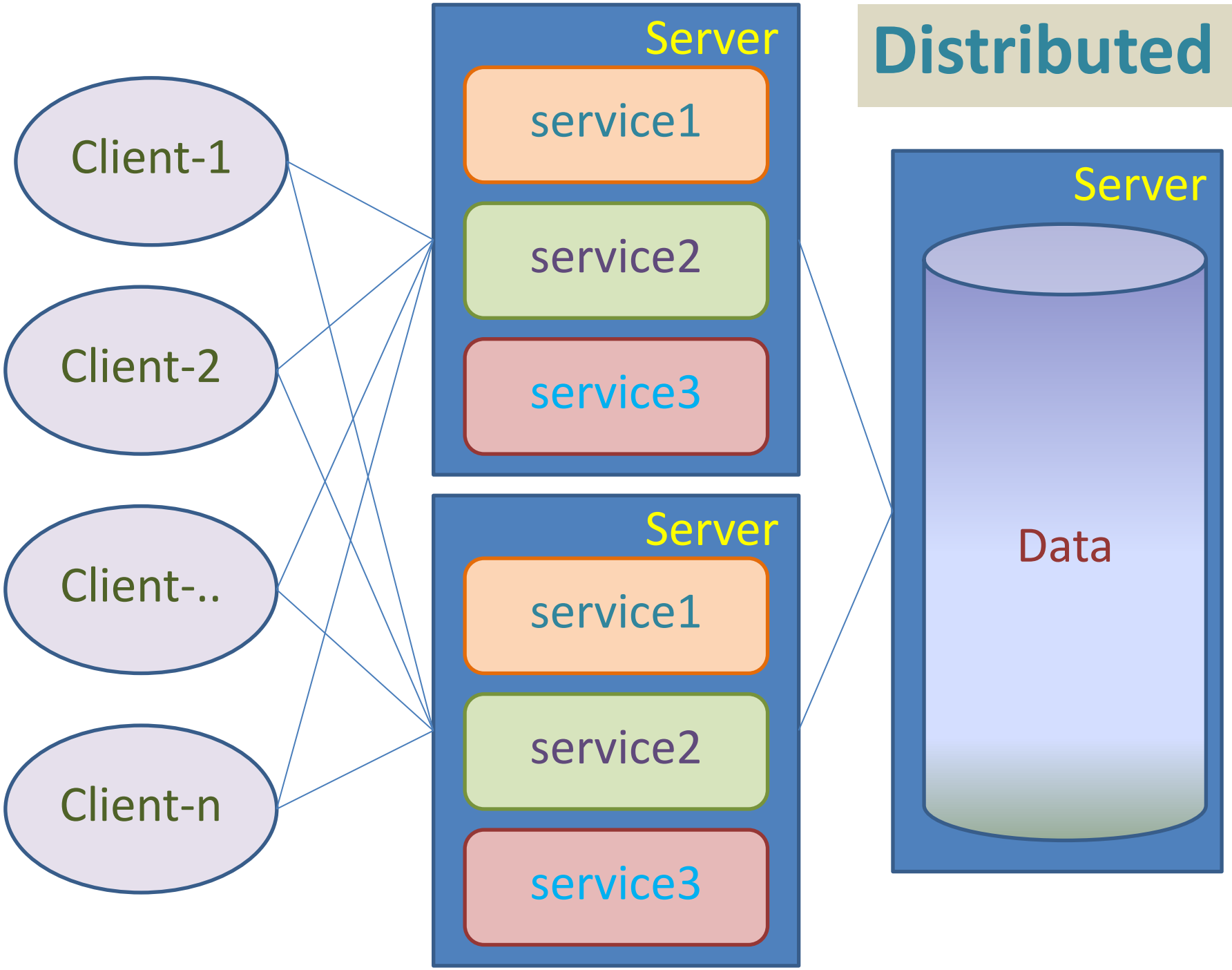
- From developer aspect
  - Client side
    - HTML
    - Web Scripting languages: JavaScript, VBScript
    - DHTML
    - AJAX
  - Server side
    - CGI -> Perl, C, C++, Pascal, Unix shell script, Python, TCL/TK, etc.
    - Server Side Include (SSI) -> .shtm or .shtml or .stm
    - Interpreter as a part of web server software -> PHP
    - Web Service, Web API, etc.

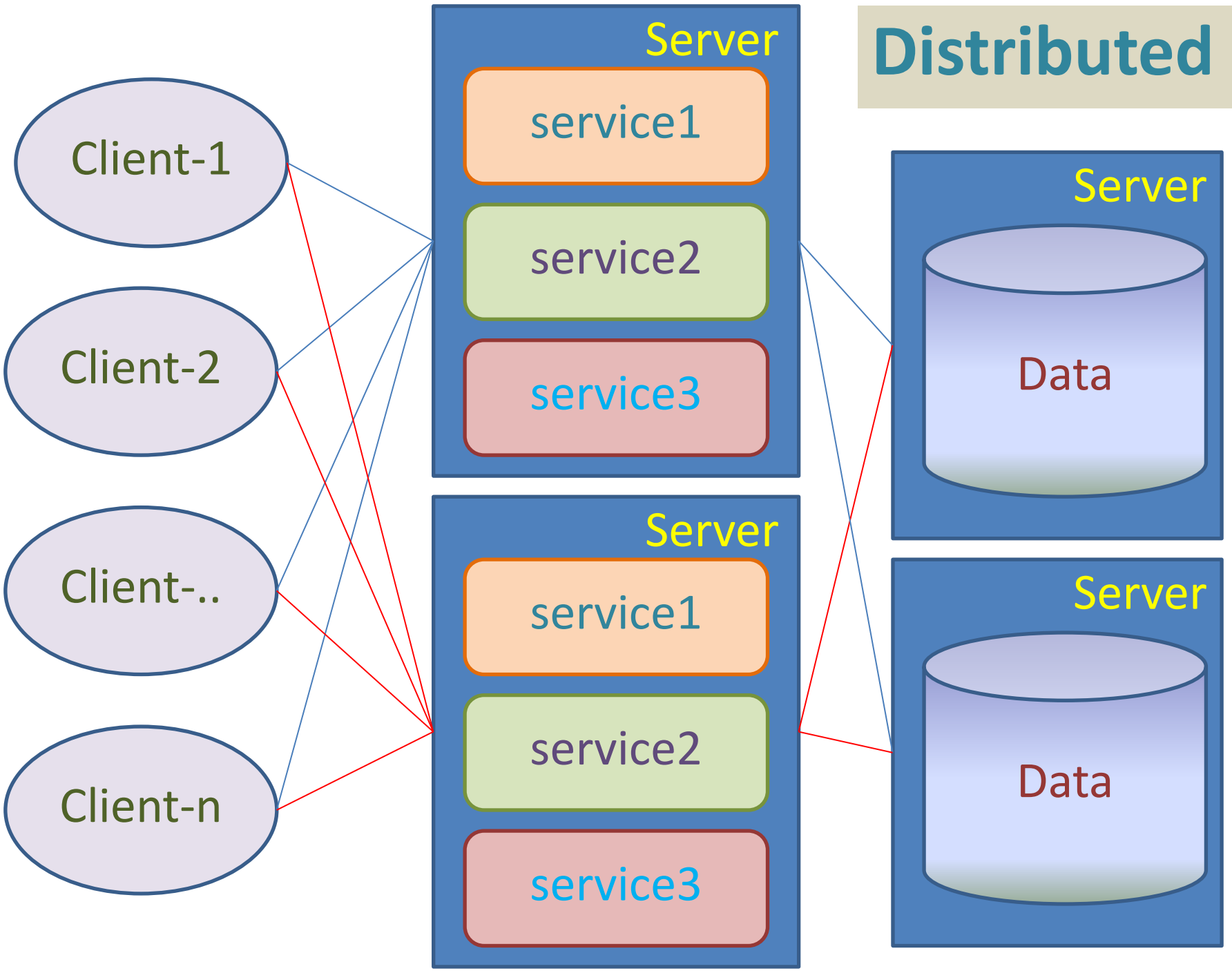
Centralize

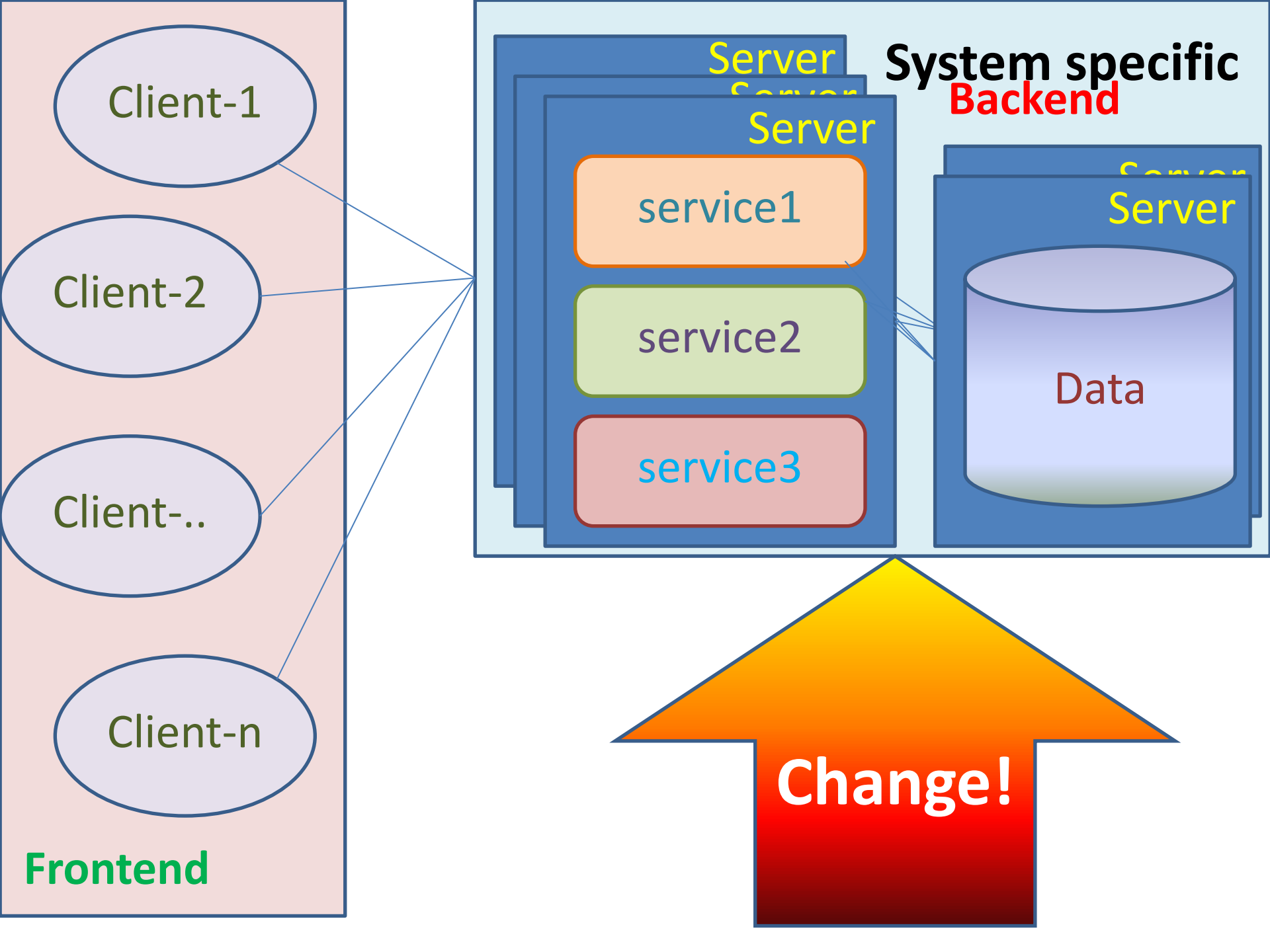

Client-1
Client-2
Client-..
Client-n
Server
service1
service2
service3
Data

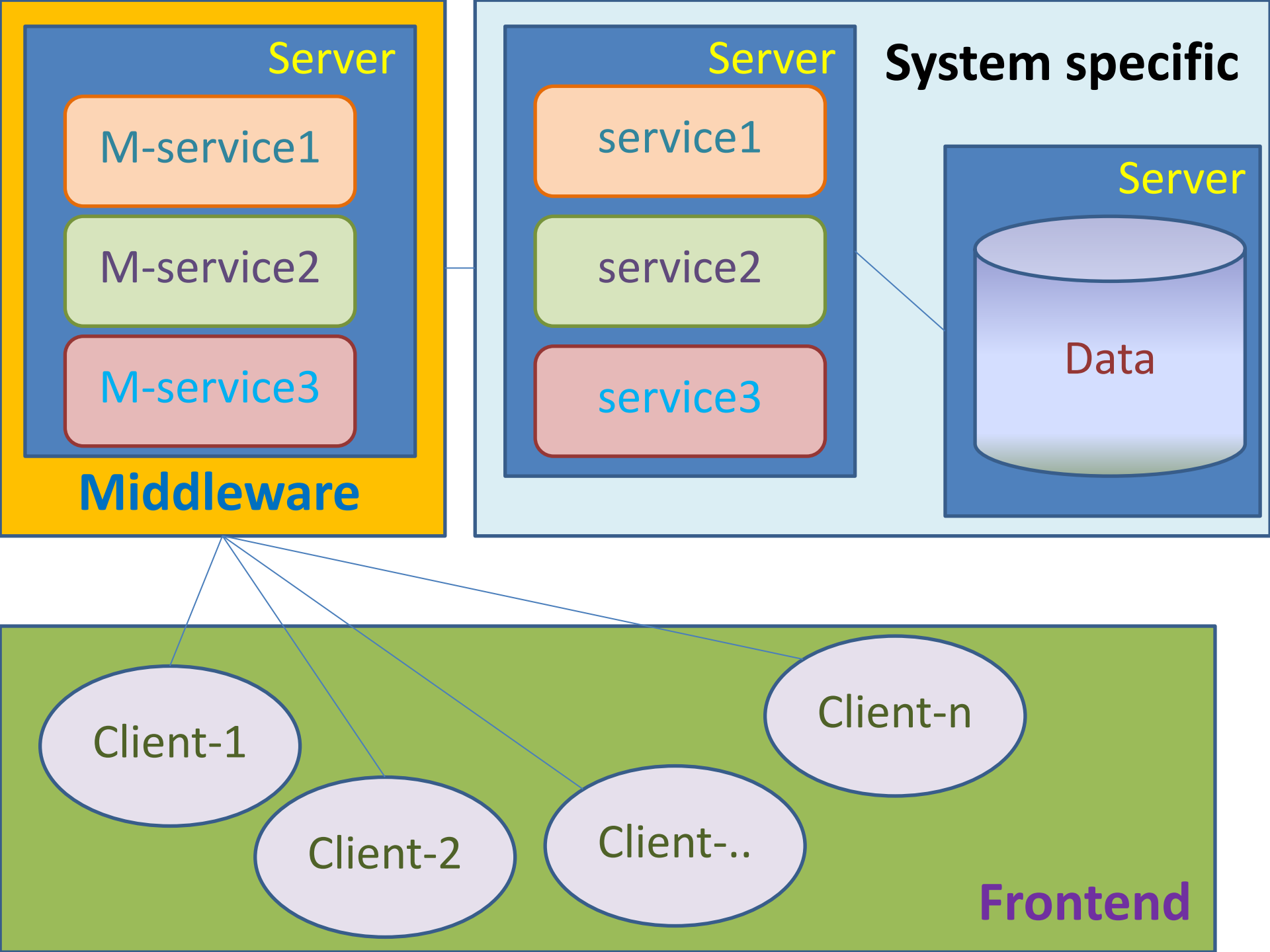
N-tier architecture




Distributed
Client-1
Client-2
Client-..
Client-n
Server
service1
service2
service3
Server
service1
service2
service3
Server
Data


Distributed
Client-1
Client-2
Client-..
Client-n
Server
service1
service2
service3
Server
service1
service2
service3
Server
Data
Server
Data


Client-1
Client-2
Client-..
Client-n
Frontend
Server
Server
Server
System specific
Backend
service1
service2
service3
Server
Data
Change!


Server
M-service1
M-service2
M-service3
Middleware
Server
service1
service2
service3
System specific
Server
Data
Client-1
Client-2
Client-..
Client-n
Frontend