# SOFTWARE ARCHITECTURE
# 12. WORLD WIDE WEB

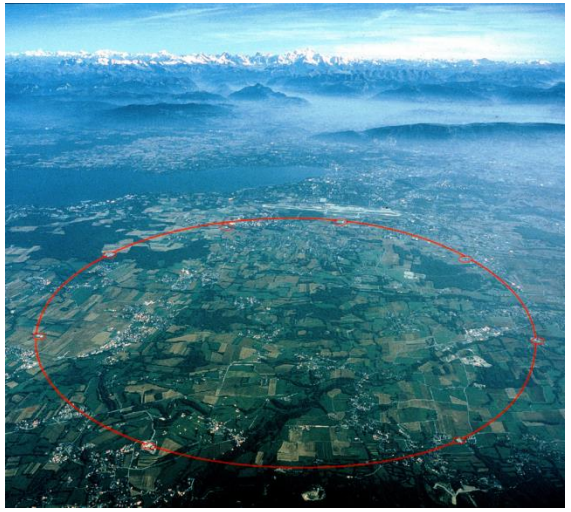
Tatsuya Hagino
hagino@sfc.keio.ac.jp

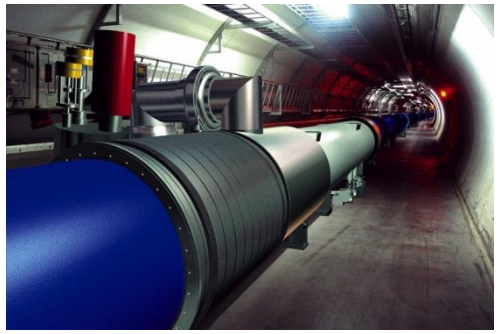
lecture URL

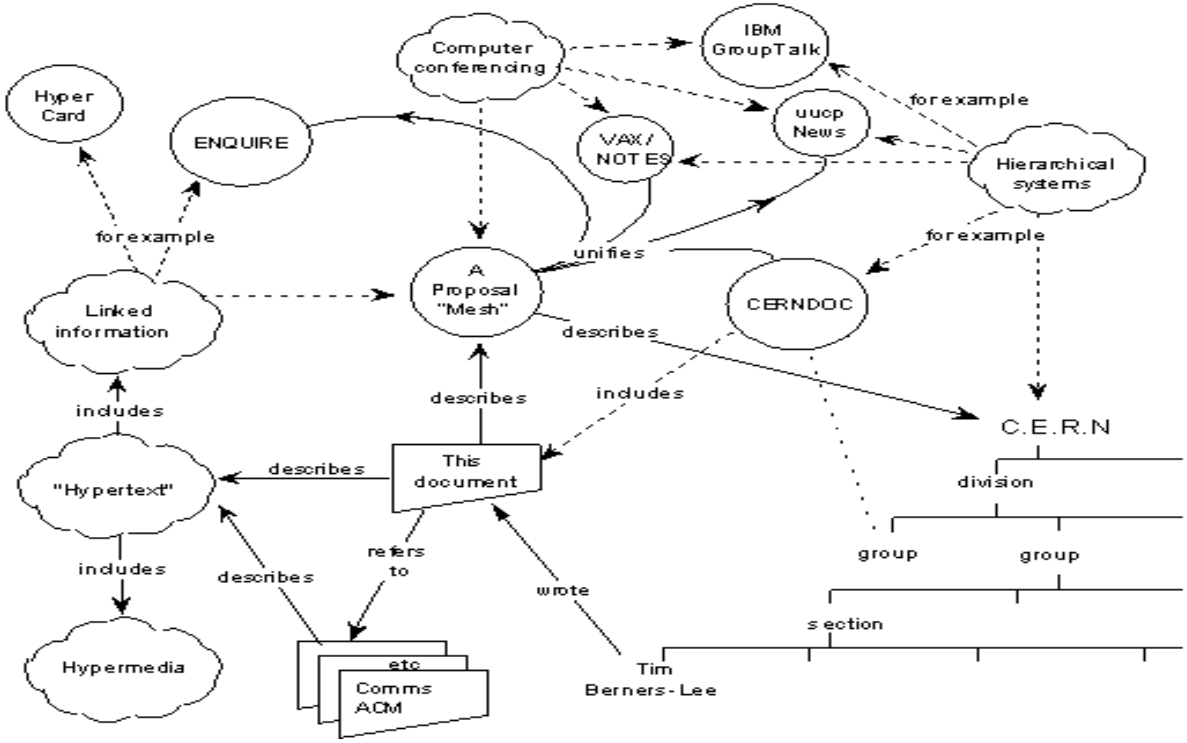
https://vu5.sfc.keio.ac.jp/slide/

# Birth of Web

- Developed around 1989 in Geneva, Switzerland
  - Tim Berners-Lee
  - Information management at CERN

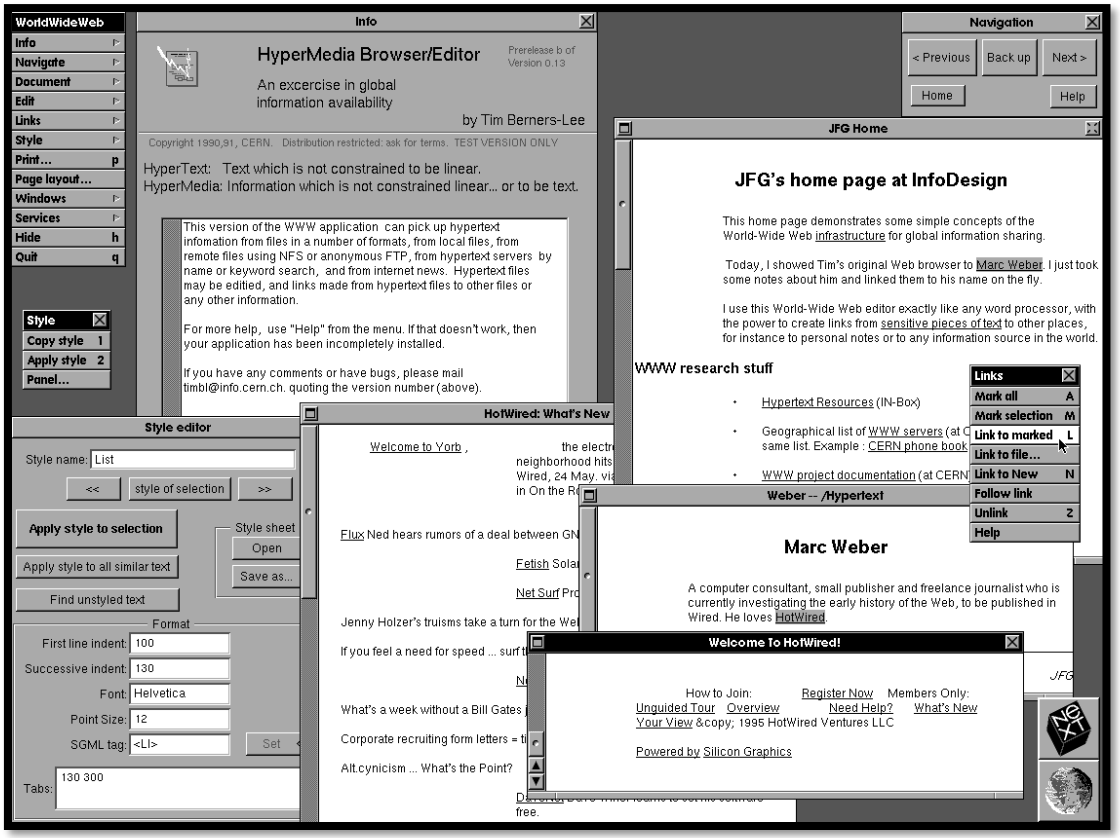







Web = Hypertext + Internet

# The first Web Server and Browser




# Compare Three Methods of Information Management

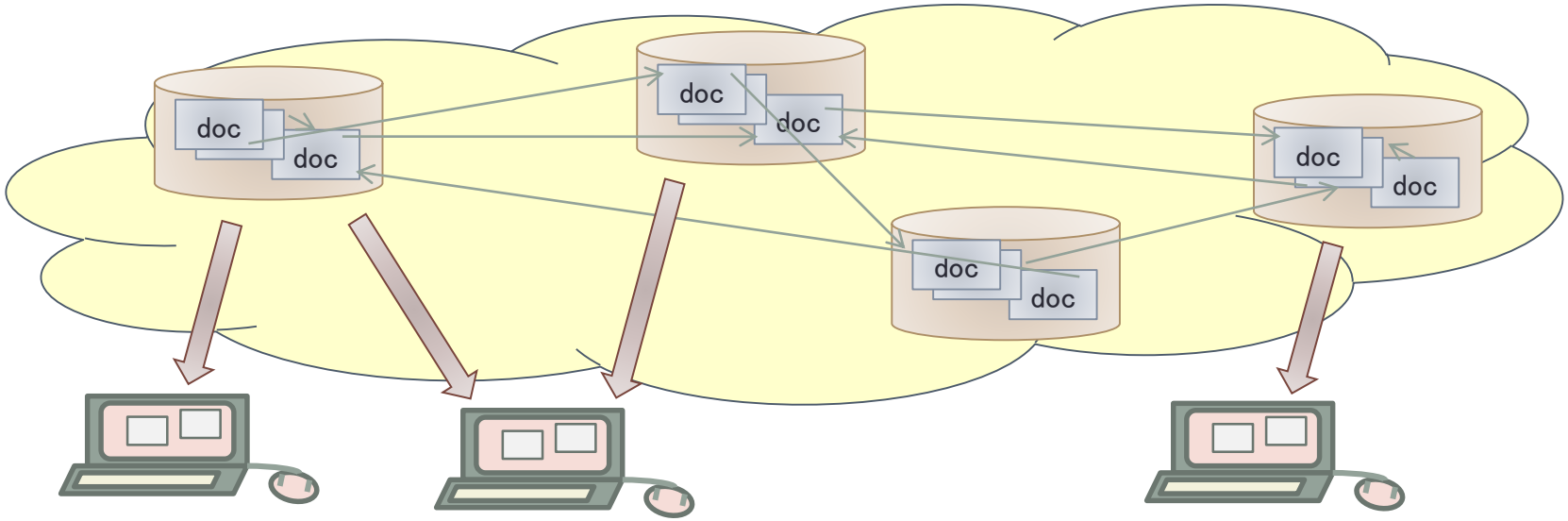
| hierarchical structure | keyword | hypertext |
| --- | --- | --- |
| Easy to manage.<br>Cannot represent real relations. | Search is fast.<br>Need to put keywords before search without knowing<br>actual user keywords. | Arbitrary structure.<br>Easy to represent real relations.<br>Keywords can be nodes. |

# What is Web?

**Web = internet + hypertext**

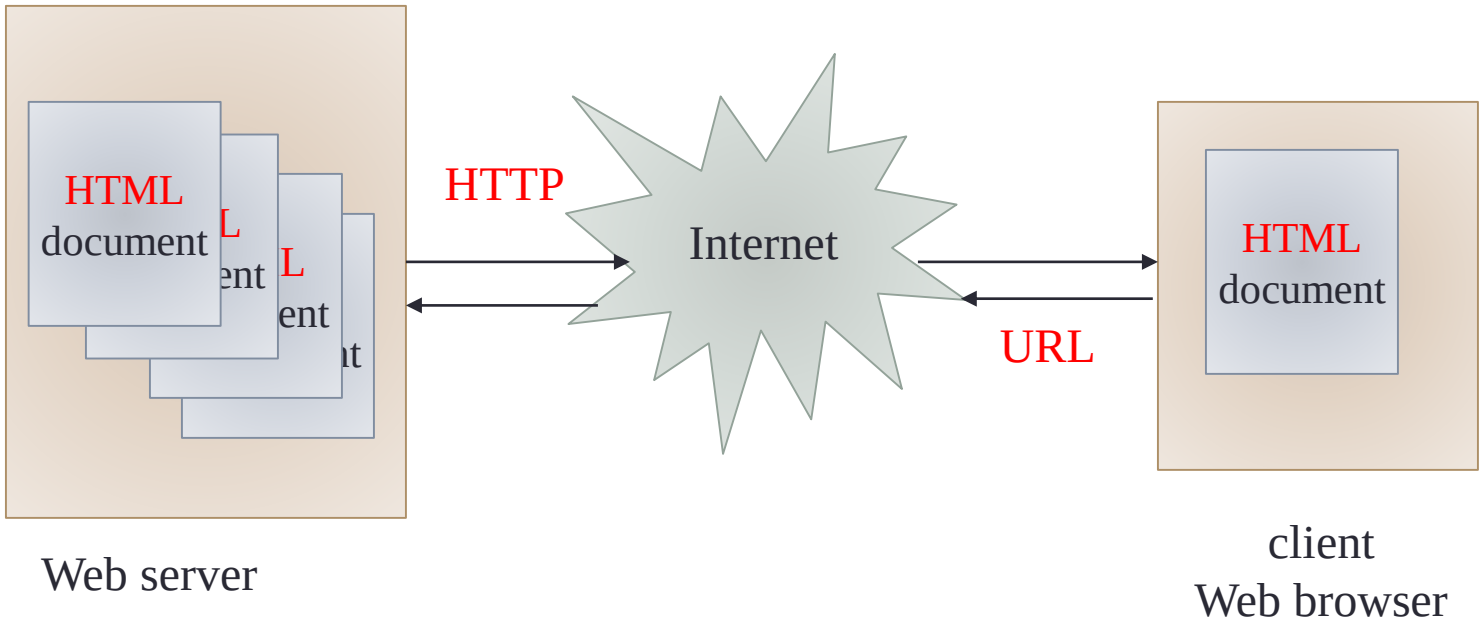
- Internet
  - Connecting global networks.

- Hypertext
  - Text with links to other texts.



# Web Basic Mechanism

- Write documents as hypertexts using HTML
- Use URL to specify locations
- Use HTTP to transfer documents from server to browser



# First Important Inventions for Web

- HTML + CSS
  - Web page description language
  - HTML: Hypertext Markup Language
    - for content
  - CSS: Cascading Style Sheet
    - for style
    - added later


- HTTP: Hypertext Transfer Protocol
  - Web page transfer protocol
  - Simplification of Anonymous FTP
  - Multimedia
  - Multi-language

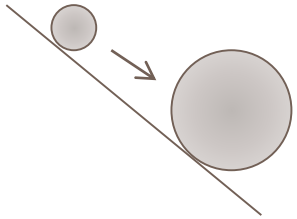
HTTP 0.8 → HTTP 1.0 → HTTP 1.1 → HTTP 2

- URL: Uniform Resource Locator
  - Web page position
  - Hypertext pointer

URL → URI → IRI

# Why Web becomes so popular?

- Free
  - Gopher had a license problem.
- Open system
  - Anybody can join.
  - Search engines automatically create indexes.
  - Network effect.



like snow ball

- Not strict
  - Broken links (404 Not Found)
  - HTML grammar error

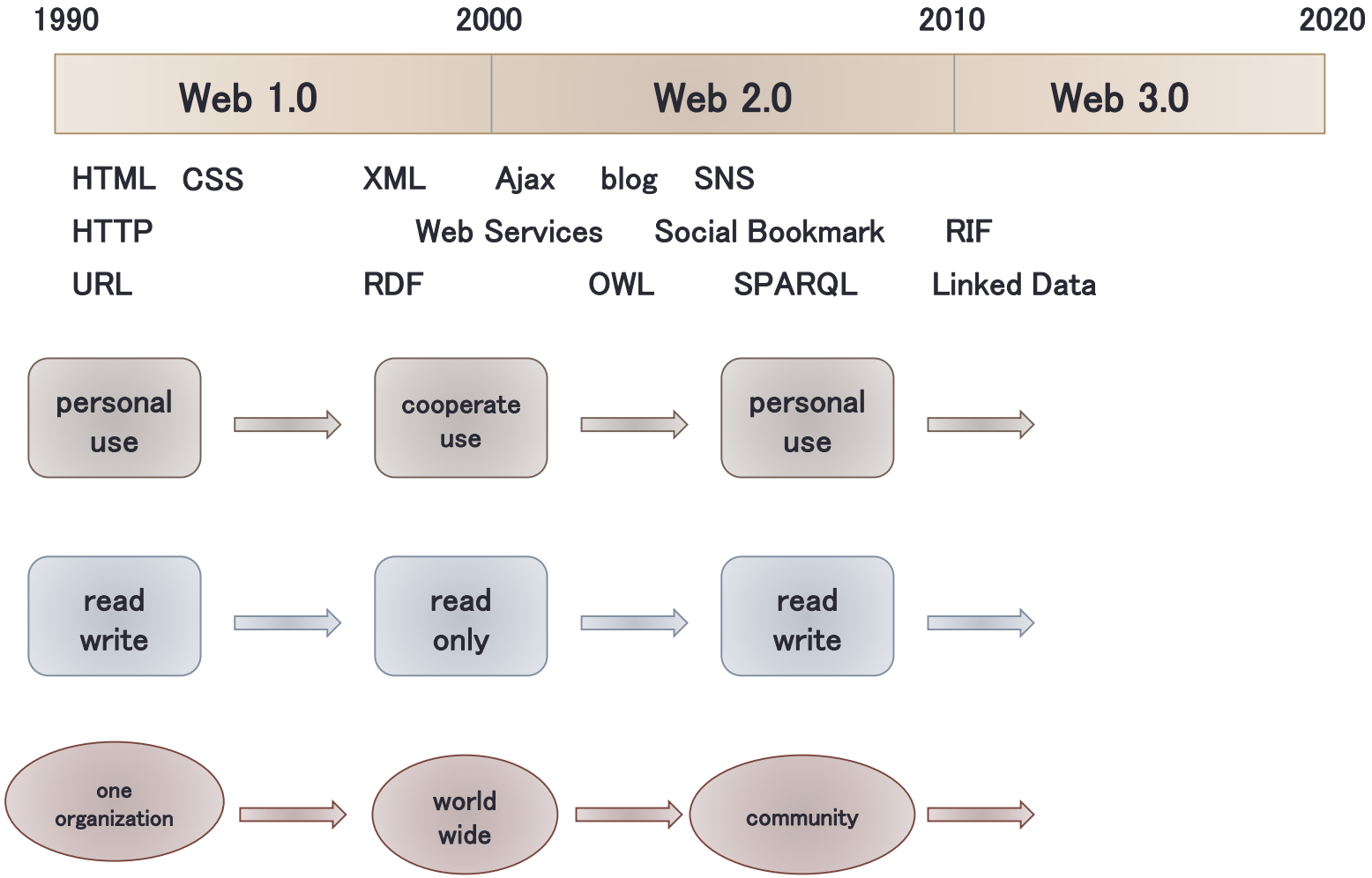
defective product
as hypertext system

- Standardization efforts
  - IETF
  - World Wide Web Consortium

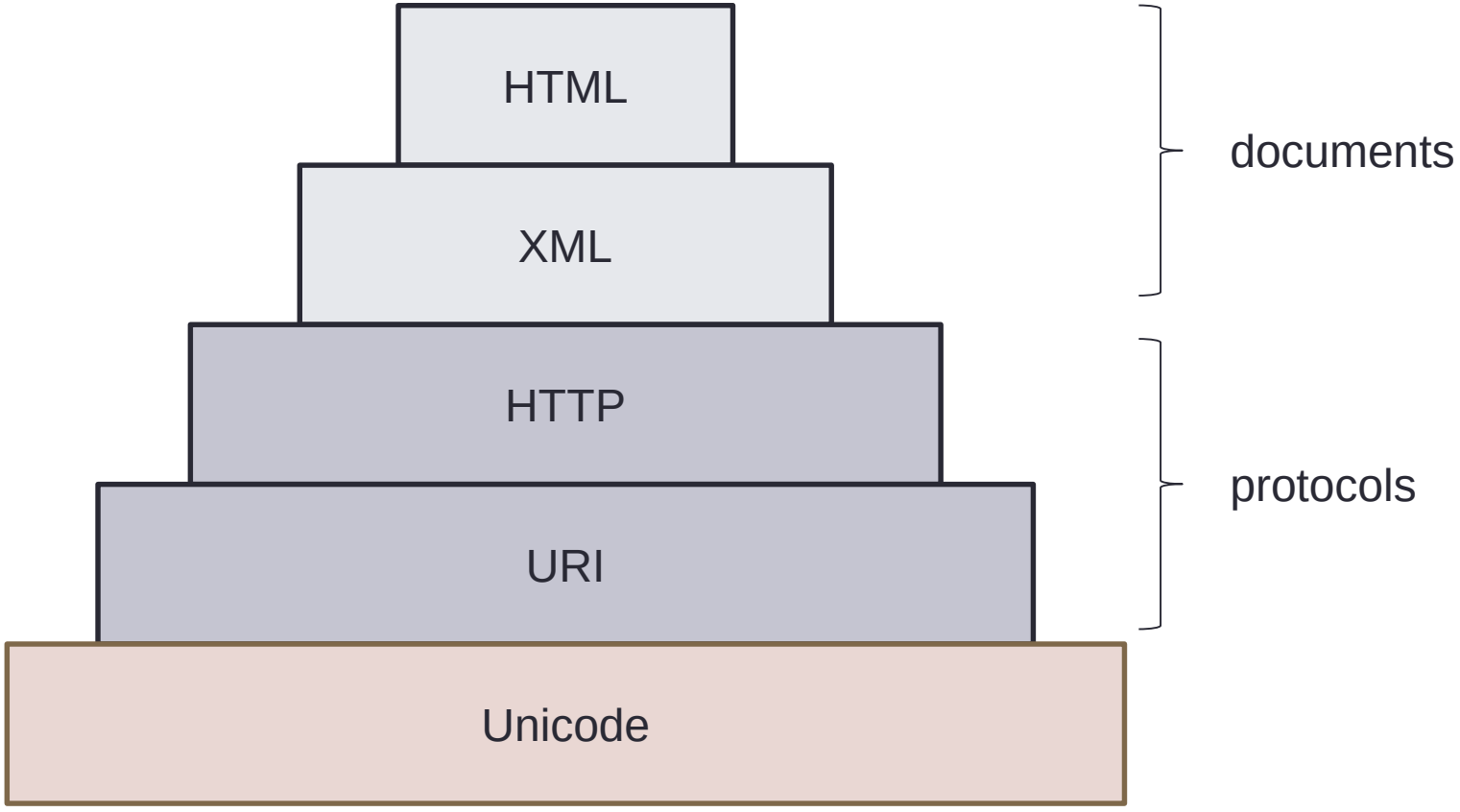
IETF

W3C WORLD WIDE WEB consortium

# From Web 1.0 to Web 2.0



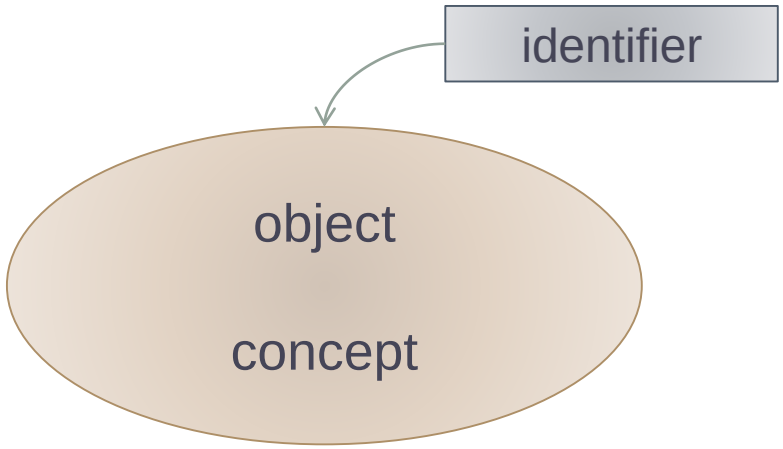
# Web Layer Structure



# URI (Uniform Resource Identifier)

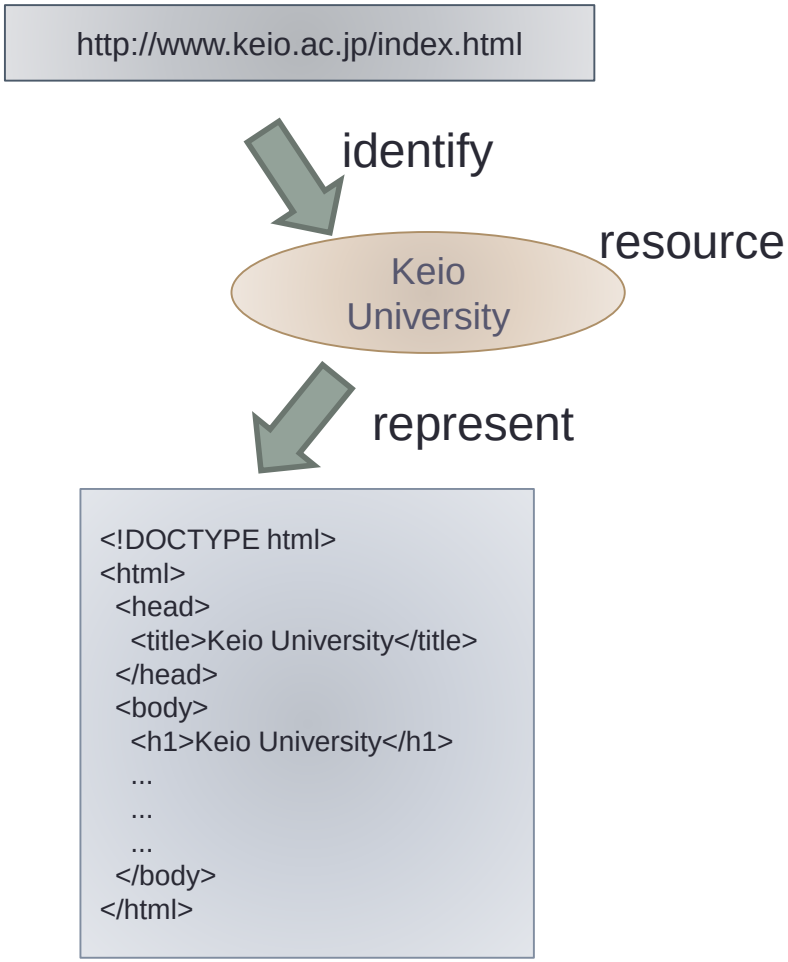
- Identifier
  - Names and numbers which identify objects and concepts.

- Identifiers around us.
  - personal identifier
  - book identifier
  - PC identifier
  - place identifier
  - ....

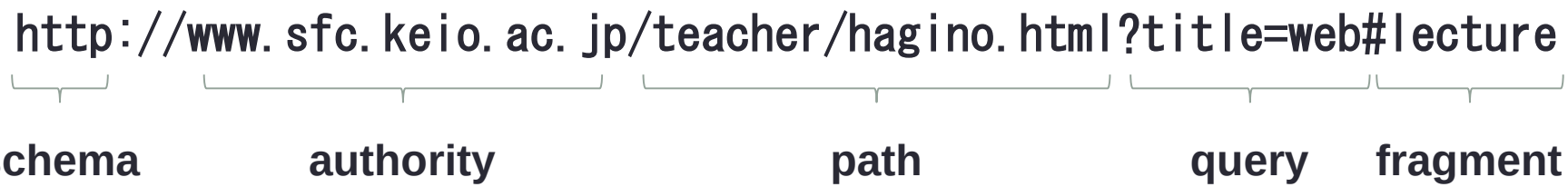


# Web Concept, Identifier and Representation

- Concept
  - Web Resource
- Identifier
  - URI (Uniform Resource Identifier)
- Representation
  - HTML+CSS
  - GIF



# URI Syntax




- Schema
  - protocol
- Authority
  - host name
  - server name
- Path
  - location in the authority
  - file name
- Query
  - query variable and value pairs
- Fragment
  - location in a document

Web server handles up to the query part and the fragment is handled by Web browser.

# URI Axioms

- Universality
  - Any Web resource has URI.
- Global Scope
  - The meaning of URI is always the same.
  - uniqueness
- Sameness
  - The same URI means the same thing.
  - Contents may be differ with the same meaning.
- Opacity
  - URI itself does not show the type of resource.
  - Resource type may depend on its representation.

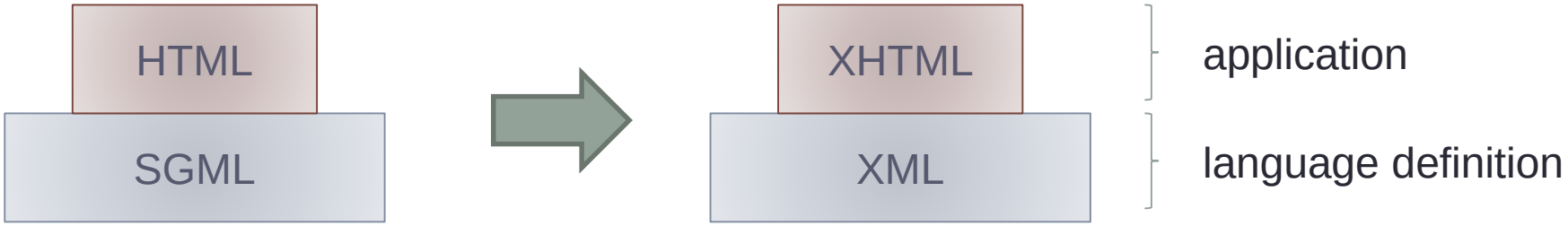
# URL, URN, IRI

- URL (Uniform Resource Locator)
  - location of resource
  - example: http, ftp
- URN (Uniform Resource Name)
  - urn:<nid>:<nss>
  - NID needs to be registered to IANA
  - http://www.iana.org/assignments/urn-namespaces/urn-namespaces.xml
  - 62 URN are registered as of 2017-06-26
  - example: ISBN
- IRI (Internationalized Resource Identifier)
  - Internationalized URI
  - URI cannot use Unicode characters but needs to represent character codes in hex.
  - Allow to use Unicode characters in host name and path.

# What is XML?

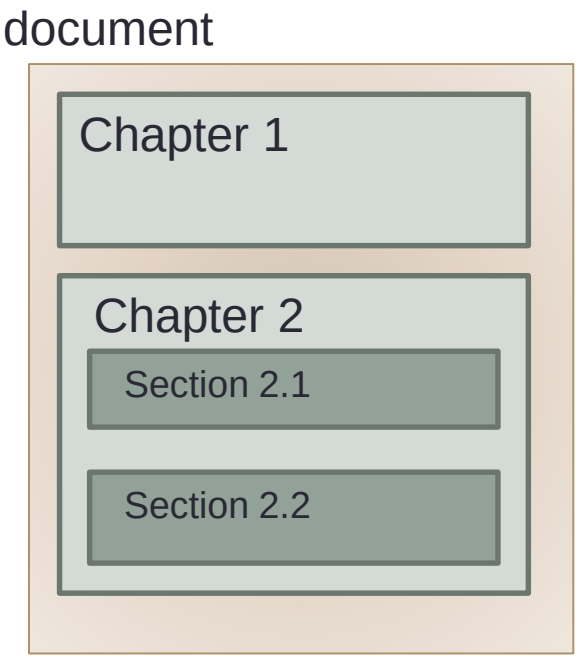
- XML (Extensible Markup Language)
  - Markup Language
  - Version 1.0 is standardized in 1998/2 by W3C.
  - Currently version 1.0 and 1.1 are available.

- Why XML is necessary?
  - SGML is too old to be used for internet documents.
  - HTML cannot be used for all the documents.
  - Documents as well as data.
  - Allow mixing multiple documents.



# Structured Document

- Each document has structure.
  - paragraph, section, chapter, index
  - CV, applications
- markup
  - Specifies document structure
  - Derived from ``marking up''
  - SGML introduced tags as markup.




<coffee price="250">Cafe Latte</coffee>

markup markup

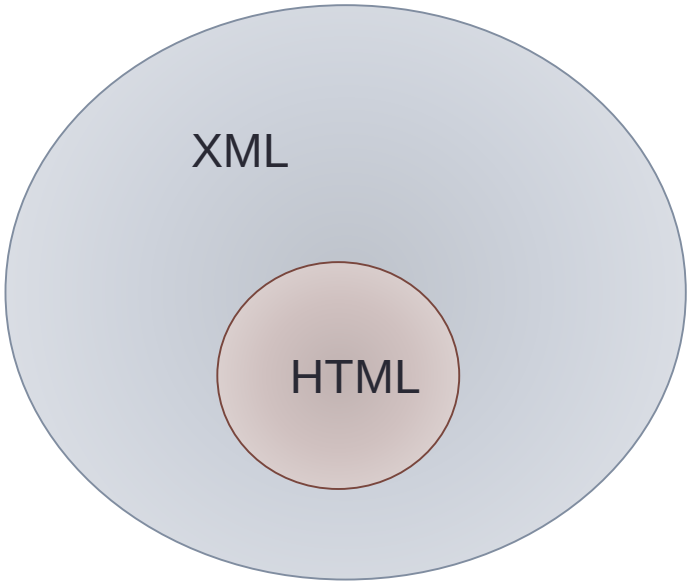
# Structure of an XML Document

- XML declaration
  - XML document
  - specified character code
- Element
  - markup using start and end tags
  - empty element tags
- CharData
  - character data
- Reference
  - character reference like < and A
- CDSect
  - CDATA section
- PI
  - processing instruction
- Comment
  - comment

```
<?xml version="1.0" encoding='EUC-JP'?>
<!-- Keio SFC Restaurant -->
<restaurant>
  <name>Keio Restaurant</name>
  <place>SFC</place>
  <menu>
    <item price="150">coffee</item>
    <item price ="250">cafe latte</item>
    <item price="400">sandwich</item>
    <item price="700">spaghetti</item>
  </menu>
  <open from="10:00" to="17:00" />
  <networks>
    <network type="wireless LAN" />
    <network type="wired LAN" />
  </networks>
  <misc>
    Enjoy international foods.
  </misc>
</restaurant>
```

# HTML

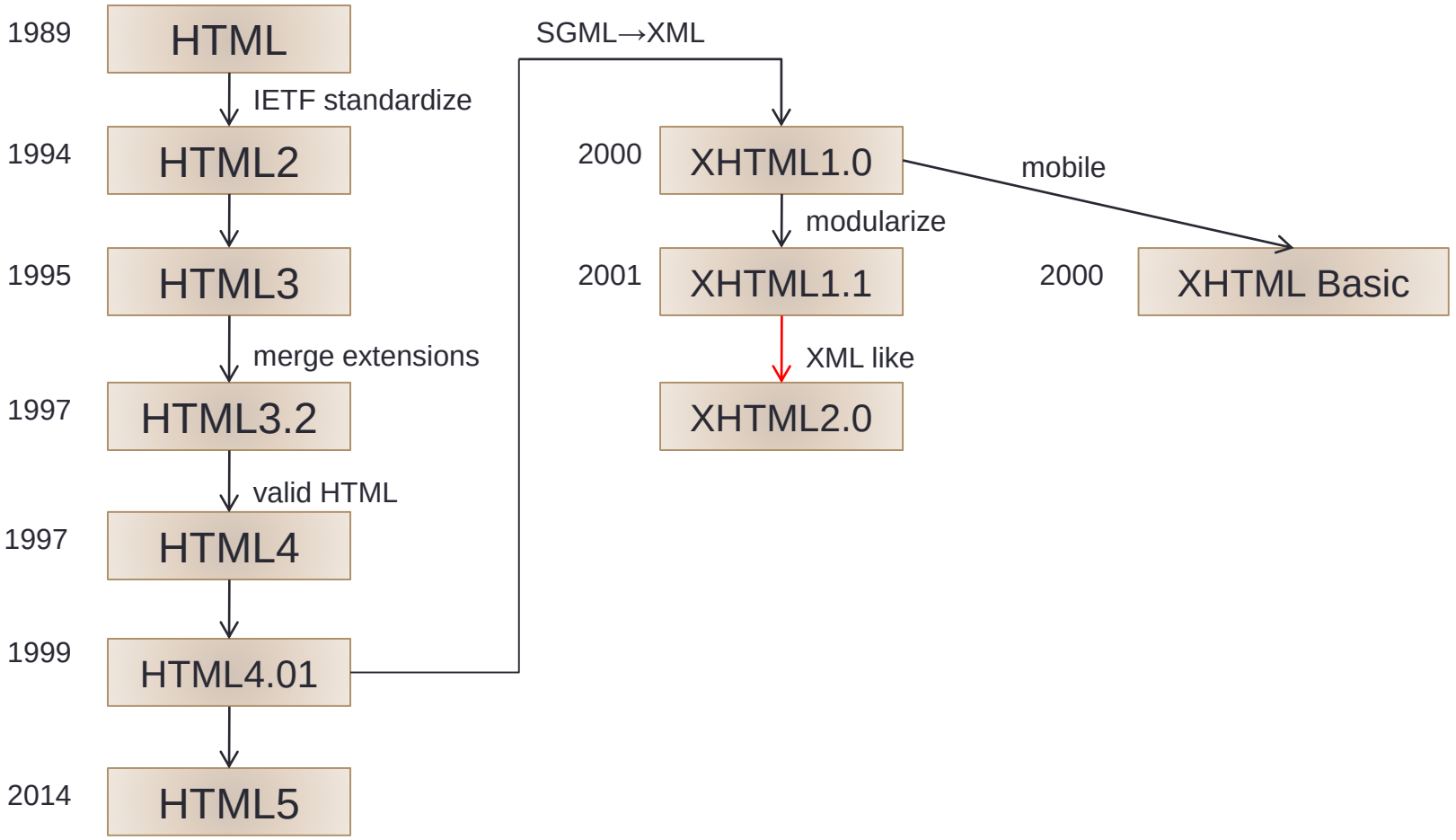
- HTML
  - XML(SGML) application
  - Hypertext document




- HTML features
  - separation of content and presentation
  - CSS for style
  - Combines orthogonal technologies:
    - content: HTML
    - presentation: CSS
    - programming: Javascript

**orthogonal technology** — **movie** — **video**, **audio**, **subtitle**

# HTML Versions



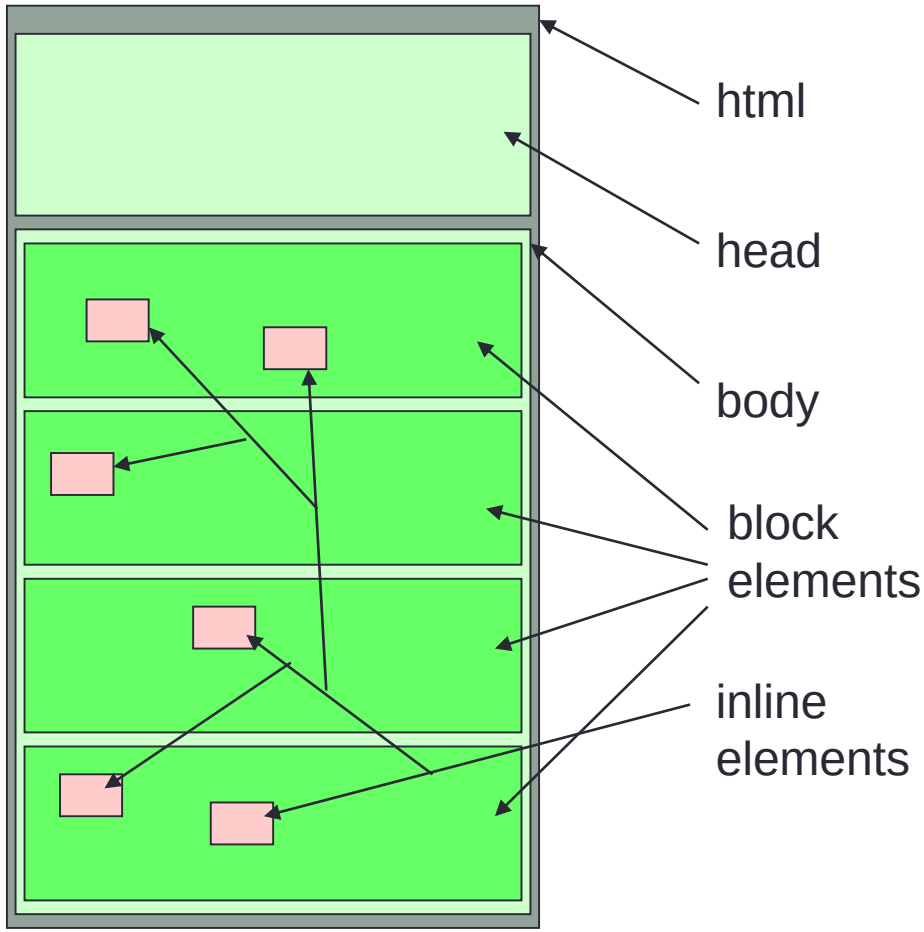
# Simple HTML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
    <title>My first HTML document</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <link rel="stylesheet" href="sample.css" type="text/css"/>
  </head>
  <body>
    <h1>Simple HTML</h1>
    <p>Anybody can write HTML documents.</p>
    <p>There are multiple paragraphs.</p>
  </body>
</html>
```

- DOCTYPE declaration specifies the version.
  - DOCTYPE declaration is too long for beginners.
  - HTML5 simply:
    - <!DOCTYPE html>

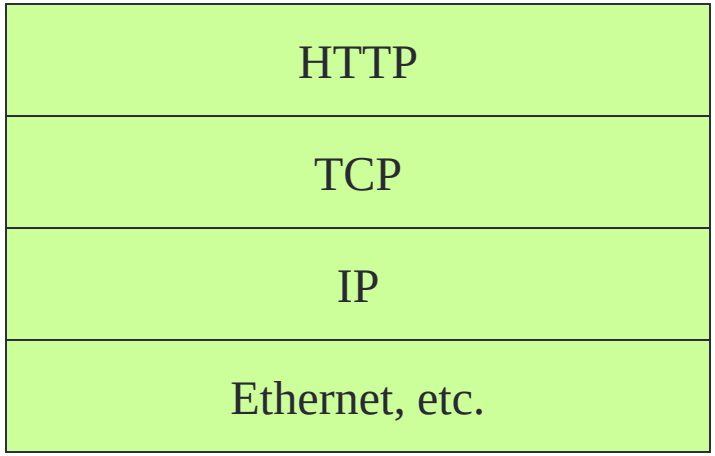
# HTML Elements

- Elements which define the whole document structure
  - html, head, body, etc.
  - section, article, etc.
- Elements to start new paragraph
  - block elements
  - h1, h2, ul, ol, table, etc.
- Elements used in a paragraph
  - inline elements (text elements)
  - i, b, em, strong, etc.



# HTTP

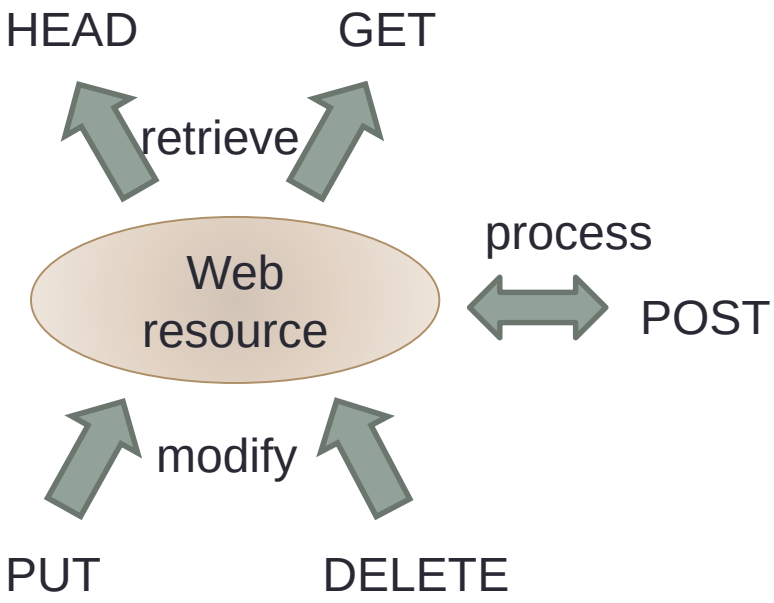
- Hyper Text Transfer Protocol
- TCP protocol
  - port 80
- Simplification of anonymous FTP



# HTTP (Hypertext Transfer Protocol)

- Protocol for manipulating Web resources

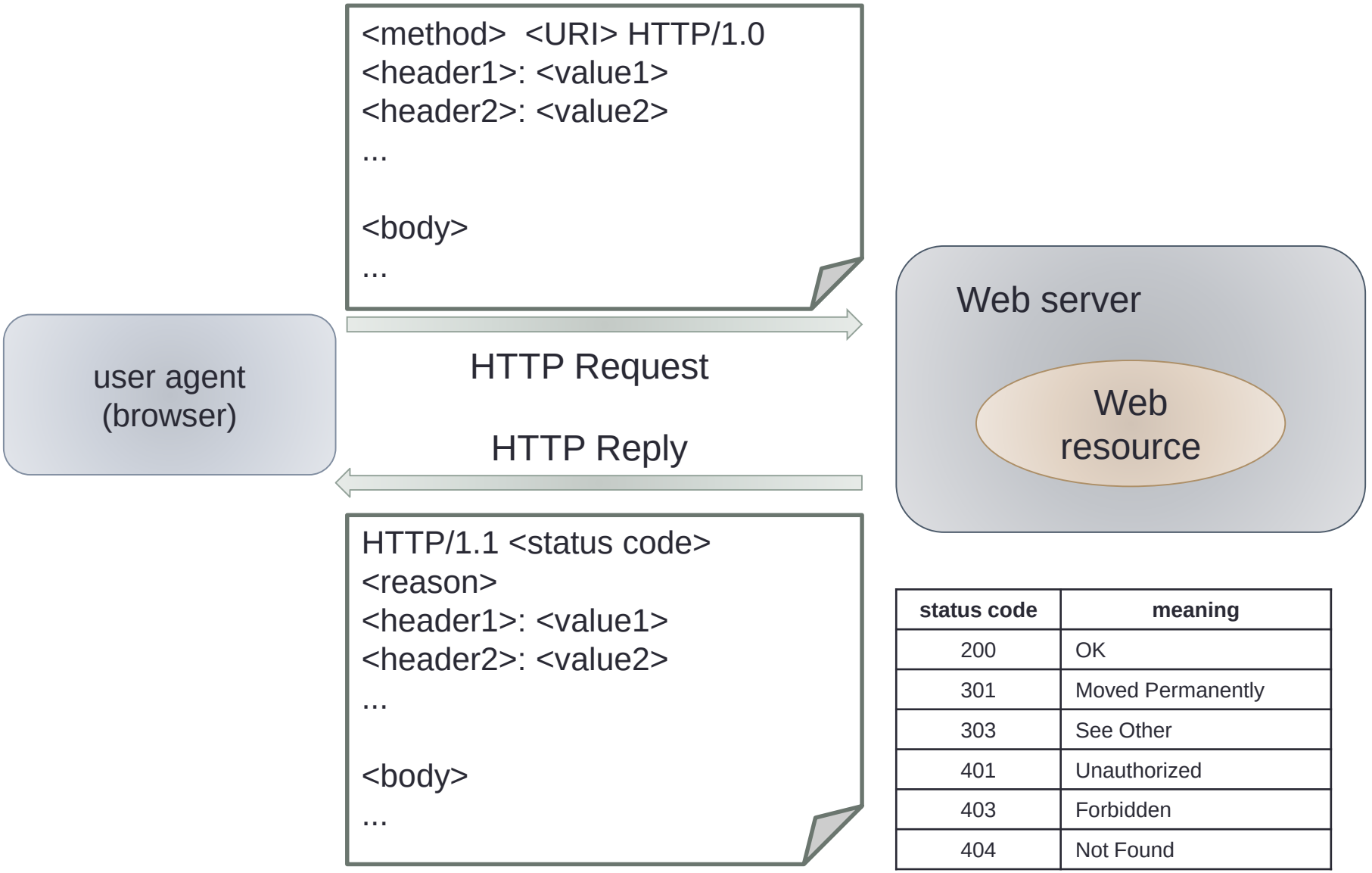
- Methods
  - HEAD
    - get information about a resource
  - GET
    - get a representation of a resource
  - PUT
    - create or modify a resource
  - DELETE
    - delete a resource
  - POST
    - send data to a resource and process it



# HTTP Versions

- HTTP/0.9
  - 1991
  - Request
    - GET <uri> CR LF
  - Reply
    - <html document>
- HTTP/1.0
  - 1992, 1996 RFC1945
  - Request
    - <Method> <uri> HTTP/1.0 CR LF
    - [<request header>*]
    - [CR LF <data>*]
  - Reply
    - HTTP/1.0 <status code> <reason> CR LF
    - [<response header>*]
    - [CR LF <data>*]
- HTTP/1.1
  - 1999/11 RFC2616
    - Virtual Hosting
    - TCP/IP Persistent Connection
    - TCP/IP Pipelining
    - Proxy Cache Control
- HTTP/2
  - 2015/5 RFC7540
    - Extend SPDY developed by Google.
    - Do not replace HTTP/1.1
    - Make HTTP/1.1 faster
    - Multiplexing connection
    - Flow control
    - Header compression
    - Server push

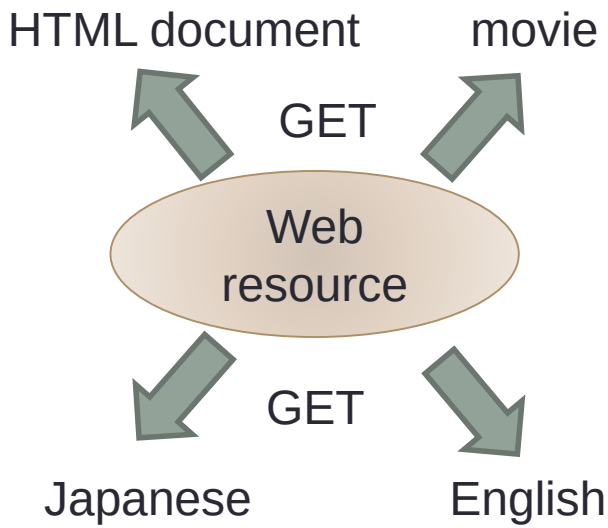
# HTTPのRequest and Reply




| status code | meaning |
|---|---|
| 200 | OK |
| 301 | Moved Permanently |
| 303 | See Other |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

# GET and HEAD Methods

- GET method
  - Get one representation of a Web resource.
  - Content negotiation
  - Language negotiation

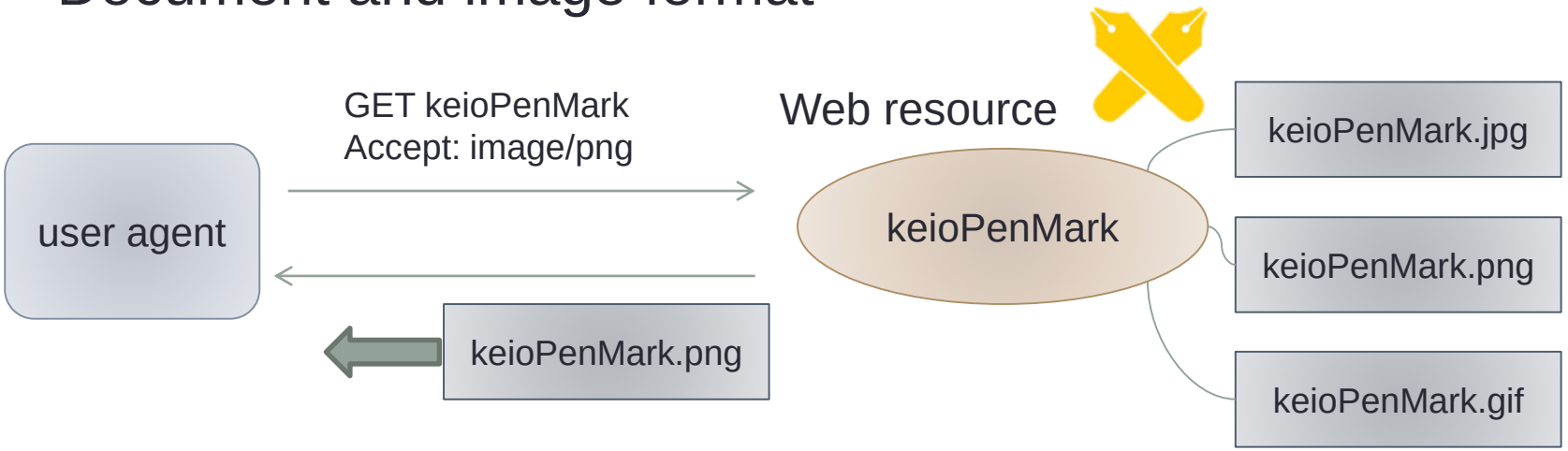



- HEAD method
  - Get information about a Web resource or its representation
  - Subset of GET

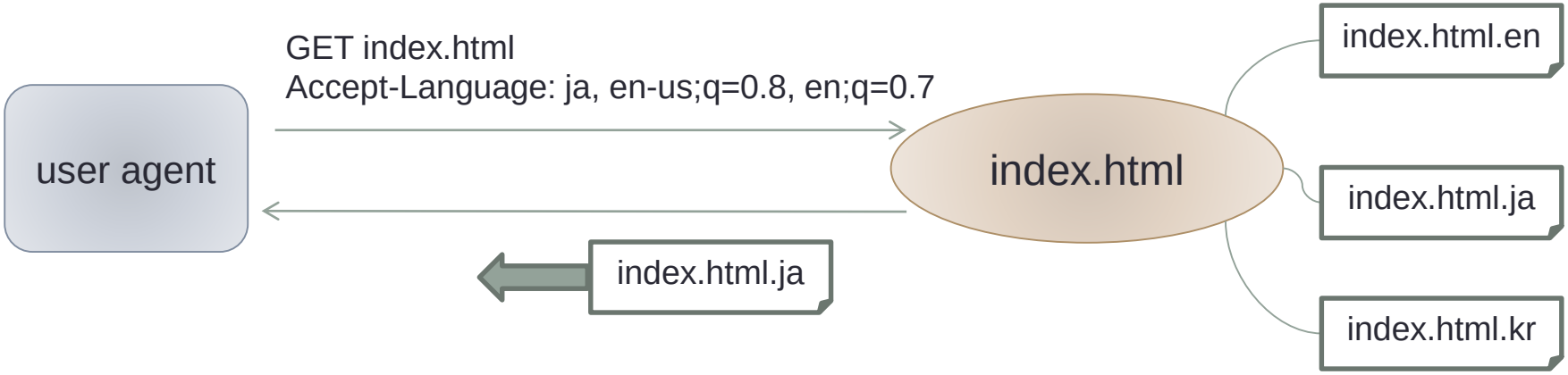
- Property of GET
  - GET is safe to use multiple times.
  - GET is idempotent.
  - GET has no side effect.

# Content Negotiation

- Document and image format

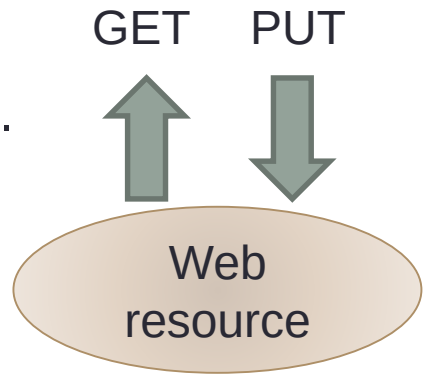



- Language format



# PUT and POST Methods

- PUT method
  - Create a new resource or modify an existing one.
  - Inverse of GET
  - Web browser does not use PUT.




- POST method
  - Send data to a resource.
  - The resource process the data.
  - HTML FORM uses POST.


- GET vs POST
  - Specify GET or POST by the method attribute of FORM.
  - When there is no side effect, use GET.
  - If the resource is updated or there is a side effect, use POST.
  - POST is not idempotent.

# HTTP Header

- General Header
  - Date
  - Pragma
- Request Header
  - Authorization
  - From
  - If-Modified-Since
  - Referer
  - User-Agent
- Response Header
  - Location
  - Server
  - WWW-Authenticate
- Entity Header
  - Allow
  - Content-Encoding
  - Content-Length
  - Content-Type
  - Expires
  - Last-Modified
- Additional Header
  - Accept
  - Accept-Charset
  - Accept-Encoding
  - Accept-Language
  - Content-Language
  - Link
  - MIME-Version
  - Refer-After
  - Title
  - URI

# HTTP Response Code

| response code | meaning |
|---|---|
| 200 | OK |
| 201 | Created |
| 202 | Accepted |
| 204 | No Content |
| 301 | Moved Permanently |
| 302 | Moved temporarily |
| 303 | See Other |
| 304 | Not Modified |
| 305 | Use Proxy |

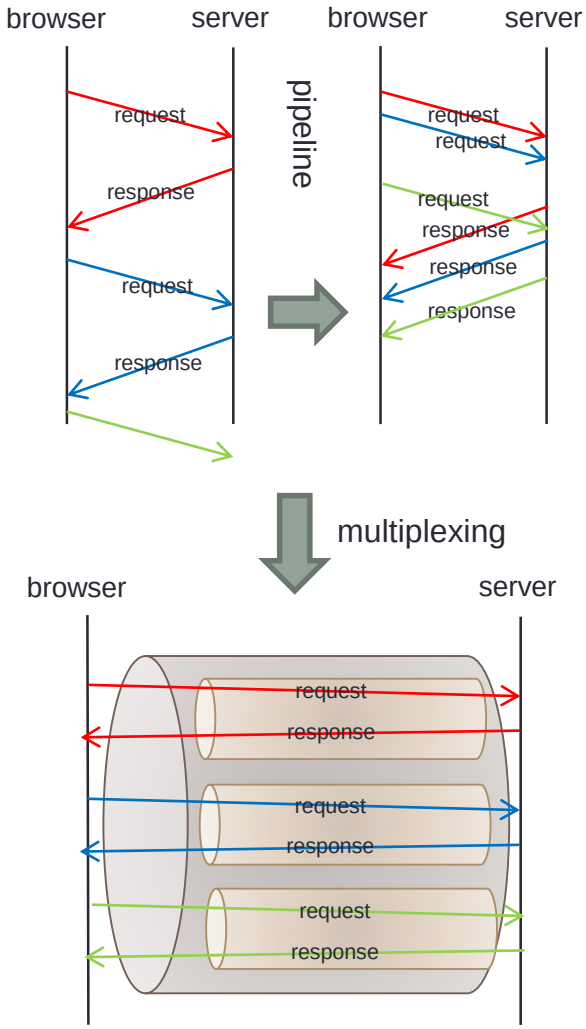
| response code | meaning |
|---|---|
| 400 | Bad Request |
| 401 | Unauthorized |
| 402 | Payment Required |
| 403 | Forbidden |
| 404 | Not Found |
| 405 | Method Not Allowed |
| 500 | Internal Server Error |
| 501 | Not Implemented |

# Other HTTP Features

- Forwarding
  - The resource is moved to a different location.
- Authentication
  - User name and password authentication
  - Basic or Digest authentication
- Virtual Host
  - Support multiple Web sites on one server.
  - Use DNS aliases.
- Effective use of TCP/IP
  - Persistent connection (keep-alive)
  - Pipeline
- Control proxy cache
  - max-age
  - no-cache
  - public or private
- Extension to WebDAV
  - COPY, MOVE, LOCK, UNLOCK

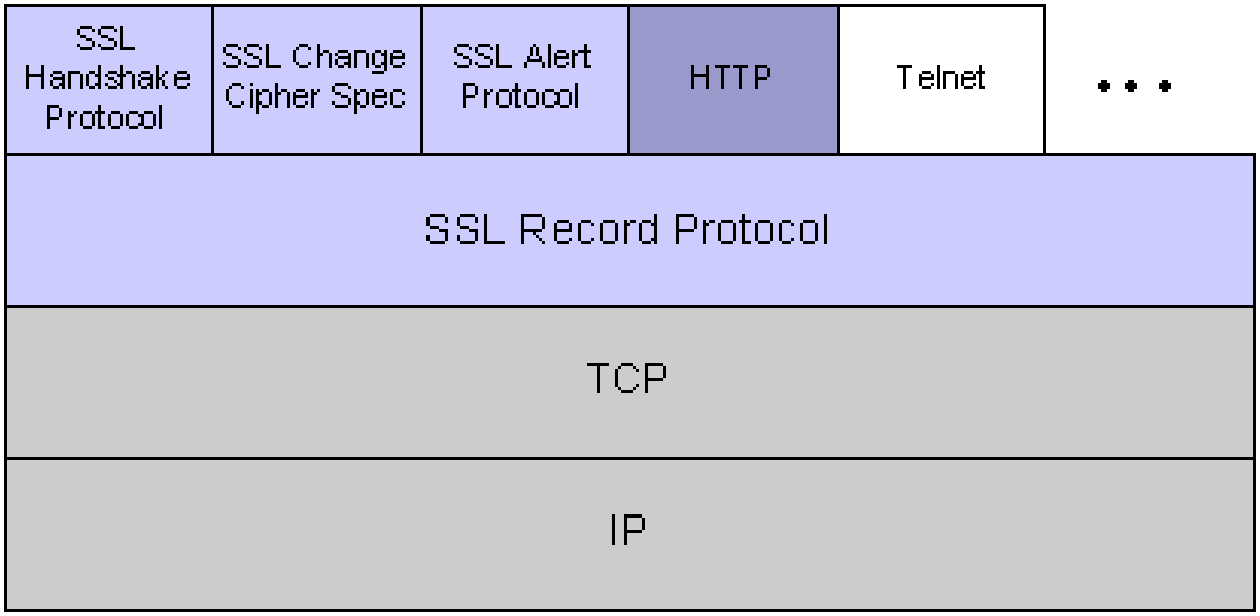
# HTTP/2

- 2015/5 RFC7540
  - Based on SPDY proposal by Google
- Features
  - Do not replace HTTP/1.1
    - Make HTTP/1.1 faster.
  - Multiplexing
    - Pipeline processes requests one by one.
    - Multiplexing can process requests depending on priority.
  - Flow control
    - Avoid overflow of receiver buffer.
  - Header compression
    - Compress header by HPACK
  - Server push
    - Server may inject data to browser cache beforehand.



# HTTPS

- HTTP on top of SSL (Secure Sockets Layer).
  - Host authentication
  - Encrypt communication




http://www.modssl.org/docs/2.8/ssl_intro.html

# Summary

- Web
  - history
  - URL
  - XML
  - HTML
  - HTTP
- Future topics:
  - Search engines (Page Rank Algorithm)
  - Web site design (Information Architecture)
  - Web data (Semantic Web, Linked Data)

# Final Exam

- Area
  - Class 1 (OS) to 12 (World Wide Web)
- Date & Time
  - July 17th, 2018,  from 9:30am for about 30min
- Condition
  - The answer need to be written on a mark sheet paper, so please bring an HB or B pencil (and a eraser).
  - You may bring in anything except electric devices like PC, mobile phones.