



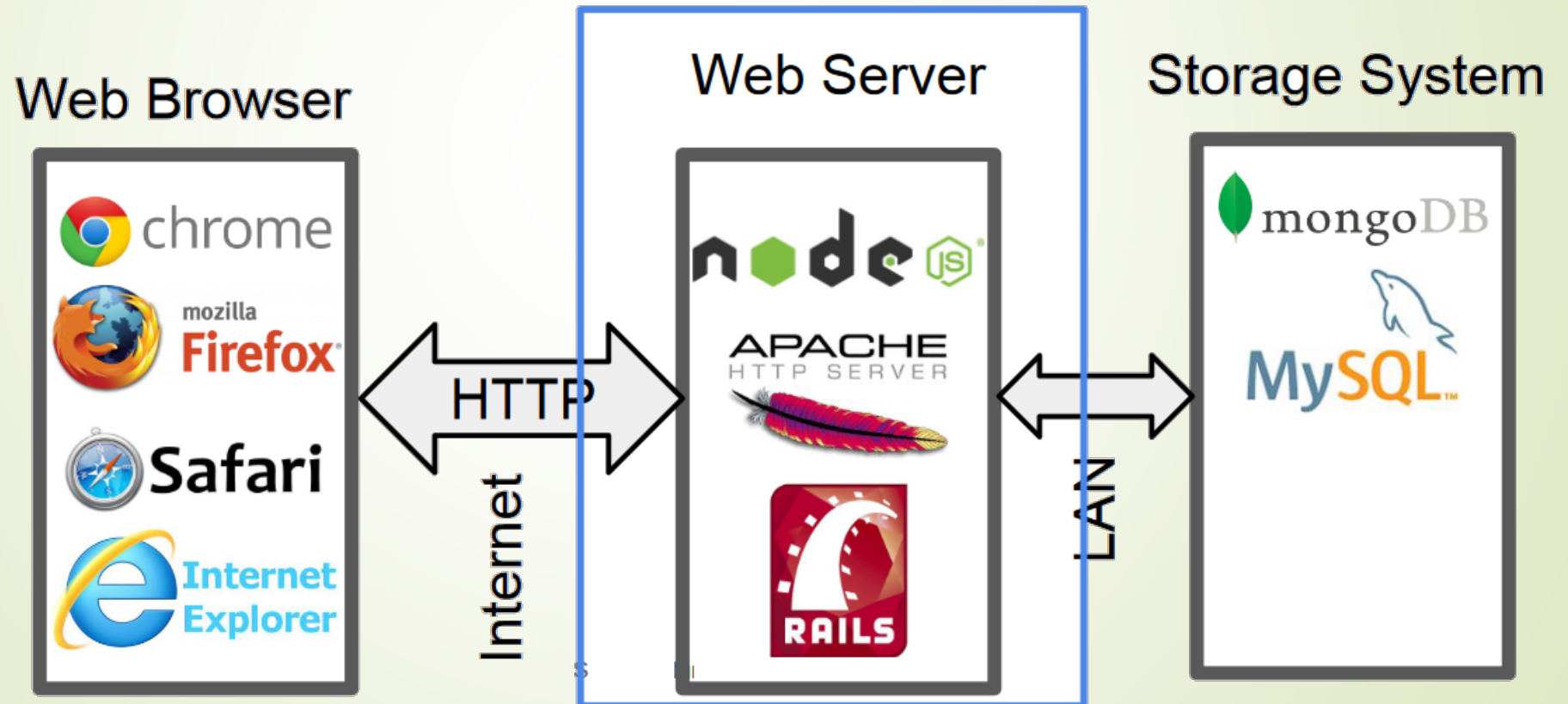
# 1 Basic Technologies

**Web Server, Server-side Website Programming**

### III.1

# Fundamental Concepts

## Web Application Architecture



# Fundamental Concepts

## Web Servers

- Browsers speak HTTP and Web Servers speak HTTP
  - Browsers: send HTTP request and get HTTP responses
  - Web Server: get HTTP requests and send HTTP responses
- HTTP is layered on TCP/IP so a web server:
  - loop forever doing:
    - accept TCP connection from browser
    - read HTTP request from TCP connection
    - process HTTP request
    - write HTTP response to TCP connection
    - shutdown TCP connection (except if Connection: keep-alive)

# Fundamental Concepts

## Web Servers – RESTful Web Services

- REST is an architectural style, not standard
- designed for distributed systems to address architectural properties such as performance, scalability, simplicity, modifiability, visibility, and portability
- REST architectural style is defined by
  - client-server autonomy
  - stateless interactions
  - service-oriented
    - resources as URI
    - operations as HTTP-methods
    - exchanging representations of resources

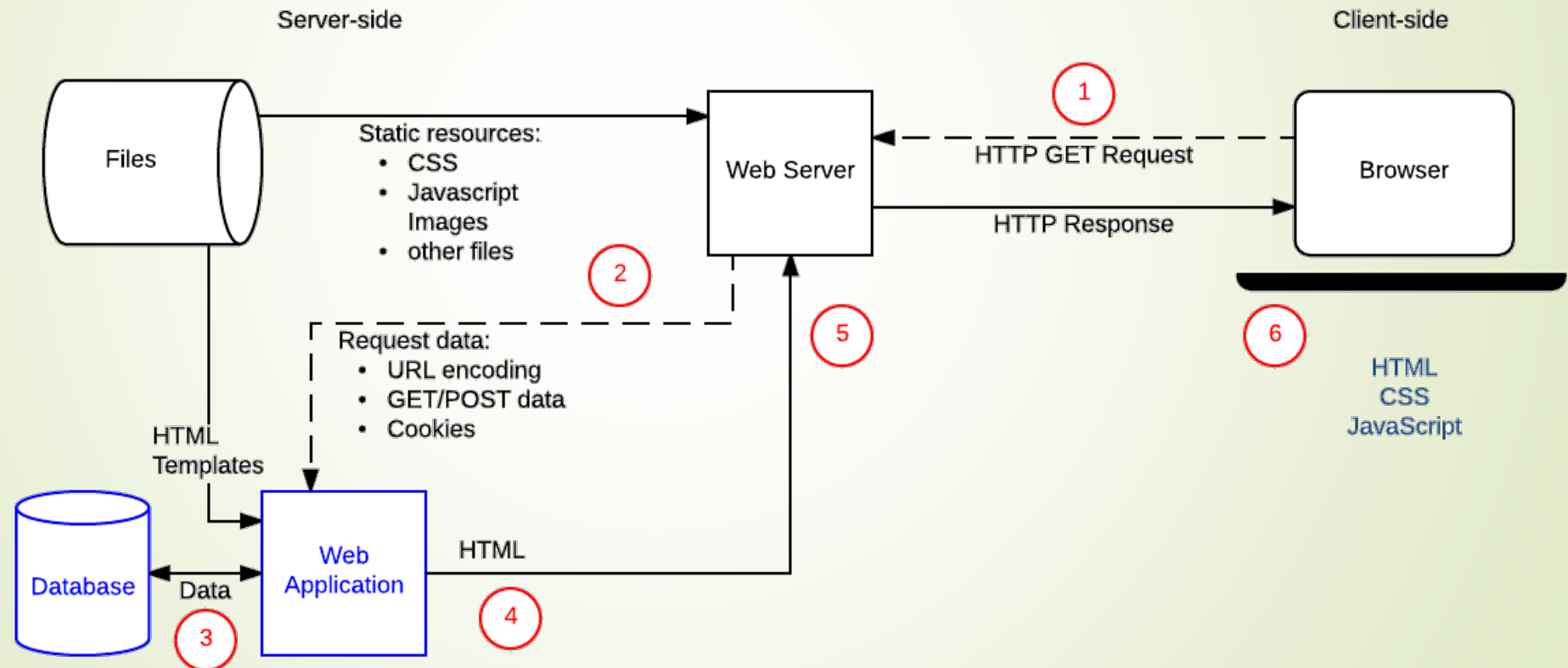
# Fundamental Concepts

## Server-side Website Programming

- allows to efficiently deliver information tailored for individual users and thereby create a much better user experience
- efficient storage and delivery of information
- customized user experience
- controlled access to content
- store session/state information
- notifications and communication

# Fundamental Concepts

## Server-side Website Programming





# Fundamental Concepts

## Server-side Web Frameworks

- Web frameworks provide tools and libraries to simplify common web development operations
- work directly with HTTP requests and responses
- route requests to the appropriate handler

### Flask

```
@app.route("/")  
def hello():  
    return "Hello World!"
```

### Django

```
urlpatterns = [  
    url(r'^$', views.index),  
    url(r'^best/(?P<team_name>\w.+?) /$',  
        views.best)  
]
```

# Fundamental Concepts

## Server-side Web Frameworks

- respond to incoming requests by running additional logic
  - specify which HTTP method is being handled
  - specify what URL routes are being handled
  - read an object representing the parsed HTTP request, which will contain fields with all the headers, other metadata, and body contents
  - interact with an object representing the in-progress HTTP response, which will contain fields and methods to help generate the final response contents
  - make it easy to access data in the request
  - abstract and simplify database access
  - rendering data



# Fundamental Concepts

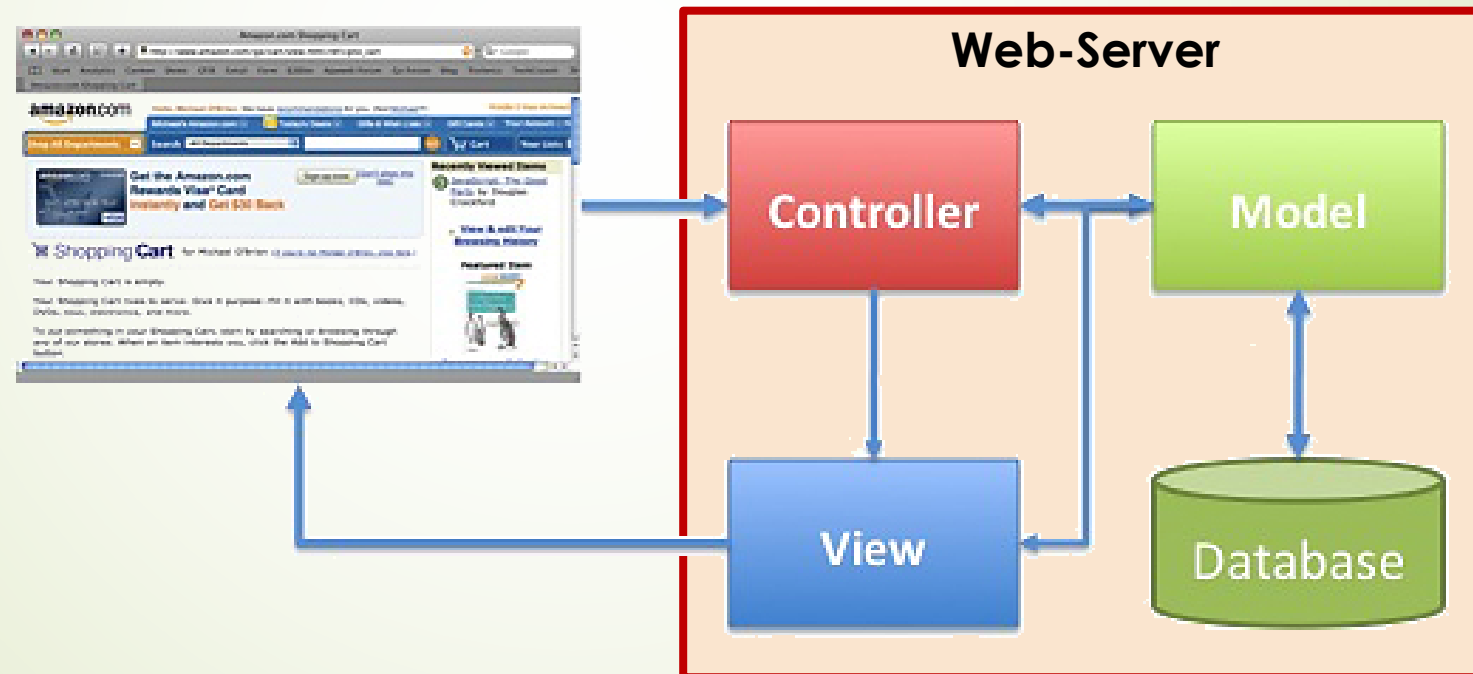
## Server-side Web Frameworks

- frameworks implement some form of middleware, which are individual chunks of logic that are combined together
  - check the URL and HTTP method
  - check cookies for a session ID and look up a user's details internally
  - read query parameters, extract data from a request body
  - connect to a database to retrieve information
  - update the database based on the calculations or the request contents
  - run some calculations
  - construct an HTML document or a JSON data structure
  - send that content back as the response body

# Fundamental Concepts

## Server-side Web Frameworks

- separation of concerns: Model View Controller Framework



# Fundamental Concepts

## Server-side Web Frameworks

