

# Lecture 11 – AJAX

## Web Application Development

February 17, 2015

Jeffrey L. Eppinger  
Professor of the Practice  
School of Computer Science

# Lecture Schedule – 1<sup>st</sup> Half

(subject to change)

#1 Intro	#9 Forms & Templates
#2 HTML & CSS	#10 Files & Images
#3 JavaScript & DOM	#11 AJAX
#4 CSS Frameworks	#12 jQuery
#5 HTTP & Django	#13 Databases
#6 Cookies & Sessions	#14 Cloud Deployment
#7 Django Models	#15 SSL
#8 Transactions	#16 Project Proposals

# Agenda

→ Course Administration

AJAX

JSON, XML

# Administrative Issues

- TA Office Hours
  - Moved to NSH 3001
  - Special: tonight from 6:30pm to 10:00pm
- HW#4 Grades
  - We are late – I hope to post grades tomorrow
- HW#5 due last night
  - Last penalty-free late day is tomorrow
- HW#6 posted tonight
  - Due Monday
- Project Proposals
  - Also due on Monday
  - We'll have time in class on Thurs to discuss concerns

# Agenda

- ✓ Course Administration

- AJAX

- JSON, XML

# What is AJAX?

- Asynchronous JavaScript And XML
  - Term coined by Jesse James Garret in 2005
    - Says it's *not* short for Asynchronous JavaScript & XML
  - Term captures use of existing technology
  - Gmail used these technologies in 2004

# AJAX is Not a New Language

- AJAX is a technique for creating more interactive web applications
  - Use an XMLHttpRequest object to make requests to the webserver for data asynchronously (or synchronously)
  - Receive server data as XML (or text or JSON)
  - Convert the XML into a DOM tree
  - Extract data from the XML DOM tree and change the HTML document's DOM tree (thereby updating the page)
- Program the above using JavaScript

# AJAX Advantages

- More interactive web sites
  - The page is not reloaded
    - You only change parts of the page
  - Requests can be executed asynchronously
    - So the user can continue to interact with the page during requests to the server
- Reduces load on server
  - A lot of the formatting of the result is offloaded to the client browser



# Objects in JavaScript

- You can use the “new” operator

- This will create an object

- Example:

```
var user = new userObject("mjs","sylvia");  
function userObject(userid, password) { ... }
```

- You can set properties on the object

- Example: `this.userid = userid;`

- Properties can be functions => instance methods

- Example: `this.salt = computeSalt;`

- Where: `function computeSalt() { ... }`

# XMLHttpRequest

- It's a native object
  - Except in IE, where it's an ActiveX object
- Important Event
  - `onreadystatechange`
- Important Methods
  - `open(method, url, async)`
  - `setRequestHeader(label, value)`
  - `send(content)`
- Important Properties
  - `readyState` (0 = before open, 4 = response received)
  - `status` (200 = OK, 404 = not found, etc)
  - `responseXML`
  - `responseText`

# Asynchronous Requests

- XMLHttpRequest
  - Requests can be sent asynchronously (or synchronously)
  - When the state of the request changes, your JavaScript method is called

# Example GET Request

Add refresh capability to shared to do list:

- `todo.js`
  - Uses XMLHttpRequest to obtain current item list
    - Uses GET method
    - Makes asynchronous call
  - Results returned in JSON
  - Updates the HTML DOM tree with results

# JSON

- JavaScript Object Notation
- “Language Independent”
- Self-describing serialization format
  - Data: number, strings, booleans, null
  - Objects (aka dictionaries): { "id": 5, text: "Sleep" }
  - Arrays: [ 1, 2, 3, 4 ]
  - Arrays of Objects, etc
- Check out W3 Schools

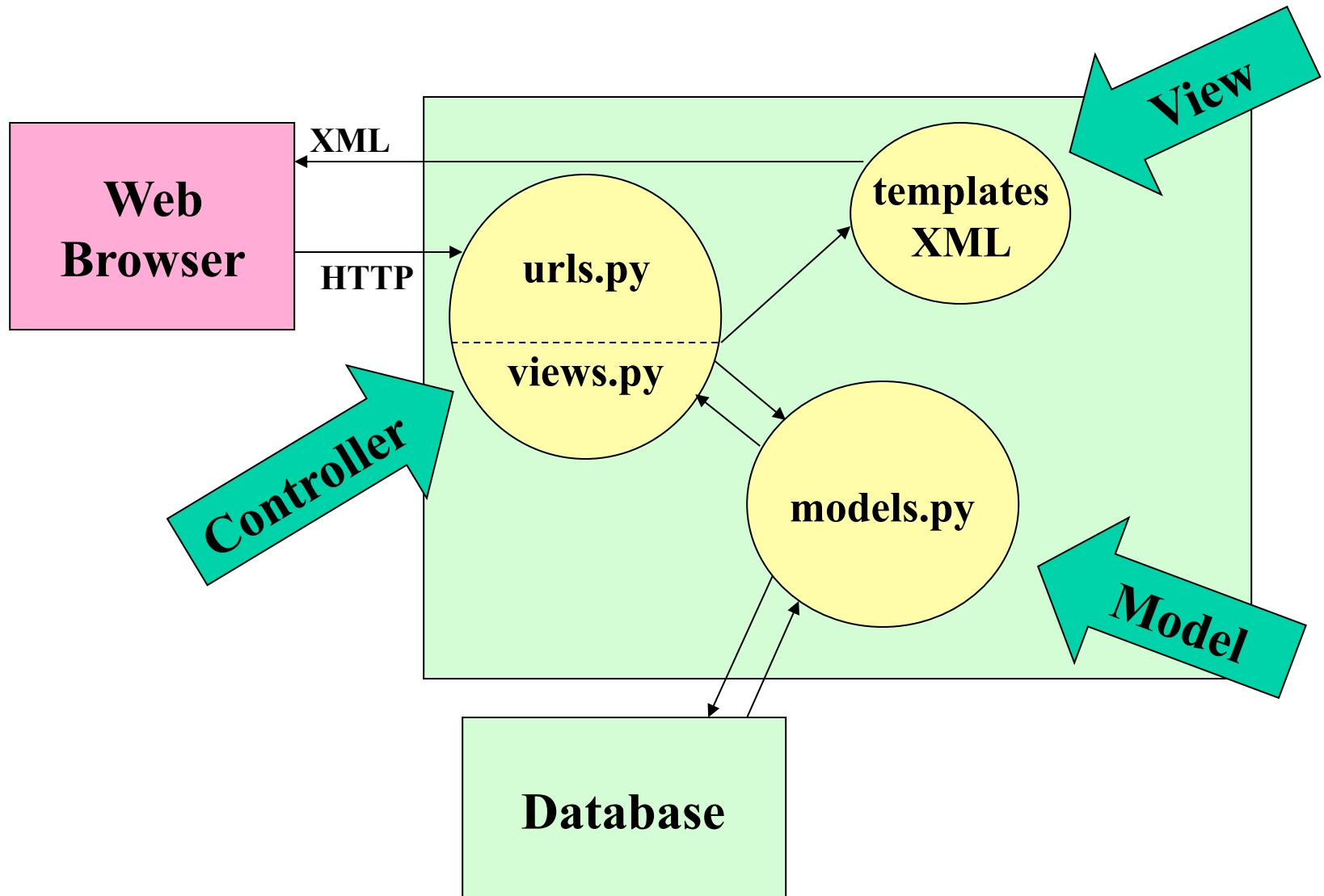
# XML

- eXensible Mark-up Language
- More language independent than JSON
- Less tightly integrated with JavaScript
- Specify parsing of XML in DOCTYPE
- To use this in JavaScript, use `reponseXML`
  - This will return data a DOM tree
  - Use same search primitives to access your data

# Other things

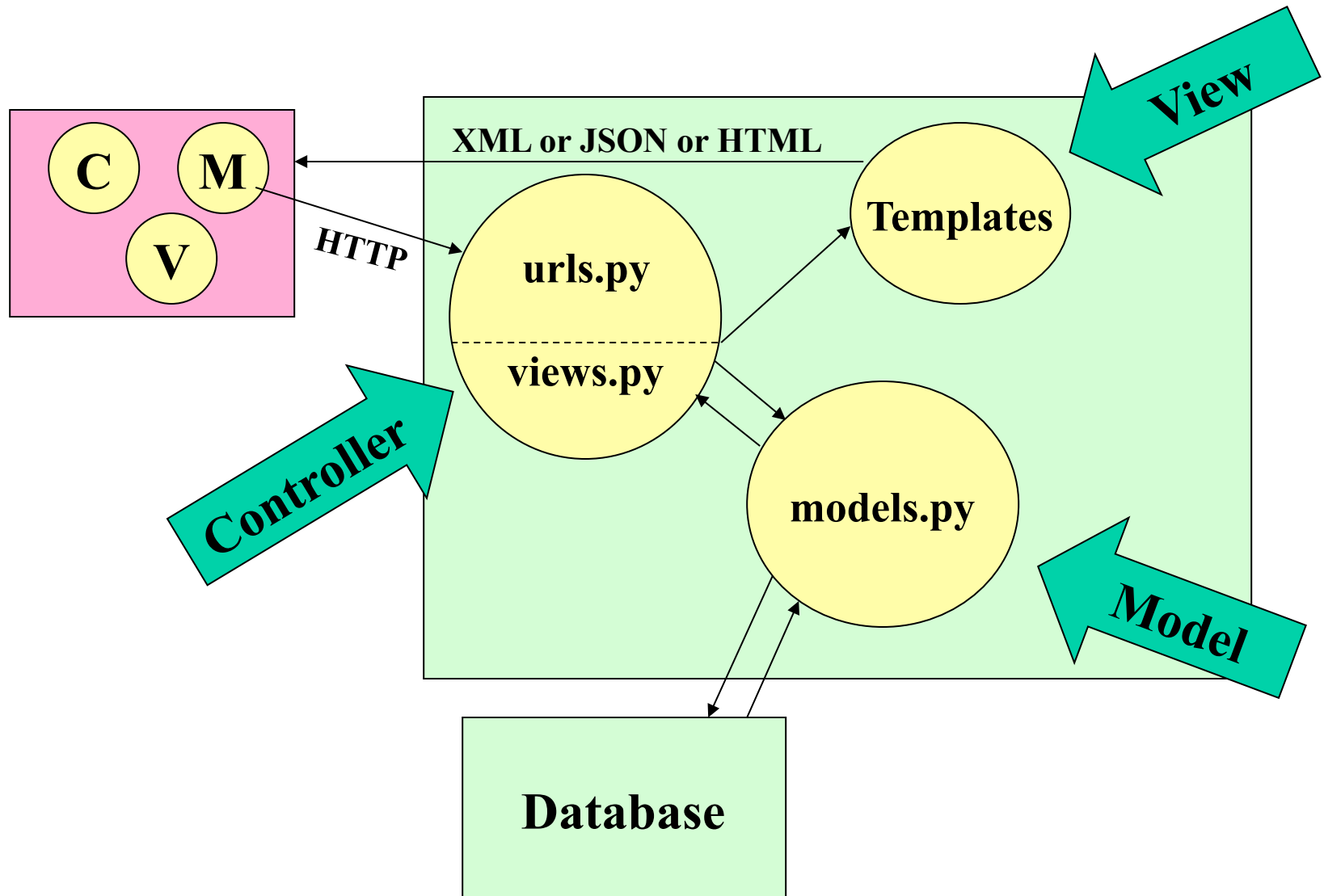
- You can have multiple requests concurrently executing
  - Just be sure to use multiple XMLHttpRequest objects
  - This example, uses only one outstanding request at a time
- POST Method
  - Should be use when updating the server

# Model-View-Controller Architecture



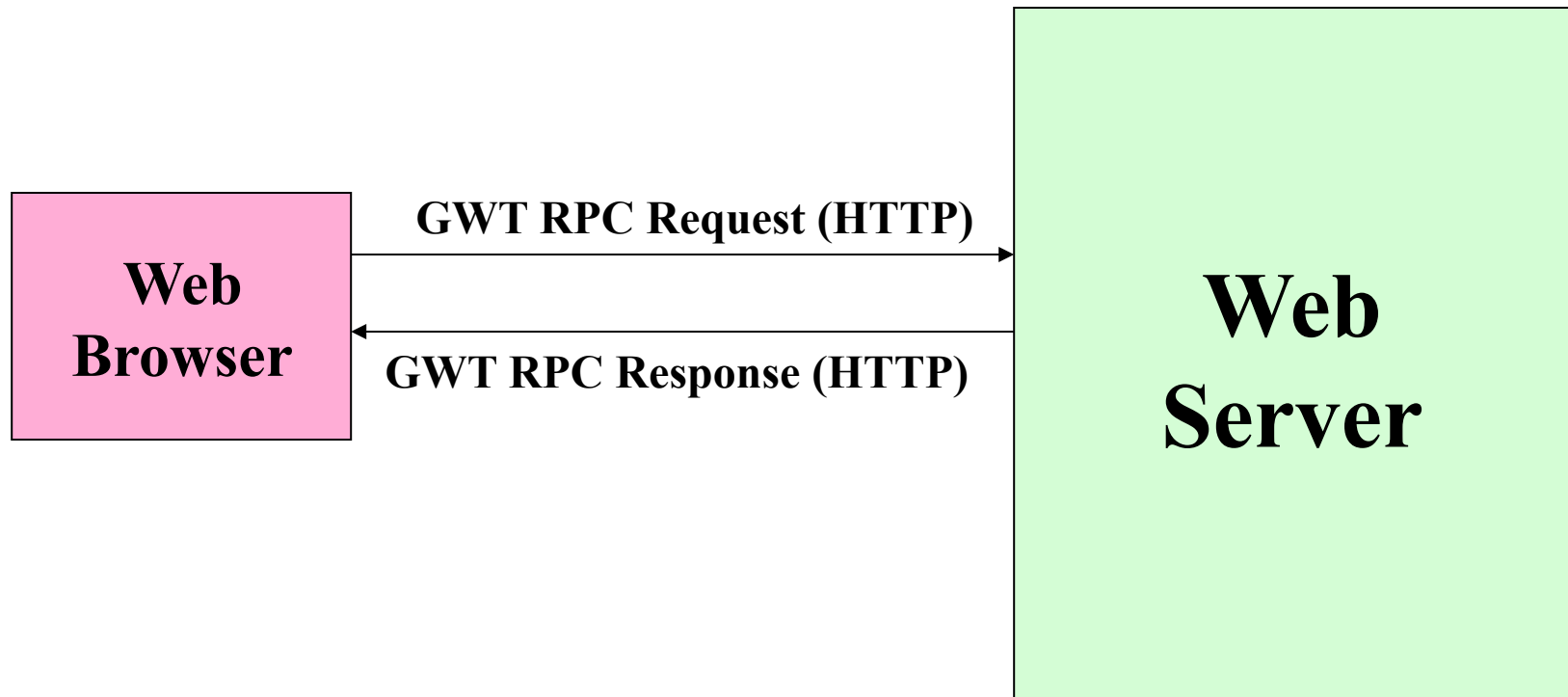


# $(MVC)^2$



# Google Web Toolkit

- App built and debugged in Java using Eclipse
- App is then cross-compiled into JavaScript for deployment



# Additional AJAX Technologies

- JavaScript Libraries to make it easier
  - JQuery – we will demonstrate this next class

# Agenda

- ✓ Course Administration
- ✓ AJAX
  - ✓ JSON, XML

# Next Lecture

- jQuery
  - This will be a short lecture
- After lecture
  - Time to meet with each other find project partners
  - Time to ask me questions about your project ideas