Introducing the MEAN stack

CS252

The MEAN stack

- MongoDB: mongodb.org
- Express: expressjs.com
- AngularJS: angularjs.org
- Node: nodejs.org

- All free and open-source
- All based on JavaScript

The LAMP stack

OS	Linux
Web Server	Apache
Database	MySQL
Server-side language	PHP

The Microsoft stack

OS	Microsoft
Web Server	IIS
Database	SQL Server
Server-side language	C#/.NET

Many moving parts

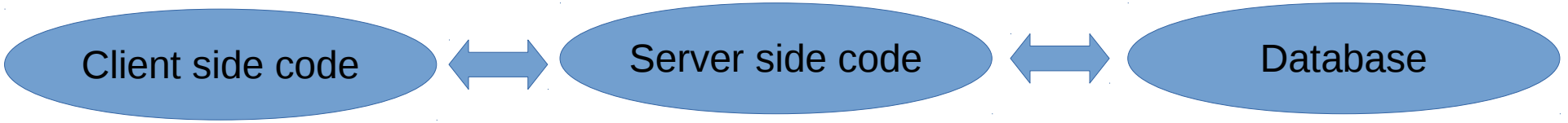
- HTML
- CSS
- JavaScript
- PHP
- MySQL
- JSON
- How do we get everything to work together?

Typical workflow

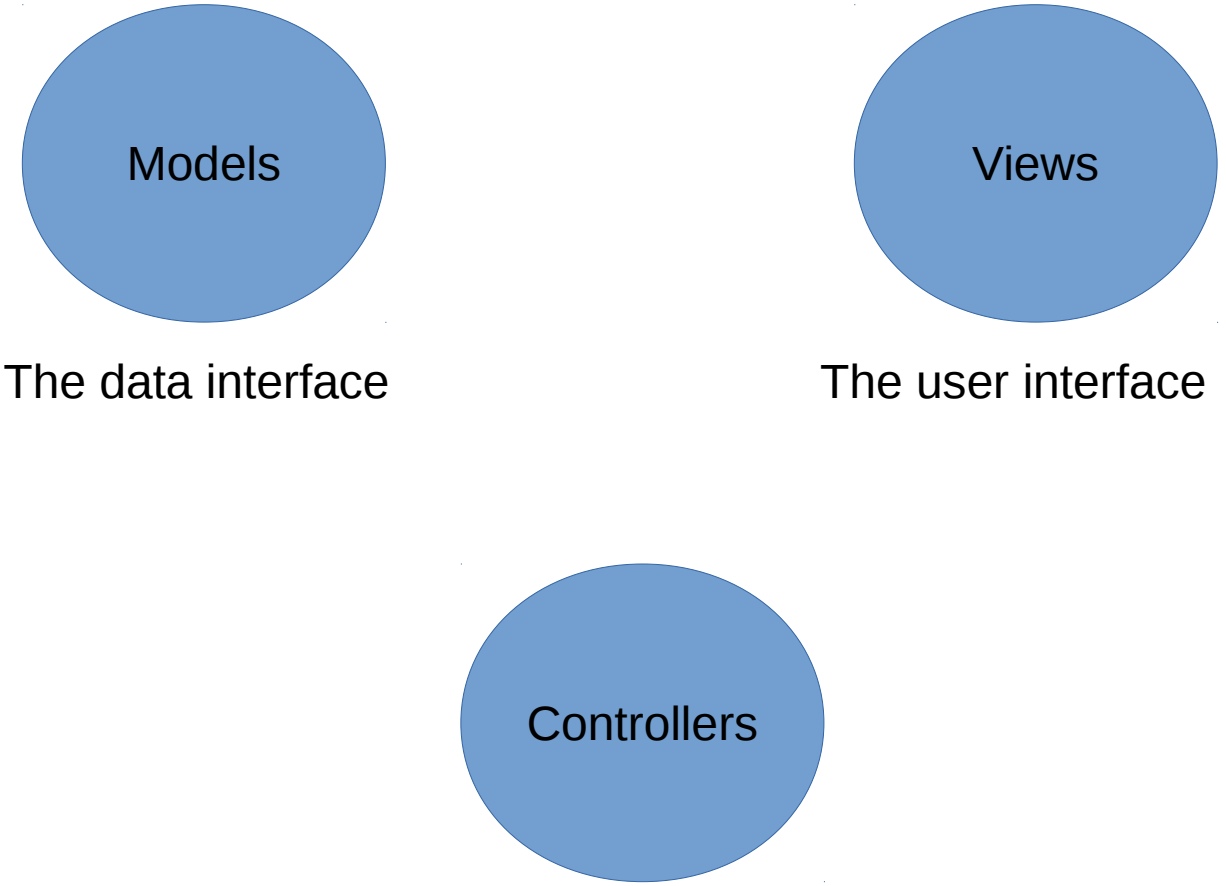
- Code mainly runs on the server
- Exception: JavaScript
- Data is mainly stored on server
- Data processing can happen either on server or at client
    - Server side processing = greater security
    - Client side processing = greater computing efficiency

Typical workflow

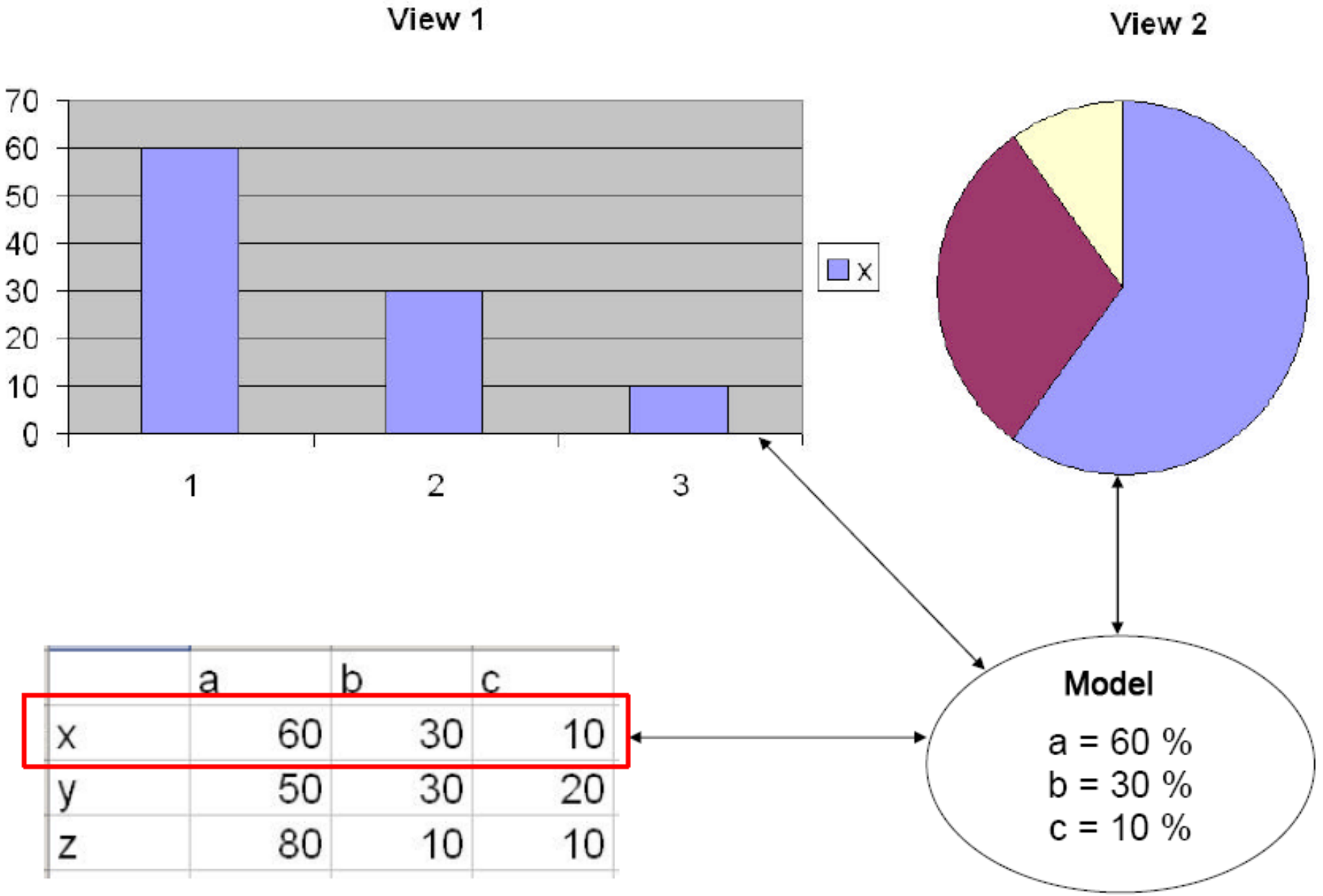
- Code mainly runs on the server
- Exception: JavaScript
- Data is mainly stored on server
- Data processing can happen either on server or at client
    - Server side processing = greater security
    - Client side processing = greater computing efficiency



The MVC abstraction



MVC – general example



The MEAN stack compared

	LAMP	Microsoft	MEAN
OS	Linux	Windows	Typically Linux
Web Server	Apache	IIS	Node
Database	MySQL	SQL Server	MongoDB
Server-side scripting	PHP	C#/.NET	JavaScript
Server-side MVC			Express
Client-side MVC			Angular

MEAN stack is JavaScript all the way through

Very fast and responsive; naturally async

Mongo

- We saw this last week
- Schemaless document storage in JSON format
- Supports indexes
- No joins
- Blazing fast

How to use?

- db.users.insert({name: ‘some_name’})
- db.users.find({name: ‘some_name’})
- db.users.find({age: {$gt: 18}}, {name: 1, address: 0}).limit(5)
- db.users.distinct(“name”)
- Very powerful concept: Aggregation
  - https://docs.mongodb.com/manual/aggregation/

Aggregation

- db.collection.aggregate([... multiple aggregation steps ....]);
- db.collection.aggregate([$match: {“pin”: 208016}])
- db.collection.aggregate([$match: {“pin”: 208016}, {$group: {_id: null, count: {$sum: 1}}}]);
- Data processing pipeline where you can chain analysis commands
- Should master – much better than map-reduce
- Great resource: education.mongodb.com

Node

- Network-enabled wrapper around Google’s V8 JavaScript engine
- Provides ready-to-run network services and command line tools
- Bundled with node package manager
- Node universe has packages for just about any network-related service you can think of
  - Just install using npm

Node usage

- Install from nodejs.org or Linux repo
- Run syntax *node somefile.js*
- Node manual: nodejs.org/api/

Express

- Server-side MVC framework for Node
- Event pipeline for request/response
  - Can chain together different data processing and user interaction activities using middleware
- Uses templating preprocessers to render views
  - Pug, EJS, CoffeeScript etc.

Middleware

- A function that does something in the request/response pipeline
  - Authentication
  - Parsing
  - Routing
  - Error handling
- Lots of third party middlewares are compatible with Express
- Read more about them at expressjs.com/api

Express usage

- Install *express-cli* using *npm*
- Usage syntax
  - *express myapp*
  - *cd myapp*
  - *npm install*
  - *npm start*
  - *<view in browser>*

In lab next week (after mid sems)

- Clone my demo MEAN app from https://github.com/nisheeths/express-demo-app
- Get it to work on your machine
- Modify it to visualize responses to the queries I asked in the MongoDB lab

Class logistics

- The SQL and MongoDB projects using PHP are now Assignment 2
- Submission deadline: 23rd September
- Ionic demo on 27th September
- Project deadline: end of September
- Lab attendance policy
  - no attendance for arrivals later than 1415