Course Review

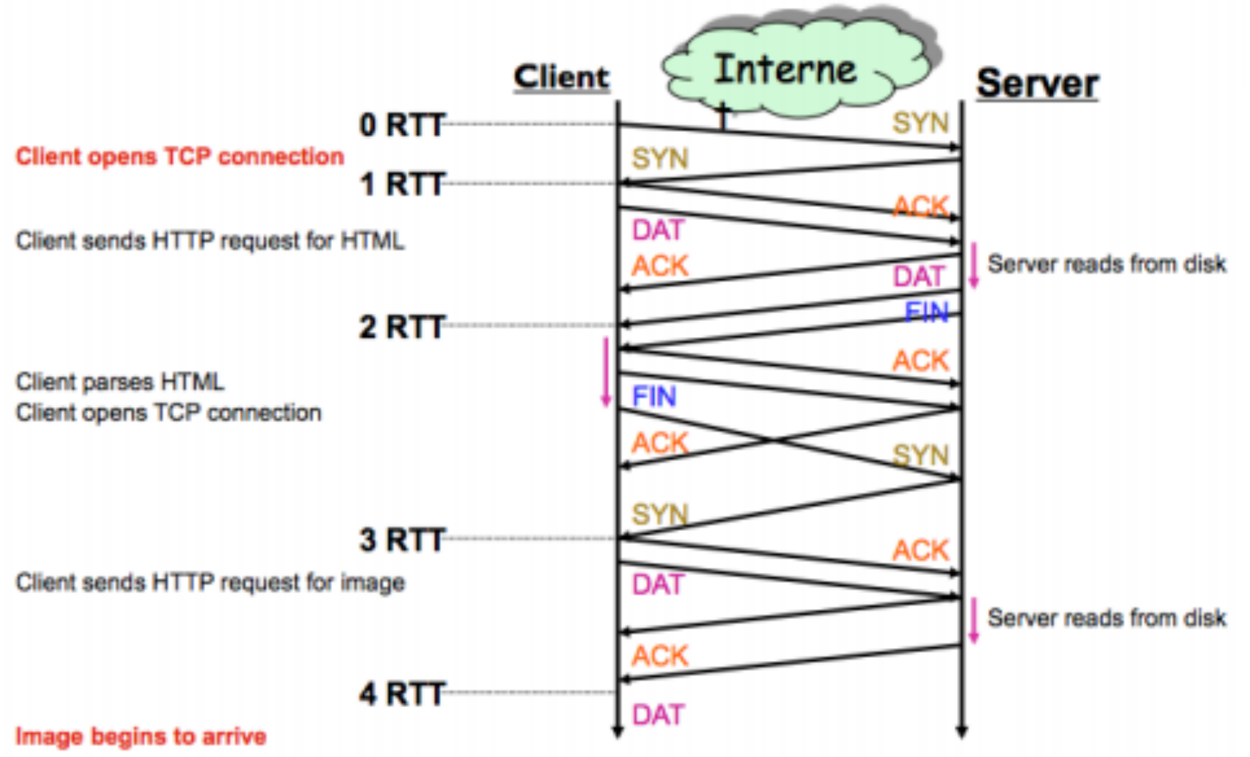
A overview of what we learned this past 3 months.

UNIVERSITY OF TORONTO

HTTP & URL

Hypertext Transport Protocol (HTTP): A set of commands understood by the web servers and sent from a browser.

—**Uniform Resource Locator (URL)**: An identifier for the location of a document on a web site



UNIVERSITY OF TORONTO

REST

"REST is just a set of conventions about how to use HTTP.

Instead of having randomly named setter and getter URLs and using `GET` for all the getters and `POST` for all the setters, we try to have the URLs identify resources, and then use the HTTP actions `GET`, `POST`, `PUT` and `DELETE` to do stuff to them. So instead of

```
GET /get_article?id=1

POST /delete_article id=1
```

You would do

```
GET /articles/1/

DELETE /articles/1/"
```

`http://stackoverflow.com/questions/2191049/what-is-the-advantage-of-using-rest-instead-of-non-rest-http`

REST

Best Practices for designing REST API

→ think about “resources”

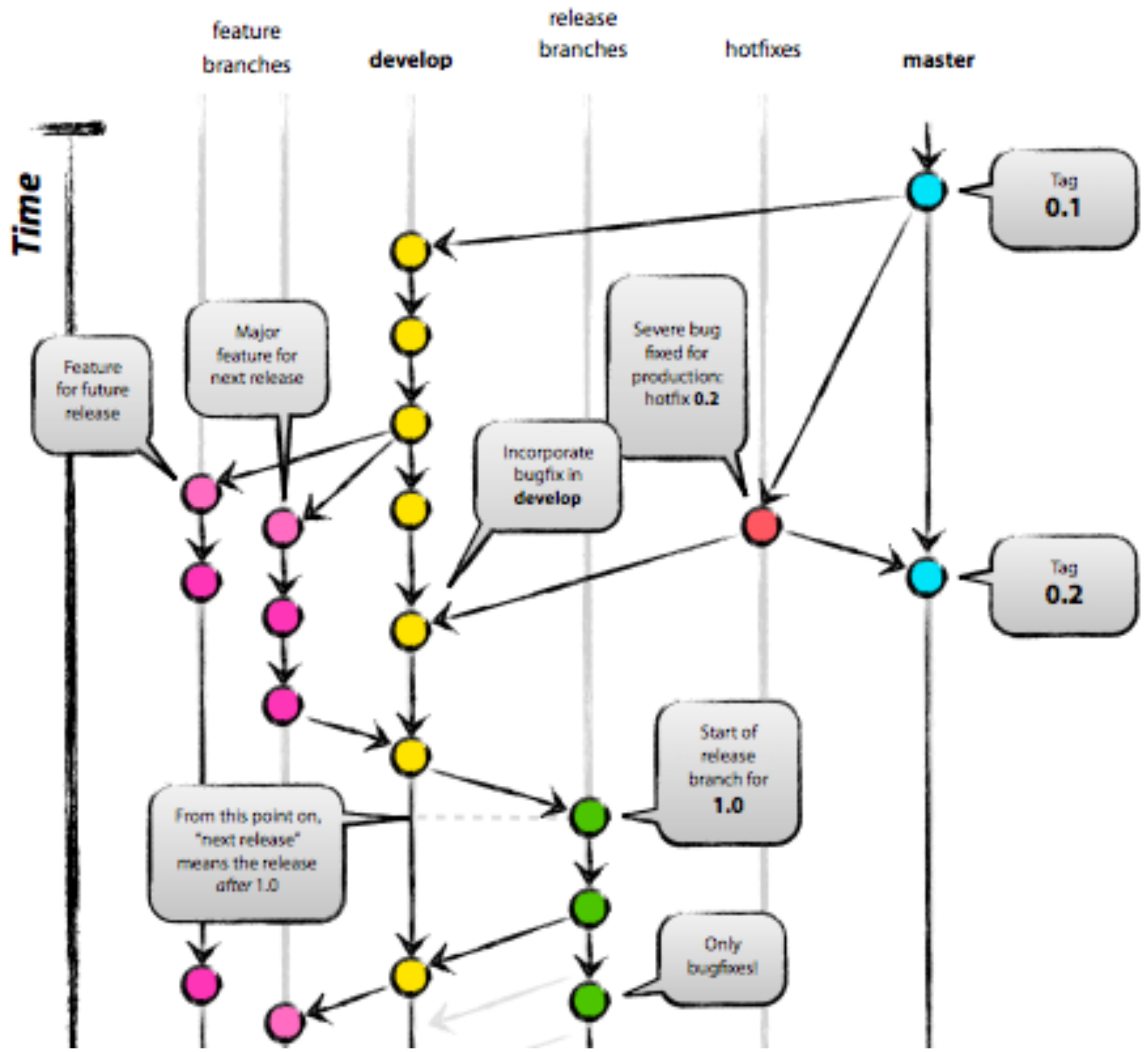
→ elements & collections

→ map out the 4 methods for each

→ Prefer Nouns, Plural, Concrete

→ Use Parameters for more advanced queries

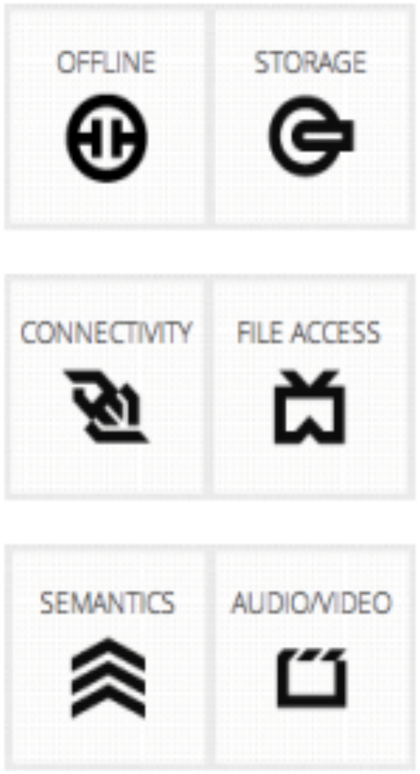
Social Coding, GIT, branches



feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Severe bug fixed for production: hotfix 0.2
Major feature for next release
Feature for future release
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!

HTML5: New Features

- Semantic elements and Markups
- Audio and video support
- Canvas
- Drag and drop
- Local data storage
- Offline applications
- Server events
- Geolocation



UNIVERSITY OF TORONTO

CSS

- HTML specifies document structure
- CSS specifies presentation

Advantage of CSS

→ Precise control over presentation
→ Simplify site maintenance
→ Faster downloads
→ Media-specific rendering

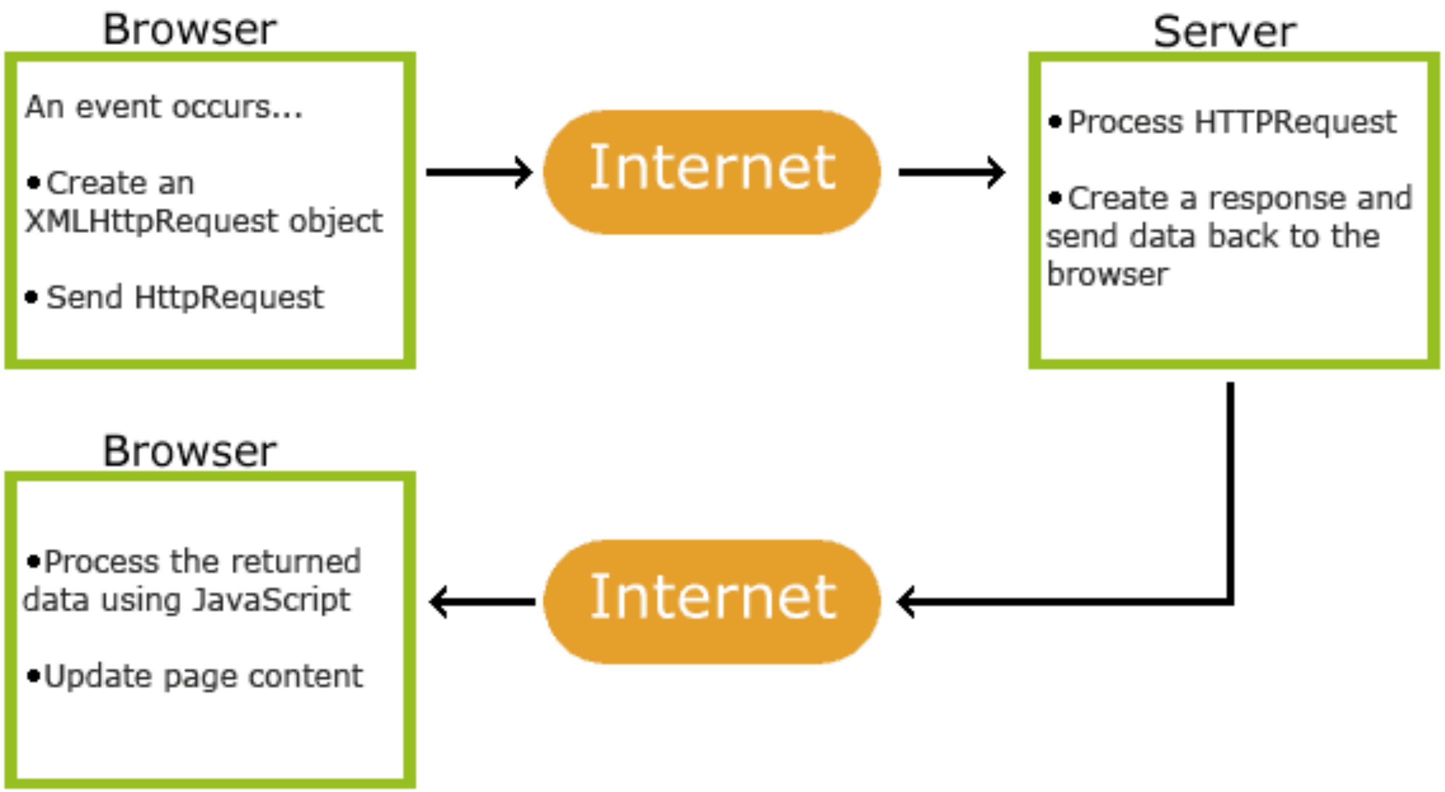
Asynchronous JavaScript and XML

Asynchronous JavaScript and XML (AJAX) is not a programming language, but a new way to use existing standards to exchanging data with a server, and updating parts of a web page - without reloading the whole page.

AJAX uses a combination of

- XMLHttpRequest object to exchange data asynchronously with a server.
- JavaScript/DOM to display/interact with the information.
- XML as a format for transferring data.
- CSS to style the data.

AJAX How it works


Browser
An event occurs...
•Create an XMLHttpRequest object
•Send HttpRequest
Internet
Server
•Process HTTPRequest
•Create a response and send data back to the browser
Internet
Browser
•Process the returned data using JavaScript
•Update page content

UNIVERSITY OF TORONTO

Ajax: JQuery write less, do more

Call a local script on the server /api/getWeather with the query parameter zipcode=97201 and replace the element #weather-temp's html with the returned text.

```
$.ajax({
 url: "/api/getWeather",
 data: {
   zipcode: 97201
 },
 success: function( data ) {
   $( "#weather-temp" ).html( "<strong>" + data + "</strong>
degrees" );
 }
});
```

JQuery write less, do more

```
<script src="http://code.jquery.com/jquery-latest.js"></script>
    <script>
      $(document).ready(function() {
        $('#mo').click(function() {
          $('#mo').append('<p>Clicked!</p>');
        });
      });
</script>
```

- Cross Browser.
- Lots of helpers and utilities.
- Very active community

JQuery UI

jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library.

JQuery Mobile

A Touch-Optimized Web Framework

jQuery Mobile is a HTML5-based user interface system designed to make responsive web sites and apps that are accessible on all smartphone, tablet and desktop devices.

Responsive Web Design

Responsive Web design (RWD) is a Web design approach aimed at crafting sites to provide an optimal viewing experience.

UNIVERSITY OF TORONTO

Frameworks

B

Foundation

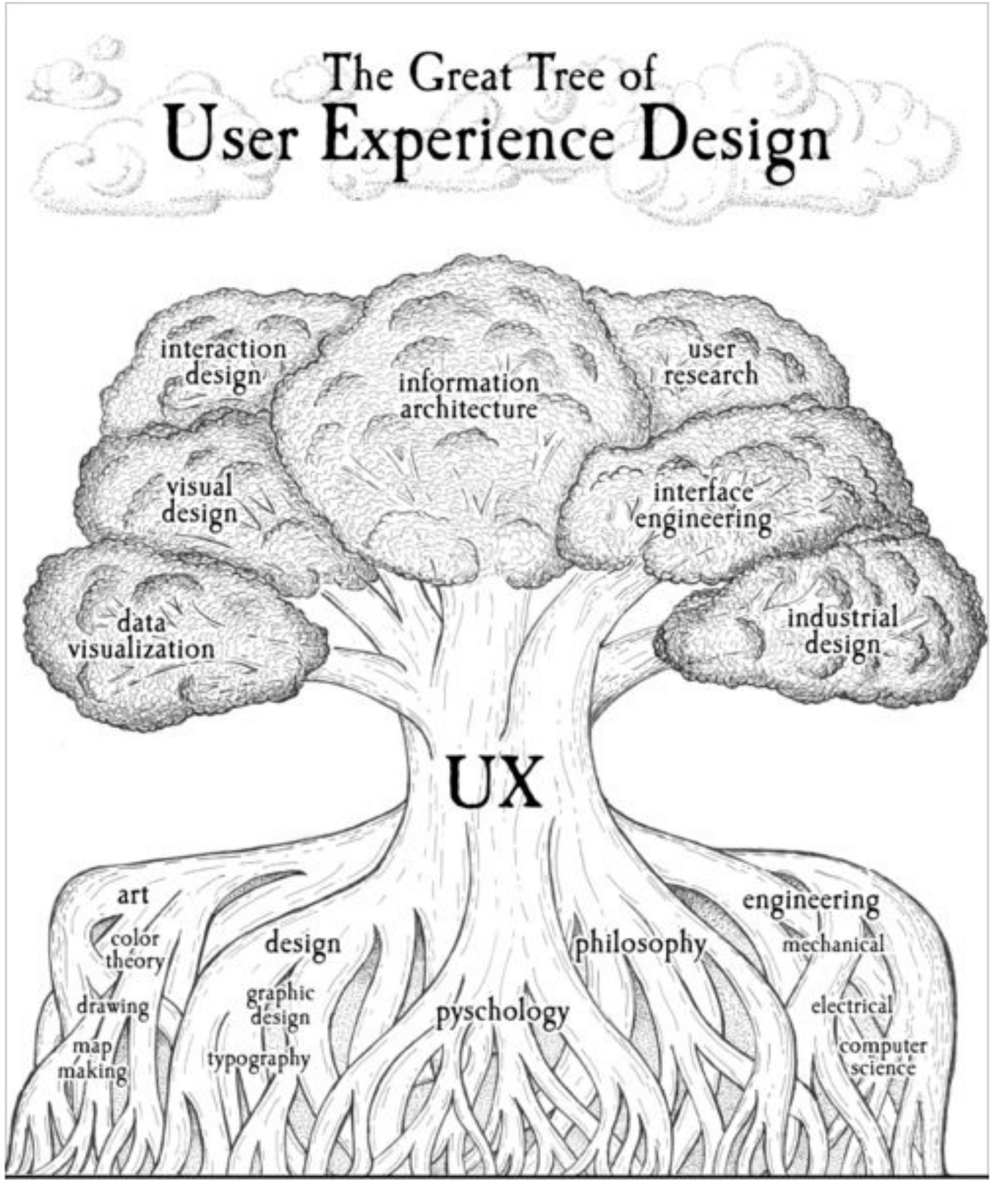
Less Framework 4

Skeleton

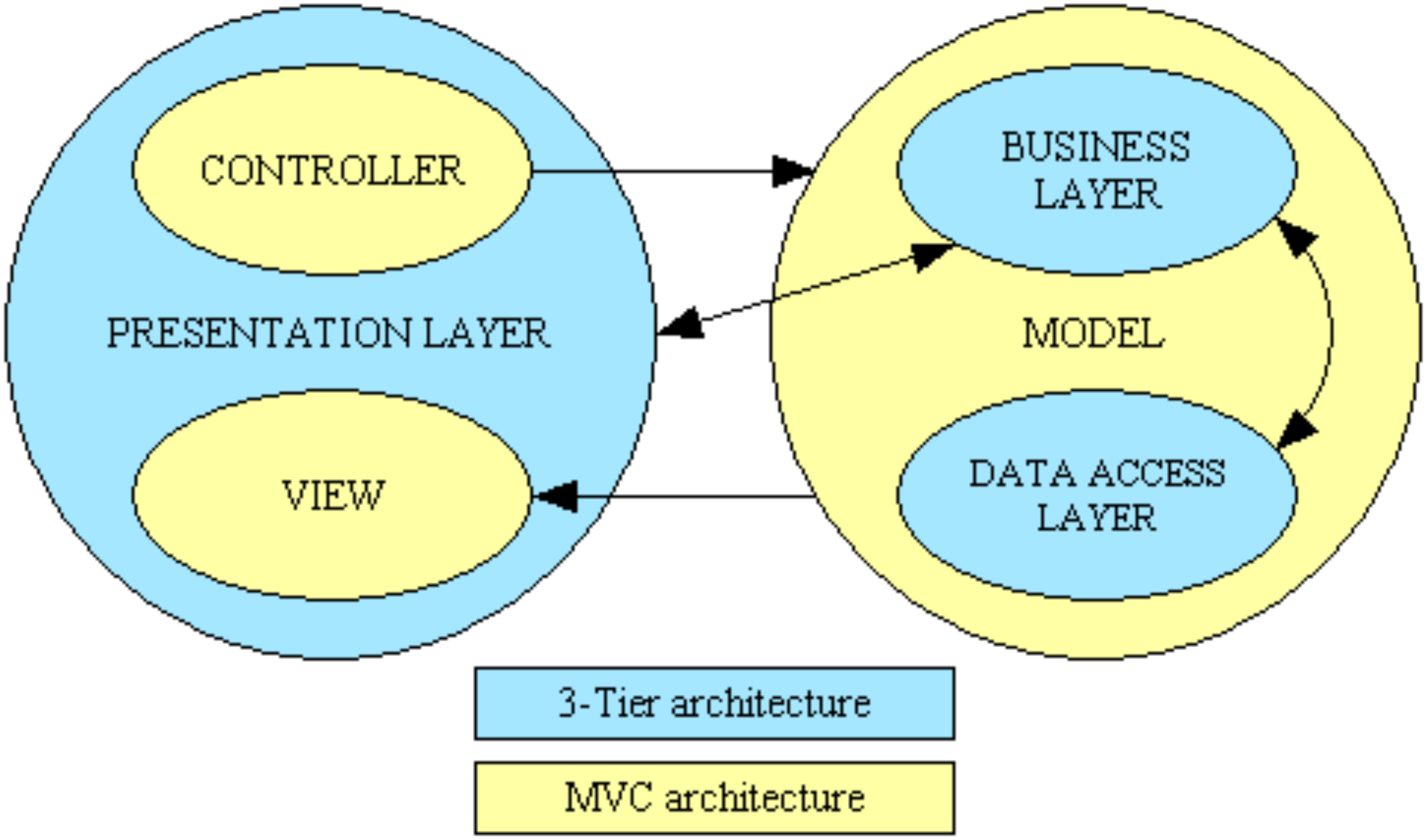
[many more to choose from]

UNIVERSITY OF TORONTO

User Experience



MVC vs. 3-tier


CONTROLLER
PRESENTATION LAYER
VIEW
BUSINESS LAYER
MODEL
DATA ACCESS LAYER
3-Tier architecture
MVC architecture

Express.js

“Express is a minimal and flexible node.js web application framework, providing a robust set of features for building single and multi-page, and hybrid web applications.”

→ Using Template Engine

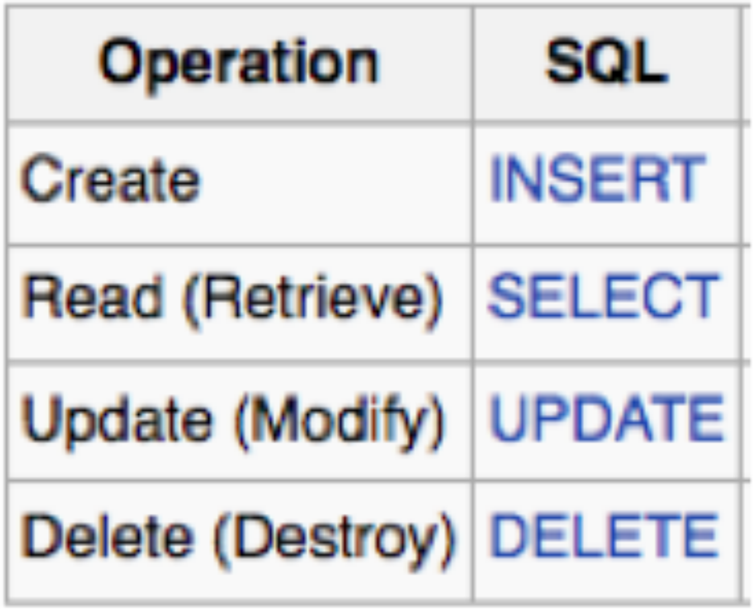
→ ODM Tool Mongoose.

→ Using REST.

UNIVERSITY OF TORONTO

CRUD Pattern

- Pattern for operating on db tables



Operation	SQL
Create	INSERT
Read (Retrieve)	SELECT
Update (Modify)	UPDATE
Delete (Destroy)	DELETE

MySQL with Node.js

Download and Install mySQL from,
http://dev.mysql.com/downloads/

Start/stop/restart your MySQL server,

```
sudo /usr/local/mysql/support-files/mysql.
server start/stop/restart
```

Install mySQL Driver (connector module) for node

`npm install mysql` **or**

`npm install felixge/node-mysql`

“NoSQL” = “Not Only SQL”

NoSQL Systems

Alternative to traditional relational DBMS

+ Flexible schema
+ Quicker/cheaper to set up
+ Massive scalability
+ Relaxed consistency → higher performance & availability

– No declarative query language → more programming
 Relaxed consistency → fewer guarantees

NoSQL Database Types

Key-value stores.

Wide-column stores.

Document.

Graph stores.

HTTP is Stateless

HTTP is stateless, it makes a lot of sense when sharing static information like html, pdf, images over HTTP (1.0).

But as we started using web application, ecommerce sites, we started adding **ad hoc** states on top of HTTP for various reasons.

Adding state to HTTP (1.0 and earlier)

There are various ways to add and maintain states on top of HTTP (not as an integral part):

Client mechanisms:

- Cookies
- Hidden variables
- URL rewriting
- Local storage (for HTTP 1.1 and we have covered this)

Server mechanisms:

- sessions

Cookies

A small amount of information sent by a server to a browser, and then sent back by the browser on future page requests.

Motivation:

→ authentication

→ user tracking

→ maintaining user preferences, shopping carts, etc.

UNIVERSITY OF TORONTO

Session - Persistent State

Current state is stored at the server (i.e., in a file, database)

- Each request includes a token identifying the browsers session (tokens can be passed via cookies, hidden variables, URL rewriting).
- At each request, the executing script uses the token to fetch session state

Session hijacking! Add unique value + signature

Security threats

Insiders	Brute Force	Stolen hardware
Phishing	Malicious Software	Sniffing
Virus	Bugs	Poor architecture

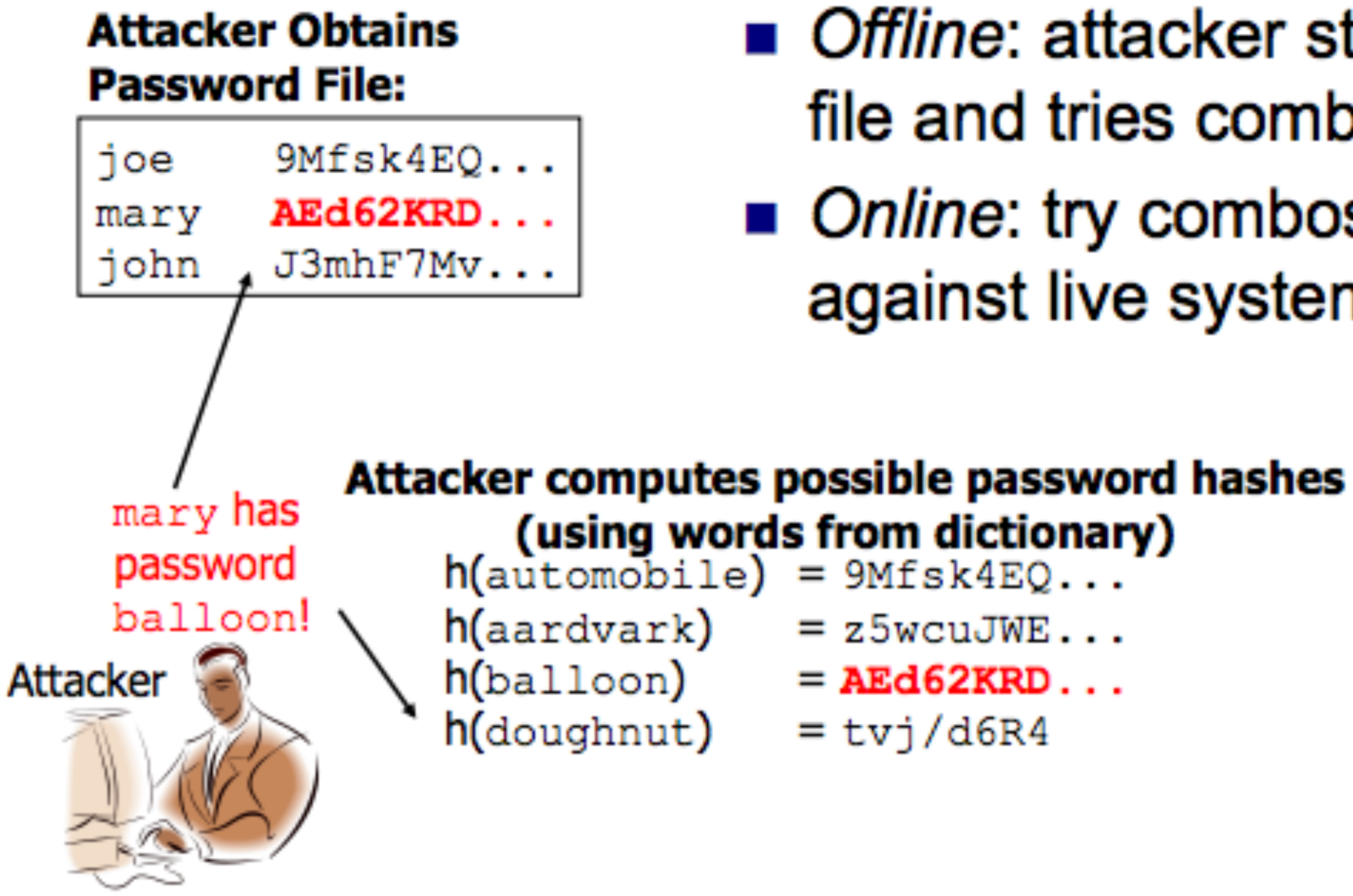
Hashing

- Encrypt passwords, don’t store “in the clear”
  - → Could decrypt (e.g. DES) to check, key storage?
  - → Even better: “one-way encryption”, no way to decrypt
  - → If file stolen, passwords not compromised
  - → Use one-way hash function, h: preimage resistant
  - → Ex: SHA-1 hashes stored in file, not plaintext passwd

```
john:9Mfsk4EQh+XD2lBcCAvputrIuVbWKqbxPgKla7u67oo=
mary:AEd62KRDHUXW6tp+XazwhTLSUlADWXrinUPbxQEfnsI=
joe:J3mhF7Mv4pnfjcnoHZ1ZrUELjSBJFOo1r6D6fx8tfwU=
```

Dictionary Attacks

- *Offline*: attacker steals file and tries combos
- *Online*: try combos against live system



Attacker Obtains Password File:

```
joe     9Mfsk4EQ...
mary    AEd62KRD...
john    J3mhF7Mv...
```

Attacker computes possible password hashes (using words from dictionary)

```
h(automobile) = 9Mfsk4EQ...
h(aardvark)   = z5wcuJWE...
h(balloon)    = AEd62KRD...
h(doughnut)   = tvj/d6R4
```

Security Measure

HoneyPot,

Password Filtering,

Limit Login Attempts,

Aging Password, etc.

Web Performance: Design

- Responsive Design
- Mobile-first Design
- Adaptive Design
- Progressive Enhancement

Changes based on Mobile, Connectivity, Screen, Browser.

UNIVERSITY OF TORONTO

Design Techniques

- Background images -> CSS gradients
- Lazy Image loading
- Multiple image resolutions
- Test capabilities (Modernizr)
- Render core elements first

HTML Resources

- Optimize images (color & size)
- Minify JS and CSS
- Use CSS Sprites to reduce image requests
- Gzip components
- Add Expires or Cache-Control Header
- Don’t forget every request sends cookies

http://css-tricks.com/css-sprites/

CSS Techniques

- Stylesheets at the Top
- Remove unused CSS rules
- Avoid universal selectors (*)
- Don't abuse "border-radius" & "transform"
- Prefer selectors with native JS support

  $('#'id) --- getDocumentById

  $('.class') --- getElementByClassName

  $('tag') --- getElementByTagName

http://perfectionkills.com

UNIVERSITY OF TORONTO

Network - Techniques

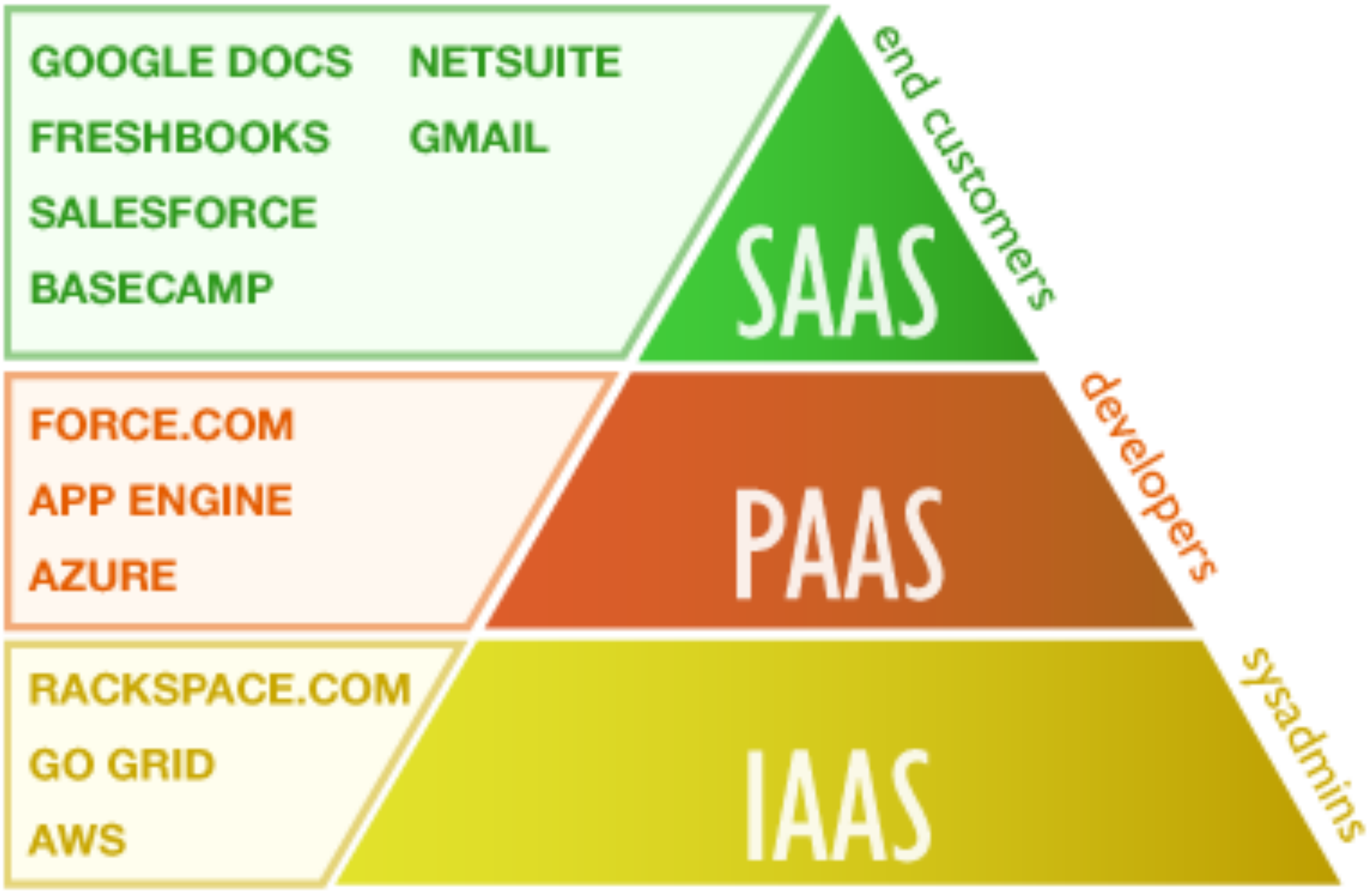
- Make Fewer HTTP requests
- Use a Content Delivery Network
- Split resources across servers - load balance
- But avoid too many DNS lookups
- Create Cookie-free domains
- Careful with redirects (301, 302)

UNIVERSITY OF TORONTO

Database - Techniques

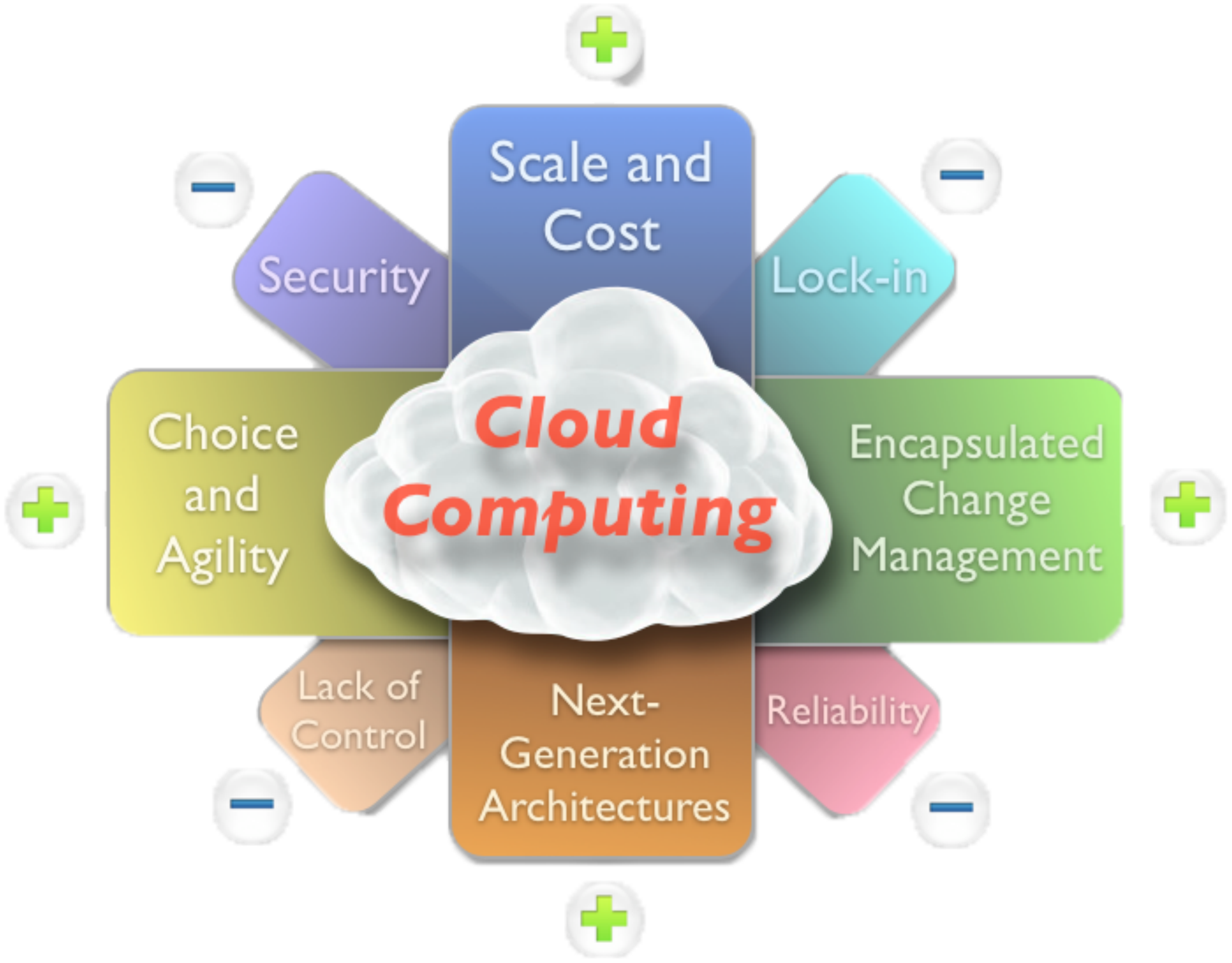
- Learn to write fast queries
  - Configure MySQL “slow” log
- Study performance of indices
  - “explain”, “show index”, “statistics
- De-normalize
- Configure Buffers, Cache, Query Cache

Cloud Service Models



Desktop as a service (DaaS), backend as a service (BaaS), and information technology management as a service (ITMaaS).

Advantage and disadvantages



Nodejs Good or Bad?

https://www.destroyallsoftware.com/talks/the-birth-and-death-of-javascript

The Power of Community

The problem:

+/- 9007199254740992

ECMA Section 8.5 - Numbers

the largest exact integral value is 2^{53}, or 9007199254740992

The Power of Community

Solusion:

https://github.com/substack/node-bigint

```
var bigint = require('bigint');
var b = bigint('782910138827292261791972728324982')
   .sub('182373273283402171237474774728373')
   .div(8);
console.log(b);
```

What App you should build using Node.js

Real-time applications:

online games,

collaboration tools,

chat rooms,

or robot? or sensor?

Basically, any application using "long-polling", you can write an application that sends updates to the user in real time.

Data Streaming

Where you should not use Node.js

Computation heavy Server app