



Web-application
I've always dreamt of

@JavaDay Lviv

2015

Victor Polischuk

Experience Summary

Overall	32
Programming	26
Java	11
Web	11
Legacy	11
Professional	13
Has something to share	<input checked="" type="checkbox"/>

infopulse



@alkovictor 
victor-cr 

Prehistoric Era [...-1995]

- Static content only
- Interaction without server:
 - Impossible
 - Unthinkable
- Server-side HTML adjustments



Welcome to Amazon.com Books!

*One million titles,
consistently low prices.*

(If you explore just one thing, make it our personal notification service. We think it's very cool!)

SPOTLIGHT! -- AUGUST 16TH

These are the books we love, offered at Amazon.com low prices. The spotlight moves **EVERY** day so please come often.

ONE MILLION TITLES

Search Amazon.com's [million title catalog](#) by author, subject, title, keyword, and more... Or take a look at the [books we recommend](#) in over 20 categories... Check out our [customer reviews](#) and the [award winners](#) from the Hugo and Nebula to the Pulitzer and Nobel... and [bestsellers](#) are 30% off the publishers list...

Medieval Era [1995-2005]

- Static content mostly
- Interaction without server: cached resources
- Server-side HTML+CSS+JS adjustments
- Frames as asynchronous data loading
- Birth of:
 - Servlet API
 - JavaScript
 - DHTML concept
 - Flash
 - Browser-compatibility hell
 - RFC1925

Google!

Search the web using Google!

10 results ▾

Google Search

I'm feeling lucky

Index contains ~25 million pages (soon to be much bigger)

[About Google!](#)

[Stanford Search](#) [Linux Search](#)

Get Google! updates monthly!

your e-mail

Subscribe

[Archive](#)

Copyright ©1997-8 Stanford University

Renaissance Era [2005-2010]

- Static content with dynamic “whistles”
- Interaction without server: simple functions
- Server-side HTML+CSS+JS adjustments
- Rising popularity of Rich Internet Applications
- Birth of:
 - jQuery
 - DOJO
 - AJAX and Comet

Web Images Video Maps News Shopping Gmail more ▾

gmail.com | My Profile | Help | My Account | Sign out

Google maps

e.g., "San Francisco, CA" or "10011" [Learn more.](#)

Real estate

Search Maps

[Remove search options](#)

[Get Directions](#) [My Maps](#)

[Print](#) [Send](#) [Link](#)

Real estate

[For rent](#) | [For sale](#) \$ to \$

Hide options

Property: Detached Semi Townhouse/Unit

Bedrooms: Studio 1 2 3 4+

Bathrooms: 1+ 1.5+ 2+ 3+

Parking: 0 1 2+

Area: to sqm

A [5 Caravel Ln, Docklands Victoria](#) ☆ - [more info](#) »



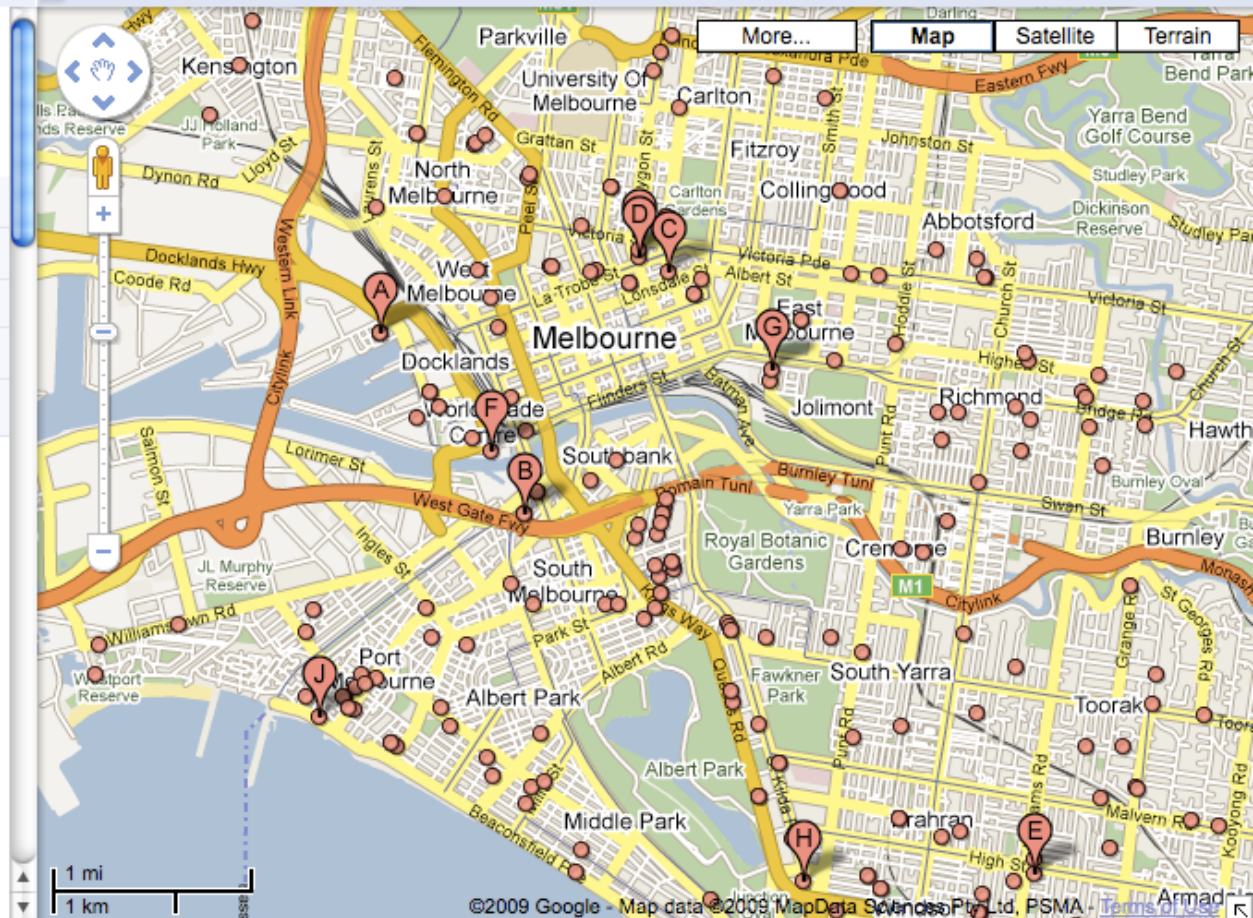
~\$600(starting) 2 bed 1 bath 1 parking

B [217/99 Whiteman St, Southbank Victoria](#) ☆ - [more info](#) »



\$479,000 2 bed 1 bath 1 parking
"Entertaining will be a treat in this amazing two bedroom apartment with spacious living areas, gourmet kitchen and large private balcony where ..." [myhome.com.au](#)

C [265 Exhibition St, Melbourne Victoria](#) ☆

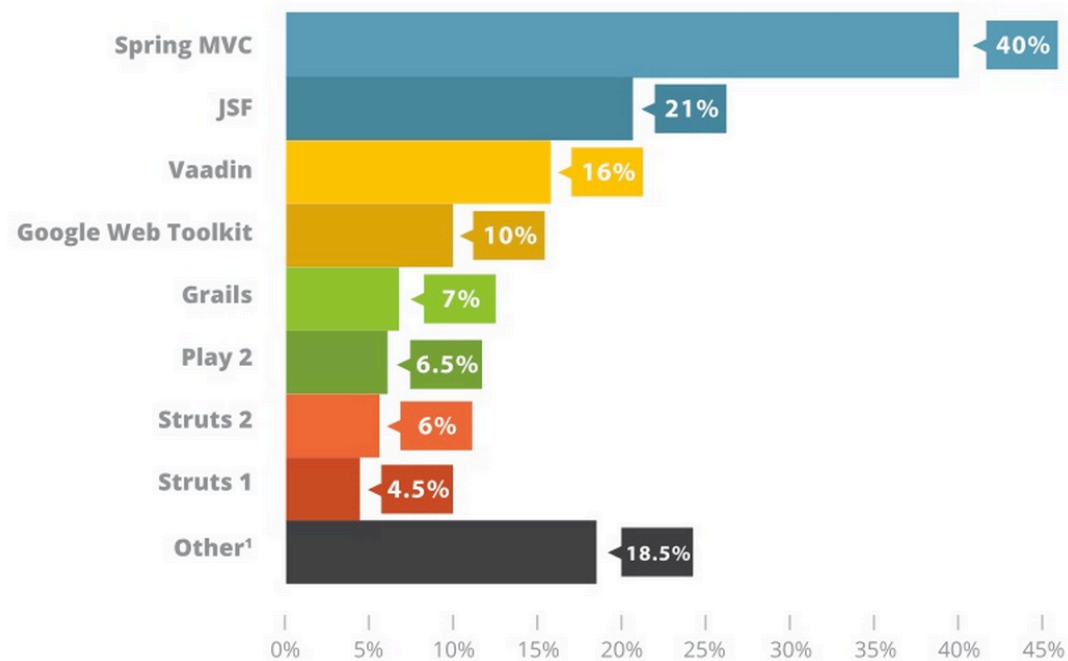


Modern Era [2010-...]

- Internet of Things
- Big Data
- Cloud Computing
- ???

Overview

Web frameworks in use *



* Multiple selections were possible and the results were normalized to exclude non-users

¹ Including Wicket, Seam, Tapestry, Play 1, ZK framework, VRaptor and about 40 others

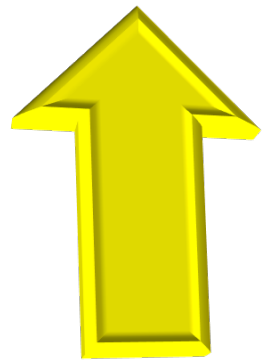
Business Needs

- Integration with anything:
 - Mobile
 - Mainframe
 - Coffee machine
- High availability:
 - Hard to predict clients behavior
- Data is the TOP priority:
 - We make money out of it
- Client's satisfaction:
 - Should be happier than those of our competitors

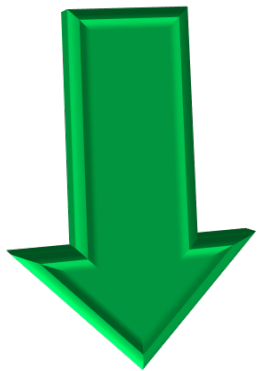
Idea

- “All problems in computer science can be solved by another level of indirection” © *David Wheeler*
- “Do One Thing and Do It Well” © *Unix community*

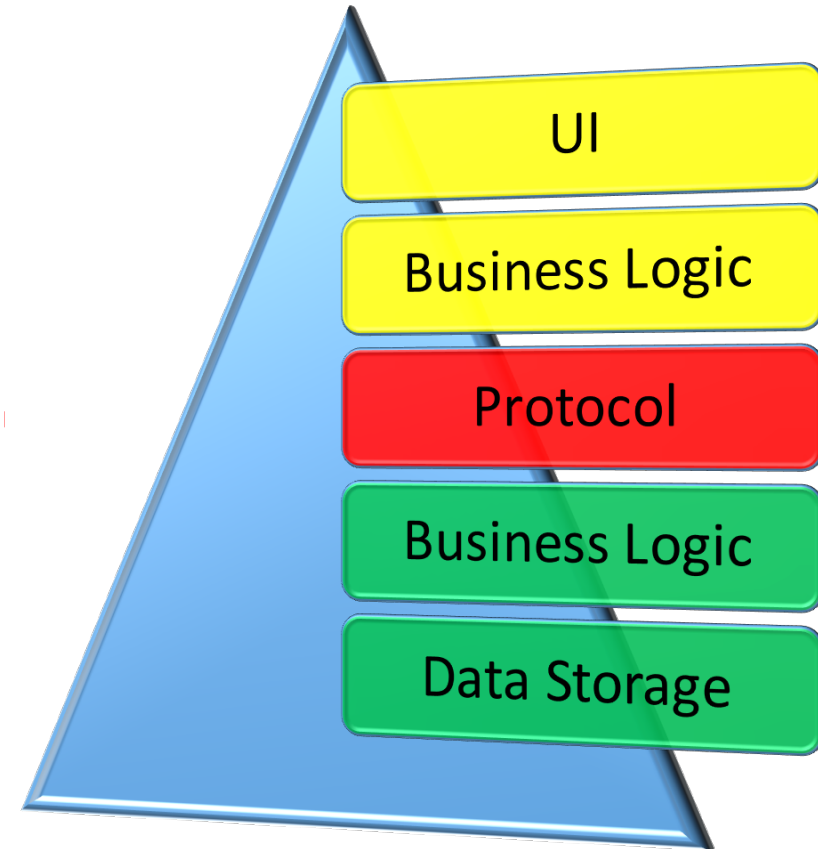
Idea



Client
(dynamic)



Server
(stability)



Client



Client

```
$http.get("/rs/stat/" + key).success(function(data) {
    self.statistics.records = data;
}).error(function(data, statusCode) {
    self.errorContent = data;

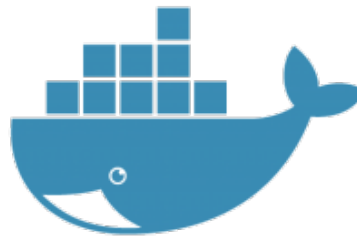
    if (statusCode === 404 || statusCode >= 500) {
        self.unavailable = true;
        self.timerId = setInterval(self.ping, 1000);
    }
});
```

```
self.ping = function() {
    $http.get("/rs/ping").success(function() {
        clearInterval(self.timerId);
        self.unavailable = false;
    });
}
```

Server



mongoDB



EHCACHE



Server

```
def receive = {  
  case GetNumber(id) => sender ! getNumber(id)  
  case GetStat(key) => sender ! getStatistics(key)  
  case AddVote(entity) => sender ! doVote(entity)  
  case AddClient(ip) => sender ! createClient(ip.value)  
  case entity: AddProposal => sender ! createNumber(entity)  
  case entity: UpdateProposal => sender ! updateNumber(entity)  
}
```

```
val route =  
  path("ping") {  
    ctx => ctx.complete("PONG")  
  } ~ (path("stat" / JavaUUID) & get) {  
    key => rejectEmptyResponse {  
      handlePerRequest {  
        GetStat(key)  
      }  
    }  
  }  
}
```

Business Needs

- Integration with anything
- High availability
- Data is the TOP priority
- Client's satisfaction

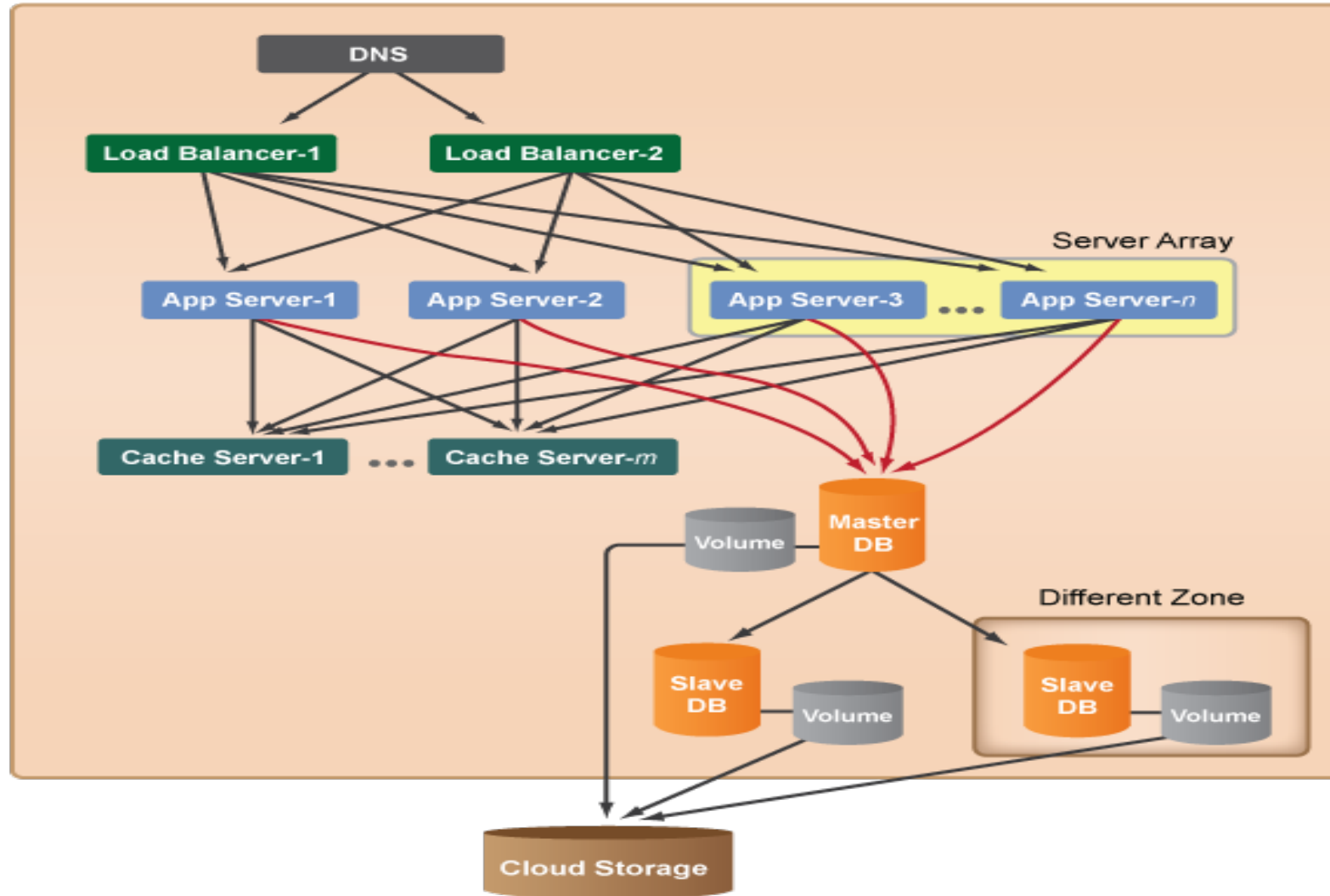
Technical Needs

- Scalability
- Security
- Testability
- “Otherity”

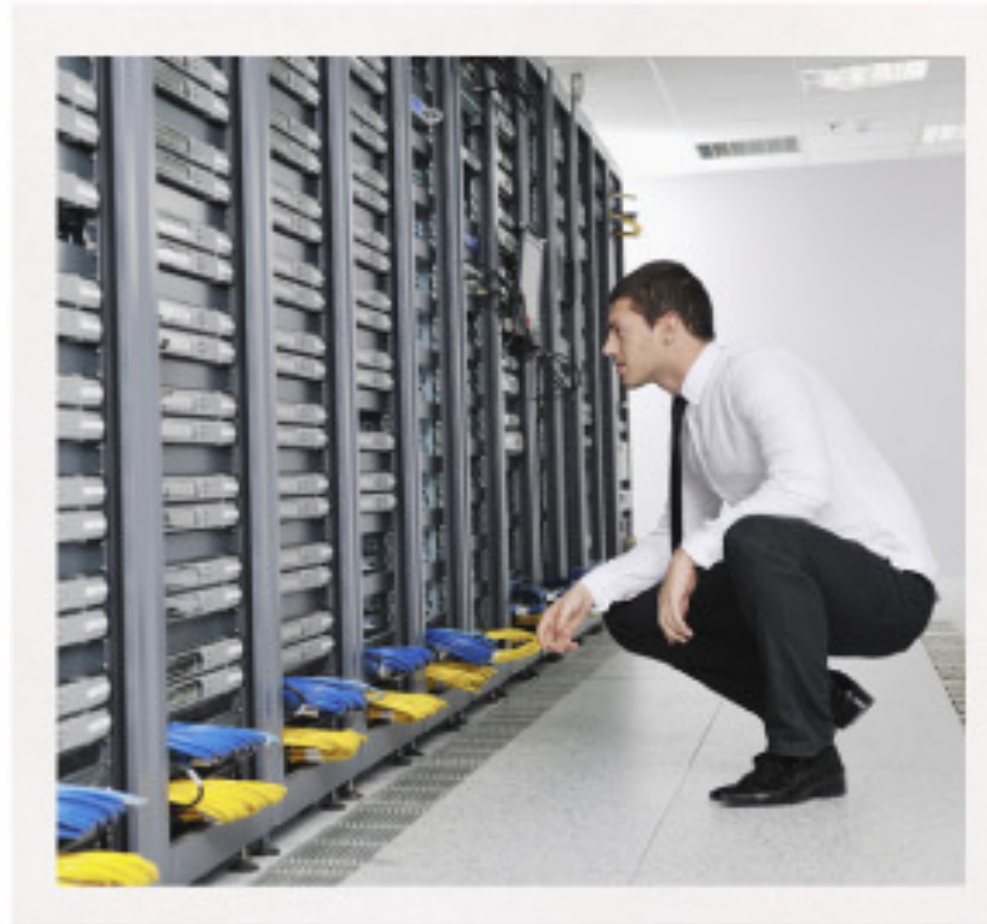
Scalability

- Server is just a bunch of endpoints
- Independent client
- State
 - Client needs it
 - Server doesn't
- SLA
 - Way to Nine-Nines
 - You do not want to lose clients, right?

...We Design



...We Imagine



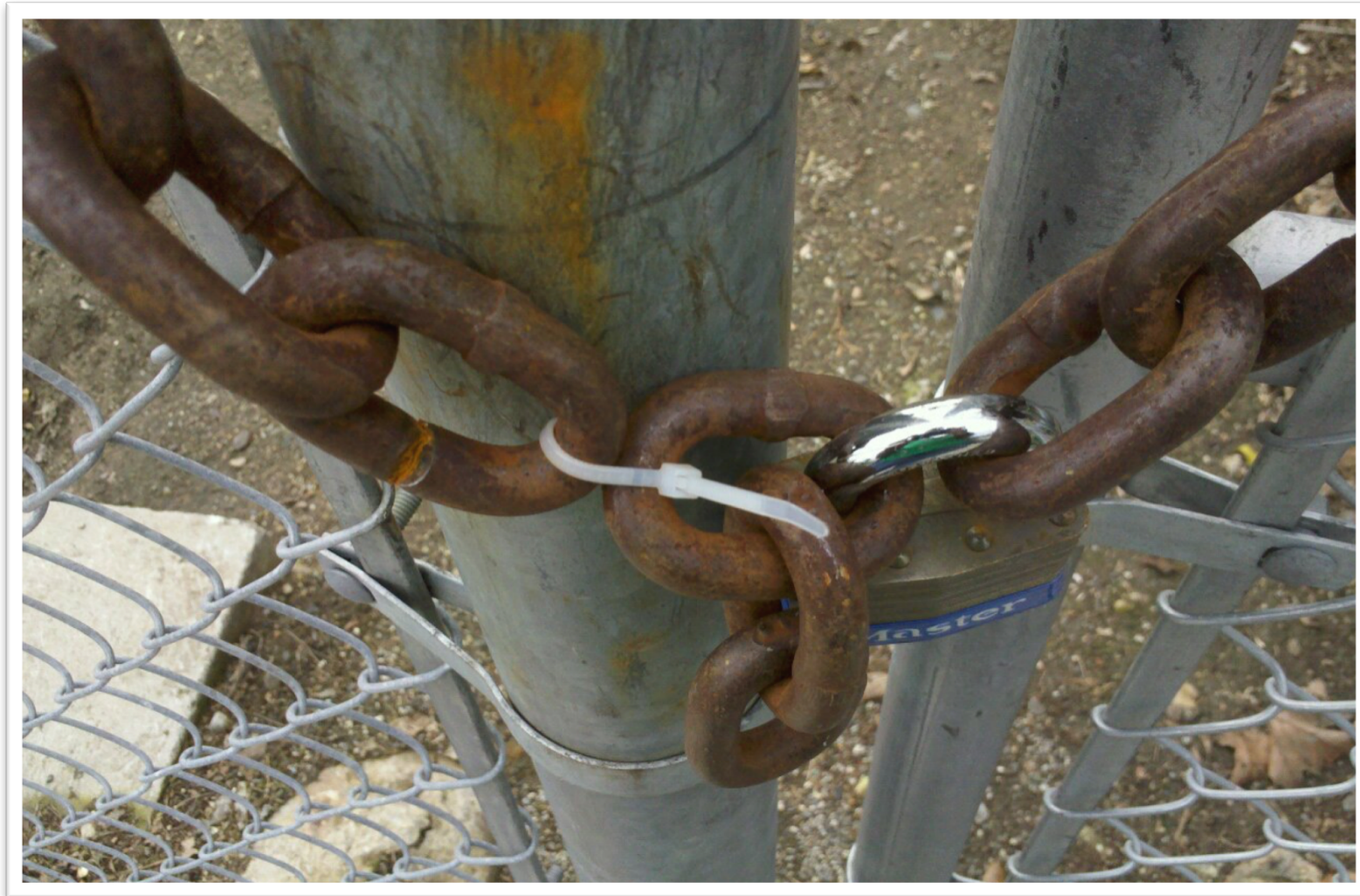
...We Build



Security

- Do you need it on each request?
- Do you know how it works?
- Do you think it is secure?

...Of Course



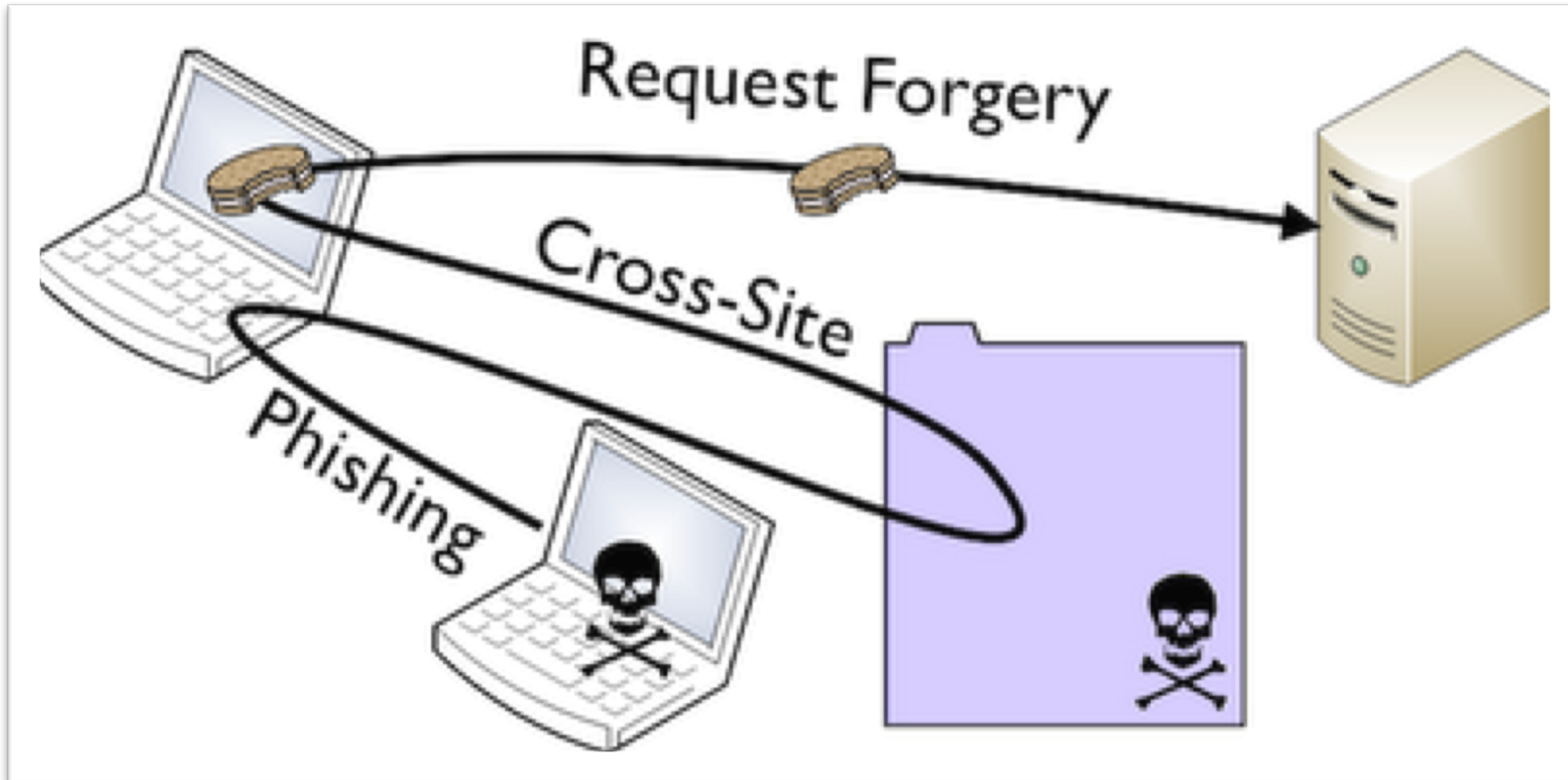
...Safe JSF

```
<h:form>
  <h:selectOneMenu value="#{control.selectedServer}">
    <f:selectItems value="#{control.availableServers}" />
  </h:selectOneMenu>

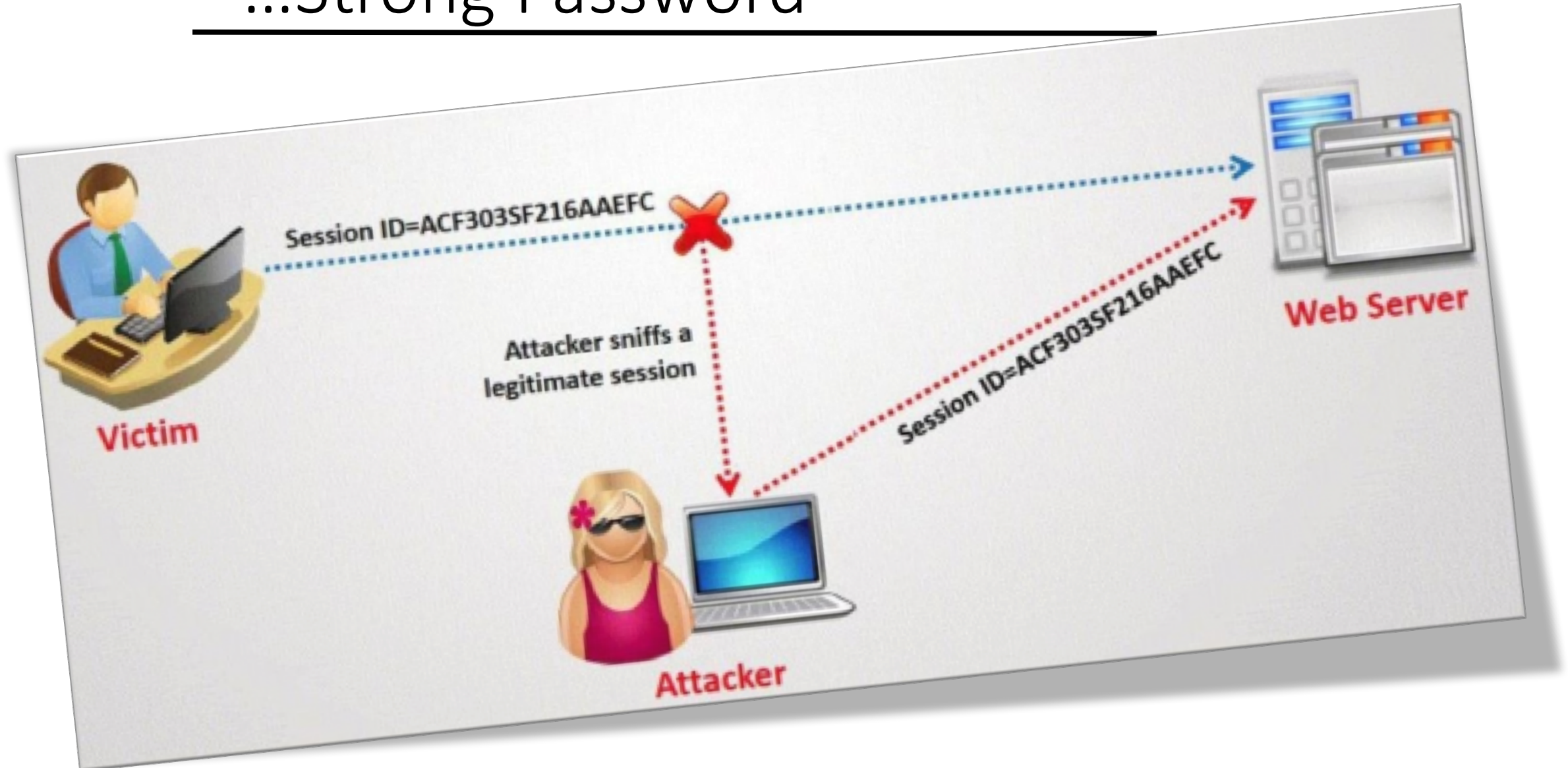
  <h:commandButton value="Reboot" action="#{control.reboot}" />
</h:form>
```

```
@ManagedBean(name="control")
@ViewScoped
public class ControlBean implements Serializable {
...
  private String selectedServer;
  private Map<String, String> availableServers;
...
  public String reboot() { ... }
```

...Safe Cookies



...Strong Password



...\$5000 bug

POST

```
https://www.youtube.com/live_events_edit_status_ajax?action_delete_live_event=1
event_id: ANY_VIDEO_ID
session_token: YOUR_TOKEN
```

```
{
  "success": 1
}
```



Testability

- Server side:
 - Unit
 - Integration
 - Contract
 - [optional] Load & Performance
- Client side:
 - Unit
 - Integration
 - Contract
 - [optional] Browser & Sanity

...Client

```
describe("A suite is just a function", function() {  
  var a;  
  it("and so is a spec", function() {  
    a = true;  
    expect(a).toBe(true);  
  });  
});
```

Jasmine 1.3.1 revision 1354556913

finished in 0.024s

•

Passing 1 spec

No try/catch

Responsibility

- Client:
 - Responsive
 - User-friendly
 - Pretty

- Server:
 - Scalable
 - Fail-proof
 - Redundant

Traffic

- Reduce useless to none
- De-increase useful:
 - Increase volume
 - Decrease size

Optimization

- Client
 - Caching
 - Pre-fetching
 - Incremental updates
- Server:
 - Caching
 - Pre-fetching
 - Incremental updates

Encapsulation

- Both parties can have their own secrets

Polygamy

- Is it moral if a client has many servers and vice versa?

Summary

- Effectiveness: you have one job
- Security: you are limiting interaction to minimum
- Testability: test JSF page, I dare you!
- State: -less server, -ful client
- Elasticity/scalability: your data volume is growing
- Network optimizations: you know what you send
- Local optimizations: you know what you do
- Many-to-many connections
- SLA

Questions

