Improving the Performance and Security of AJAX Web Applications

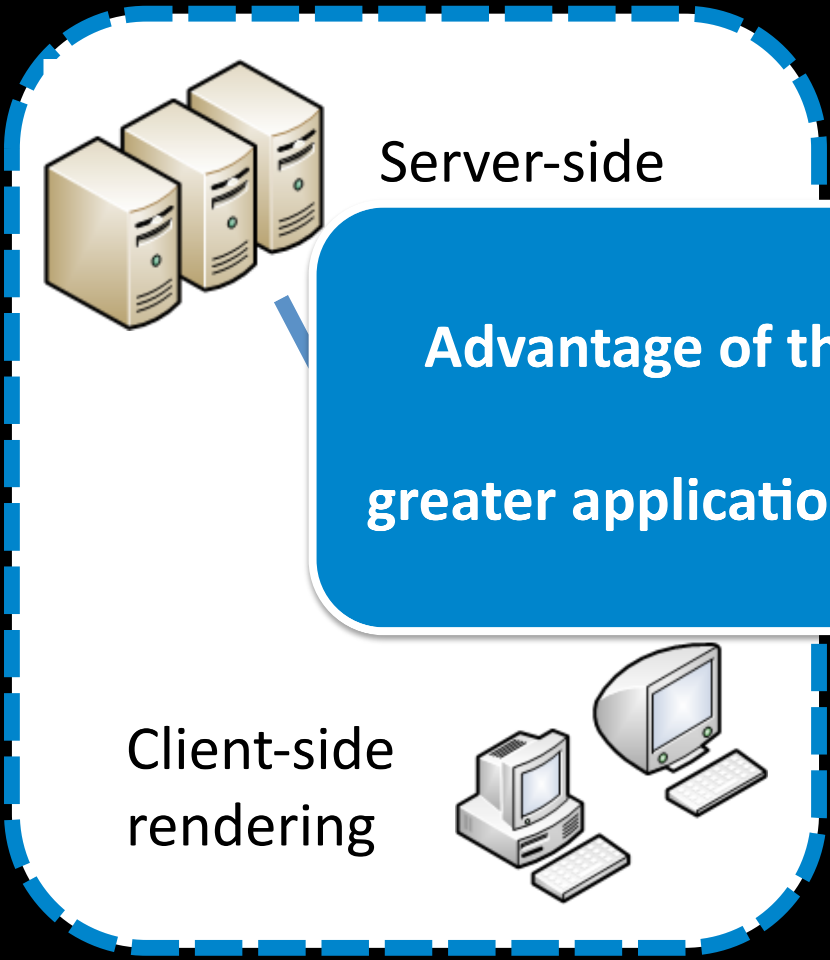
Ben Livshits
Microsoft Research
Redmond, WA

http://research.microsoft.com/~livshits/

Web 2.0 is Upon Us

Source:
flickr.com

Web 1.0 → Web 2.0



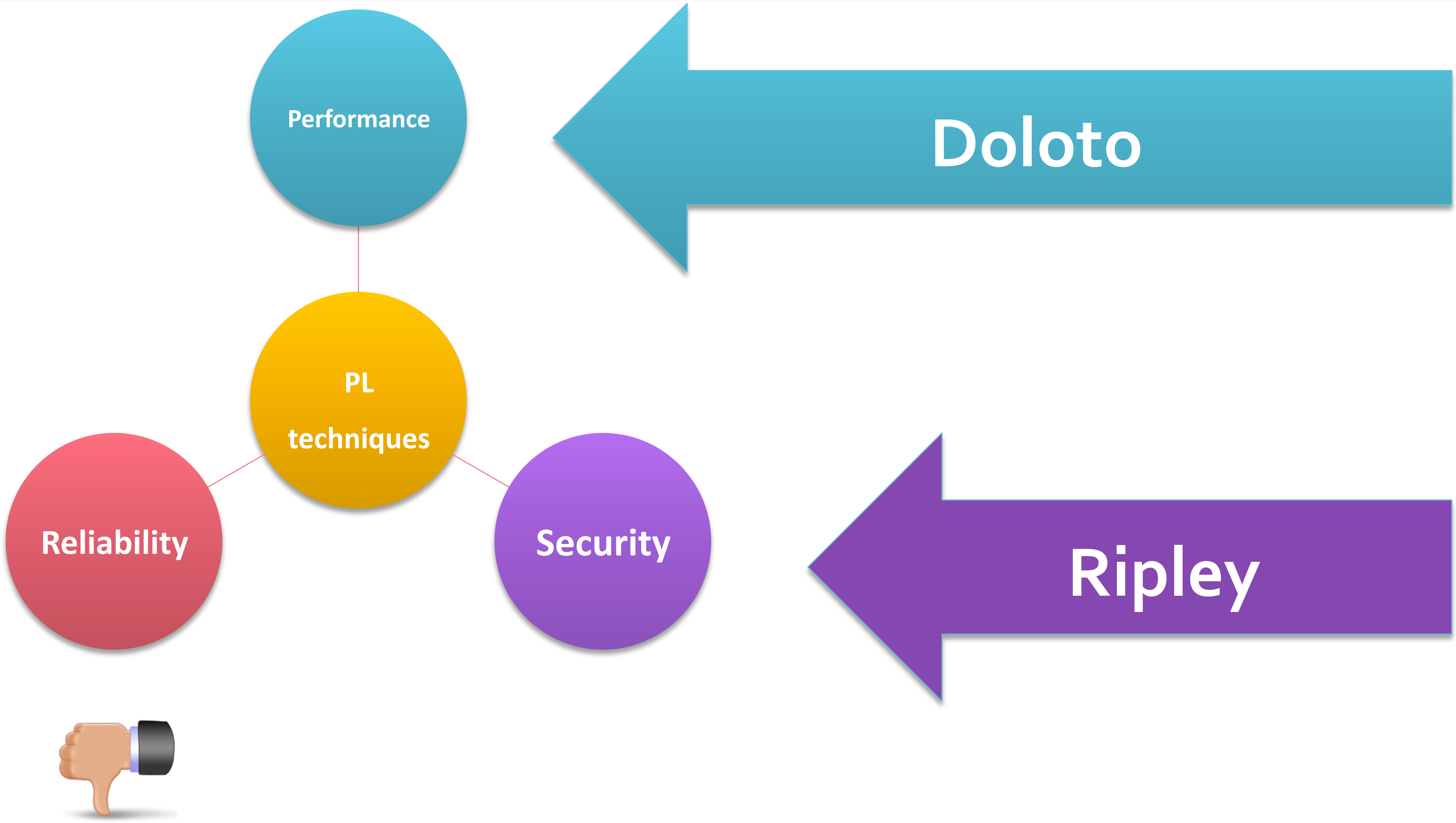
Server-side

Client-side rendering

Advantage of the AJAX model:

greater application responsiveness

Outline of the Talk



Performance

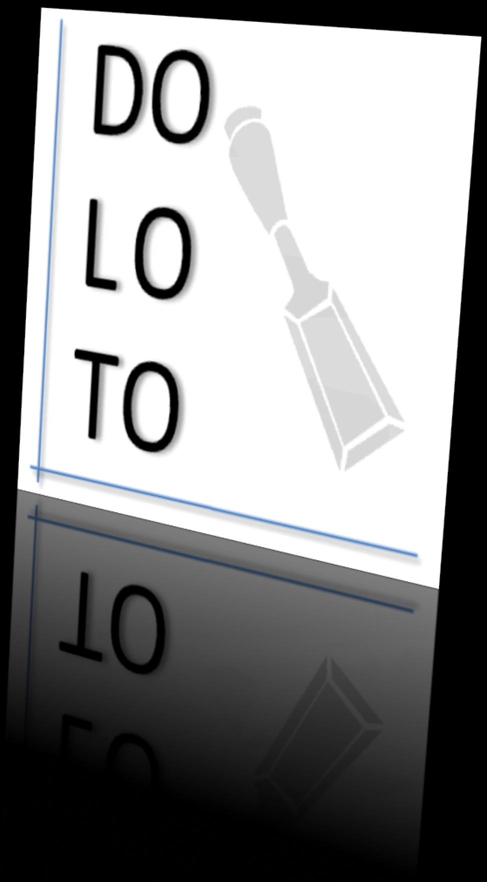
PL techniques

Reliability

Security

Doloto

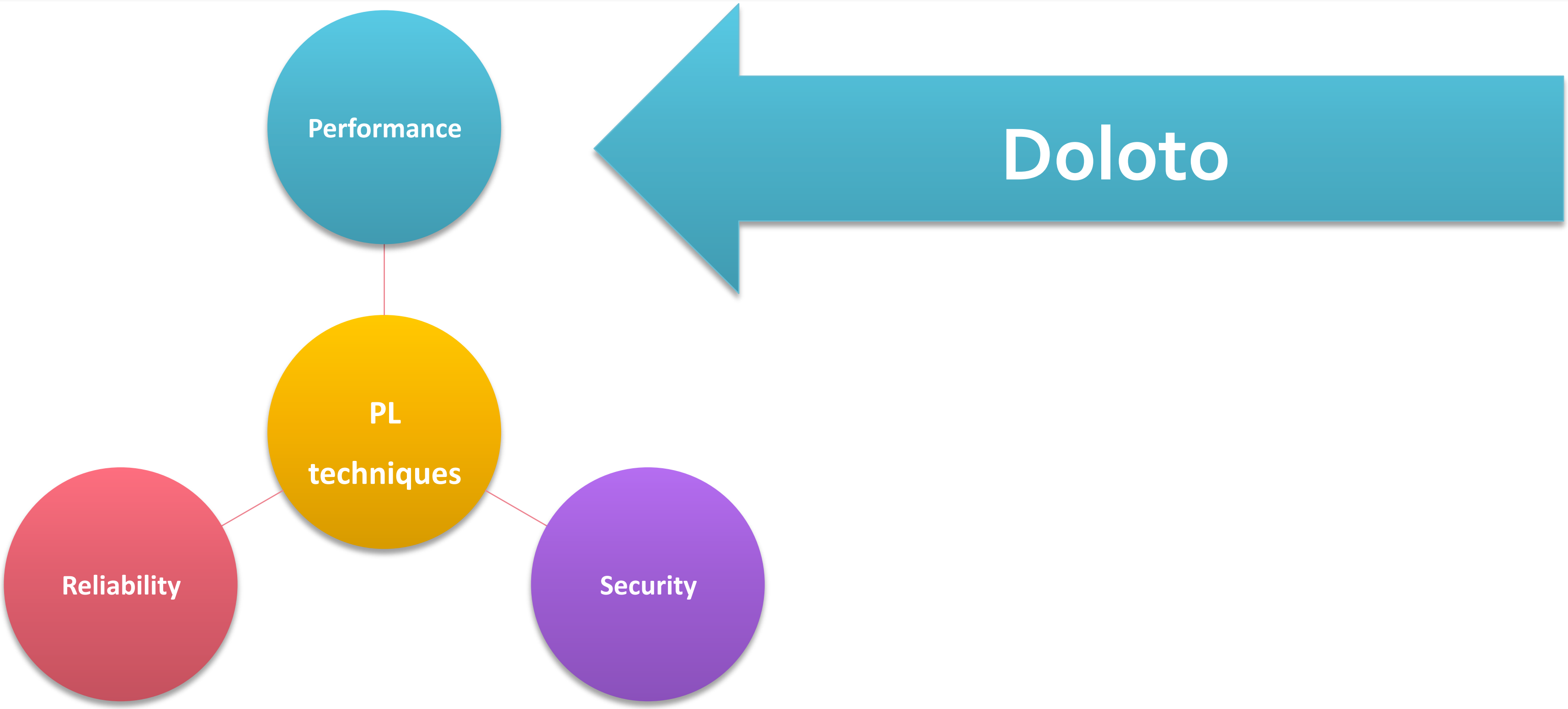
Ripley



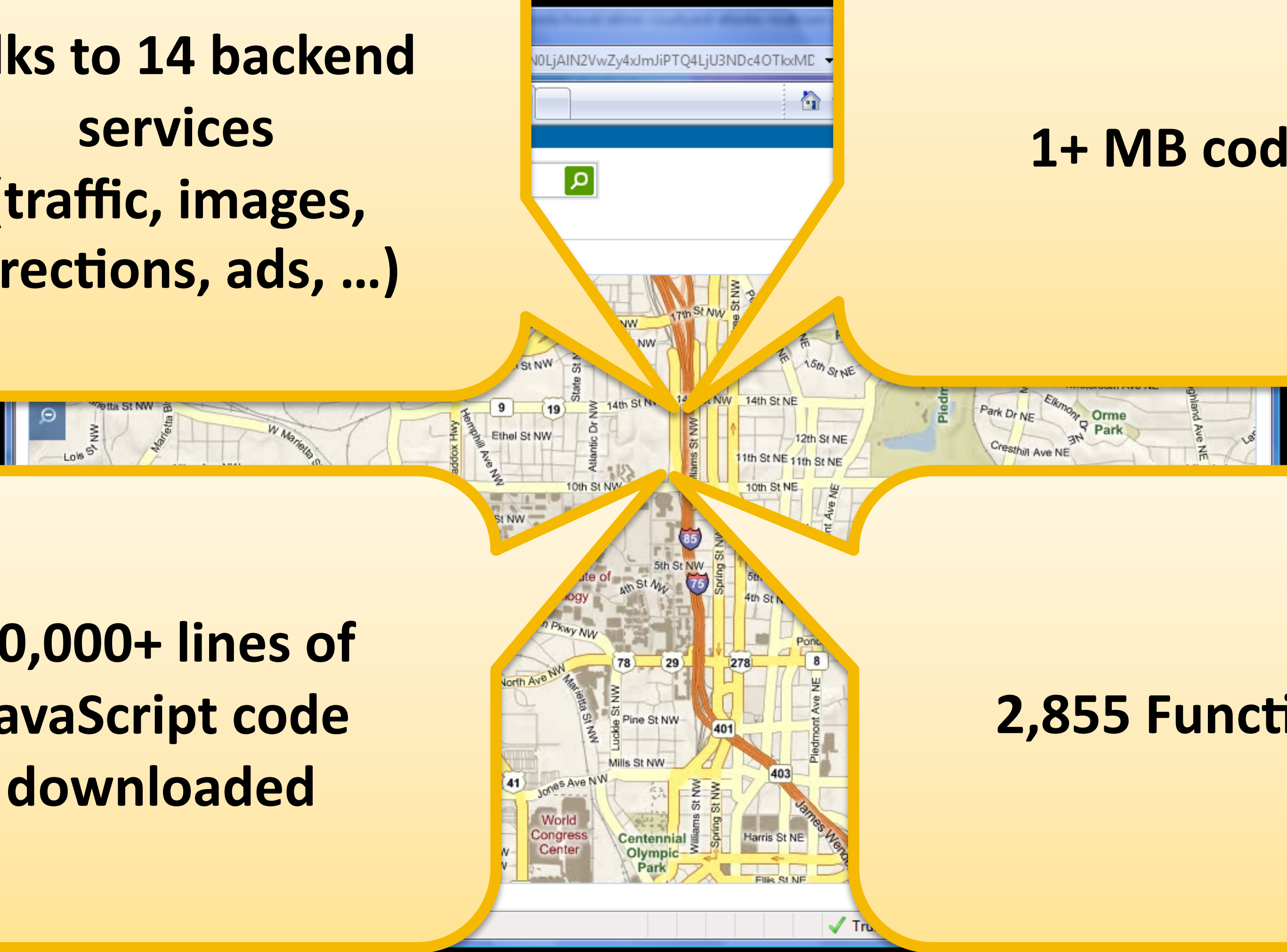
Doloto

Code Splitting for AJAX Applications

Outline of the Talk



A Web 2.0 Application Disected



Talks to 14 backend services (traffic, images, directions, ads, …)

1+ MB code

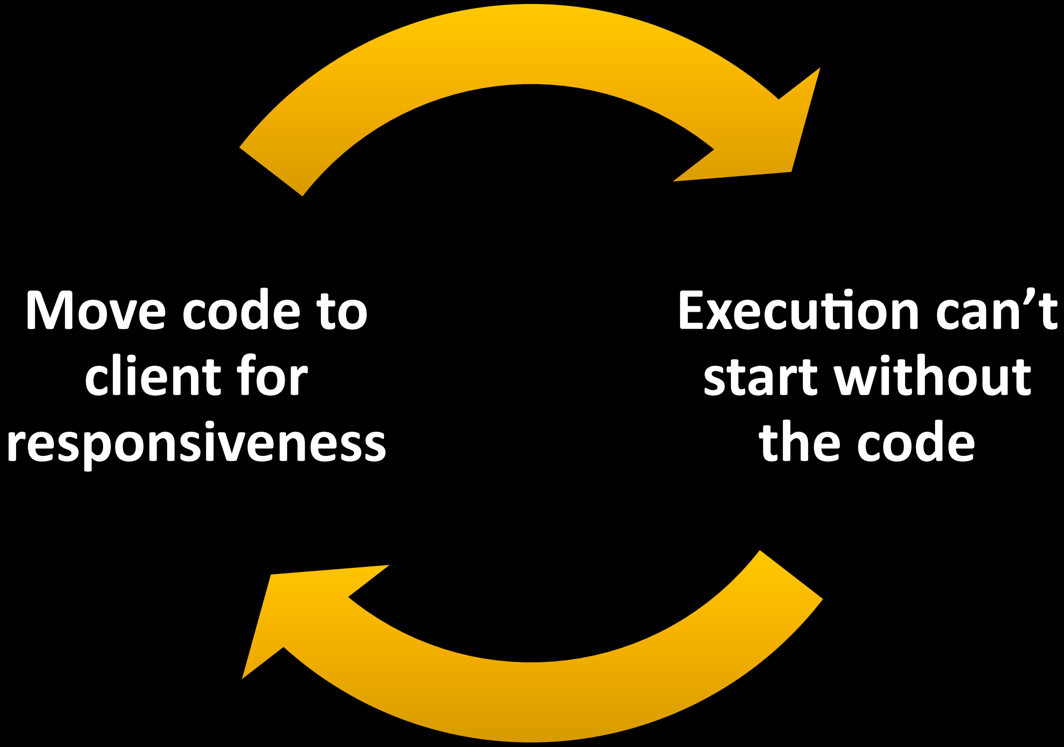
70,000+ lines of JavaScript code downloaded

2,855 Functions

Most of the application download is JavaScript code

Slows down application execution

AJAX Responsiveness: Catch-22



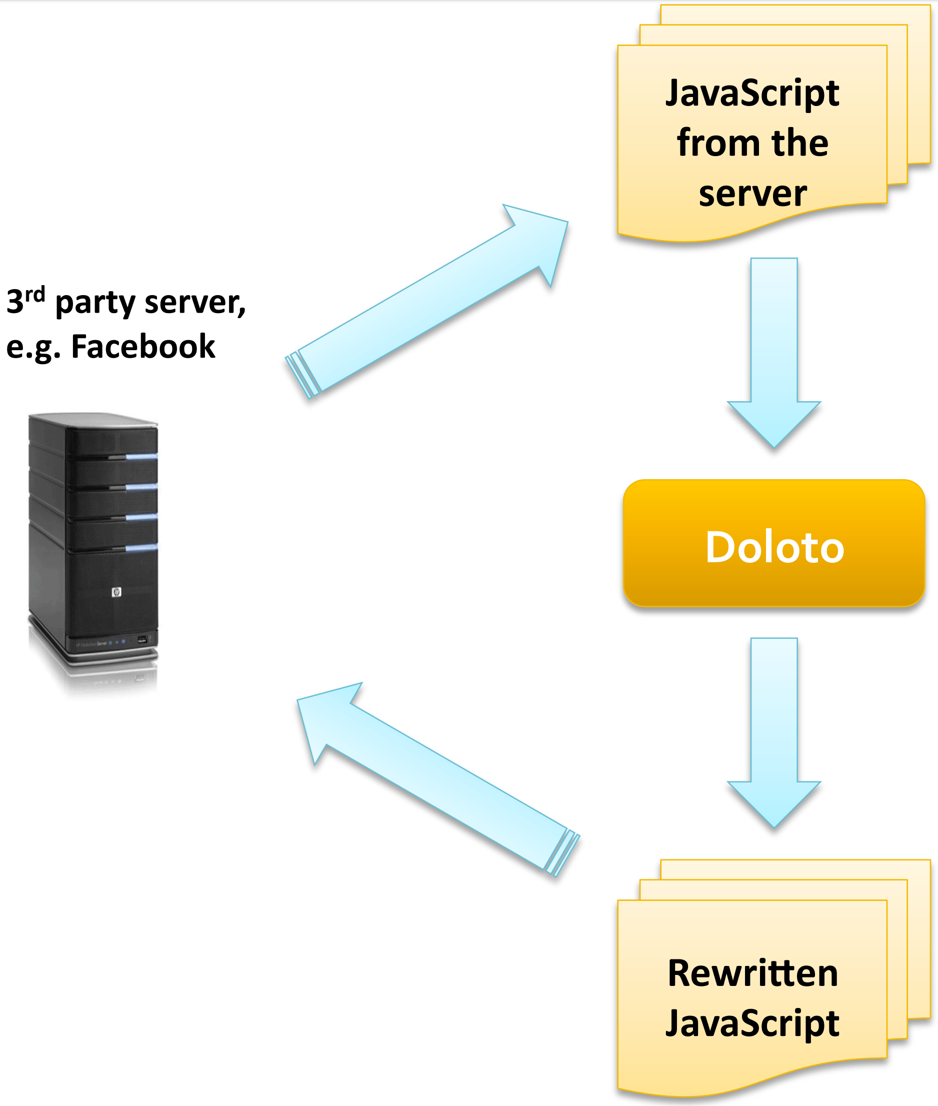
Move code to client for responsiveness

Execution can’t start without the code

Motivation for Doloto

- Idea behind Doloto
  - Start with a small piece of code on the client
  - Download required code on demand (pull)
  - Send code when bandwidth available (push)
- Leads to better application responsiveness
  - Interleave code download & execution
  - Faster startup times
  - Rarely executed code is rarely downloaded

Doloto Workflow



3rd party server,
e.g. Facebook

JavaScript from the server

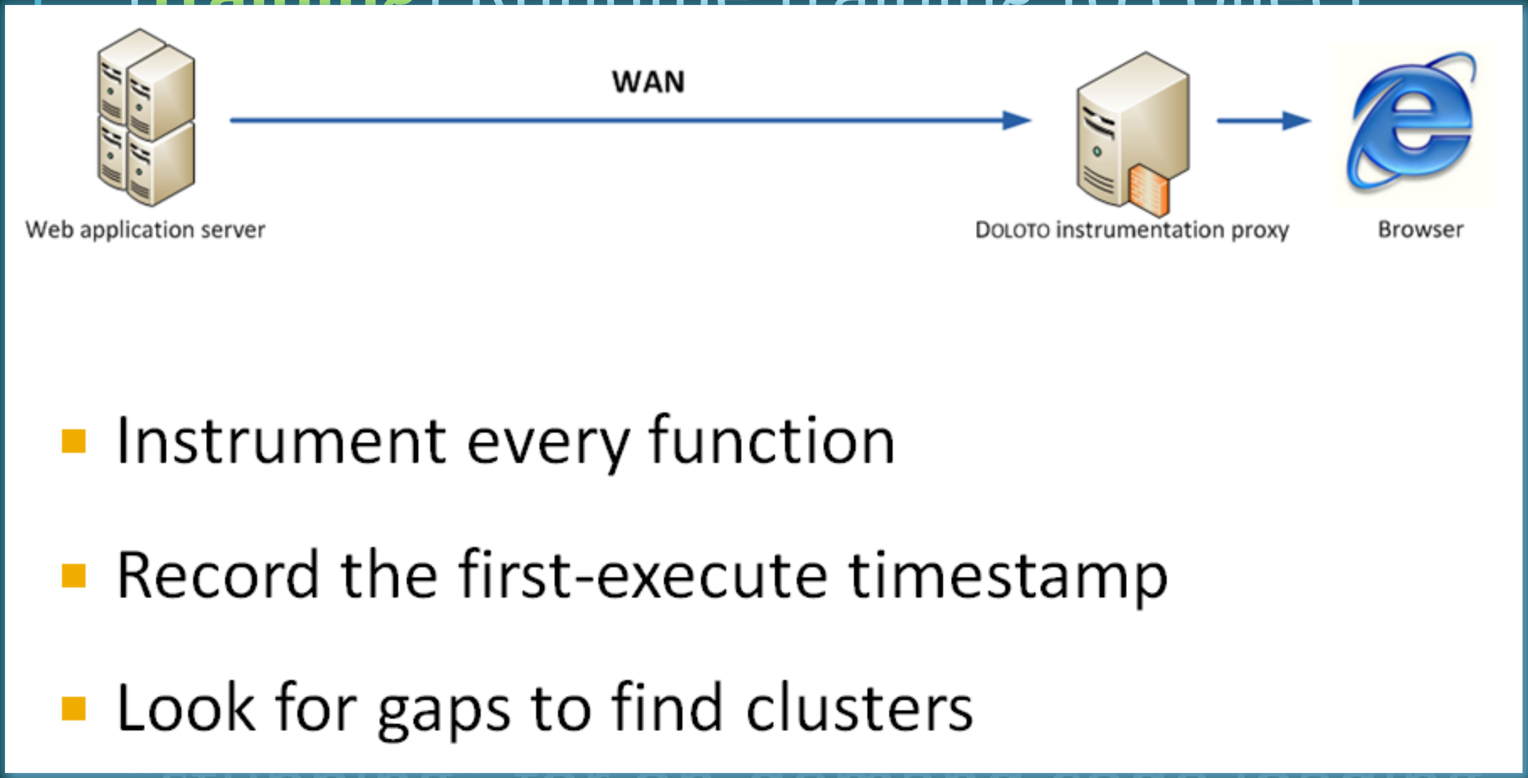
Doloto

Rewritten JavaScript

- Doloto intercepts JavaScript from the server using a proxy
- Instruments and rewrites it on the client
- Deploy it back to the server as the last step

Doloto: the Steps

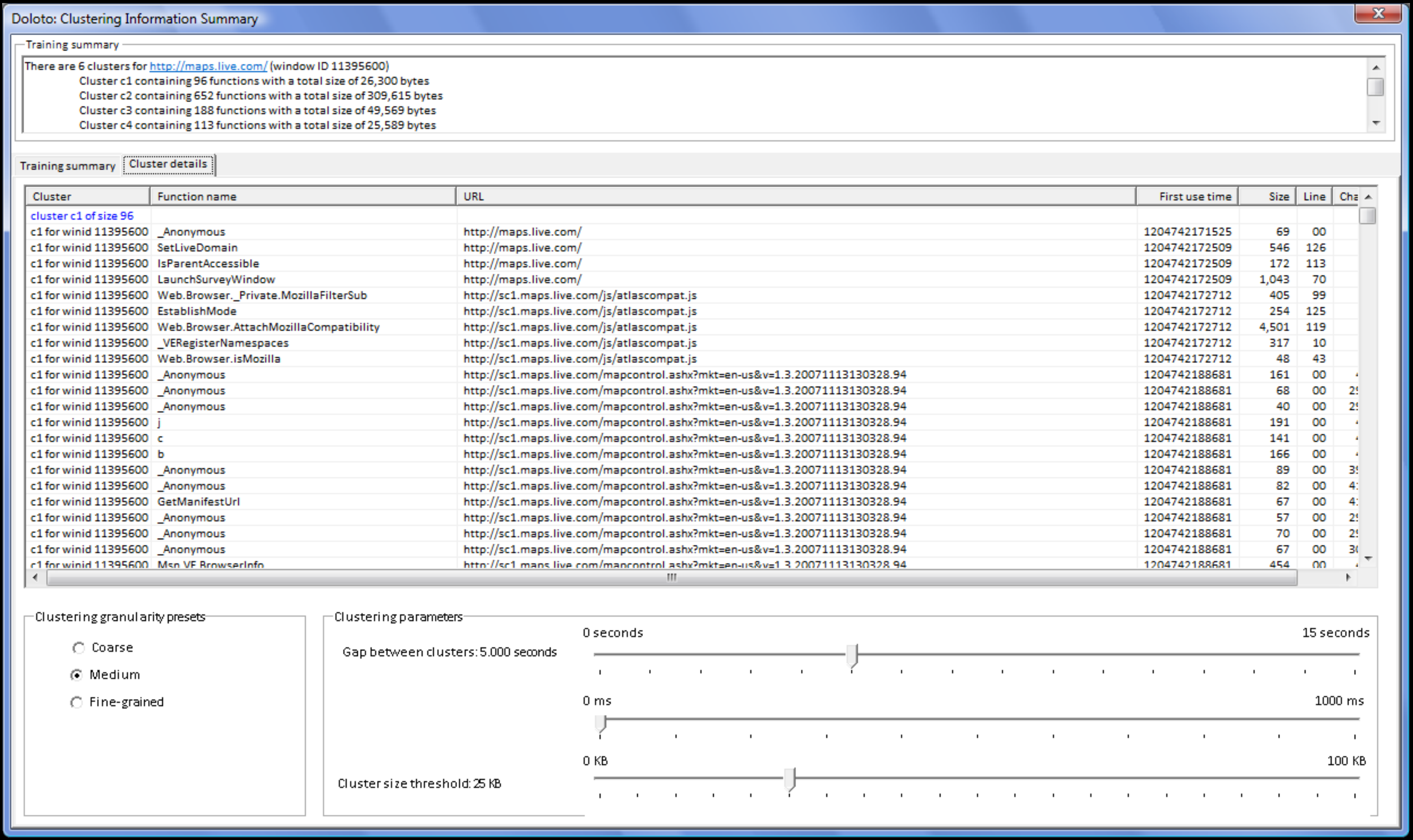
1. **[training]** Runtime training to collect



- Instrument every function
- Record the first-execute timestamp
- Look for gaps to find clusters

stubbing" for on-demand code loading

Doloto Training Tool



There are 6 clusters for http://maps.live.com/ (window ID 11395600)
Cluster c1 containing 96 functions with a total size of 26,300 bytes
Cluster c2 containing 652 functions with a total size of 309,615 bytes
Cluster c3 containing 188 functions with a total size of 49,569 bytes
Cluster c4 containing 113 functions with a total size of 25,589 bytes

Cluster	Function name	URL	First use time	Size	Line	Cha
cluster c1 of size 96						
c1 for winid 11395600	_Anonymous	http://maps.live.com/	1204742171525	69	00	
c1 for winid 11395600	SetLiveDomain	http://maps.live.com/	1204742172509	546	126	
c1 for winid 11395600	IsParentAccessible	http://maps.live.com/	1204742172509	172	113	
c1 for winid 11395600	LaunchSurveyWindow	http://maps.live.com/	1204742172509	1,043	70	
c1 for winid 11395600	Web.Browser._Private.MozillaFilterSub	http://sc1.maps.live.com/js/atlascompat.js	1204742172712	405	99	
c1 for winid 11395600	EstablishMode	http://sc1.maps.live.com/js/atlascompat.js	1204742172712	254	125	
c1 for winid 11395600	Web.Browser.AttachMozillaCompatibility	http://sc1.maps.live.com/js/atlascompat.js	1204742172712	4,501	119	
c1 for winid 11395600	_VERegisterNamespaces	http://sc1.maps.live.com/js/atlascompat.js	1204742172712	317	10	
c1 for winid 11395600	Web.Browser.isMozilla	http://sc1.maps.live.com/js/atlascompat.js	1204742172712	48	43	
c1 for winid 11395600	_Anonymous	http://sc1.maps.live.com/mapcontrol.ashx?mkt=en-us&v=1.3.20071113130328.94	1204742188681	161	00	4
c1 for winid 11395600	_Anonymous	http://sc1.maps.live.com/mapcontrol.ashx?mkt=en-us&v=1.3.20071113130328.94	1204742188681	68	00	2
c1 for winid 11395600	_Anonymous	http://sc1.maps.live.com/mapcontrol.ashx?mkt=en-us&v=1.3.20071113130328.94	1204742188681	40	00	2
c1 for winid 11395600	j	http://sc1.maps.live.com/mapcontrol.ashx?mkt=en-us&v=1.3.20071113130328.94	1204742188681	191	00	4
c1 for winid 11395600	c	http://sc1.maps.live.com/mapcontrol.ashx?mkt=en-us&v=1.3.20071113130328.94	1204742188681	141	00	4
c1 for winid 11395600	b	http://sc1.maps.live.com/mapcontrol.ashx?mkt=en-us&v=1.3.20071113130328.94	1204742188681	166	00	4
c1 for winid 11395600	_Anonymous	http://sc1.maps.live.com/mapcontrol.ashx?mkt=en-us&v=1.3.20071113130328.94	1204742188681	89	00	3
c1 for winid 11395600	_Anonymous	http://sc1.maps.live.com/mapcontrol.ashx?mkt=en-us&v=1.3.20071113130328.94	1204742188681	82	00	4
c1 for winid 11395600	GetManifestUrl	http://sc1.maps.live.com/mapcontrol.ashx?mkt=en-us&v=1.3.20071113130328.94	1204742188681	67	00	4
c1 for winid 11395600	_Anonymous	http://sc1.maps.live.com/mapcontrol.ashx?mkt=en-us&v=1.3.20071113130328.94	1204742188681	57	00	2
c1 for winid 11395600	_Anonymous	http://sc1.maps.live.com/mapcontrol.ashx?mkt=en-us&v=1.3.20071113130328.94	1204742188681	70	00	2
c1 for winid 11395600	_Anonymous	http://sc1.maps.live.com/mapcontrol.ashx?mkt=en-us&v=1.3.20071113130328.94	1204742188681	67	00	3
c1 for winid 11395600	Msn.VE.BrowserInfo	http://sc1.maps.live.com/mapcontrol.ashx?mkt=en-us&v=1.3.20071113130328.94	1204742188681	454	00	4

Clustering granularity presets

○ Coarse
● Medium
○ Fine-grained

Clustering parameters

Gap between clusters: 5.000 seconds

Cluster size threshold: 25 KB

Architecture of Doloto

1. [**training**] Runtime training to collect access profile

3. [**clustering**] Grouping related functions together

4. [**rewriting**] Function rewriting or “stubbing” for on-demand code loading

Inserting Function Stubs

```
var g = 10;
function f1(y){
    var x=g+1;
    …
    …
    …
    …
    …
    return …;
}
```

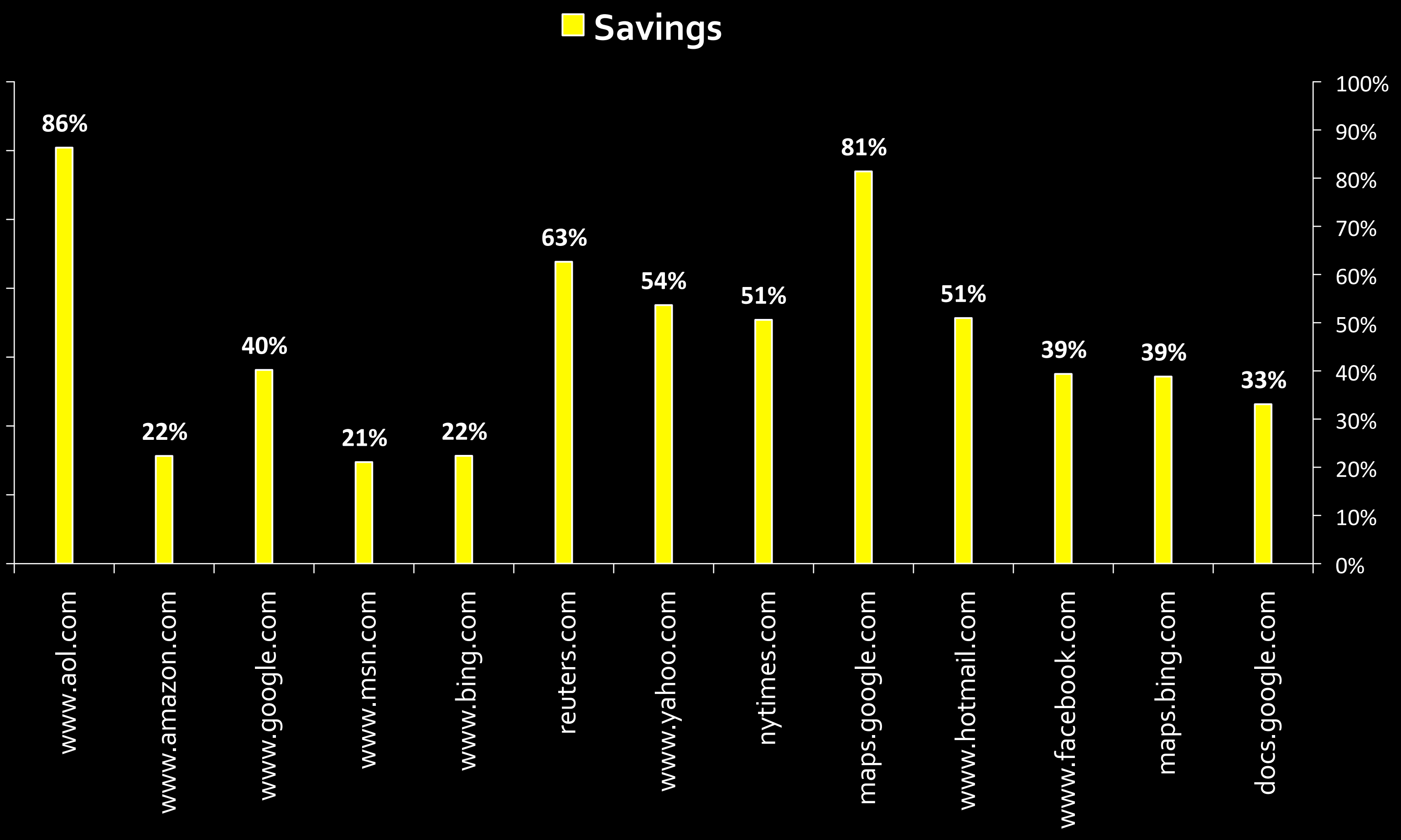
```
var g = 10;

var real_f1;
function f1(y) {
    if(!real_f1){
        var code = load("f1");
        real_f1 = eval(code);
        f1 = real_f1;
    }
    return real_f1.apply(this,
                         arguments);
}
```

```
eval($exp("f1"), "y");          // 22 bytes
```

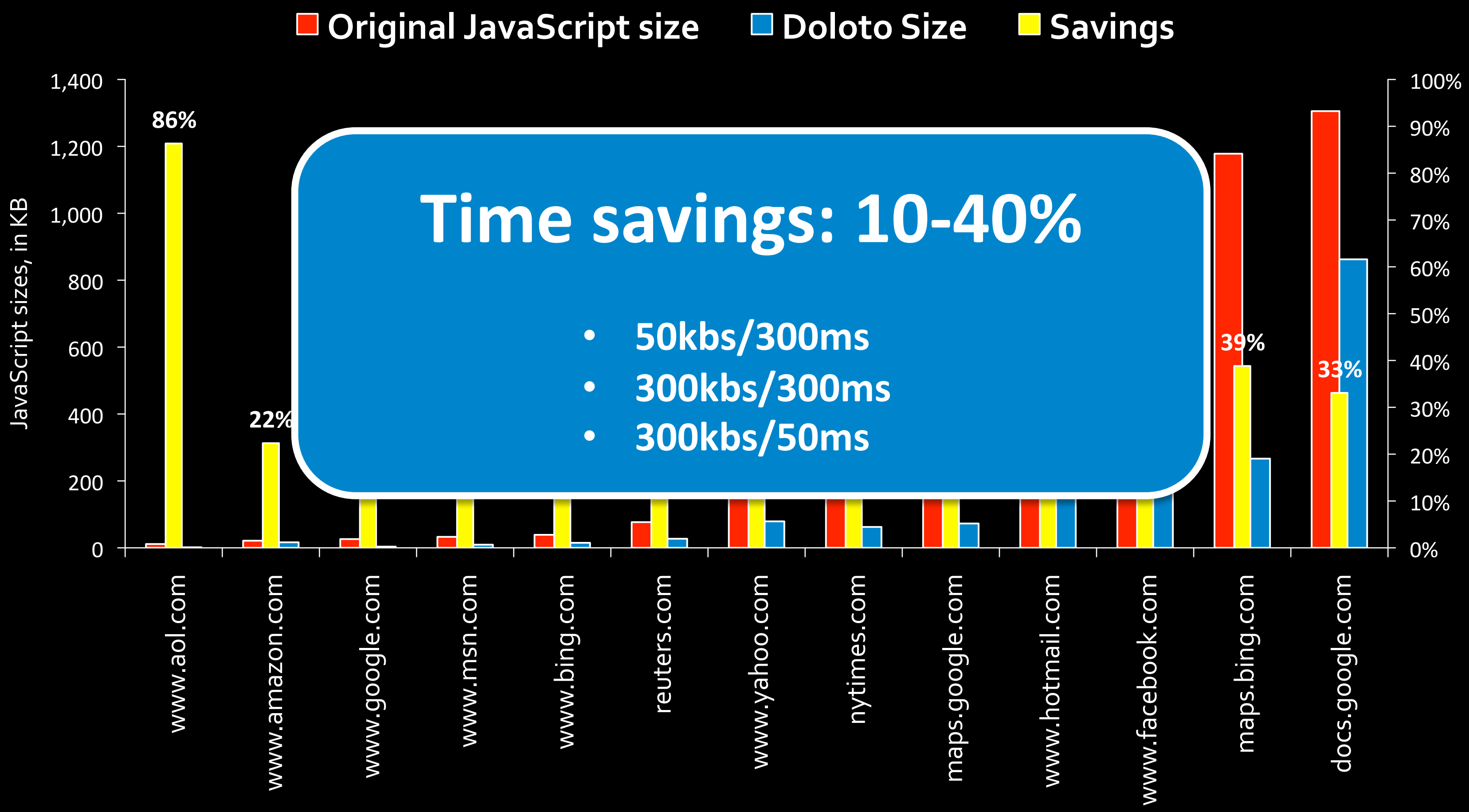
Profile applications using a proxy

Deploy rewritten code + cluster file to the server

Doloto Savings



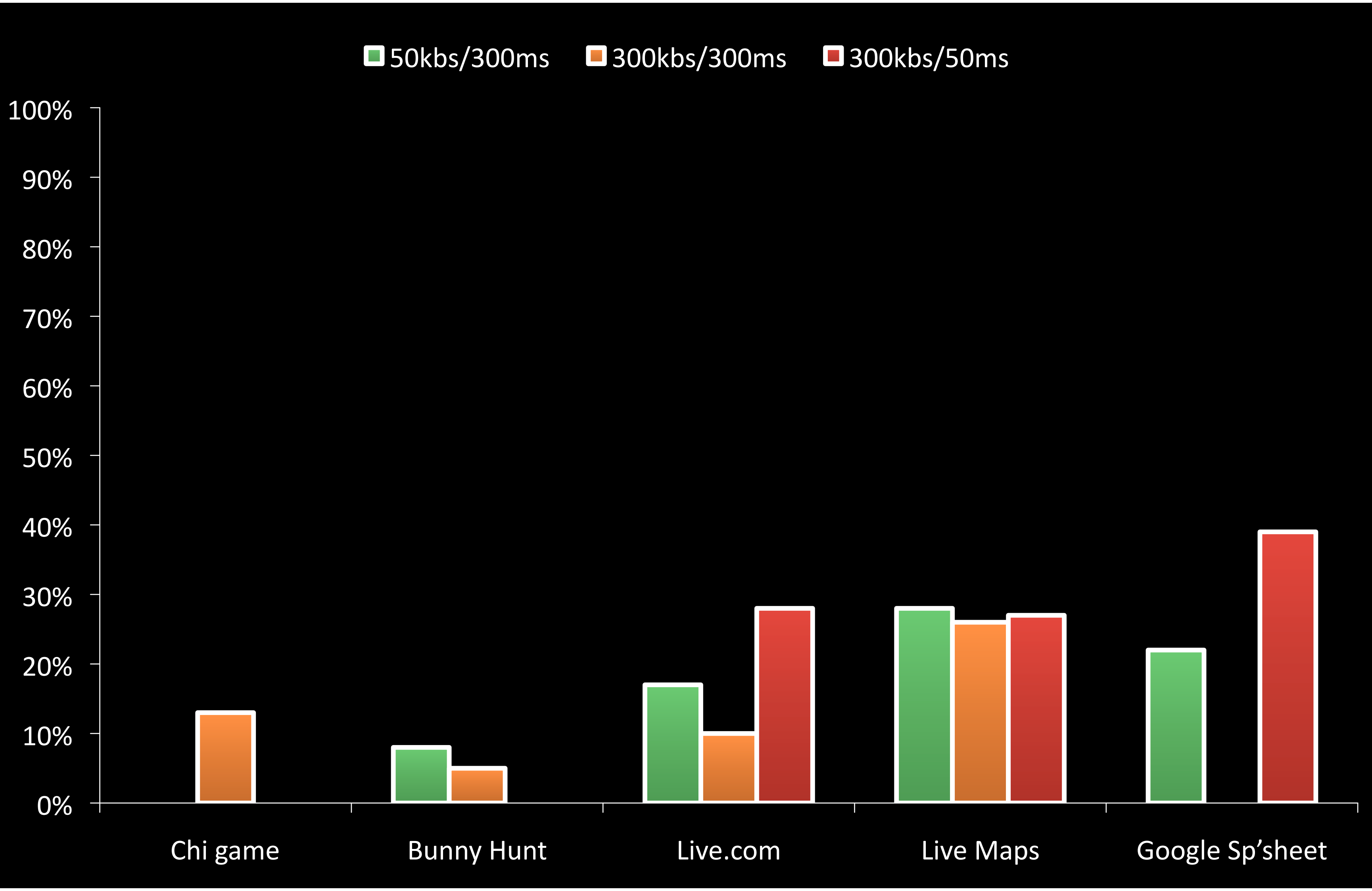
Doloto Savings



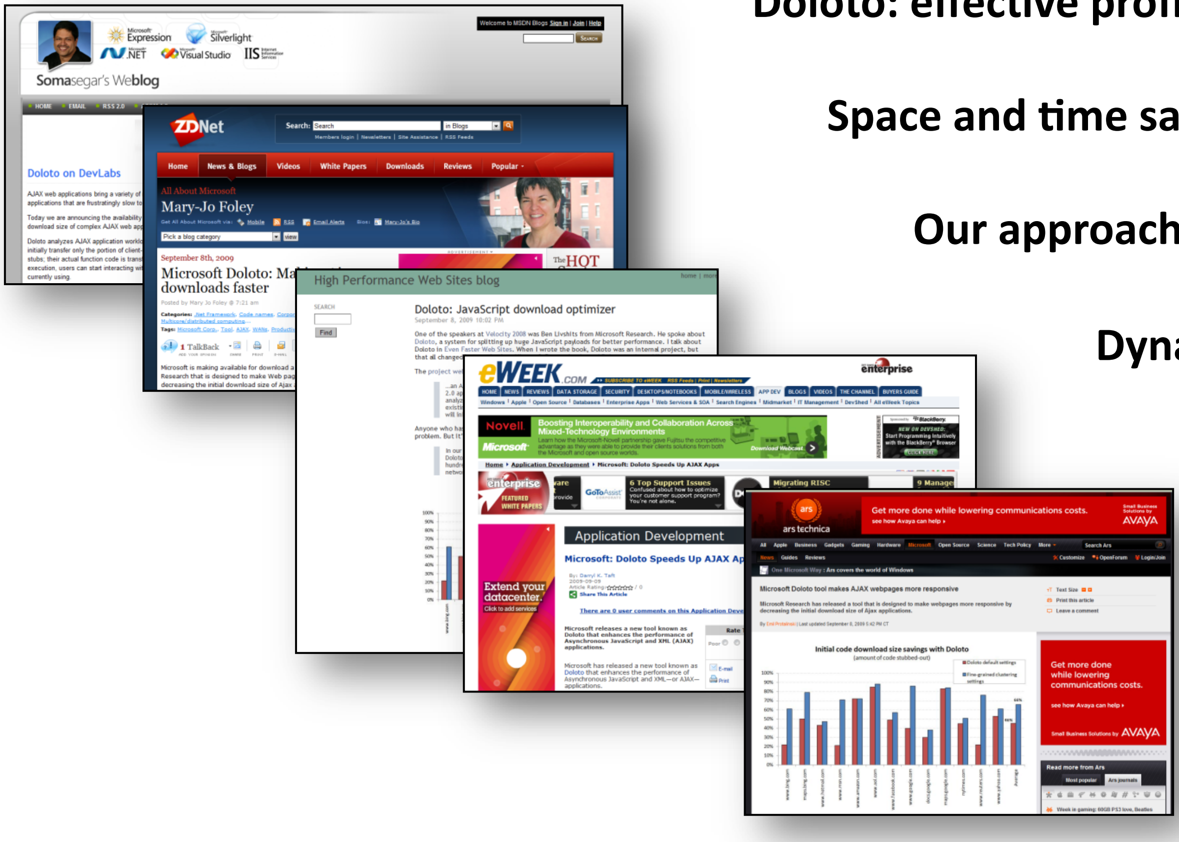
Time savings: 10-40%

- 50kbs/300ms
- 300kbs/300ms
- 300kbs/50ms

Runtime Savings with Doloto



Doloto: Conclusions



Somasegar's Weblog

Doloto on DevLabs

ZDNet

All About Microsoft

Mary-Jo Foley

September 8th, 2009

Microsoft Doloto: Ma

downloads faster

High Performance Web Sites blog

Doloto: JavaScript download optimizer

eWEEK.COM

Application Development

Microsoft: Doloto Speeds Up AJAX Ap

ars technica

Microsoft Doloto tool makes AJAX webpages more responsive

Microsoft Research has released a tool that is designed to make webpages more responsive by decreasing the initial download size of Ajax applications.

Initial code download size savings with Doloto

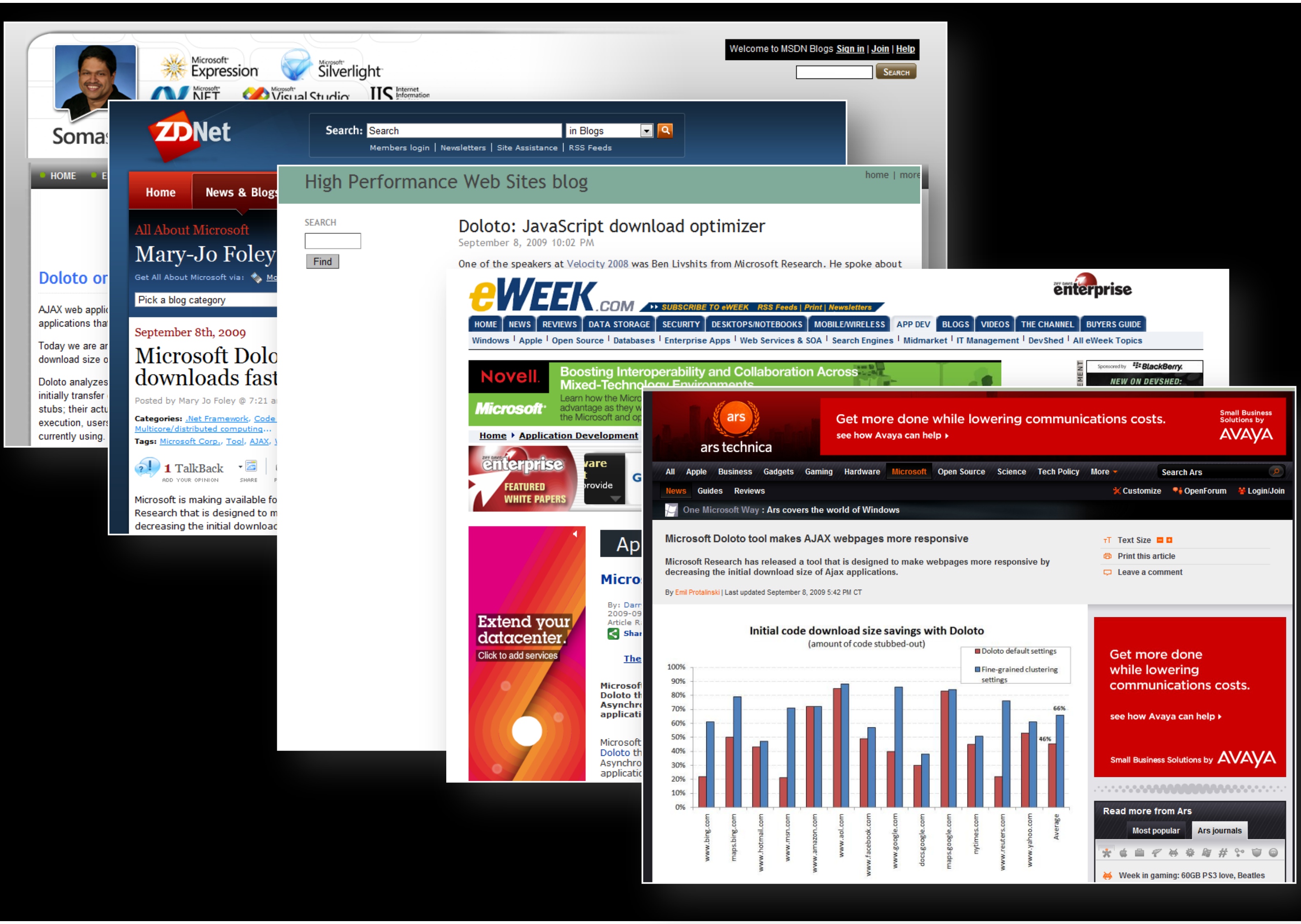
Doloto: effective profile-driven optimization

Space and time savings

Our approach is general: Silverlight

Dynamic code loading for future web apps

Press Coverage



Welcome to MSDN Blogs Sign in | Join | Help

Doloto or

AJAX web applic applications tha

Today we are ar download size o

Doloto analyzes initially transfer stubs; their actu execution, users currently using.

All About Microsoft

Mary-Jo Foley

September 8th, 2009

Microsoft Dolo downloads fast

Posted by Mary Jo Foley @ 7:21 a

Categories: .Net Framework, Code, Multicore/distributed computing...

Tags: Microsoft Corp., Tool, AJAX,

Microsoft is making available fo Research that is designed to m decreasing the initial downloac

High Performance Web Sites blog

Doloto: JavaScript download optimizer

September 8, 2009 10:02 PM

One of the speakers at Velocity 2008 was Ben Livshits from Microsoft Research. He spoke about

Home ▸ Application Development

One Microsoft Way : Ars covers the world of Windows

Microsoft Doloto tool makes AJAX webpages more responsive

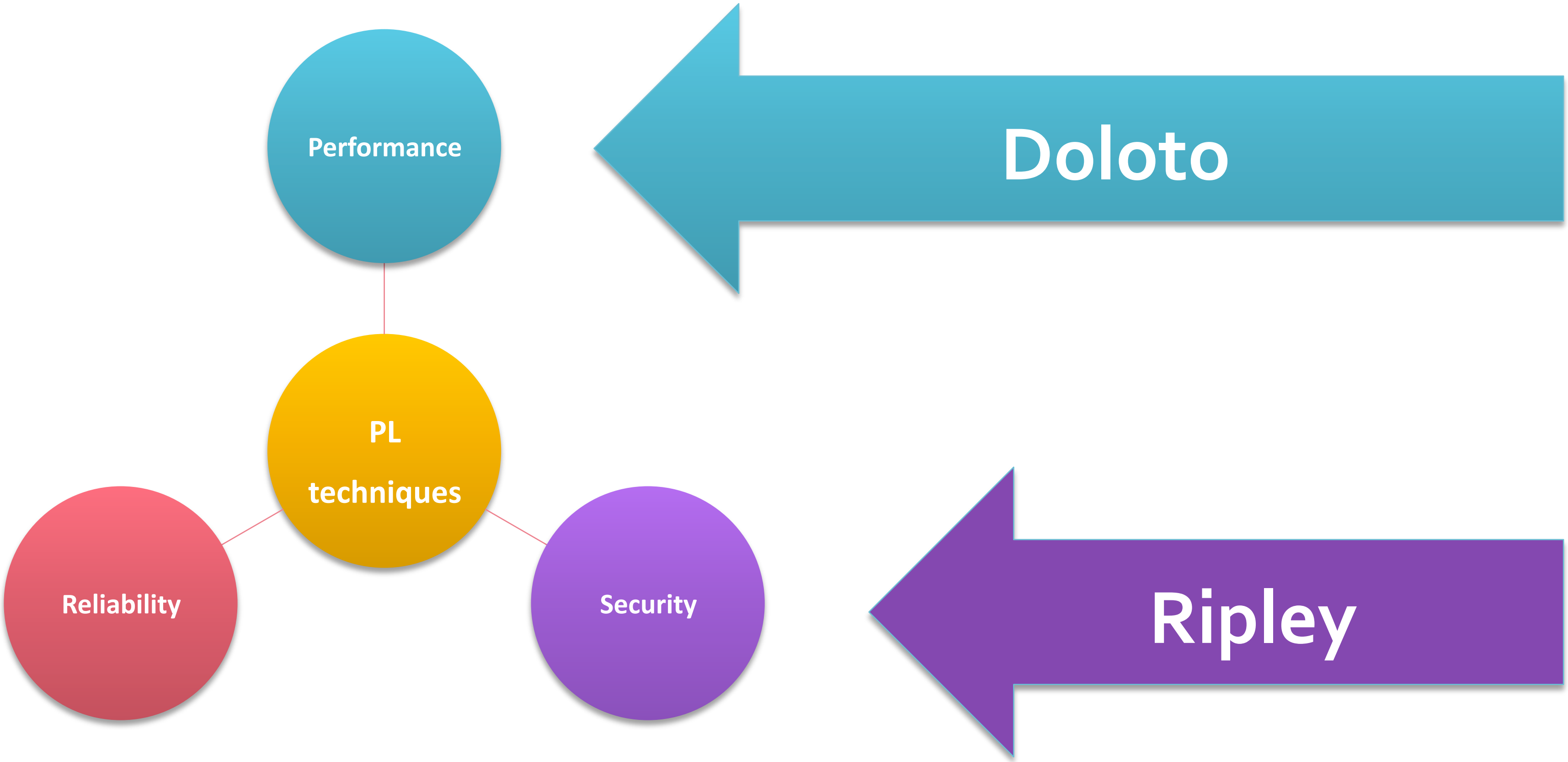
Microsoft Research has released a tool that is designed to make webpages more responsive by decreasing the initial download size of Ajax applications.

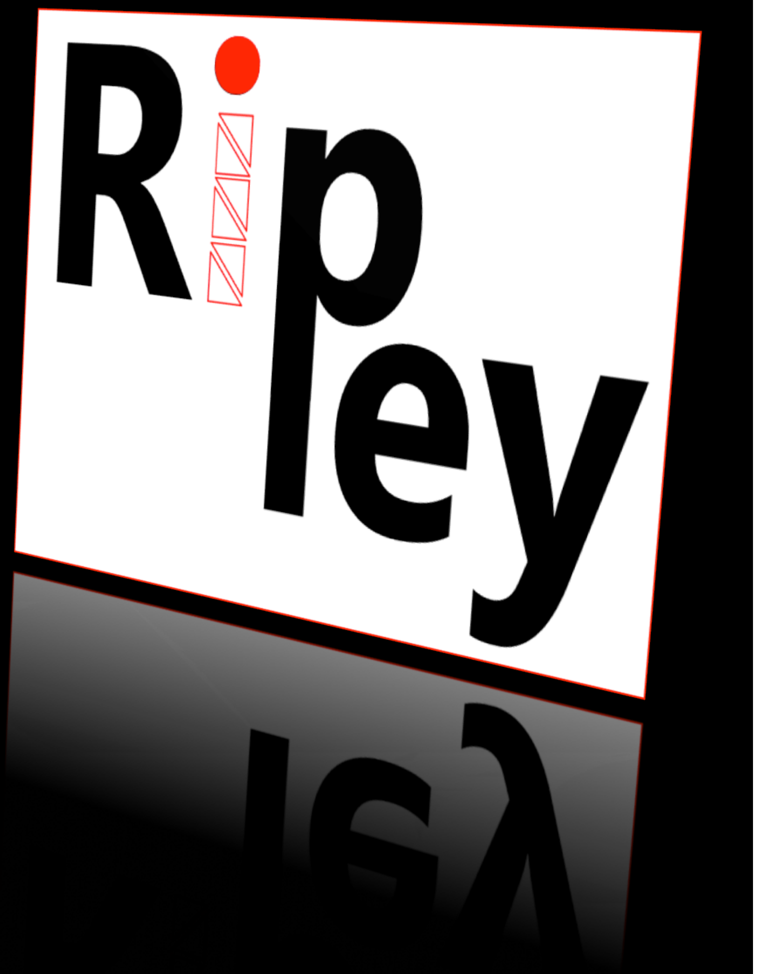
By Emil Protalinski | Last updated September 8, 2009 5:42 PM CT

Initial code download size savings with Doloto

(amount of code stubbed-out)

Outline of the Talk

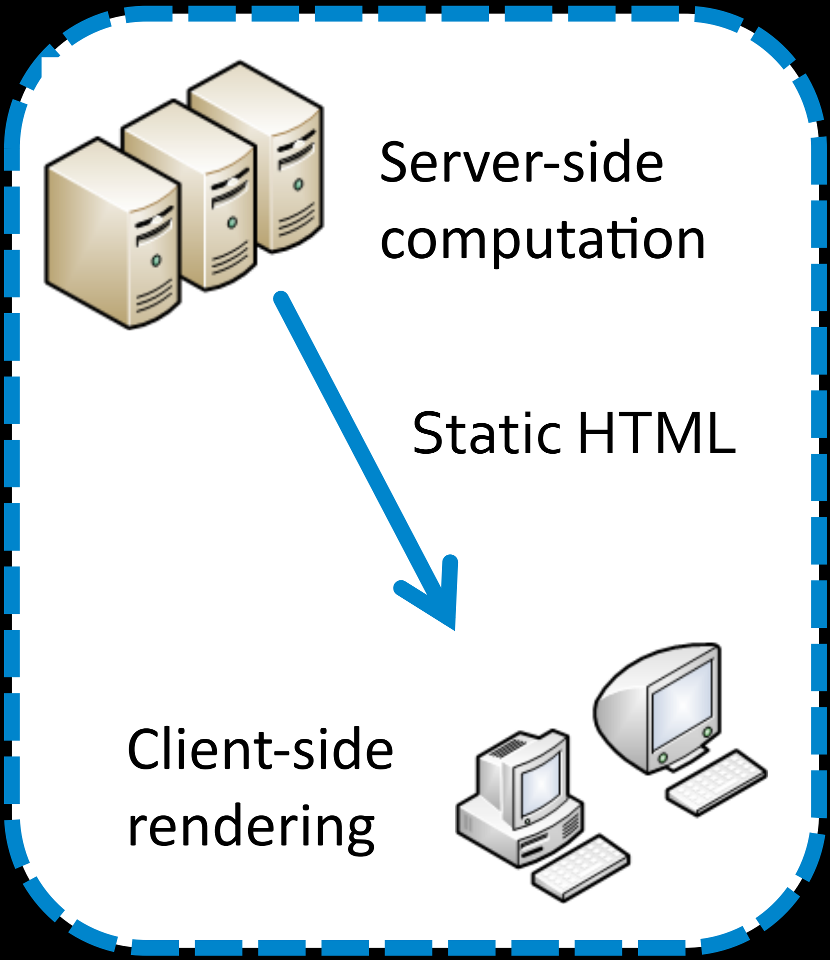




Automatically securing web 2.0 applications

through replicated execution

Web 1.0 → Web 2.0



Server-side computation

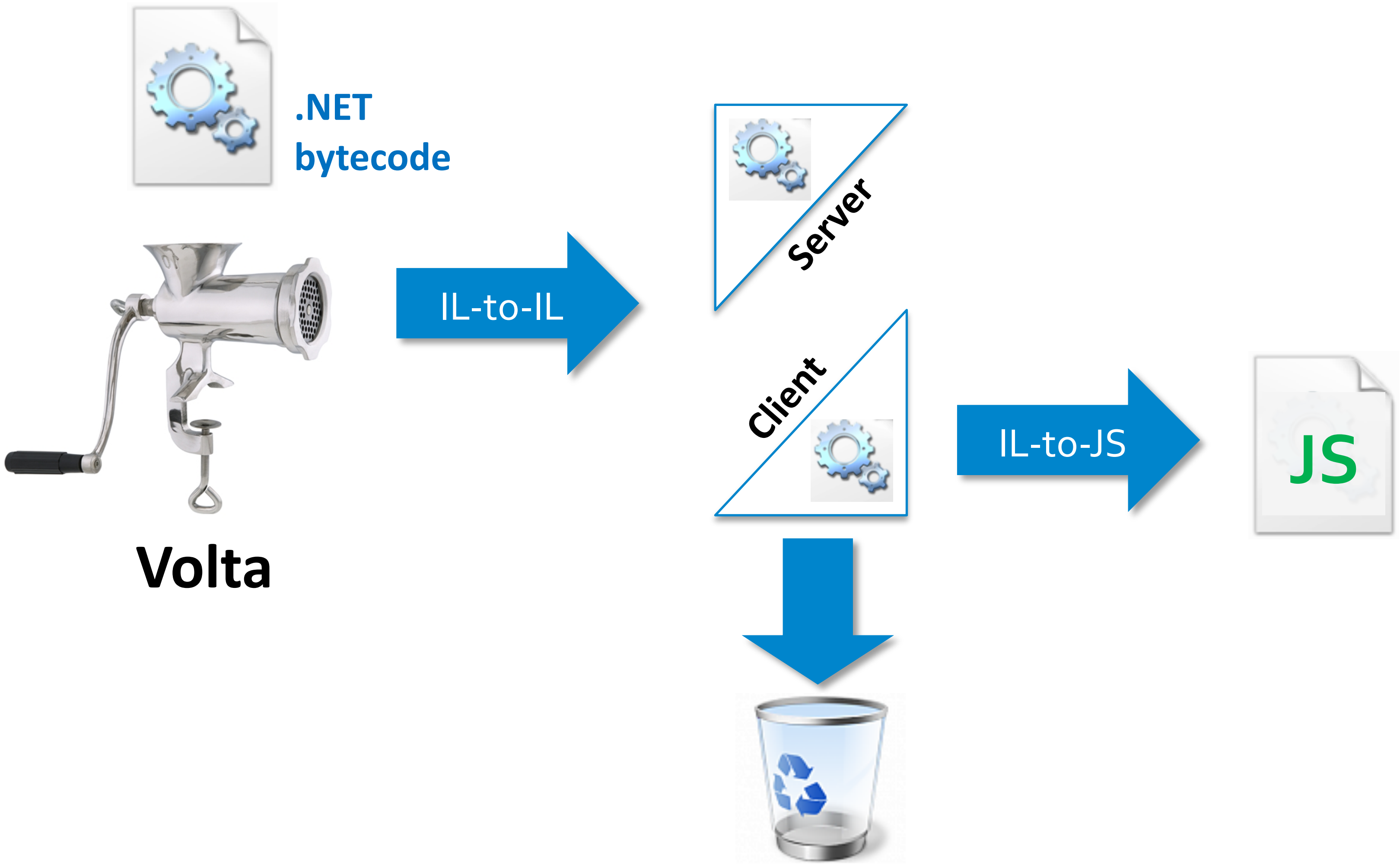
Static HTML

Client-side rendering

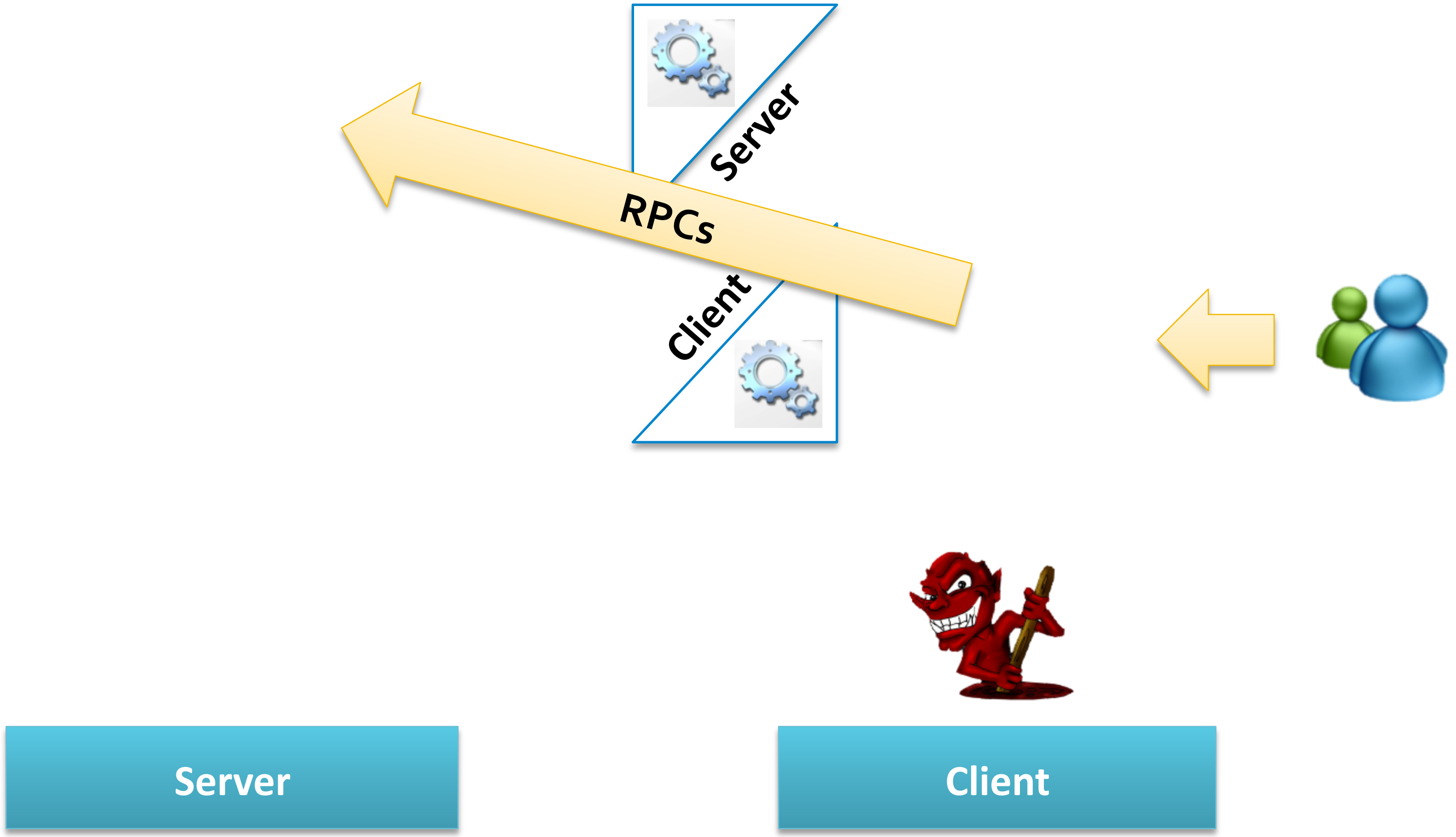
How Do We Program This Mess?

- Currently:
  - J2EE + JavaScript?
  - PHP + Flash?
  - ASP.NET + Silverlight?
- One alternative:
  - Distributing compilers
  - Volta, GWT, Hops, Links

The Volta Distributing Compiler Illustrated

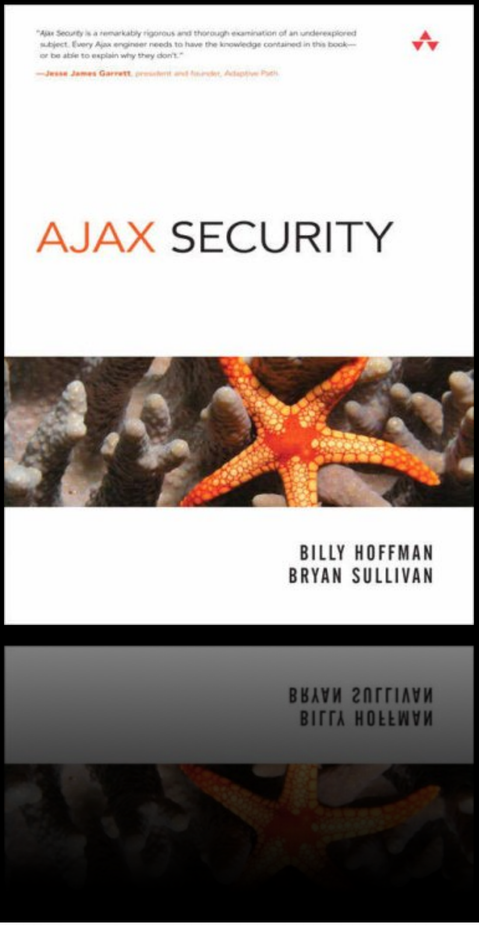


The Volta Distributing Compiler: Deployment



Web Developer’s Mantra

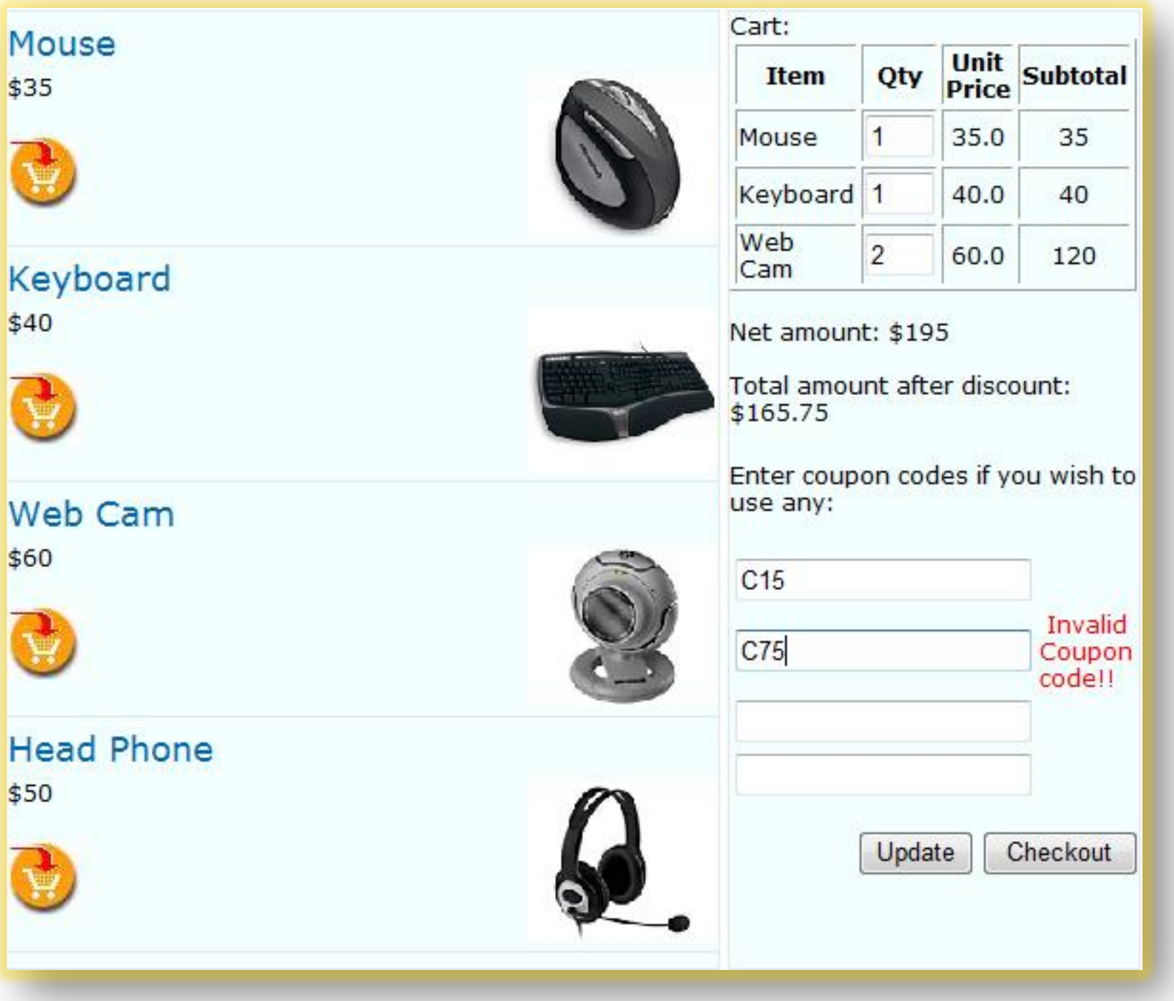
Thou shall not trust the client


"Ajax Security is a remarkably rigorous and thorough examination of an underexplored subject. Every Ajax engineer needs to have the knowledge contained in this book—or be able to explain why they don't."
—Jesse James Garrett, president and founder, Adaptive Path
AJAX SECURITY
BILLY HOFFMAN
BRYAN SULLIVAN

C:\Users\livshits\Desktop\chert.jpg - Windows Internet Explorer
Favorites
Add

- ✗ No data integrity
- ✗ No code integrity

AJAX-based Shopping Cart



Mouse

$35

Keyboard

$40

Web Cam

$60

Head Phone

$50

Cart:

Item	Qty	Unit Price	Subtotal
Mouse	1	35.0	35
Keyboard	1	40.0	40
Web Cam	2	60.0	120

Net amount: $195

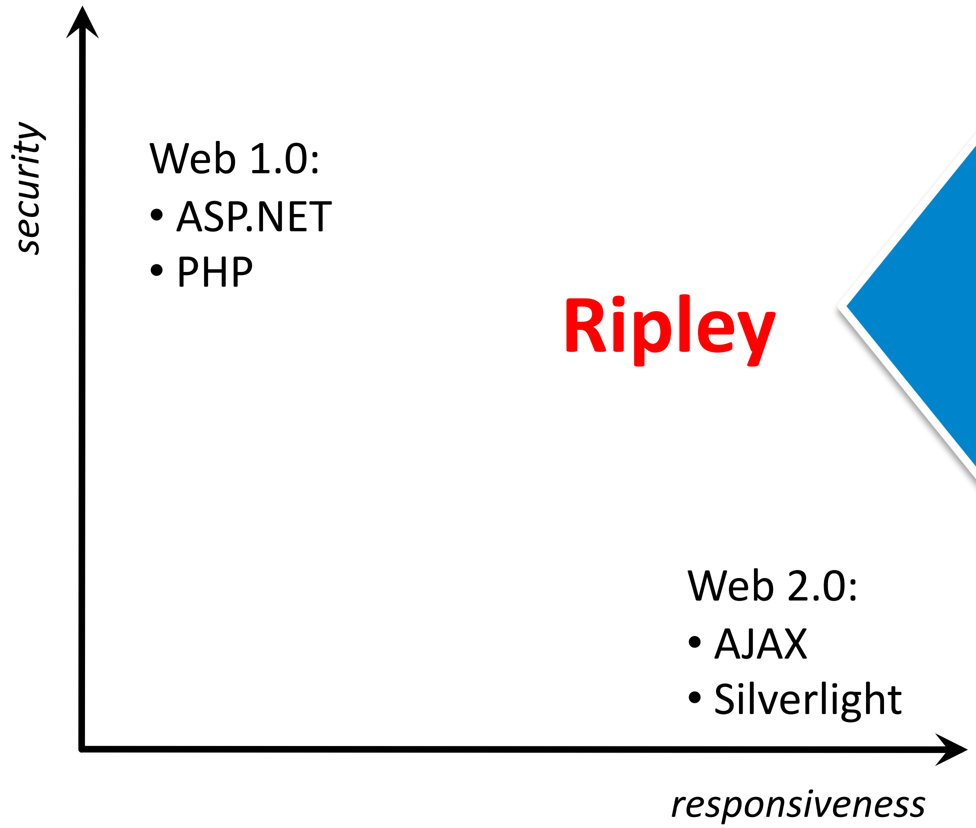
Total amount after discount: $165.75

Enter coupon codes if you wish to use any:

C15

C75 Invalid Coupon code!!

Security vs. Performance



Web 1.0:

- ASP.NET
- PHP

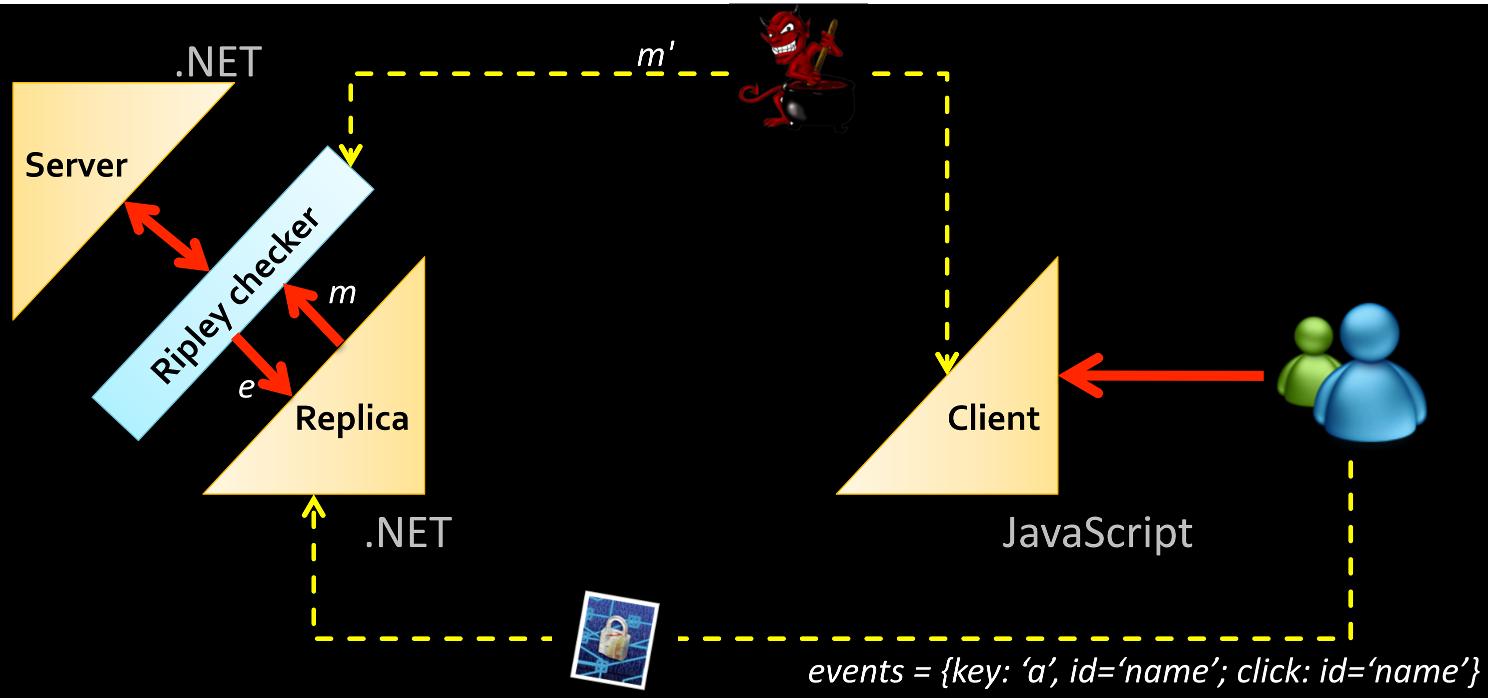
Ripley

Web 2.0:

- AJAX
- Silverlight

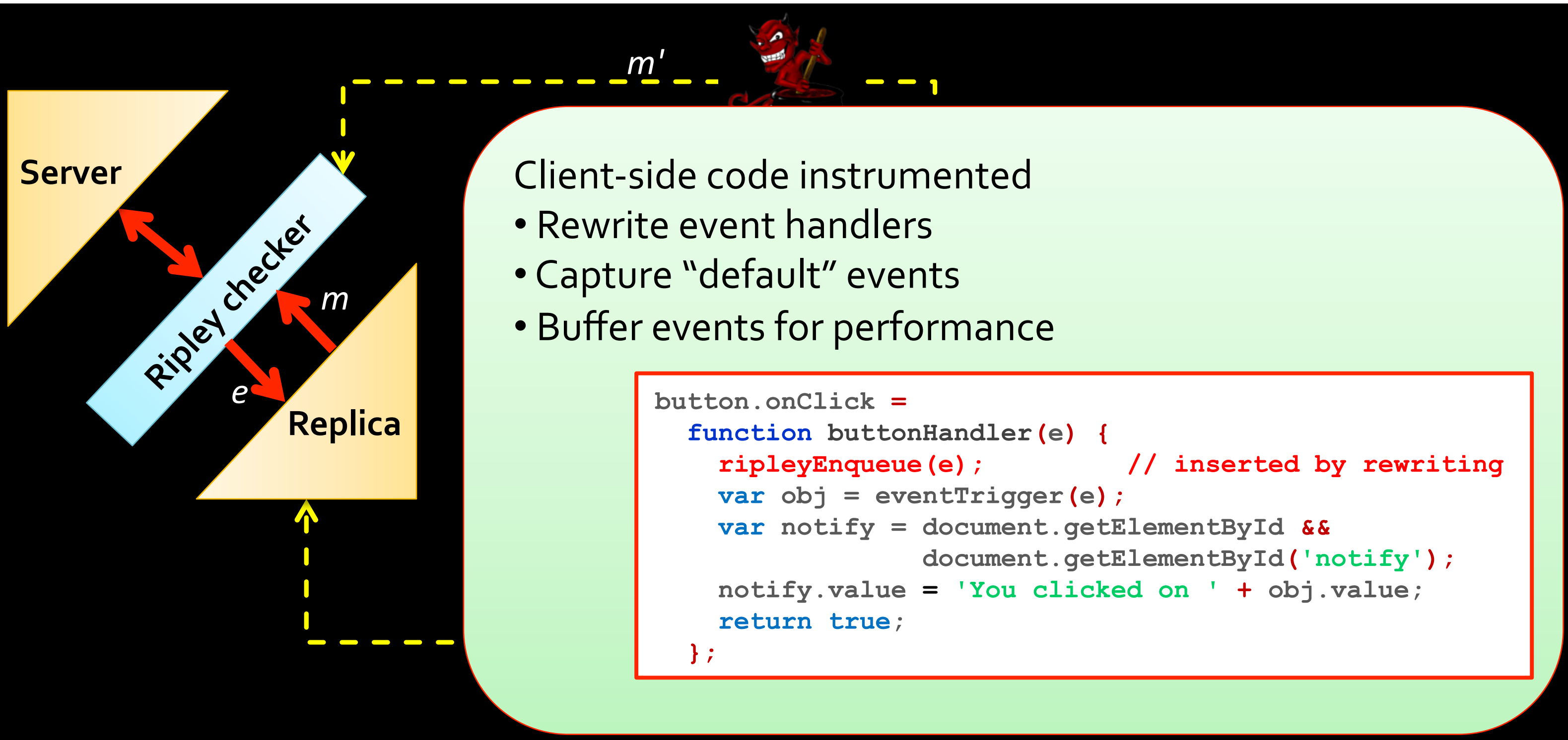
With Ripley, placing computation on the client ***does not reduce*** computational integrity

Ripley Architecture



1. Keep a replica of the client code
2. Capture user events & transmit to server for replay
3. Compare server and client results

Ripley Architecture



Client-side code instrumented

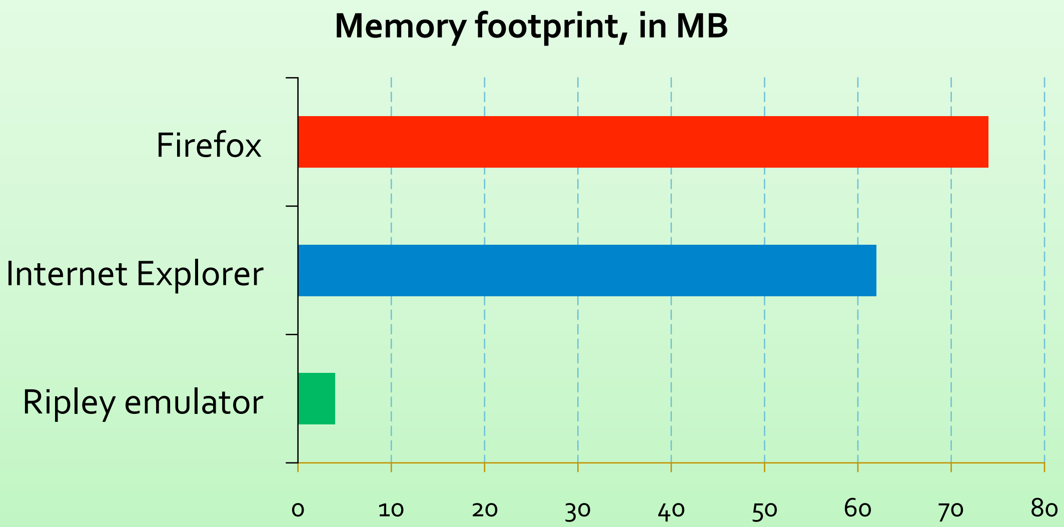
- Rewrite event handlers
- Capture “default” events
- Buffer events for performance

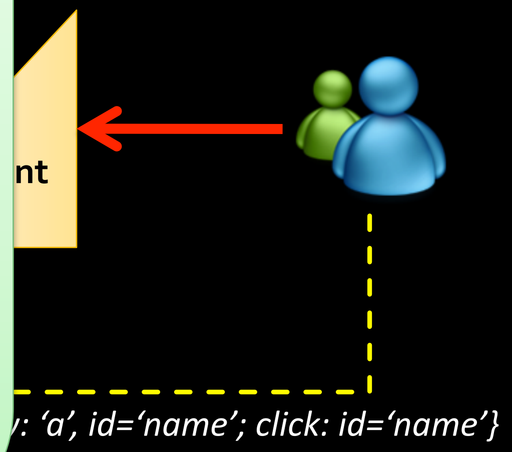
```
button.onClick =
  function buttonHandler(e) {
    ripleyEnqueue(e);          // inserted by rewriting
    var obj = eventTrigger(e);
    var notify = document.getElementById &&
                 document.getElementById('notify');
    notify.value = 'You clicked on ' + obj.value;
    return true;
  };
```

1. Keep a replica of the client code
2. Capture user events & transmit to server for replay
3. Compare server and client results

Ripley Architecture

- Run replica in a Ripley emulator
- Run in .NET, not in JavaScript, 100x speed increase

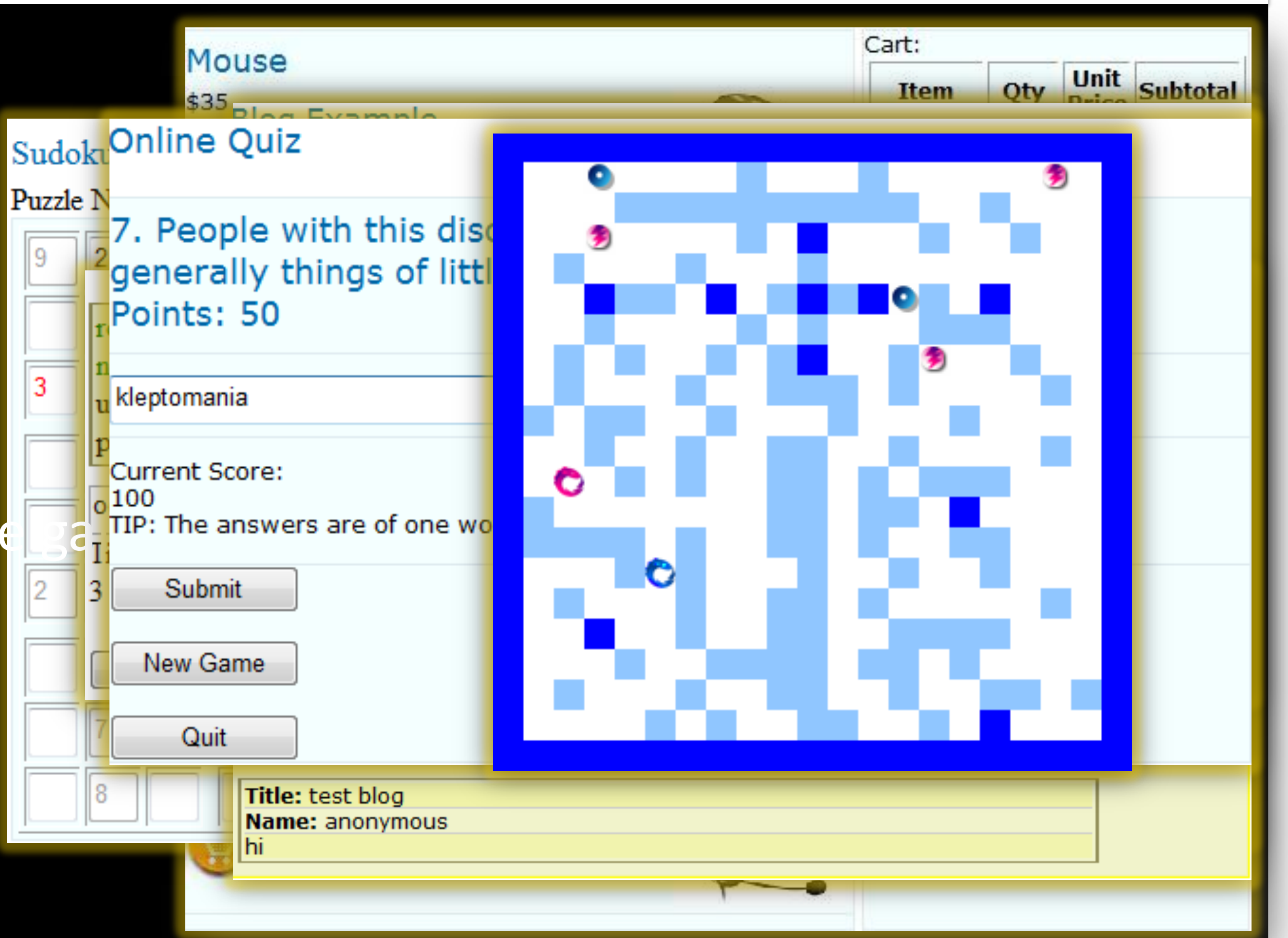




1. Keep a replica of the client code
2. Capture user events & transmit to server for replay
3. Compare server and client results

Ripley Applications

- ✓ Shopping cart
- ✓ Sudoku
- ✓ Blog
- ✓ Speed typing
- ✓ Online Quiz
- ✓ Distributed online ga



7. People with this disc
generally things of littl
Points: 50

kleptomania

Current Score:
100
TIP: The answers are of one wo

Title: test blog
Name: anonymous
hi

Performance Overhead Summary

Network:

- 2-3 bytes per user event (key press, mouse, etc.)
- Event stream compresses extremely well

CPU:

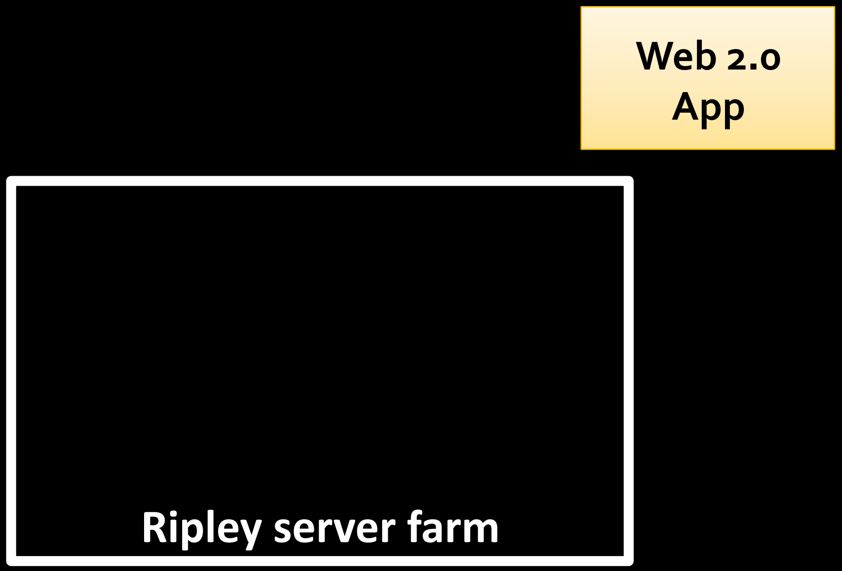
- Client: Several *ms* of overhead added for event capture
- Server: Several *ms* for server-side checking

Memory:

- About 1 MB per connected client
- Can scale to 1,000’s of clients per server

Ripley: Vision for the Future

- Security by construction



Web 2.0 App

Ripley server farm

For More...

Doloto|Ripley MSR _

Doloto: Code

Network-Bound

Ripley: Automatically Securing Distributed W

Applications Through Replicated Exec

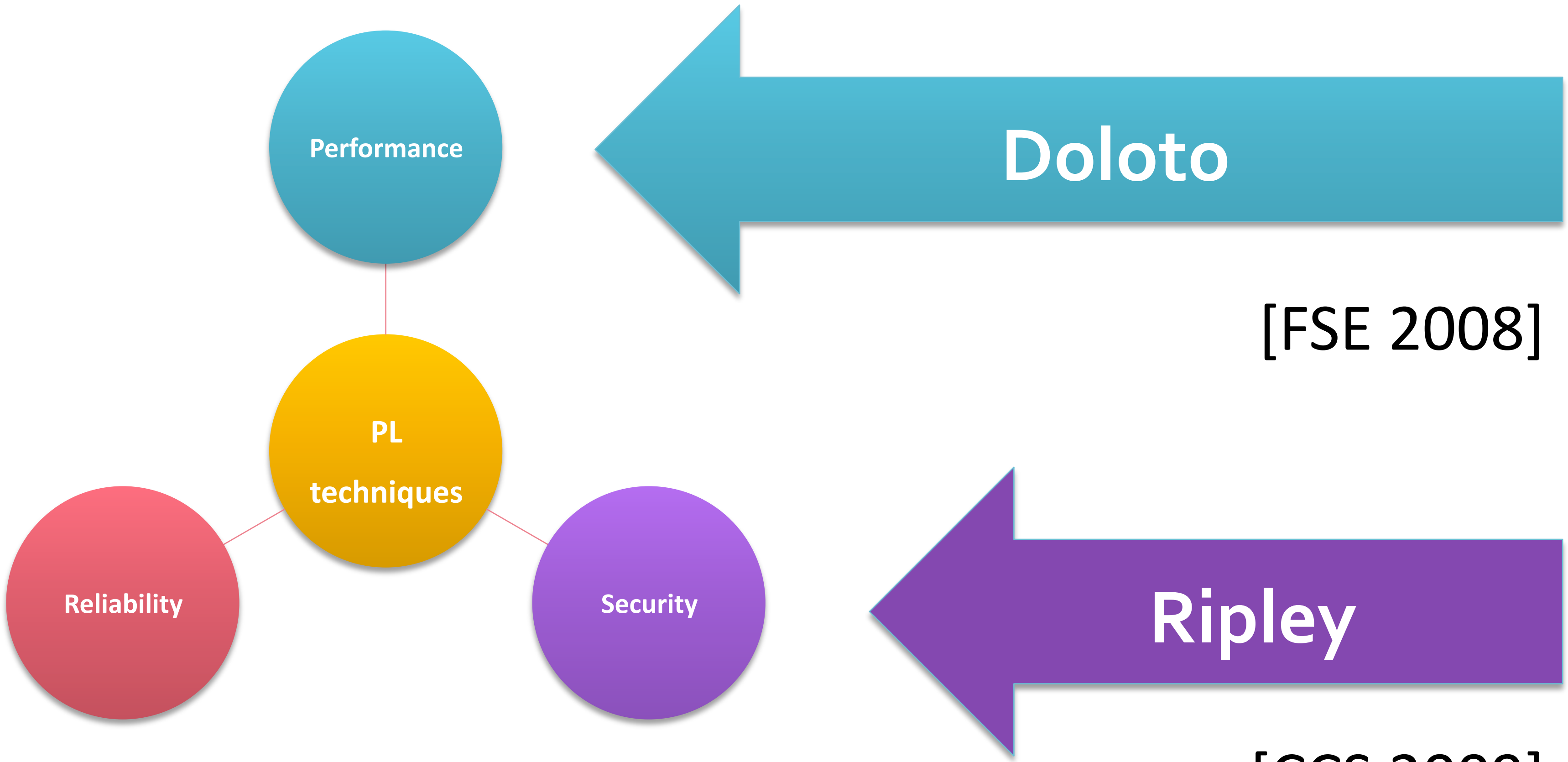
K. Vikram
Cornell University

Abhishek Prateek
IIT Delhi

Abstract

Rich Internet applications are becoming in
by the popularity of AJAX or Web 2.0 app
Facebook, and many others. A typical
and XML) application consists of a
ASP.NET and a client-side compo
cation is more performant and r
client, and thus avoids unnec
However, once a port
easily subvert the clie
server state. In this
to *automatically*
a copy of the
is transfer
the com
repli

Summary



Performance

PL techniques

Reliability

Security

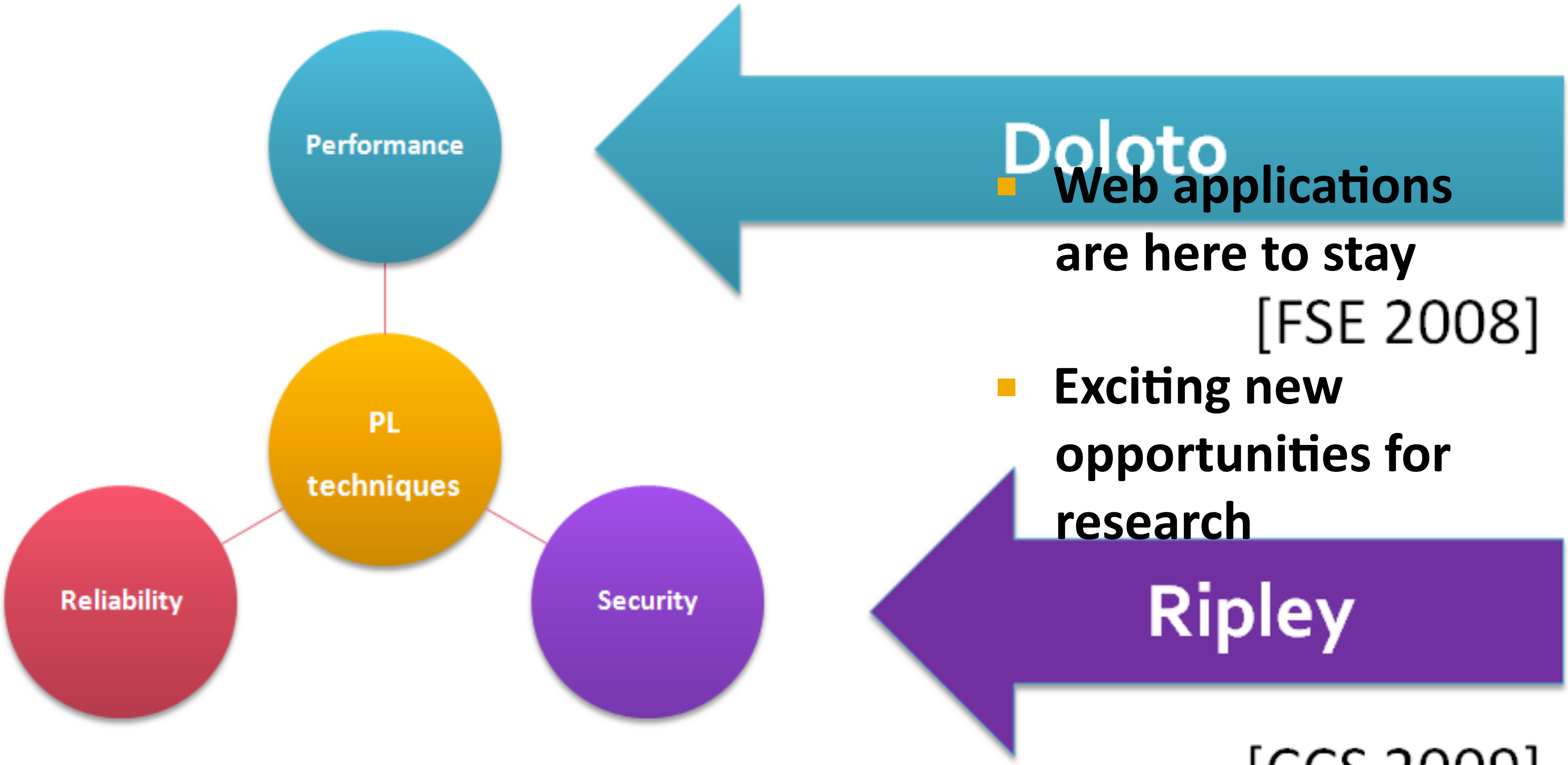
Doloto

Ripley

[FSE 2008]

[CCS 2009]

Call to Arms



- Web applications are here to stay

[FSE 2008]

- Exciting new opportunities for research

[CCS 2009]