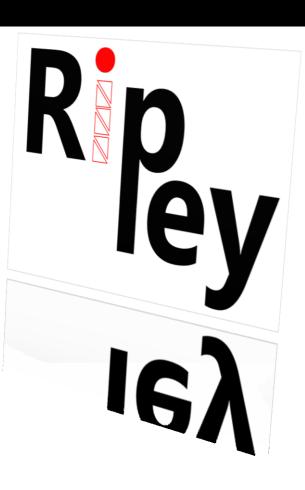
SECURING WEB 2.0 APPLICATIONS THROUGH REPLICATED EXECUTION

K. Vikram <u>Corn</u>ell University Abhishek Prateek IIT Delhi

Microsoft Research



2

Web 2.0 is Upon Us





Web 1.0 \rightarrow Web 2.0



Server-side

Advantage of the AJAX model:

greater application responsiveness

Client-side rendering

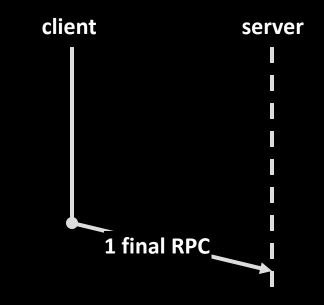


Motivation



AJAX-based Shopping Cart (Fantasy)

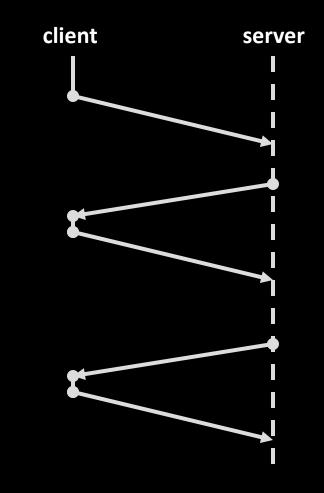
Favorites 🏾 🏉 C:\Users\livshits\Desktop\shot.jpg					
Mouse	Cart:	Cart:			
\$35	Item	Qty	Unit Price	Subtotal	
	Mouse	1	35.0	35	
	Keyboard	1	40.0	40	
Keyboard	Web Cam	2	60.0	120	
Web Cam	\$165.75	Enter coupon codes if you wish to			
\$60	C15	C15			
S 2	C75	C75			
Head Phone \$50		Upda	te (Checkout	





Shopping Cart (Reality)

Marian	Cart:			
Mouse	Item	Qty	Unit Price	Subtotal
1	Mouse	1	35.0	35
9	Keyboar	d 1	40.0	40
Keyboard	Web Cam	2	60.0	120
Web Cam \$60	\$165.75 Enter cou	Enter coupon codes if you wish t use any:		
	C75	C75		
•	9			
Head Phone				



Motivation



Web Developer's Mantra

**

Thou shall not trust the client



- X No data integrity
- 🔀 No code integrity

*Apit Security is a remarkably rigorous and thorough examination of an underexplored subject. Every Apic engineer needs to have the knowledge contained in this book or be able to explain why they don't."
—Jesse James Garrett, prevalent and faileder, Astiptive Poti-

AJAX SECURITY



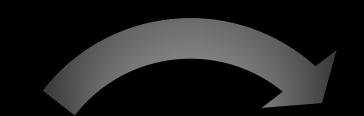
BILLY HOFFMAN Bryan Sullivan

Motivation



Tension Headaches





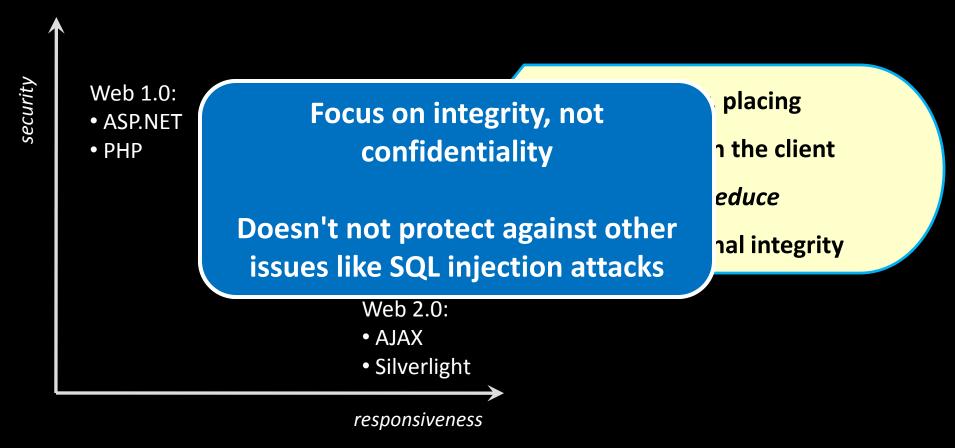
Move code to client for performance Move code to the server for security







Security vs. Performance

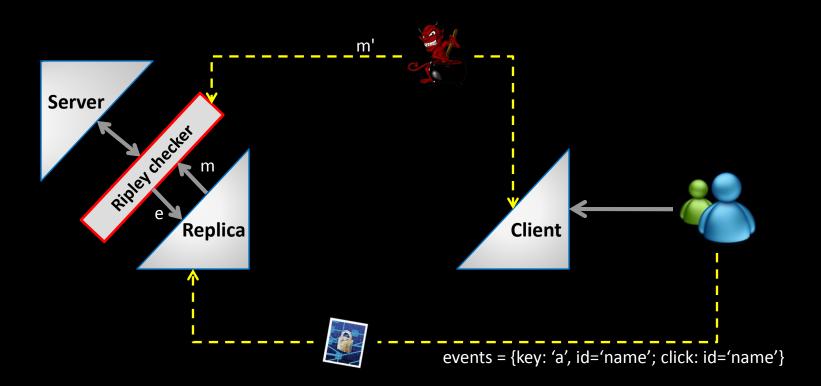


Motivation

Architecture



Ripley Architecture

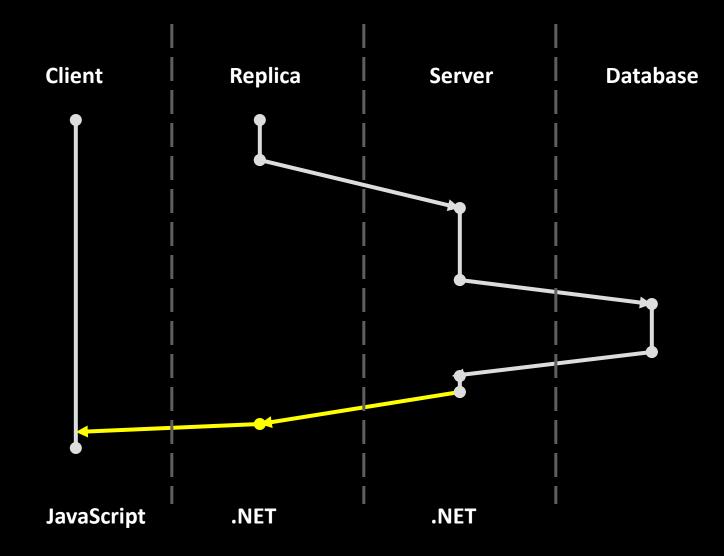


- 1. Keep a replica of the client code
- 2. Capture user events & transmit to server for replay
- 3. Compare server and client results

gorithms



Zero-latency RPCs







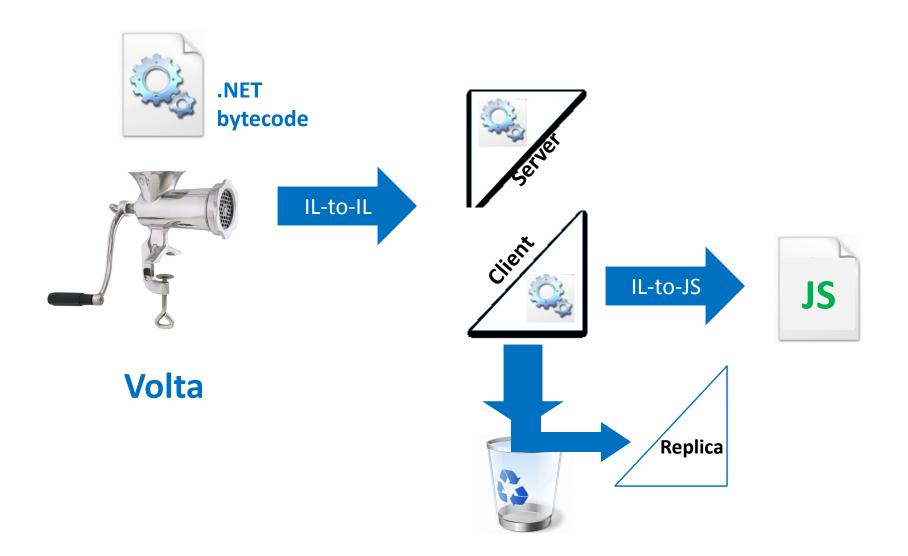
Seems Too Much Like Magic. Is this Feasible?

- Create deterministic replay system
 - How to we replicate JavaScript code?
 - Cross-browser differences?
 - Non-determinism?
- How do we scale it?
 - Replica overhead on server
 - Hundreds of concurrent replicas



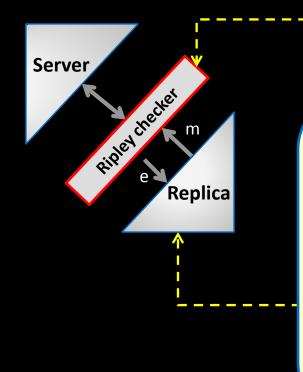


The Volta Distributing Compiler Illustrated





Ripley Architecture



- Client-side code instrumented
 - Rewrite event handlers
 - Capture "default" events
- Network overhead
 - Buffer events for performance
 - Piggy-back on existing RPCs

1. Keep a replica of the c

gorithms

2. Capture user events & transmit to server for replay

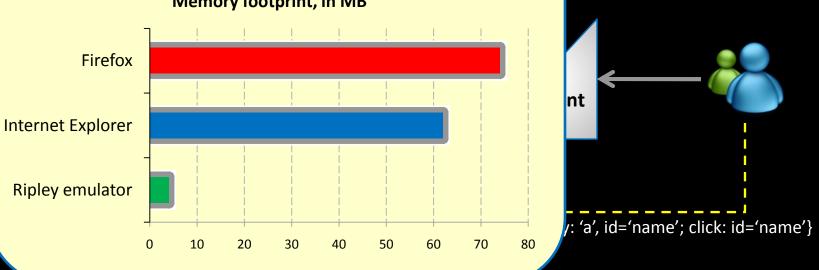
m

3. Compare server and client results



Ripley Architecture

- Run replica in a Ripley emulator
- In .NET, not in JavaScript, 10-100x speed increase



Memory footprint, in MB

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

gorithms

Experiments



Ripley Applications

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

Mouse		Cart:			
\$35 Plac Example	-	Item	Qty	Unit Duice Subtotal	
okuOnline Quiz					
e N	•			۳	
= 7. People with this disc	9			- - -	
² generally things of littl					
Points: 50			0		
n			۲		
u kleptomania					
P					
Current Score: 0100	0				
TIP: The answers are of one wo					
Ti 3 CSubmit	0				
New Game					
7 Quit					
8 Title: test blog					
hi					
U					





Performance Overhead: Volta Benchmarks

Network:

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



Memory:

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



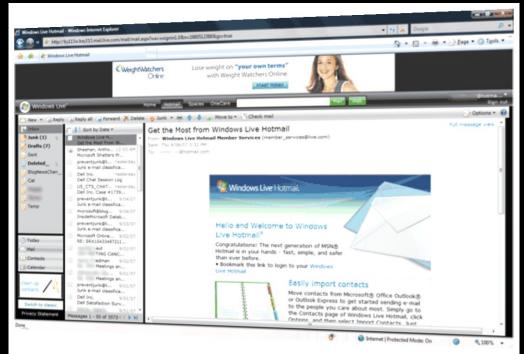
CPU:

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





Replicating Hotmail



• Hotmail size

xperiments

- 793 KB download
- 703 KB JavaScript
- 31,000+ lines of code

- 10 minutes of normal use
- Requests: 617 KB
- Responses: 3,045 KB



Replicating Hotmail



Ripley traffic:

- 491 keyboard & mouse events
- 1.4% without compression (8.6 KB)
- 0.4% otherwise (2.8 KB)



Memory:

- DOM state in memory: 350 -- 450 KB
- JavaScript heap state: 1.3 MB
- < 1.75 MB in total
- Can scale up to hundreds of clients



CPU overhead small:

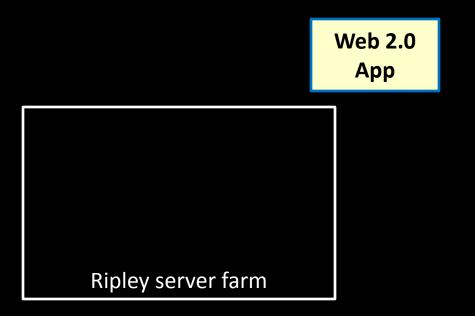
- Most: < 15 *ms*
- Email message processing: 125 ms
- Most time spent in HTML rendering and data marshaling code





Ripley: Vision for the Future

• Secure-by-construction Software + Services



Contact us

Ben Livshits (livshits@microsoft.com)

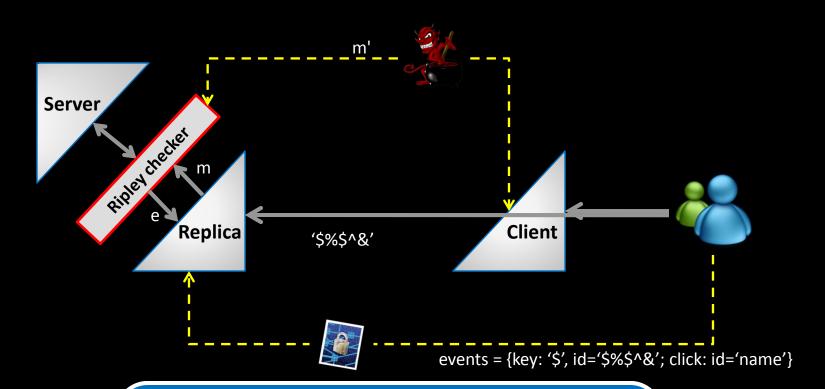
Microsoft Research Ripley project







Malicious Event Stream



Every attack against integrity of the Ripley-protected application was possible against the standalone app

