

Comet: Low Latency Data For Browsers


Alex Russell

alex@jot.com

Project Lead, The Dojo Toolkit

Please Ask Questions and Interrupt

- ▶ These slides are online at:
 - ▶ <http://alex.dojotoolkit.org>
- ▶ But you didn't come to ETech to read slides

The image features three vertical rectangular bars of varying shades of green. The central bar is the darkest, while the two flanking bars are lighter. The text is centered on the darkest bar.


Ajax is *me*
driven



Social apps
are also driven
by *others*

Eyes On The User

- ▶ The goal is *responsiveness*
- ▶ Ajax improves responsiveness for:
 - ▶ Single-user CRUD
 - ▶ Write-only interactions
 - ▶ Read-only apps where large lag is acceptable
- ▶ The web is inherently multi-user
- ▶ Ajax is only half the answer



To any user,
the server *is*
other users

The Multi-User Web

- ▶ Single interaction updates are not enough
- ▶ Users in the same “space” need live updates of:
 - ▶ Their own changes
 - ▶ The changes others make
- ▶ Updates to context affect available actions
- ▶ Stale context may mean the wrong decision



If the web is a
conversation...

The image features three vertical rectangular bars of varying shades of green. The central bar is the darkest, while the two flanking bars are lighter. The text is centered on the darkest bar.

...then stale
context kills

Latency Matters


- ▶ Conversation mediums are defined by latency, interrupt, and bandwidth
 - ▶ Snail-mail
 - ▶ Email
 - ▶ IRC
 - ▶ SMS
 - ▶ IM
 - ▶ Phone
 - ▶ Face-to-face

Example: Wikis As Conversations


- ▶ Wikis are conversation enablers
 - ▶ Traditionally medium-to-high latency
 - ▶ Not well suited to high-volume changes
- ▶ Locking/overwrite issues
- ▶ Ajax allows more context to go stale
 - ▶ What is changing in the Wiki while I edit?
 - ▶ Who wants to break my lock?
 - ▶ Have attachments been added?
 - ▶ Is the text of the page itself changing?

Conversations Are Ordered Events

- ▶ Granular interfaces require granular events
 - ▶ Granular conversations are more immediate
 - ▶ IM vs. Email
- ▶ Social apps are event busses
- ▶ Social web apps just batch changes today
- ▶ No effective way to “subscribe” to server events today
- ▶ To fix the context, syndicate the events
- ▶ Does “SOA” ring a bell? How about JMS?



Broadcast Is Synchronization For
“Shared Nothing”





Comet

Comet: Server Push Data

- ▶ New term, old tech
- ▶ Unburdened by previous definitions and tools
- ▶ Long-lived HTTP connections instead of polling
- ▶ Similarities to Ajax:
 - ▶ No plugins
 - ▶ “Plain old” HTTP
 - ▶ Asynchronous
 - ▶ Broad browser support
 - ▶ Payload can be anything textual

Applications Implementing Comet

- ▶ GMail + GTalk
- ▶ JotLive
- ▶ Renkoo
- ▶ Meebo
- ▶ cgi:irc
- ▶ KnowNow
- ▶ others?

Why Now?



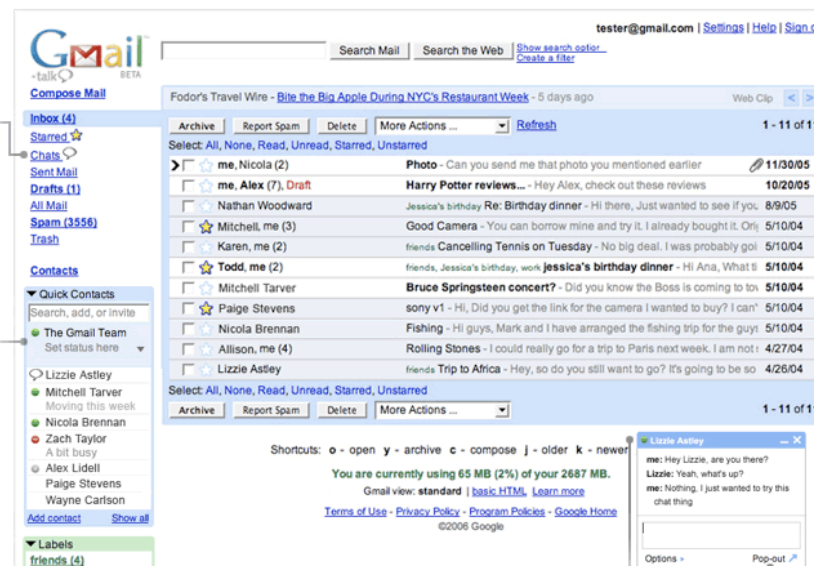
Live, group note-t
Five people in a meeting? Take one set of



Check out the latest buzz: our new Gmail chat features.

Chat History
Chats can be saved and searched for just like email messages

Quick Contacts
The people you email most often show up automatically. Customize this area to email and IM the friends you want with just one click



Chat Now you can chat with your friends from directly within Gmail

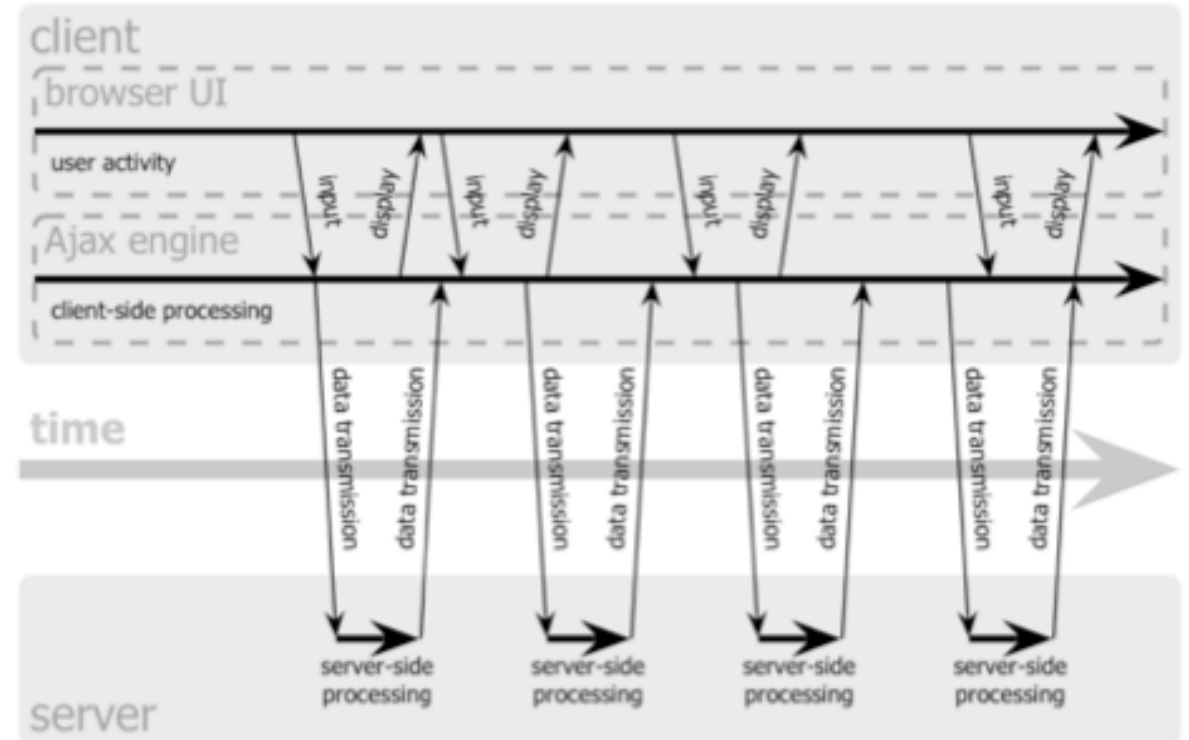
Pop-out Chats Give your chat session its own window



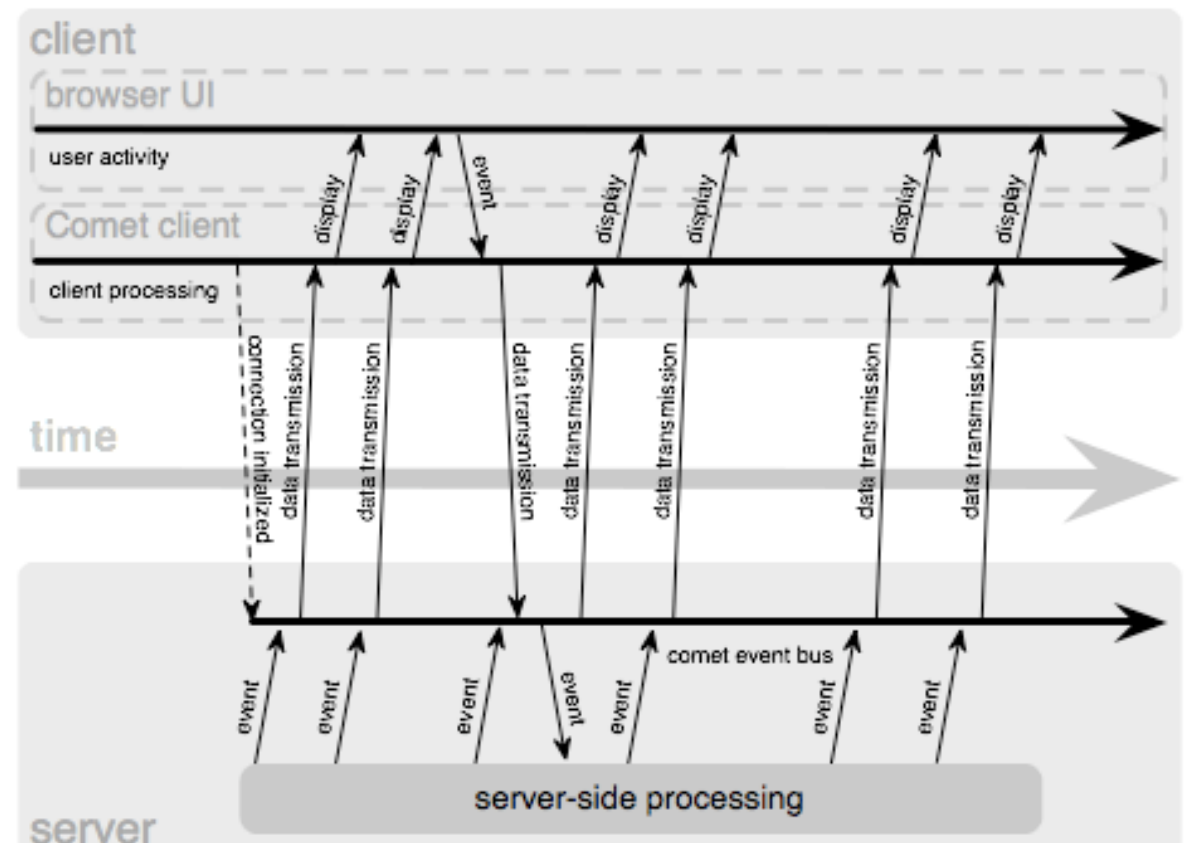
How Is This Different From Ajax?

Servers push data in addition to clients requesting it

Ajax web application model (asynchronous)

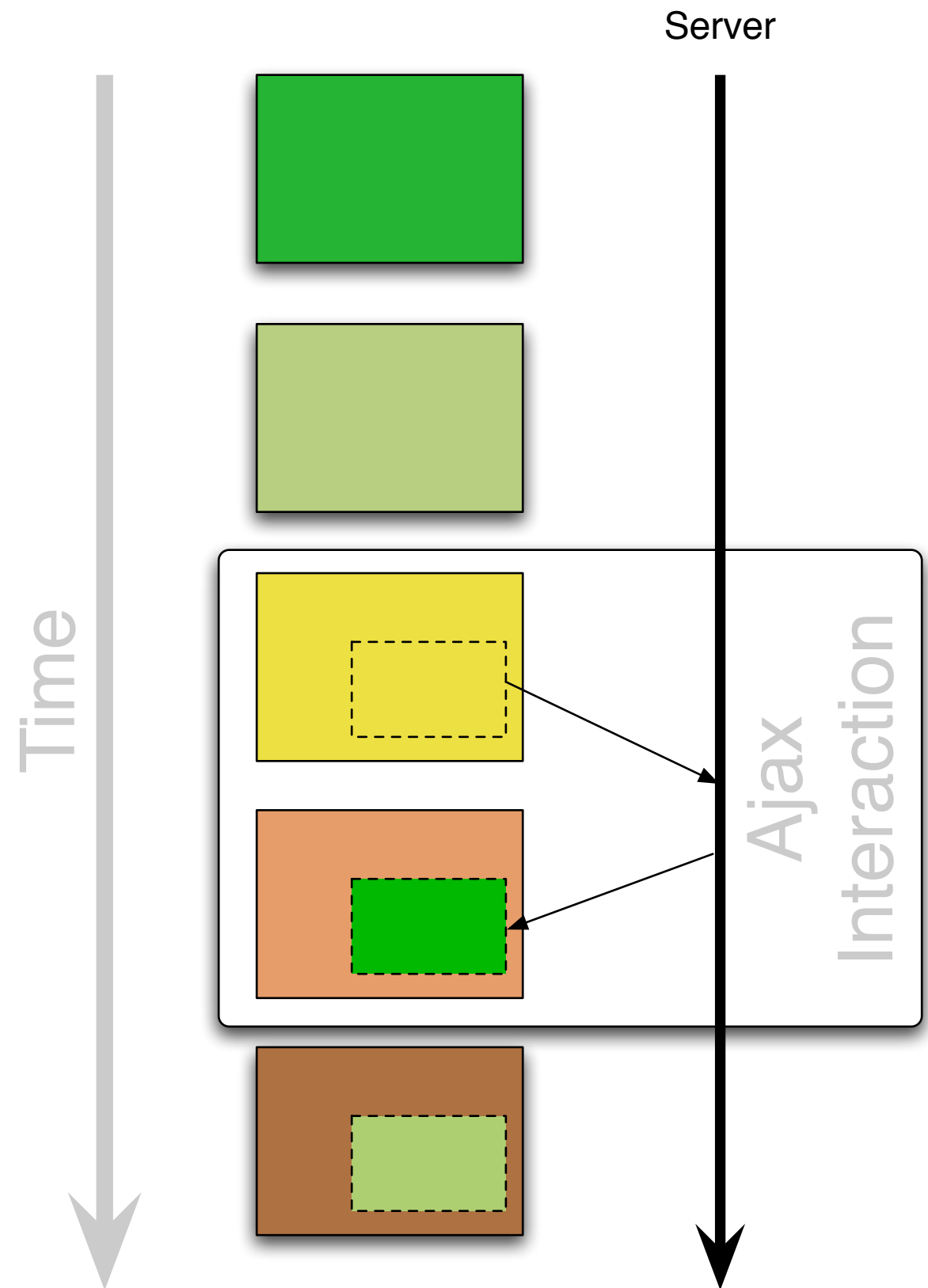


Comet web application model



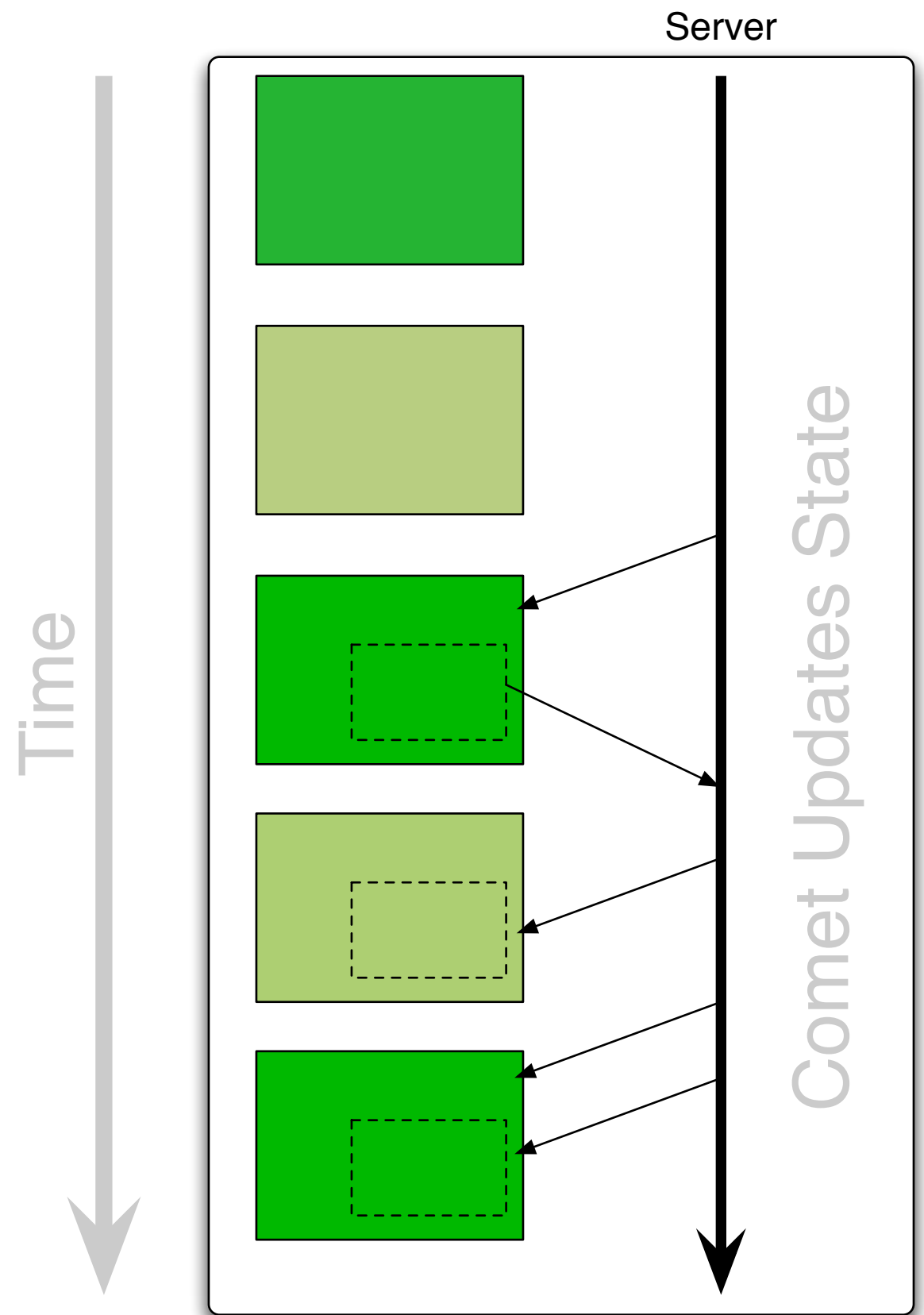
Stale Ajax

Context and manipulated content **go stale** at different rates



Keeping Up

Pushing state updates keeps
Ajax interactions and page
context in sync



How Comet Fights Lag

- ▶ Avoids HTTP & TCP/IP set-up and tear-down
- ▶ **Single connection is re-used** (in some forms)
- ▶ Ajax + Polling latency:
 - ▶ Time since last polling request +
 - ▶ TCP and HTTP request setup +
 - ▶ Data transmission time/latency
- ▶ Comet latency:
 - ▶ Data transmission time/latency
- ▶ Lots of “zombie” connections! (C10K+ problem)



Transfer Only Necessary Data,
Exactly When It's Most Relevant





Demo

Implementation Styles

▶ Long-poll

- ▶ Examples: Meebo, Live Page
- ▶ Reconnect after every datagram
- ▶ Server might package multiple datagrams together
- ▶ Simple to implement w/ XMLHTTP object

▶ Multipart XMLHTTP

- ▶ No known system does this portably today
- ▶ Similar to the forever-frame technique
- ▶ Different delimiters for IE and FF, no Safari

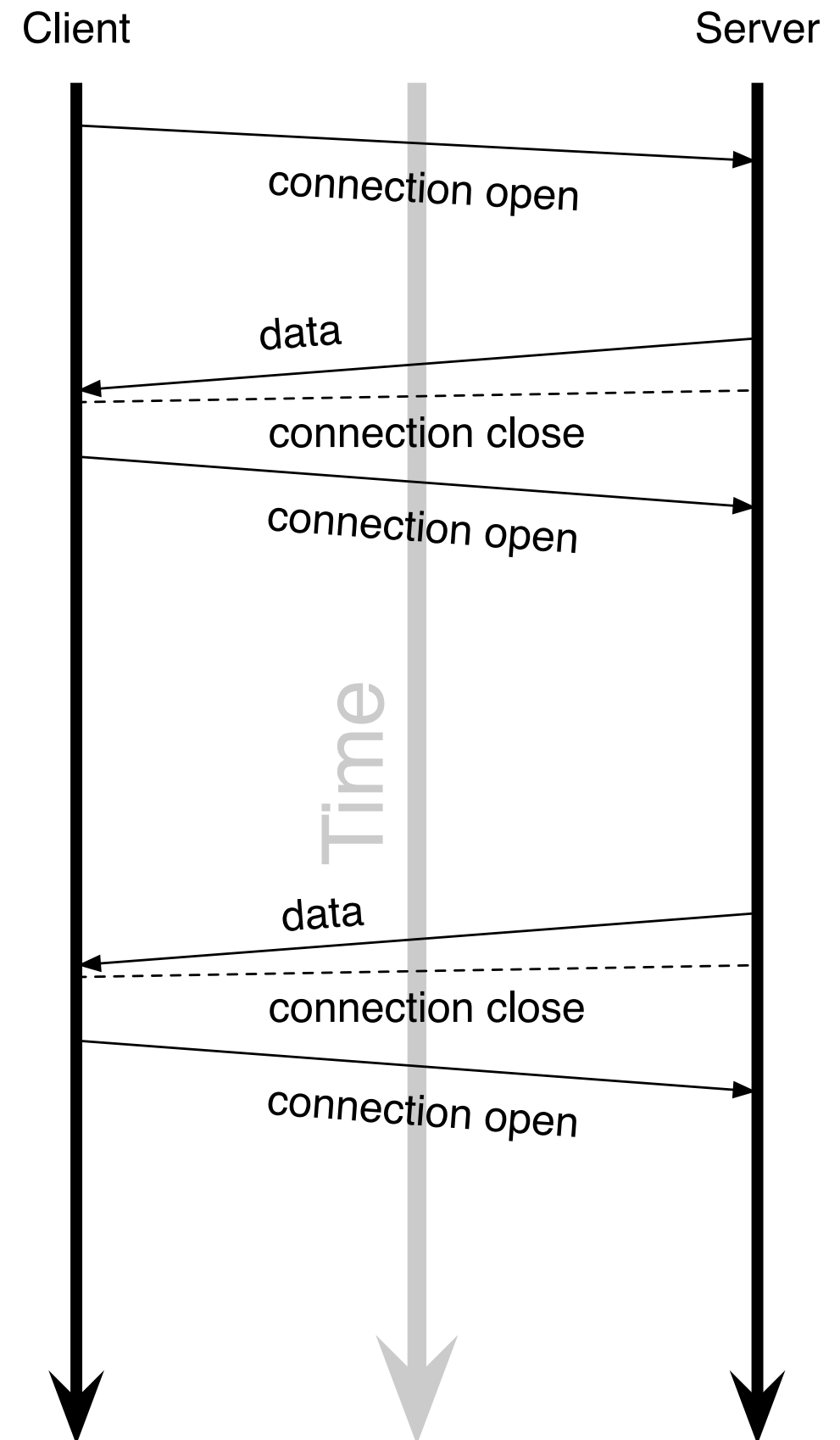
Contd.

- ▶ The “forever frame”
 - ▶ frame or iframe
 - ▶ Browser incremental rendering hack
 - ▶ `<script>` blocks sent to iframe are evaluated after some sort of “flush” token tickles the browser
 - ▶ Highly portable
 - ▶ Allows connections to sub-domains
 - ▶ `document.domain`
 - ▶ Important in designing workable architectures

The Long Poll

Data transfer ends the connection

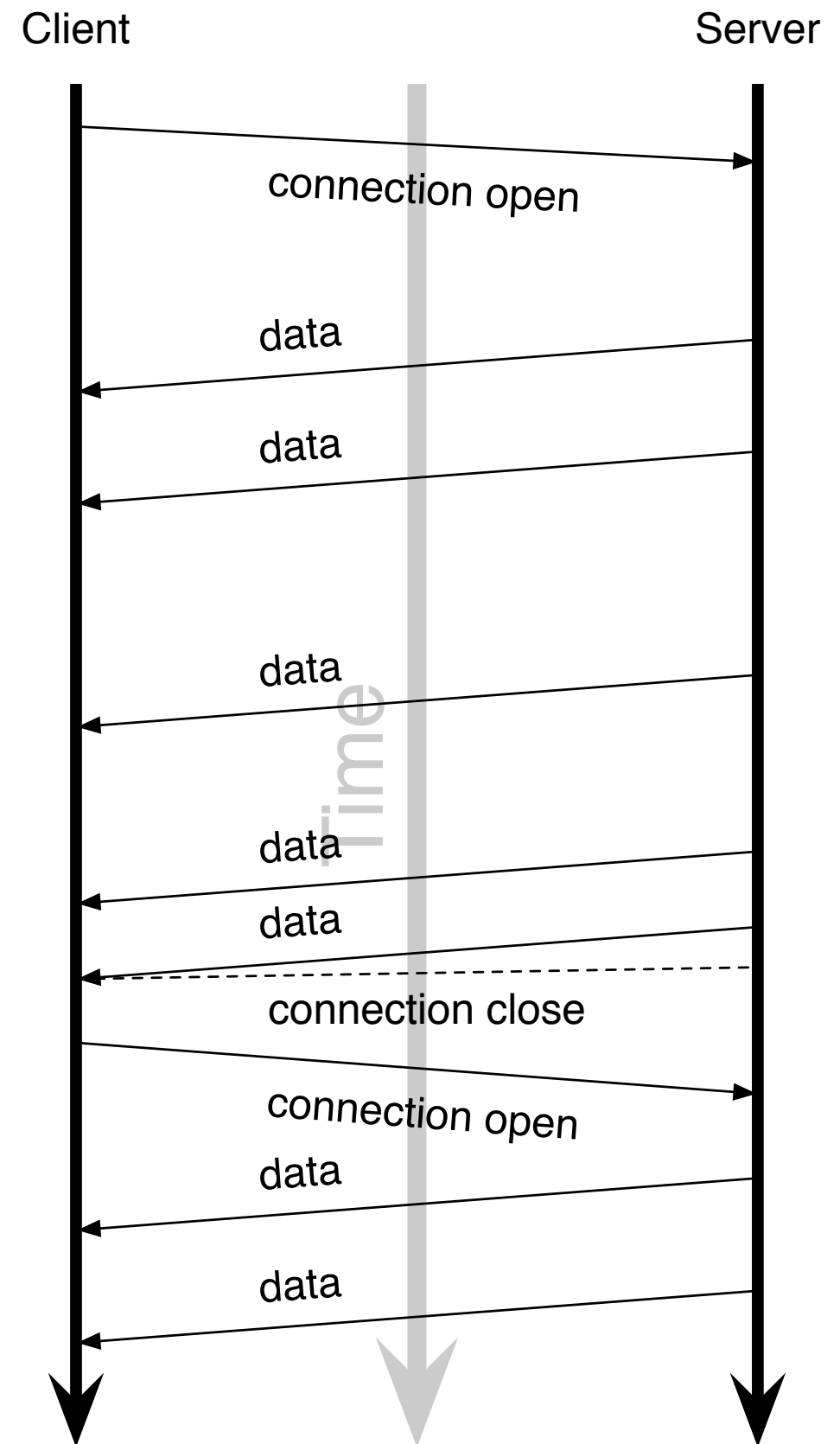
Clients re-connect after every datagram



Forever Frames and Multipart

Connection only closes on errors or connection “recycling”

Data is encoded in “envelopes”



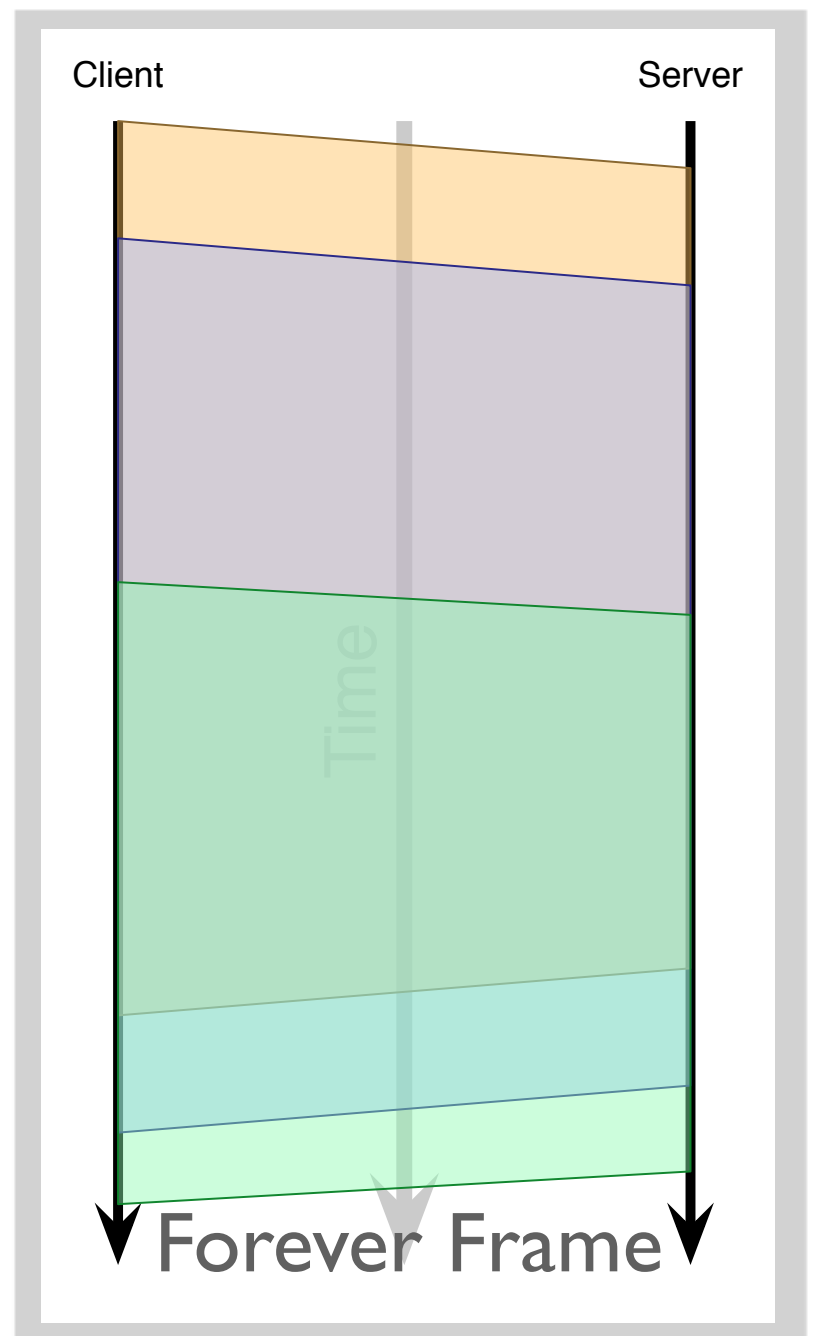
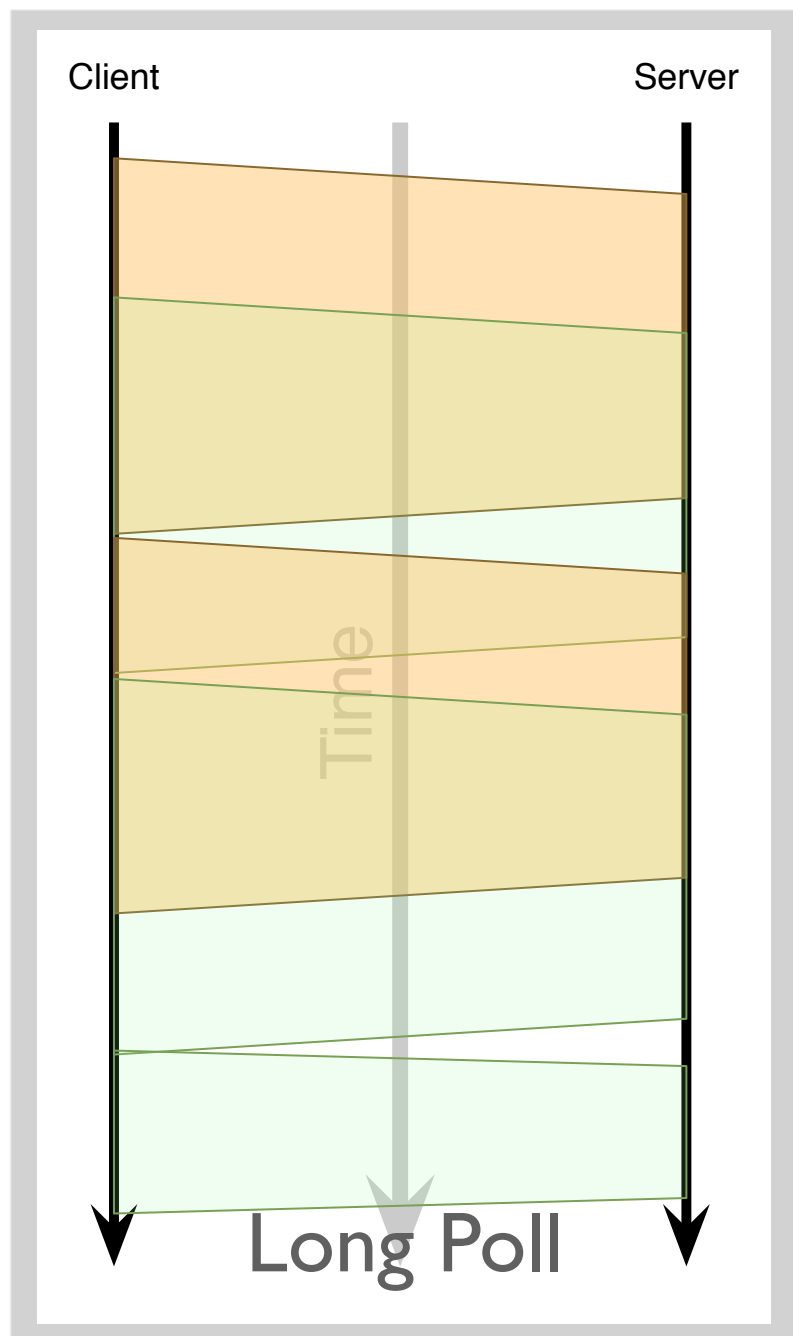
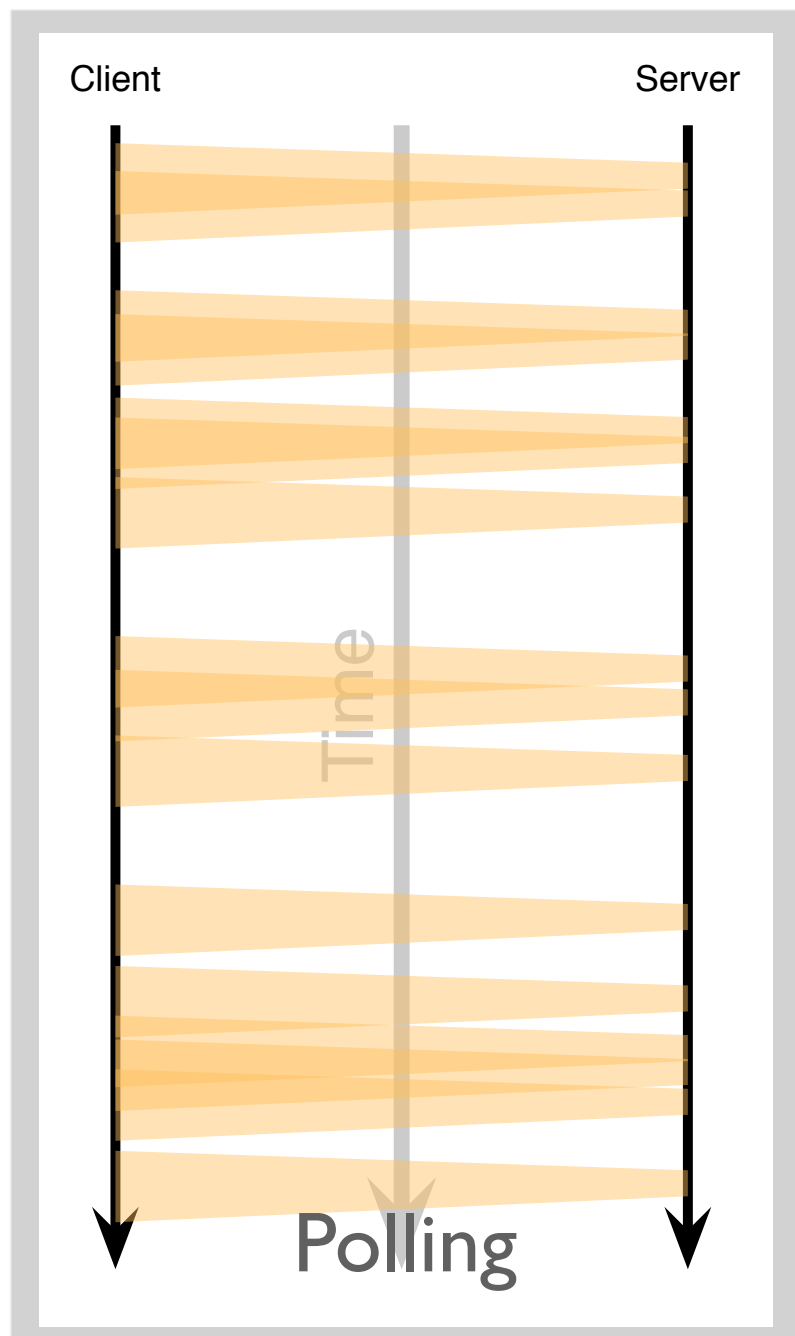


Copyright Simon James

Today's Web Servers Won't Cut It

Comet *Can* Reduce Load

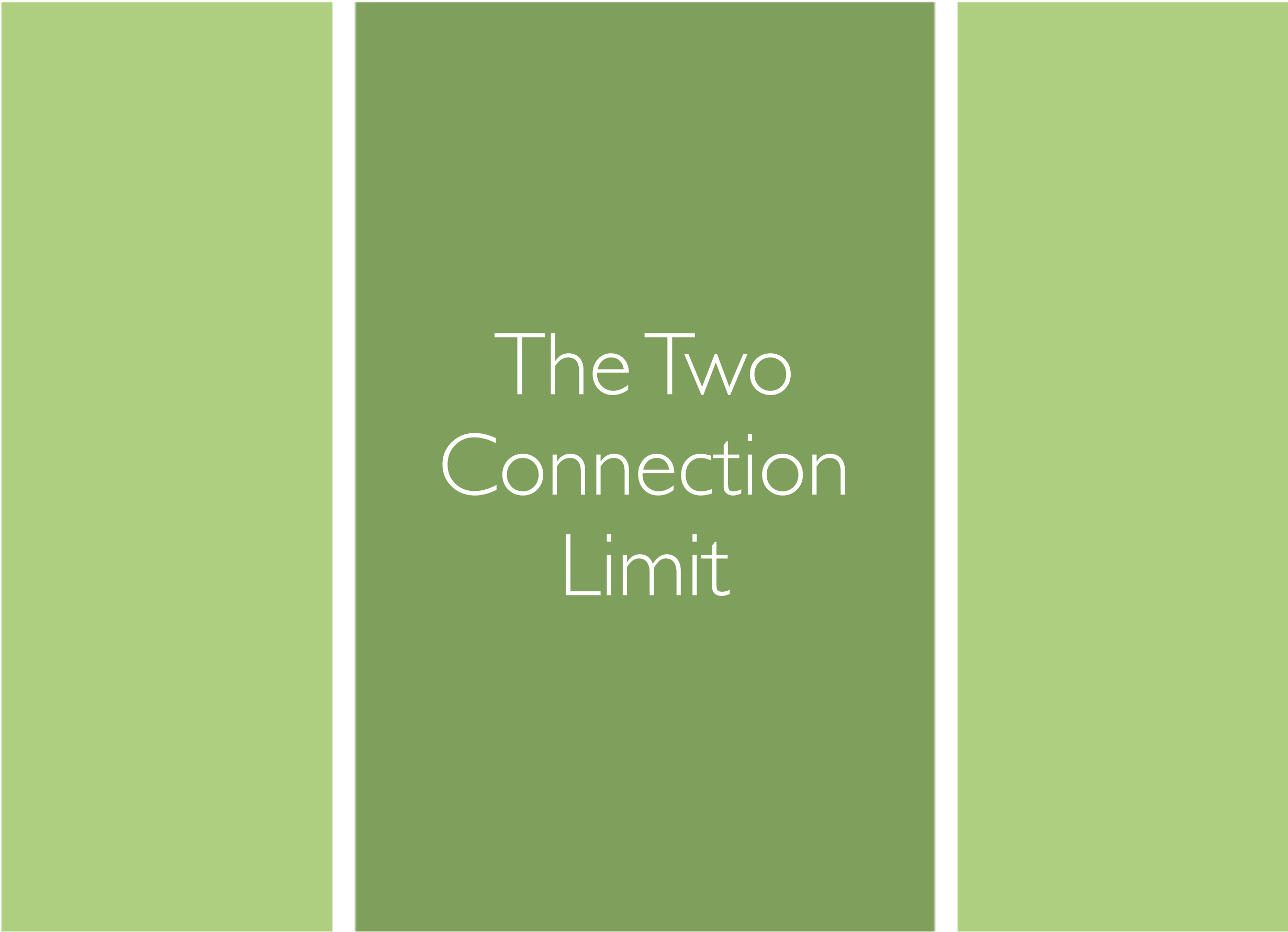
- ▶ But not on your current web infrastructure
- ▶ Polling is a latency/load trade-off
- ▶ Comet is an architectural complexity trade-off (today)
- ▶ Most of today's web servers use threads or processes
 - ▶ Threads consume fixed resources per request
 - ▶ Do not free them until end of connection
 - ▶ Comet does not free connections quickly
 - ▶ Polling frees resources quickly, but makes many times as many requests



Load Profiles

Event-Based Tools

- ▶ **OS level**
 - ▶ kqueue (FreeBSD)
 - ▶ epoll (Linux)
- ▶ **Network level**
 - ▶ **POE** (Perl)
 - ▶ **Twisted** (Python)
 - ▶ **Yaws** (Erlang)
 - ▶ **event_mpm** (Apache 2.2, unstable)
 - ▶ **Jetty** (Java)



The Two Connection Limit

Workarounds

- ▶ Multiplex!
 - ▶ Events for multiple components must come over the same connection
 - ▶ Prevent creation of multiple tunnels
- ▶ DNS hackery
 - ▶ document.domain + subdomains, wildcard DNS
- ▶ Flash
 - ▶ XMLSocket + Flash 8 “ExternalInterface”

Is Comet Good For My App?

- ▶ Do users collaborate on shared data?
- ▶ Can presence data improve the conversation?
- ▶ Can your users benefit from “fresher” data?
- ▶ Can it not be attempted any other way?
 - ▶ Can my architecture handle it?
- ▶ How long do users stay in a single page?
- ▶ If lag is acceptable, can polling work/scale instead?

Early Design Lessons

- ▶ Work with interaction designers
- ▶ Learn from your desktop competition
 - ▶ They had the same design problems
- ▶ Be consistent
- ▶ Let users know *why* the data is changing
- ▶ Let users know *who* changed the data
- ▶ Communicate connection failures clearly
- ▶ Push data updates, not functionality changes



Evolution, Not Revolution





Questions?

