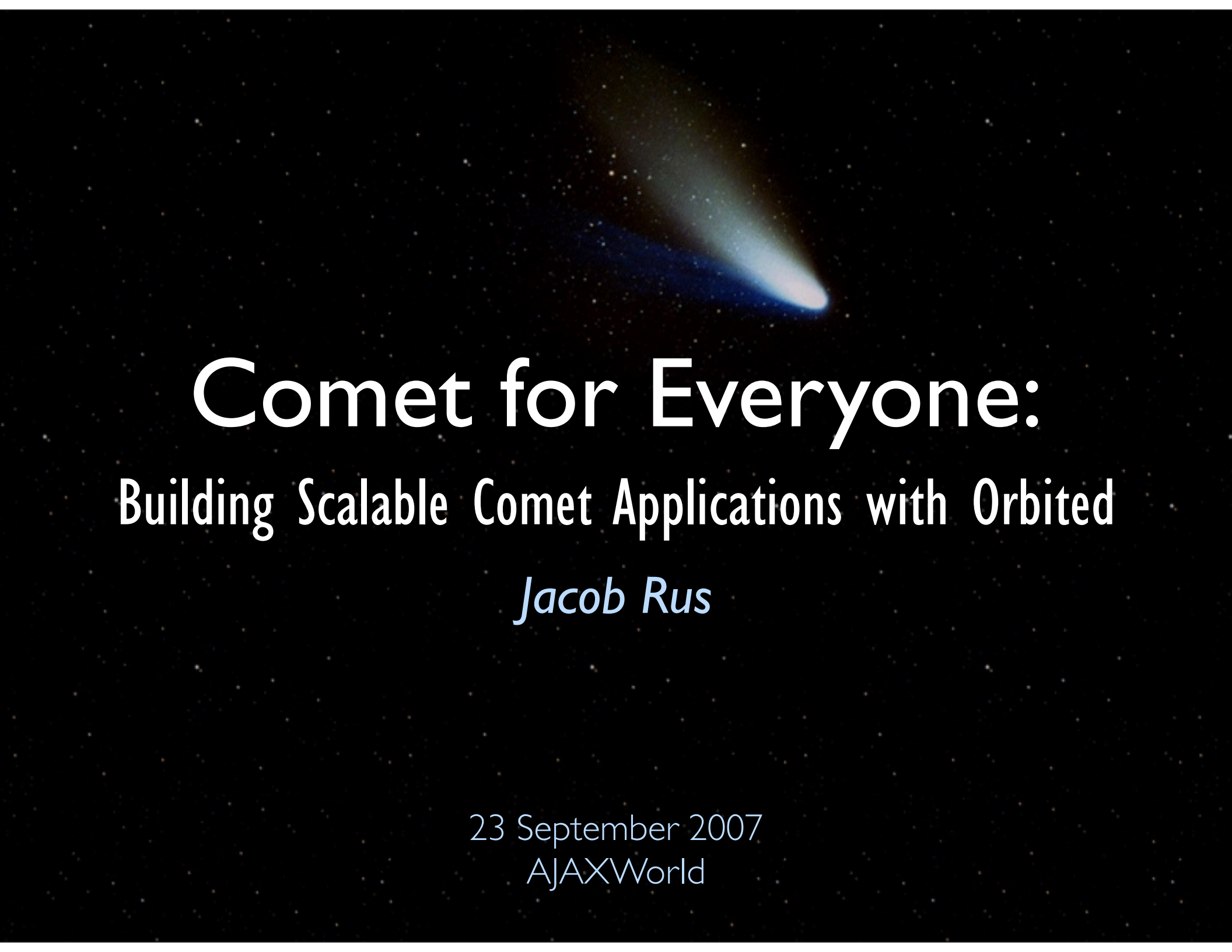


Comet for Everyone:

Building Scalable Comet Applications with Orbited

Jacob Rus

23 September 2007
AJAXWorld

What is Comet?

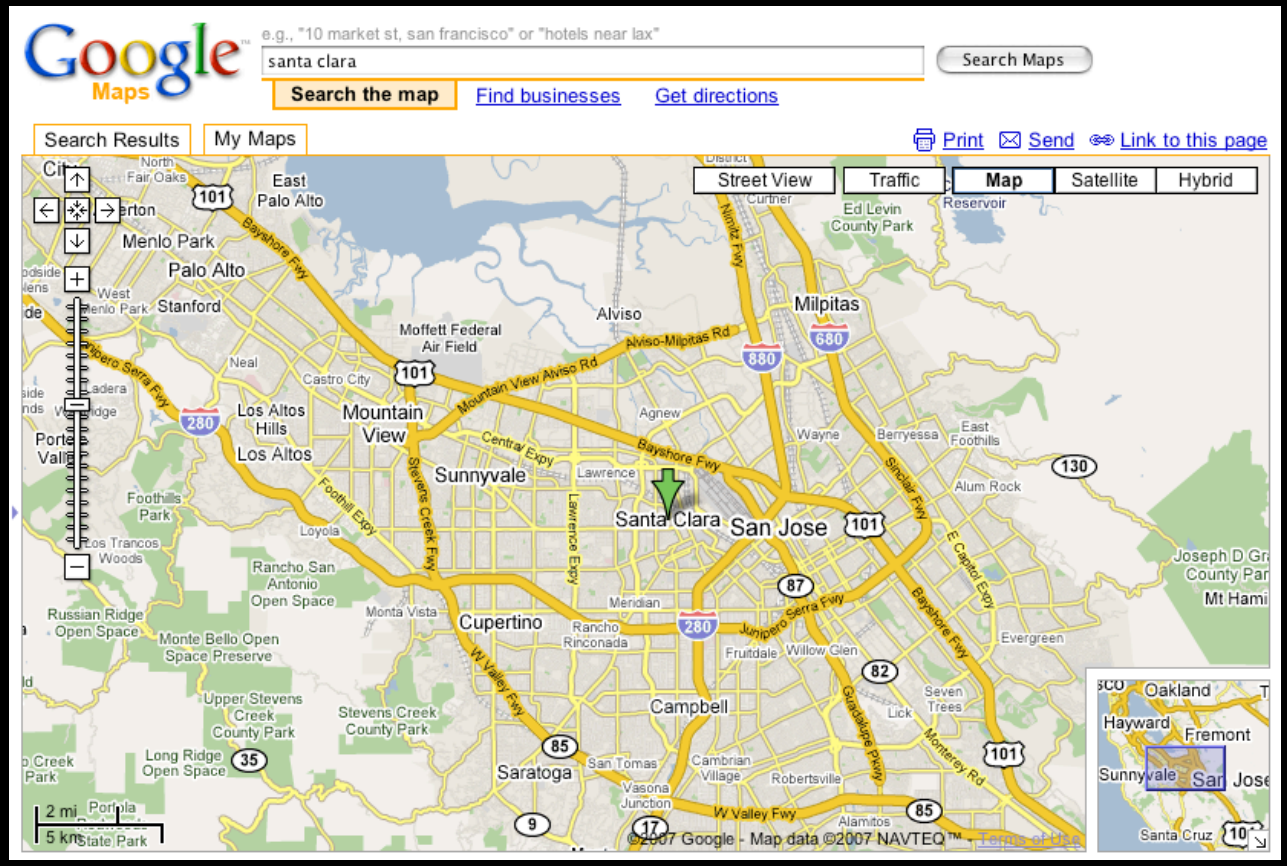
Pushing real-time updates to web browsers.

Original web: browser requests whole pages

What is Comet?

Pushing real-time updates to web browsers.

AJAX web: browser requests chunks of data



What is Comet?

Pushing real-time updates to web browsers.

Comet web: server pushes data to browser

Demo

Talking to myself with CherryChat

What is Orbited?

Taking care of comet so you don't have to

- Without Orbited: Each application re-implements its own comet server
  - Months of developer time
  - If done poorly, impossible to scale
- With Orbited: Every application uses Orbited
  - Easy to add comet; app server unchanged
  - Stable, scalable implementation

Dead-Simple Interface

You only need to know "event"

```
from pyorbited.simple import Client
orbit = Client()

user_key = "username, 0, /example"
message = "Hello, browser!"
orbit.event(connection_keys, message)
```

It's really that simple.

Sessions

Every session has a unique connection key

Example from our previous server-side code:

```
user_key = "username, 0, /example"
```

- Username: One for each user
- Session key: Generally one session key for each logged-in browser per user
- Location: The name of a particular chat room, for example

URLs

Clients request very simple URLs

http://domain/location|username,session,transport

for instance:

http://domain/example|username,0,iframe

- The first three colored bits are the connection key from the previous slide
- Orbited matches this URL with the events sent by the application
- Transport: describes the transmission method

Transports

Easy to add new transports to Orbited.

- No extra client-side javascript to handle iframe transport
- XMLHTTPRequest long polling requires 2 dozen lines of javascript

```
function long_polling() {
  xhr = createXMLHttpRequest()
  xhr.onreadystatechange = function() {
    orbited_event(xhr);
  }
  xhr.open("GET", "/location|" +
    username + ',0,xhr')
  xhr.send(null);
}
```

```
function orbited_event(xhr) {
  try {
    if (xhr.status  == 200) {
      if (xhr.readyState == 4) {
        old_xhr = xhr;
        data = eval(xhr.responseText)
        if (data != undefined) {
          event(data)
        }
        long_polling();
      }
    }
    else { alert("failed"); }
  }
  catch(e) {}
}
```

Demo

Sending events to the browser
from Ruby and Python

Orbited Proxy

Avoids cross-domain scripting conflicts

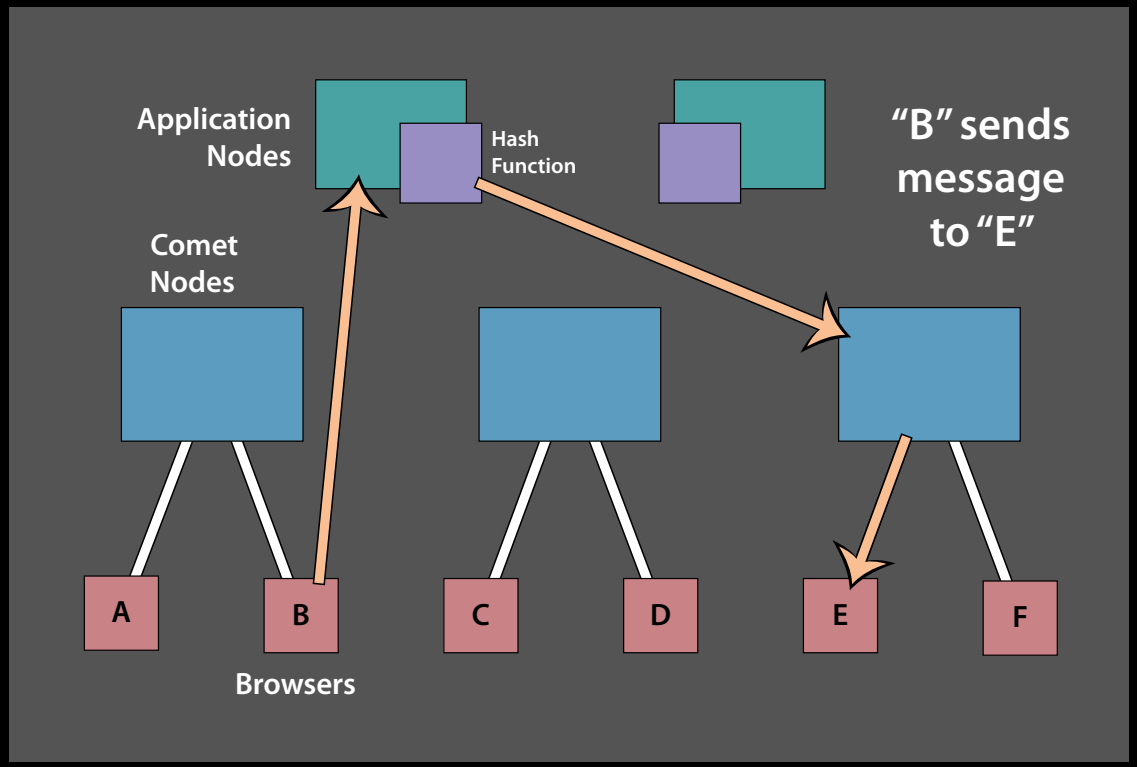
Cross-domain scripting conflicts make writing JavaScript for a site more complicated. The Orbited proxy gets rid of them

- Grabs Orbited events; passes the rest through
- Great for development: helps you figure out your app functionality fast
- Can run multiple apps behind it
- Not for large-scale production use.

Scaling Orbited Apps

Orbited makes scaling Comet achievable

- To learn about scaling in depth, see Michael's talk
- Scaling Comet is not an easy problem
- Orbited's simplicity helps it scale



Go Build Some Real-Time Web Apps!

Check out http://orbited.org for code, tutorials, documentation,