RCB: A Simple and Practical Framework for Real-time Collaborative Browsing

Chuan Yue, Zi Chu, and Haining Wang
The College of William and Mary

W&M Computer Science
WILLIAM AND MARY

End-user Real-time Communication

Document Sharing and Collaboration

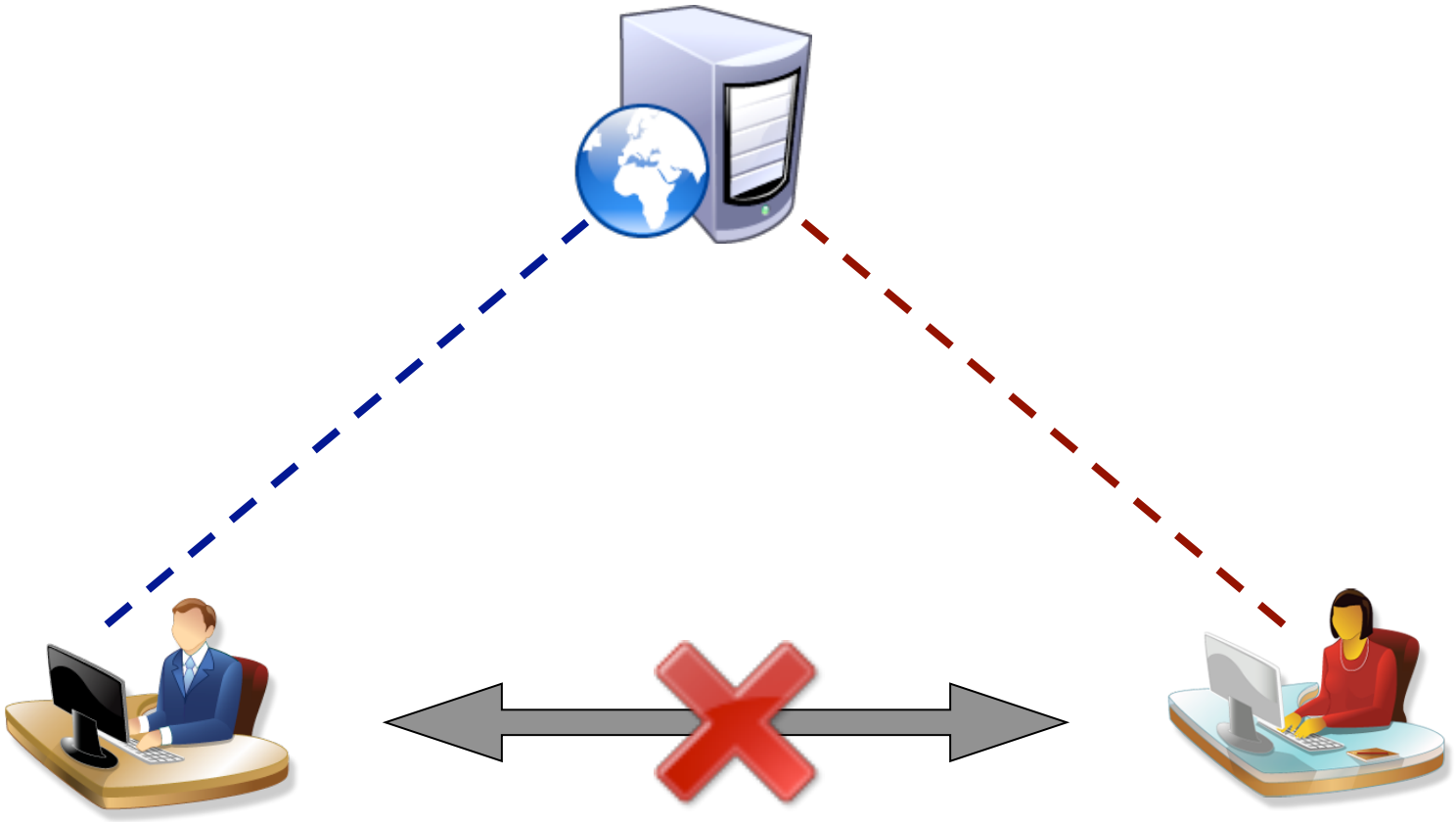
Google Docs BETA

Microsoft Office Live Beta

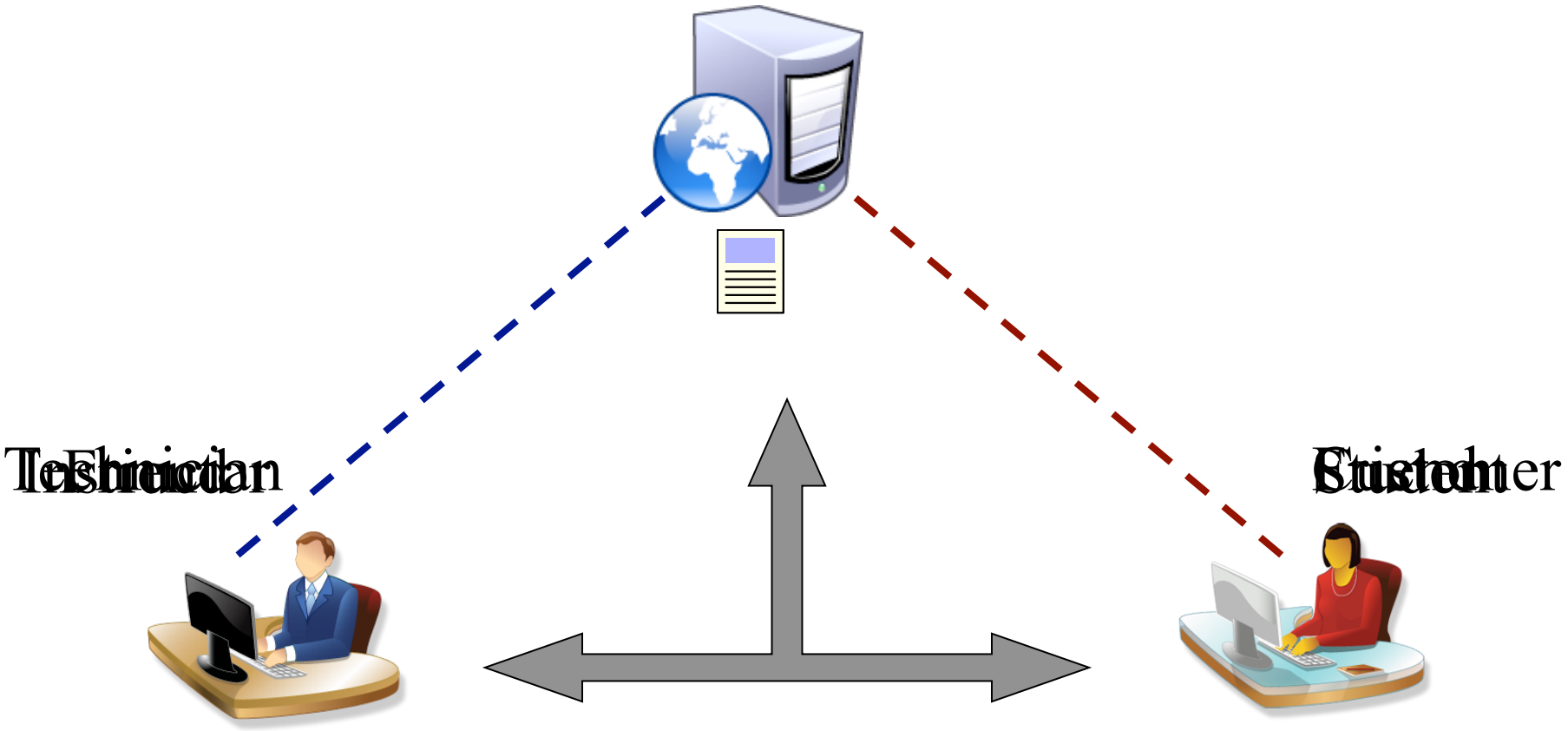
Adobe Buzzword

Zimbra

Web Browsing: Heavily Isolated



Collaborative Browsing (Co-browsing)



Simple Co-browsing via URL sharing

- E.g., instant messenger tools/browser extensions
- Limited collaboration
  - Can at most view webpages
- Narrow scope of webpages
  - Cannot access session-protected or dynamic webpages

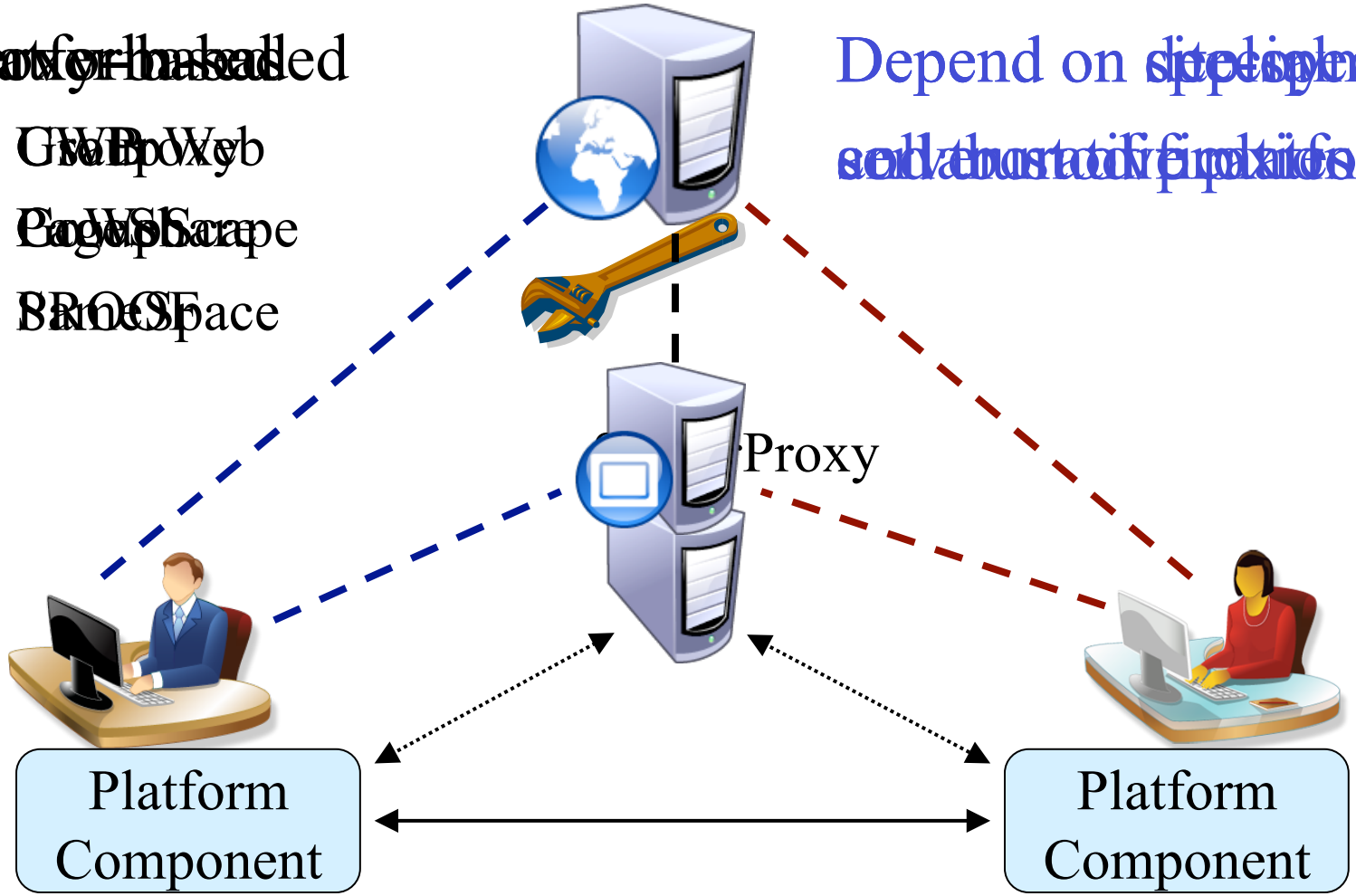
Complex Co-browsing via Screen Sharing

- E.g., screen or application sharing software
- High demands on network bandwidth and security
  - Grant the control of a whole screen or application

Specific Co-browsing Solutions

- Platform-based
  - GWB
  - CoWSE
  - BrowseSpace

Depend on specific platforms



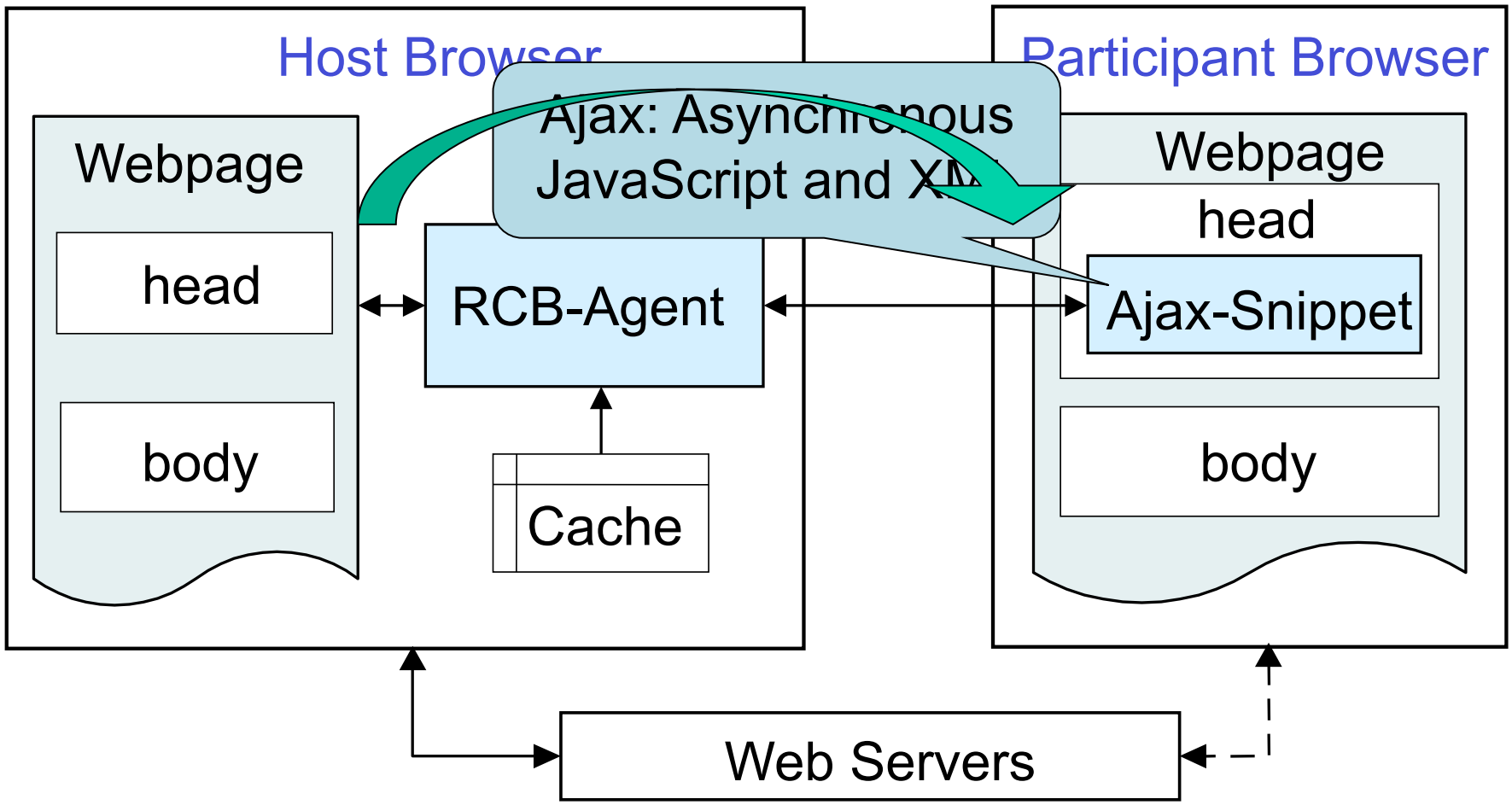
Our RCB Solution

- Pure browser-based solution
- Simple and Practical
- Almost everywhere, various webpages
- Fine-grained, high-quality

Outline

- Introduction
- Framework Design
- Implementation
- Evaluation

Architecture of the RCB Framework

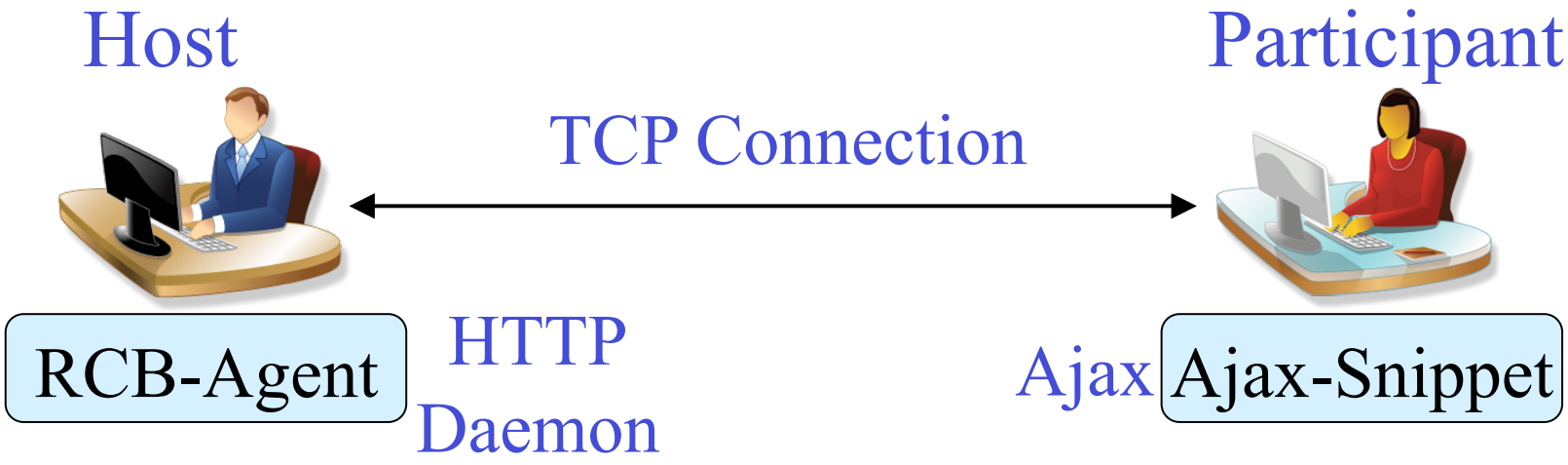


A Typical RCB Co-browsing Session

1. Start the agent
2. Establish connection
3. Visit a webpage
4. Clone and modify
5. Synchronize document
6. Replace HTML elements
7. Download object (non-cache)
8. Download object (cache)
9. Synchronize changes/actions

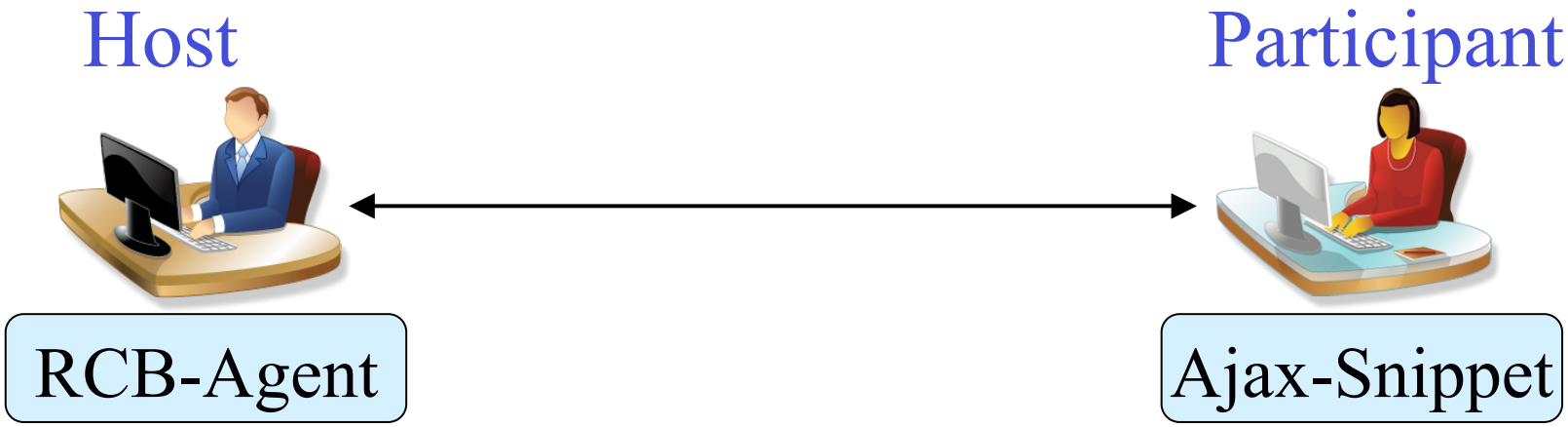
Repeat steps 3 ~ 9 !

Three Design Decisions



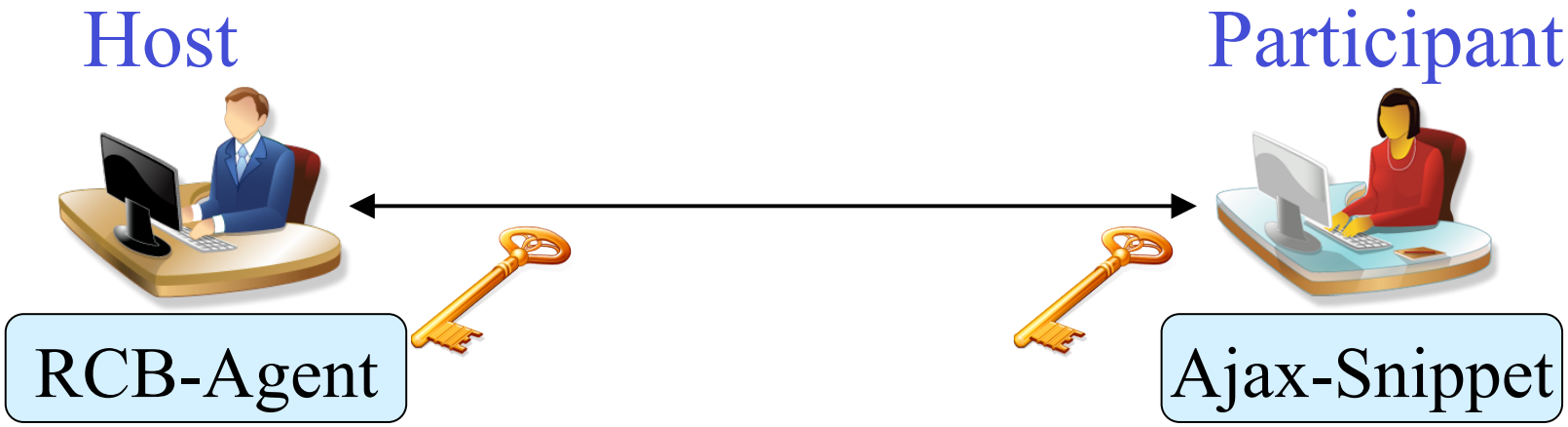
- Direct communication model
- HTTP-based service model
- Poll-based synchronization model

Co-browsing Topologies and Policies



- Multiple participants, free join/leave, awareness
- RCB-Agent enforces high-level policies

Security Design



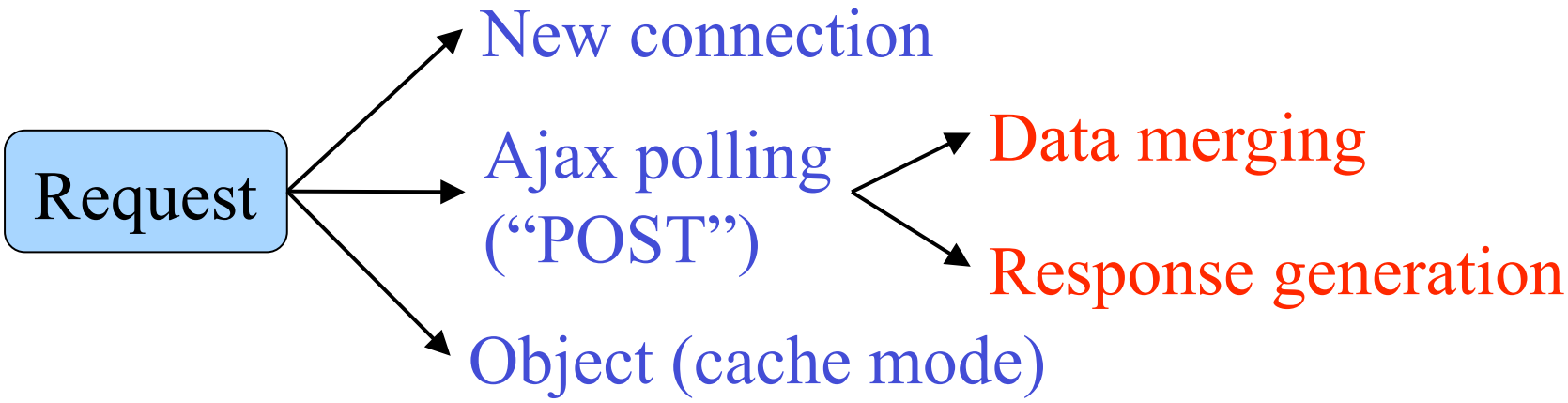
- Similar to visiting a trusted HTTP website
- Protect RCB-Agent by authenticating requests
  - HMAC (keyed-Hash Message Authentication Code)

Implementation Overview

- RCB-Agent
  - Firefox Extension 
  - Pure JavaScript
  - Possible for other browsers
- Ajax-Snippet
  - JavaScript objects/functions
  - Support different browsers  

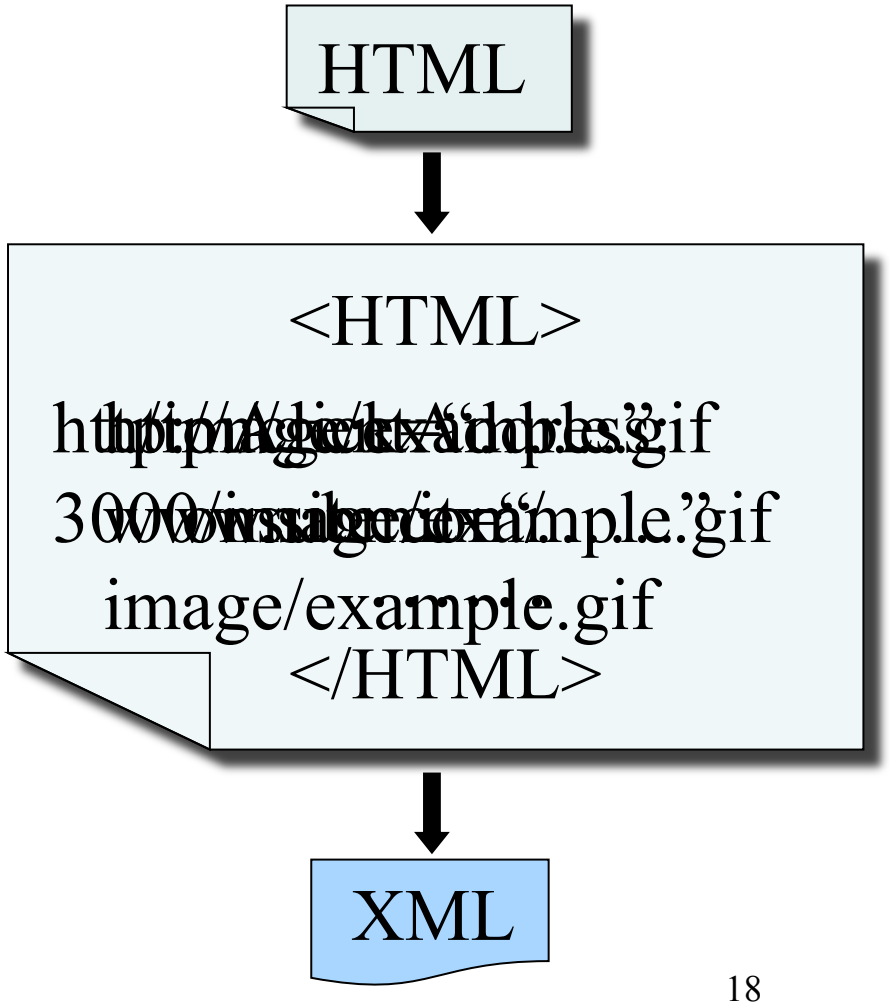
RCB-Agent Request Processing

- Implement a server socket object
  - Asynchronously accept new TCP connections
  - Asynchronously process HTTP requests
- Three types of HTTP requests



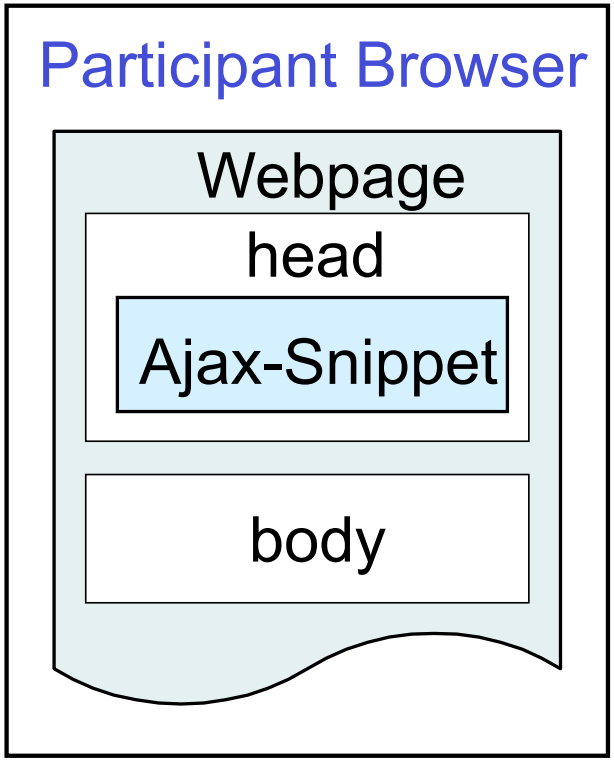
RCB-Agent Response Content Generation

1. Clone
2. Change object URL
   (Relative → Absolute)
3. Change object URL
   (Absolute → Agent)
4. Rewrite event handler
5. Generate response



Ajax-Snippet

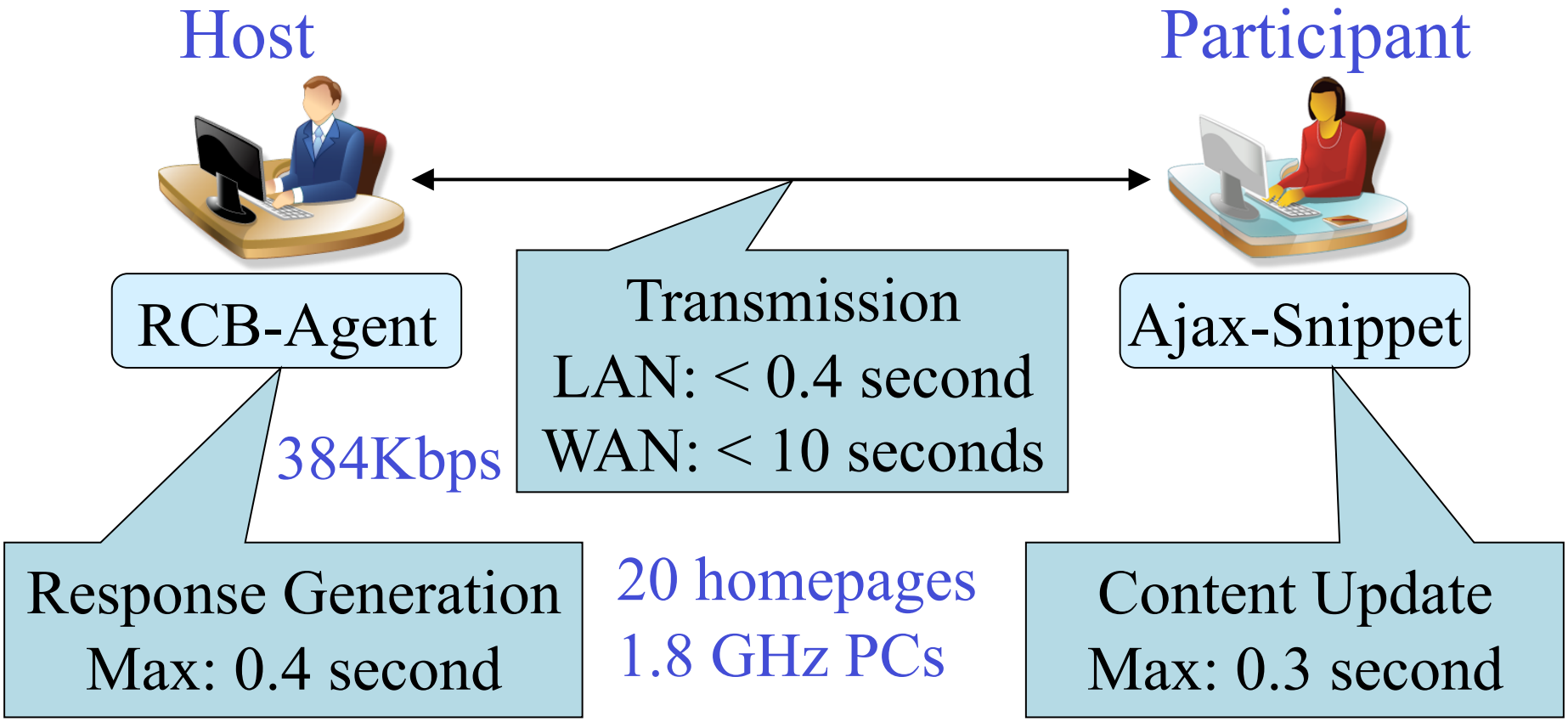
- Request Sending
  - XMLHttpRequest
  - “POST”, asynchronous
- Response Processing
  - Clean up and set head
  - Clean up and set other



Evaluation of RCB

- Performance Evaluation
  - The real-time performance of RCB
  - LAN environment and WAN environment
- Usability Evaluation
  - Whether RCB is helpful and easy to use
  - Using Google Maps and shopping online

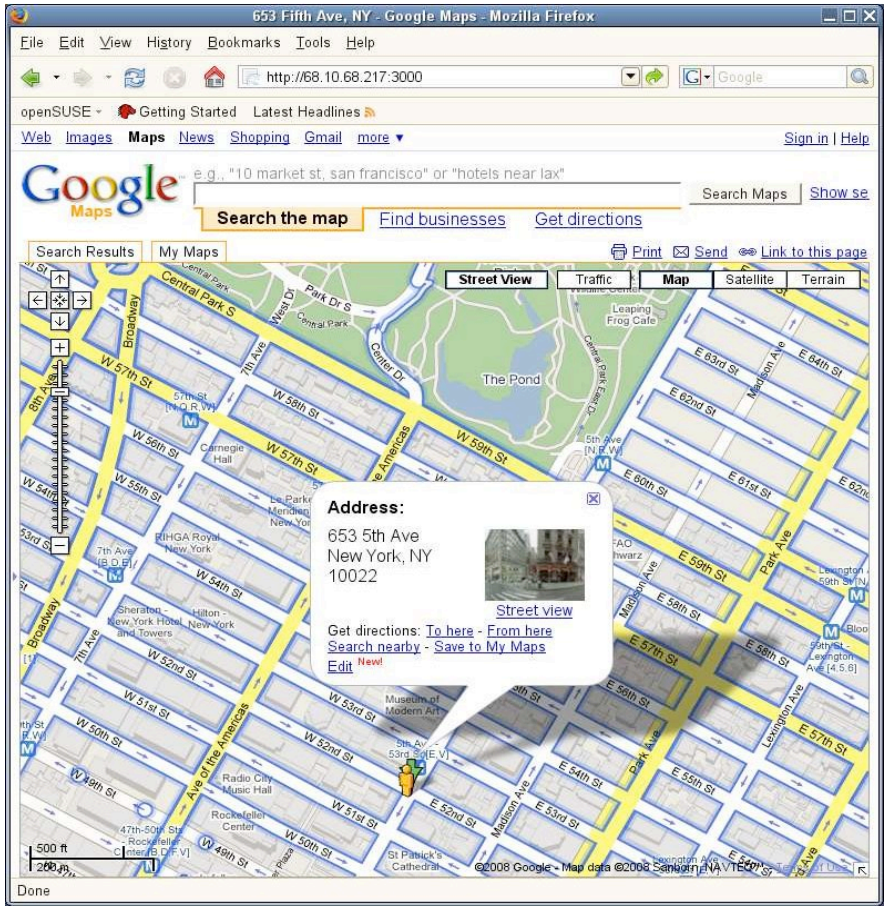
Performance Evaluation



Coordinating a Meeting Spot via Google Maps

- Bob hosts
- Alice joins
- Bob may
  - Search, zoom in/out, drag, switch views
- Alice sees same pages

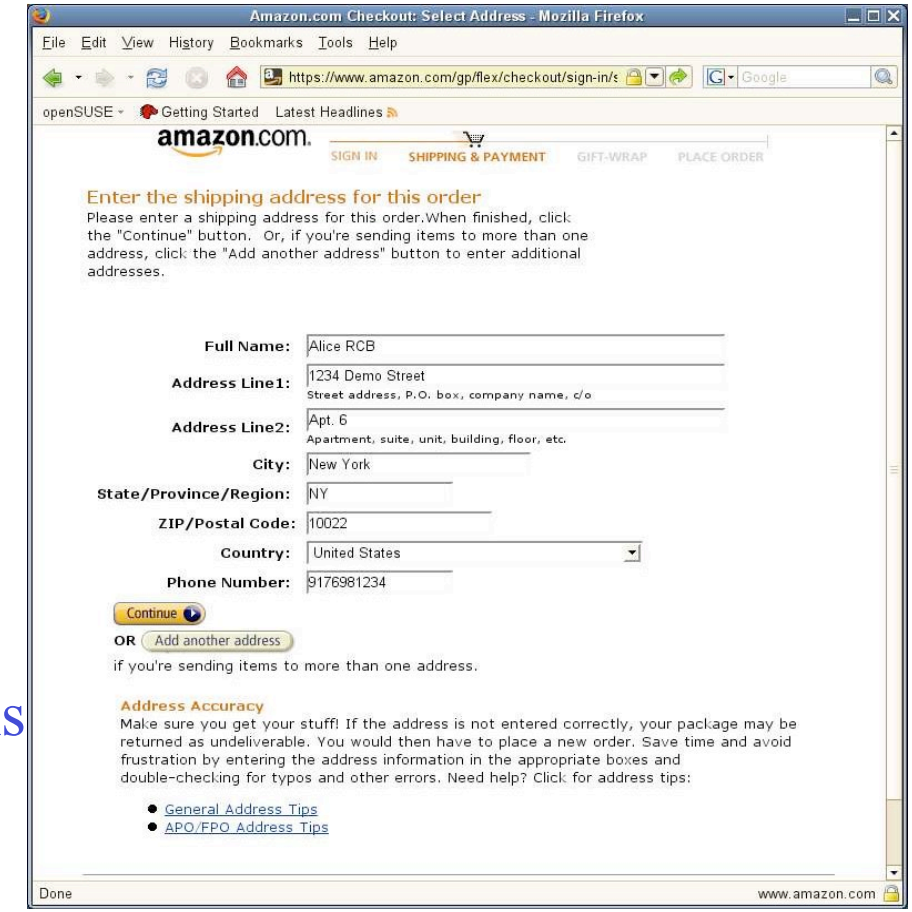
Rich content, communication intensive webpages



Online Co-shopping at Amazon.com

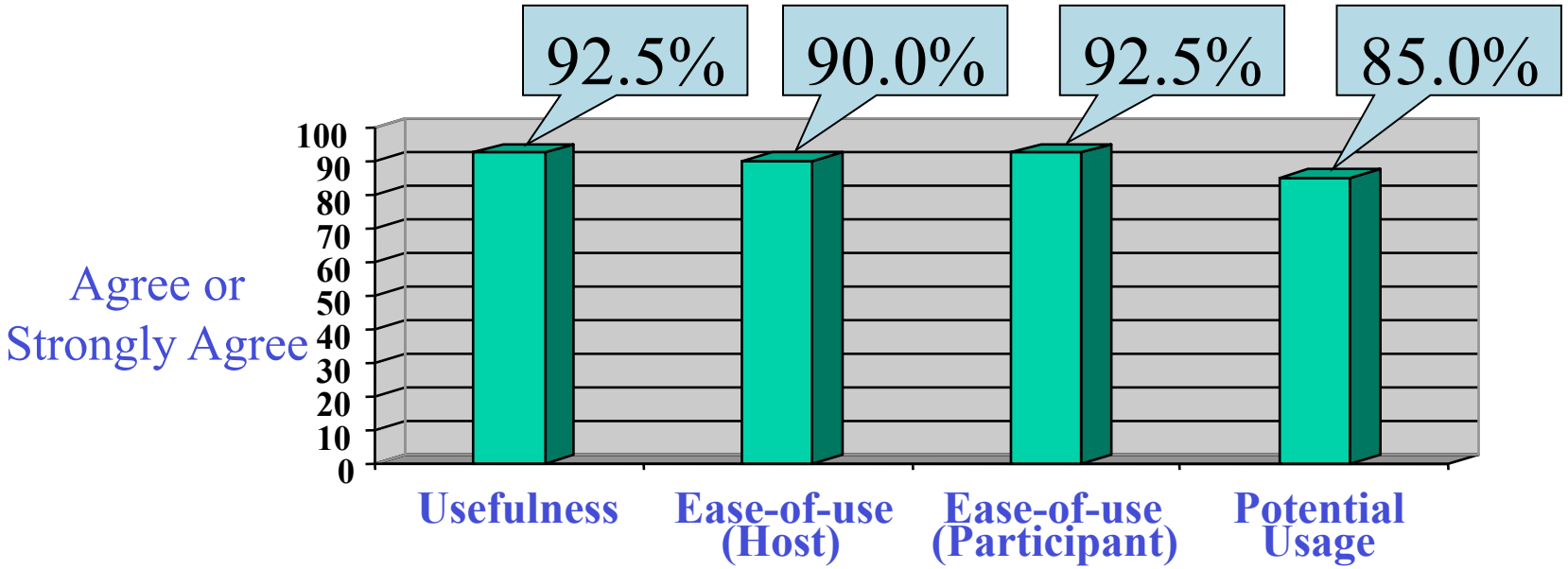
- Bob hosts
- Alice joins
- Bob or Alice may
  - Type in, search, click, fill/submit form

Dynamic/session-protected webpages, various interactions



Usability Evaluation

- Twenty students come from nine degree programs
- Ten pairs perform the two scenarios in a session
- Observation and questionnaire results



Summary

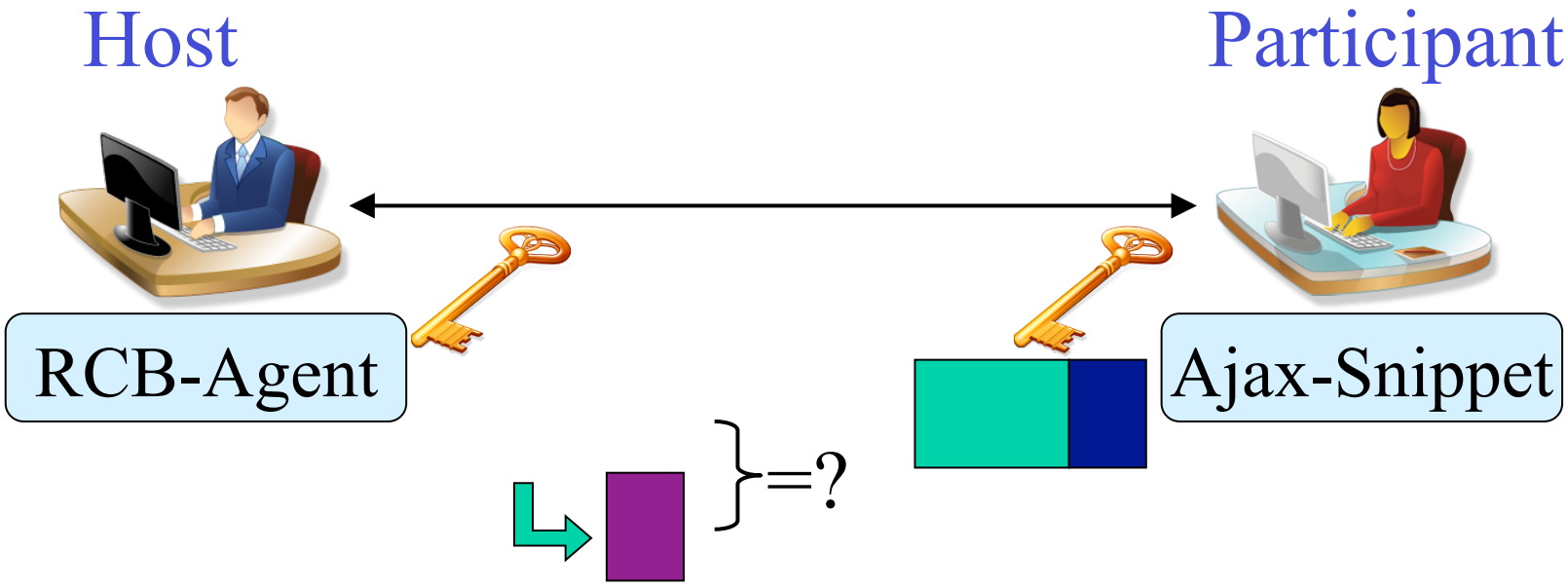
- Pure browser-based co-browsing solution
- Simple and Practical
- Implemented as a Firefox extension
- Efficient, high-quality, helpful and easy to use

Thank You!

cyue@cs.wm.edu
http://www.cs.wm.edu/~cyue

Backup Slides

Security Design



- Similar to visiting a trusted HTTP website
- Protect RCB-Agent by authenticating requests
  - HMAC (keyed-Hash Message Authentication Code)

XML Format Response Content

```
<?xml version=’1.0’ encoding=’utf-8’?>
<newContent>
  <docTime>documentTimestamp</docTime>
  <docContent>
    <docHead>
      <hChild1><![CDATA[escape(hData1)]]></hChild1>
      <hChild2><![CDATA[escape(hData2)]]></hChild2>
    </docHead>
    <docBody><![CDATA[escape(bData)]]></docBody>
  </docContent>
  <userActions>userActionData</userActions>
</newContent>
```