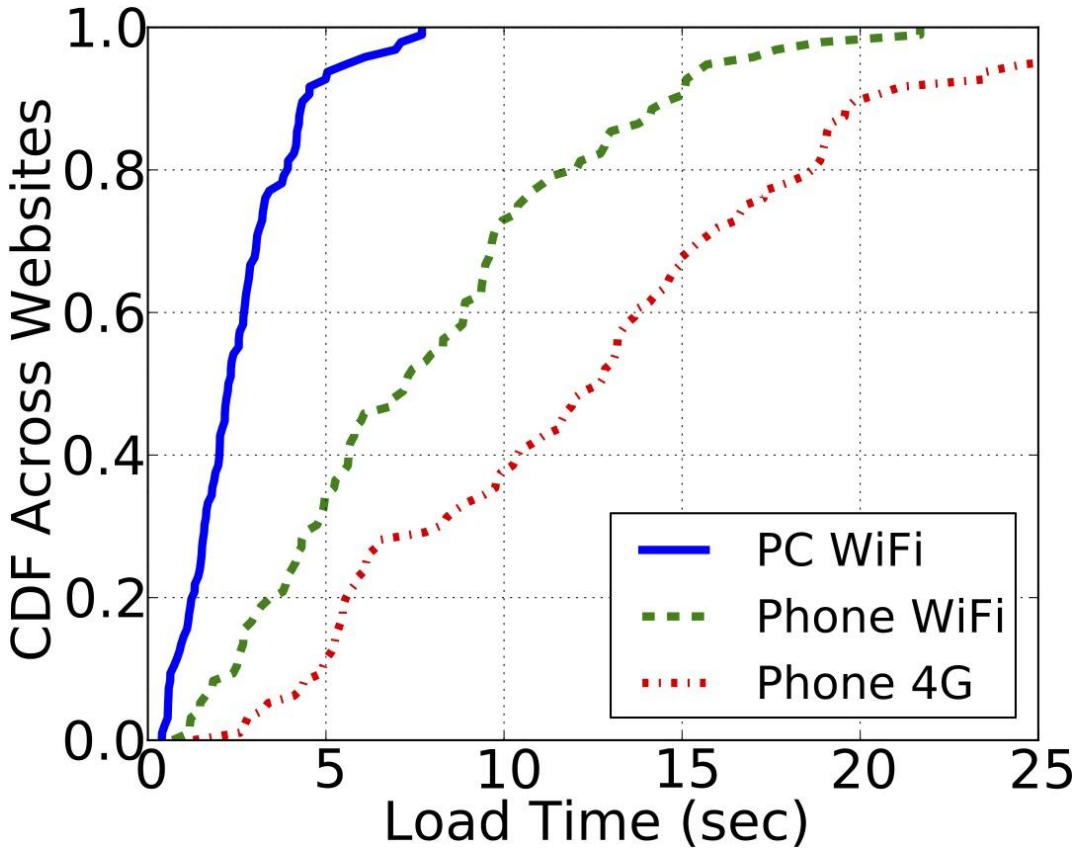
Klotski: Reprioritizing Web Content to Improve User Experience on Mobile Devices

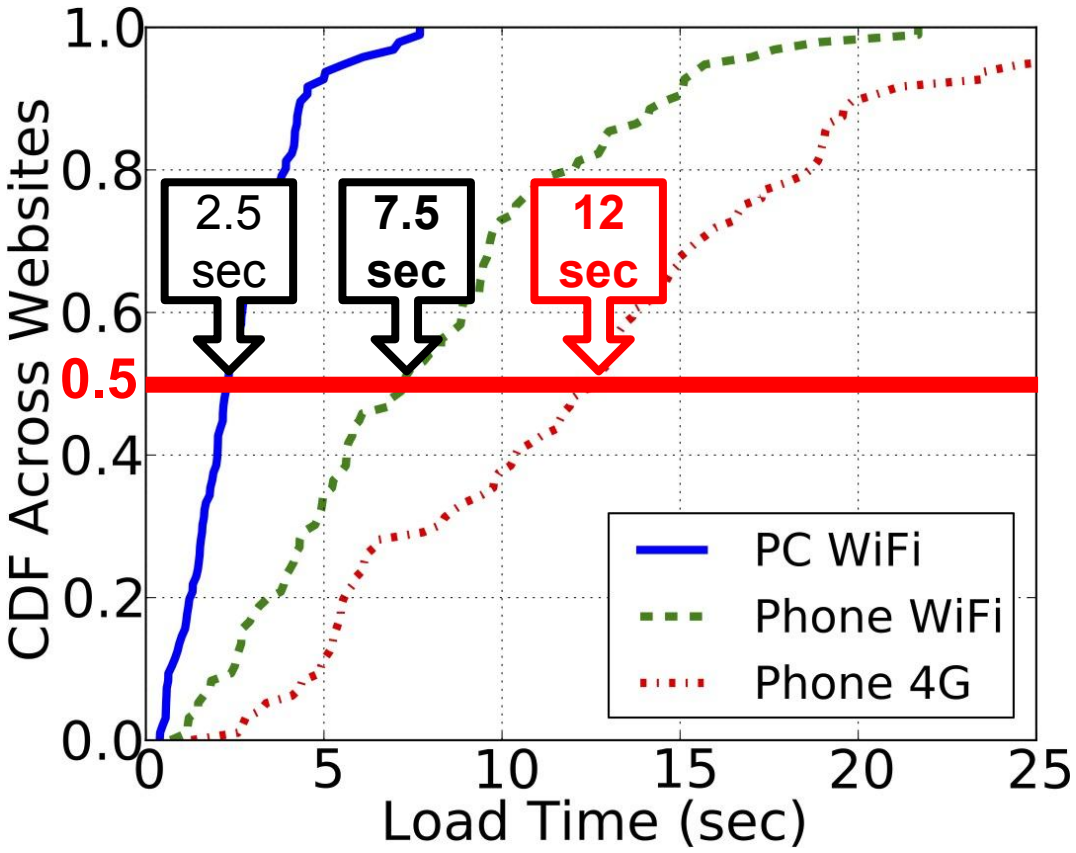
Michael Butkiewicz♕, Daimeng Wang♕,
Zhe Wu♖, Harsha V. Madhyastha♖, Vyas Sekar♘

♕ UC Riverside ♖ University of Michigan ♘ CMU

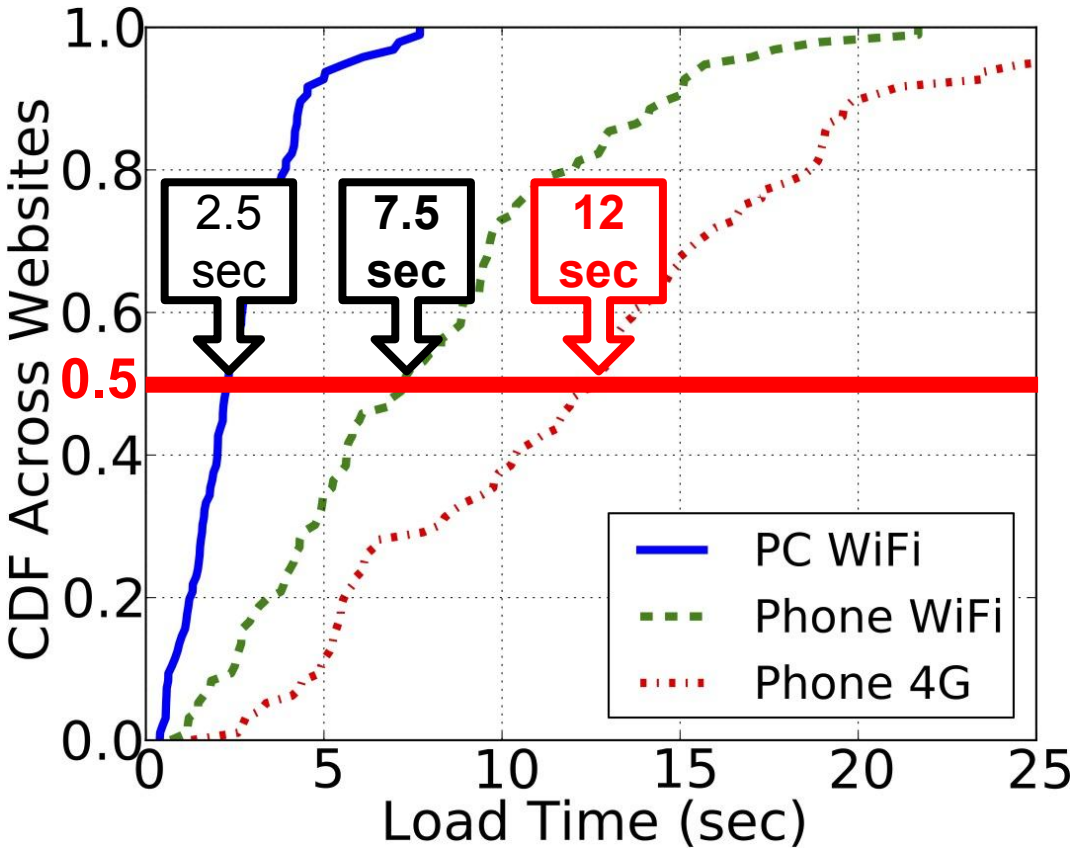
Motivation: Slow Mobile Web



Motivation: Slow Mobile Web



Motivation: Slow Mobile Web



Slow page loads → Less users, Lost business

Wide Range of Existing Solutions

Wide Range of Existing Solutions

Compression

Caching

Wide Range of Existing Solutions

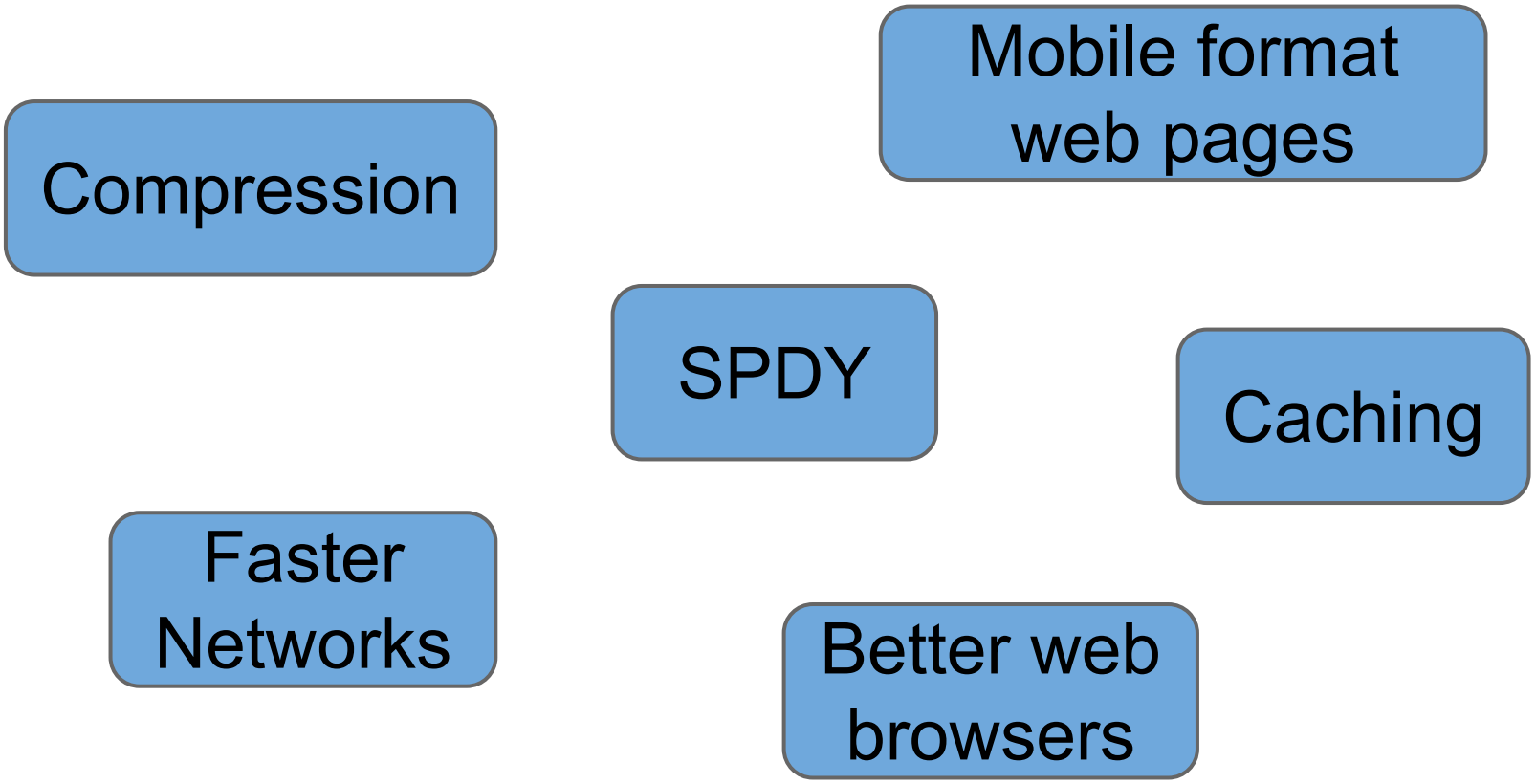
Compression

Mobile format web pages

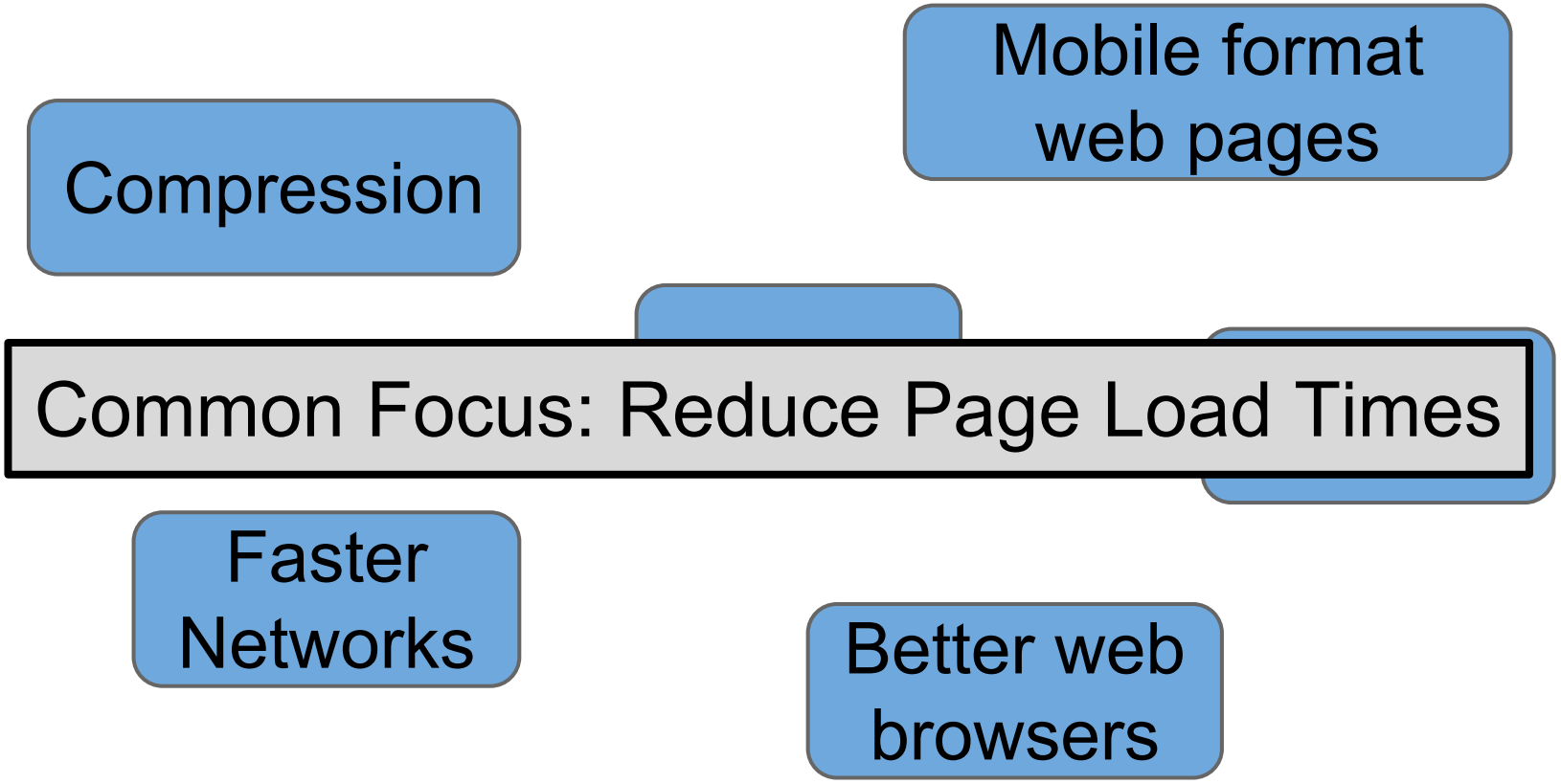
SPDY

Caching

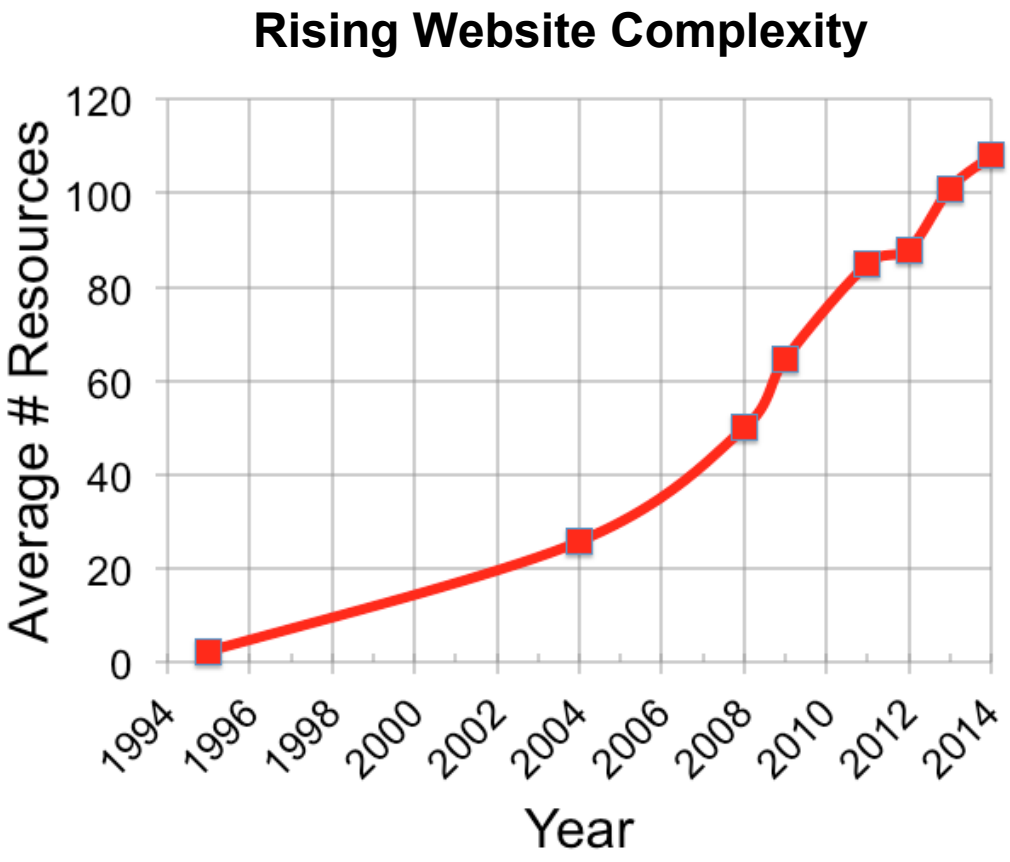
Wide Range of Existing Solutions



Wide Range of Existing Solutions

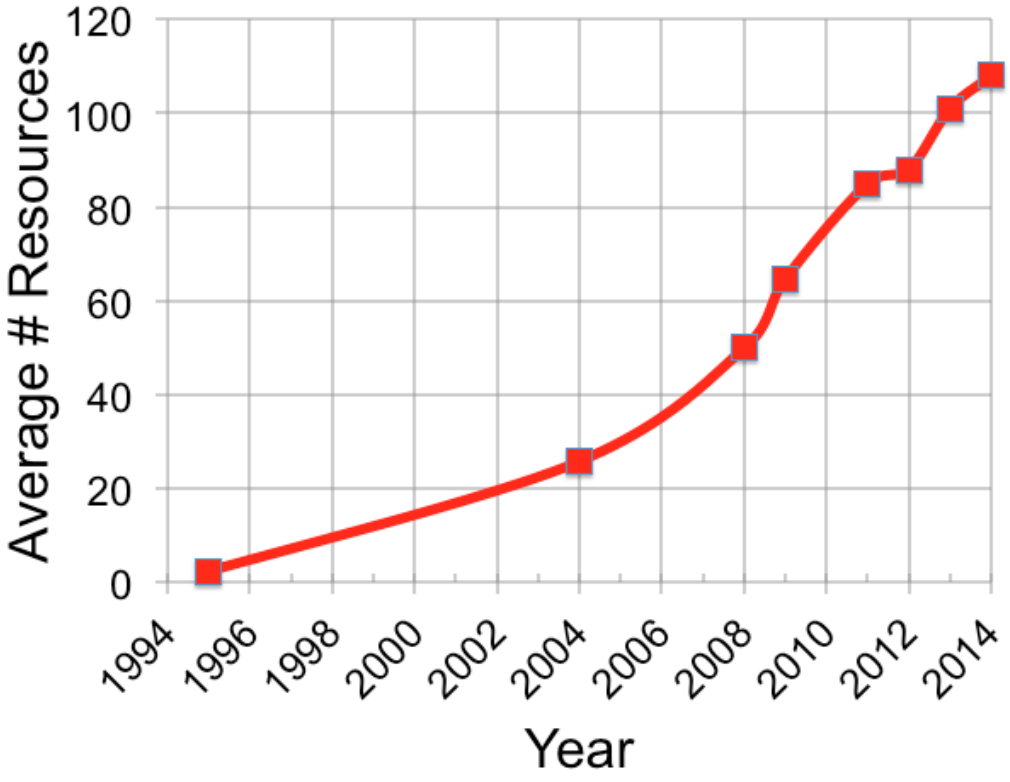


Reducing Load Time is Not Enough

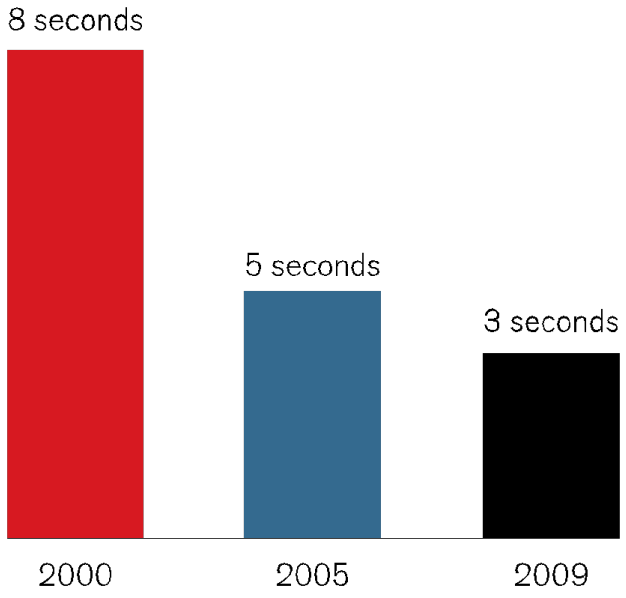


Reducing Load Time is Not Enough

Rising Website Complexity



Falling User Tolerance



Source: Akamai

Our Approach:
Dynamic Reprioritization

- Forseeable Future: High load times norm, not exception

Our Approach:
Dynamic Reprioritization

- Forseeable Future: High load times norm, not exception
- Reformulate the problem:

Our Approach: Dynamic Reprioritization

- Forseeable Future: High load times norm, not exception
- Reformulate the problem:

  How to reduce page load time?

Our Approach:
Dynamic Reprioritization

- Forseeable Future: High load times norm, not exception
- Reformulate the problem:

  ~~How to reduce page load time?~~

  How to ***Prioritize*** the content most ***important*** to user?

Our Approach: Dynamic Reprioritization

- Forseeable Future: High load times norm, not exception
- Reformulate the problem:

  ~~How to reduce page load time?~~

  How to ***Prioritize*** the content most ***important*** to user?

  - Typical tolerance limit of 2-4 seconds
  - Deliver “high utility” resources within time budget

Our Approach: Dynamic Reprioritization

- Forseeable Future: High load times norm, not exception
- Reformulate the problem:

  ~~How to reduce page load time?~~

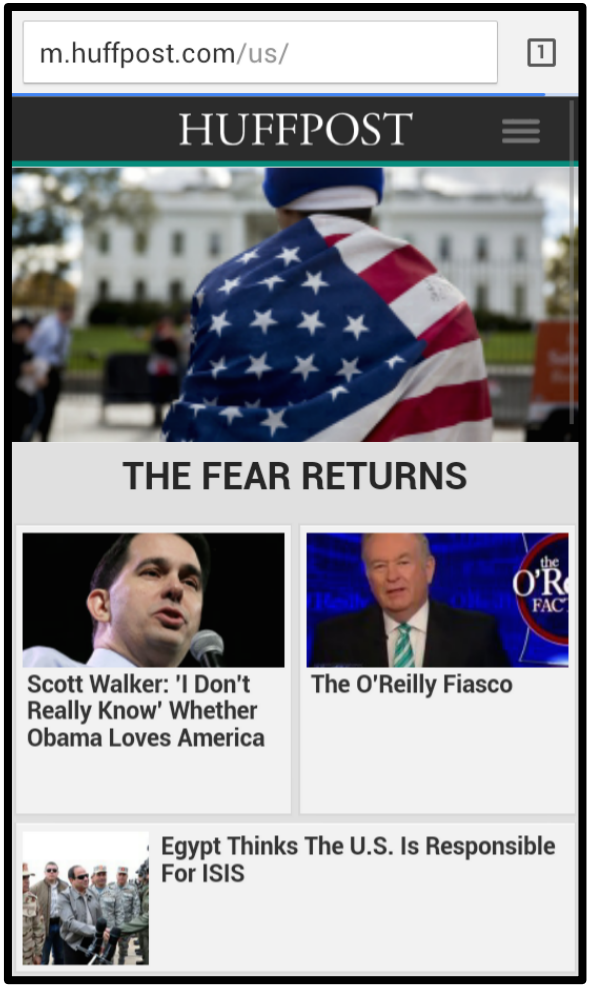
  How to ***Prioritize*** the content most ***important*** to user?

  - Typical tolerance limit of 2-4 seconds
  - Deliver “high utility” resources within time budget
- Our solution: Klotski proxy
  - No modifications to clients and web servers!

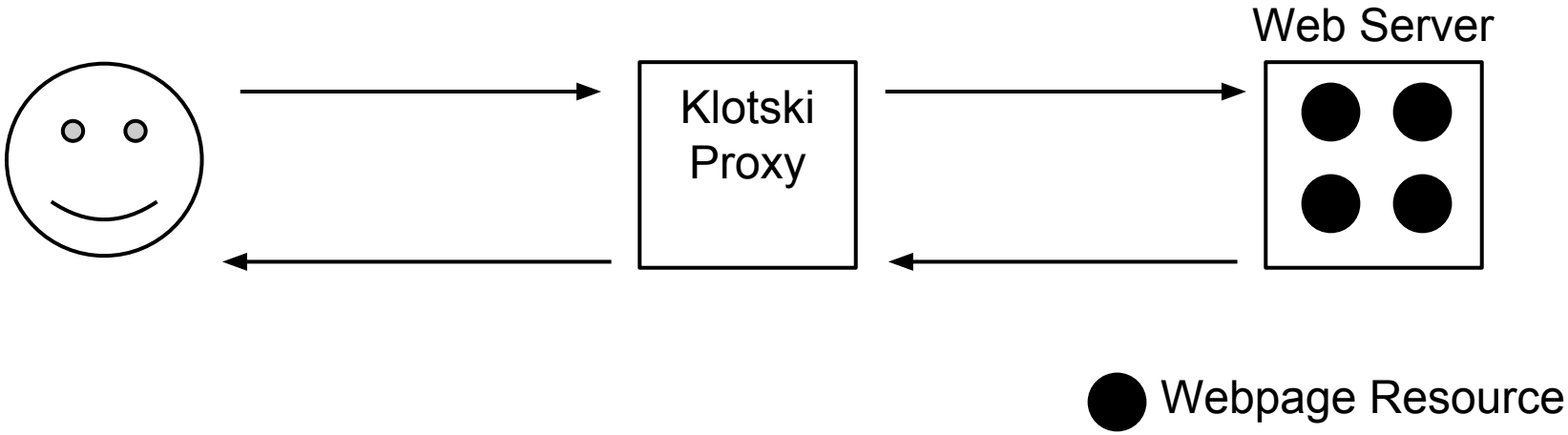
Klotski in Action!

Original Page Load at 3s

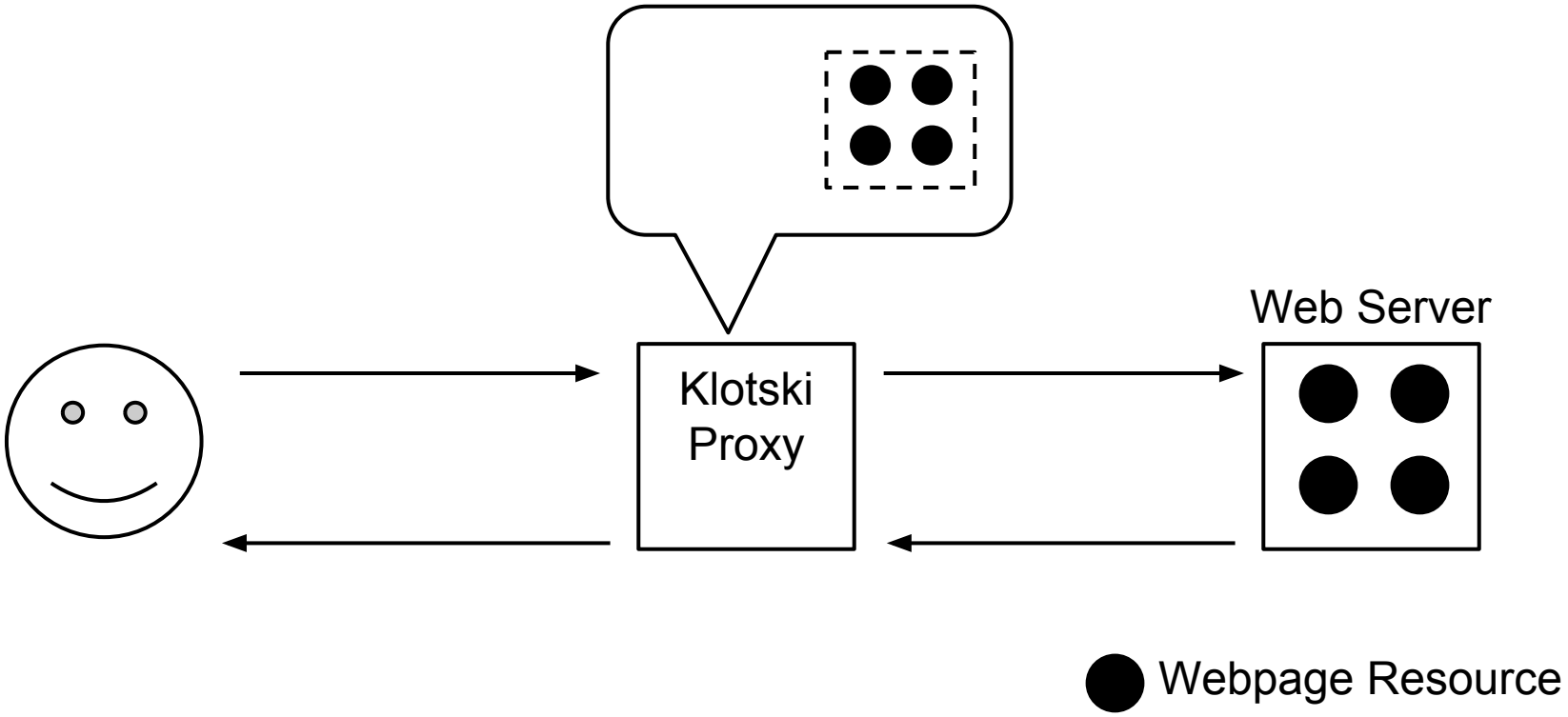
Klotski Page Load at 3s



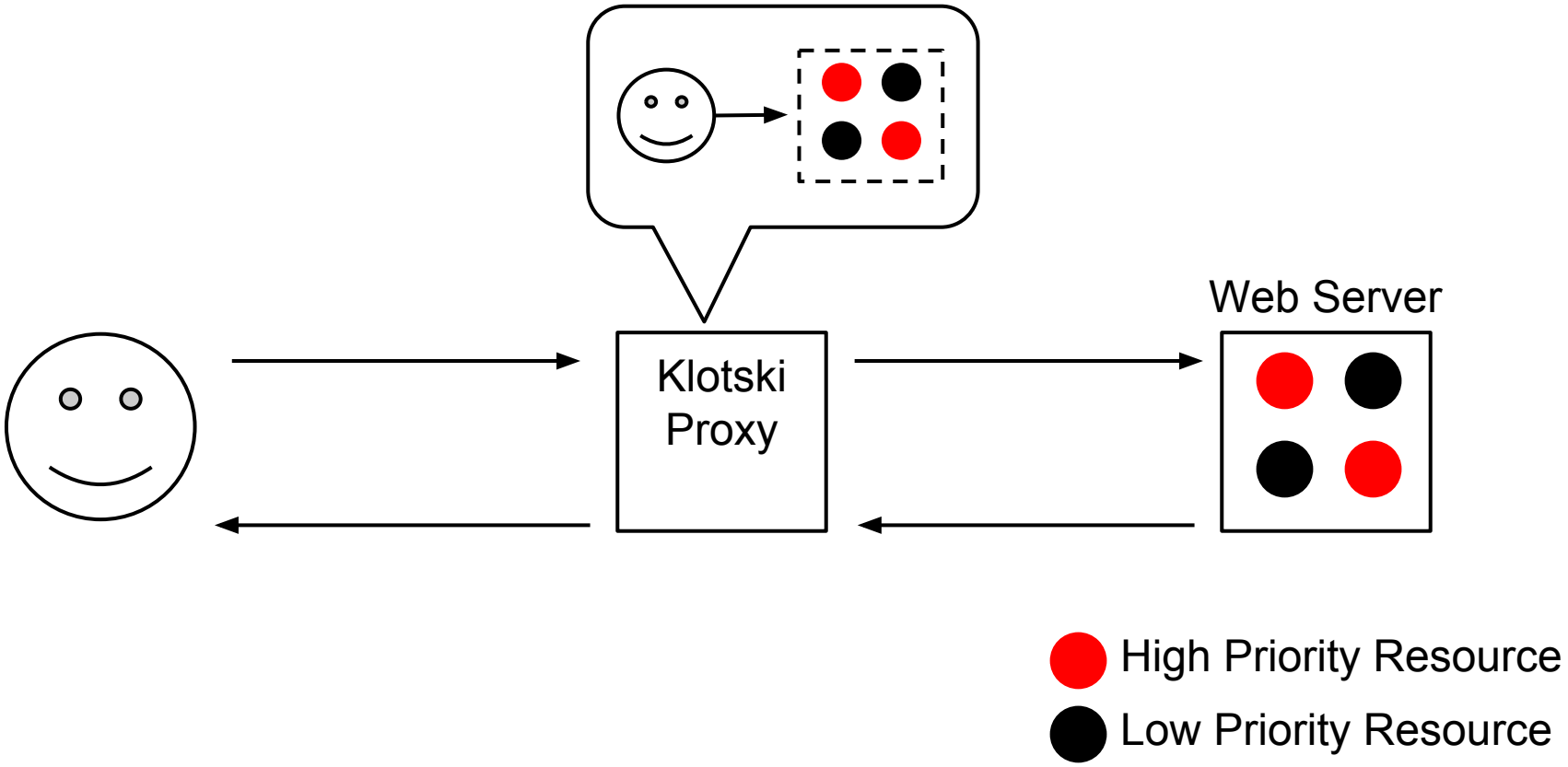
Klotski: Idealized View



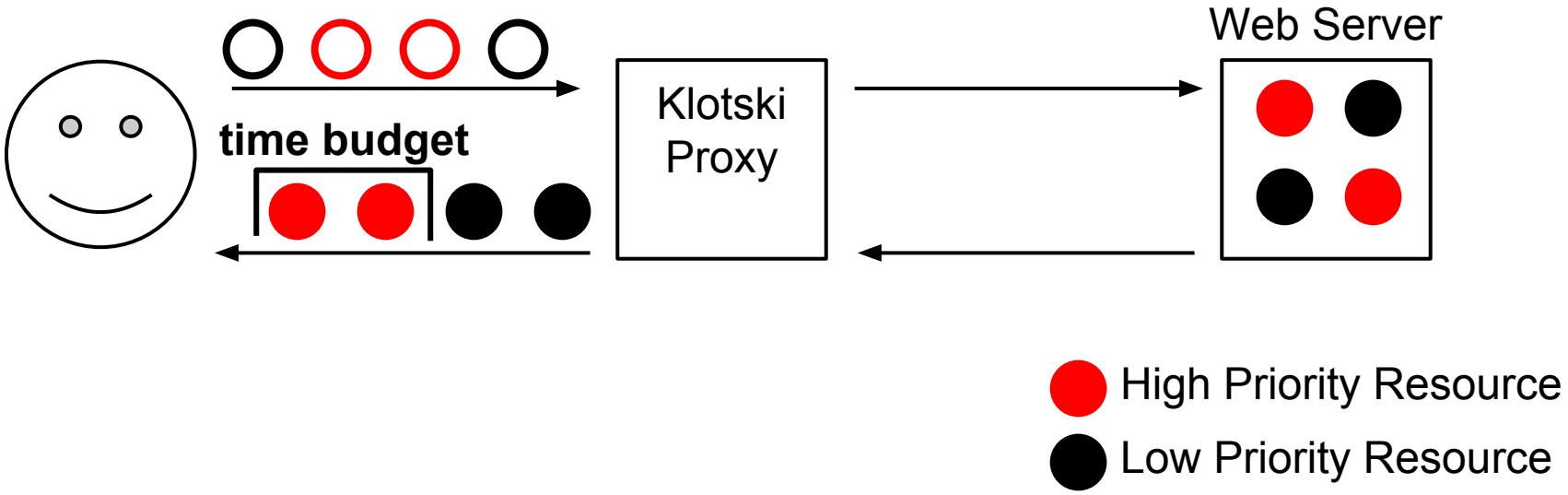
Klotski: Idealized View



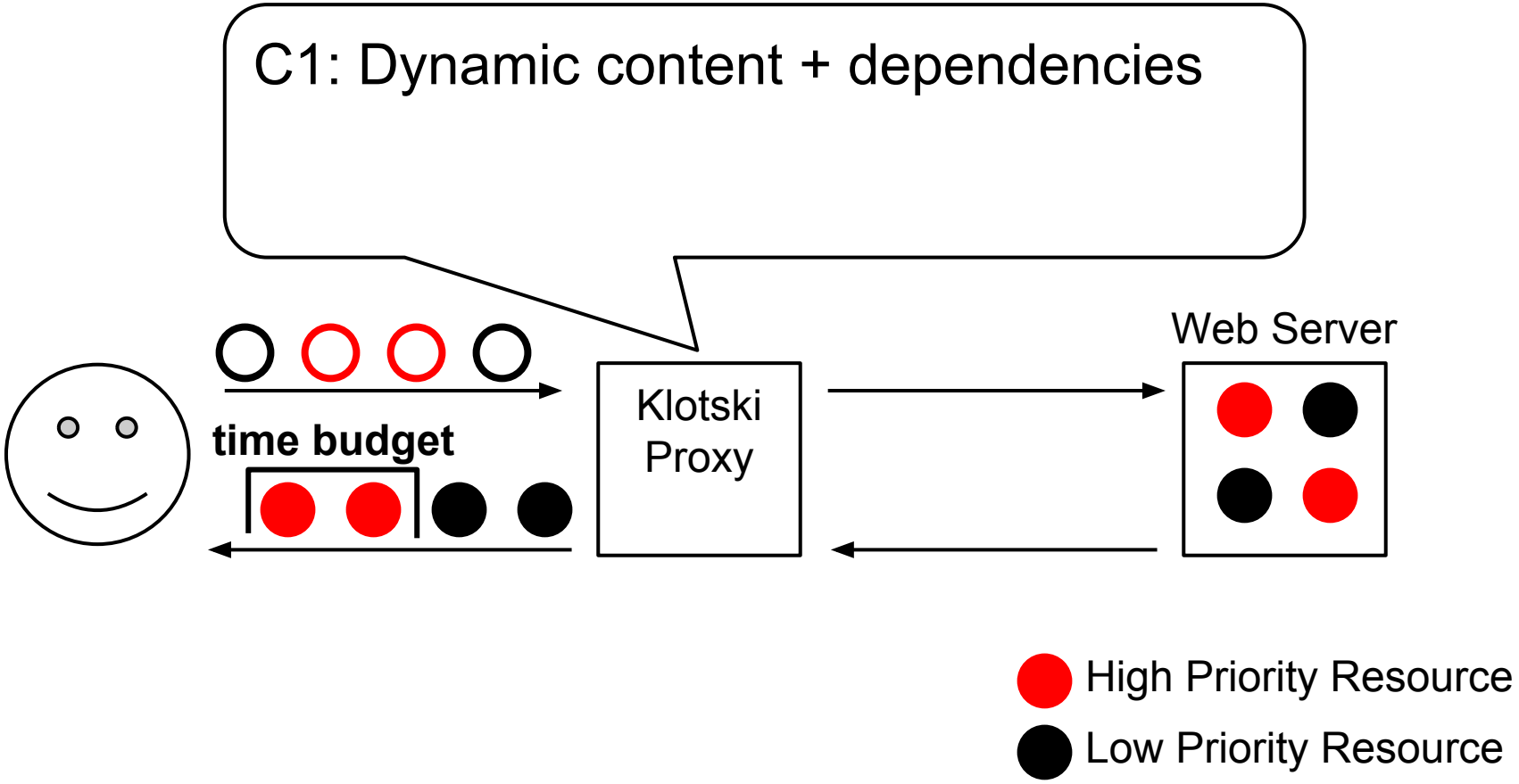
Klotski: Idealized View



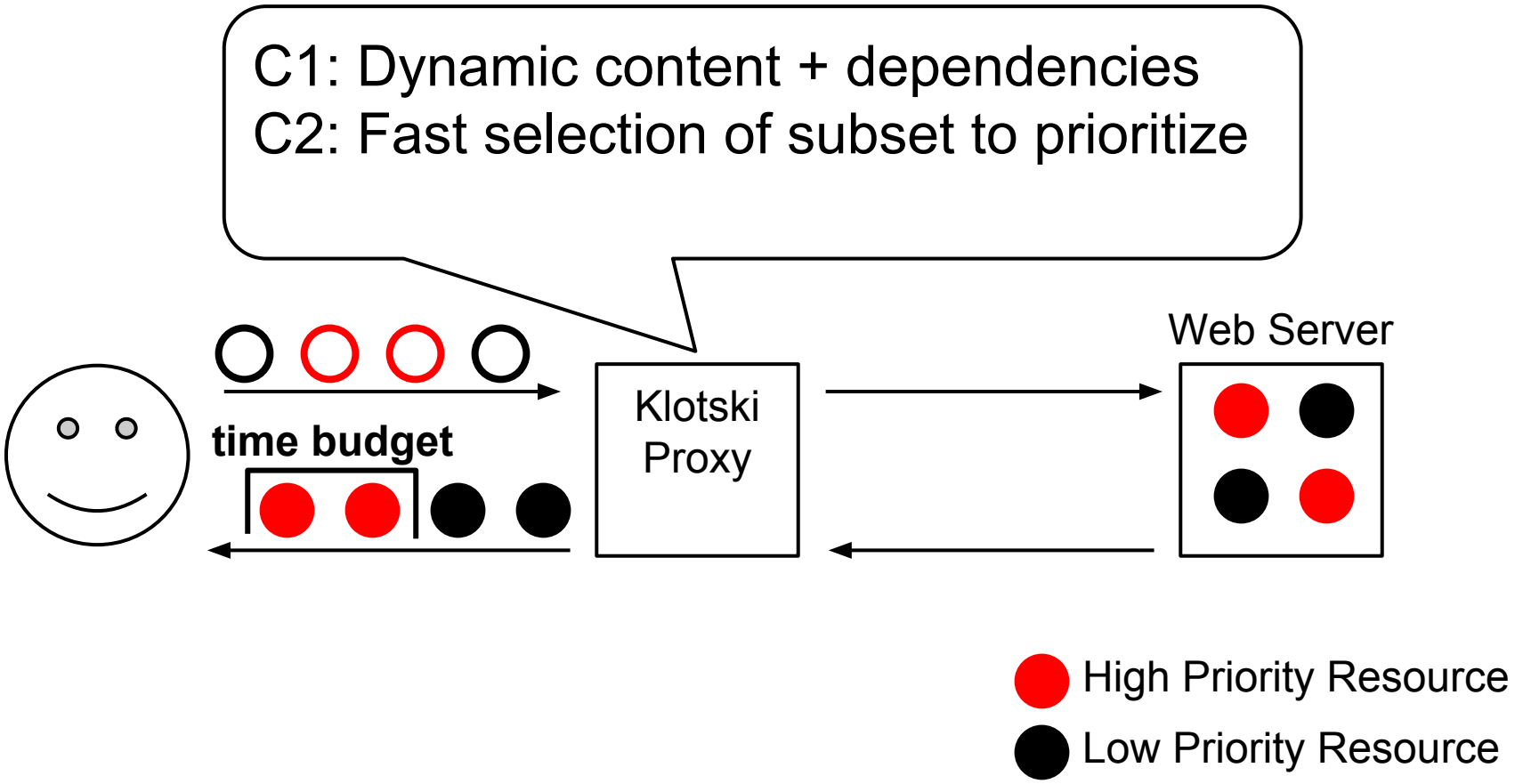
Klotski: Idealized View



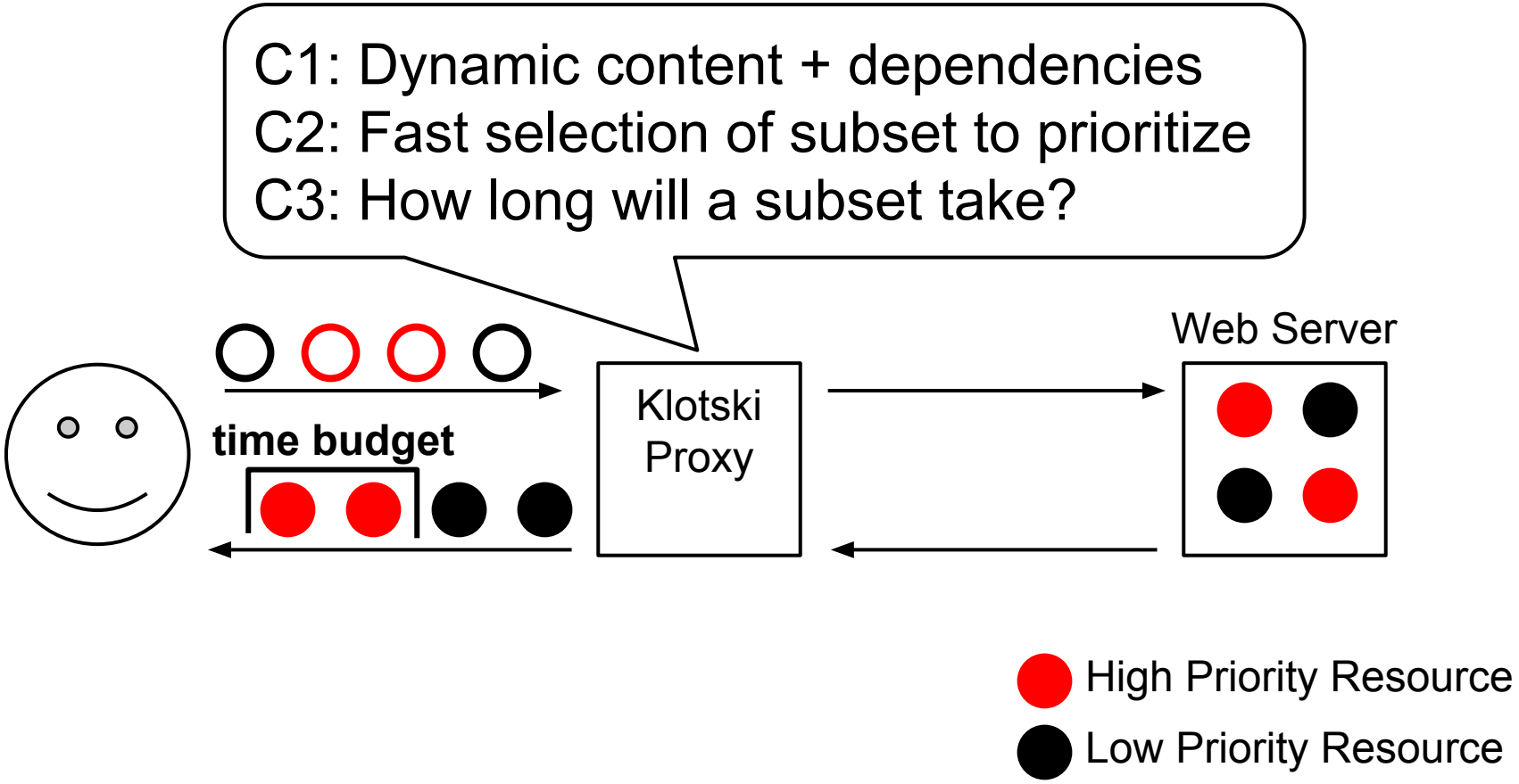
Challenges with Idealized View



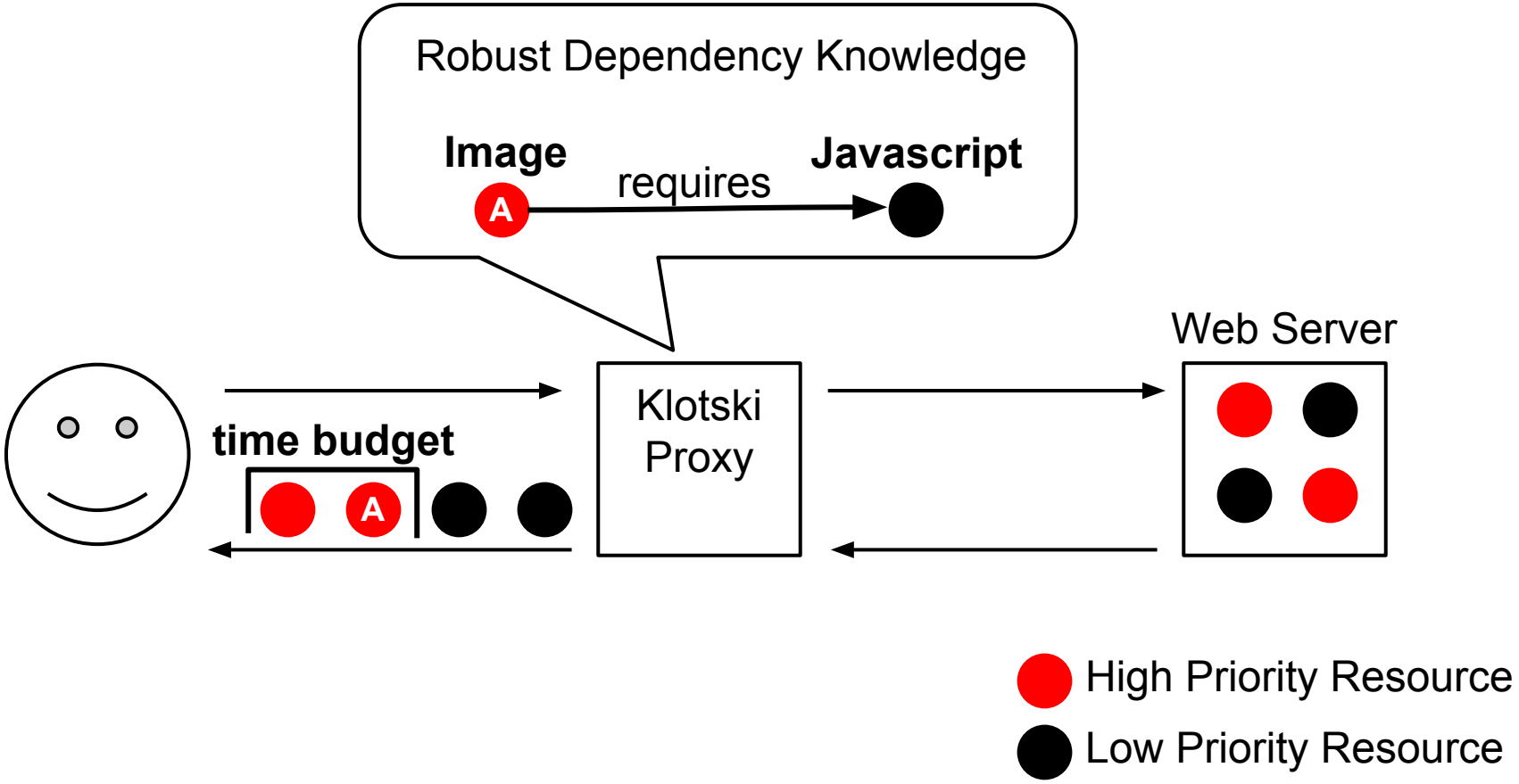
Challenges with Idealized View



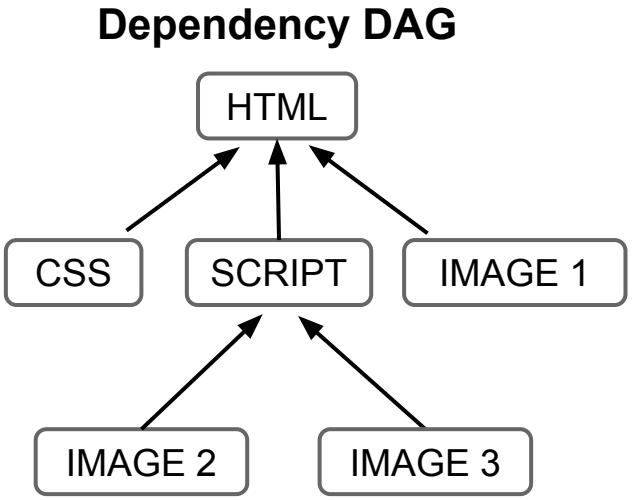
Challenges with Idealized View



Challenge 1:
Dynamic Content and Dependencies

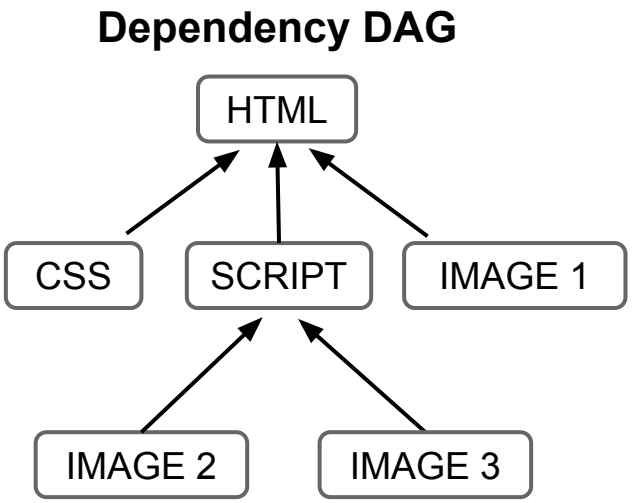


Intuition: Page structure is stable



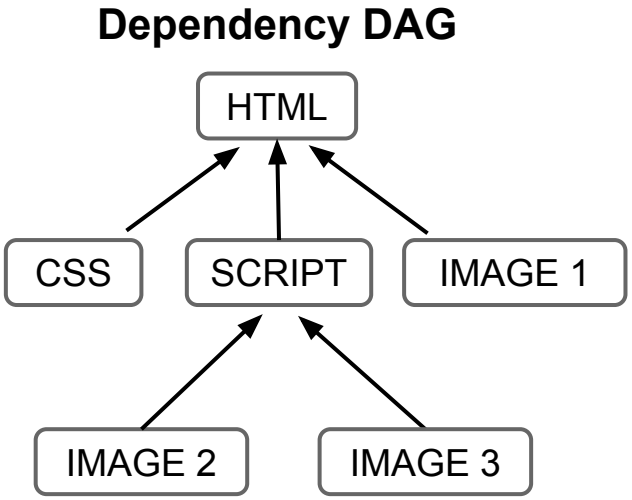
- Prior work on static dependencies
  - E.g., WebProphet, WProf

Intuition: Page structure is stable



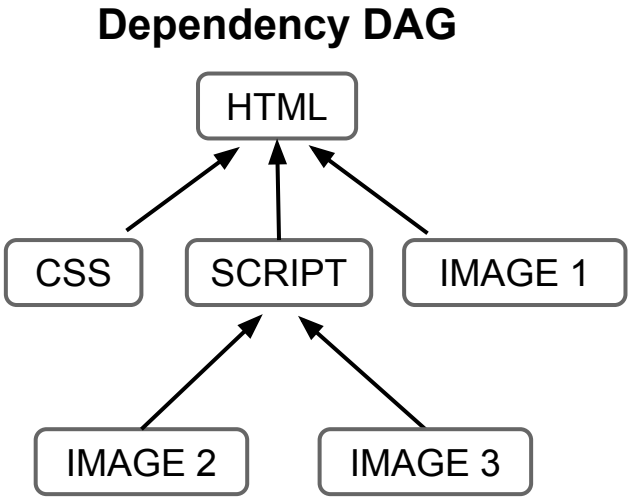
- Prior work on static dependencies
  - E.g., WebProphet, WProf
- Problem: Dependencies not reusable due to dynamic content

Intuition: Page structure is stable



- Prior work on static dependencies
  - E.g., WebProphet, WProf
- Problem: Dependencies not reusable due to dynamic content
- Our observation:
  - Nodes in DAG change
  - DAG structure largely stable

Intuition: Page structure is stable



- Prior work on static dependencies
  - E.g., WebProphet, WProf
- Problem: Dependencies not reusable due to dynamic content
- Our observation:

Load page repeatedly to generate *fingerprint*:

- DAG structure with a URL pattern at every node
- Pattern generalizes URL of dynamic resources

le

Learn URL Patterns

- Generalize known prior URLs of a dynamic resource

foo.com/SG39HZ78/a.js
foo.com/SHFS2732/a.js
⟶ foo.com/*/a.js

Learn URL Patterns

- Generalize known prior URLs of a dynamic resource

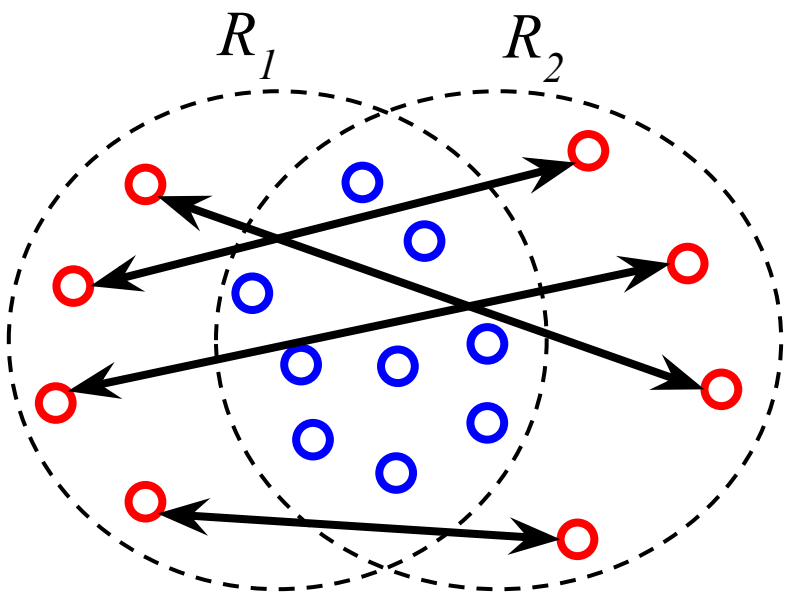
  foo.com/SG39HZ78/a.js
  foo.com/SHFS2732/a.js
  ⟶ foo.com/*/a.js

- 3 Cases = 90% of Replacements:
  - Single token in URL changes
  - Only URL argument changes: www.site.org/a.js?FOO=1...
  - CDN node name: {CDN2.bar.com/x.jpg, CDN5.bar.com/x.jpg}

Identify Resource Replacements

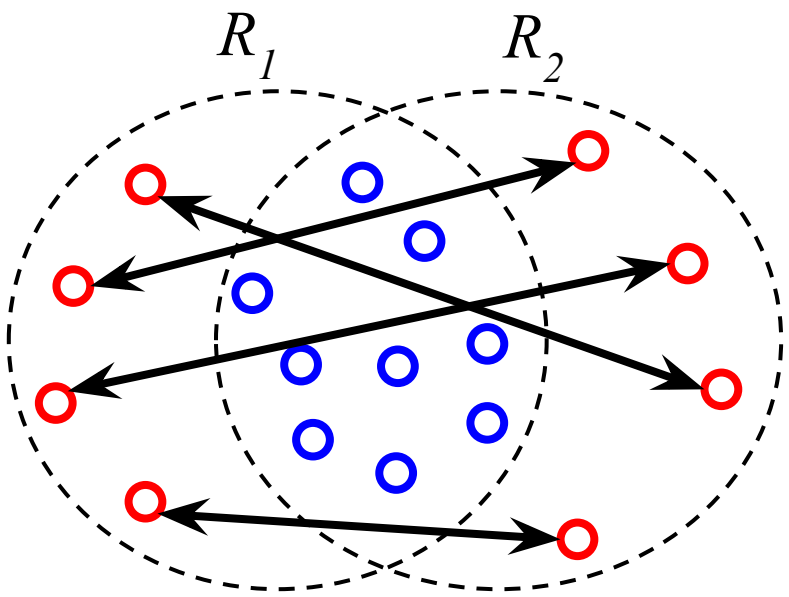
- Capture set of prior URLs:
  Track Replacement of Resource
  over multiple page loads

Identify Resource Replacements



- Capture set of prior URLs:
  Track Replacement of Resource
  over multiple page loads

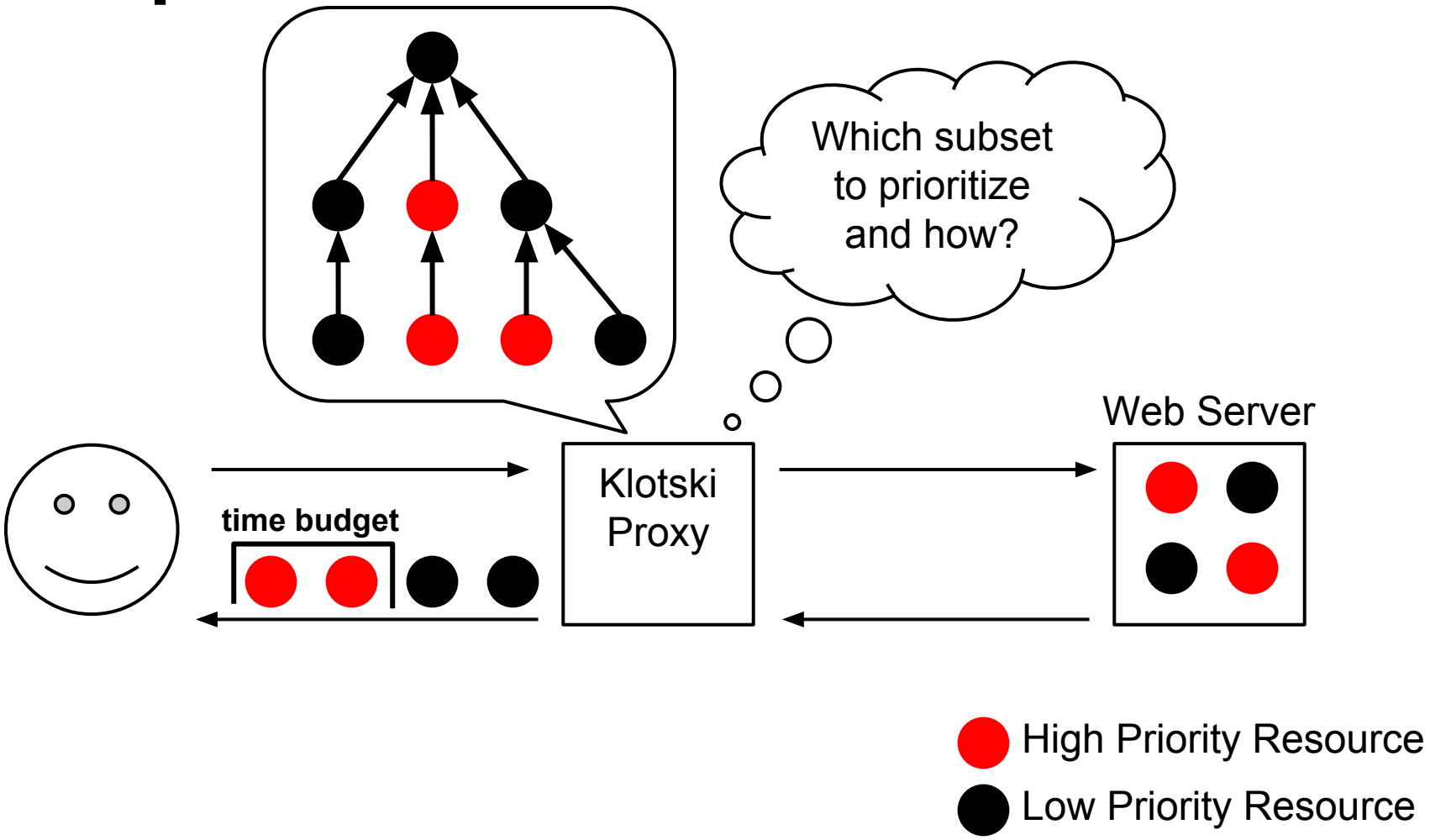
Identify Resource Replacements



- Capture set of prior URLs:
  Track Replacement of Resource
  over multiple page loads

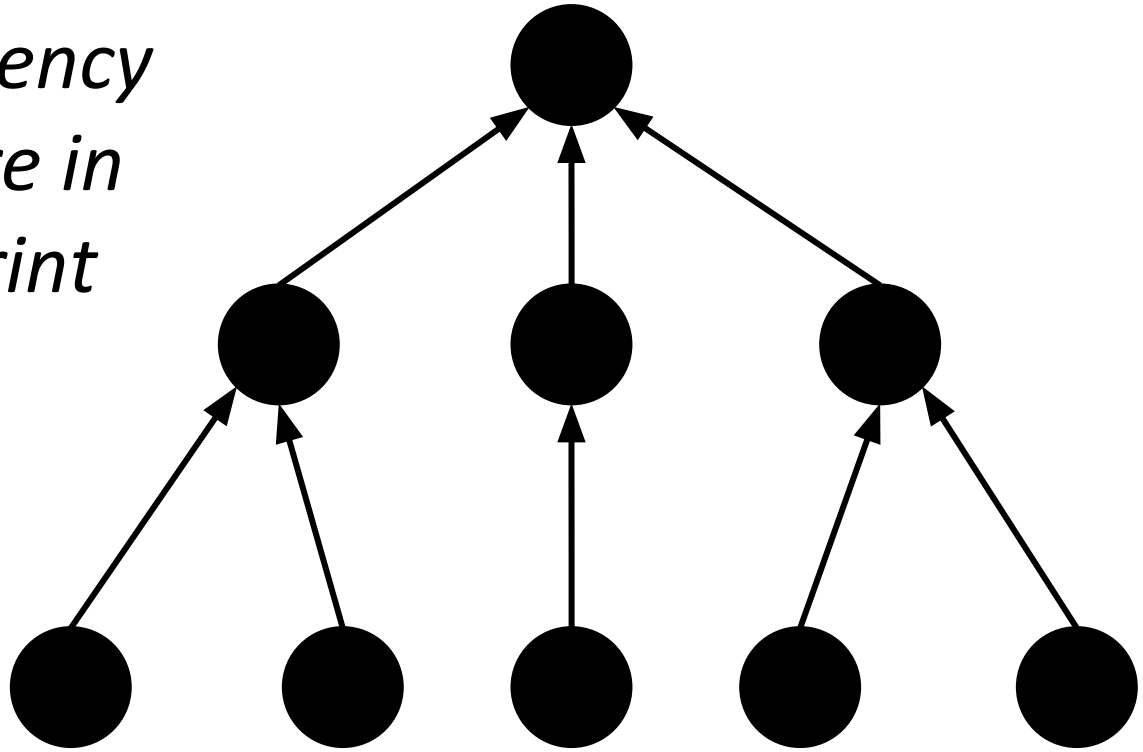
- Combination of techniques:
  - Match position in DAG dependency structure
  - Identical position on screen
  - Identical reference in source

Challenge 2: Optimal Prioritization Schedule



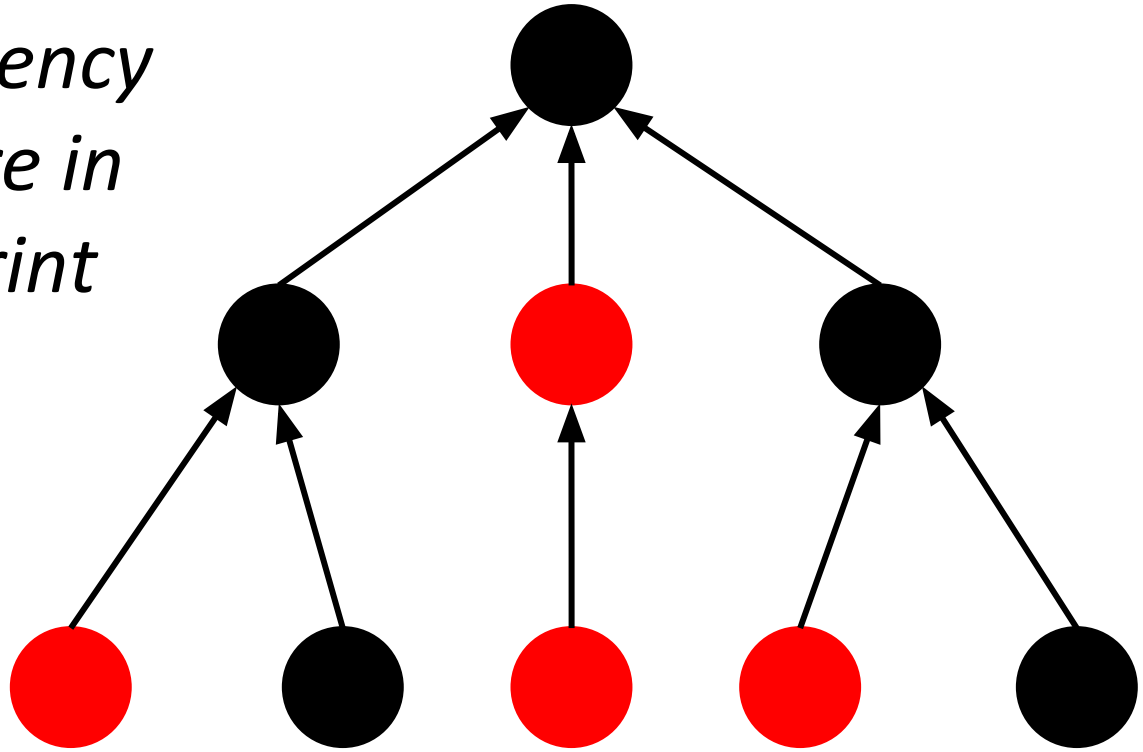
Select Subset of High Utility Resources

Dependency structure in fingerprint



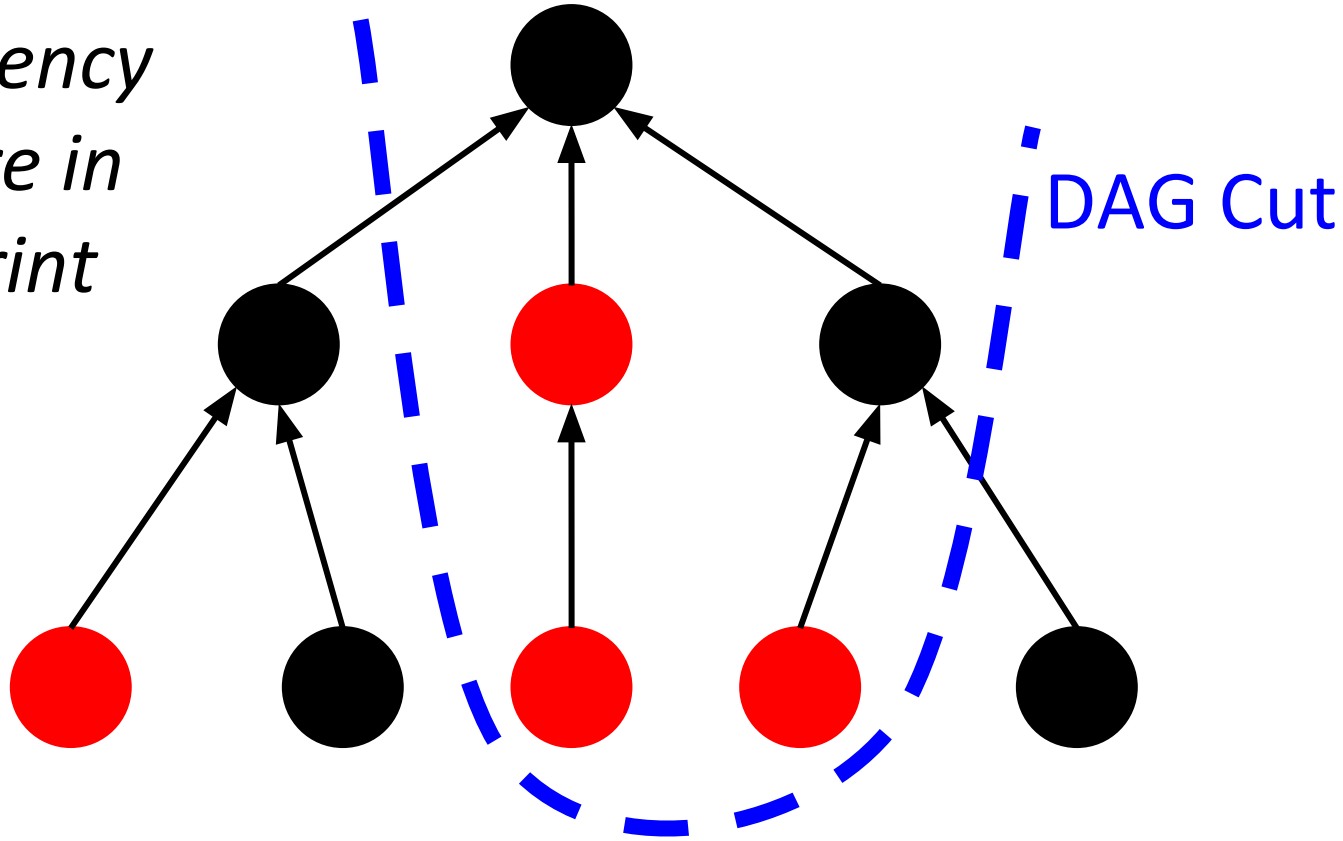
Select Subset of High Utility Resources

Dependency structure in fingerprint



Select Subset of High Utility Resources

Dependency structure in fingerprint



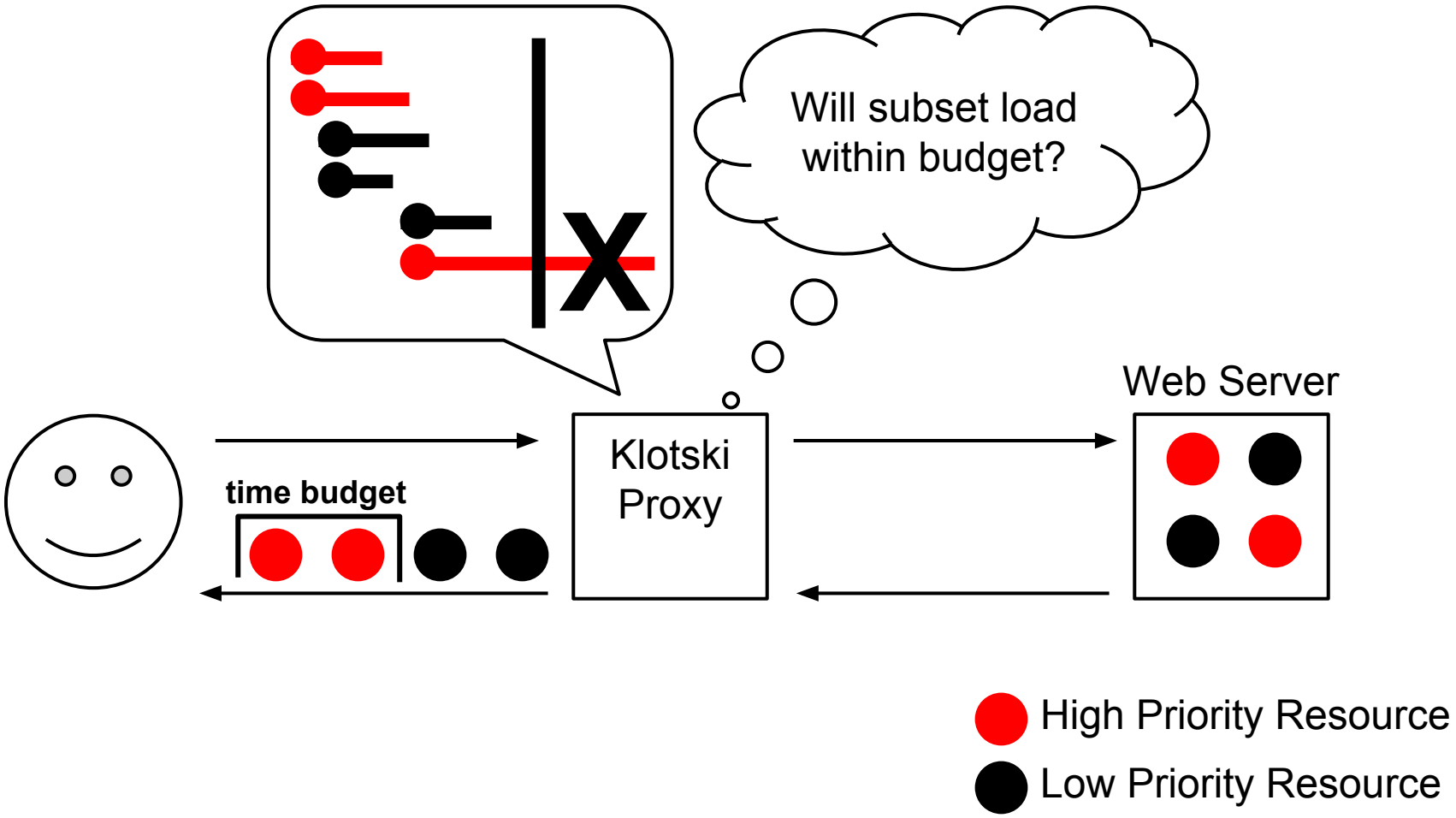
- Reduces to knapsack with dependencies: NP-Hard
- Apply greedy heuristic

Prioritizing High-Utility Resources

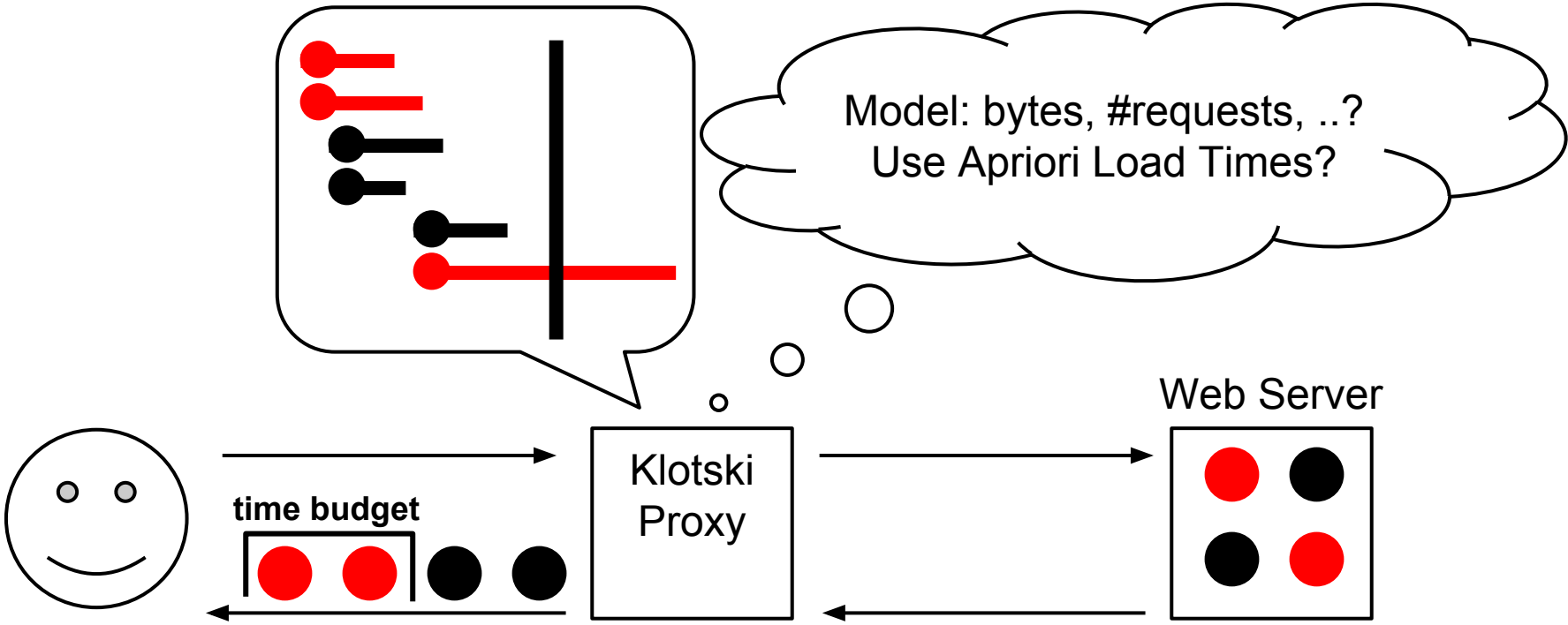
- Static URLs: Use ***SPDY PUSH*** to pre-emptively deliver as soon as main HTML is requested

- Dynamic URLs: Prioritize delivery if ***match with regular expression*** of a selected resource

Challenge 3:
Estimating Load Times in the “Wild”



Seemingly Natural Non-Solutions

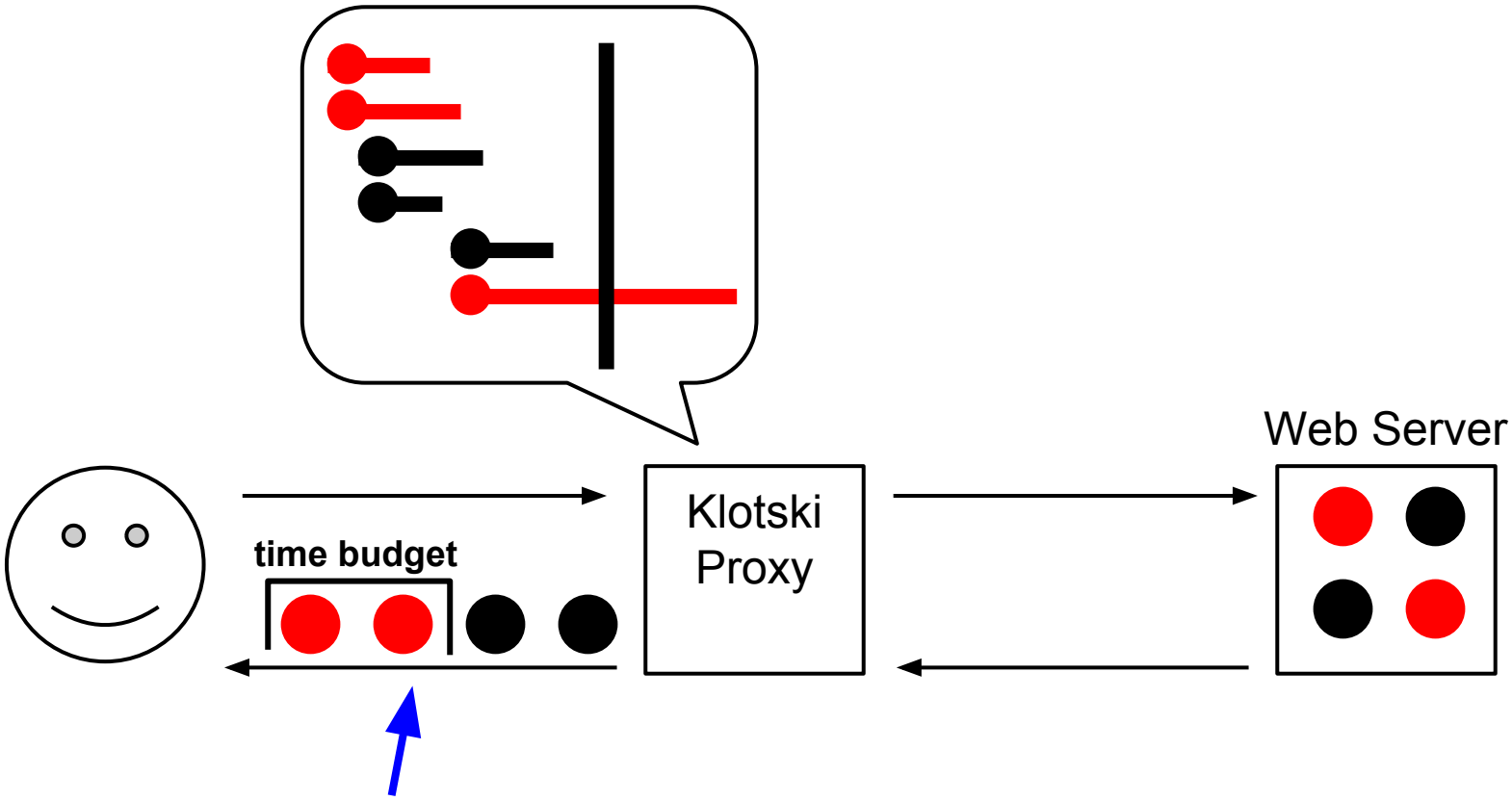


Model: Too simplistic to capture browser effects
Apriori Loads: No longer valid with reprioritization
Cannot capture diversity in client conditions

esource

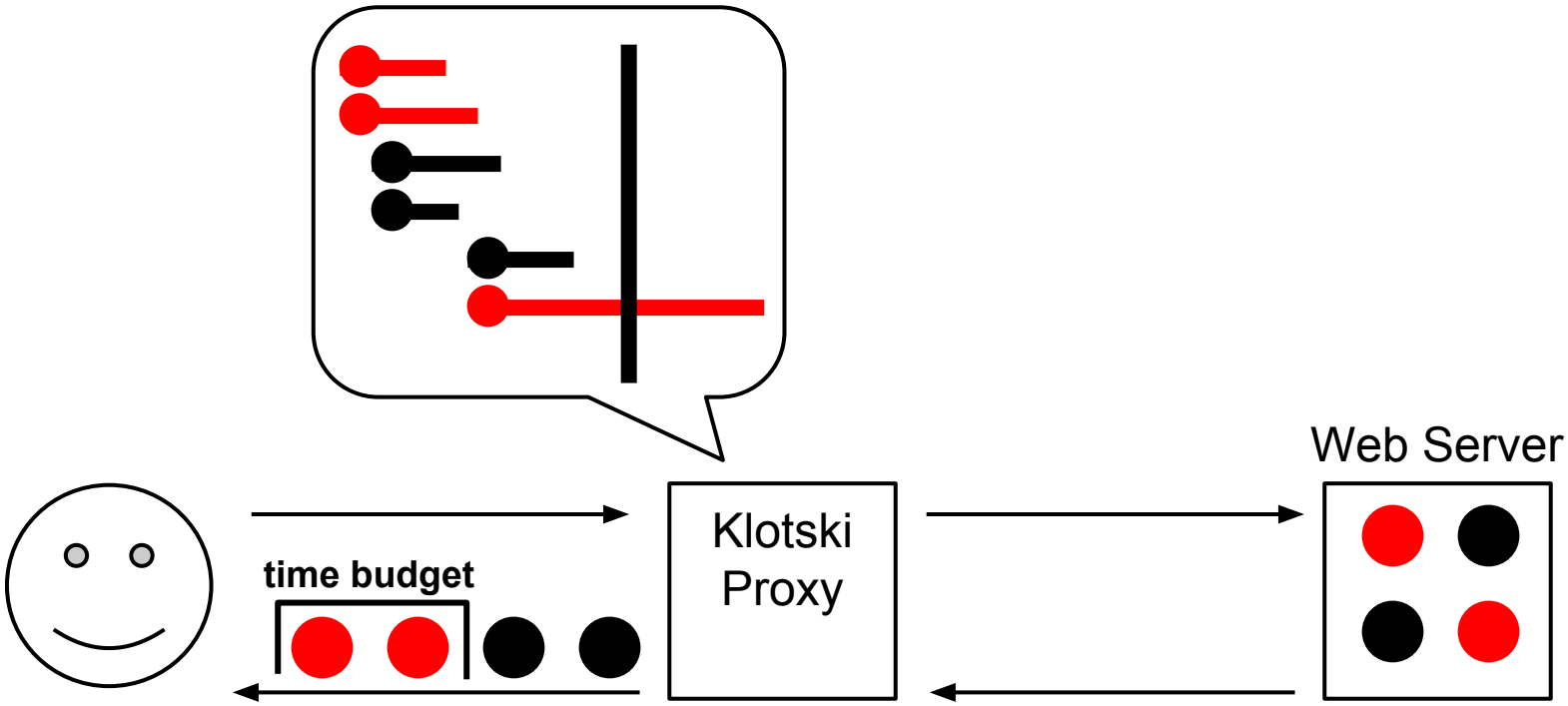
source

Intuition Behind Klotski Estimator



Bottleneck = Client-Proxy Link (e.g, 4G)

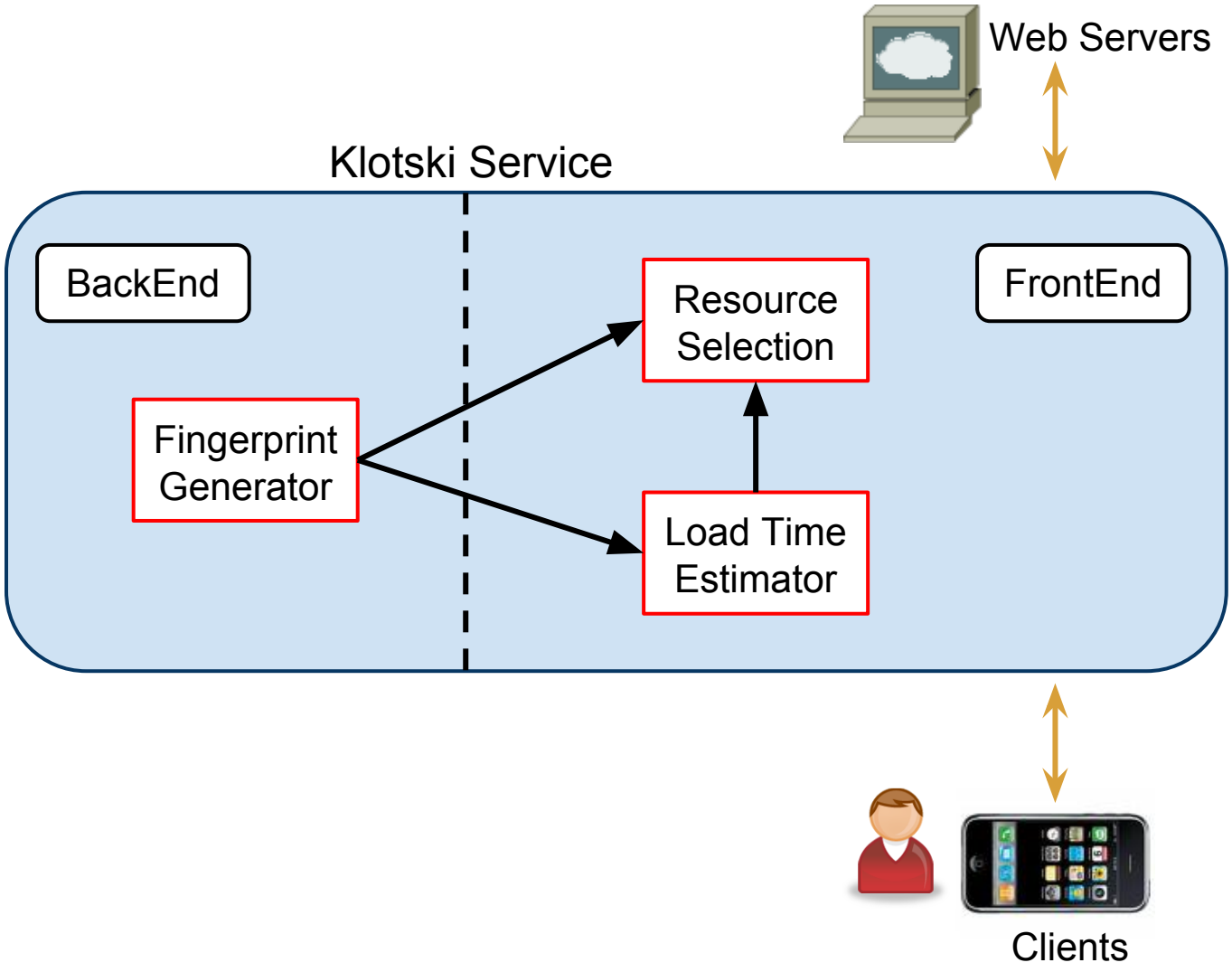
Intuition Behind Klotski Estimator



Build Fluid Model Simulation

- Proxy as work conserving scheduler
- Priorities/Dependencies, fairly shared bw for concurrent transfers
- Some subtle issues: PUSH, client processing delays

Klotski: System Architecture



Klotski: Experimental Results

Data Set

Websites

- 50 Random From Alexa Top 200

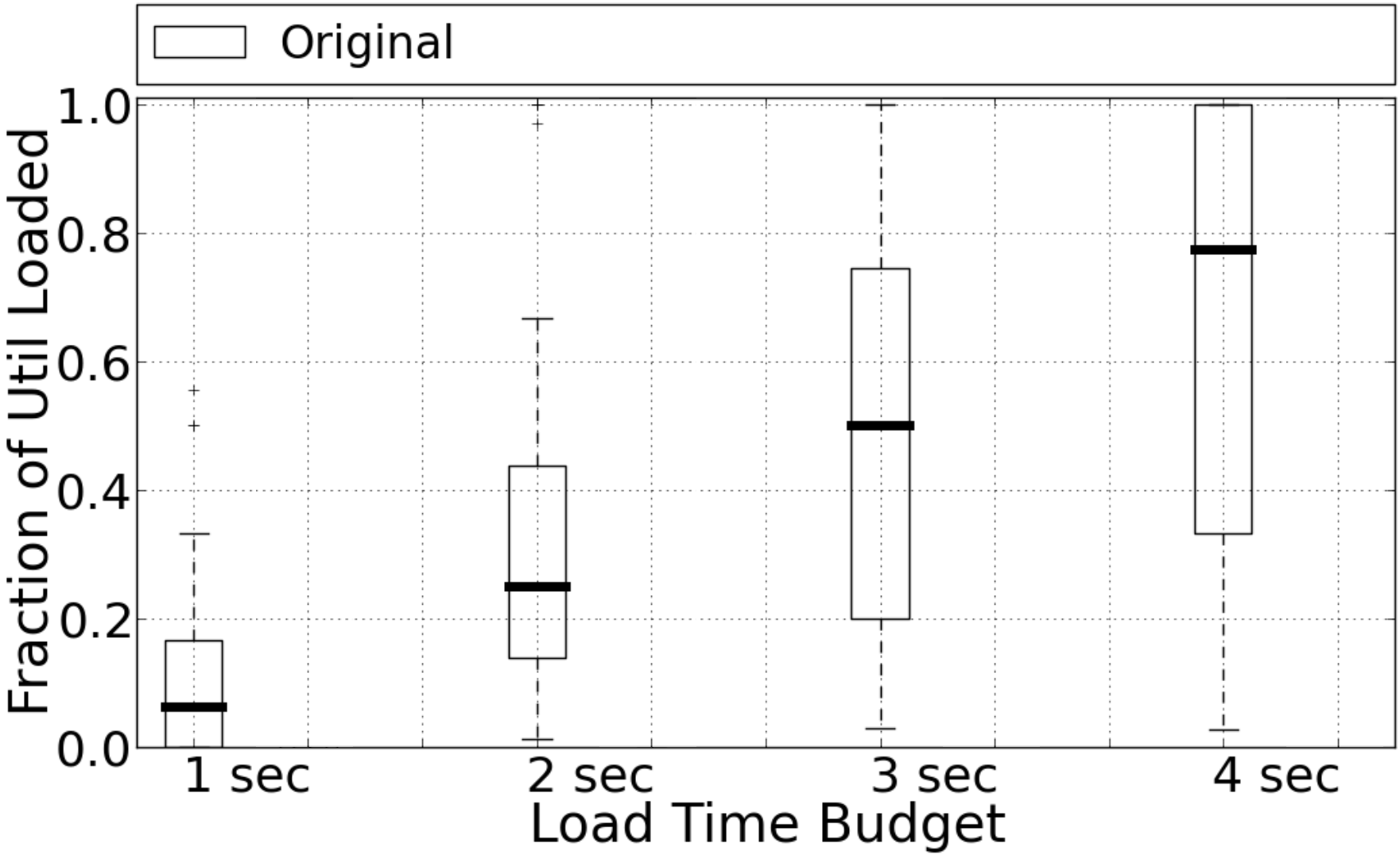
Mobile Device

- Android Smartphone
- 4G Connection
- Google Chrome

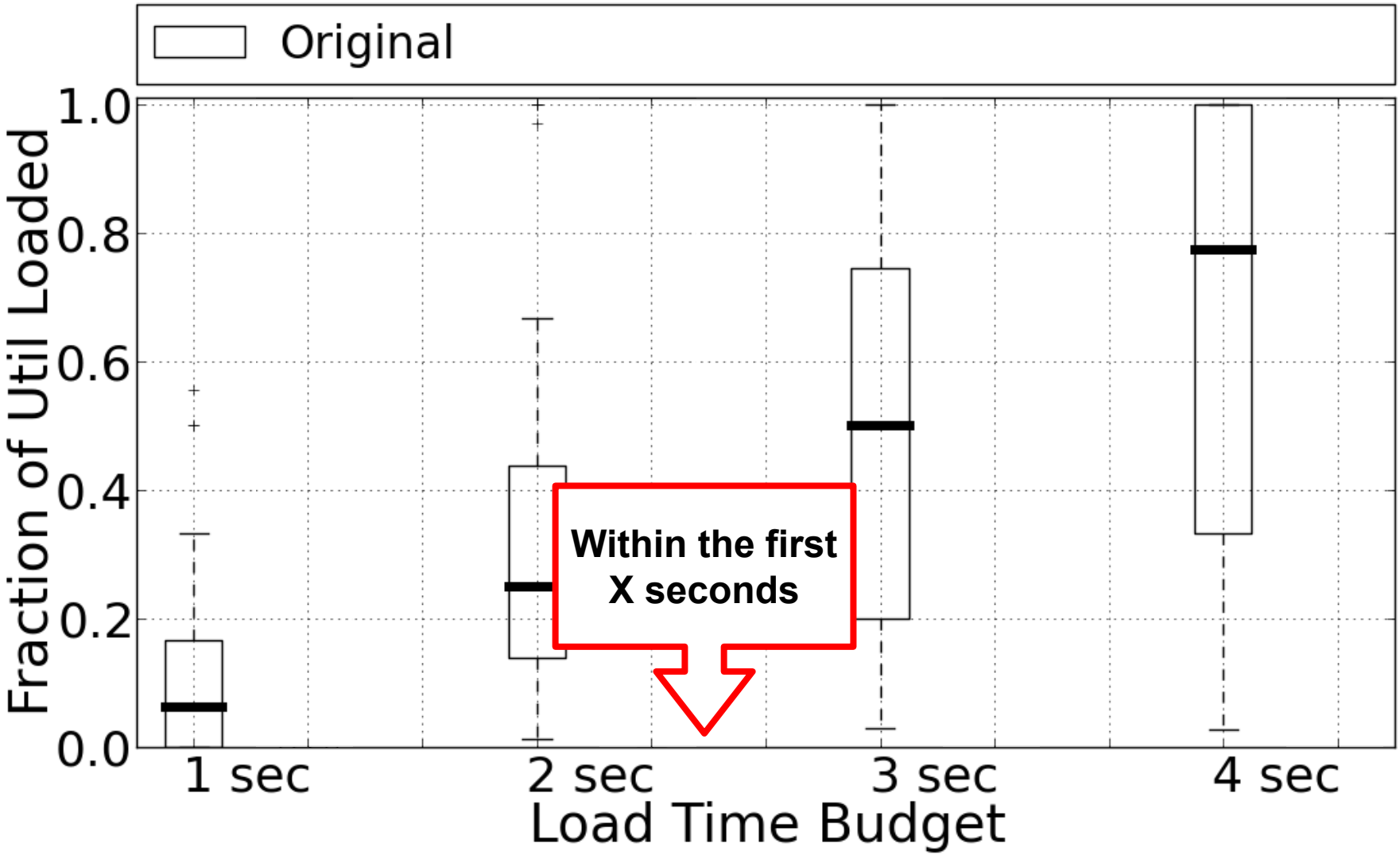
Paper Only Results

- Desktop PC + Ethernet
- Smartphone + Full Website
- User study on utility preferences
- Resource churn over time

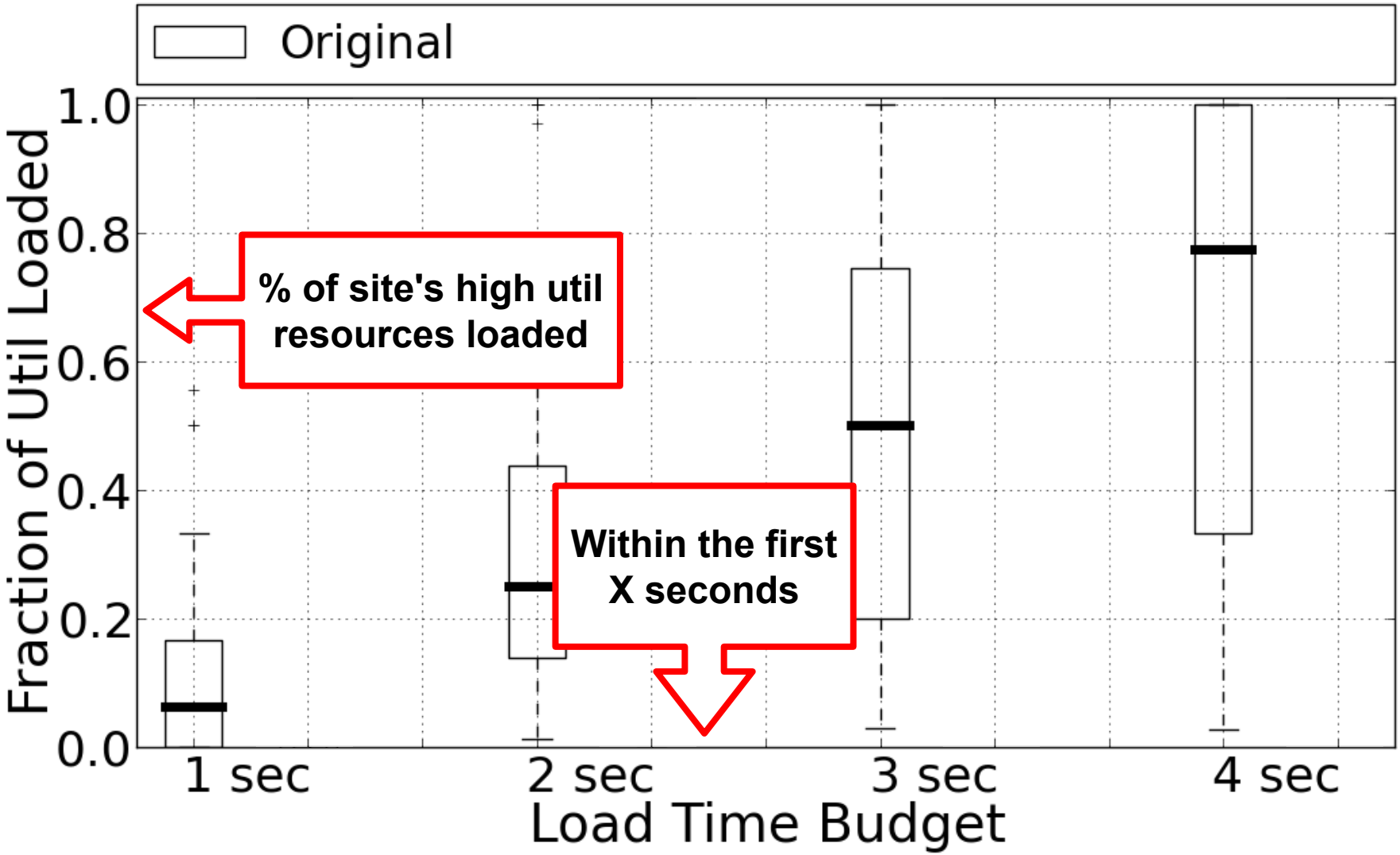
Klotski Improves User Experience



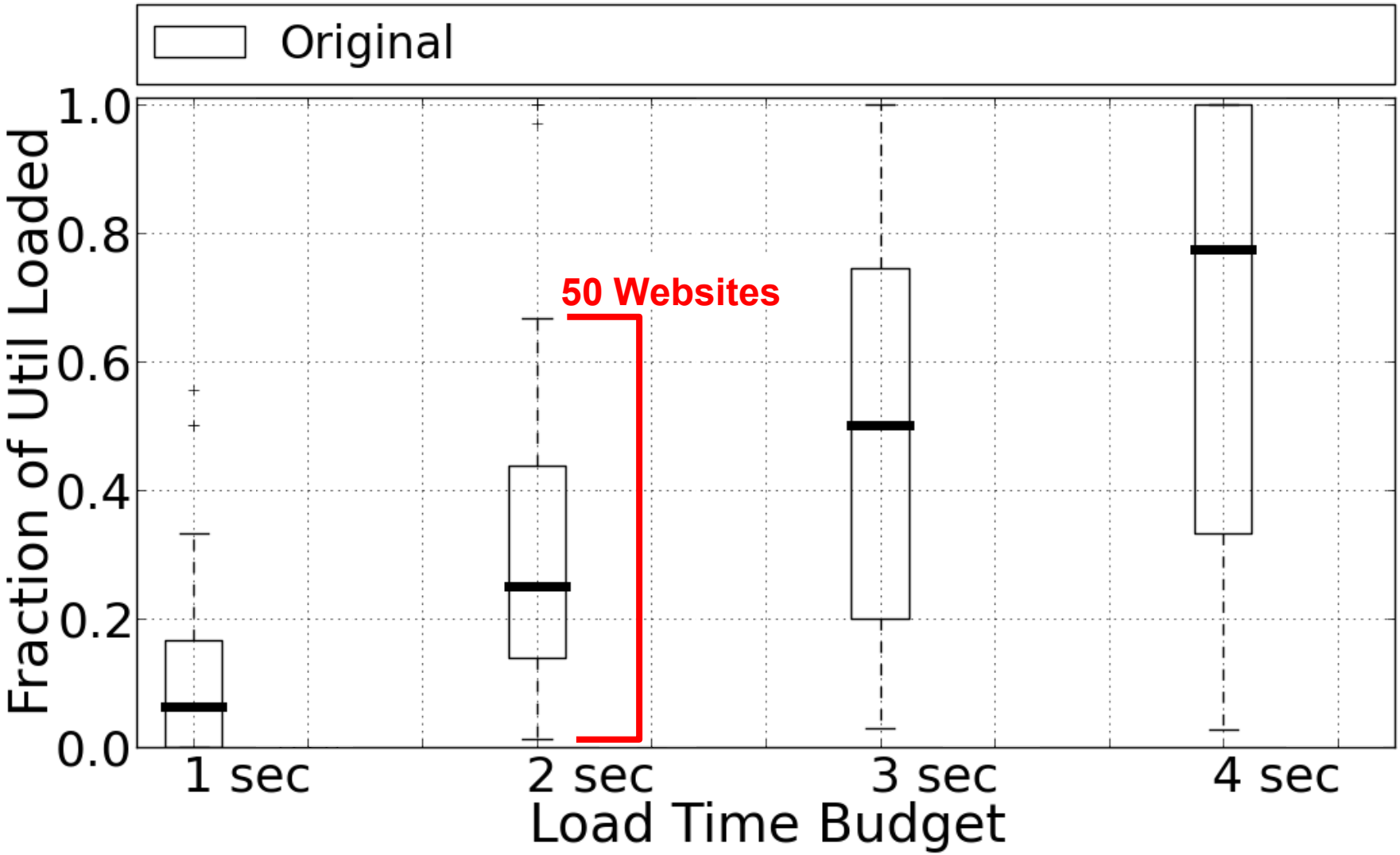
Klotski Improves User Experience



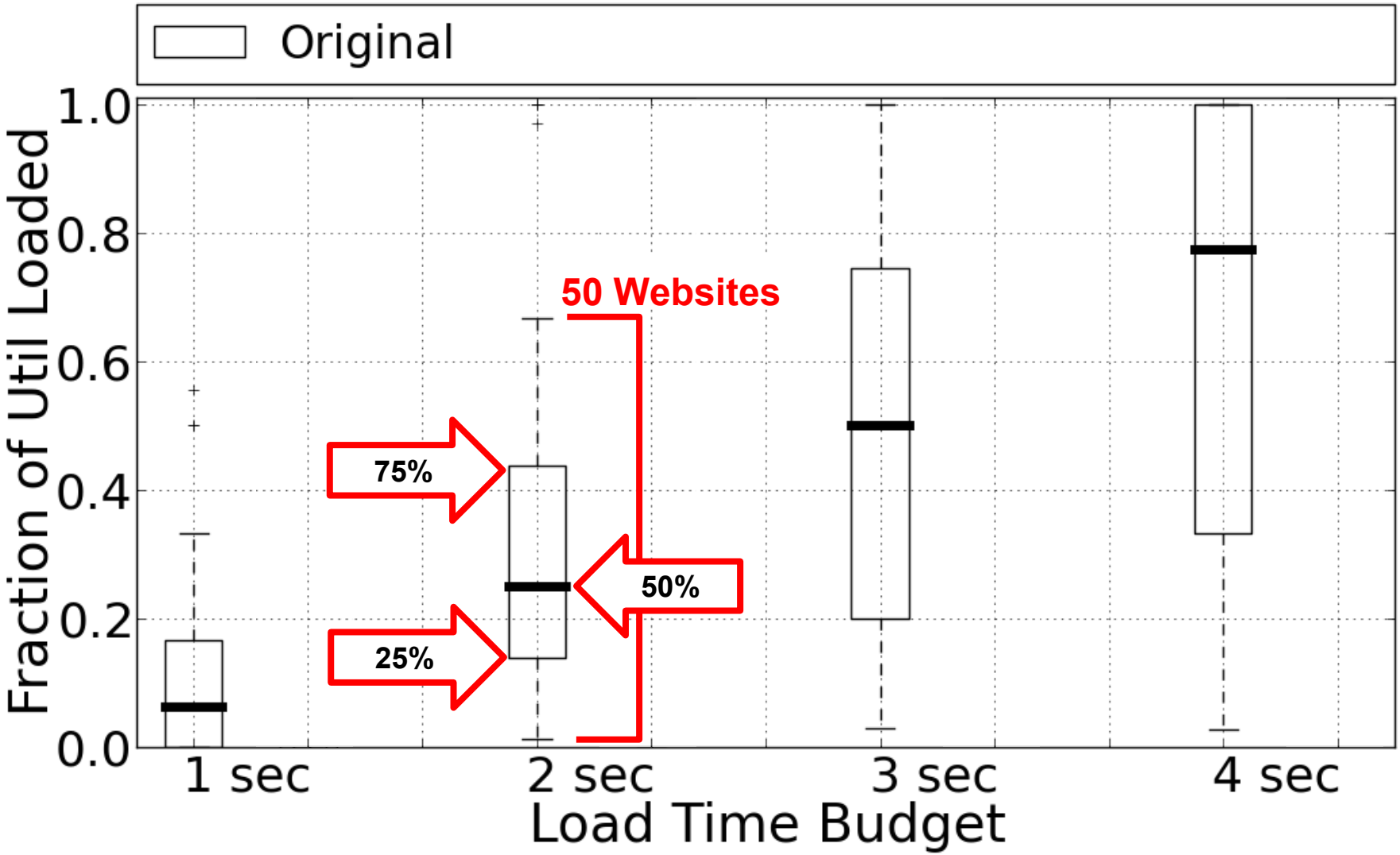
Klotski Improves User Experience



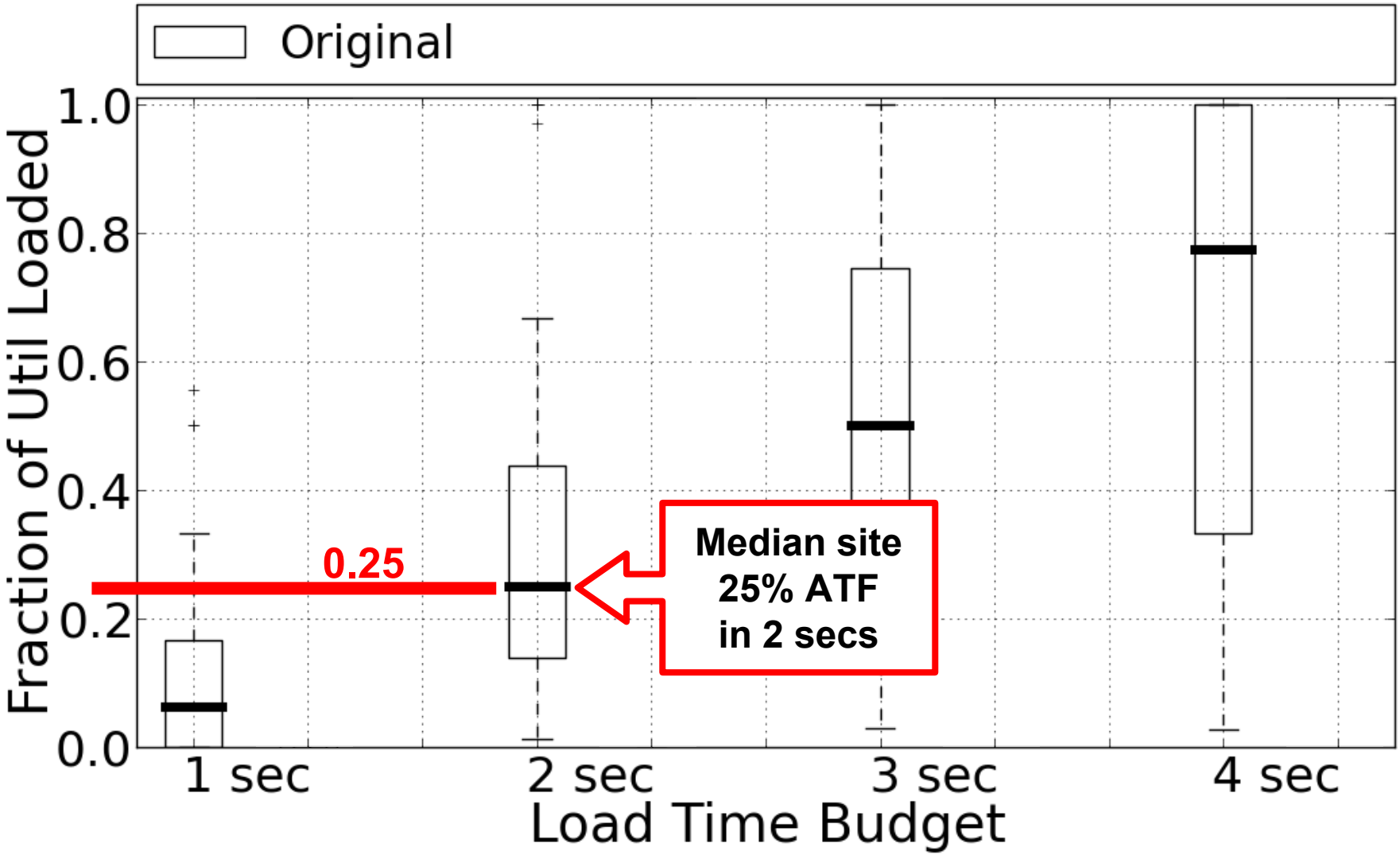
Klotski Improves User Experience



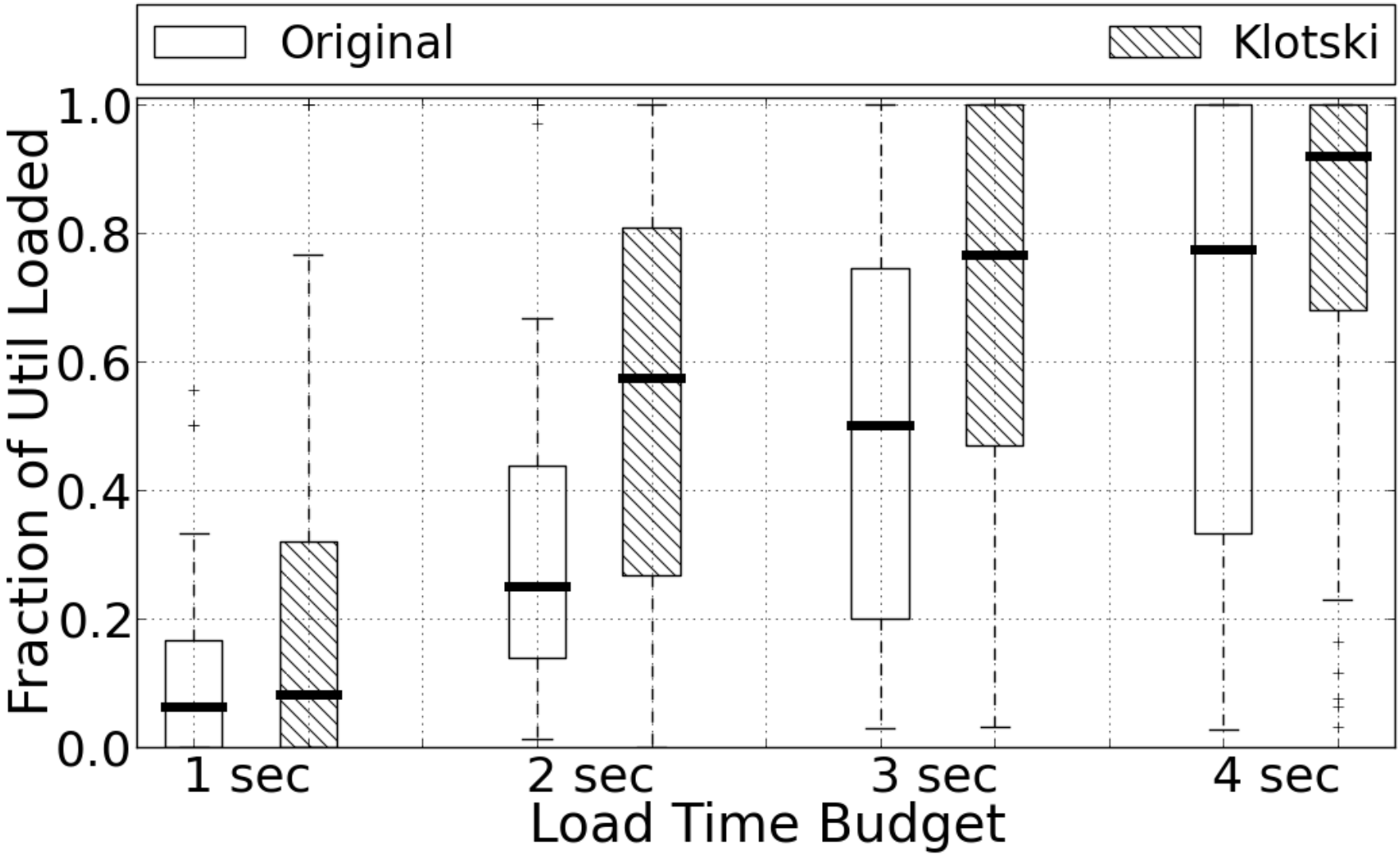
Klotski Improves User Experience



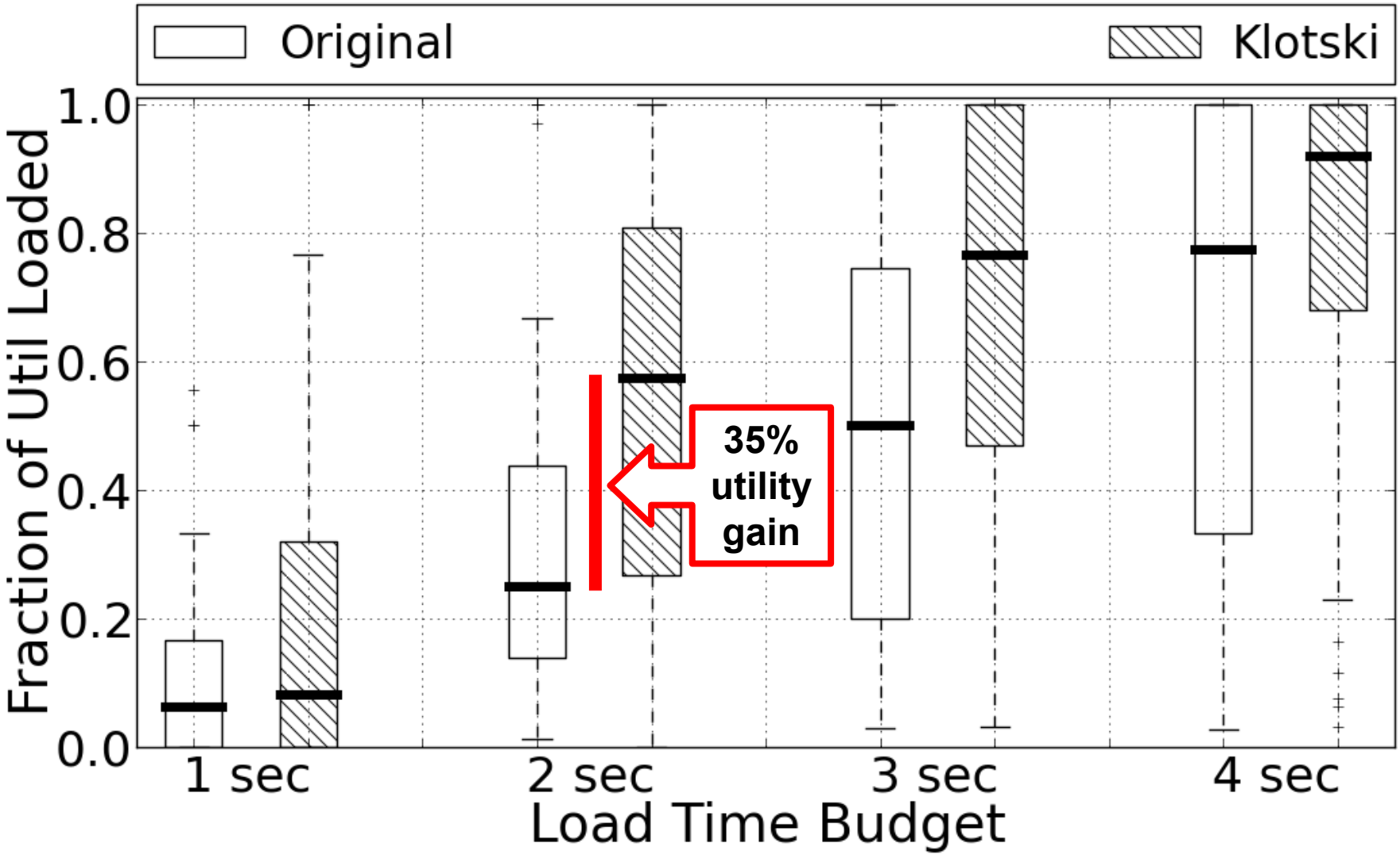
Klotski Improves User Experience



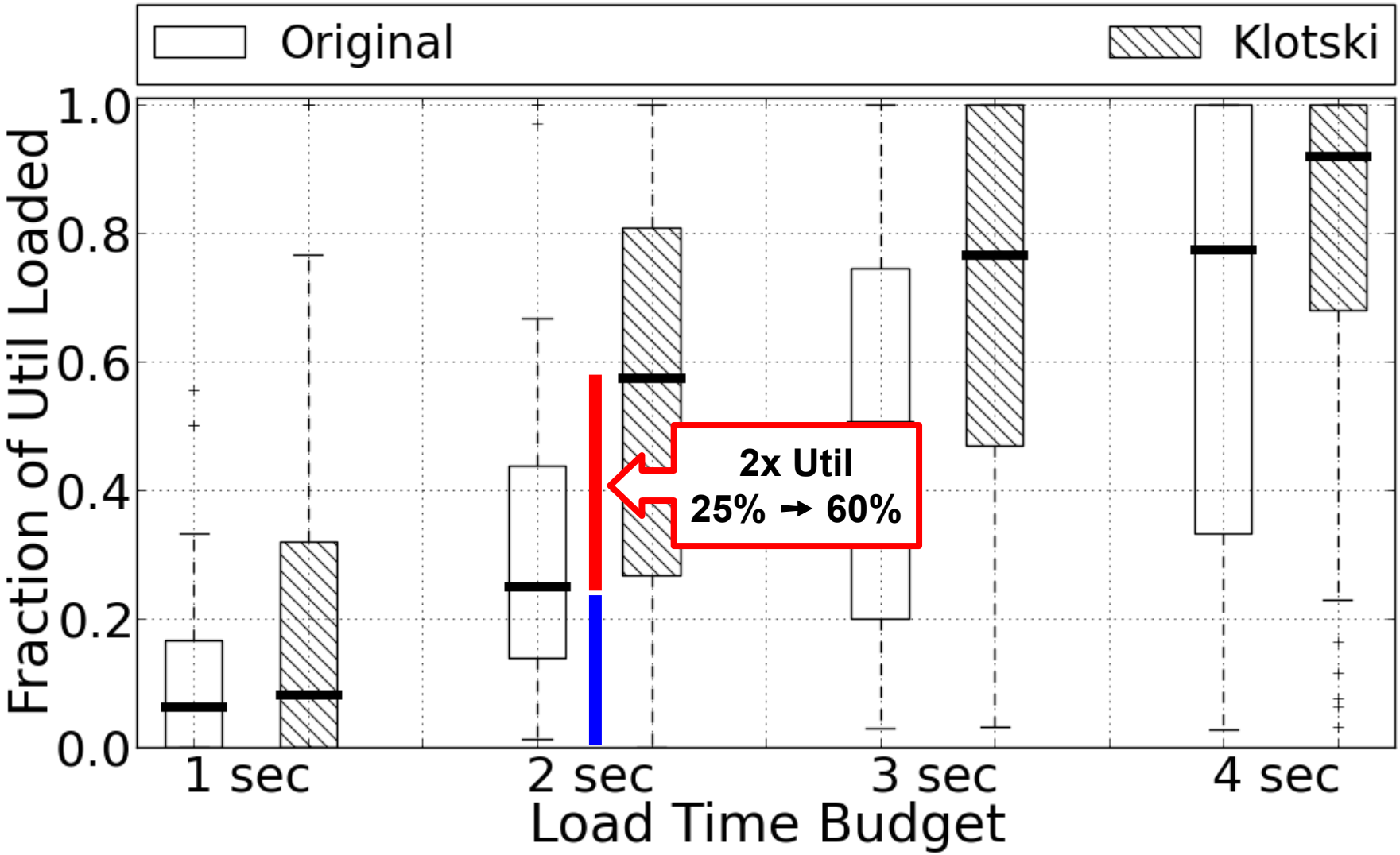
Klotski Improves User Experience



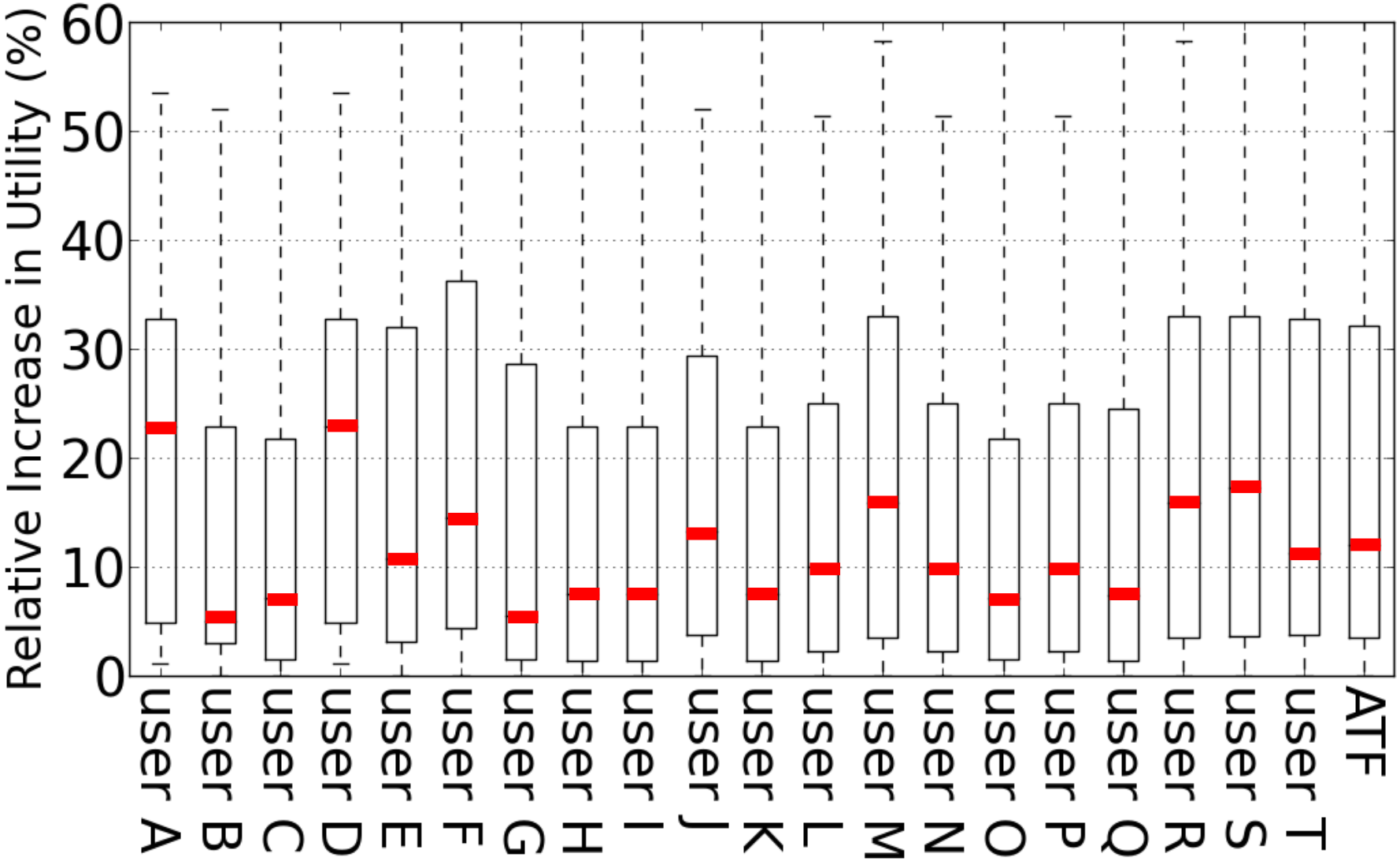
Klotski Improves User Experience



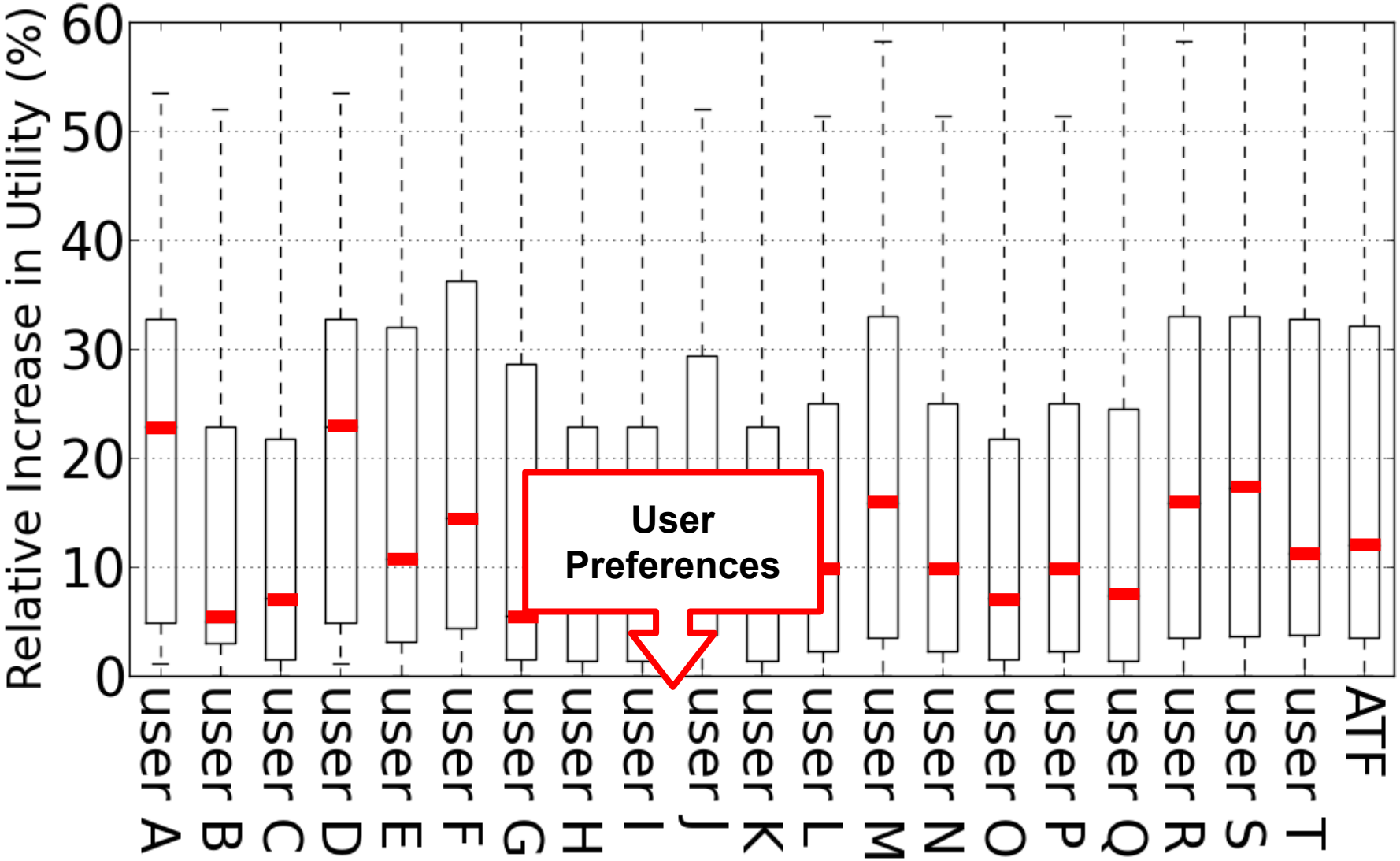
Klotski Improves User Experience



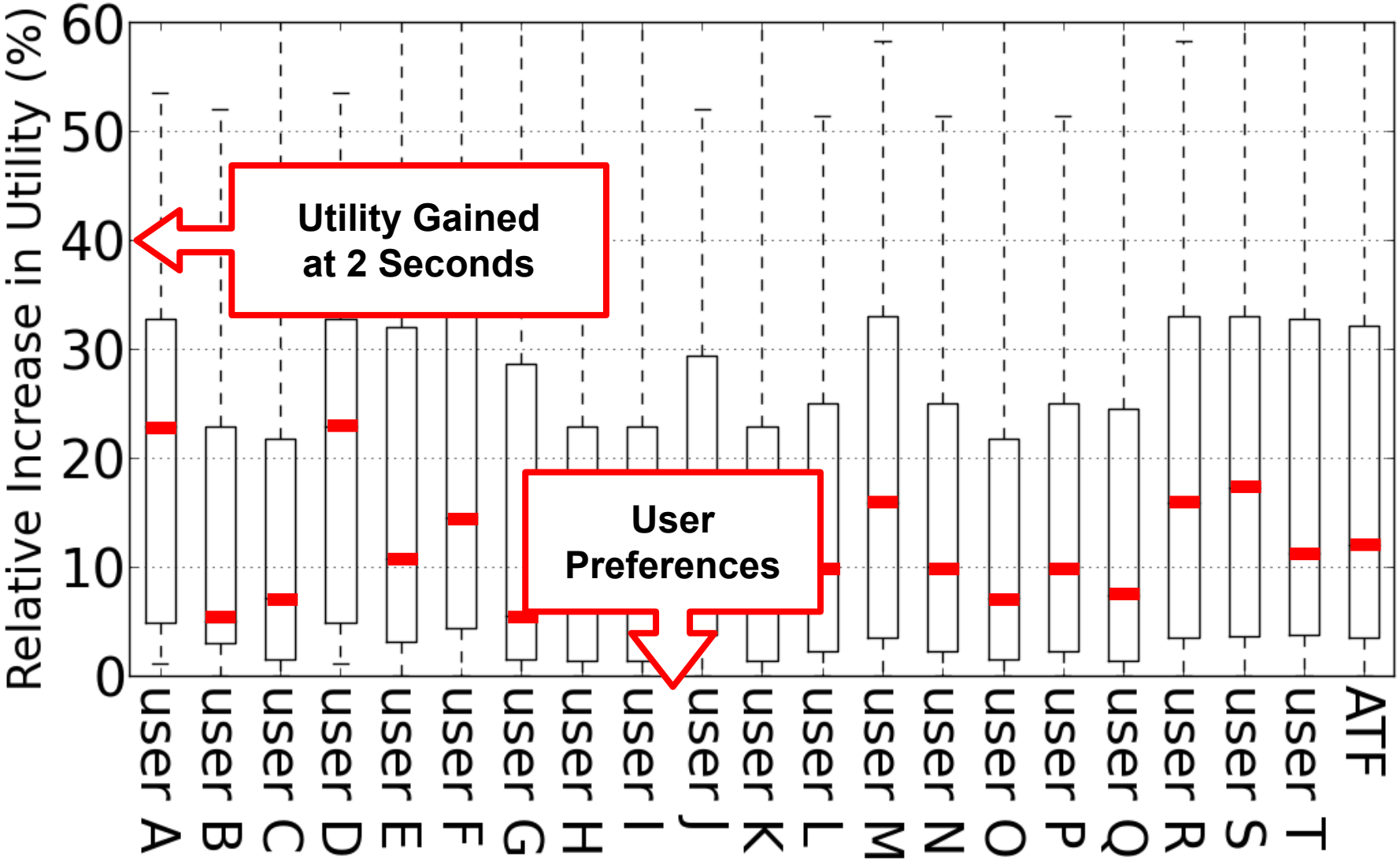
High Utility Gain For Diverse Users



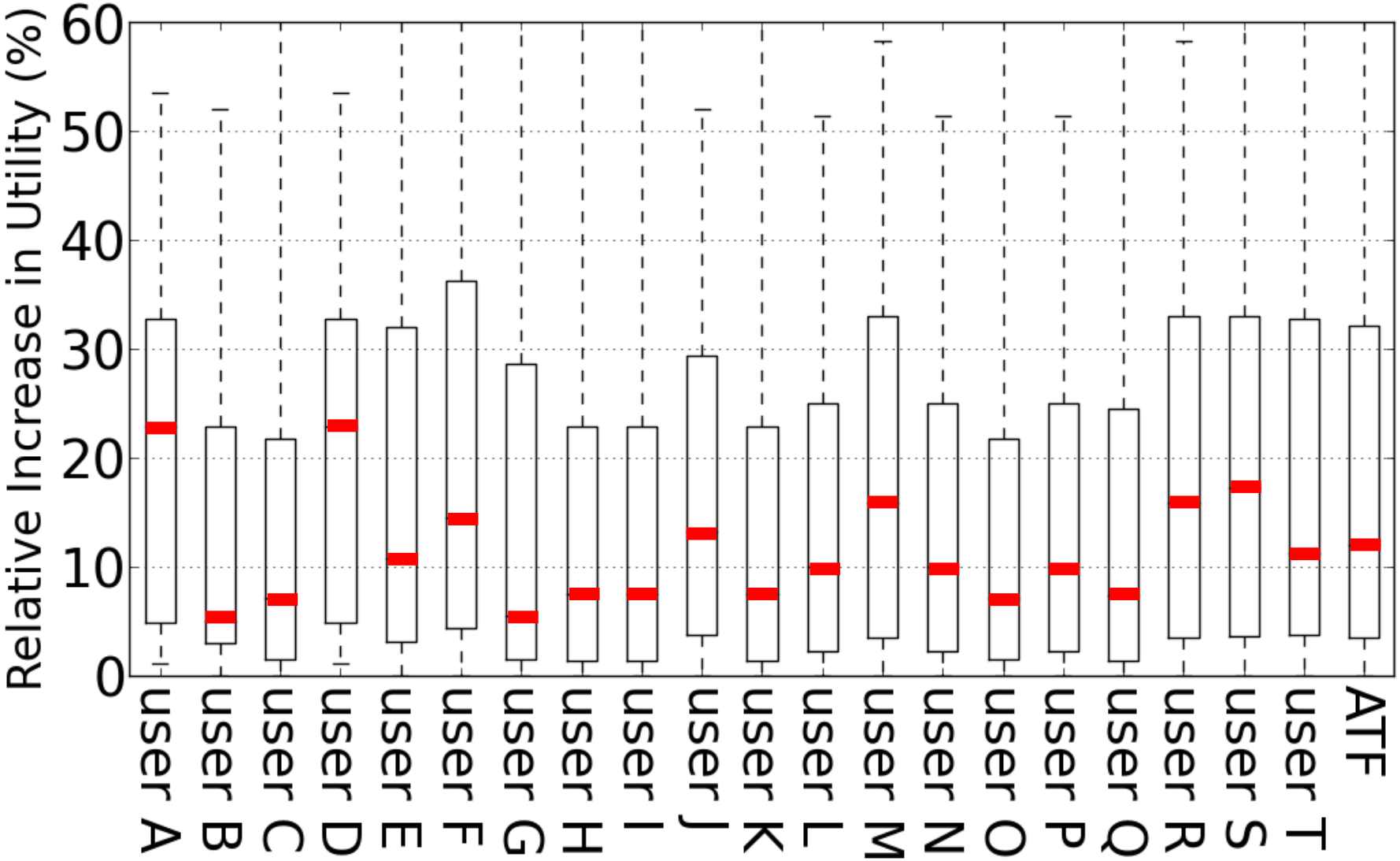
High Utility Gain For Diverse Users



High Utility Gain For Diverse Users



High Utility Gain For Diverse Users



Conclusions

- Mobile web continues to be a pain point
  - Focus on load time alone is likely insufficient
- Instead we focus on dynamic reprioritization
- Key challenges we address:
  - dynamic dependency representation
  - fast resource selection
  - load time estimation
- Klotski greatly improves user experience
  - for diverse preferences