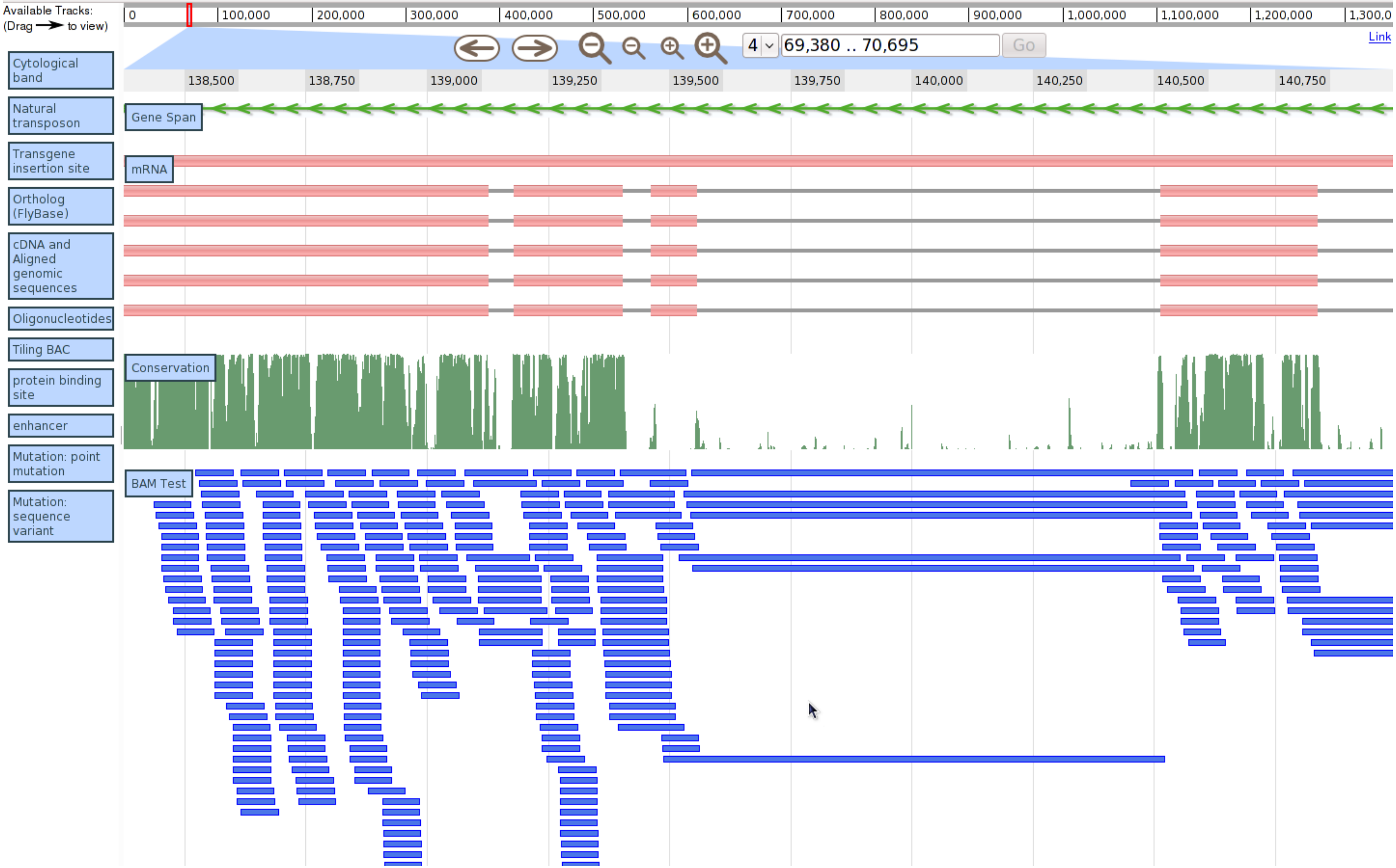
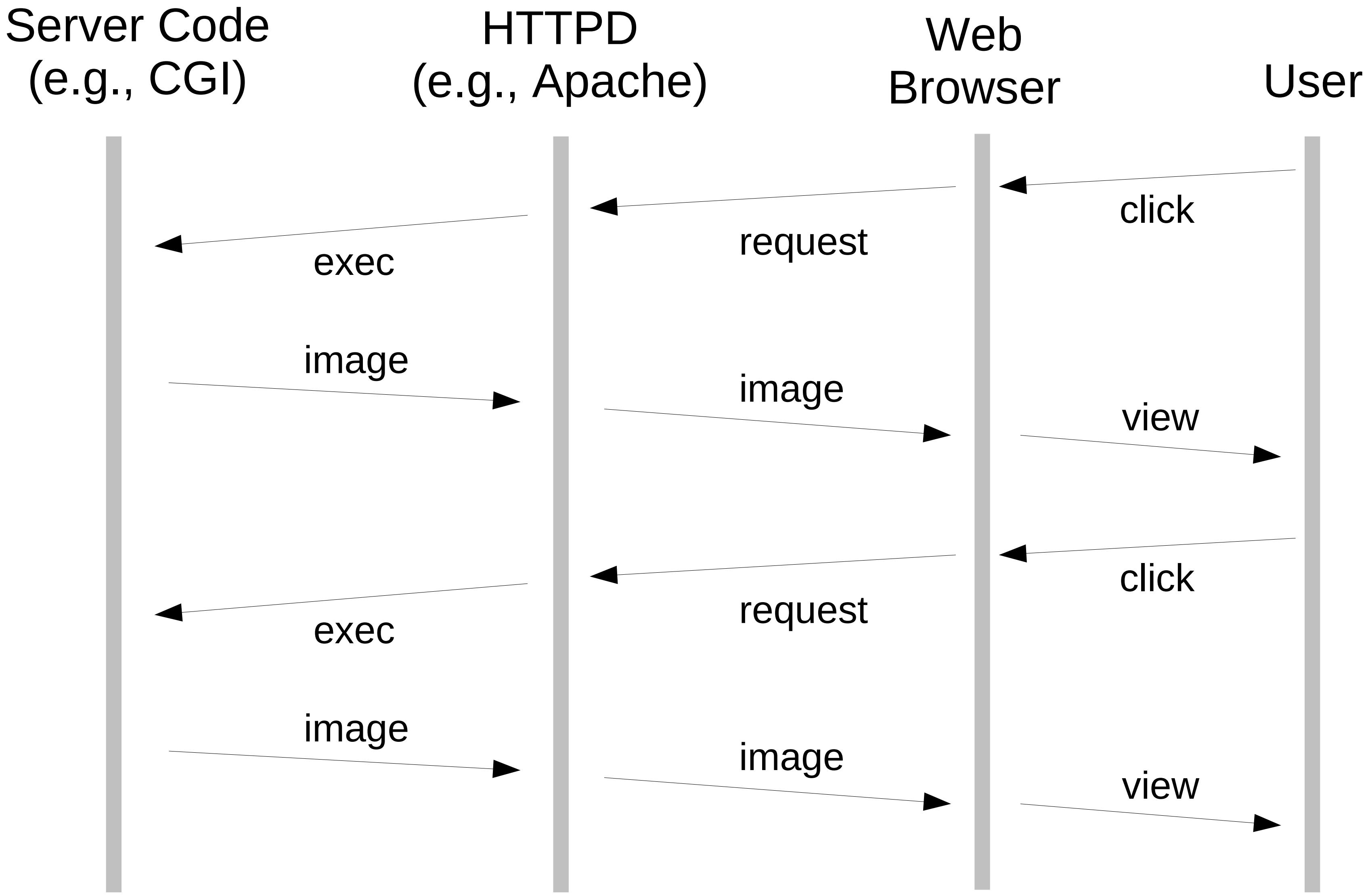
JBrowse

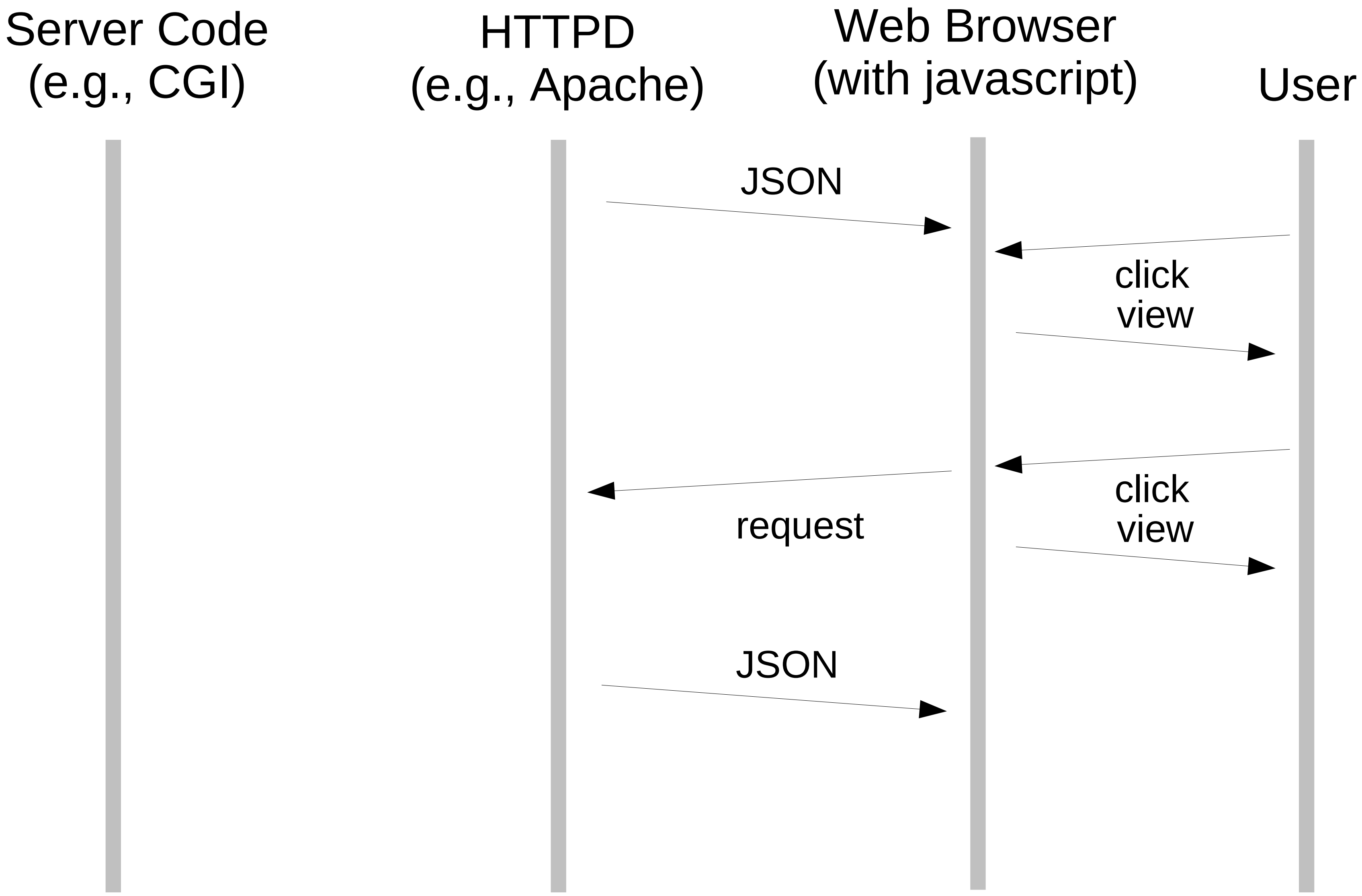

Available Tracks:
(Drag → to view)
Cytological band
Natural transposon
Transgene insertion site
Ortholog (FlyBase)
cDNA and Aligned genomic sequences
Oligonucleotides
Tiling BAC
protein binding site
enhancer
Mutation: point mutation
Mutation: sequence variant
0
100,000
200,000
300,000
400,000
500,000
600,000
700,000
800,000
900,000
1,000,000
1,100,000
1,200,000
1,300,0
4
69,380 .. 70,695
Go
Link
138,500
138,750
139,000
139,250
139,500
139,750
140,000
140,250
140,500
140,750
Gene Span
mRNA
Conservation
BAM Test

How JBrowse is different

Most Web-based Genome Browsers



JBrowse



Server Code
(e.g., CGI)

HTTPD
(e.g., Apache)

Web Browser
(with javascript)

User

JSON

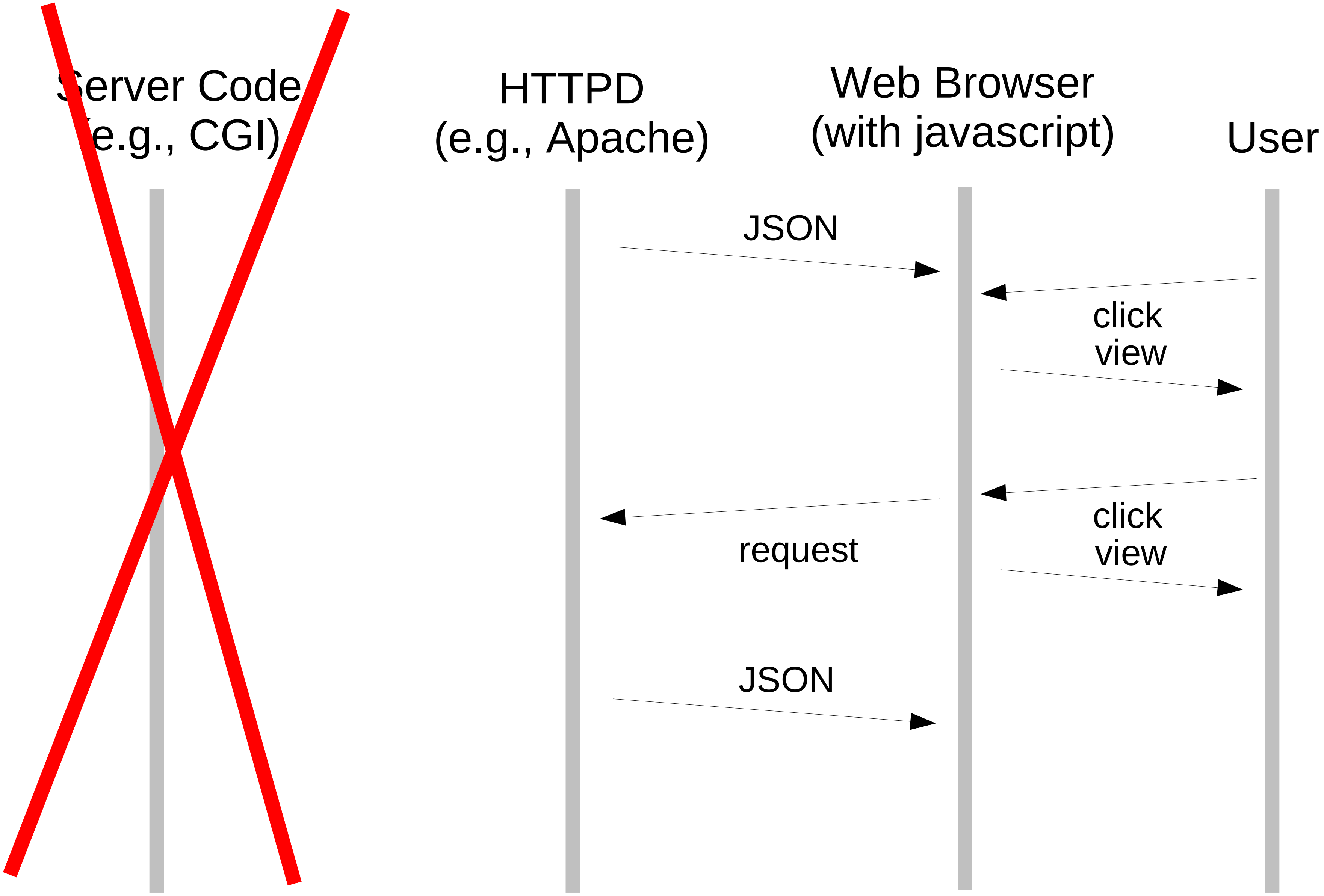
click
view

click
view

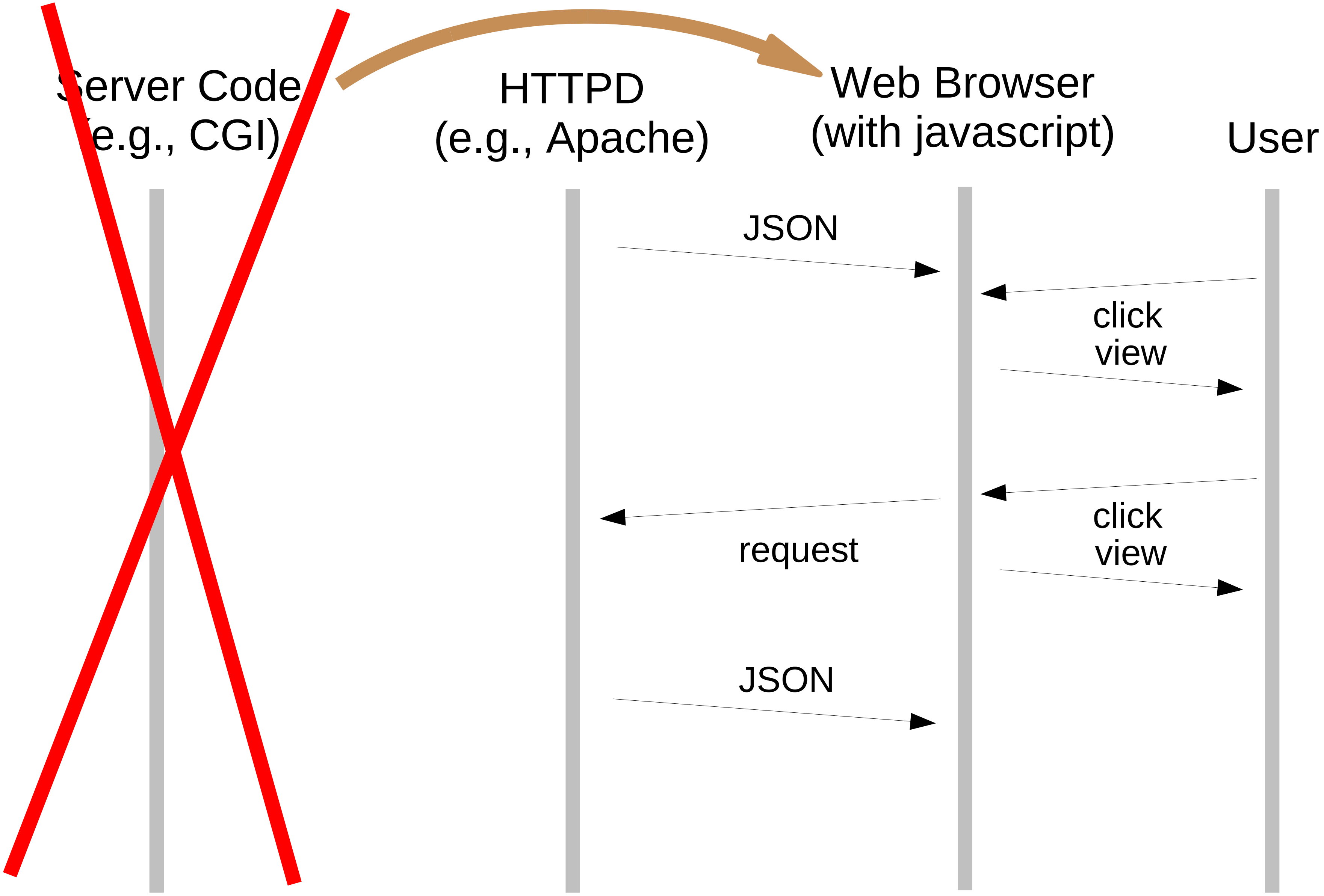
request

JSON

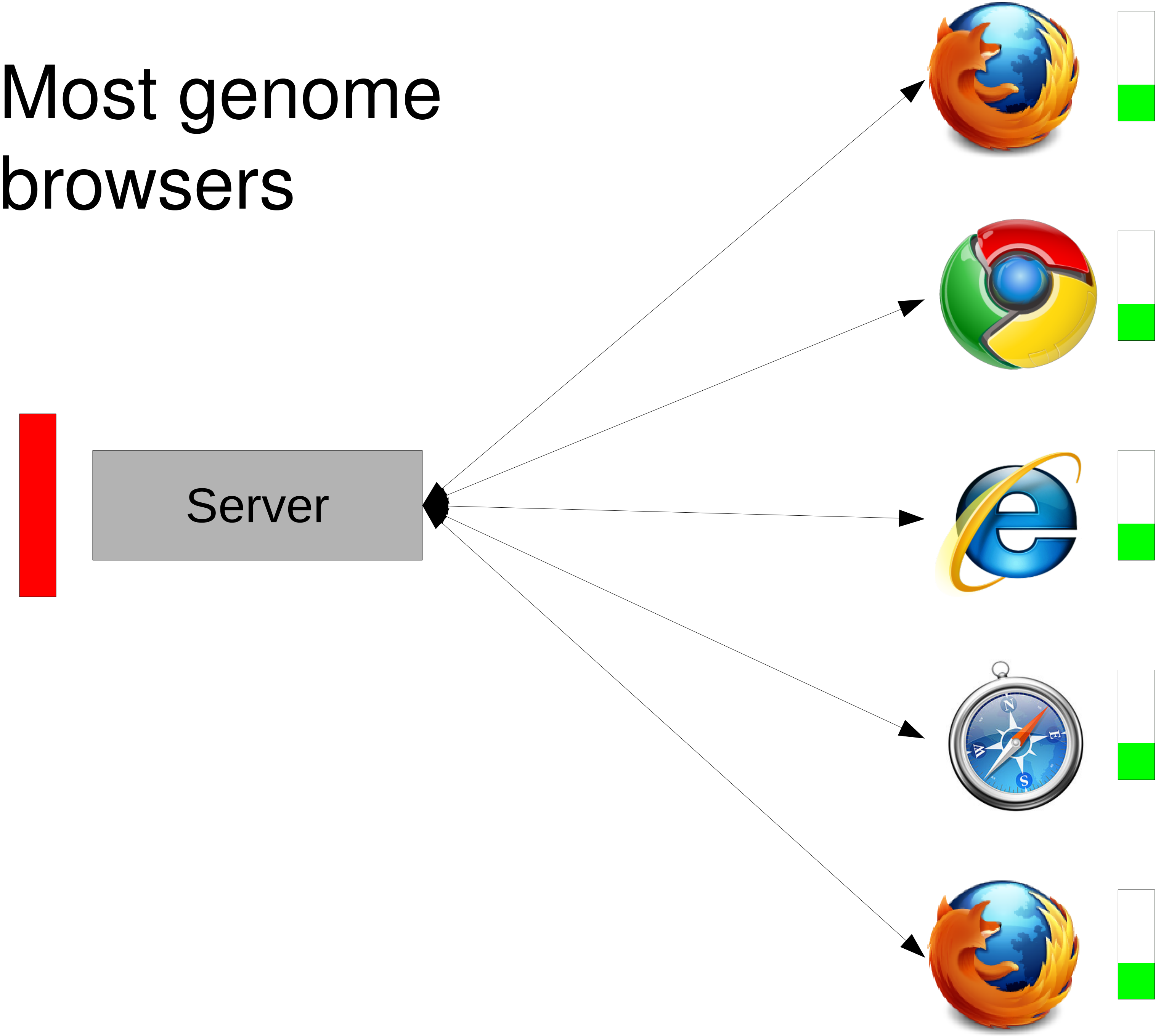
JBrowse


Server Code
(e.g., CGI)
HTTPD
(e.g., Apache)
Web Browser
(with javascript)
User
JSON
click
view
request
click
view
JSON

JBrowse

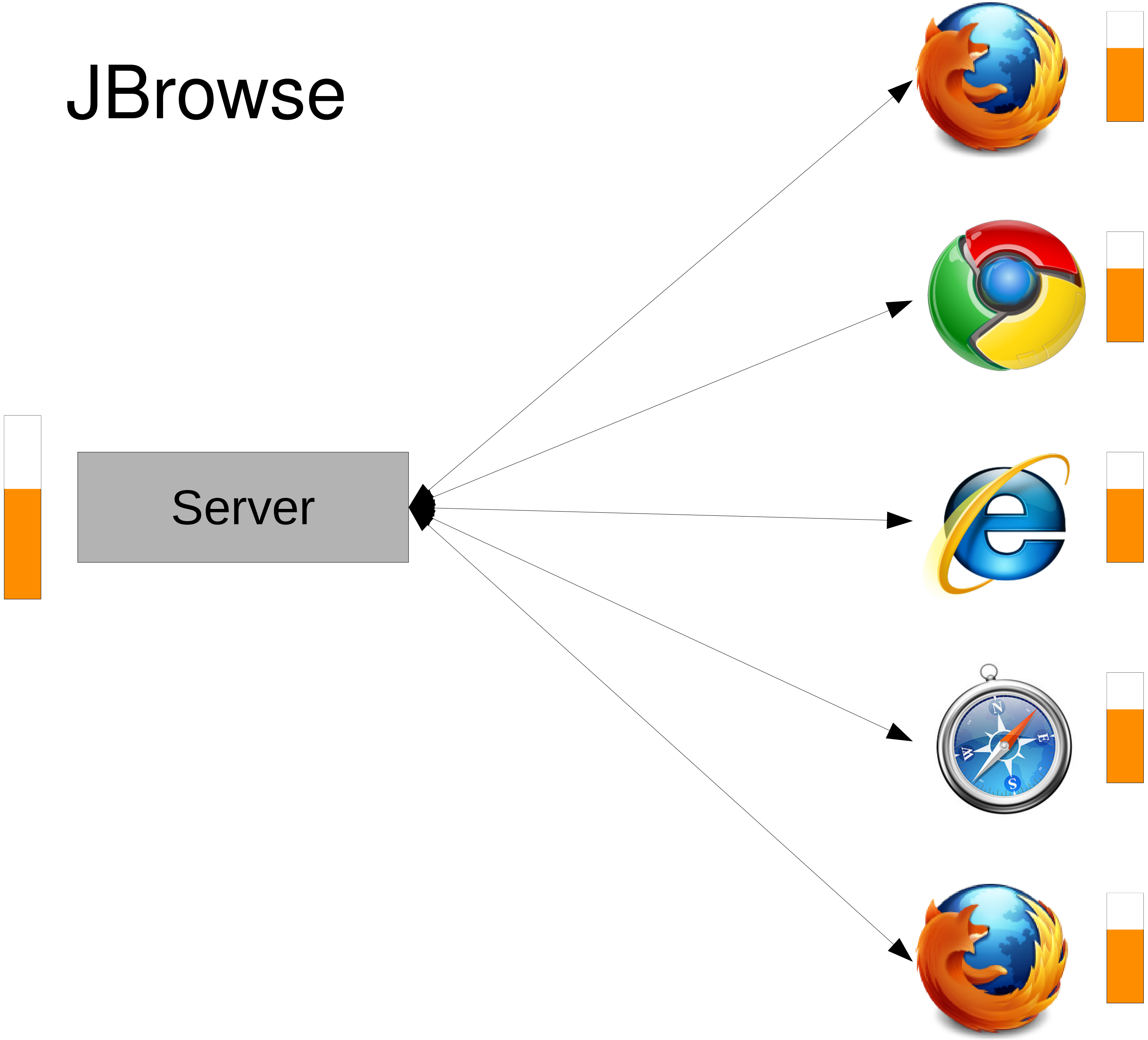

Server Code
(e.g., CGI)
HTTPD
(e.g., Apache)
Web Browser
(with javascript)
User
JSON
click
view
click
view
request
JSON

Most genome browsers



Server

JBrowse

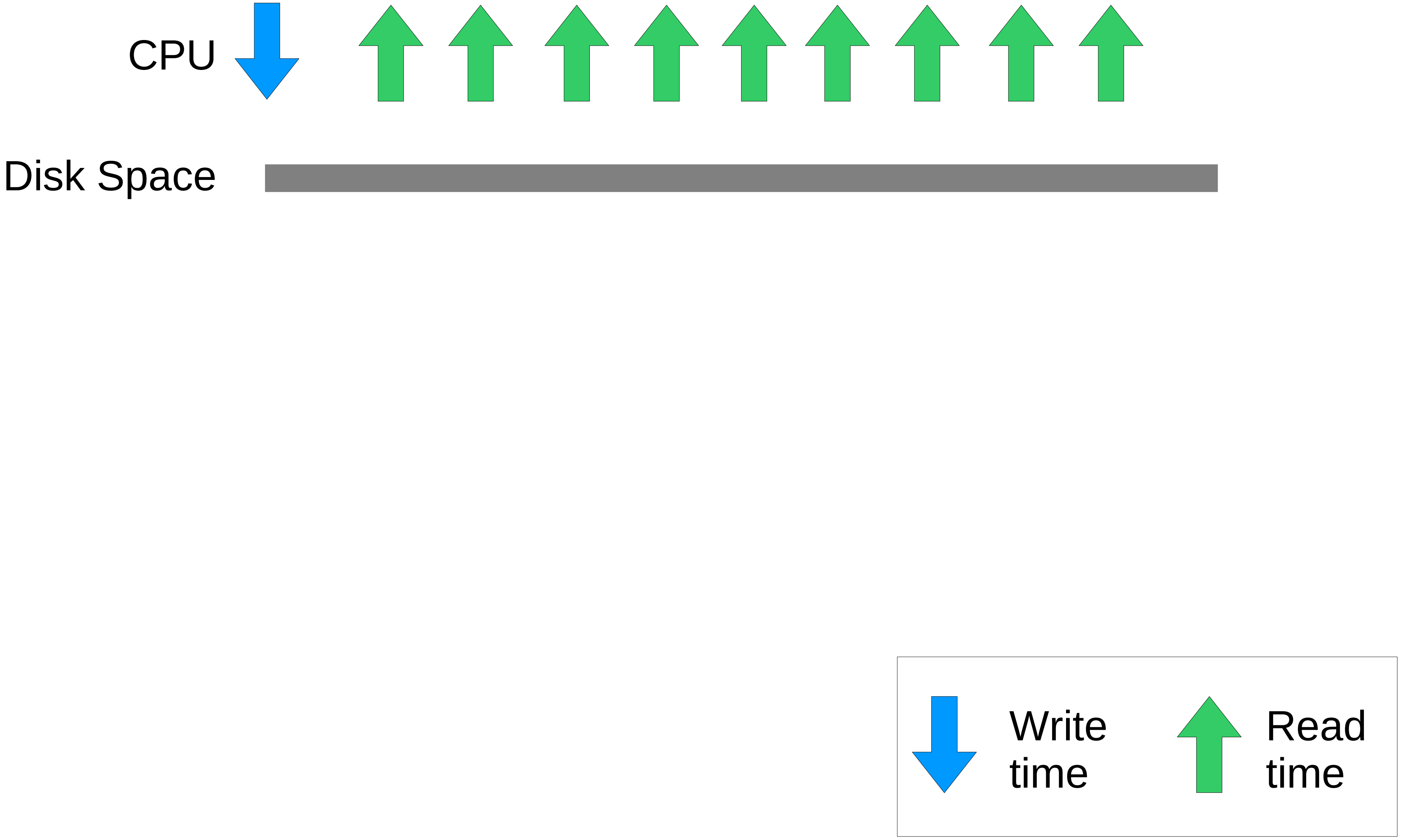


JBrowse moves work:

Server -> client

On the server:
Read-time -> write-time

Most Web-based Genome Browsers



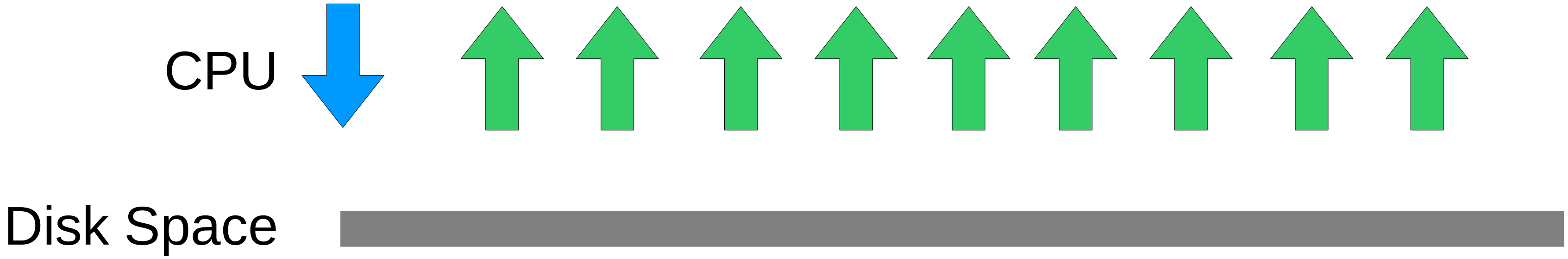
CPU

Disk Space

Write time

Read time

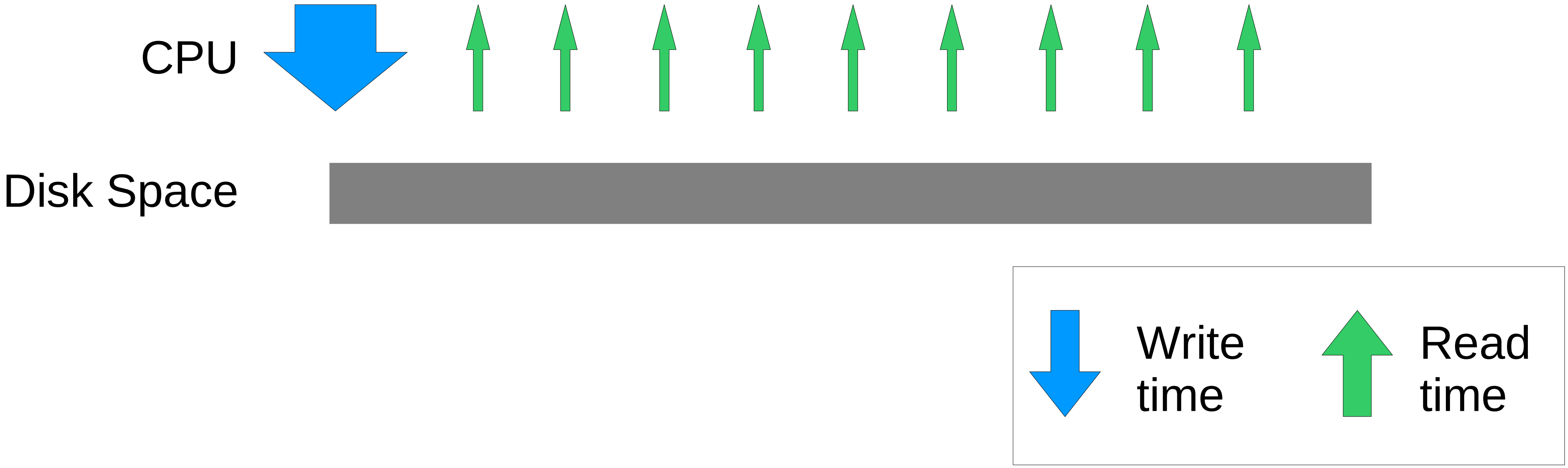
Most Web-based Genome Browsers



CPU

Disk Space

JBrowse



CPU

Disk Space

Write time

Read time

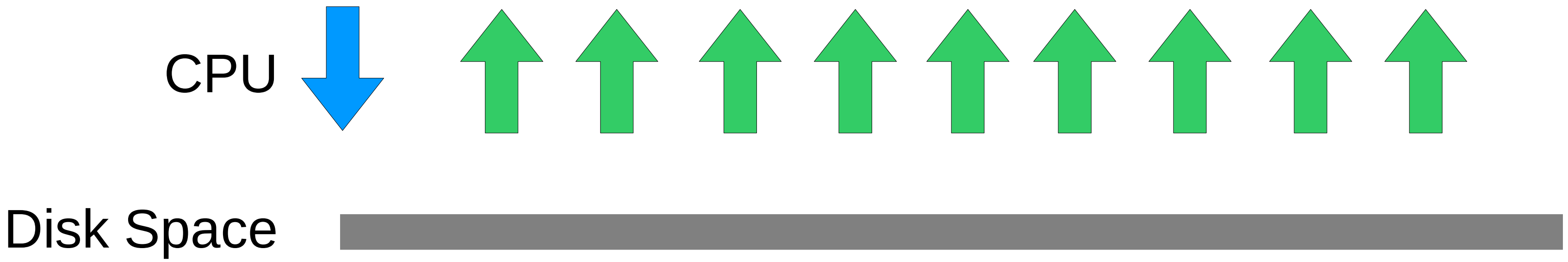
BAM example

- On one test data set:
  - 4.4 million features
  - 8 minutes to process
    - From 242 megabyte BAM file
    - Not paired-end
  - Used 400 megabytes of RAM
  - 330 megabytes on disk (without sequence)
    - Broken into ~40 kilobyte chunks
  - Compresses down to 80 megabytes

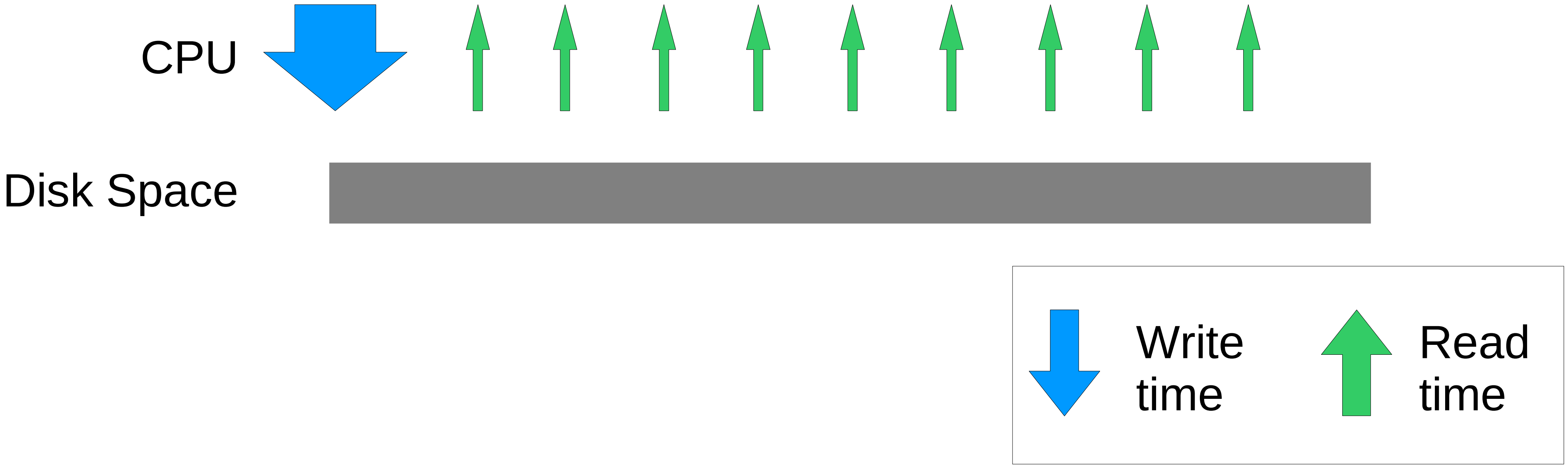
Wiggle tracks: pre-rendered

- Only rendered up to 1 base per pixel
- Implemented in C++
- ~12 min to generate tiles for Dmel conservation track (1 data point per base)
    - => ~1min per 10 million bases
- Wiggle tiles compress well
    - ~5 bytes/base, half of which is filesystem overhead
- They could also be rendered on the fly

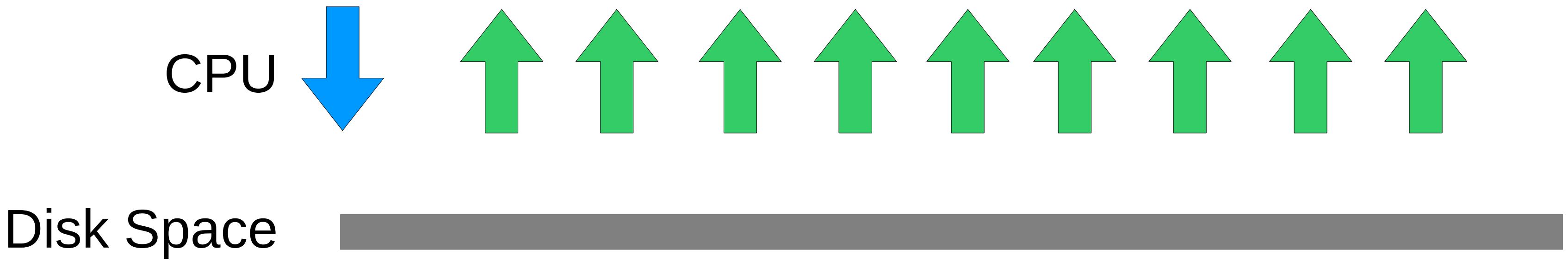
Most Web-based Genome Browsers



JBrowse



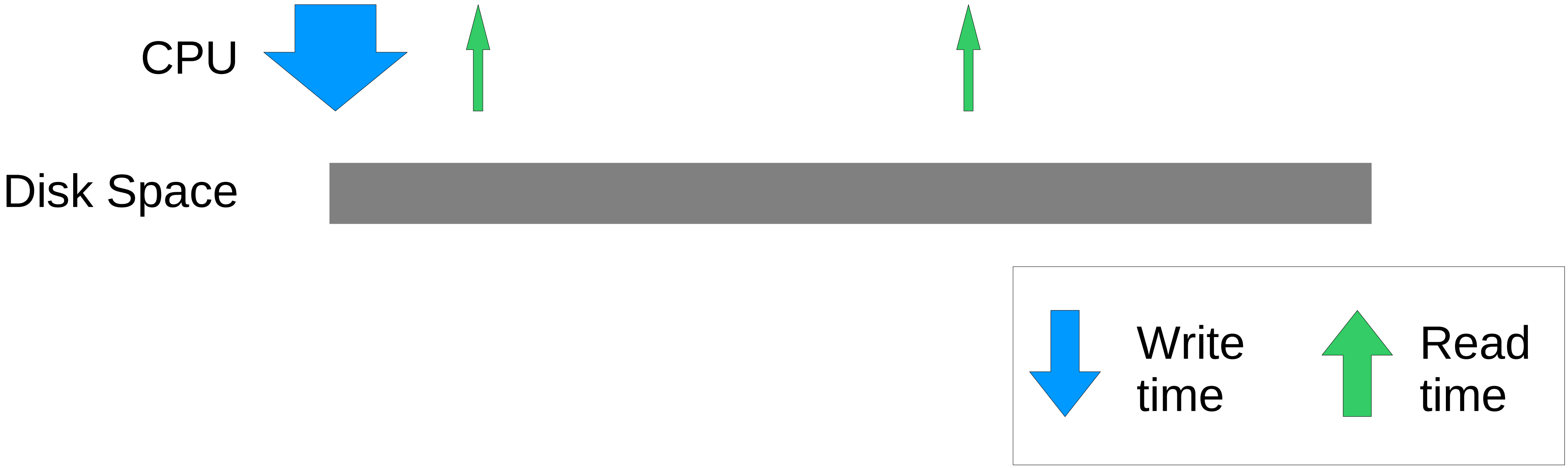
Most Web-based Genome Browsers



CPU

Disk Space

JBrowse



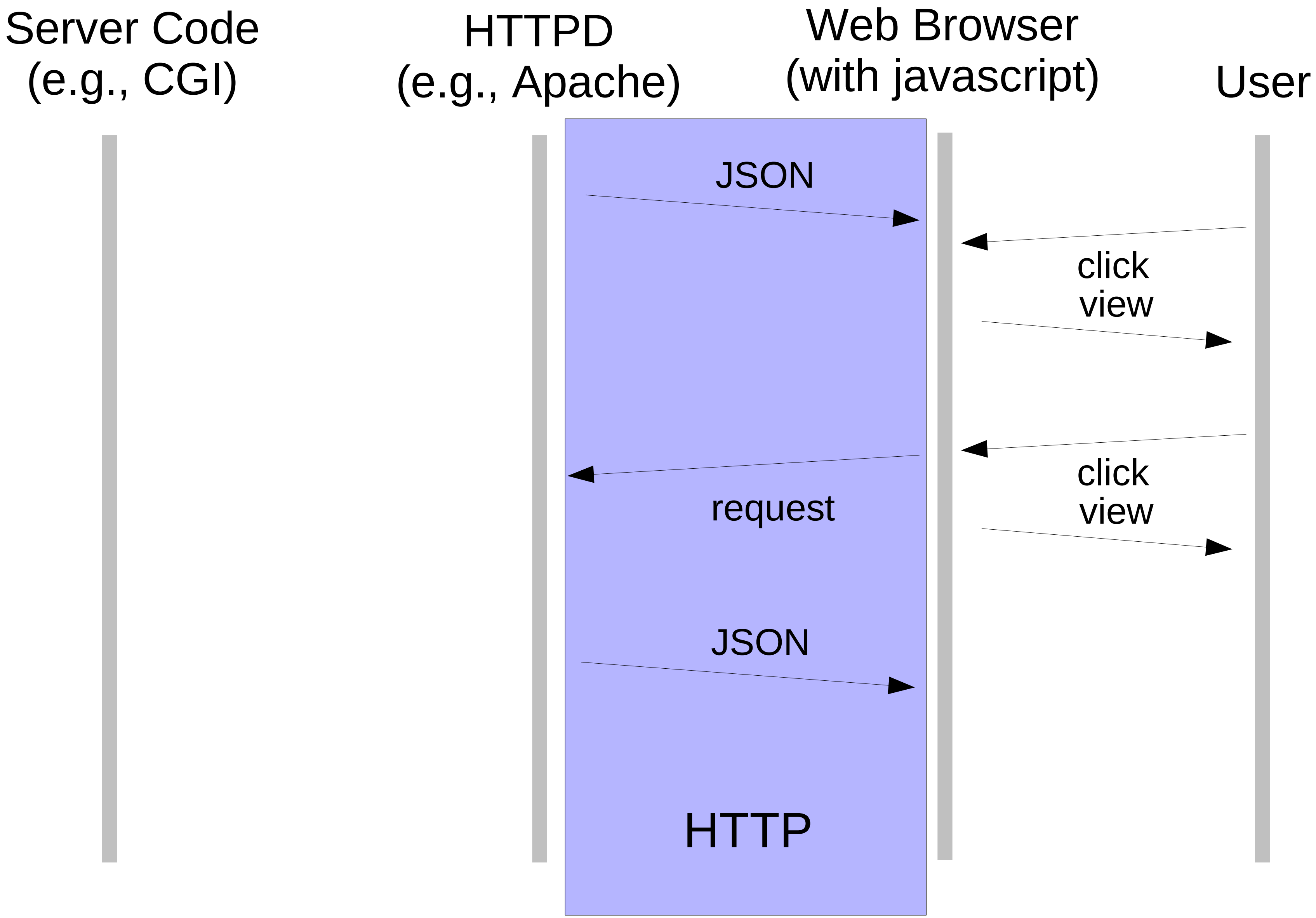
CPU

Disk Space

Write time

Read time

JBrowse



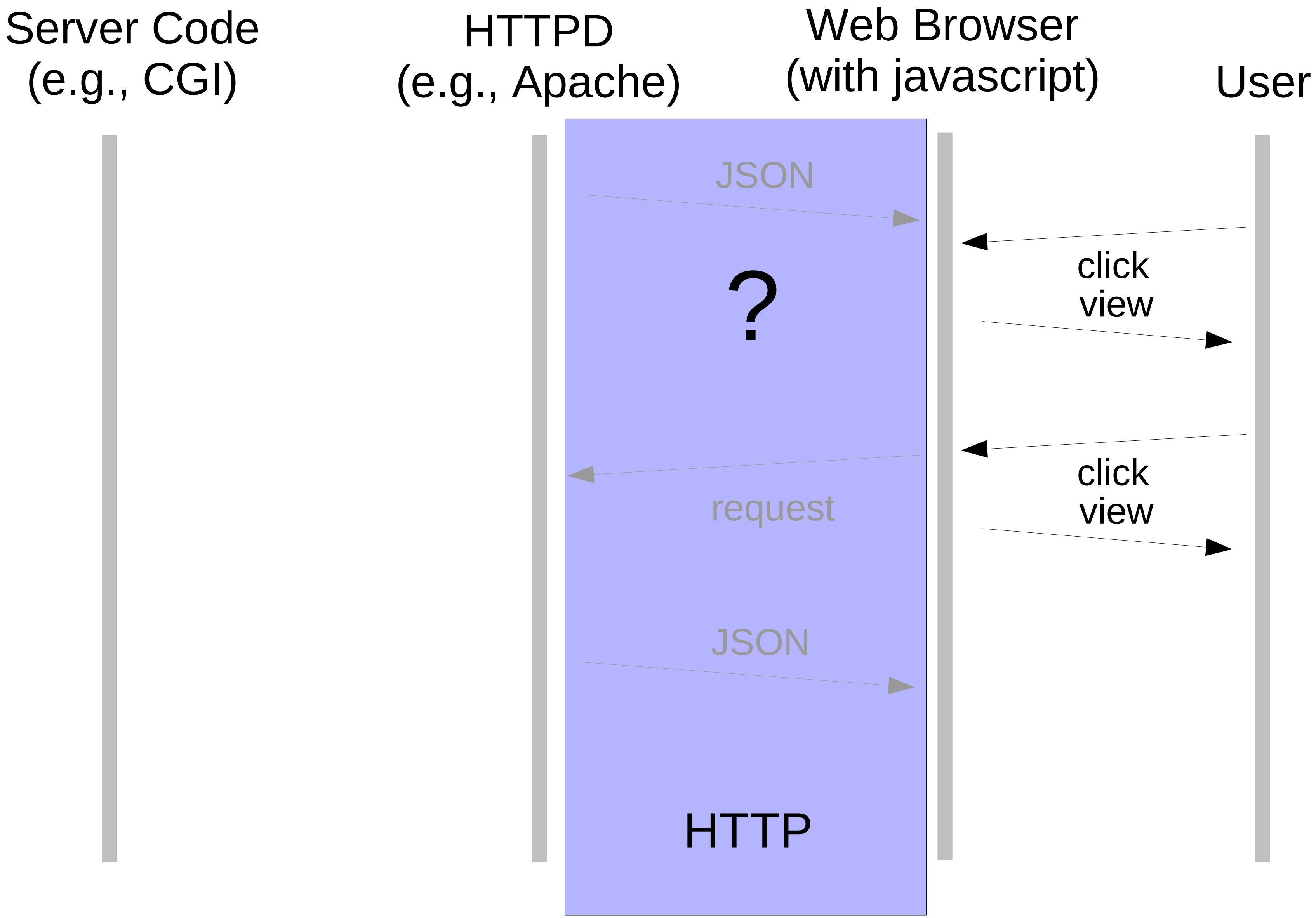
Server Code
(e.g., CGI)

HTTPD
(e.g., Apache)

Web Browser
(with javascript)

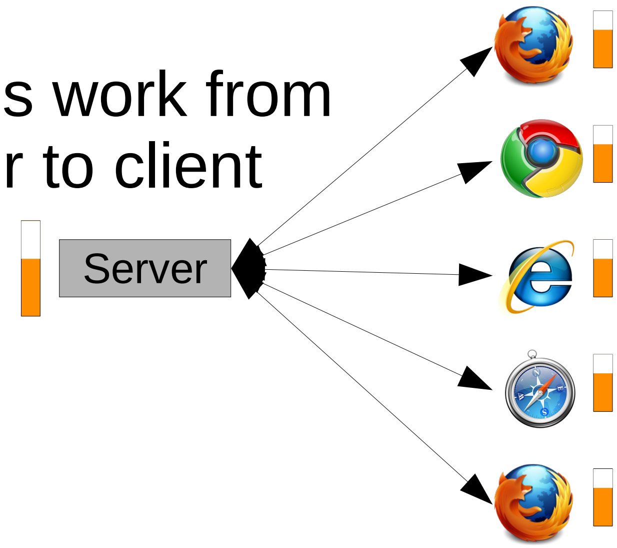
User

JBrowse

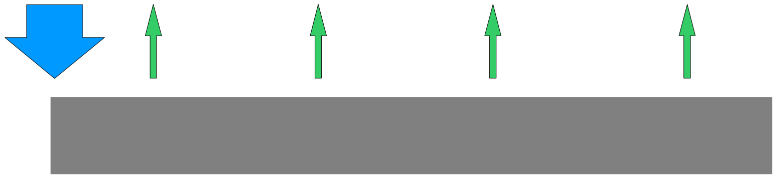

Server Code
(e.g., CGI)
HTTPD
(e.g., Apache)
Web Browser
(with javascript)
User
JSON
?
click
view
request
click
view
JSON
HTTP

Summary: JBrowse...

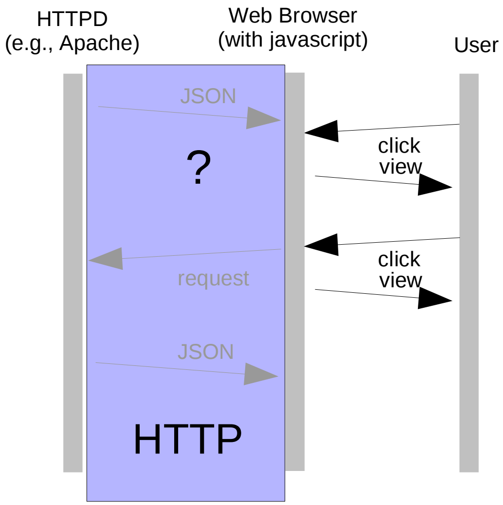
Moves work from server to client



Moves work from read-time to write-time



Allows the client to cache useful amounts of data



Lower server load means:

- The user waits much less
- Cheaper/easier to host a genome browser

Client-side approach

- Richer interaction
  - Smooth, continuous transitions
    - Help the user build an intuitive sense of where things are relative to one another
  - Client-rendered graphics: client can filter, highlight, etc.
- Web Apollo

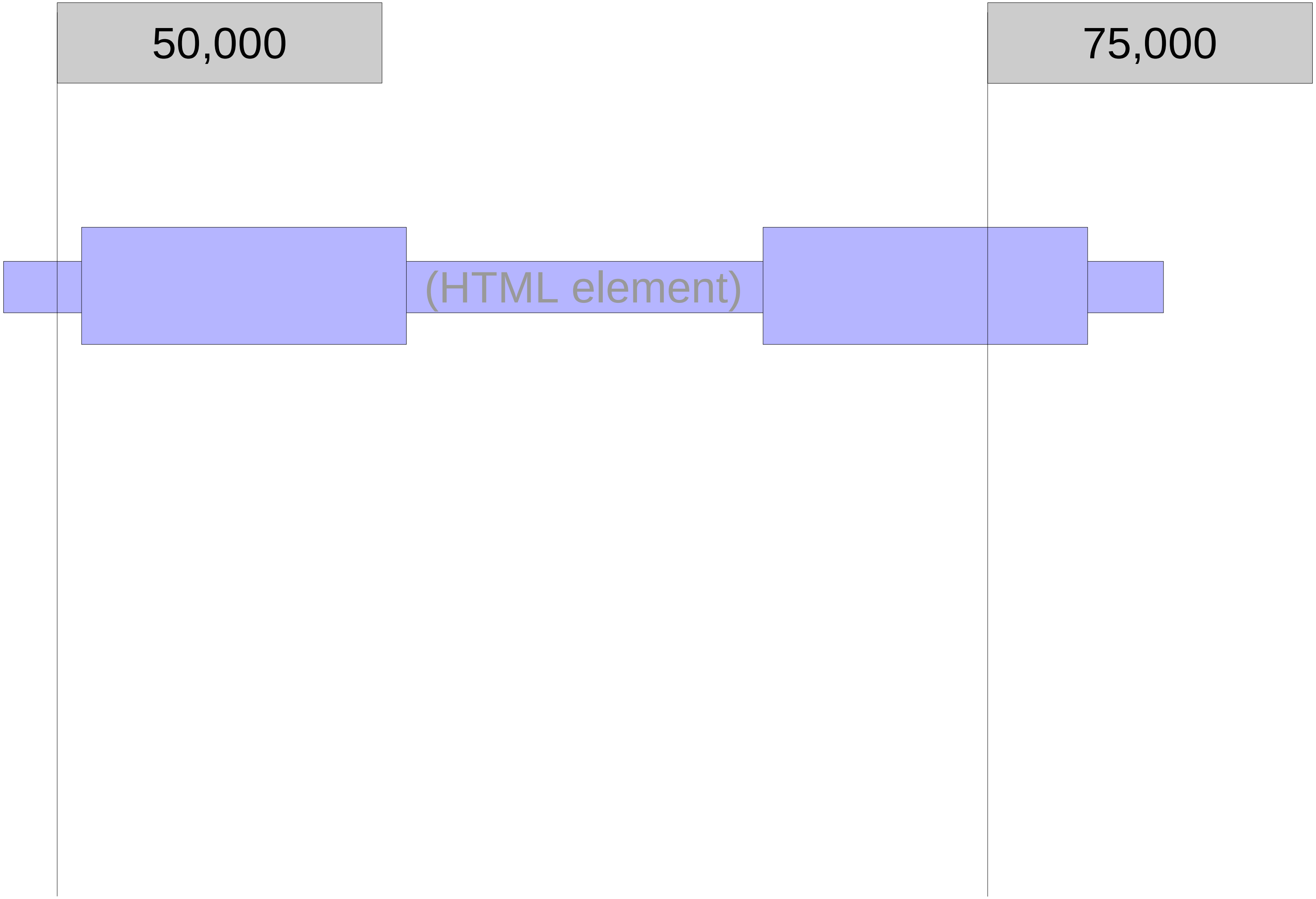
Why it wasn't done earlier

- Getting the web browser to do the work is non-trivial
  - Some web browsers have mechanisms intended to enable the browser to render graphics (e.g., SVG, canvas)
  - None of those mechanisms work in all browsers (Internet explorer doesn't have SVG or canvas)

HTML element

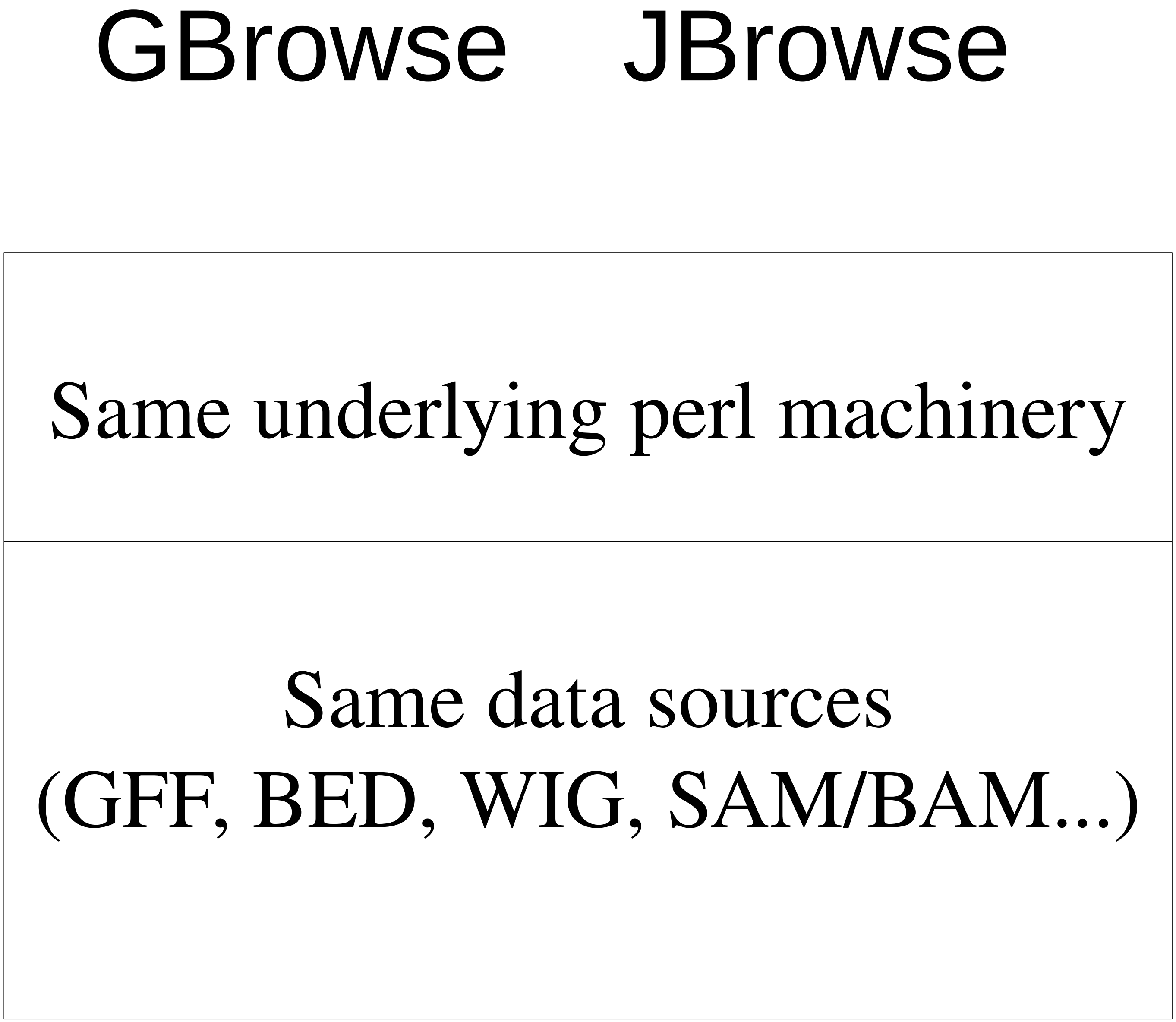
HTML element

HTML element


50,000
75,000
(HTML element)

GBrowse JBrowse

Older (2002)	Newer (2009)
More functionality	Faster, smoother UI
Does work on server	Moves work to web browser



GBrowse JBrowse

Same underlying perl machinery

Same data sources
(GFF, BED, WIG, SAM/BAM...)