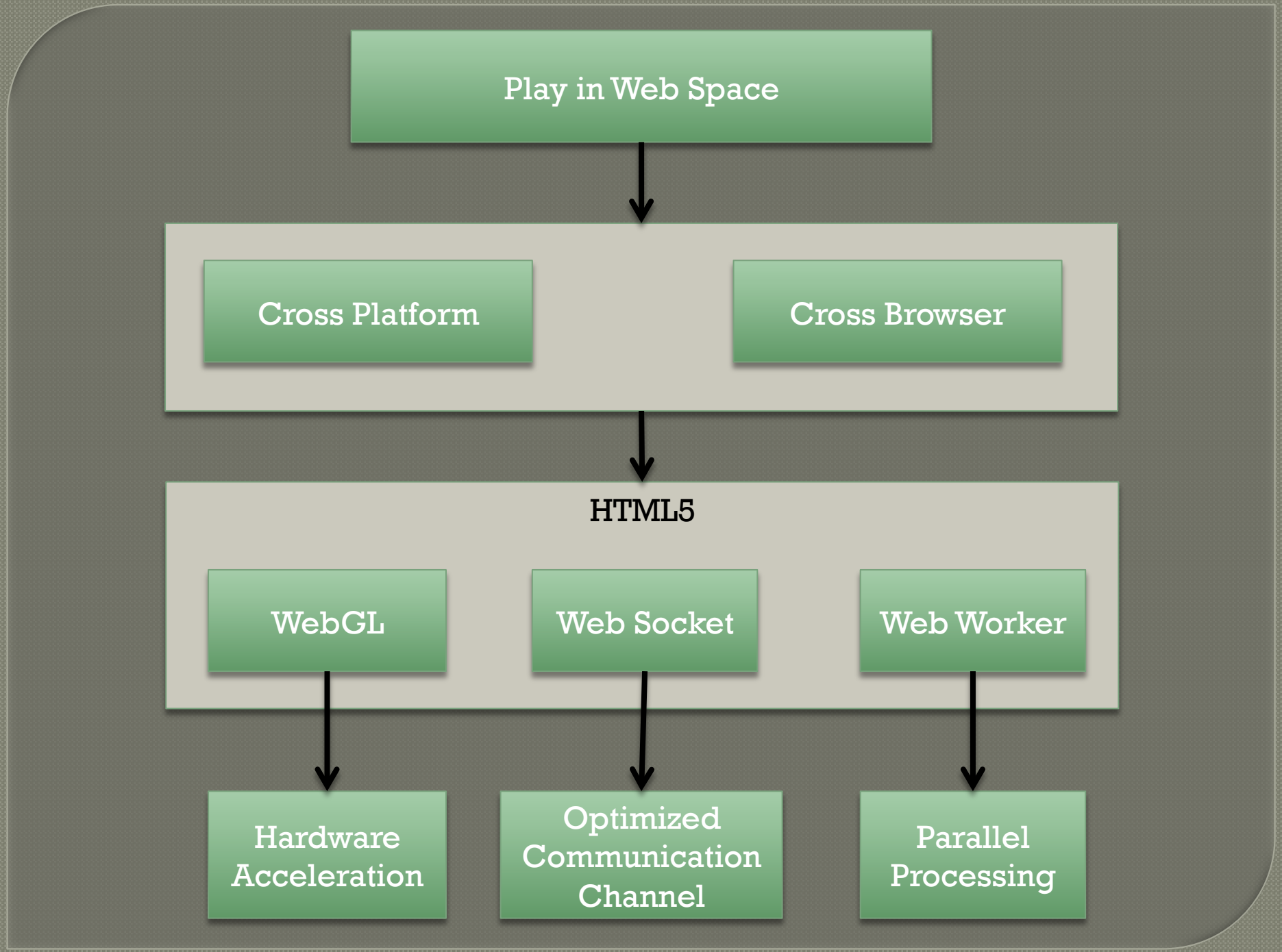
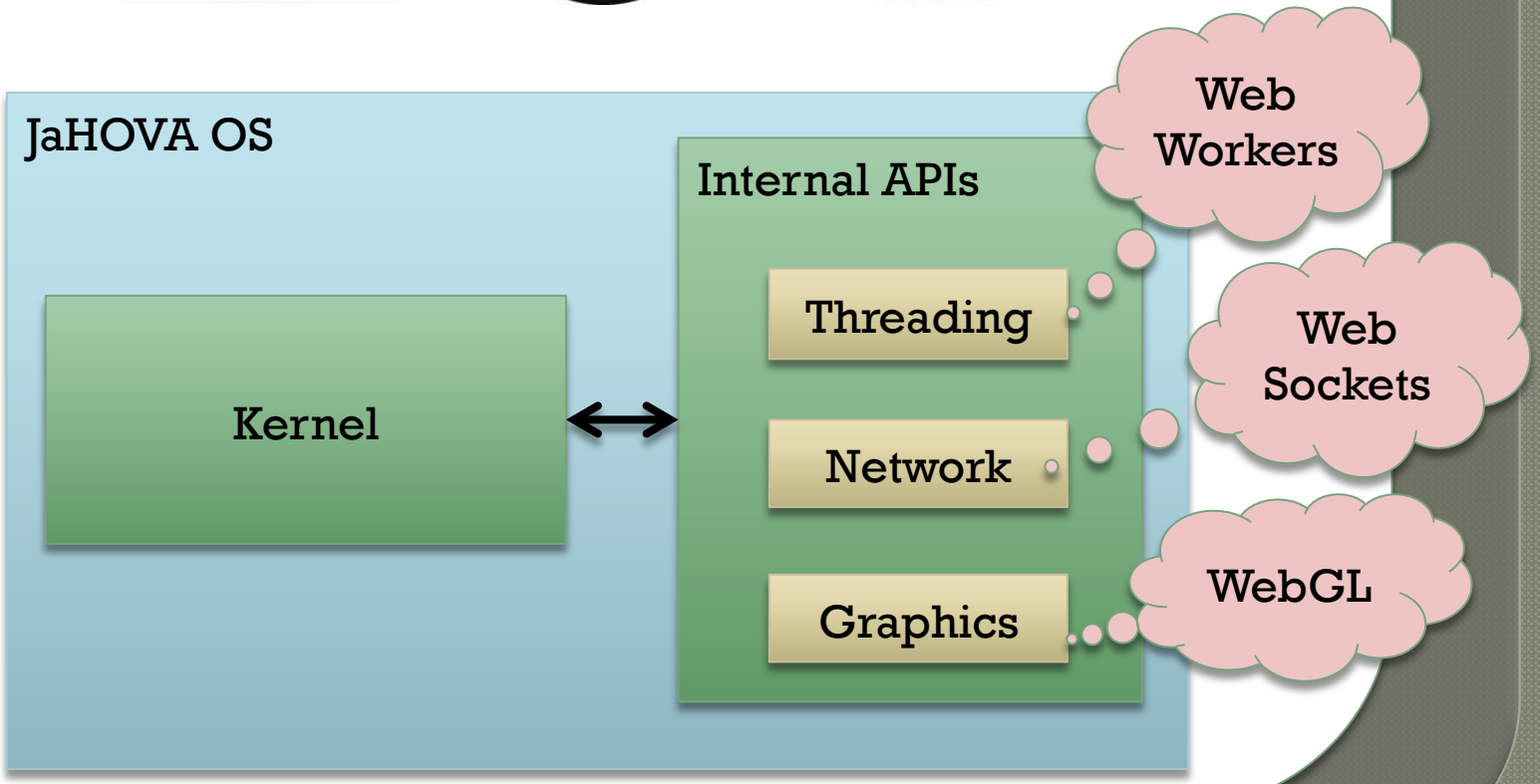
Building an HTML 5 Multithreaded Game Engine

Corey Clark Ph.D.
Derek Carpenter


Play in Web Space
Cross Platform
Cross Browser
HTML5
WebGL
Web Socket
Web Worker
Hardware Acceleration
Optimized Communication Channel
Parallel Processing

Browsers

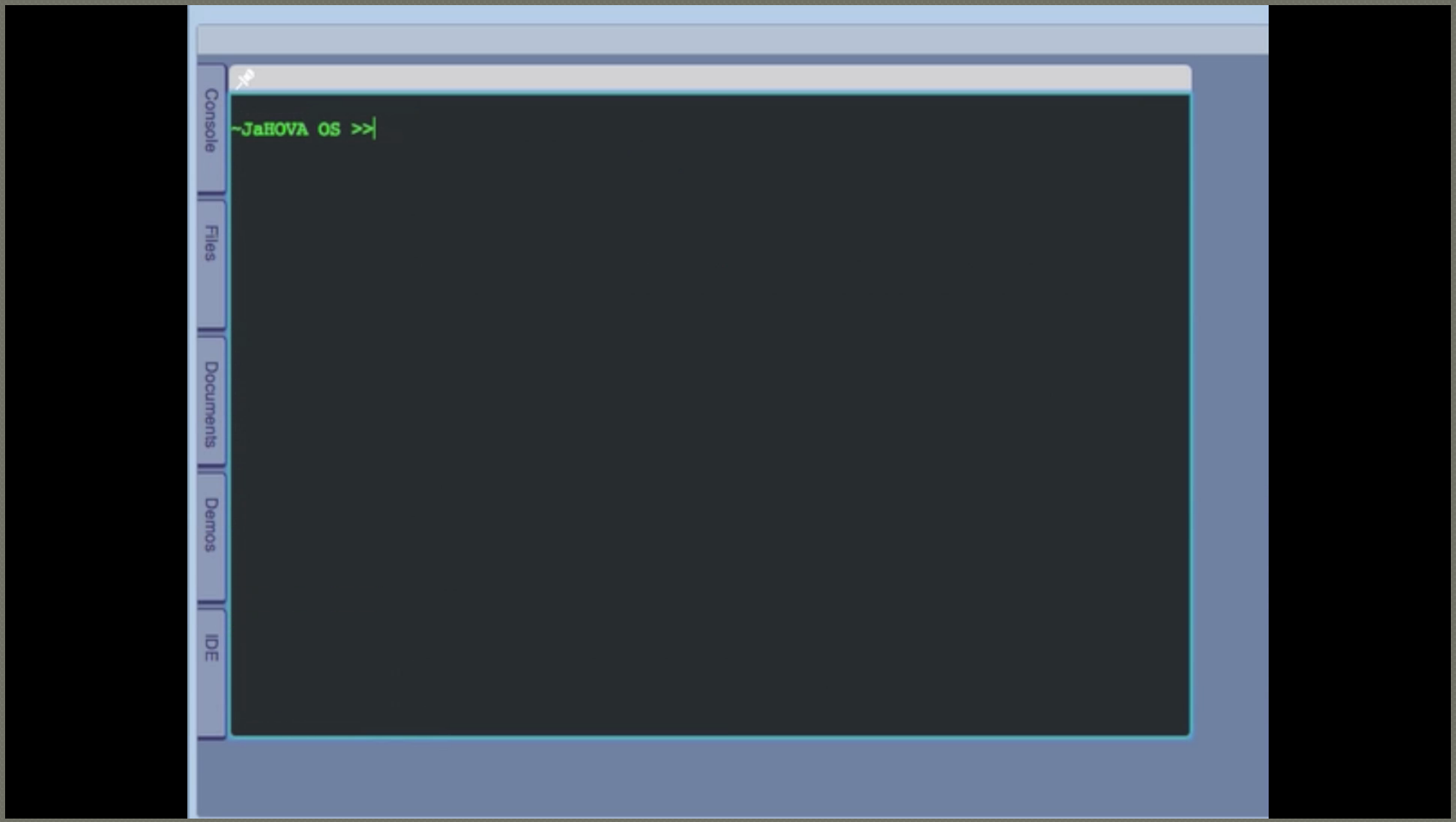

JaHOVA OS
Kernel
Internal APIs
Threading
Network
Graphics
Web Workers
Web Sockets
WebGL

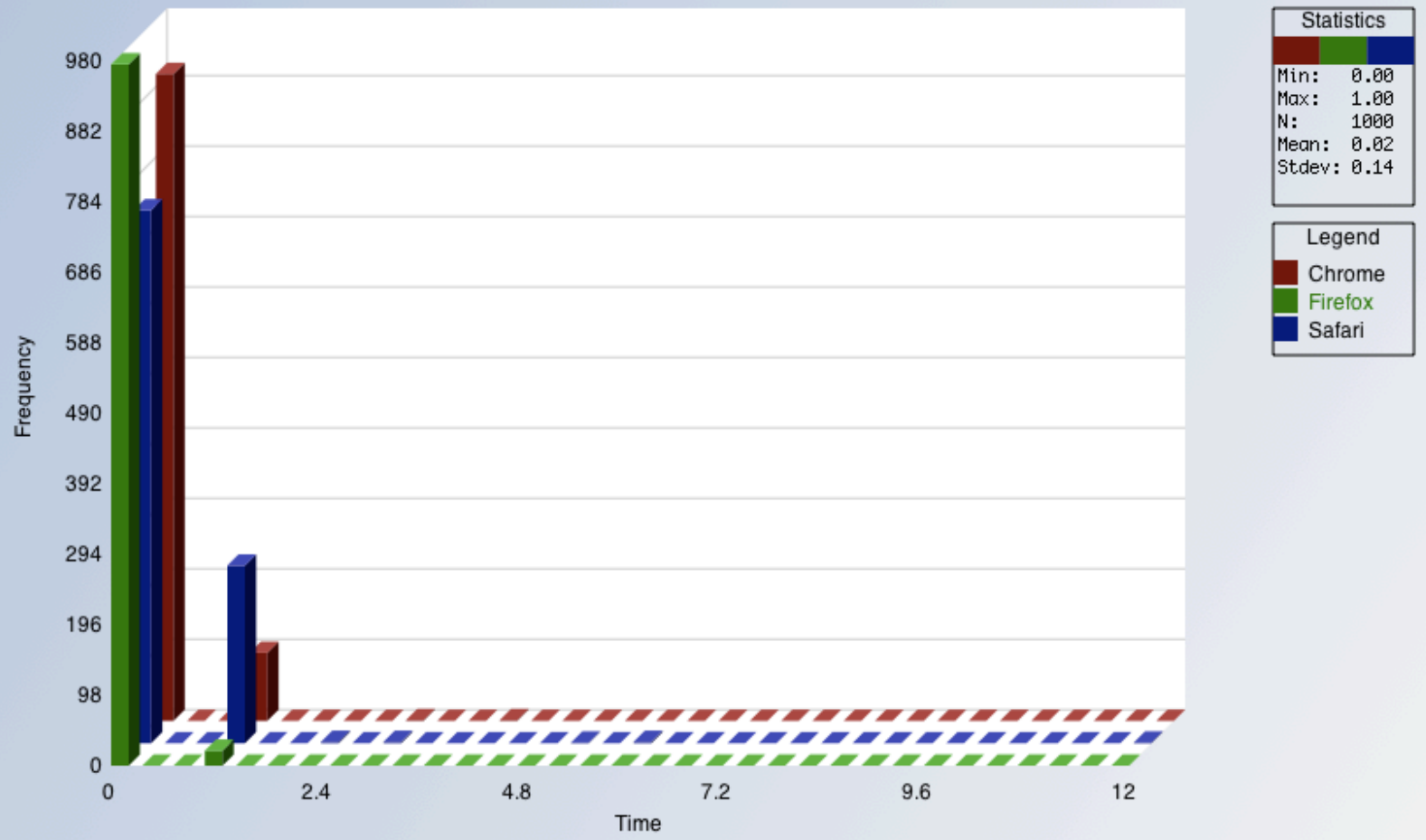
Web Workers and Multithreading

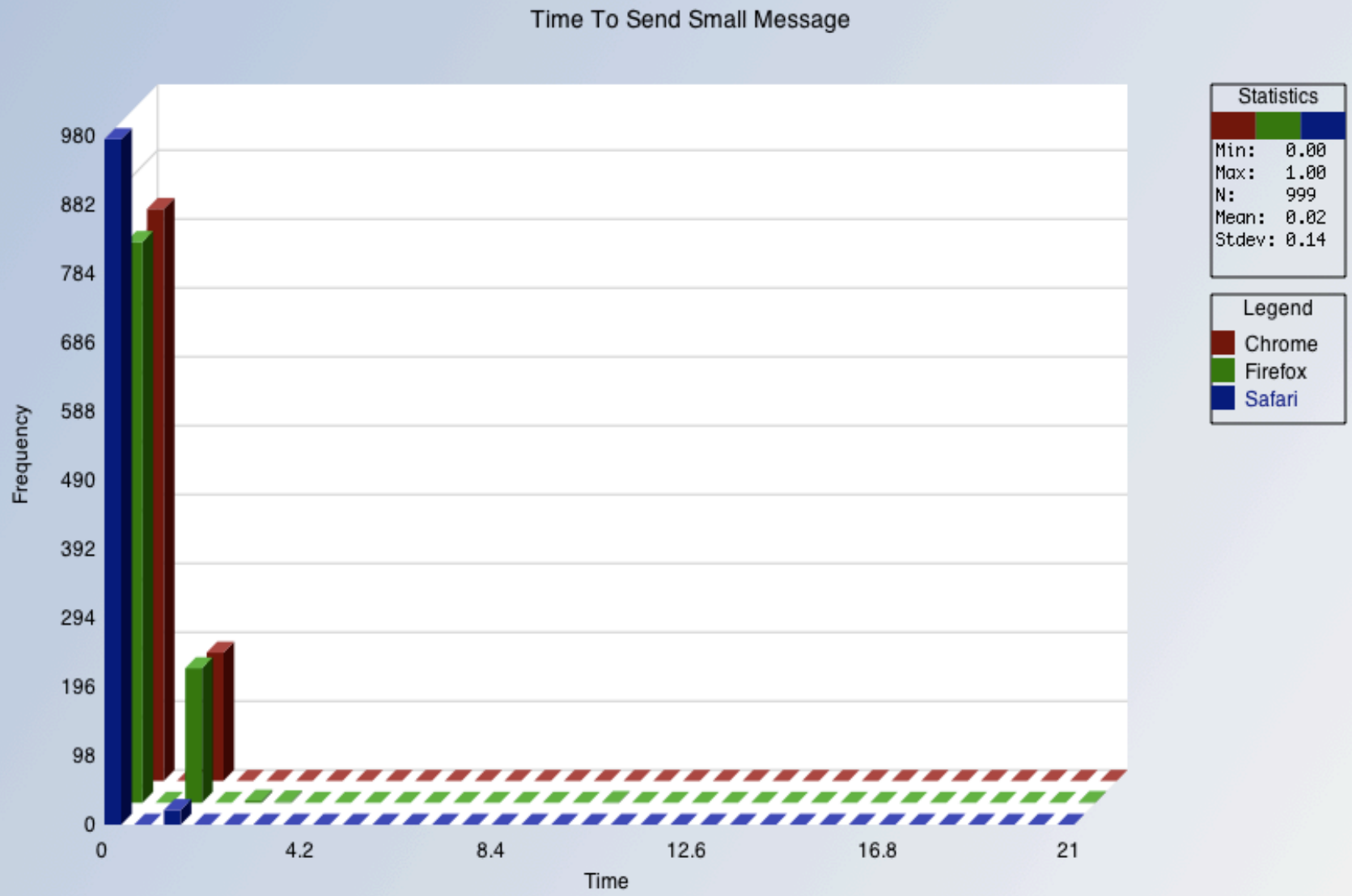
- Parallel Execution
- Communicates through Message
- Executes in Isolated Thread

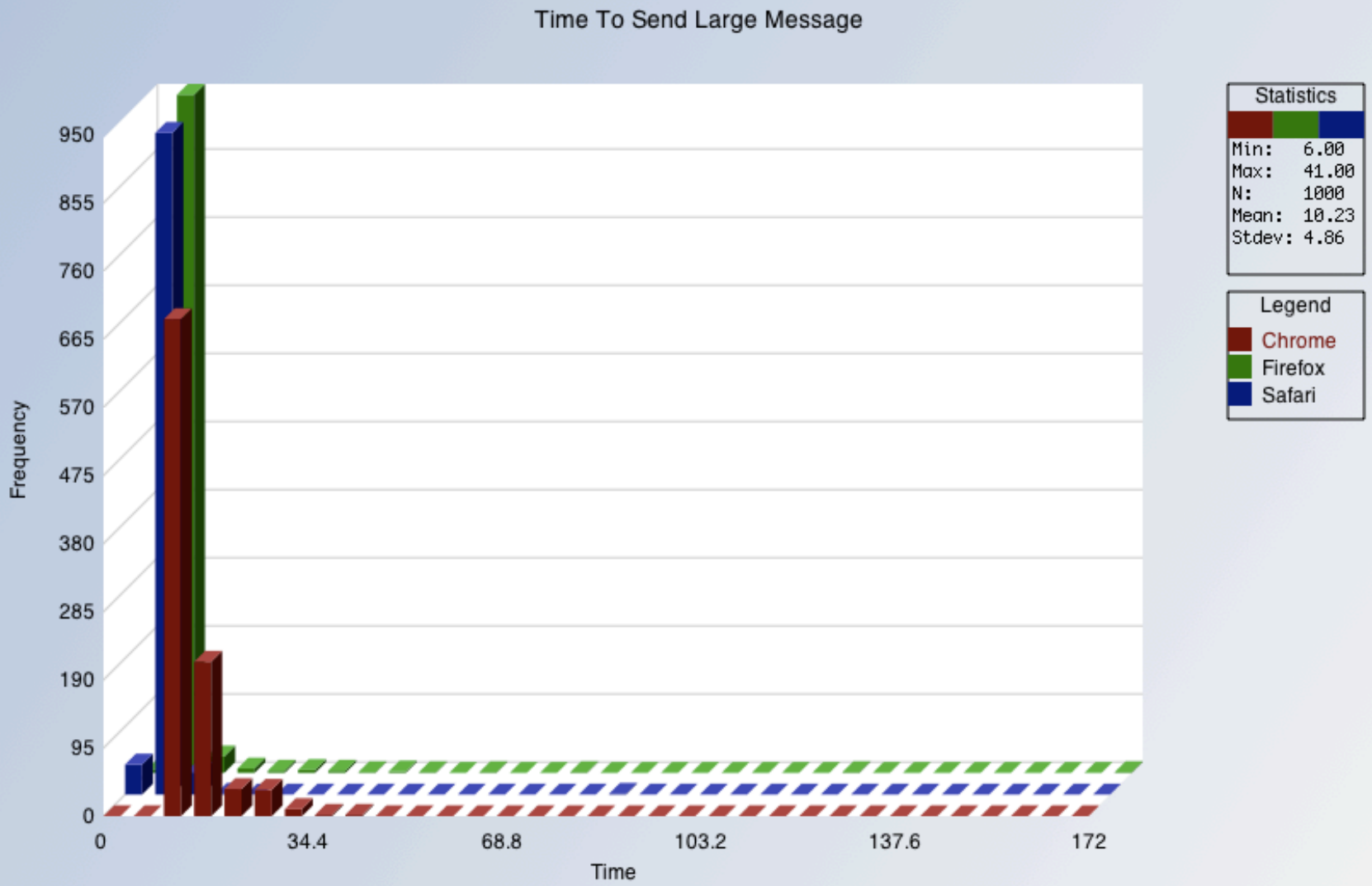
- No Access To
  - DOM
  - Window
  - Document
  - Parent
  - No Shared Memory
- But You Do Have…
  - XHR / WebSockets
  - Navigator
  - Location
  - setTimeout/setInterval
  - App Cache
  - importScript

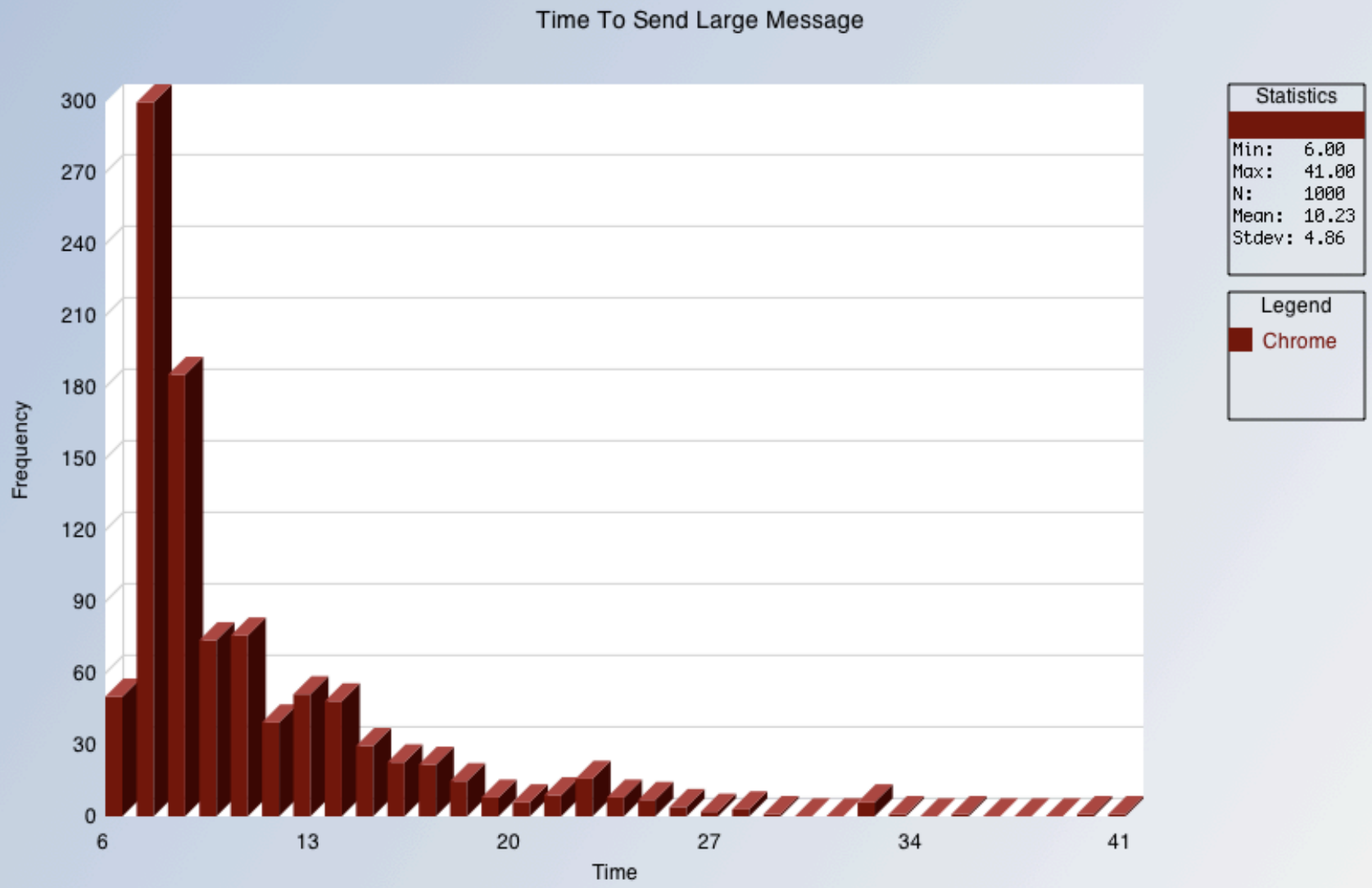
10 Thread Demo




Time To Create Second Worker
Frequency
980
882
784
686
588
490
392
294
196
98
0
0
2.4
4.8
7.2
9.6
12
Time
Statistics
Min: 0.00
Max: 1.00
N: 1000
Mean: 0.02
Stdev: 0.14
Legend
Chrome
Firefox
Safari


Time To Send Small Message
Frequency
980
882
784
686
588
490
392
294
196
98
0
0
4.2
8.4
12.6
16.8
21
Time
Statistics
Min: 0.00
Max: 1.00
N: 999
Mean: 0.02
Stdev: 0.14
Legend
Chrome
Firefox
Safari

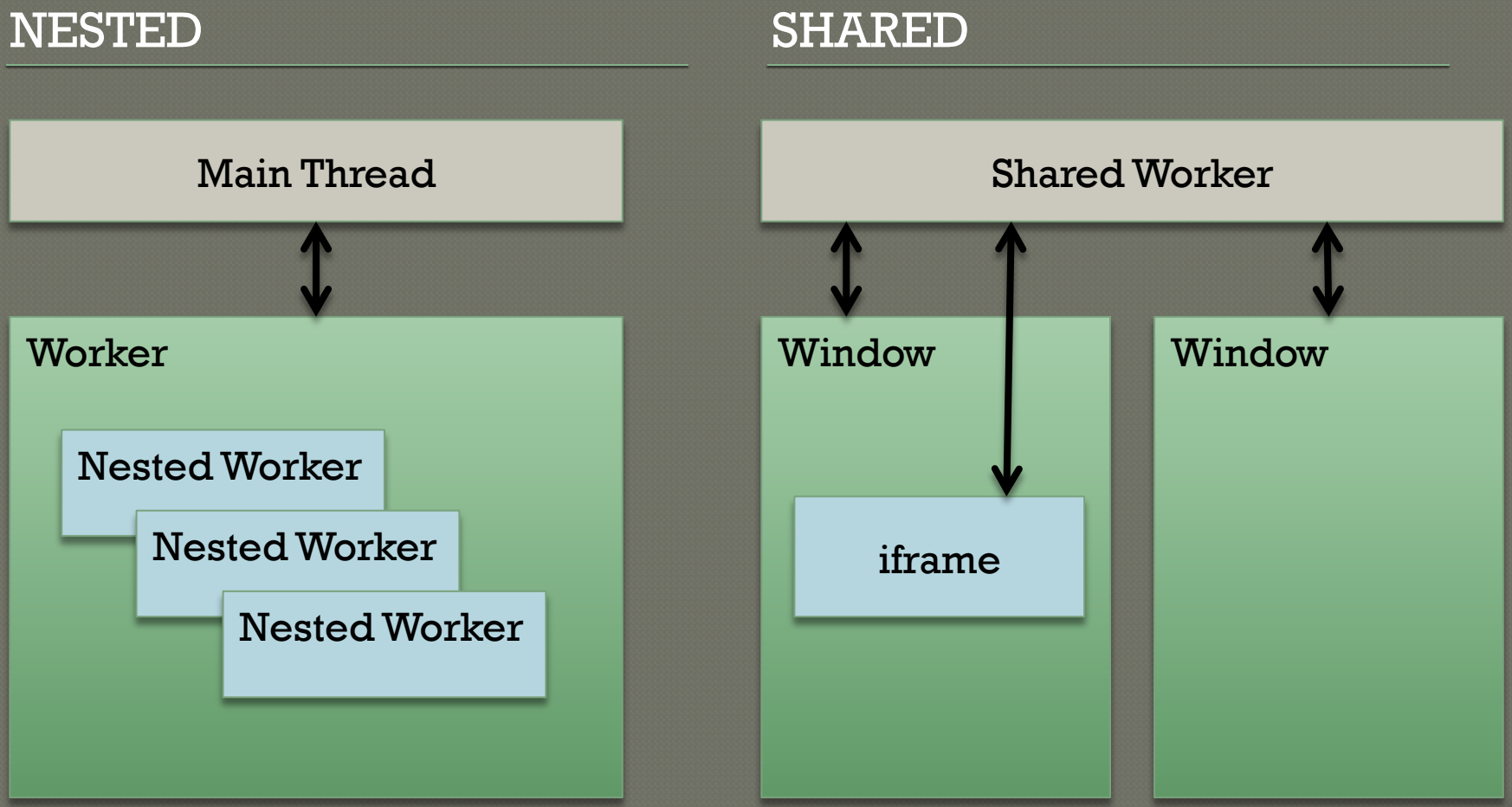

Time To Send Large Message
Frequency
950
855
760
665
570
475
380
285
190
95
0
0
34.4
68.8
103.2
137.6
172
Time
Statistics
Min: 6.00
Max: 41.00
N: 1000
Mean: 10.23
Stdev: 4.86
Legend
Chrome
Firefox
Safari


Time To Send Large Message
Frequency
300
270
240
210
180
150
120
90
60
30
0
6
13
20
27
34
41
Time
Statistics
Min: 6.00
Max: 41.00
N: 1000
Mean: 10.23
Stdev: 4.86
Legend
Chrome

Web Worker Variations

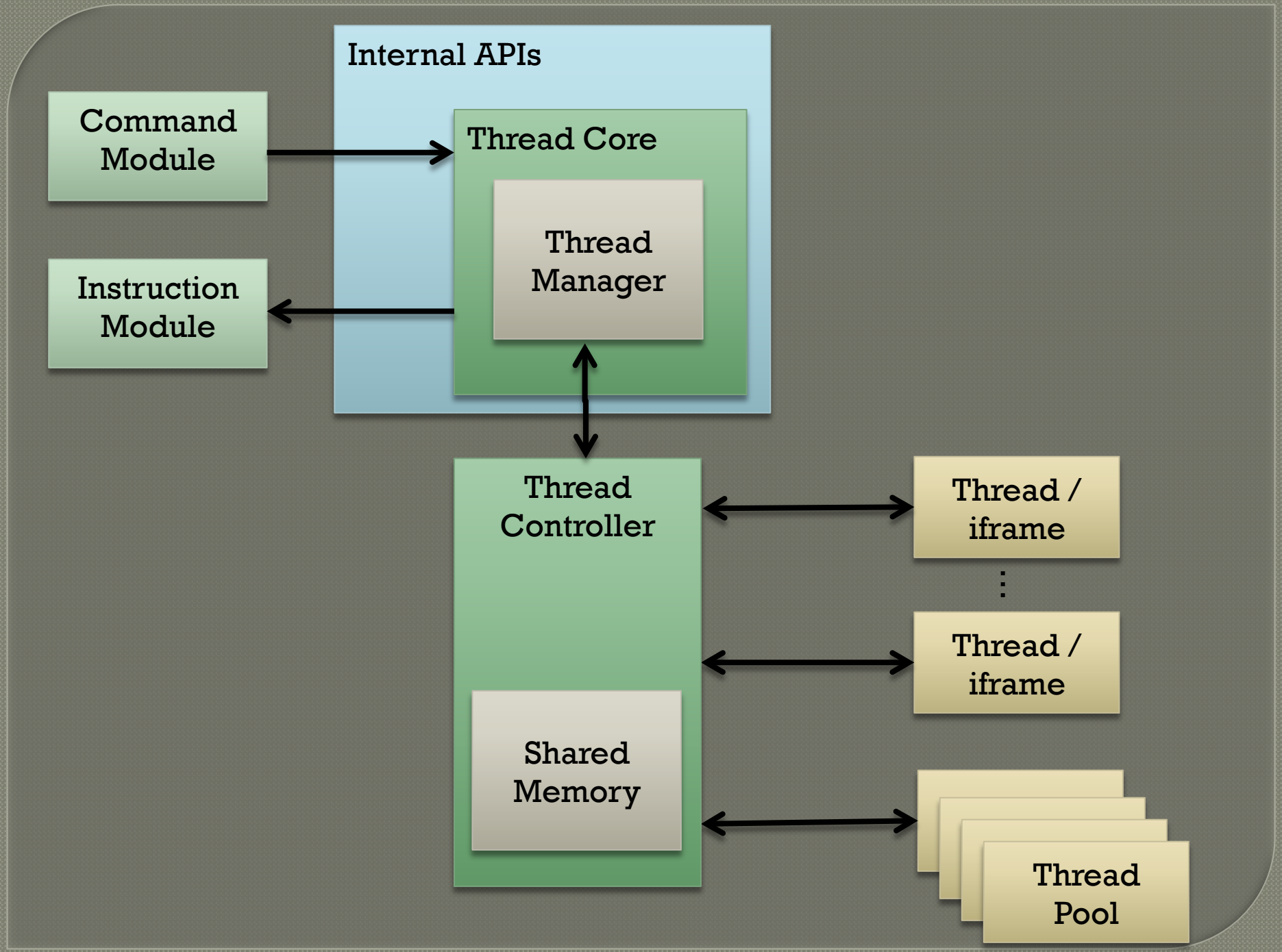
- Inline vs. External
  - BlobBuilder
  - Inline only supported by FF and Chrome
  - No difference in performance
- Dynamic
  - Method determined at run time from JSON string
  - Altered and changed by user at run time

Nested and Shared Workers



Platform Support

- Chrome/Firefox/Safari
  - Chrome/Safari does not support Nested
  - Firefox does not support Shared
- Android – (via Firefox Browser)
- iOS (as of Yesterday!)
- No IE, but expected


Internal APIs
Command Module
Thread Core
Thread Manager
Instruction Module
Thread Controller
Thread / iframe
Thread / iframe
Shared Memory
Thread Pool

What is WebGL?

WebGL

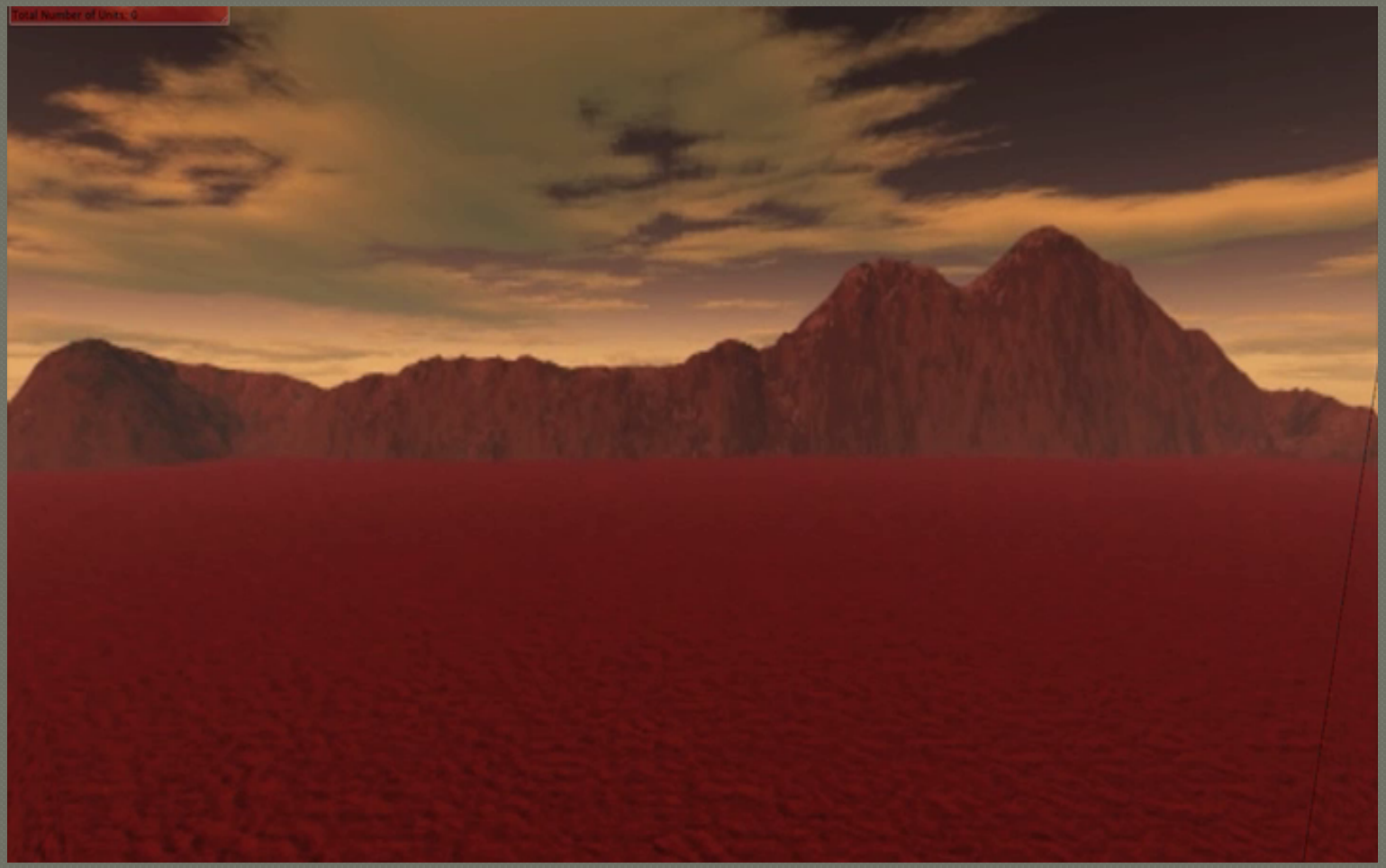
THE BASICS

- JavaScript Wrapper for OpenGL ES 2.0
- Programmable Graphics Pipeline (GLSL)
- Hardware Acceleration

SAMPLE API/LIBRARIES

- SpiderGL
- Copperlicht
- CubicVR
- Gladius

Performance Test



WebGL Hardware Test

Verts	Polys	Draw	FPS	CPU	RAM	Video	OS
858,750	485,292	654	30	2.2GHz Intel i7 Quad Core	4GB	AMD Radeon 6750M 1GB	OSX
634,179	353,386	483	30	2.53GHz Intel Core 2 Duo	4GB	NVIDIA 9800+ 1GB	Windows 7
590,898	333,924	450	33.33	2.53GHz Intel Core 2 Duo	8GB	NVIDIA GeForce 9600M GT 512MB	OSX

Chrome

Browser vs Browser

Browser	Verts	Polys	Draw Calls
Chrome	590,898	333,924	450
Aurora	426,773	241,174	325
Safari	295,473	166,974	225
Firefox	262,648	148,424	200

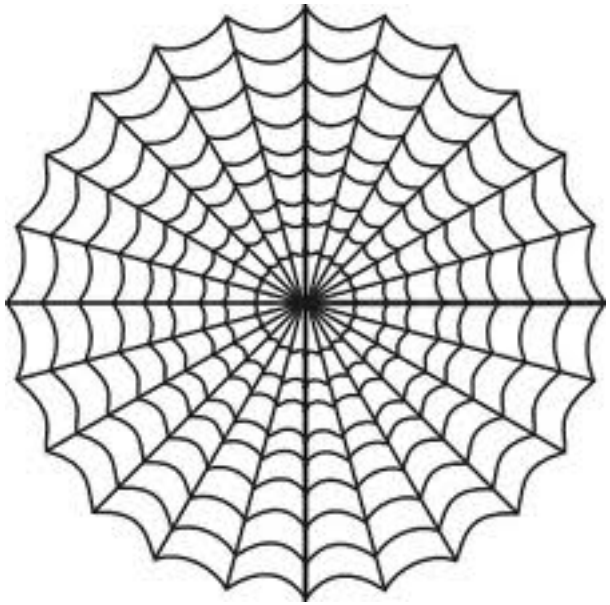
WebGL Tid Bits

TIPS

- WebGLDebugUtils.makeDebugContext(get3DContext(canvas));
- WebGL Inspector
- requestAnimFrame

UP AND COMING

- Firefox moving to multiprocess architecture
- WebGL Running in WebWorker?



+

+

Web Sockets

- Full Duplex Communication
- No HTTP overhead
- Has Secure Transport Protocol
- Cross Origin Communication
- Very Simple Client Side API
  - Socket.onmessage = function(){}
  - Socket.onopen = function(){}
  - Socket.open()
  - Socket.send()
  - Socket.close()

Nitty Gritty

- 2 Bytes Overhead Per Message
  - XHR ~871 Bytes
- Constant Connection
  - No need to re-establish connection (COMET)
- Connecting to Non Browser Applications (via Proxy/Server)
  - Byte Arrays
  - Bit Shifting

WebSocket Server Side

- Standard HTTP Request/ Response model performs poorly for high number of users
- Server Needs
  - Thread or Non Blocking IO Design
  - High concurrency at Low Performance cost
- Existing Servers
  - Node.js
  - Jetty (Java)
  - Ruby (Event Machine)
  - Python (Twisted)

node.js

FEATURES

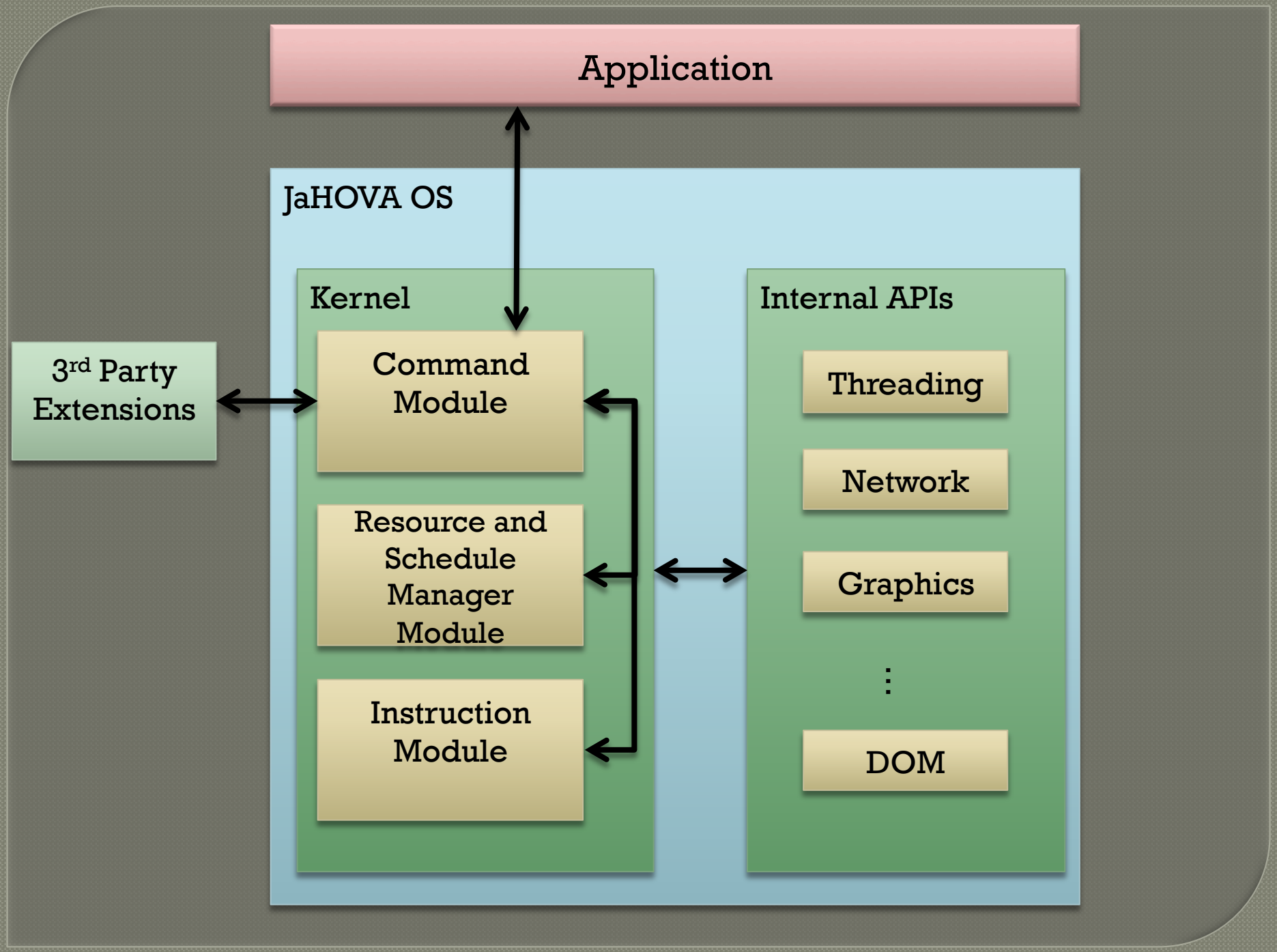
- Google V8 Engine
- JavaScript
- Event Driven
  - Event Loop with Callbacks
  - Non Blocking
- Native C Bindings
- Fast!!

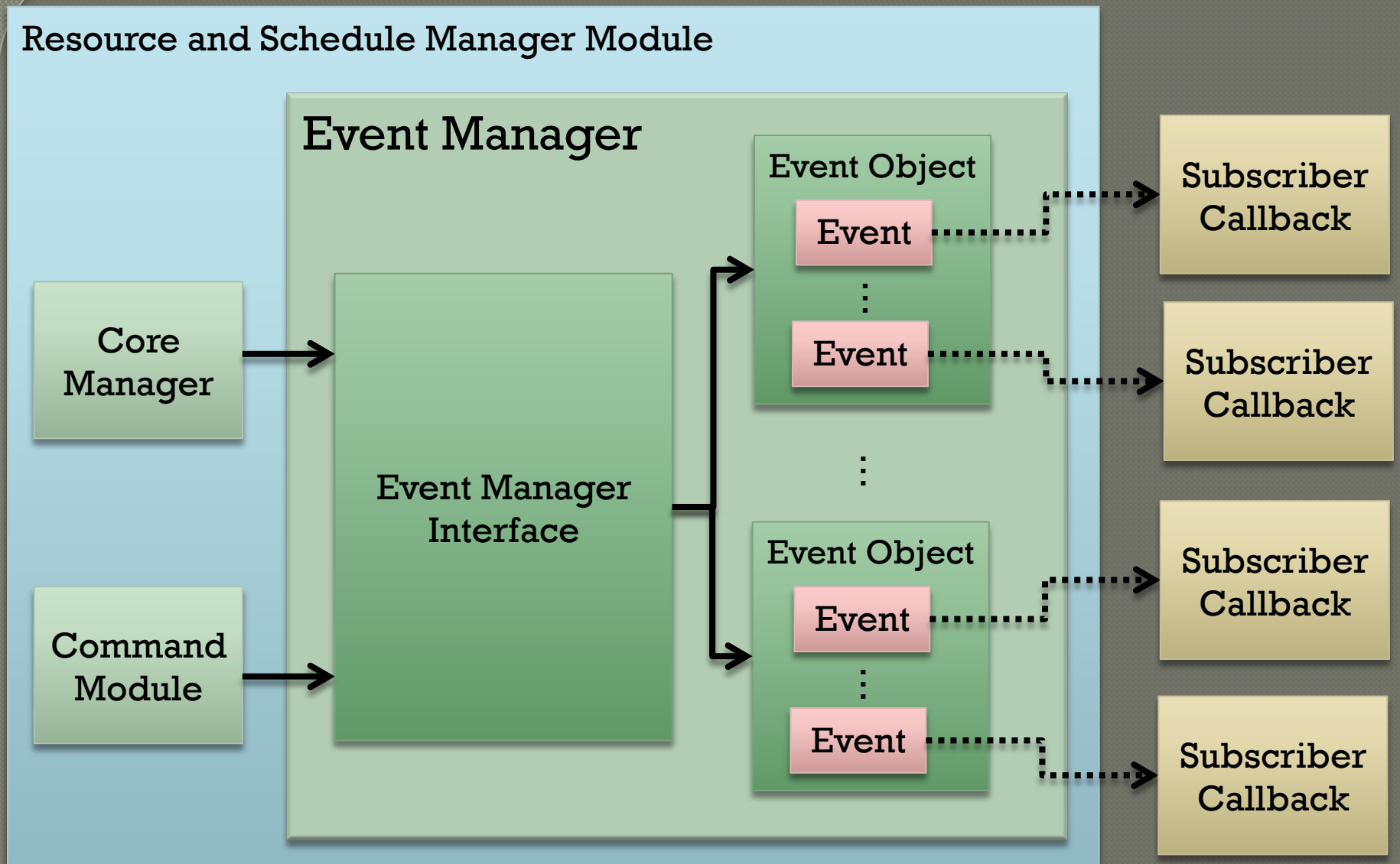
MODULES (NPM)

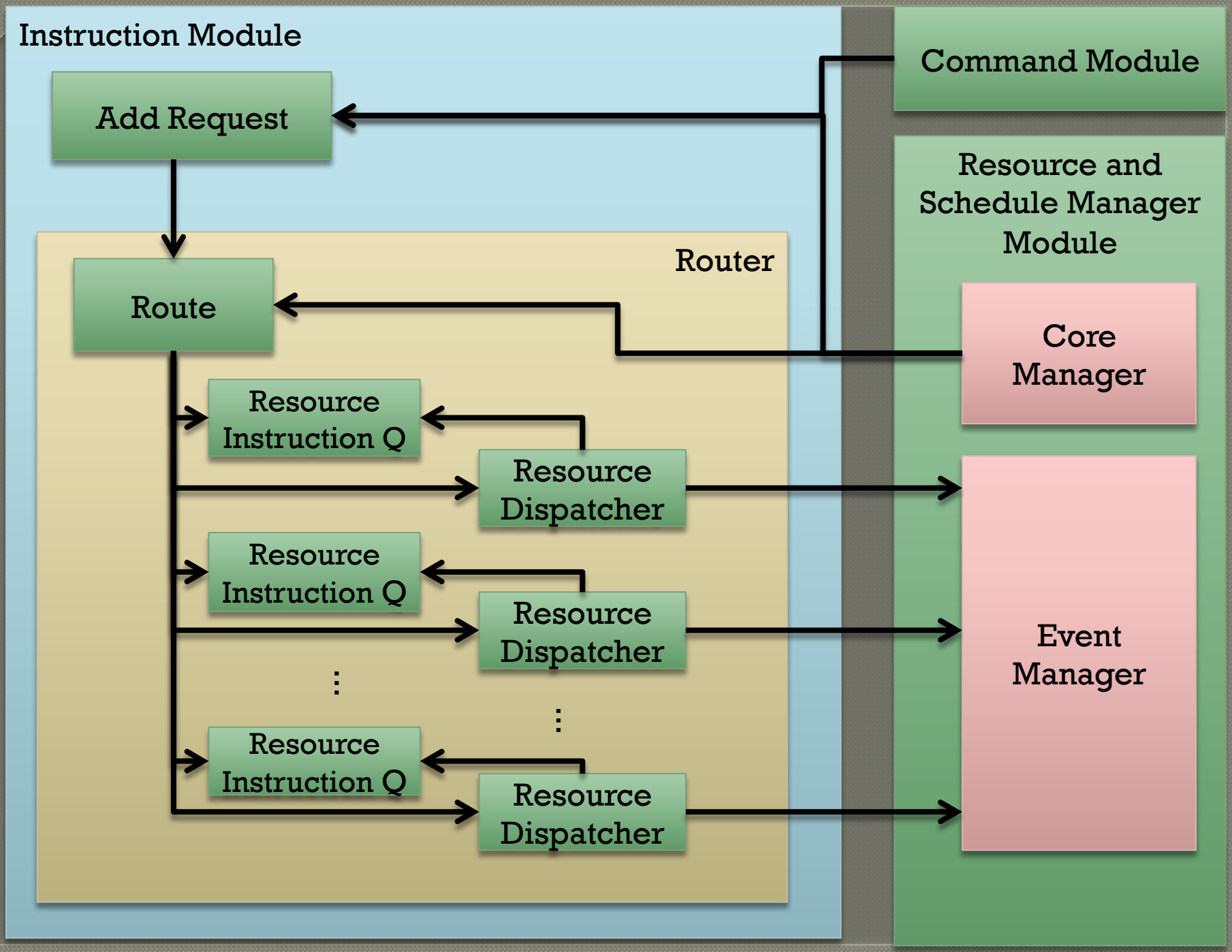
- WebSocket Server
  - Socket IO
  - Kaazing Gateway
- TCP Server
- File Server
- MySQL

Engine Architecture

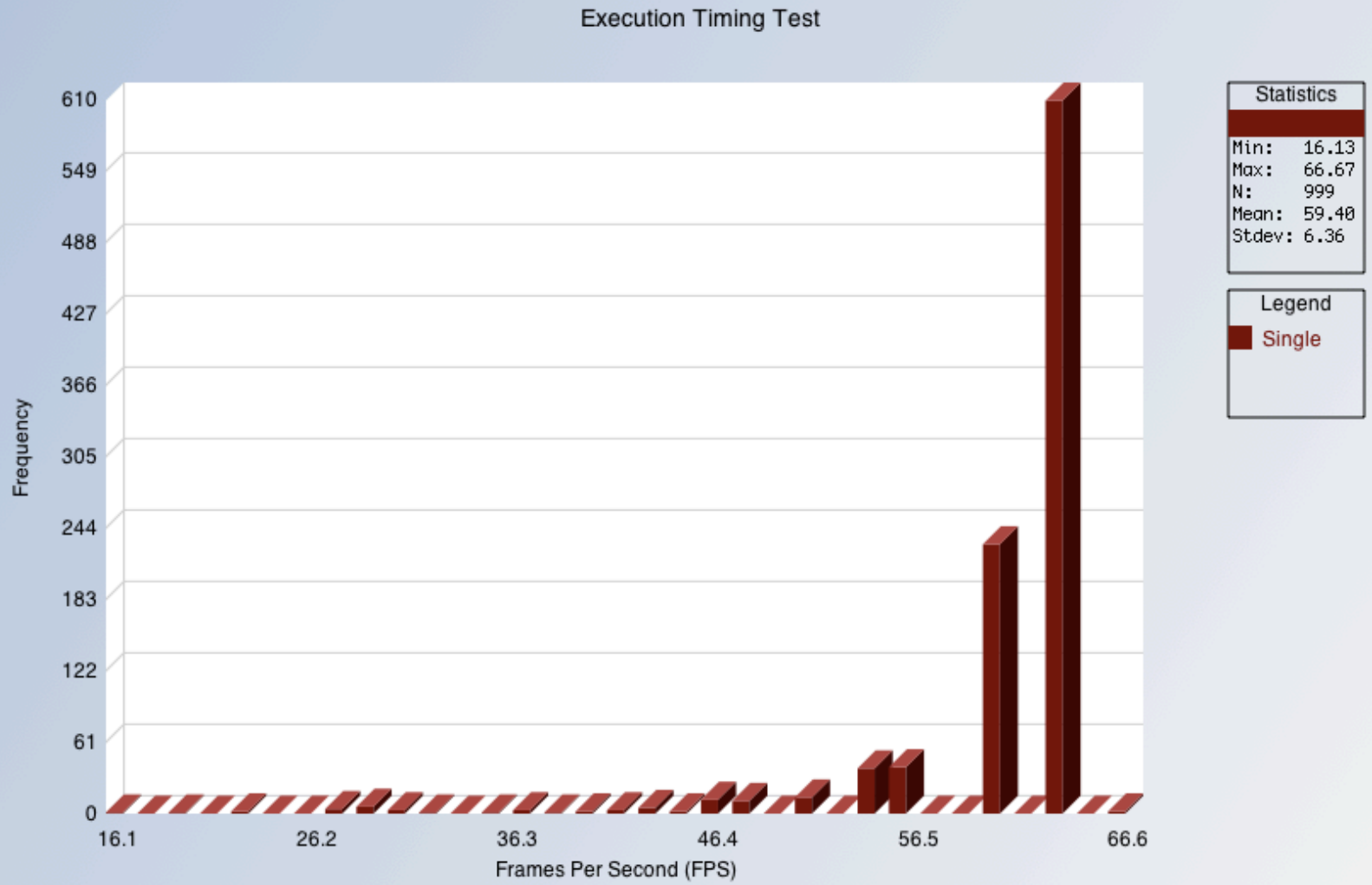
- Dynamic Nested/Shared Web Worker
  - Thread Controller
  - Shared Memory Pool
- Custom Event Management System
- Available Engines
  - Akihabara (2D)
  - Effect Game (2D)
  - Isogenic Engine
  - Rocket Engine


Application
JaHOVA OS
Kernel
Command Module
Resource and Schedule Manager Module
Instruction Module
Internal APIs
Threading
Network
Graphics
DOM
3rd Party Extensions


Resource and Schedule Manager Module
Event Manager
Core Manager
Command Module
Event Manager Interface
Event Object
Event
Event
Event Object
Event
Event
Subscriber Callback
Subscriber Callback
Subscriber Callback
Subscriber Callback

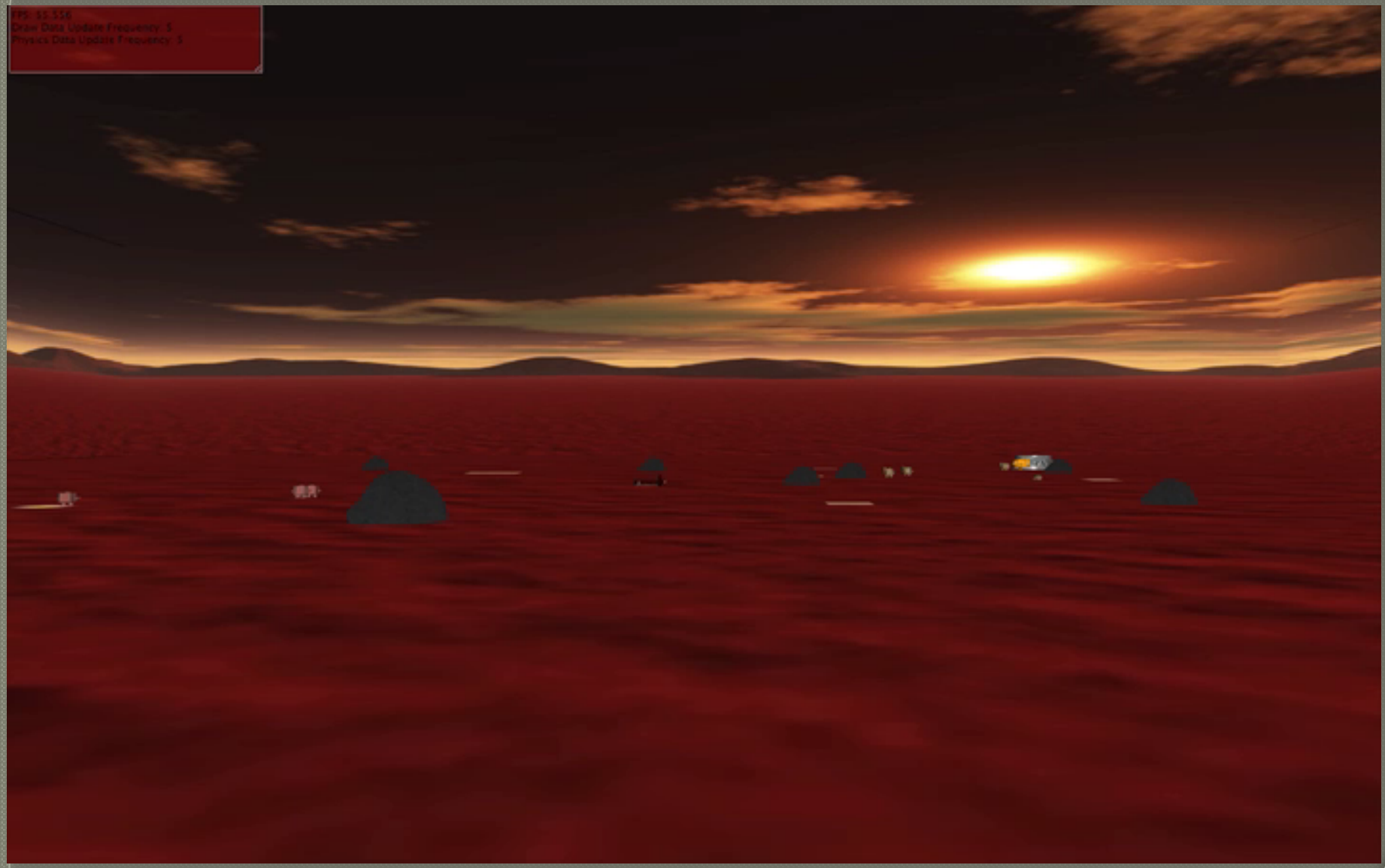

Instruction Module
Command Module
Add Request
Resource and Schedule Manager Module
Router
Route
Core Manager
Resource Instruction Q
Resource Dispatcher
Resource Instruction Q
Resource Dispatcher
Event Manager
Resource Instruction Q
Resource Dispatcher

Single vs Multithread Performance



Engine Demonstration

- WebGL
- Thread Controller
  - WebSocket
  - Physics
- Node.js
  - TCP
  - WebSocket
- PC C++/DirectX Clients


FPS: 33.336
Draw Data Update Frequency: 5
Physics Data Update Frequency: 5

Final Thoughts

- Initialize Threads at Startup
- Careful with Debuggers and Web Workers
- See more at
  - http://jahovaos.com
- Code samples at
  - http://demo.jahovaos.com