Presentation #4

MASTERING STREAMS IN NODE JS

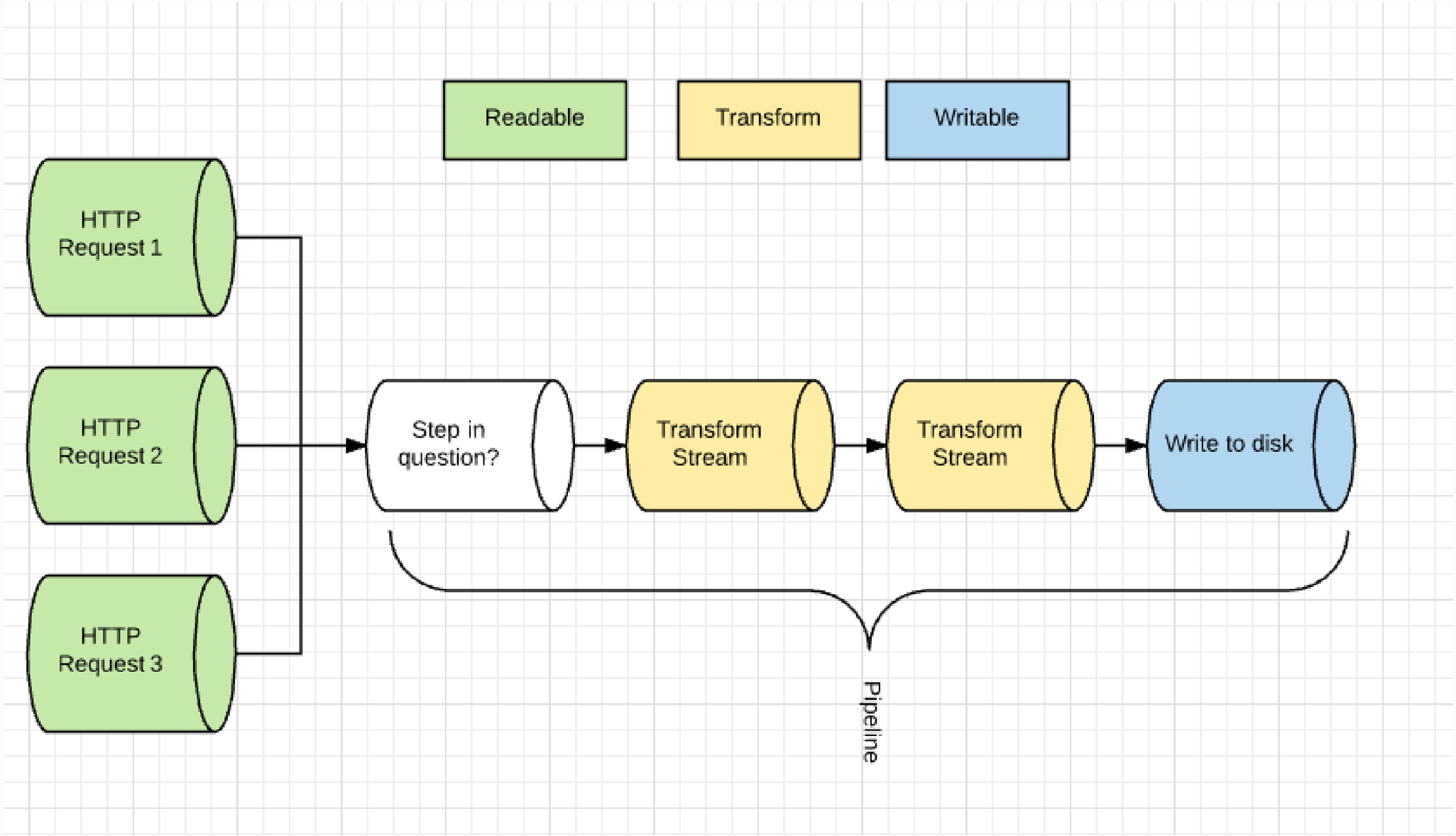
Github Organization

CREATED BY

KETI ELIZBARASHVILI

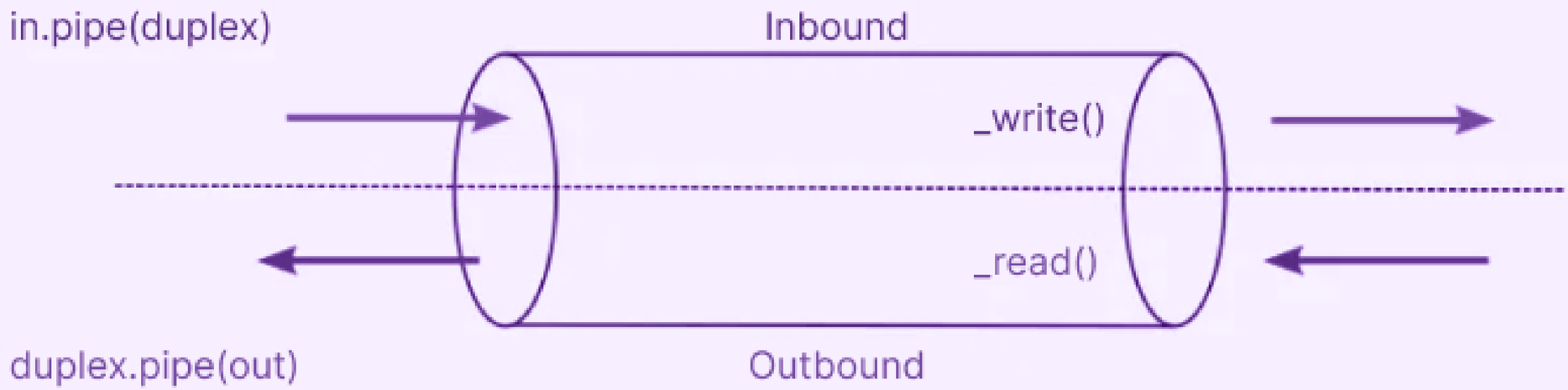
Agenda

1. Introduction to Streams
   - Definition of Streams in Node.js
   - Types of Streams: Readable, Writable, Duplex, and Transform
   - Importance of Streams in handling large amounts of data efficiently
2. Understanding Streams
   - How Streams Work
   - Buffering and Flowing Modes
   - Event-driven architecture of Streams
3. Readable Streams
   - Creating a Readable Stream
   - Handling 'data', 'end', and 'error' events
   - Practical Example: Reading from a file
4. Writable Streams
   - Creating a Writable Stream
   - Writing data using 'write' method
   - 'finish' and 'error' events
   - Practical Example: Writing to a file
5. Duplex and Transform Streams
   - Understanding Duplex Streams
   - Creating and using Transform Streams
   - Practical Example: Transforming data on-the-fly
6. Implementing Backpressure
   - What is Backpressure?
   - Handling Backpressure in Node.js Streams
   - Example scenarios demonstrating backpressure
7. Piping Streams
   - The concept of Piping
   - Using 'pipe' method to connect streams
   - Chaining multiple streams
   - Practical Example: Piping streams for file transfer
8. Error Handling in Streams
   - Common errors in Stream handling
   - Best practices for robust error handling
9. Advanced Topics
   - Stream Implementations in popular Node.js modules
   - Custom Streams: Creating your own Streams
10. Conclusion and Q&A
    - Recap of key points
    - Open floor for questions


Readable
Transform
Writable
HTTP Request 1
HTTP Request 2
HTTP Request 3
Step in question?
Transform Stream
Transform Stream
Write to disk
Pipeline


Input
Chunk
Chunk
Chunk
Node.js
program
Chunk
Chunk
Chunk
Output

Duplex Stream



Practical Work