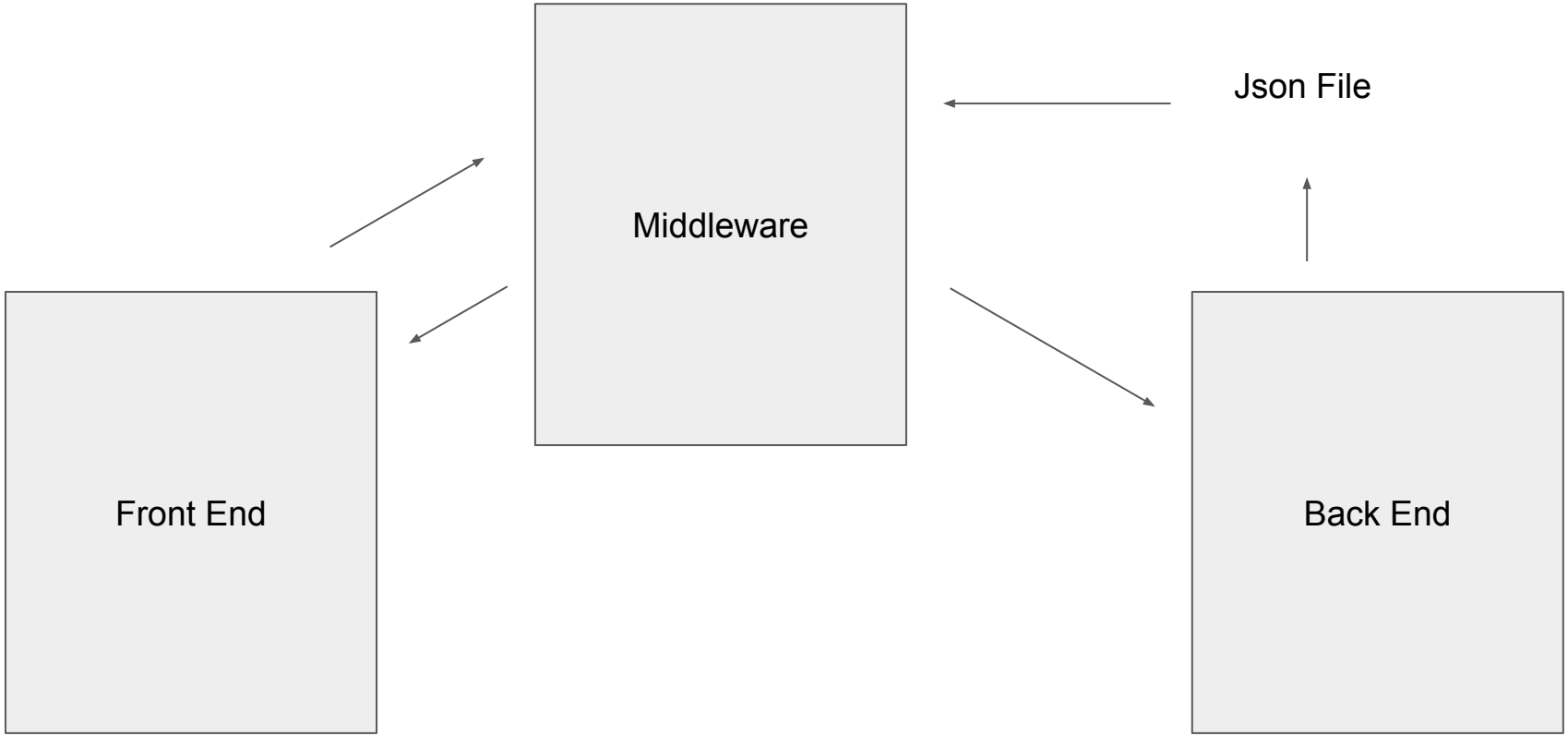
Lab Meeting


Middleware
Json File
Front End
Back End

Methods

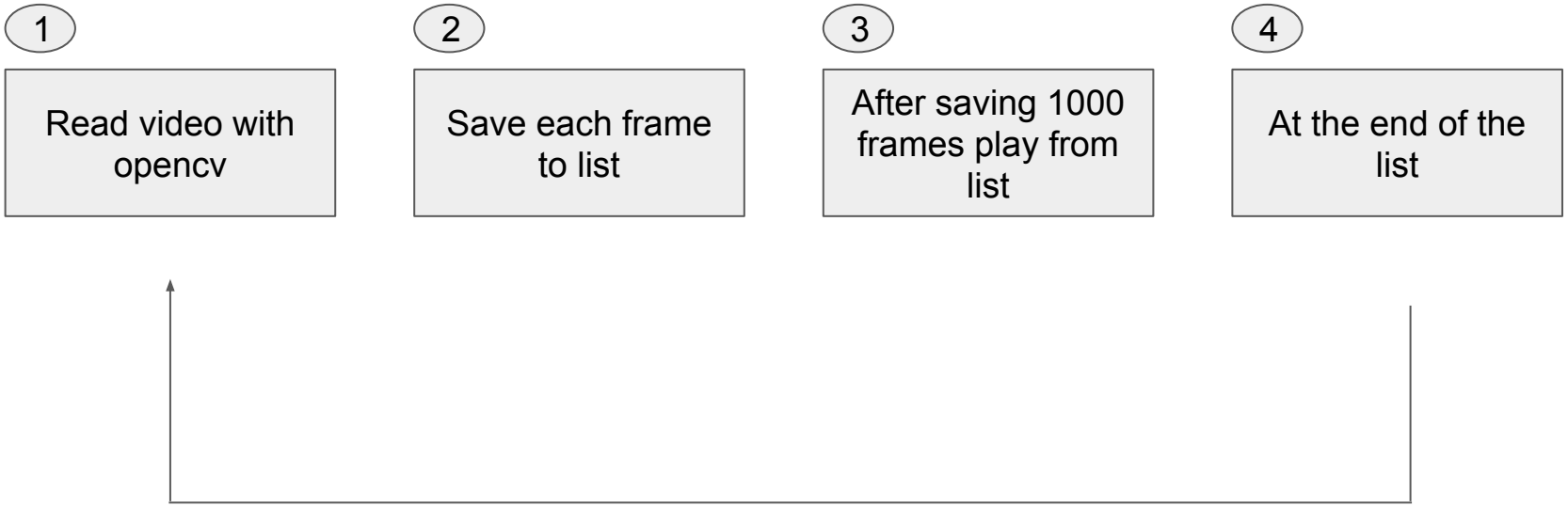
Python	C++	Javascript
- Kivy - OpenCV	- Openframeworks	- Electron - OpenCV

Main reason for front-back integration

- Multiple messages to be sent back and forth
  - It will overcomplicate the process
    - Will require numerous functions to handle each edge cases

- Native Integration
  - Each framework is natively integrated into each language

Python


1
Read video with opencv
2
Save each frame to list
3
After saving 1000 frames play from list
4
At the end of the list

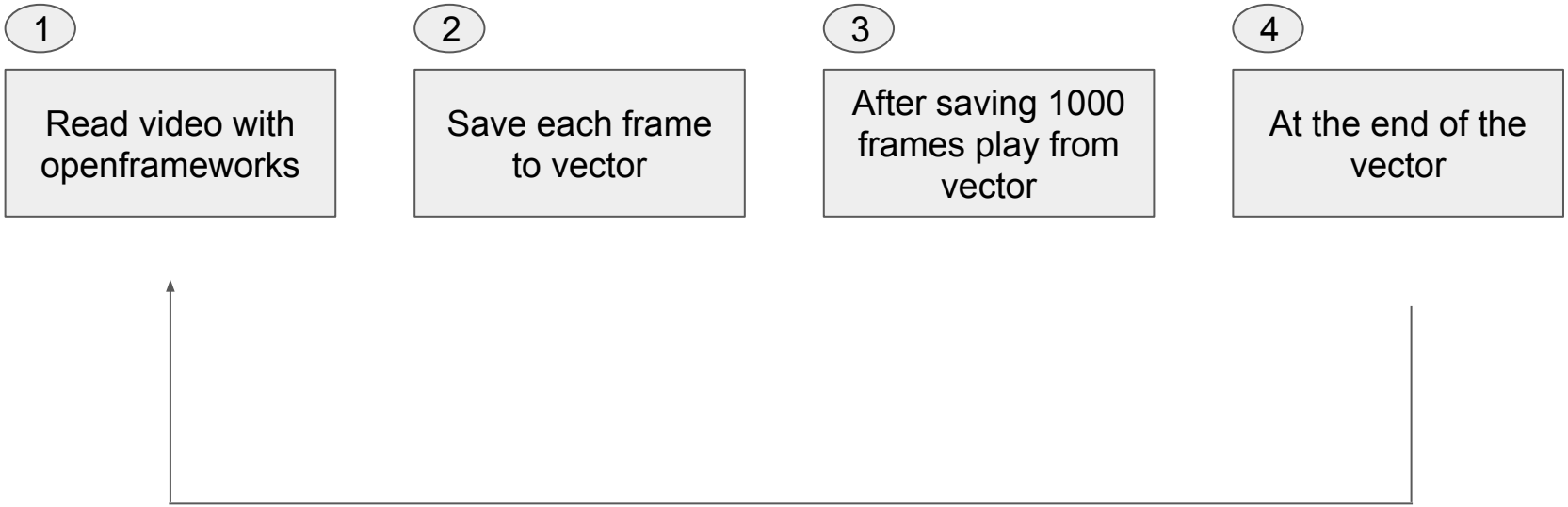
Reasons

1. Read from OpenCV
    a. Supports user friendly functions for drawing on each frame
2. Save to list
    a. Faster than storing in numpy array
    b. Encode/Decode is for network communication, so there had to be one more step for encoding/decoding, which resulted in slower process
    c. Buffer also consumed a lot of time, which resulted in slower process
    d. Storing 1000(Resolution=1920x1080, Duration=50s) frames took about 15 seconds with list
3. Kivy is implemented in C-level for time-critical functionalities
    a. Cross-platform library

Problems

1. Converting the frame to kivy format slowed down the process
    a. Resolution up to 720p was played in normal speed
    b. The high resolution was not a problem
        i. Tested on both opencv and kivy
2. Storing slowed down significantly after 1500-2000 frames
    a. As the list stores more frames, it slowed down more
    b. Have to restore the frames into a new list

C++


1
Read video with openframeworks
2
Save each frame to vector
3
After saving 1000 frames play from vector
4
At the end of the vector

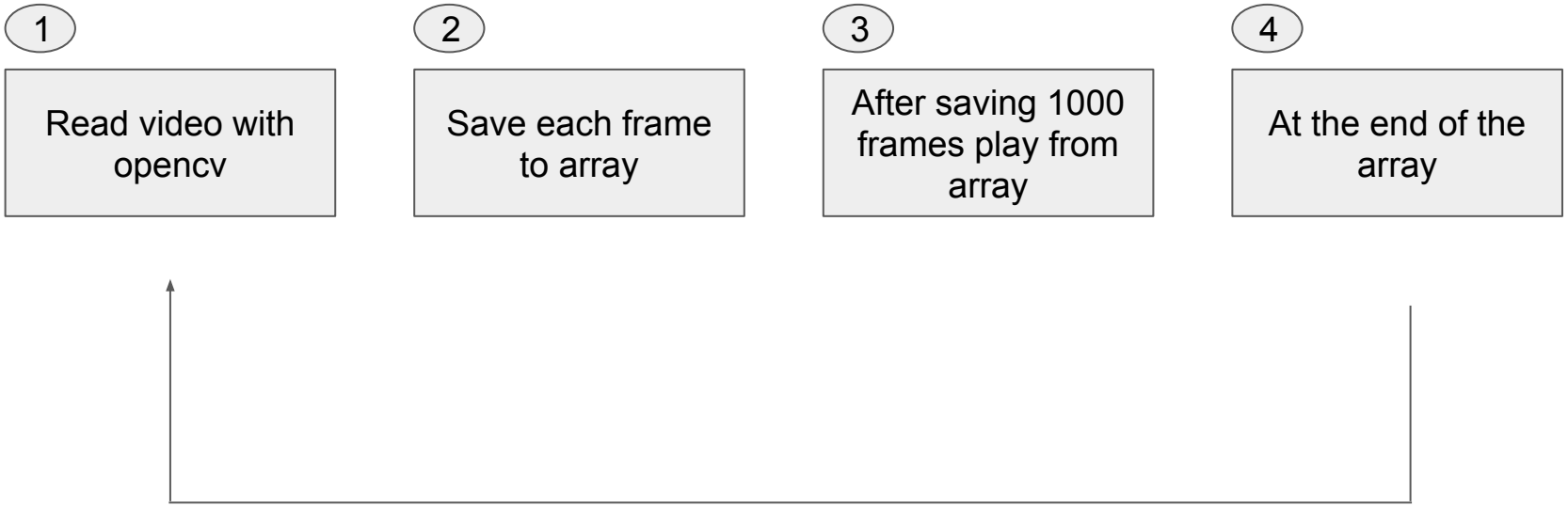
Reasons

1. Openframeworks is designed for visual design
    a. Provide more functions than OpenCV
    b. OpenCV is already included in openframeworks
2. C++ is already faster than Python or Javascript
    a. 5-7 faster than Javascript
    b. Python is up to 400 slower than C++
3. Able to play 1920x1080 resolution video with normal speed or even faster

Problems

1. High resolution video play was not smooth
    a. Resolution up 720p was smooth
2. May need more steps to run in cross-platform environment
    a. Need a separate mac environment to build the executable file
3. Playing a .mp4 video with openframeworks needed a separate codec installed
    a. Uses Direct-show
        i. Developed by Microsoft
        ii. Used for C++ development purpose
    b. K-lite codec download

Javascript


1
Read video with opencv
2
Save each frame to array
3
After saving 1000 frames play from array
4
At the end of the array

Reason

1. Only 5-7 times slower than C++
2. Supports abundant features for both front and back ends
    a. HTML/CSS used for web design purposes and Javascript for backend
3. Electron supports cross-platform
    a. Easy to install on Mac, Windows, and Linux

Problems

1. Hard to install on Electron
    a. Still in process of installing opencv on Electron