

# Accessing Preserved? Archived? Conserved? Games using Emulation

Knowledge Exchange Event @ Melbourne University (June 2014)

Ian Welch and Stuart Marshall

*School of Engineering and Computer Science*

*Te Kura Mātai Pūkaha, Pūrорohiko,*

*Victoria University of Wellington, New Zealand*



**Visitors (physical or virtual) should be able to interact with preserved software as easily as borrowing a book from the library.**

**What are the available technical options? What are the tradeoffs?**



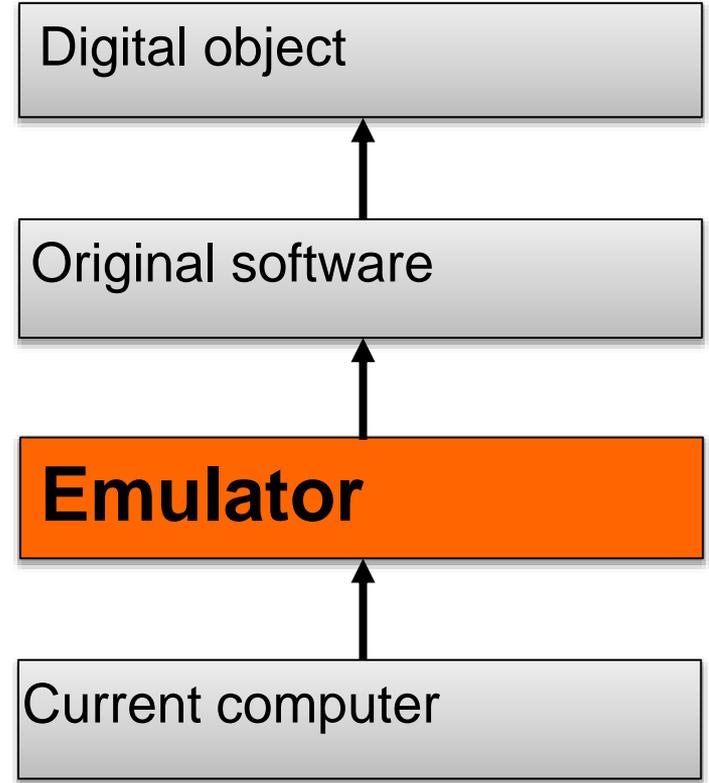
# Emulation for Access

Emulators recreate the original hardware/software environment in a new hardware/software environment.

Allows you to run the original software “as is”.

Client-side emulation requires you to install an emulator on the device.

Within an institution might be PCs provided locally, a service provided over the Internet (**legal issues!**) or BYO devices such as mobile phones etc.



# Technologies

Community has developed many emulators.

Native applications (for example, **ubee512**) that you install on your device (PC in this case).

Flash-based (for example, <http://nesbox.com/>) only require browser plugin (downside – won't work on iOS).

HTML5/Javascript based emulators provide greater portability depending upon capabilities of the browser.

1. The generic approach (Internet Archive) (<http://jsmess.textfiles.com/> port of jsmess).
2. The per-machine approach (<http://nanowasp.org>).



## What is NESbox?

NESbox is an emulator of NES, Super Nintendo, Sega Mega Drive video consoles, built on Adobe Flash technology and it can only be run directly in your browser's window.

[Learn more >](#)



# Pro's and Con's

## Pro's:

- Browser-based emulators = zero installation.
- Scales with the number of visitors.
- No security concerns (for the institution).
- Once browser closed, copyright material removed from the client's machine.

## Con's:

- Javascript does vary between browsers, **won't work with all browsers.**
- Responsiveness depends on capabilities of the client's machine. **Works well on PCs but mileage varies on mobile devices.**
- Machine-specific emulators often assume a physical keyboard.
- Generic emulators tend to be slower (but could get better! JsMESS is in beta).



## What is NESbox?

NESbox is an emulator of NES, Super Nintendo, Sega Mega Drive video consoles, built on Adobe Flash technology and it can only be run directly in your browser's window.

[Learn more >](#)



# Remote Access to Emulators

Run the emulator on remote machines.

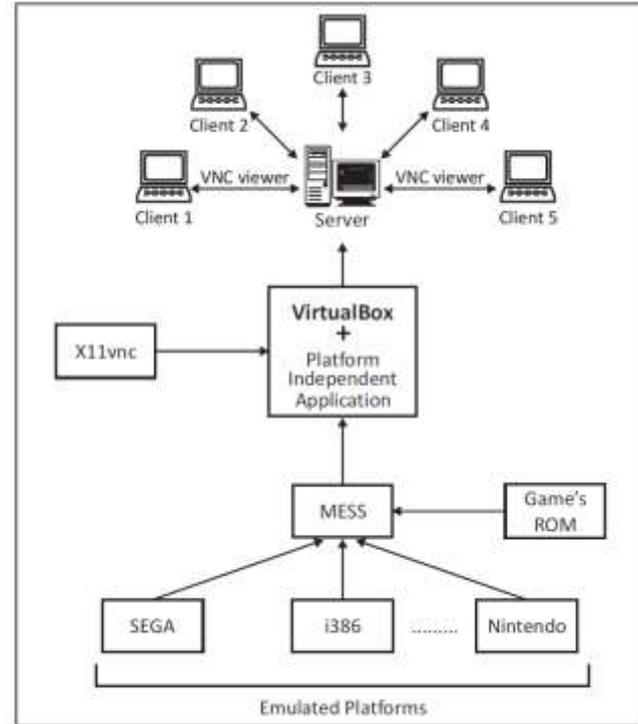
Use generic viewer running on a client.

## Pro's:

- Client is much simpler than the emulator (reduces client hardware requirements).
- **One** client can be used to **access a wide number of different emulators**.
- Full control over access to copyrighted content by the institution (except for artwork ...).

## Con's:

- Responsiveness is affected by network performance (latency and bandwidth).
- Need to provision machines to run the emulators (possibly could use the cloud or provide your own cluster).
- Need to ensure remote user cannot break out of the emulator or the virtual machine.



(Ausama Al-Sahaf, 2012 – Evaluation of Remote Client Approach using VNC).

# Technologies

We have built some prototypes demonstrating practicality of the approach and investigated minimum performance requirements.

bwFLA — Emulation as a Service (<http://bw-fla.uni-freiburg.de/>) is a state of the art implementation by Dirk von Suchodoletz et. al. from Freiburg University (came out of KEEP and related EU projects).

Scalable -- spawn environment on demand based upon **technical details**.

Portable approach for accessing the service can be installed:

- as native applications for PCs and mobile devices.
- as browser-based zero install applications for PCs and mobile devices.

1. Example of the Hobbit running on a local server accessed via a browser client.

[http://192.168.43.61:6080/vnc\\_auto.html](http://192.168.43.61:6080/vnc_auto.html)

2. Example of using it to provide access to hypercard (served from Germany):

<http://bw-fla.uni-freiburg.de/demo-flusser.html>

# What next?

Want to use these technologies to provide access to the Popular Memory Archive.

Key idea is to provide two levels of access for different types of audience:

- (1) large number of casual users accessing particular examples of historic software (via the website);
- (2) small number of expert users who need access to a wider range of systems and more faithful emulation (perhaps onsite).

Provide **access** to casual gamers and casual programmers via javascript/html5 emulators.

Focus on specific emulators for specific platforms/games of historical value (want good performance).

Assume users using particular browser on a PC/Mac. Evaluate mobile device access.

Longer term:

- Build upon the work done by bwFLA to allow access to on-demand emulation (mobile device access, performance engineering, security issues, definition of technical metadata needed for on-demand emulation).
- Improve the performance of JSmess as a generic solution for emulation so can be used for casual users?