



**Carnegie Mellon University**

**Preserving Software:**  
*The Olive Archive*

*Presenter*

**Daniel Ryan**

Curator of Executable Content

# The Problem

**Born digital interactive and executable content represents an increasing fraction of the creative output of society.**

- How can we capture this content in a faithful and reliable manner?
- What constitutes reliable preservation?
- Which software should we preserve?

# What about hardware?

**Hardware preservation is very important.**

That said, it is not a sustainable approach to digital preservation at scale.

- This approach costs us something in the user experience department
- We are focused on preserving the bits, which seem to slip away at an alarming rate

# Our Approach

**Olive is a virtual machine and emulation based preservation ecosystem for complex and interactive digital content.**

The basics:

- One piece of software = one virtual machine (space is cheap)
- Open source from the ground up to avoid a single point of failure
- Like YouTube for software

The code is available on github:

<https://github.com/cmusatyalab/vmnetx>

# The User Experience



## Olive Executable Archive

Olive is a collaborative project seeking to establish a robust ecosystem for long-term preservation of software, games, and other executable content. Born at [Carnegie Mellon University](#), Olive addresses the current gap in preservation technology by providing a curated environment for the preservation and distribution of executable content.

### What is Olive?

Learn more about the Olive Project.

[About Olive »](#)

### Papers & Presentations

View presentations and read papers by the Olive team.

[View Documents »](#)

### News & Updates

Get the latest news from the Olive Project.

[Read the Blog »](#)

© Olive Archive 2013

# The Magic

## **How it works:**

- VM Streaming (Disk is demand-paged over the network)
- Access control
- Cloud-based or local execution
- Archival copy is pristine, but individuals can still modify and interact with a working copy

## **Why that's good:**

- Buffering of disk access (a la YouTube)
- Emulators go inside a VM, eliminating complex emulator rewrites
- Single point of maintenance (VM monitors)
- Low barrier to entry – can you make a VM?

# Challenges

- Intellectual Property
- Best practices
- Sustainability
- Technological
- Psychological





# Thanks!

URL: <https://olivearchive.org>

GitHub: CMUSatyaLab/vmnetx

