Digital Vellum and Archives

Vint Cerf
Google

August 2016

Scope of Digital Preservation

- Digital object structures, representations, vocabulary and standard terminology (schema, OWL, …)
- Identifier spaces, registries, resolution mechanisms
  - The irony of WWW, URLs, DNS (TBL was at CERN)
  - Robert Kahn: Digital Object Architecture, CNRI
- Standard, rigorous ingestion processes
- Metadata (about the data, provenance, authenticity, calibration, ...)
- Legal frameworks for preservation (copyright, patents, licensing, special treatment for perserving bodies)
- Business Models for extended, long term operation

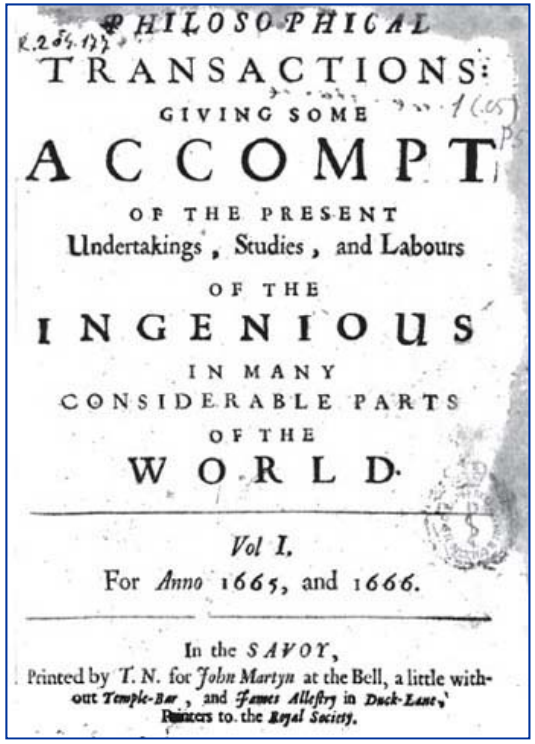
Recursive Processes

- Representation Information and the Designated Community
  - The longer the preservation period, the more we must assume lack of specific knowledge of the content and its structure and semantics
  - But the recursive loop has to stop somewhere
- Standard descriptive definitions, terminology, system representation (e.g. Open Archival Information System [OAIS])
- Specializations for particular kinds of data, objects, especially those with structure

Systematic Approach

- We need to adopt a systematic approach to long term archiving and preservation of information (such as the OAIS methods)
- The PCAST and OSTP initiative to include data archiving for all federally-funded research needs practical and affordable solutions
- Distributed archiving systems to spread load and risk
- Policy frameworks for incentive to implement and maintain solutions

Archiving Static Content


PHILOSOPHICAL
TRANSACTIONS:
GIVING SOME
ACCOMPT
OF THE PRESENT
Undertakings, Studies, and Labours
OF THE
INGENIOUS
IN MANY
CONSIDERABLE PARTS
OF THE
WORLD.
Vol I.
For Anno 1665, and 1666.
In the SAVOY,
Printed by T. N. for John Martyn at the Bell, a little without Temple-Bar, and James Allestry in Duck-Lane,
Printers to the Royal Society.

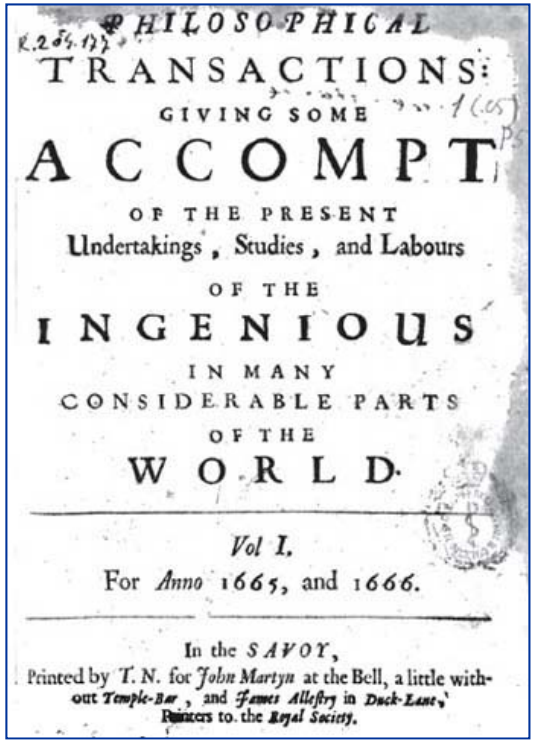

boedem
FLOPPY DISK
5,25" HD,DS


BUYDIG.COM


TDK


DVD
Blu-ray Disc

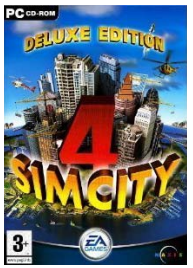
Archiving Static Text/Image Content

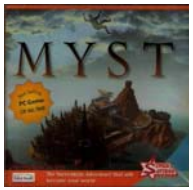

PHILOSOPHICAL
TRANSACTIONS:
GIVING SOME
ACCOMPT
OF THE PRESENT
Undertakings, Studies, and Labours
OF THE
INGENIOUS
IN MANY
CONSIDERABLE PARTS
OF THE
WORLD.
Vol I.
For Anno 1665, and 1666.
In the SAVOY,
Printed by T. N. for John Martyn at the Bell, a little without Temple-Bar, and James Allestry in Duck-Lane,
Printers to the Royal Society.

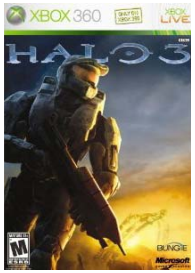
22nd Century
Doris Kearns Goodwin

- A Team of Rivals (Lincoln)
  - How did she reconstruct the dialog??
  - 100 Libraries and repositories w/physical correspondence
- What will the 22nd C. Doris Kearns Goodwin find?
- What will the National Archives be able to offer?
- What will our descendants know of our 21st Century?
  - Correspondence, entertainment, advertising, education, jobs, family life,…

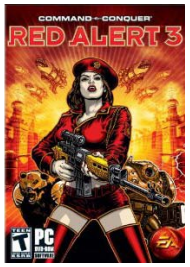
What About Executable Content?



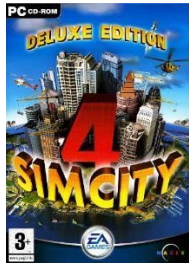


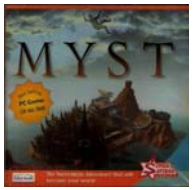


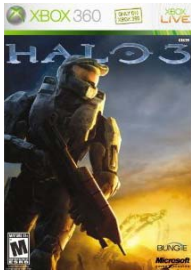
Games



What About Executable Content?

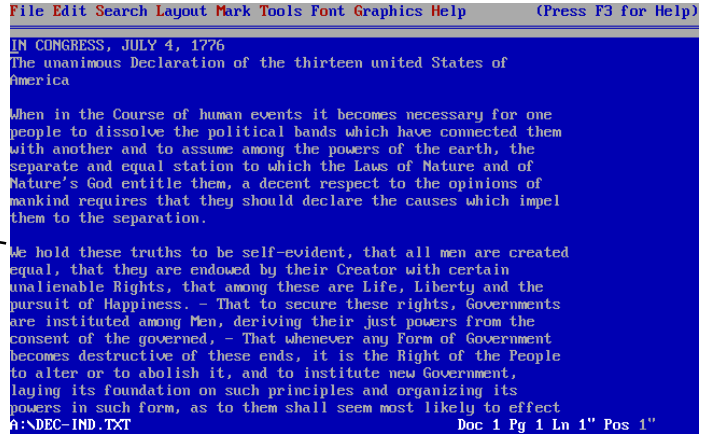


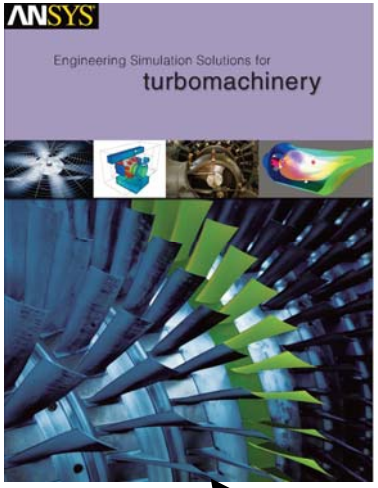




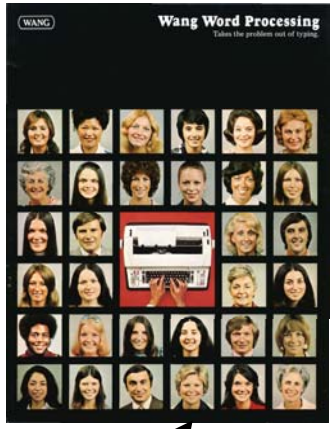
Games

WordPerfect 1.0 doc
Can you read it today?
100 years from now?





Application-specific content



Simulation model
Can you re-run old model with new data?

Original Wang doc
Can you read it today?
100 years from now?

Challenges

- Interpretation of bits
- Metadata capture
- Source or executable code
- “Digital X-ray”
- Capacity for BIG DATA
- Bankruptcies, sunsetting of apps, OS, hardware
- Intellectual Property Rights
- Legal frameworks, exceptions for preservation

The OLIVE Project

- Carnegie-Mellon University
- Mahadev Satyanarayanan (“Satya”)
- NSF funded project on digital preservation

Execution Fidelity

Ability to precisely reproduce execution

Many moving parts

- **hardware**
- **operating system**
- **dynamically linked libraries**
- **configuration parameters**
- **language settings**
- **time zone settings**
- **…**

Inspiration: "Digital X-Ray" of the hardware and operating software

Very difficult to achieve and then maintain

Transform into a Scaling Problem

Pack up and carry the entire environment with you

- **including the OS**
- **transitive closure of everything you need**

Central idea of a (hardware) *virtual machine (VM)*

But VMs are huge

- **many GB to tens of GB**
- **waiting to download → long launch delay**
- **inspiration from YouTube: *stream* instead of downloading**

VM Streaming Not So Easy

Access to VM image is not linear

Reference pattern depends on many runtime factors

- **data dependencies**
- **human interaction**
- **spatial and temporal locality (program behavior)**

Our approach

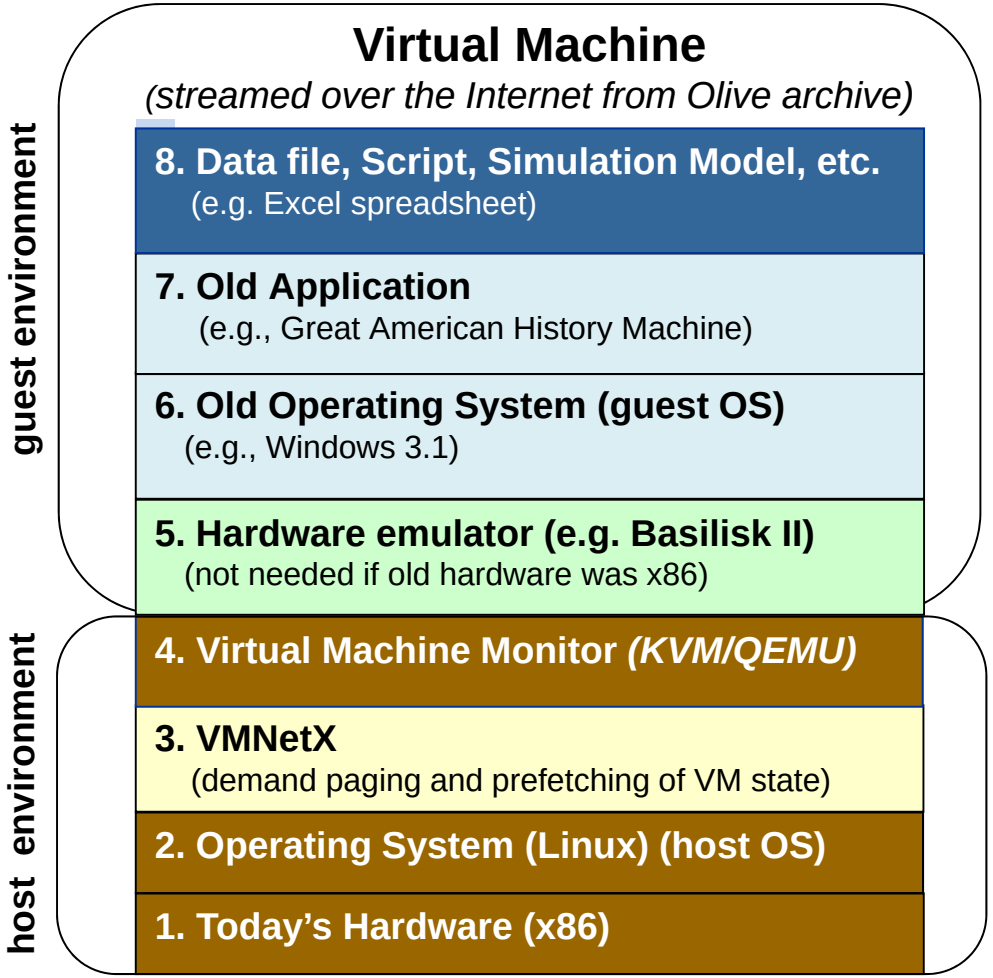
- ***demand paging***

  intercept missing VM pieces and fetch over Internet

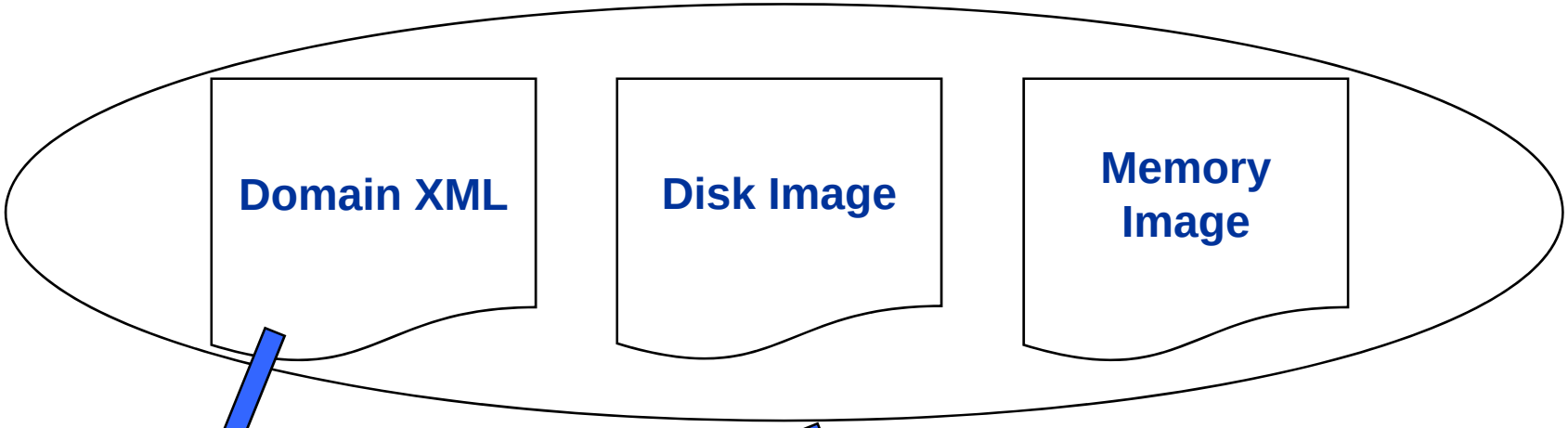
- ***prefetching***

  mask stalls due to demand misses
  (if hints are good)

Client Structure



VM Image Representation

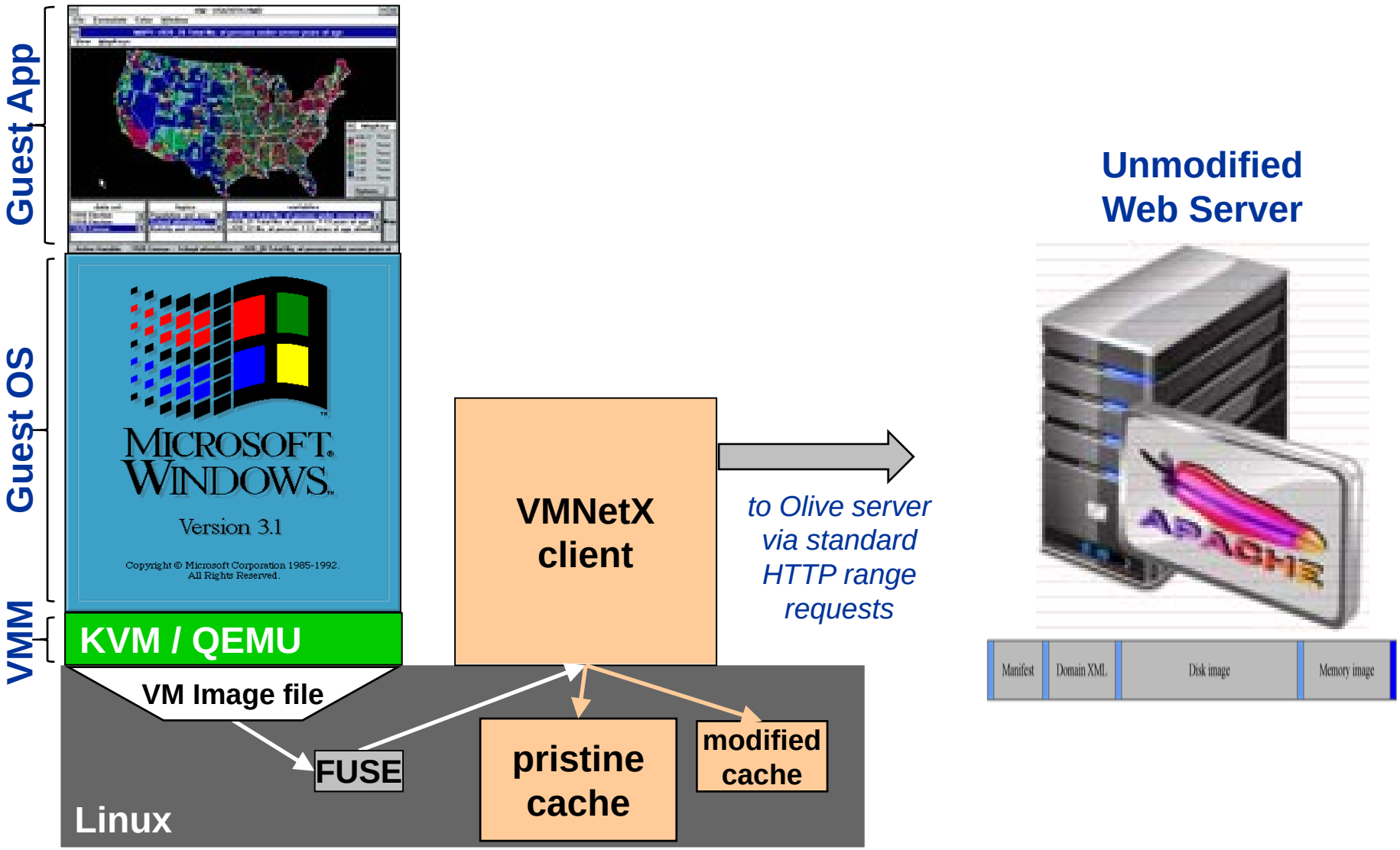


Machine details

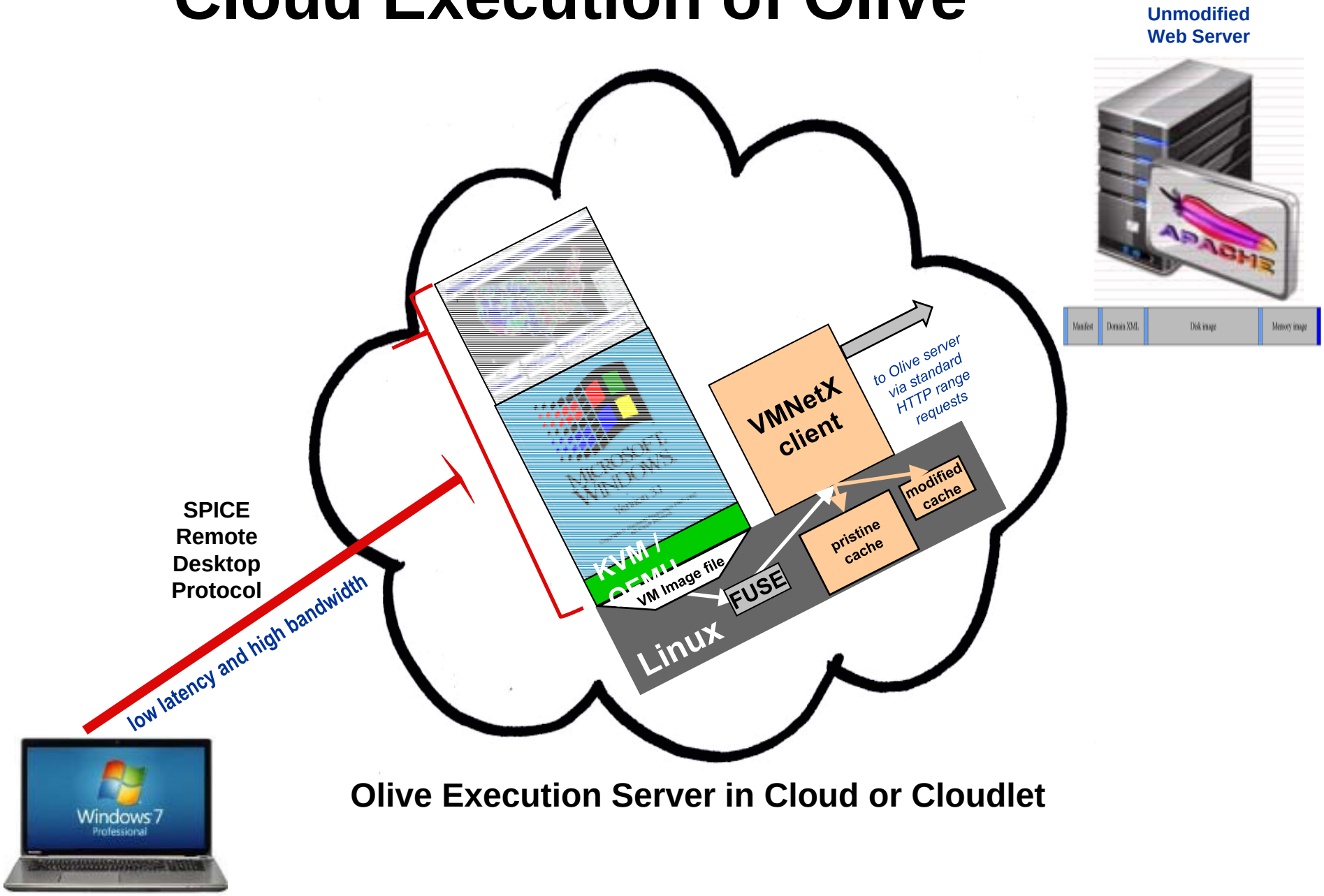
```
<domain type="kvm">
  <name>manual</name>
  <uuid>a7434757-631b-496d-a1ba-638014c74cc4</uuid>
  <memory>65536</memory>
  <currentMemory>65536</currentMemory>
  <vcpu>1</vcpu>
  <os>
    <type arch="i686" machine="pc">hvm</type>
    <boot dev="hd"/>
  </os>
  <features>
    <pae/>
  </features>
  <clock offset="utc"/>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type="file" device="disk">
      <driver name="qemu" type="raw"/>
      <source file="/disk.img"/>
      <target dev="hda" bus="ide"/>
      <address type="drive" controller="0" bus="0" unit="0"/>
    </disk>
    <controller type="ide" index="0">
      <address type="pci" domain="0x0000" bus="0x00" slot="0x01" function="0x1"/>
    </controller>
    <interface type="user">
      <mac address="52:54:00:03:a0:11"/>
      <address type="pci" domain="0x0000" bus="0x00" slot="0x03" function="0x0"/>
    </interface>
    <input type="mouse" bus="ps2"/>
    <graphics type="vnc" autoport="yes"/>
    <video>
      <model type="vga" vram="9216" heads="1"/>
      <address type="pci" domain="0x0000" bus="0x00" slot="0x02" function="0x0"/>
    </video>
  </devices>
</domain>
```

Single file representation

Olive Implementation



Cloud Execution of Olive



Olive Execution Server in Cloud or Cloudlet

Many Future Technical Challenges

We are a long way from being "done"!

Scaling and performance issues

- VMs keep getting bigger, networks are never fast enough
- clever prefetching techniques

Precise emulation of hardware

- even x86 extended memory modes not quite right in QEMU (can't boot Windows 95 in KVM/QEMU)
- exotic hardware platforms
- host compatibility (e.g. CPU flags in x86) vs performance
- hardware performance accelerators (e.g. GPUs)

Multi-VM ensembles (e.g. HPC environments)

Tools for easy building of VMs (physical to virtual?)

Archiving entire cloud services

… *many others* …

A Self-Archiving Web

Vint Cerf

Internet Lessons

- Collaborate and Cooperate
- Open design and evolution process
- Anyone can join: follow the protocols
- Multiple business models acceptable
- Modularity, layered evolution (info hiding)
- E pluribus Unum

Thinking Out Loud

- Compression schemes
- Tarball and related formats
- Only Storing and recovering particular objects
- Software storage/development w/versioning
  - Programmer's Workbench et al

But the WWW is a complex reference structure….

Brewster's Internet Archive

Static snapshot of the Web

- Time indexed, copies of web pages
- Hyperlinks re-formed to resolve to "in-Archive" pages
- Self-contained
- Requires continuous Web Crawl
- When to create a new instance of a page?
- …is there a role for RSS-like feed?
- Random: does the set of all sets contain itself...

Environmental Thoughts

- The Web can hardly contain itself – is there really room to replicate it (LOCKSS)?
- Hyperlinks deteriorate with time for many reasons – do we need a permanent link system (Digital Object Identifiers maybe?)
- HTMLx rendering challenge: backward compatibility
- Permissions, access control, copyrights?

Some Basic Thoughts

- Automatic, cooperative replication of created web pages (is this part of the editing process? Part of the “publication” process?)
- Is there a role for the Google Doc property of replication/real-time synchronization?
- Reference space held in common by the cooperating web-archiving entities?
- Think about Google sharding and Drive referencing.

More Thinking Out Loud

- Is there a role for a Pub/Sub mechanism for cooperating archive entities?
- A lot of metadata is needed to replicate Brewster's Time Machine feature
- A library of rendering/interpretation software is needed for the "leaves" of the web (ok, mixed metaphor)
- Whole subtext of permission to use here…

Surfing the S/A Web

- Multiple, alternative resolution targets (should not matter which is chosen if the system works right)
- Note: the static media of newspaper and magazines and books are snapshots of a “work” in time. We have “editions”. One can “see” the work as it existed at a given time. What’s the snapshot rate of S/A Web?

Some Desirable Properties?

- Automatic archiving on “publication”
- A service you sign up for? How is it funded?
- Registration of rendering engines (and permissioning system?)
- Automatic Malware Filters??
- Fidelity levels: everything works, surface display only (no links), other?

Additional Desirable Properties?

- Once archived, is a page an indelible and unalterable instance, useful in a court of law?
- Is there an “official records” side-effect of making the S/A Web work?
- Can this system work for encrypted content?
- Can access to parts of the archive be access controlled (e.g. release after 25 years)?

Is there a role for containers?

- Google is putting the entire Android Framework into a container, all the way down to the Hardware Abstraction Layer.
- It's a lot like virtualization, but a virtual machine would sandbox the apps away from other apps and the rest of the OS. Containers let the Android apps access the underlying OS, which allows them to communicate with each other. Connectivity, storage, audio, touch, and all the other inputs and outputs get connected to Chrome OS, allowing Android to interact with the outside world. Containerization is also designed to be lightweight—it doesn't use anywhere near the storage resources virtualization uses, since you aren't including a copy of the OS with every single app. The apps can all share a single Android app framework. Containers also don't need to allocate individual user storage for each instance.

Other Projects

- The Internet Archive – Brewster Kahle et al
  - Library of Alexandria backup among others
  - Digital content, books, software
- The Computer History Museum
  - Software and computing artifacts
- Google Book Scans and Cultural Institute
- Digital Object Architecture and Identifiers (CNRI)