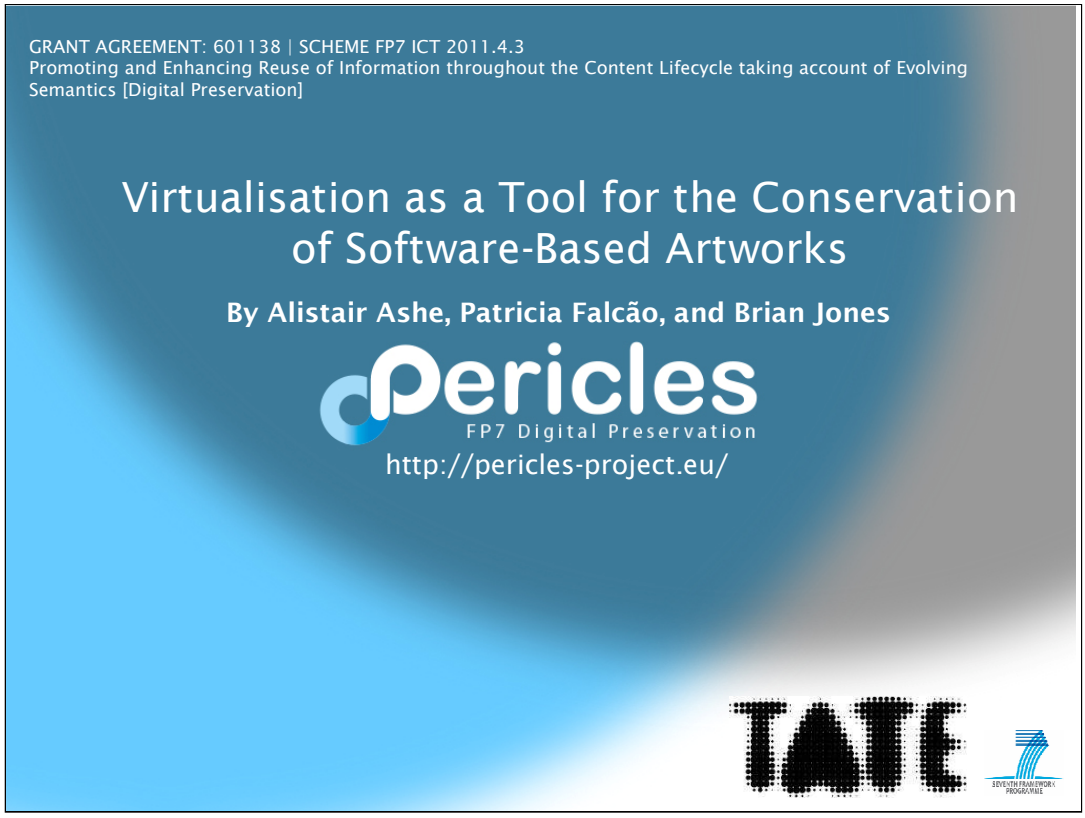


Hello, good morning all,
In this paper we will present the results of our research project, in which we tested the use of virtualisation for the preservation of software-based art.
The project was developed under the umbrella of Pericles, an European funded project addressing the impact of change in the long-term preservation of digital objects.
Software-based artworks are one of the use cases that Tate is bringing to Pericles, looking at technological change.
Our project lasted 20 days and involved 3 Departments; Research, Conservation and IT.
We were a team of 3, Alistair and myself from Conservation and Brian Jones from the IT department, who has the practical expertise in terms of virtualisation.
The aim of the project was to test whether virtualisation would be an appropriate tool for preservation, and whether it would be possible to implement it as part of the preservation workflow for works in the collection.

The presentation has 2 parts, first I will give you the context in which we are working, the Institution, the Collection and the Artworks. I will speak briefly on why we chose virtualisation and then I will handover to Alistair, who will speak about the project itself and the resulting workflow. Our best outcome for this presentation would be to have comments on our workflow, as we are about to start implementing it.

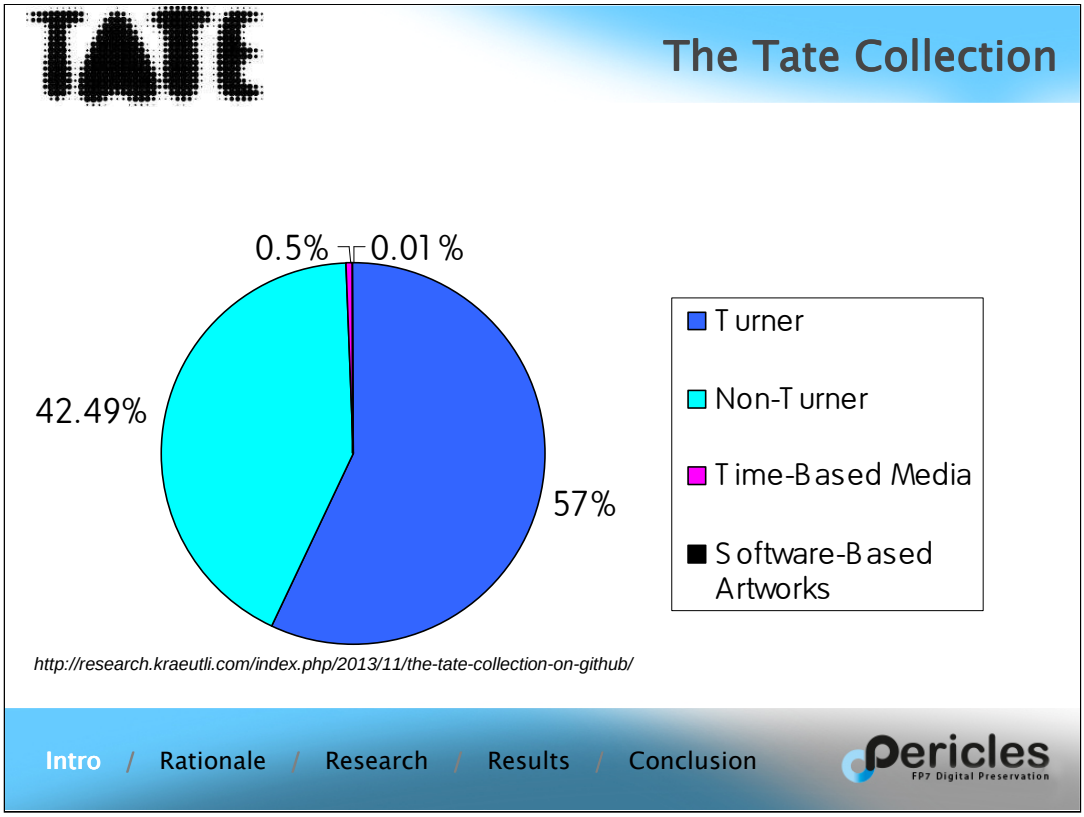
Tate is composed of 4 Galleries, Tate Britain and Tate Modern in London, Tate Liverpool in the North and Tate St. Ives in the South West. The Collection as well as the 3 Departments involved in the project are based in London.

The Research department supports research about the Art Collection, including the development of new strategies for its care. In this case we made a proposal for this project, which was selected as relevant by the Research Department. They then provided a framework to develop the project within Pericles and in collaboration with the IT Department and Pericles Partners.

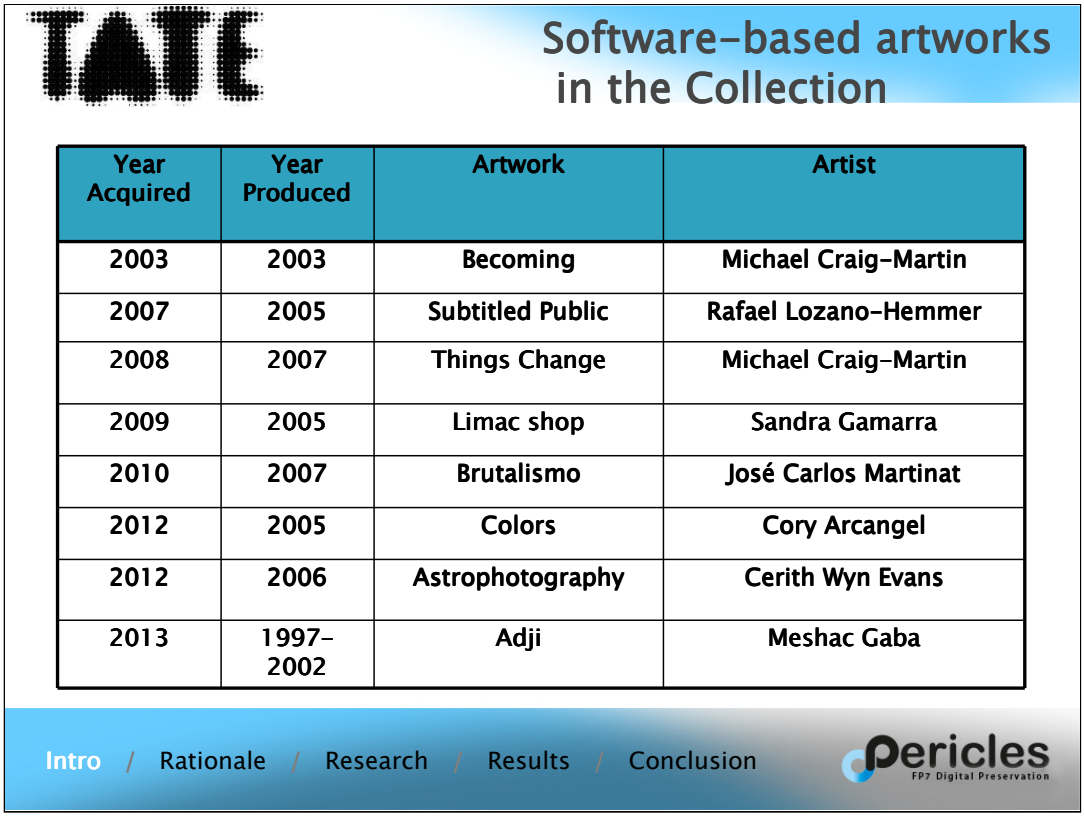
Conservation, is responsible for the long-term preservation of the Collection. Our team, Time-based media conservation, specialises in the preservation and display of film, video, slide, audio and software-based art, as well as performance. The team is composed of 4 Conservators and 5 Technicians, which preserve and install all the time-based media artworks across the different Tate sites.

Last but definitely not least the IT department is responsible for the whole IT infrastructure at Tate. It was the first time we collaborated with them in a conservation project, as before we tended to use external specialists. That the IT department made collaboration with research and conservation a priority is a big change in the institutional tradition, and it was a good experience, with much learning at all ends.

And now a little about the Collection:



The Collection contains about 70,000 artworks, British Art between 1500 and the present, and both British and International Contemporary Art. This diagram is an adaptation of one created by Florian Kraeutli, based on Collection Data published online at GitHub.
Tate‘s Collection is not very balanced, as you can see.... 56% of the collection are artworks by JW Turner.
As you can see from the diagram, Time-based media- so, film, video, slide, audio and software-based artworks make about 0.5% of the Collection, and specifically software-based artworks are only 0,001%.

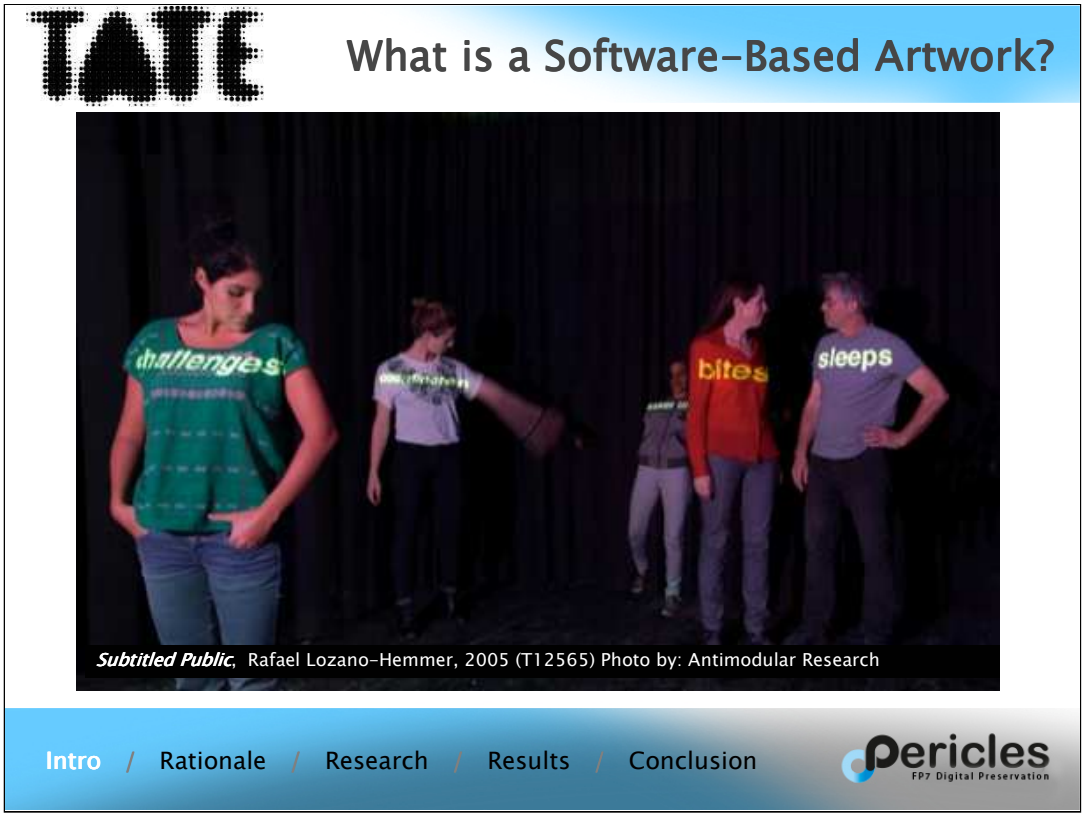


Those 0,001% translate to the 8 software-based artworks shown in this table. These works, even if they are not many, cover a spectrum of types. From one small screen displaying images through and executable running on Windows XP to a large installation including a network of computers tracking visitors in a dark gallery space and projecting images on them, there is quite a variety.

As you can see from the table, Tate's earliest software-based artwork was created in 2003 and after 10 years the issues around aging technologies are becoming more obvious and new strategies for preservation are more urgently needed. The number of artworks being acquired and displayed is increasing and therefore better workflows must be developed to accommodate this increase.

Eight may seem like a small number of artworks to preserve. Yet currently the amount of resources needed to preserve one software-based artwork is much higher than the resource needed for an average TiBM artwork. This is due to the technical complexity of the artworks, their uniqueness, but also because the workflows required are not fully established.

Now let's see what I mean with software-based artwork:



We define as software-based any artwork for which software is the core medium.

In the majority of cases the artist did not programme them themselves, but have worked with programmers, to which they tend to "remain faithful". The exception in the Collection is *Colors* (2005), by Cory Arcangel, an artist who does his own programming. Some of the actions performed include: analysing video, mapping the location of visitors in a gallery, or displaying randomly composed tableaux of vector images.

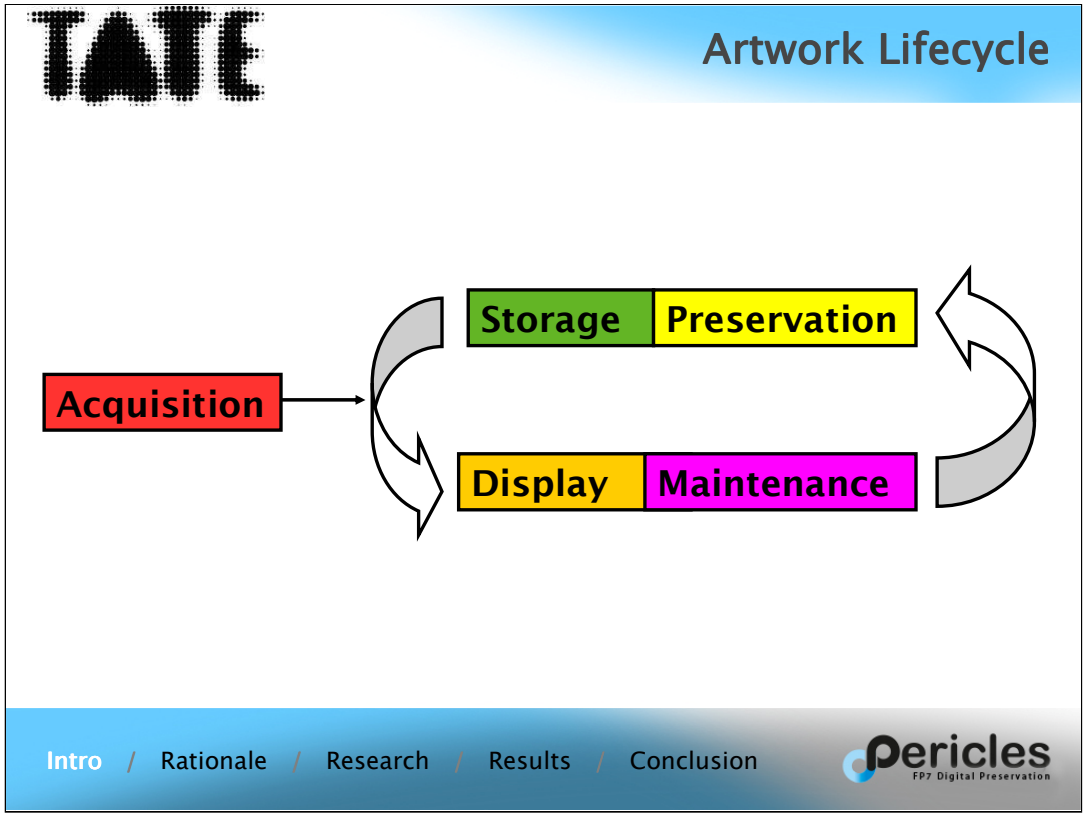
Software-based artworks are usually supplied to Tate on a computer, with the software installed and with any peripheral needed (except projectors). A package ready for display in the galleries.

The works use different Hardware, Operating Systems (Windows XP, Mac OSX and Ubuntu), and the code itself is often bespoke and written in different programming languages, from Java to Delphi.

The peripherals required include, among others, a Morse translator, video cameras and till printers.

Pip Laurenson refers that Software-base artworks are often like prototypes, where the artist/programmer are developing a system from scratch, and that many adjustments are made when the work is displayed, either to correct bugs that show up over time, or to adapt it to work in that specific situation. My experience is that when an artwork is installed and the artist is involved there is a high probably that changes will be made to the software.

I will talk about the artwork's lifecycle in the collection next



This Diagram shows the lifecycle of an artwork in the collection

The whole process of acquiring the artwork into the collection starts before the Acquisition stage depicted in the diagram, with often lenghty discussions between the curators and artists or galleries. Conservation is involved just before acquisition to create a pre-acquisition condition report where we identify what the artwork is, and what materials we would expect to receive, the main risks for preservation and the resource implications of acquiring and displaying the work. To be able to produce this report we need to understand te production process of the artwork, and so usually there is a period of intense communication with either the artist, their studios and technicians or commercial galleries.
When the acquisition is agreed we contact the seller/donor for delivery of what is required, and on arrival we condition check whichever media and equipment is delivered, to ensure that everything is working as expected. For software-based artworks we usually create a copy of the system supplied at acquisition. We found that doing this will usually highlight any specificities of the system that are not obvious if you have a running system. An example are particular libraries that the artwork‘s software may require but that are not installed by default in the Operation System in use.

The work will then either go into storage or be displayed. While in storage we will carry out preservation steps that may be required, but currently for software-based artworks that amounts mostly to backing up files and maintaining the equipment. The use of virtualisation will mean that new processes will be required, probably carried out in conjuction with out IT department.
The Installation/Display moment is particularly important, because often

When I first came across the concept of Significant Property I thought that it was very powerful, and so I was keen to use it. I believe this was the initial definition by InSPECT. But I had trouble to use them directly to an artwork, as the Significant Properties of an artwork must be closely related to the artistic intent. For Instance, for both works in our case studies the artists were happy for us use a different software to obtain the same results. So re-coding would be an option for preservation. For some artists the code is the art, and in those cases the only option is to find ways to run the original software.

At Tate we are keen to keep the technical history of the production of the artwork, but that must go hand by and with being able to display the work in the galleries. So a degree of change is to be expected and our systems and processes must accommodate for that as well.

I felt that, because very often out digital objects are not the artwork itself we needed to change that definition to accommodate this. The following was my suggestion.

Because what we are trying to keep is the artwork and its properties and not just the digital media that it composes of.

For Subtitled Public, for instance, we may have a perfectly preserved version of the software, but if we don't have the information about the conditions required in the gallery, or the requirements for the equipment needed to display it we will not be making the artwork justice.

In the next few slides we will look at the example from one of our case studies:

This work is “Becoming”, by Michael Craig-Martin, from 2003. It is one of our Case Studies.

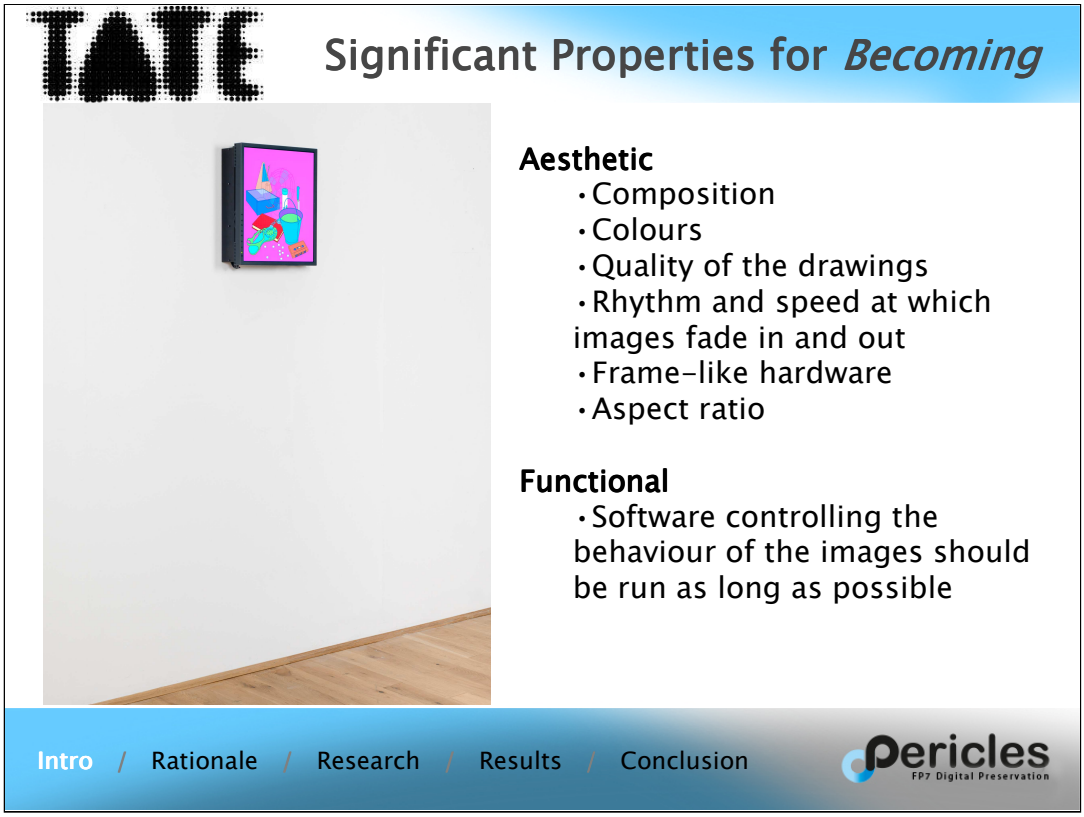
It is a computer-generated animation playing on a wall-mounted (LCD) screen with an in-built computer. The monitor has a simple black frame, which has been professionally re-sprayed to render it anonymous in appearance. A series of Craig-Martin’s elementary line drawings, vividly coloured, fade in and out of visibility against the fuchsia pink background. At any one time, eighteen objects - a chair, a pair of pliers, a tape cassette, a fan, a pitchfork, a sandal, a light bulb, a drawer, a metronome, a book, a bucket, a TV, a flashlight, a safety-pin, a knife, a pair of handcuffs and a medicine jar spilling pills – may appear onscreen. For this project, AVCO developed a programme that generates the random appearance and disappearance of the objects.

As part of the acquisition process we discussed the preservation options with the artist, which helped us understand what he feels is relevant about the work and therefore what its significant properties are. In terms of it’s preservation the artist said that it would be fine to use different equipment, but that the work must be shown as a framed screen on the wall, like a painting.

The size of the screen could be different, if necessary, but if a change of aspect ratio were necessary we would want to adjust the composition of the images on the screen

The artist is not attached to the software and is happy to change it as long as the rhythm and speed at which the images appear is maintained.

As part of the preservation strategy a second set of computer and screen was built. Because the computer specifications were different we requested the artist’s programmer to write a piece of software to measure the speed at which the images appear and disappear on the screen.



From this process we drew a list of the significant properties for the work:

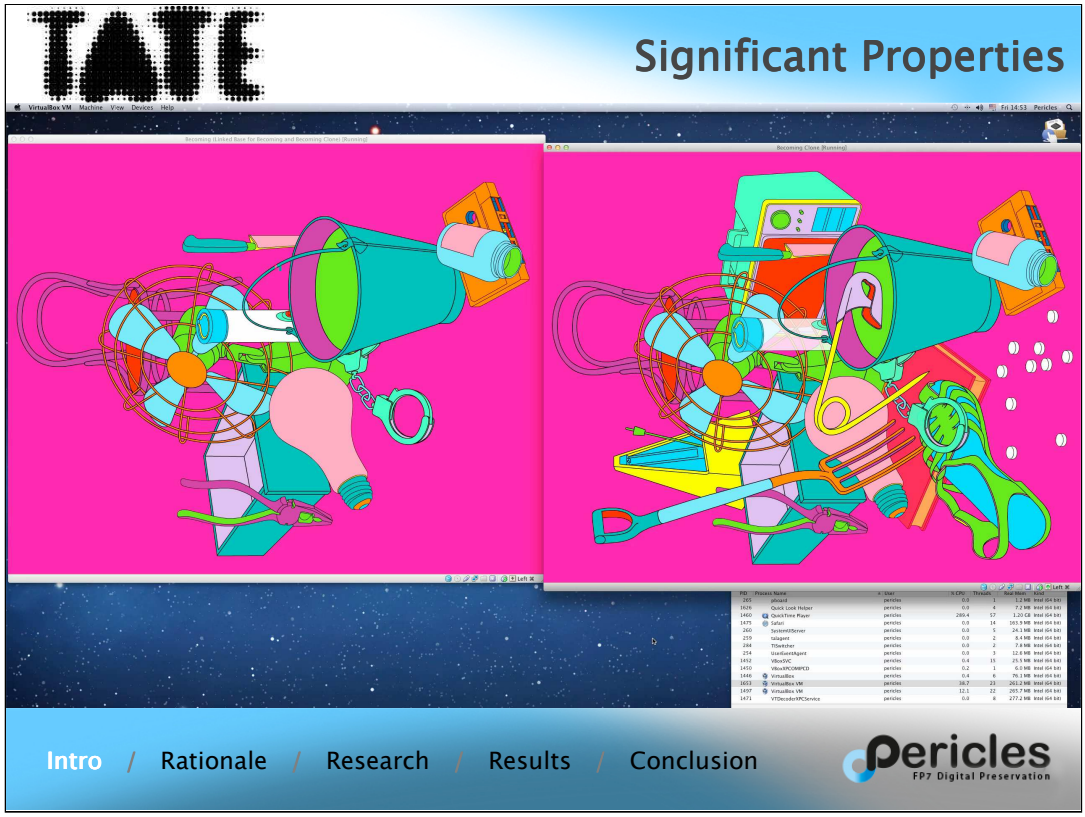
Aesthetic

- Composition
- Colours
- Quality of the drawings
- Rhythm and speed at which the images fade in and out
- Frame-like hardware
- Aspect ratio

Functional

- Algorithm controlling the behaviour of the images should be maintained as long as possible

On the next slide you see an example:



In this slide we have a video of 2 virtual machines running, with the settings changed to illustrate the correct playback speed (right) and the software running on a much faster processor (left).

As mentioned, the significant properties will vary with the artwork, and are very closely related to the artist's intent. The same property can be significant or not depending on the value an artist attributes to that particular property. Defining them and finding ways of evaluating these properties in both the physical and virtual machines is in our opinion, the main challenge and the most important step in the process.

As I described before, Daniel Jackson from AVCO, designed a software tool to measure the speed. For this one work this tool provides a concrete way to measure this one significant property of the work. However, this tool is specific to this artwork, and it is unlikely that its usefulness would be applicable to a different work. This indicates that there may be the need to write specific tools for other artworks as well. How to measure quantifiable significant properties is one further strand of research that we need to develop.

We decided to test virtualisation for a number of reasons, that I will explain.

Our aim is to be create an alternative way to run the software, that is not dependent on the original hardware.

Eventually the original hardware will fail, and by having a virtualised version we may either be able to run the software from the virtualisation or we can use the virtualised version to compare any new versions made and ensure that the significant properties are the same.

We defined that the virtualisation needed to:

run the artwork's software in a form as similar as possible to the original hardware/software combination.

maintain the significant properties of the artwork unchanged –

support the use of the different peripherals that are required to display the work in the form devised by the artist.

In practical terms there were a series of advantages to virtualisation:

We have the infrastructure and expertise in-house

All our artworks use systems based on the x86 architecture, and the oldest operating system is Windows XP, so the virtualisation tools available fully support them.

The resulting virtual machines can be saved as files in Tate's Digital Repository

Limitations:

VMWare will only support systems based on the x86 architecture, so if we acquire any earlier software we will need to look at emulation instead.

We chose 2 artworks, that you can see here. With both:

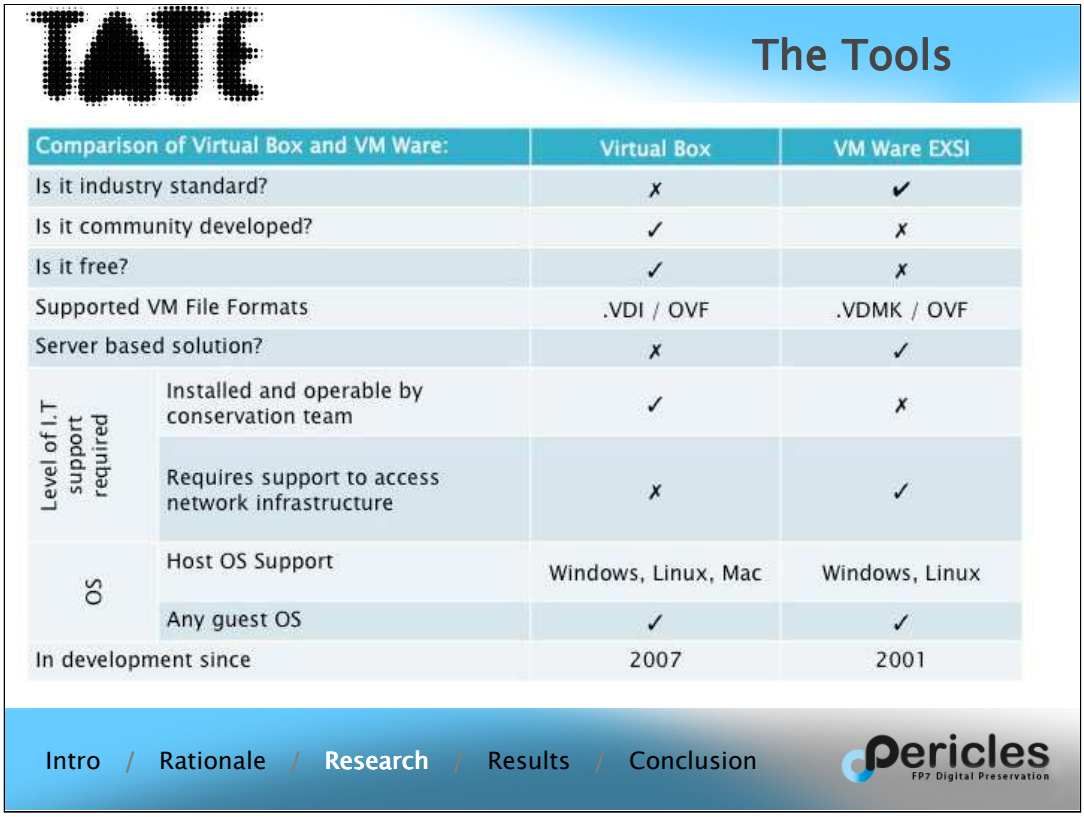
- software interlinked with artistic intention
- Software forms main significant properties
- Because of their difference, they represent a sample of complexity

Becoming, by Michael Craig-Martin

- 18" LCD screen with in-built computer
- Becoming presents eighteen vividly coloured line drawings
- fading randomly, and slowly
- Uses Windows XP, that will stop being supported by Microsoft as of coming April
- The work should be experienced as if it were a painting.

Brutalismo, by Jose Carlos Martinat Mendoza:

- model of the Peruvian Army Secret Service building in Lima
- incorporates a computer and four printers that are placed on top of the sculpture, and visitors can take away printouts.
- software runs on Ubuntu Linux
- works by searching the internet for phrases
- stores the results of these searches in a database
- prints the results accessed from the database



As part of our workflow, we use two tools that we tested as part of research – Virtual Box and VMWare EXSI.

We evaluated these tools:
•Sustainability
•Table shows some parameters

For example - widespread use
•Mitigate against obscelscence
•Open virtual machine format (ovf) support – not dependant
Both tools widespread and supported the ovf format.

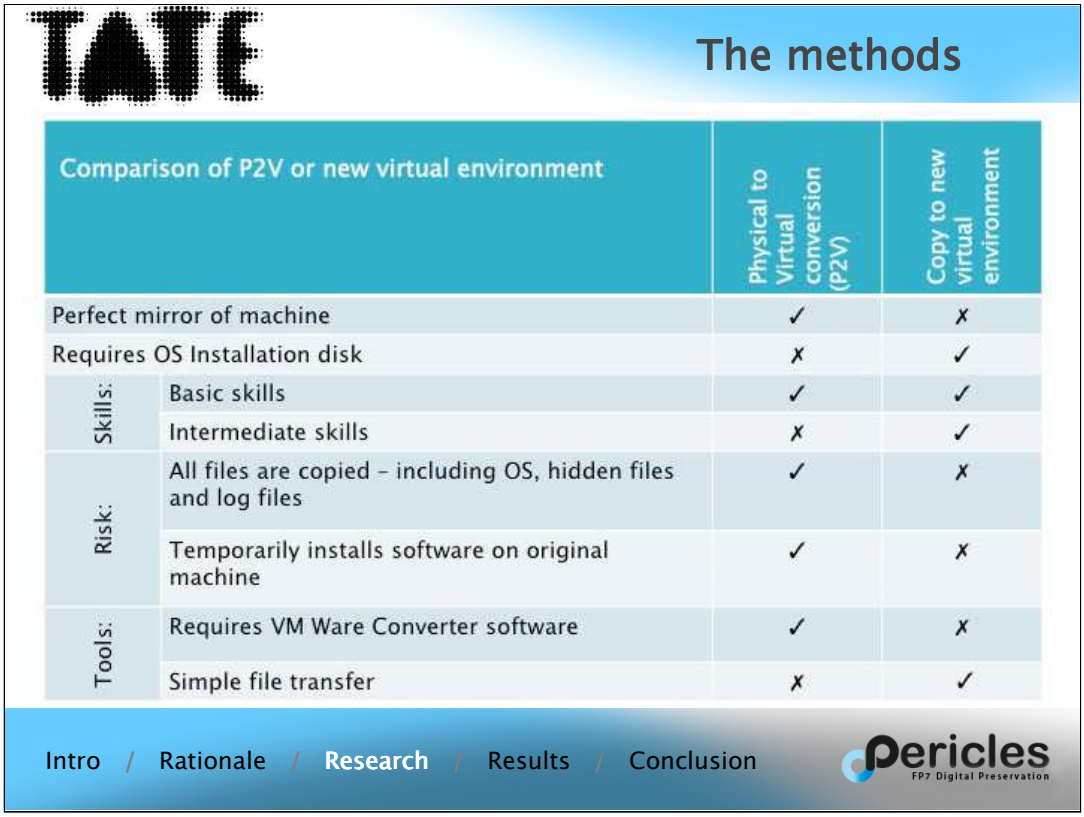
VM Ware EXSI is:
•An industry standard, server based solution
•Tate Information System's preferred solution
•Which makes it our main tool, because:

- Expertise
- Infrastructure
- And existing protocols, like upgrading and maintaining virtualisation infrastructure, not parallel system for conservation

Virtual Box has some other advantages:
•It is client based
•Easier to use
•And can be run on Mac, and run Mac virtual machines

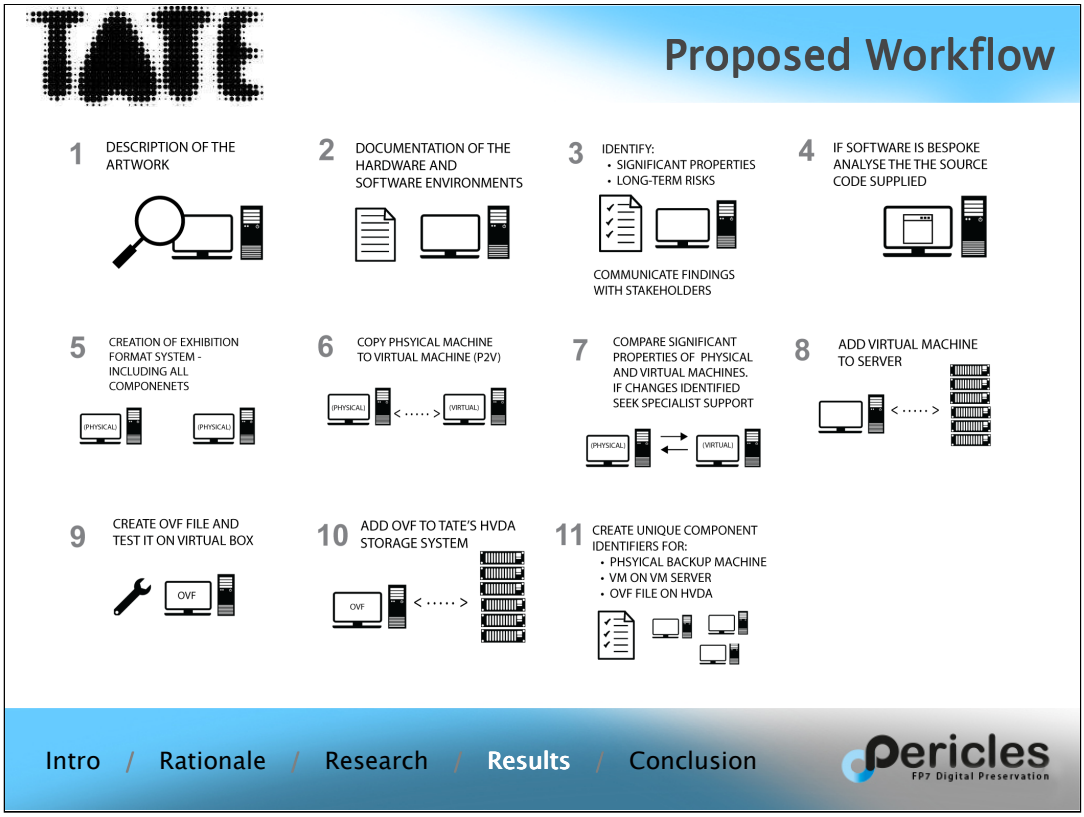
Both tools limited - Mac OS:
•limitation of the OS - it must be run on a Mac host computer.
•Because Virtual Box available for Mac OS, it can run Mac vms.
•VM Ware can store, but not run Mac OS virtual machines, because it is only available for windows and Linux.

Workarounds are needed for this, but we could not test them for this project.



As part of our research, we tested 2 methods

- p2v conversion:
  -
  -

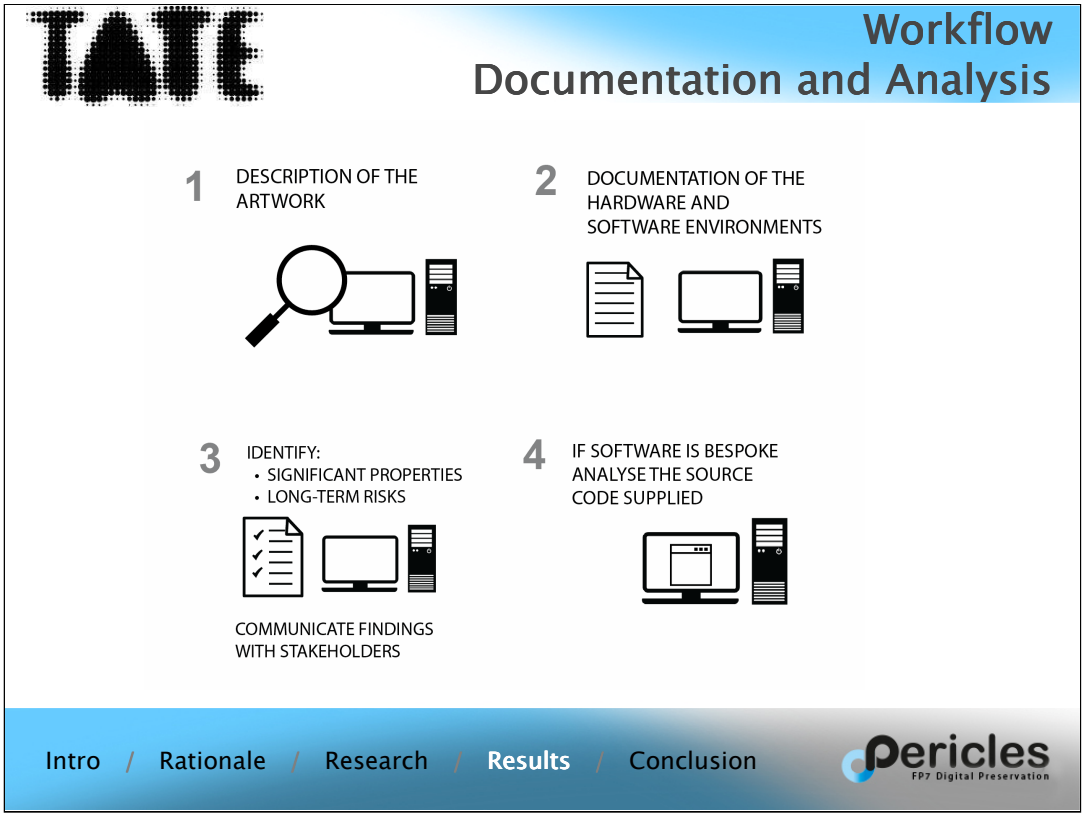


Proposed workflow as result of our project

Stages documentation and analysis / duplication and diagnostics / storing and tracking

Developed to compliment existing systems and workflows of the institution.

Importantly, each stage is not completely distinct, and each refers to one another and are often conducted at the same time, roughly in the wider stages presented.



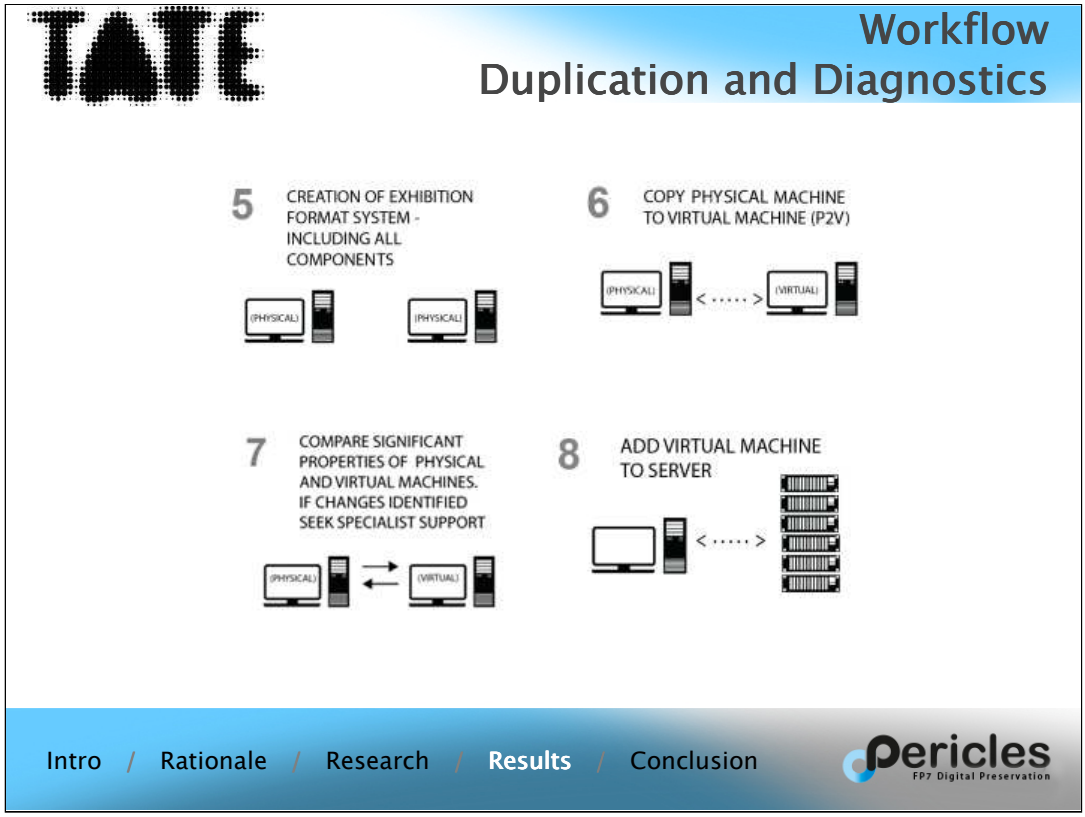
The documentation and analysis phase of the proposed workflow….

Step 1, involves locating and checking updating the software based artwork's technical, and other contact with the work's suppliers and producers about the intended effects and experience of the artwork.

Step 2, involves the checking and updating of the known technical requirements and technical history of the artwork, both in respects of it's software and hardware environments. This is to include, where appropriate, video documentation of the artwork running to capture the experience of the artwork, and interviews with the artist and artists' programmer about the process of developing the solution.

Working from the generated and existing information about the work, in stage 3 the significant properties and long term risks of the software based artwork are identified . This may require further communication with experts and the artist again. These finding should be communicated to the project stakeholders.

Stage 4 is a delve into the commented source code of the artwork by someone other than the original programmer in order to discern further, from the perspective of it's coding, how the software works. This step also plays a role in confirming the information produced by the other stages, and generates an understanding of how the software works in relation to artistic intention, but also any problems or risks identifiable within the source code itself, including

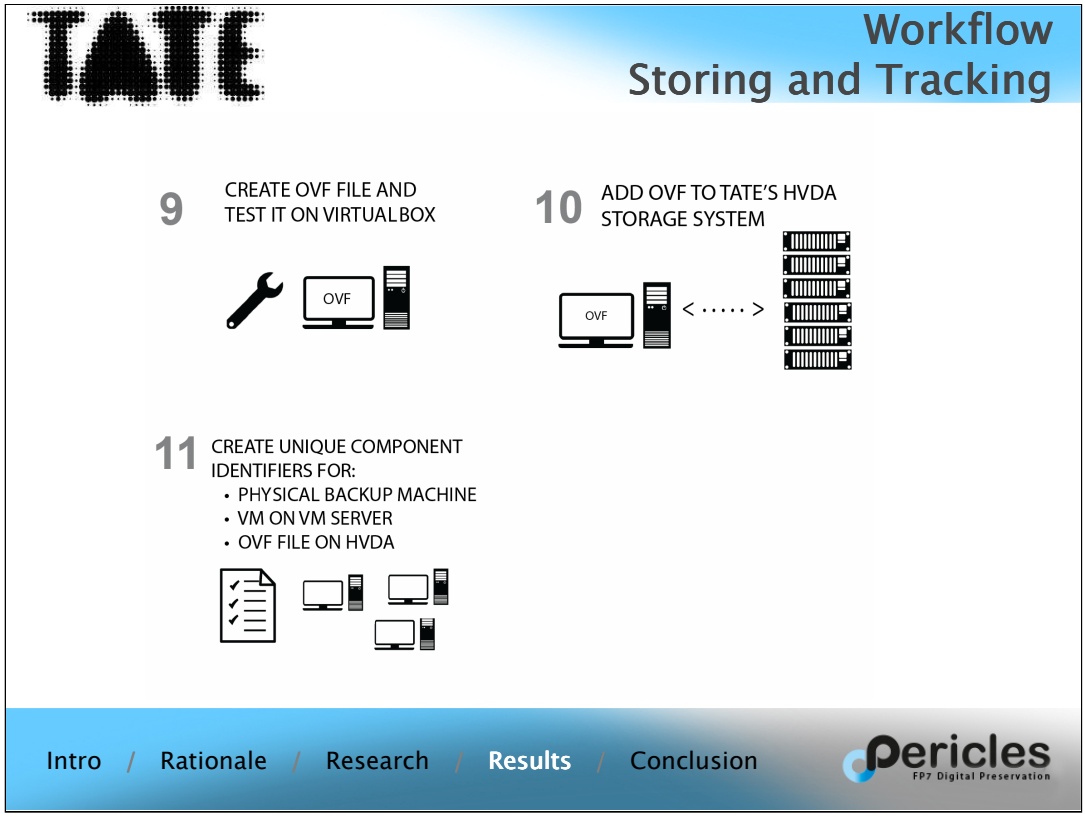


The next phase of the proposed workflow is covers duplication and diagnostics.

Stage 5, follows the existing protocol for software based artworks whereby the entire software based artwork's physical machine supplied to Tate is duplicated on a new machine to match the original's specifications as much as possible (including peripherals and consumables). These duplicate physical machines are then used for exhibition to limit the risk of wear and tear on the original unit. Interestingly, this process normally throws up additional knowledge about the software components and their compatibility, in turn contributing again to the documentation of the work, for If any extra software must be added to the physical computer (e.g. libraries or drivers for particular printers) then this should also be made a documented component of the artwork.

Stage 6 is the virtualization of the software based artwork's computer by means of physical to virtual virtualization. This involves connecting the software based artwork's machine to a pc over a local network. VMWare's converter software will make a virtual machine file. Importantly, we have also imagined a process whereby the virtualization process will be conducted from a disk image of the software based artwork's machine, because VMWare converter temporarily installs a small piece of software of the machine to be copied. In addition, currently our exact process only supports PC and Linux virtual machines.

Stage 7 is a comparison between the software based artwork running on its supplied hardware and the virtualized version. The first time the virtualised machine is opened, it will require the automated installation of virtual drivers for display etc to replace the hardware previously installed. Specialist technical support is likely to be required to identify the less visible differences between the physical and virtual machines. A very obvious example of the kinds of fin

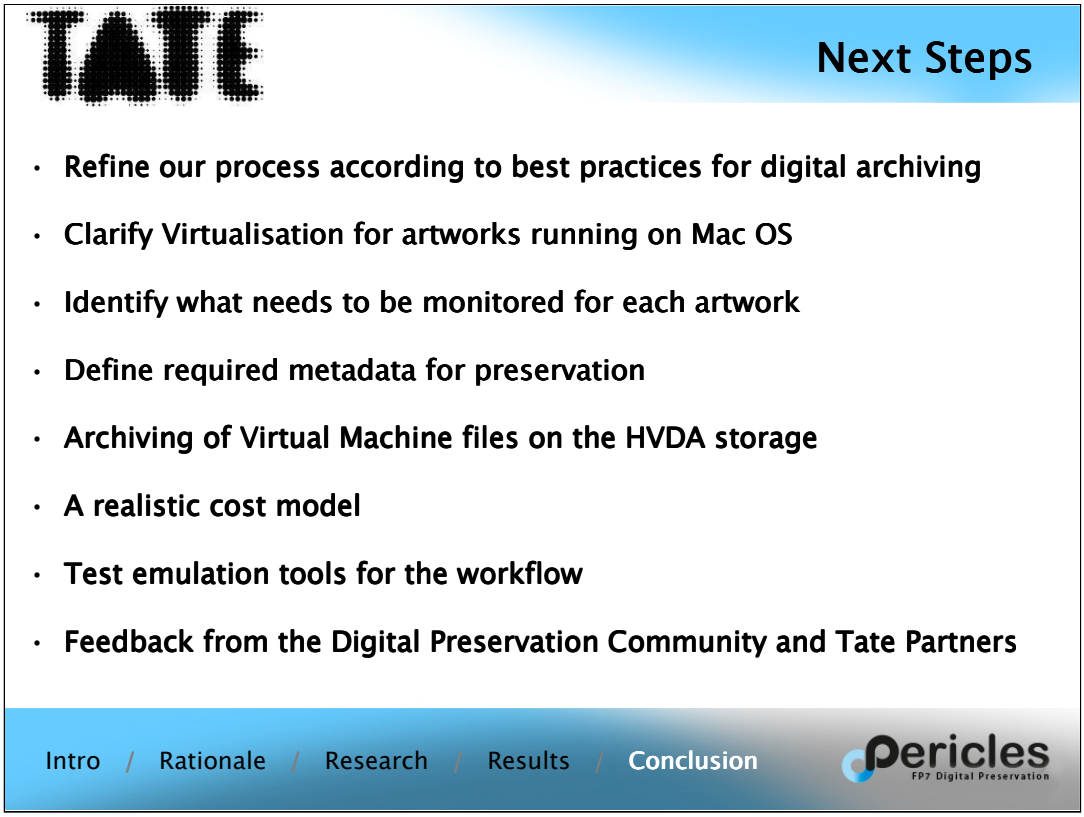


The final phase is the storing and tracking phase.

Stage 9, involves the creation of an ovf (open virtualisation format) virtual machine. Testing the virtual machine file can be conducted via virtual box in accordance with the significant properties outlined earlier to make sure that these are maintainable.

Stage 10, is to add the file to Tate's High Value Digital Asset (HVDA) storage system, as Patrica described earlier.

The final stage, stage 11, is to ensure that all the new elements have unique component identifiers and added to Tate's collection management system called "The Museum System" or TMS for short. Component numbers need to be created, and continue to be updated based on condition and location, for the physical backup machine, the virtualised version on Tate's virtual machine server, the OVF virtual machine file on the High Value Digital asset server.



We have found a strategy that would be applicable to possibly all SBA artworks in the Collection

We would need to develop a plan by:

Refining the process of producing virtual machines – with some input from some of Tate Partners and people from the Digital Preservation community

Look at issues with Mac OS.

Identify what needs to be monitored for each artwork to identify if significant changes happen when virtualising

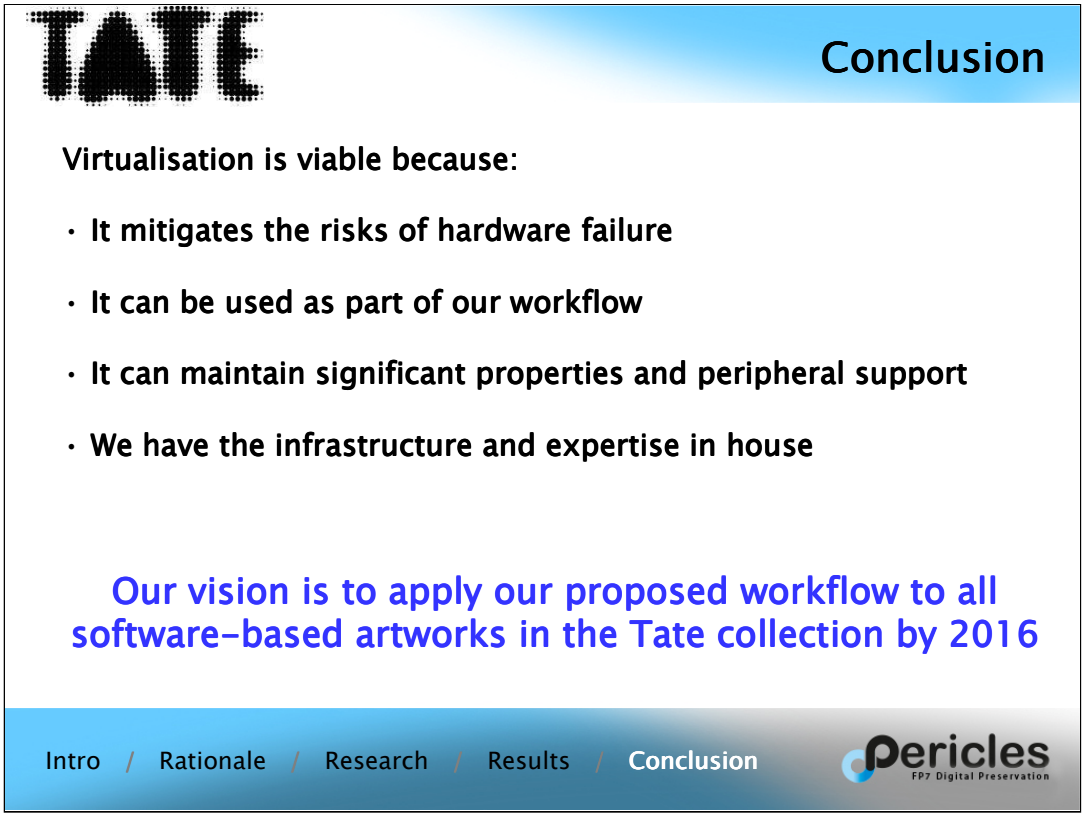
Define the Metadata required for preservation, and whether it can be extracted automatically

See how we could save virtual machine files in the HVDA storage, what would be required

Creating a realistic cost model

As we mentioned earlier, for now virtualisation will not work for systems that are not x86 architecture. Also, it is likely that we will buy earlier software that is not x86, this is what we are seeing for video and film. Therefore in future we may need to explore emulation, for example using tools like **bwfla and Qemu**. However, our use case means that we will be looking for tools that fit within our existing systems, workflows, and expertise, like virtualisation does in the way we have outlined.

Continue to discuss these issues, and learning from, the Digital Preservation Community and Tate Partners like MoMA and SFMoMA.



(Our vision includes collection works which may have been duplicated but not yet virtualised).

The project identified virtualisation as a step towards a viable strategy for the preservation of our software-based artworks. Virtual machines will also in turn become obsolete. It may be that the virtual machines can be migrated, or that alternative strategies may be developed in the future to keep the software-based artwork operational. As with any digital object, preservation will need to involve the active monitoring and management of material to ensure that it remains accessible.

Virtualisation provides a complete environment within which the software runs, this enables comparison with our original systems; making it possible to check that they behave in the same way. An important aspect of this strategy is creating a virtualised version of the work, whilst the original can be confirmed as still running correctly.

Each software-based artwork is different, and it is an important aspect of the challenge of conservation to identify the significant properties of a particular artwork. Finding the best way to compare physical and virtual machines will also have to be decided on a case by case basis.

Conclusions:

(other points)

Discussion –

Feedback

X86 architecture

Mac OS