netlabs.org - The Voyager Project

A Workplace for the 21th Century

Adrian Gschwend

netlabs.org - Open Source Software for OS/2 and eCS

Warpstock Europe 2006, Cologne, Germany

Outline

1. History
2. The Voyager Project
3. Voyager Components
4. Roadmap

Warning

This is not a very technical presentation!
(Please don’t leave the room now ;-)

One Year Ago

After last years presentation:

- Joy
- Shock
- *Why Voyager, my eComStation works just fine!?*

One Year Ago

After last years presentation:

- Joy
- Shock
- *Why Voyager, my eComStation works just fine!?*

One Year Ago

After last years presentation:

- Joy
- Shock
- *Why Voyager, my eComStation works just fine!?*

Motivation, Hardware

eCS and OS/2 works just fine ^{TM}

- Might be true nowadays
- Might be hidden for many users
- We (the developers) know the technical limitations
- Annoying kernel limitations and bugs
- Hardware evolves

Motivation, Hardware

eCS and OS/2 works just fine ^{TM}

- Might be true nowadays
- Might be hidden for many users
- We (the developers) know the technical limitations
- Annoying kernel limitations and bugs
- Hardware evolves

Motivation, Hardware

eCS and OS/2 works just fine ^{TM}

- Might be true nowadays
- Might be hidden for many users
- We (the developers) know the technical limitations
- Annoying kernel limitations and bugs
- Hardware evolves

Motivation, Hardware

eCS and OS/2 works just fine TM

- Might be true nowadays
- Might be hidden for many users
- We (the developers) know the technical limitations
- Annoying kernel limitations and bugs
- Hardware evolves

Motivation, Hardware

eCS and OS/2 works just fine $^{\mathrm{TM}}$

- Might be true nowadays
- Might be hidden for many users
- We (the developers) know the technical limitations
- Annoying kernel limitations and bugs
- Hardware evolves

Motivation, Other Issues

It’s not just about the hardware. . .

- The valuable stuff is done by the same few people
- Motivation is not what it used to be for various reasons
- It is annoying and frustrating to
  - write device drivers
  - code around bugs
  - port software and toolkits
  - know that the work will be useless in a few years
- Almost no new people are joining the community

Motivation, Other Issues

It’s not just about the hardware…

- The valuable stuff is done by the same few people
- Motivation is not what it used to be for various reasons
- It is annoying and frustrating to
  - write device drivers
  - code around bugs
  - port software and toolkits
  - know that the work will be useless in a few years
- Almost no new people are joining the community

Motivation, Other Issues

It's not just about the hardware…

- The valuable stuff is done by the same few people
- Motivation is not what it used to be for various reasons
- It is annoying and frustrating to
    - write device drivers
    - code around bugs
    - port software and toolkits
    - know that the work will be useless in a few years
- Almost no new people are joining the community

Motivation, Other Issues

It’s not just about the hardware…

- The valuable stuff is done by the same few people
- Motivation is not what it used to be for various reasons
- It is annoying and frustrating to
  - write device drivers
  - code around bugs
  - port software and toolkits
  - know that the work will be useless in a few years
- Almost no new people are joining the community

Motivation, Other Issues

It’s not just about the hardware...

- The valuable stuff is done by the same few people
- Motivation is not what it used to be for various reasons
- It is annoying and frustrating to
    - write device drivers
    - code around bugs
    - port software and toolkits
    - know that the work will be useless in a few years
- Almost no new people are joining the community

Motivation, Other Issues

It’s not just about the hardware…

- The valuable stuff is done by the same few people
- Motivation is not what it used to be for various reasons
- It is annoying and frustrating to
  - write device drivers
  - code around bugs
  - port software and toolkits
  - know that the work will be useless in a few years
- Almost no new people are joining the community

Motivation, Other Issues

It’s not just about the hardware. . .

- The valuable stuff is done by the same few people
- Motivation is not what it used to be for various reasons
- It is annoying and frustrating to
  - write device drivers
  - code around bugs
  - port software and toolkits
  - know that the work will be useless in a few years
- Almost no new people are joining the community

Motivation, Other Issues

It's not just about the hardware...

- The valuable stuff is done by the same few people
- Motivation is not what it used to be for various reasons
- It is annoying and frustrating to
  - write device drivers
  - code around bugs
  - port software and toolkits
  - know that the work will be useless in a few years
- Almost no new people are joining the community

Conclusion

Conclusion

If we do *business as usual* it will not go on one day.

Wouldn’t it be nice?

We can now either fall into depression or come up with some idea.

Why not

- rewrite what we like?
- profit from existing Open Source Software?
- *do it right* ourself?
- start the whole idea on eComStation?

Thus the idea of Voyager was born!

Wouldn’t it be nice?

We can now either fall into depression or come up with some idea.
Why not

- rewrite what we like?
- profit from existing Open Source Software?
- *do it right* ourself?
- start the whole idea on eComStation?

Thus the idea of Voyager was born!

Wouldn't it be nice?

We can now either fall into depression or come up with some idea.
Why not

- rewrite what we like?
- profit from existing Open Source Software?
- *do it right* ourself?
- start the whole idea on eComStation?

Thus the idea of Voyager was born!

Wouldn’t it be nice?

We can now either fall into depression or come up with some idea.
Why not

- rewrite what we like?
- profit from existing Open Source Software?
- *do it right* ourself?
- start the whole idea on eComStation?

Thus the idea of Voyager was born!

The Idea I

The Story so Far...

- Long process of thinking about the future for several years
- First idea with Kernel of MacOS X in Summer 2004
- First presentation of that idea at Developers Workshop 2005 in Dresden
- Reconsideration of this idea because it doesn't solve the main problem: Desktop

The Idea II

- New idea with OpenGL based Desktop with well known toolkits, developed at SYSTEMS fair in Munich
- Talks to various people and first presentation of that idea at Warpstock Europe 2005 in Dresden
- Presentation of first concept and design studies at Developers Workshop 2006 in Biel, Switzerland
- License decision during Summer 2006

The Idea III

- First 0.1 release of *The Design of Voyager* released to the public for Warpstock Canada 2006

What Voyager *is not*

- Voyager is **not** the OS/2 and eCS killer!
- netlabs.org will **not** stop development for eCS software
- Voyager is not something you can use right now (yet)
- It is not vaporware

The Goal

There are many free desktop environments available, like KDE and GNOME. Our focus is different:

- SOM like object model, binary compatible (unlike everything else out there on Unix-like systems)
- Provide a WPS like desktop environment (*OS/2 Feeling*)
- Well integrated applications (drag & drop, CUA, etc)
- Focus on localization right from the beginning (Unicode/UTF-8/16/32)
- Keep unique ideas like IOProcs and re-implement them
- Use as much existing code/apps as possible (as long as it makes sense), don't reinvent the wheel
- Document what we do!

The Goal

There are many free desktop environments available, like KDE and GNOME. Our focus is different:

- SOM like object model, binary compatible (unlike everything else out there on Unix-like systems)
- Provide a WPS like desktop environment (*OS/2 Feeling*)
- Well integrated applications (drag & drop, CUA, etc)
- Focus on localization right from the beginning (Unicode/UTF-8/16/32)
- Keep unique ideas like IOProcs and re-implement them
- Use as much existing code/apps as possible (as long as it makes sense), don't reinvent the wheel
- Document what we do!

The Goal

There are many free desktop environments available, like KDE and GNOME. Our focus is different:

- SOM like object model, binary compatible (unlike everything else out there on Unix-like systems)
- Provide a WPS like desktop environment (*OS/2 Feeling*)
- Well integrated applications (drag & drop, CUA, etc)
- Focus on localization right from the beginning (Unicode/UTF-8/16/32)
- Keep unique ideas like IOProcs and re-implement them
- Use as much existing code/apps as possible (as long as it makes sense), don’t reinvent the wheel
- Document what we do!

The Goal

There are many free desktop environments available, like KDE and GNOME. Our focus is different:

- SOM like object model, binary compatible (unlike everything else out there on Unix-like systems)
- Provide a WPS like desktop environment (*OS/2 Feeling*)
- Well integrated applications (drag & drop, CUA, etc)
- Focus on localization right from the beginning (Unicode/UTF-8/16/32)
- Keep unique ideas like IOProcs and re-implement them
- Use as much existing code/apps as possible (as long as it makes sense), don't reinvent the wheel
- Document what we do!

The Goal

There are many free desktop environments available, like KDE and GNOME. Our focus is different:

- SOM like object model, binary compatible (unlike everything else out there on Unix-like systems)
- Provide a WPS like desktop environment (*OS/2 Feeling*)
- Well integrated applications (drag & drop, CUA, etc)
- Focus on localization right from the beginning (Unicode/UTF-8/16/32)
- Keep unique ideas like IOProcs and re-implement them
- Use as much existing code/apps as possible (as long as it makes sense), don't reinvent the wheel
- Document what we do!

The Goal

There are many free desktop environments available, like KDE and GNOME. Our focus is different:

- SOM like object model, binary compatible (unlike everything else out there on Unix-like systems)
- Provide a WPS like desktop environment (*OS/2 Feeling*)
- Well integrated applications (drag & drop, CUA, etc)
- Focus on localization right from the beginning (Unicode/UTF-8/16/32)
- Keep unique ideas like IOProcs and re-implement them
- Use as much existing code/apps as possible (as long as it makes sense), don't reinvent the wheel
- Document what we do!

The Goal

There are many free desktop environments available, like KDE and GNOME. Our focus is different:

- SOM like object model, binary compatible (unlike everything else out there on Unix-like systems)
- Provide a WPS like desktop environment (*OS/2 Feeling*)
- Well integrated applications (drag & drop, CUA, etc)
- Focus on localization right from the beginning (Unicode/UTF-8/16/32)
- Keep unique ideas like IOProcs and re-implement them
- Use as much existing code/apps as possible (as long as it makes sense), don't reinvent the wheel
- Document what we do!

Development Path

- Development of Voyager should be possible on many platforms, starting on eCS
- Support for the most popular Unix-like systems is required
- eCS developers should be motivated to use SOM for new ideas because code can be partially reused
- Users can continue to use eCS as we know it today and still get new features
- Part-by-part replacement for smooth migration

Development Path

- Development of Voyager should be possible on many platforms, starting on eCS
- Support for the most popular Unix-like systems is required
- eCS developers should be motivated to use SOM for new ideas because code can be partially reused
- Users can continue to use eCS as we know it today and still get new features
- Part-by-part replacement for smooth migration

Development Path

- Development of Voyager should be possible on many platforms, starting on eCS
- Support for the most popular Unix-like systems is required
- eCS developers should be motivated to use SOM for new ideas because code can be partially reused
- Users can continue to use eCS as we know it today and still get new features
- Part-by-part replacement for smooth migration

Development Path

- Development of Voyager should be possible on many platforms, starting on eCS
- Support for the most popular Unix-like systems is required
- eCS developers should be motivated to use SOM for new ideas because code can be partially reused
- Users can continue to use eCS as we know it today and still get new features
- Part-by-part replacement for smooth migration

Development Path

- Development of Voyager should be possible on many platforms, starting on eCS
- Support for the most popular Unix-like systems is required
- eCS developers should be motivated to use SOM for new ideas because code can be partially reused
- Users can continue to use eCS as we know it today and still get new features
- Part-by-part replacement for smooth migration

The Vision

There is more to do…

- A team of interested developers might start working on an OS/2 compatibility layer for Unix-like systems
- In long term we need a new kernel (discussion is open)
- Most of the OS/2 coders don't like the Linux design so other options are preferred
- If you can help out on that project, you are very welcome :)
- Final goal: our own distribution based on existing and new software
- And finally: *Phase 3* (aka *World Domination*^{TM})

The Vision

There is more to do…

- A team of interested developers might start working on an OS/2 compatibility layer for Unix-like systems
- In long term we need a new kernel (discussion is open)
- Most of the OS/2 coders don't like the Linux design so other options are preferred
- If you can help out on that project, you are very welcome :)
- Final goal: our own distribution based on existing and new software
- And finally: *Phase 3* (aka *World Domination*^{TM})

The Vision

There is more to do...

- A team of interested developers might start working on an OS/2 compatibility layer for Unix-like systems
- In long term we need a new kernel (discussion is open)
- Most of the OS/2 coders don't like the Linux design so other options are preferred
- If you can help out on that project, you are very welcome :)
- Final goal: our own distribution based on existing and new software
- And finally: *Phase 3* (aka *World Domination*^{TM})

The Vision

There is more to do…

- A team of interested developers might start working on an OS/2 compatibility layer for Unix-like systems
- In long term we need a new kernel (discussion is open)
- Most of the OS/2 coders don't like the Linux design so other options are preferred
- If you can help out on that project, you are very welcome :)
- Final goal: our own distribution based on existing and new software
- And finally: *Phase 3* (aka *World Domination*^{TM})

The Vision

There is more to do...

- A team of interested developers might start working on an OS/2 compatibility layer for Unix-like systems
- In long term we need a new kernel (discussion is open)
- Most of the OS/2 coders don't like the Linux design so other options are preferred
- If you can help out on that project, you are very welcome :)
- Final goal: our own distribution based on existing and new software
- And finally: *Phase 3* (aka *World Domination*^{TM})

The Vision

There is more to do...

- A team of interested developers might start working on an OS/2 compatibility layer for Unix-like systems
- In long term we need a new kernel (discussion is open)
- Most of the OS/2 coders don't like the Linux design so other options are preferred
- If you can help out on that project, you are very welcome :)
- Final goal: our own distribution based on existing and new software
- And finally: *Phase 3* (aka *World Domination*^{TM})

The Codename

Why *The Voyager Project*?

- Fits into OS/2 codenames (Star Trek)
- NASA Spacecraft: Probe since 1977, longest-lasting mission of NASA, visited Jupiter and Saturn and provided detailed images of them
- It is really just a codename, not a definitive name for a product.

The Codename

Why *The Voyager Project*?

- Fits into OS/2 codenames (Star Trek)
- NASA Spacecraft: Probe since 1977, longest-lasting mission of NASA, visited Jupiter and Saturn and provided detailed images of them
- It is really just a codename, not a definitive name for a product.

The Codename

Why *The Voyager Project*?

- Fits into OS/2 codenames (Star Trek)
- NASA Spacecraft: Probe since 1977, longest-lasting mission of NASA, visited Jupiter and Saturn and provided detailed images of them
- It is really just a codename, not a definitive name for a product.

Voyager Components, now

- Netlabs Object Model (SOM)
- Voyager Desktop (WPS)
- Cairo (GPI/GDI, what Peter loves ;)
- GTK+ (GUI Toolkit, PM)
- Triton (Multimedia Subsystem, MMPM/MMOS2)

Voyager Components, now

- Netlabs Object Model (SOM)
- Voyager Desktop (WPS)
- Cairo (GPI/GDI, what Peter loves ;)
- GTK+ (GUI Toolkit, PM)
- Triton (Multimedia Subsystem, MMPM/MMOS2)

Voyager Components, now

- Netlabs Object Model (SOM)
- Voyager Desktop (WPS)
- Cairo (GPI/GDI, what Peter loves ;)
- GTK+ (GUI Toolkit, PM)
- Triton (Multimedia Subsystem, MMPM/MMOS2)

Voyager Components, now

- Netlabs Object Model (SOM)
- Voyager Desktop (WPS)
- Cairo (GPI/GDI, what Peter loves ;)
- GTK+ (GUI Toolkit, PM)
- Triton (Multimedia Subsystem, MMPM/MMOS2)

Voyager Components, now

- Netlabs Object Model (SOM)
- Voyager Desktop (WPS)
- Cairo (GPI/GDI, what Peter loves ;)
- GTK+ (GUI Toolkit, PM)
- Triton (Multimedia Subsystem, MMPM/MMOS2)

Voyager Components, next steps

- Proper security concept
- Neptune (Window Manager for Cairo and Xlib)
- OpenGL based Cairo backend
- Xorg OpenGL device drivers (no Xlib necessary in mid-term)

Voyager Components, next steps

- Proper security concept
- Neptune (Window Manager for Cairo and Xlib)
- OpenGL based Cairo backend
- Xorg OpenGL device drivers (no Xlib necessary in mid-term)

Voyager Components, next steps

- Proper security concept
- Neptune (Window Manager for Cairo and Xlib)
- OpenGL based Cairo backend
- Xorg OpenGL device drivers (no Xlib necessary in mid-term)

Voyager Components, next steps

- Proper security concept
- Neptune (Window Manager for Cairo and Xlib)
- OpenGL based Cairo backend
- Xorg OpenGL device drivers (no Xlib necessary in mid-term)

Netlabs Object Model

- SOM is a binary compatible object model, no need to recompile objects (unlike on GNOME for example)
- There are some design documents from IBM itself (published in ACM)
- There is quite some documentation available written by former IBM employees
- Chris Wohlgemuth started to re-implement the concepts under the name NOM
- This is the base for a WPS like desktop
- Chris: Status? :-)

Netlabs Object Model

- SOM is a binary compatible object model, no need to recompile objects (unlike on GNOME for example)
- There are some design documents from IBM itself (published in ACM)
- There is quite some documentation available written by former IBM employees
- Chris Wohlgemuth started to re-implement the concepts under the name NOM
- This is the base for a WPS like desktop
- Chris: Status? :-)

Netlabs Object Model

- SOM is a binary compatible object model, no need to recompile objects (unlike on GNOME for example)
- There are some design documents from IBM itself (published in ACM)
- There is quite some documentation available written by former IBM employees
- Chris Wohlgemuth started to re-implement the concepts under the name NOM
- This is the base for a WPS like desktop
- Chris: Status? :-)

Netlabs Object Model

- SOM is a binary compatible object model, no need to recompile objects (unlike on GNOME for example)
- There are some design documents from IBM itself (published in ACM)
- There is quite some documentation available written by former IBM employees
- Chris Wohlgemuth started to re-implement the concepts under the name NOM
- This is the base for a WPS like desktop
- Chris: Status? :-)

Netlabs Object Model

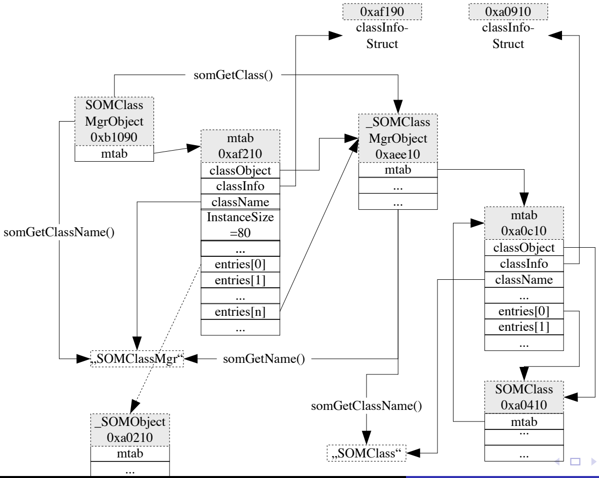
- SOM is a binary compatible object model, no need to recompile objects (unlike on GNOME for example)
- There are some design documents from IBM itself (published in ACM)
- There is quite some documentation available written by former IBM employees
- Chris Wohlgemuth started to re-implement the concepts under the name NOM
- This is the base for a WPS like desktop
- Chris: Status? :-)

Netlabs Object Model

- SOM is a binary compatible object model, no need to recompile objects (unlike on GNOME for example)
- There are some design documents from IBM itself (published in ACM)
- There is quite some documentation available written by former IBM employees
- Chris Wohlgemuth started to re-implement the concepts under the name NOM
- This is the base for a WPS like desktop
- Chris: Status? :-)

NOM Design

SOMClassMgrObject layout



Netlabs Object Model

Things that work right now:

- Binding files creation (IDL compiler)
- Class creation from IDL files
- Subclassing
- Explicit metaclasses
- Method overriding

Netlabs Object Model

Things that work right now:

- Binding files creation (IDL compiler)
- Class creation from IDL files
- Subclassing
- Explicit metaclasses
- Method overriding

Netlabs Object Model

Things that work right now:

- Binding files creation (IDL compiler)
- Class creation from IDL files
- Subclassing
- Explicit metaclasses
- Method overriding

Netlabs Object Model

Things that work right now:

- Binding files creation (IDL compiler)
- Class creation from IDL files
- Subclassing
- Explicit metaclasses
- Method overriding

Netlabs Object Model

Things that work right now:

- Binding files creation (IDL compiler)
- Class creation from IDL files
- Subclassing
- Explicit metaclasses
- Method overriding

Netlabs Object Model

Thinks that do not work yet:

- Class replacement
- Dynamical loading of classes (have to be linked in atm)
- . . .

Difference to SOM:

- Use of an environment pointer in each method call (CORBA exception handling)
- IDL is 100% CORBA
- SOM specific IDL extensions are not supported. Implemented as macros instead.
- . . .

Netlabs Object Model

Thinks that do not work yet:

- Class replacement
- Dynamical loading of classes (have to be linked in atm)
- . . .

Difference to SOM:

- Use of an environment pointer in each method call (CORBA exception handling)
- IDL is 100% CORBA
- SOM specific IDL extensions are not supported. Implemented as macros instead.
- . . .

Netlabs Object Model

Thinks that do not work yet:

- Class replacement
- Dynamical loading of classes (have to be linked in atm)
- ...

Difference to SOM:

- Use of an environment pointer in each method call (CORBA exception handling)
- IDL is 100% CORBA
- SOM specific IDL extensions are not supported. Implemented as macros instead.
- ...

Netlabs Object Model

Thinks that do not work yet:

- Class replacement
- Dynamical loading of classes (have to be linked in atm)
- . . .

Difference to SOM:

- Use of an environment pointer in each method call (CORBA exception handling)
- IDL is 100% CORBA
- SOM specific IDL extensions are not supported. Implemented as macros instead.
- . . .

Netlabs Object Model

Thinks that do not work yet:

- Class replacement
- Dynamical loading of classes (have to be linked in atm)
- . . .

Difference to SOM:

- Use of an environment pointer in each method call (CORBA exception handling)
- IDL is 100% CORBA
- SOM specific IDL extensions are not supported. Implemented as macros instead.
- . . .

Netlabs Object Model

Thinks that do not work yet:

- Class replacement
- Dynamical loading of classes (have to be linked in atm)
- . . .

Difference to SOM:

- Use of an environment pointer in each method call (CORBA exception handling)
- IDL is 100% CORBA
- SOM specific IDL extensions are not supported. Implemented as macros instead.
- . . .

Netlabs Object Model

Thinks that do not work yet:

- Class replacement
- Dynamical loading of classes (have to be linked in atm)
- . . .

Difference to SOM:

- Use of an environment pointer in each method call (CORBA exception handling)
- IDL is 100% CORBA
- SOM specific IDL extensions are not supported. Implemented as macros instead.
- . . .

Voyager Desktop

- WPS base classes in development now. Already some code available
- Snapshot available but not for mere mortals
- Subversion available but no check-in yet
- Todo lists will be online soon
- Right now based on Xlib (Everblue) because of GTK+

Voyager Desktop

- WPS base classes in development now. Already some code available
- Snapshot available but not for mere mortals
- Subversion available but no check-in yet
- Todo lists will be online soon
- Right now based on Xlib (Everblue) because of GTK+

Voyager Desktop

- WPS base classes in development now. Already some code available
- Snapshot available but not for mere mortals
- Subversion available but no check-in yet
- Todo lists will be online soon
- Right now based on Xlib (Everblue) because of GTK+

Voyager Desktop

- WPS base classes in development now. Already some code available
- Snapshot available but not for mere mortals
- Subversion available but no check-in yet
- Todo lists will be online soon
- Right now based on Xlib (Everblue) because of GTK+

Voyager Desktop

- WPS base classes in development now. Already some code available
- Snapshot available but not for mere mortals
- Subversion available but no check-in yet
- Todo lists will be online soon
- Right now based on Xlib (Everblue) because of GTK+

Cairo

- modern, open source, cross-platform 2D API
- PDF/PS-like 2D API (hint: MacOS X Quartz)
- multiple output systems (screen, printer)
- OS/2 port exists, not accelerated so far
- OpenGL based backend exists, called Glitz (hint: MacOS X Quartz :)

Cairo

- modern, open source, cross-platform 2D API
- PDF/PS-like 2D API (hint: MacOS X Quartz)
- multiple output systems (screen, printer)
- OS/2 port exists, not accelerated so far
- OpenGL based backend exists, called Glitz (hint: MacOS X Quartz :)

Cairo

- modern, open source, cross-platform 2D API
- PDF/PS-like 2D API (hint: MacOS X Quartz)
- multiple output systems (screen, printer)
- OS/2 port exists, not accelerated so far
- OpenGL based backend exists, called Glitz (hint: MacOS X Quartz :)

Cairo

- modern, open source, cross-platform 2D API
- PDF/PS-like 2D API (hint: MacOS X Quartz)
- multiple output systems (screen, printer)
- OS/2 port exists, not accelerated so far
- OpenGL based backend exists, called Glitz (hint: MacOS X Quartz :)

Cairo

- modern, open source, cross-platform 2D API
- PDF/PS-like 2D API (hint: MacOS X Quartz)
- multiple output systems (screen, printer)
- OS/2 port exists, not accelerated so far
- OpenGL based backend exists, called Glitz (hint: MacOS X Quartz :)

GTK+ Toolkit

- Most used toolkit on Unix-like systems nowadays
- LGPL license (binary linking possible, unlike qt)
- Abstraction layer called GLIB
- Ongoing development
- wxWidgets and SWT are implemented on top of it
- Not yet available on eCS, PM port is evaluated right now
- Everblue already provides some support for it

GTK+ Toolkit

- Most used toolkit on Unix-like systems nowadays
- LGPL license (binary linking possible, unlike qt)
- Abstraction layer called GLIB
- Ongoing development
- wxWidgets and SWT are implemented on top of it
- Not yet available on eCS, PM port is evaluated right now
- Everblue already provides some support for it

GTK+ Toolkit

- Most used toolkit on Unix-like systems nowadays
- LGPL license (binary linking possible, unlike qt)
- Abstraction layer called GLIB
- Ongoing development
- wxWidgets and SWT are implemented on top of it
- Not yet available on eCS, PM port is evaluated right now
- Everblue already provides some support for it

GTK+ Toolkit

- Most used toolkit on Unix-like systems nowadays
- LGPL license (binary linking possible, unlike qt)
- Abstraction layer called GLIB
- Ongoing development
- wxWidgets and SWT are implemented on top of it
- Not yet available on eCS, PM port is evaluated right now
- Everblue already provides some support for it

GTK+ Toolkit

- Most used toolkit on Unix-like systems nowadays
- LGPL license (binary linking possible, unlike qt)
- Abstraction layer called GLIB
- Ongoing development
- wxWidgets and SWT are implemented on top of it
- Not yet available on eCS, PM port is evaluated right now
- Everblue already provides some support for it

GTK+ Toolkit

- Most used toolkit on Unix-like systems nowadays
- LGPL license (binary linking possible, unlike qt)
- Abstraction layer called GLIB
- Ongoing development
- wxWidgets and SWT are implemented on top of it
- Not yet available on eCS, PM port is evaluated right now
- Everblue already provides some support for it

GTK+ Toolkit

- Most used toolkit on Unix-like systems nowadays
- LGPL license (binary linking possible, unlike qt)
- Abstraction layer called GLIB
- Ongoing development
- wxWidgets and SWT are implemented on top of it
- Not yet available on eCS, PM port is evaluated right now
- Everblue already provides some support for it

Triton

- Well-defined subsystem
- Core library
- Easy-to-use API for the application developers to visualize and control multimedia contents
- Easy-to-use API for the plugin developers to be able to extend the system with support of new multimedia formats

Triton

- Well-defined subsystem
- Core library
- Easy-to-use API for the application developers to visualize and control multimedia contents
- Easy-to-use API for the plugin developers to be able to extend the system with support of new multimedia formats

Triton

- Well-defined subsystem
- Core library
- Easy-to-use API for the application developers to visualize and control multimedia contents
- Easy-to-use API for the plugin developers to be able to extend the system with support of new multimedia formats

Triton

- Well-defined subsystem
- Core library
- Easy-to-use API for the application developers to visualize and control multimedia contents
- Easy-to-use API for the plugin developers to be able to extend the system with support of new multimedia formats

Triton Status

- First audio codec implemented: MP3 (April 2006)
- First image codecs implemented: GBM module support
- First video codec implemented: (can't remember which one ;)

Triton Status

- First audio codec implemented: MP3 (April 2006)
- First image codecs implemented: GBM module support
- First video codec implemented: (can't remember which one ;)

Triton Status

- First audio codec implemented: MP3 (April 2006)
- First image codecs implemented: GBM module support
- First video codec implemented: (can't remember which one ;)

Neptune Window Manager

A WM has to take care of

- the open client windows
- the client stacking order
- the client to virtual desktop assignment
- the session
- multiple screens
- client decorations
- ...

Status: Prototype done, developer is very busy with other things. Source code of prototype available soon.

Neptune Window Manager

A WM has to take care of

- the open client windows
- the client stacking order
- the client to virtual desktop assignment
- the session
- multiple screens
- client decorations
- ...

Status: Prototype done, developer is very busy with other things. Source code of prototype available soon.

Neptune Window Manager

A WM has to take care of

- the open client windows
- the client stacking order
- the client to virtual desktop assignment
- the session
- multiple screens
- client decorations
- ...

Status: Prototype done, developer is very busy with other things. Source code of prototype available soon.

Neptune Window Manager

A WM has to take care of

- the open client windows
- the client stacking order
- the client to virtual desktop assignment
- the session
- multiple screens
- client decorations
- ...

Status: Prototype done, developer is very busy with other things. Source code of prototype available soon.

Neptune Window Manager

A WM has to take care of

- the open client windows
- the client stacking order
- the client to virtual desktop assignment
- the session
- multiple screens
- client decorations
- ...

Status: Prototype done, developer is very busy with other things. Source code of prototype available soon.

Neptune Window Manager

A WM has to take care of

- the open client windows
- the client stacking order
- the client to virtual desktop assignment
- the session
- multiple screens
- client decorations
- ...

Status: Prototype done, developer is very busy with other things. Source code of prototype available soon.

Neptune Window Manager

A WM has to take care of

- the open client windows
- the client stacking order
- the client to virtual desktop assignment
- the session
- multiple screens
- client decorations
- ...

Status: Prototype done, developer is very busy with other things. Source code of prototype available soon.

OpenGL Backend

- OpenGL is the standard nowadays, only MS is doing their own game mit DirectX
- OpenGL allows fancy stuff like blending and 3D effects, no fun on 2D Hardware
- powerful API
- MacOS X is using it as backend, Xorg Project is moving towards OpenGL as well

OpenGL Backend

- OpenGL is the standard nowadays, only MS is doing their own game mit DirectX
- OpenGL allows fancy stuff like blending and 3D effects, no fun on 2D Hardware
- powerful API
- MacOS X is using it as backend, Xorg Project is moving towards OpenGL as well

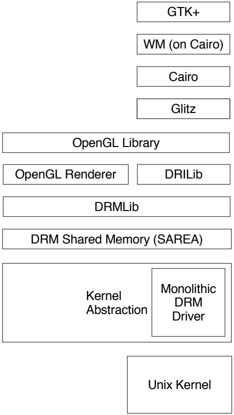
OpenGL Backend

- OpenGL is the standard nowadays, only MS is doing their own game mit DirectX
- OpenGL allows fancy stuff like blending and 3D effects, no fun on 2D Hardware
- powerful API
- MacOS X is using it as backend, Xorg Project is moving towards OpenGL as well

OpenGL Backend

- OpenGL is the standard nowadays, only MS is doing their own game mit DirectX
- OpenGL allows fancy stuff like blending and 3D effects, no fun on 2D Hardware
- powerful API
- MacOS X is using it as backend, Xorg Project is moving towards OpenGL as well

Xorg OpenGL Design



Simplified design of the Xorg OpenGL backend (taken from official docs), Xlib stripped out

Some Ideas

People do ask for binary compatibility. We see the following options:

- Use a pure VM based solution - easy and will work well, already possible
- Rewrite the whole OS as proposed by some people - unrealistic, waste of resources
- Implement a minimal OS/2 personality on top of an existing kernel and get binary compatibility to work

Some Ideas

People do ask for binary compatibility. We see the following options:

- Use a pure VM based solution - easy and will work well, already possible
- Rewrite the whole OS as proposed by some people - unrealistic, waste of resources
- Implement a minimal OS/2 personality on top of an existing kernel and get binary compatibility to work

Some Ideas

People do ask for binary compatibility. We see the following options:

- Use a pure VM based solution - easy and will work well, already possible
- Rewrite the whole OS as proposed by some people - unrealistic, waste of resources
- Implement a minimal OS/2 personality on top of an existing kernel and get binary compatibility to work

OS/2 OS Personality

To get a minimal OS/2 personality to work we would have to implement:

- `DOS*`, `MOU*`, `KBD*` and `VIO*` API's
- LX loader (Knuts kLoader)
- GRADD driver for OpenGL backend
- Some other things

Some things exist, like JdeBP - unfortunately not open source. Work on some parts started, much work left.

OS/2 OS Personality

To get a minimal OS/2 personality to work we would have to implement:

- DOS*, MOU*, KBD* and VIO* API's
- LX loader (Knuts kLoader)
- GRADD driver for OpenGL backend
- Some other things

Some things exist, like JdeBP - unfortunately not open source. Work on some parts started, much work left.

OS/2 OS Personality

To get a minimal OS/2 personality to work we would have to implement:

- `DOS*`, `MOU*`, `KBD*` and `VIO*` API's
- LX loader (Knuts kLoader)
- GRADD driver for OpenGL backend
- Some other things

Some things exist, like JdeBP - unfortunately not open source. Work on some parts started, much work left.

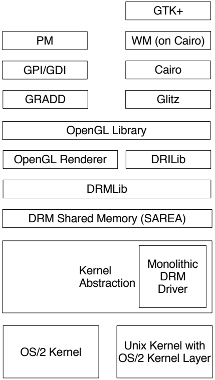
OS/2 OS Personality

To get a minimal OS/2 personality to work we would have to implement:

- `DOS*`, `MOU*`, `KBD*` and `VIO*` API's
- LX loader (Knuts kLoader)
- GRADD driver for OpenGL backend
- Some other things

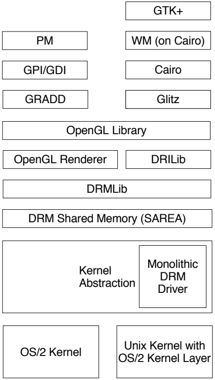
Some things exist, like JdeBP - unfortunately not open source. Work on some parts started, much work left.

GRADD on OpenGL



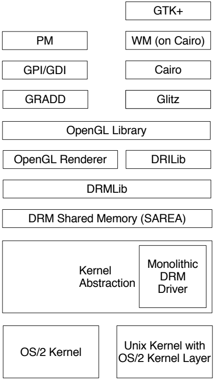
- OpenGL GRADD driver could be used on OS/2 already, Xorg should work (with some effort)
- As soon as a OS/2 personality would work, one could migrate to the new kernel
- This might become an important project for eCS anyway

GRADD on OpenGL



- OpenGL GRADD driver could be used on OS/2 already, Xorg should work (with some effort)
- As soon as a OS/2 personality would work, one could migrate to the new kernel
- This might become an important project for eCS anyway

GRADD on OpenGL



- OpenGL GRADD driver could be used on OS/2 already, Xorg should work (with some effort)
- As soon as a OS/2 personality would work, one could migrate to the new kernel
- This might become an important project for eCS anyway

Voyager License

- we do not want to use GPL for our own code for various reasons
- the object model allows binary code, GPL would make that difficult
- we might use (L)GPLed code for parts when it makes sense to do that
- LGPL/CDDL dual license for Voyager core
- APL for sample applications

Voyager License

- we do not want to use GPL for our own code for various reasons
- the object model allows binary code, GPL would make that difficult
- we might use (L)GPLed code for parts when it makes sense to do that
- LGPL/CDDL dual license for Voyager core
- APL for sample applications

Voyager License

- we do not want to use GPL for our own code for various reasons
- the object model allows binary code, GPL would make that difficult
- we might use (L)GPLed code for parts when it makes sense to do that
- LGPL/CDDL dual license for Voyager core
- APL for sample applications

Voyager License

- we do not want to use GPL for our own code for various reasons
- the object model allows binary code, GPL would make that difficult
- we might use (L)GPLed code for parts when it makes sense to do that
- LGPL/CDDL dual license for Voyager core
- APL for sample applications

Voyager License

- we do not want to use GPL for our own code for various reasons
- the object model allows binary code, GPL would make that difficult
- we might use (L)GPLed code for parts when it makes sense to do that
- LGPL/CDDL dual license for Voyager core
- APL for sample applications

Timeline

- First source code is online!
  - Triton
  - Netlabs Object Model
- First version of *The Design of Voyager* got released in October 2006
- New server for netlabs.org, new CMS with dedicated Voyager pages (work in progress)
- Goal: provide development environments for at least two platforms ASAP

Timeline

- First source code is online!
  - Triton
  - Netlabs Object Model
- First version of *The Design of Voyager* got released in October 2006
- New server for netlabs.org, new CMS with dedicated Voyager pages (work in progress)
- Goal: provide development environments for at least two platforms ASAP

Timeline

- First source code is online!
  - Triton
  - Netlabs Object Model
- First version of *The Design of Voyager* got released in October 2006
- New server for netlabs.org, new CMS with dedicated Voyager pages (work in progress)
- Goal: provide development environments for at least two platforms ASAP

Timeline

- First source code is online!
  - Triton
  - Netlabs Object Model
- First version of *The Design of Voyager* got released in October 2006
- New server for netlabs.org, new CMS with dedicated Voyager pages (work in progress)
- Goal: provide development environments for at least two platforms ASAP

Timeline

- First source code is online!
  - Triton
  - Netlabs Object Model
- First version of *The Design of Voyager* got released in October 2006
- New server for netlabs.org, new CMS with dedicated Voyager pages (work in progress)
- Goal: provide development environments for at least two platforms ASAP

Timeline

- First source code is online!
  - Triton
  - Netlabs Object Model
- First version of *The Design of Voyager* got released in October 2006
- New server for netlabs.org, new CMS with dedicated Voyager pages (work in progress)
- Goal: provide development environments for at least two platforms ASAP

Join the Project

- Go to `http://voyager.netlabs.org`
- Check the Voyager Wiki and FAQ
- Join the Voyager Mailinglist at
- join the `#netlabs` IRC channel (see `http://www.ecomstation.com/chat.phtml`)
- Contribute to *The Design of Voyager*
- Contribute to NOM, Triton, compatibility layer etc.
- Translate as soon as first text is available
- Support us with money

Join the Project

- Go to `http://voyager.netlabs.org`
- Check the Voyager Wiki and FAQ
- Join the Voyager Mailinglist at
- join the `#netlabs` IRC channel (see `http://www.ecomstation.com/chat.phtml`)
- Contribute to *The Design of Voyager*
- Contribute to NOM, Triton, compatibility layer etc.
- Translate as soon as first text is available
- Support us with money

Join the Project

- Go to `http://voyager.netlabs.org`
- Check the Voyager Wiki and FAQ
- Join the Voyager Mailinglist at
- join the `#netlabs` IRC channel (see `http://www.ecomstation.com/chat.phtml`)
- Contribute to *The Design of Voyager*
- Contribute to NOM, Triton, compatibility layer etc.
- Translate as soon as first text is available
- Support us with money

Join the Project

- Go to `http://voyager.netlabs.org`
- Check the Voyager Wiki and FAQ
- Join the Voyager Mailinglist at
- join the `#netlabs` IRC channel (see `http://www.ecomstation.com/chat.phtml`)
- Contribute to *The Design of Voyager*
- Contribute to NOM, Triton, compatibility layer etc.
- Translate as soon as first text is available
- Support us with money

Join the Project

- Go to `http://voyager.netlabs.org`
- Check the Voyager Wiki and FAQ
- Join the Voyager Mailinglist at
- join the `#netlabs` IRC channel (see `http://www.ecomstation.com/chat.phtml`)
- Contribute to *The Design of Voyager*
- Contribute to NOM, Triton, compatibility layer etc.
- Translate as soon as first text is available
- Support us with money

Join the Project

- Go to `http://voyager.netlabs.org`
- Check the Voyager Wiki and FAQ
- Join the Voyager Mailinglist at
- join the `#netlabs` IRC channel (see `http://www.ecomstation.com/chat.phtml`)
- Contribute to *The Design of Voyager*
- Contribute to NOM, Triton, compatibility layer etc.
- Translate as soon as first text is available
- Support us with money

Join the Project

- Go to `http://voyager.netlabs.org`
- Check the Voyager Wiki and FAQ
- Join the Voyager Mailinglist at
- join the `#netlabs` IRC channel (see `http://www.ecomstation.com/chat.phtml`)
- Contribute to *The Design of Voyager*
- Contribute to NOM, Triton, compatibility layer etc.
- Translate as soon as first text is available
- Support us with money

Join the Project

- Go to `http://voyager.netlabs.org`
- Check the Voyager Wiki and FAQ
- Join the Voyager Mailinglist at
- join the `#netlabs` IRC channel (see `http://www.ecomstation.com/chat.phtml`)
- Contribute to *The Design of Voyager*
- Contribute to NOM, Triton, compatibility layer etc.
- Translate as soon as first text is available
- Support us with money

Voyager?

We talked about what Voyager is not, so what is it then?

- It is a great motivation to go on with eCS
- It is a ambitious project
- It is a great challenge
- It will become a great piece of software
- The first step to *World Domination*^{TM}

Voyager?

We talked about what Voyager is not, so what is it then?

- It is a great motivation to go on with eCS
- It is a ambitious project
- It is a great challenge
- It will become a great piece of software
- The first step to *World Domination*^{TM}

Voyager?

We talked about what Voyager is not, so what is it then?

- It is a great motivation to go on with eCS
- It is a ambitious project
- It is a great challenge
- It will become a great piece of software
- The first step to *World Domination*^{TM}

Voyager?

We talked about what Voyager is not, so what is it then?

- It is a great motivation to go on with eCS
- It is a ambitious project
- It is a great challenge
- It will become a great piece of software
- The first step to *World Domination*^{TM}

Voyager?

We talked about what Voyager is not, so what is it then?

- It is a great motivation to go on with eCS
- It is a ambitious project
- It is a great challenge
- It will become a great piece of software
- The first step to *World Domination*^{TM}

Q&A

Questions?