

# Mirage as a Xen.org incubated project

We propose to make Mirage an incubated project within Xen.org. This document aims to provide some background to Mirage and explain the relevance of Mirage to Xen (and the Xen Community).

## Background - The aims of Mirage

The cloud has so far mostly been used to consolidate existing operating systems and manage them more conveniently. The interface between guest kernels, applications and VMs are all managed very differently, leading to serious inefficiencies, unreliability and insecurity.

Our Mirage OS revisits the [library OS](#) concept (built as Nemesis in the Cambridge Computer Lab in the late '90s), and finds it a practical approach to improve this situation. Applications are written in a high-level functional programming language (OCaml) and compiled directly into microkernels that run on the Xen hypervisor. These microkernels can be treated as stand-alone *appliances* which use the Xen hypervisor interfaces directly.

Treating the hypervisor as a stable hardware platform means we can focus on high-performance protocol implementations without worrying about having to support the thousands of device drivers found in a traditional OS.

Mirage includes clean-slate functional implementations of protocols ranging from [TCP/IP](#), [DNS](#), [SSH](#), [Openflow](#) (switch/controller), [HTTP](#), XMPP and Xen inter-VM transports. Some of the new applications we are building using Mirage include the next-generation [Xen Cloud Platform](#) (a widely deployed open-source Xen distribution), dubbed [Project Windsor](#).

An example of a current Mirage appliance can be seen at [openmirage.org](#), which is a self-hosted site deployed on the public cloud (in this case, on Amazon EC2). With some further work, it should also be possible to 'encapsulate' an existing piece of software using Mirage. One example might be to use Mirage to form an appliance from an existing Content Management System (e.g. Drupal or Joomla). Such systems typically need several components and expect to be installed on a server. Mirage could be used to encapsulate a CMS, retaining only the components and bindings required by the CMS and removing the need for the rest of the Operating System. Such an appliance could be auto-configured and deployed directly to the public cloud (e.g. Rackspace or Amazon EC2).

## Relevance to Xen and its Community

Mirage has a very close dependency on the Xen hypervisor since it targets the hypervisor as a stable platform. As Mirage is written entirely in OCaml, a statically type-safe functional language, the resulting microkernels are protected

against low-level memory errors, and made convenient to program via high-level interfaces. OCaml is also the implementation language for the XCP which makes integration with these components straightforward.

Mirage produces microkernels that offer an order of magnitude reduction in code size and smaller VM memory footprints without significant performance penalty. The library-based architecture enables security, efficiency and performance optimisations, such as a single-address space that can be made immutable via an optional hypervisor extension (linked above), and compile-time configuration specialisation and inlining.

The key insight is that the hypervisor provides a virtual hardware abstraction that can be scaled dynamically — vertically with memory and virtual CPUs, and horizontally by spawning more VMs. One benefit of this approach is the ability to make use of multicore. As everything runs over the Xen VMM, multicore is simply a case of running multiple instances of a given microkernel over Xen, communicating using the standard protocol stack.

### Current status of Mirage

Mirage has already been in development for over three years and a paper will be presented at an upcoming, top-tier conference, [ASPLOS 2013](#). The paper will cover the technical background of the project as well as some of the evaluation. In the meantime, a previous paper is also available (see the [HotCloud Paper](#) of 2010). In addition, there are around 50 members on the current mailing list ([view the archives](#)) and a large body of code, which can be found on Github at <https://github.com/mirage>. Much work has been done to date on creating pure OCaml implementations of protocols, as listed previously.

The existing maintainers will continue to work on Mirage while it is incubated at xen.org. Some of the people involved in Mirage include:

- [Anil Madhavapeddy](#), University of Cambridge - Technical Lead
- [Richard Mortier](#) University of Nottingham - Project Lead at Horizon Digital Economy Research
- [Dave Scott](#), Citrix Systems - Xen and Citrix
- [Thomas Gazagnaire](#), OCamlPro - Packaging
- [Jonathan Ludlam](#), Citrix Systems - Xen Cloud Platform integration
- [Haris Rotsos](#), University of Cambridge - DNS implementation
- [Raphael Proust](#), University of Cambridge - Linear Typing
- [Balraj Singh](#), University of Cambridge - TCP/IP implementation

- [Gabor Pali](#), FreeBSD Core Team - FreeBSD kernel port
- [Robert Watson](#) University of Cambridge - FreeBSD and MIPS/BERI

**Interested parties and dependencies** In addition to the current mailing list members and contributors, Citrix also has an interest. Mirage has been used internally during tests and benchmarking and there is continued support from key Citrix members to continue development (e.g Dave Scott).

### Mirage during incubation

The aim of making Mirage an incubated project is to raise its profile among the Xen community and draw more developers into contributing code and working towards a stable release. The current project has reached a stage where others can make use of the work to date and their input can accelerate the improvements and adoption of Mirage. In general, support from the Xen community is important to the long-term success of Mirage. Engagement from vendors such as Rackspace, Amazon, Citrix and others is also key.

One of the things we hope to gain during incubation is to benefit from access to test infrastructure. This infrastructure will be used to test the build and deployment of Mirage as well as the packages it requires.

The roles during incubation will be as follows:

*Project Lead:* [Anil Madhavapeddy](#) - Anil has led the Mirage project since its inception and would continue to do so during the incubation phase.

*Project Sponsor:* [Tim Deegan](#) - Tim is one of the maintainers of the Xen hypervisor and has agreed to act as Sponsor for this project. He was previously involved with a predecessor called Melange, which was published at the EuroSys conference in 2007 (paper available at: <http://anil.recoil.org/papers/2007-eurosys-melange.pdf>)

*Project Mentor:* [Lars Kurth](#) - Lars is the Community Manager for xen.org and has agreed to act and the project's Mentor.

### License

The current codebase is released under the ISC license, with some code released under LGPLv2 (the Xen Cloud Platform). The ISC license is listed by the Open Source Initiative as one of their approved licences (<http://opensource.org/licenses/ISC>) and is also compatible with the GNU GPL (<http://www.gnu.org/licenses/license-list.html#ISC>).

As an example, Mirage code releases have a statement of the following form at the beginning of each file:

Copyright (c) 2011 Firstname Surname [email@example.org](mailto:email@example.org)

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED “AS IS” AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

**Rationale for License** As Mirage appliances are statically linked microkernels, it is necessary to use a permissive open source licence. The use of the ISC licence maximises the the number of applications that can be linked against and thus can be supported by Mirage. As noted above the ISC is an approved licence of the Open Source Initiative and is compatible with the GNU GPL. Therefore, it meets the license requirements of projects being incubated by Xen.org.