

Winged Monkey

**The nature of Winged Monkey is
irrepressible**

Context

To understand Winged Monkey, two concepts which have been discussed this week need to be borne in mind:

- **Personas.** Winged Monkey is targeted firmly at self-service users
- **Modularisation.** Winged Monkey has the potential to succeed as an discrete upstream project.

Caveats

Winged Monkey is a new project, and little is set in stone.

Also, it is an upstream, developer-led project.

Elements of what follows are my opinion.

Future reality may vary.

Definition

Winged Monkey is. . .

- A standalone application
- A simplified UI layer in front of cloud providers
- Intended for self-service consumers, not for administrators

It is a "user portal", as requested by multiple consumers of Cloud Engine.

Guidelines

- Focus on ease of use and aesthetics (built on Converge UI)
- Do a small number of things well
- Be *largely* stateless, representing state from underlying provider's API
- If the provider doesn't support it, don't show it.
 - (No Account multiplexing. No cloud brokering)
 - (Don't gradually turn into a poorly implemented Conductor clone)

Multiple Providers

Winged Monkey connects to **one of**:

oVirt, vSphere, OpenStack, Conductor, Heat.
DeltaCloud

It will act as user portal for all of those projects, only displaying the capabilities that the current provider enables.

By providing a consistent UI, it will allow users to plan to change their underlying provider later, whilst retaining a consistent user experience.

Winged Monkey passes credentials through to the provider and shows whatever those credentials allow access to.

Core features

- Browse a list of applications which I can launch
- Launch applications, with the minimum necessary steps to that goal
- See the status of applications which I have launched
- Terminate applications which I have launched
- See a history of the applications I have run in the past

Extended (optional) features

- Events
- Instance stats
- instance history
- Cost
- Quota
- More or less, that's it

Conductor will, naturally, be the best provider.

Wot no deltaxloud?

"At this point in the development, we can just as easily use rbovirt to talk to Ovirt clusters.

However, we plan on adequately encapsulating that interaction with providers, in rails engines, so that if we choose to later replace that method of cloud connectivity by using deltaxloud, we can.

Why not just make Conductor prettier?

Imagine you're a Winged Monkey developer:
Connecting directly to multiple providers
without additional dependencies is a feature

Contributions from other provider projects. A
discrete, simple project, with a clear purpose,
which can succeed upstream.

How Winged Monkey works with Conductor

- Each resource zone acts as a provider
- No Conductor functionality is descoped
- Winged Monkey will not become a prerequisite for Conductor

Achievable goals

Winged Monkey is specific and limited in scope.

Within a matter of months, we should reach the point where the UI development is "done".

Thereafter, we might:

- add a couple more gizmos to the UI
- add support for additional providers

That's it.

Progress to date

- Creation of the project on Github, and the developer mailing list.
- Lots of thinking.
- Greg and Mainn are thinking about provider access.
- Jeremy, Jaromir and Jirka are thinking about the UI.
- Angus is nagging Mike about API requirements.