

Not-So-Foreign Functions

Calling Ruby from Smalltalk

Martin McClure

2011 ESUG



Why Foreign Functions?



Shared library call

Distributed object system

Shared library call

Distributed object system

**Is there an
in-between way?**

Shared library call

Distributed object system

Multi-language VM

Shared library call

Multi-language VM

MagLev: Ruby and Smalltalk
in the same VM

How does that work?

How does that work?

An object...

How does that work?

An object...

...is an object...

How does that work?

An object...

...is an object...

...is an object...

How does that work?

A class...

How does that work?

A class...

...is a class...

How does that work?

A class...

...is a class...

...is a class...

How does that work?

BUT...

How does that work?

A message...

How does that work?

A message...

...is Smalltalk...

How does that work?

A message...

...is Smalltalk...

...or is Ruby.

How does that work?

A class...

How does that work?

A class...

...may define different
Ruby and Smalltalk
methods...

How does that work?

A class...

How does that work?

A class...

...may have different
Ruby and Smalltalk
names...

How does that work?

A class...

How does that work?

A class...

...may have different
Ruby and Smalltalk
superclasses.

Selector Supplements

- Environment ID
- Number/type of sent arguments

Selector Supplements

- Default send environment = compile environment
- Must override this default

**Onward
to the
Demo**

Thanks to:

Peter McLain

Tim Felgentreff

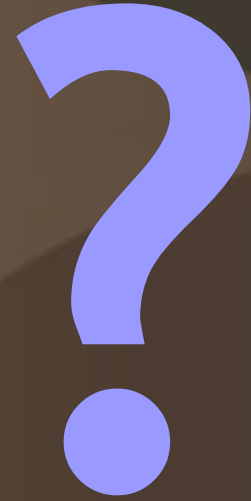
rubygems.org
> 25000 gems

maglev-gem install mailfactory



What's Next?

- Name “RubyService”?
- You tell us...



Martin McClure

2011 ESUG

