

StrokeDB

Yet Another Database?

Yurii Rashkovskii & Oleg Andreev

2008

A close-up photograph of a koala's face and upper body. The koala has dark brown fur with lighter patches around its eyes and ears. It is clinging to a light-colored, textured tree trunk with its front paws. The background is blurred, showing more of the tree and some greenery.

Trees, maps and
sortables are hard with
RDBMS

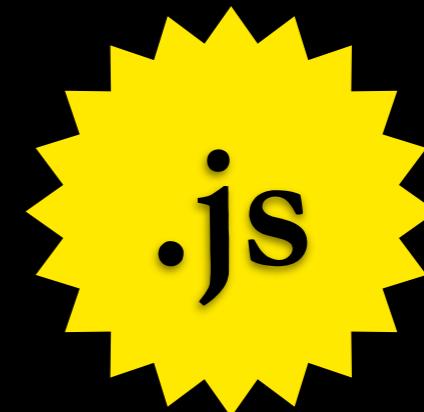


We need
flexible
data formats

document.rb



```
{J:{S:[0,"N"]}}}
```



```
{  
  "name": "Yuri",  
  "height": 180,  
  "interests": ["ruby", "lisp"]  
}
```



```
{  
    "name": "Yuri",  
    "height": 180,  
    "interests": ["ruby", "lisp"]  
}
```

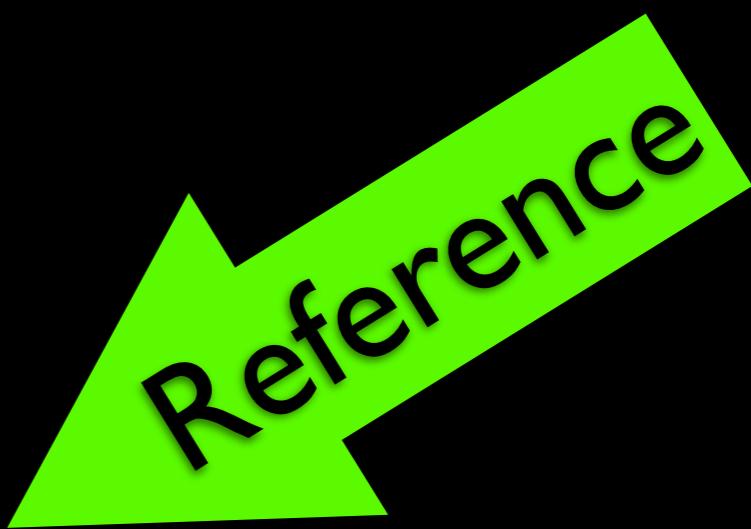


```
{  
  "name" => "Yuri",  
  "height" => 180,  
  "interests" => ["ruby", "lisp"]  
}
```

```
person = Document.new
person.name == "Yurii"      # String
person.height == 180         # Fixnum
person.interests[0] == "ruby"
person.save!
```

```
{  
  "name": "Yurii",  
  "height": 180,  
  "friends":  
    ["@#12345678-1234-5678-..."]  
}
```

```
{  
  "name": "Yurii",  
  "height": 180,  
  "friends":  
    ["@#12345678-1234-5678-..."]  
}
```



```
person.name == “Yurii”
person.height == 180
person.interests[0] == “ruby”
person.friends[0].name == “Oleg”
```

Auto dereferencing

```
person.name == "Yurii"  
person.height == 180  
person.interests[0] == "ruby"  
person.friends[0].name == "Oleg"
```

person.inspect

#<{User, Admin} “name”=>“Yurii” ...>

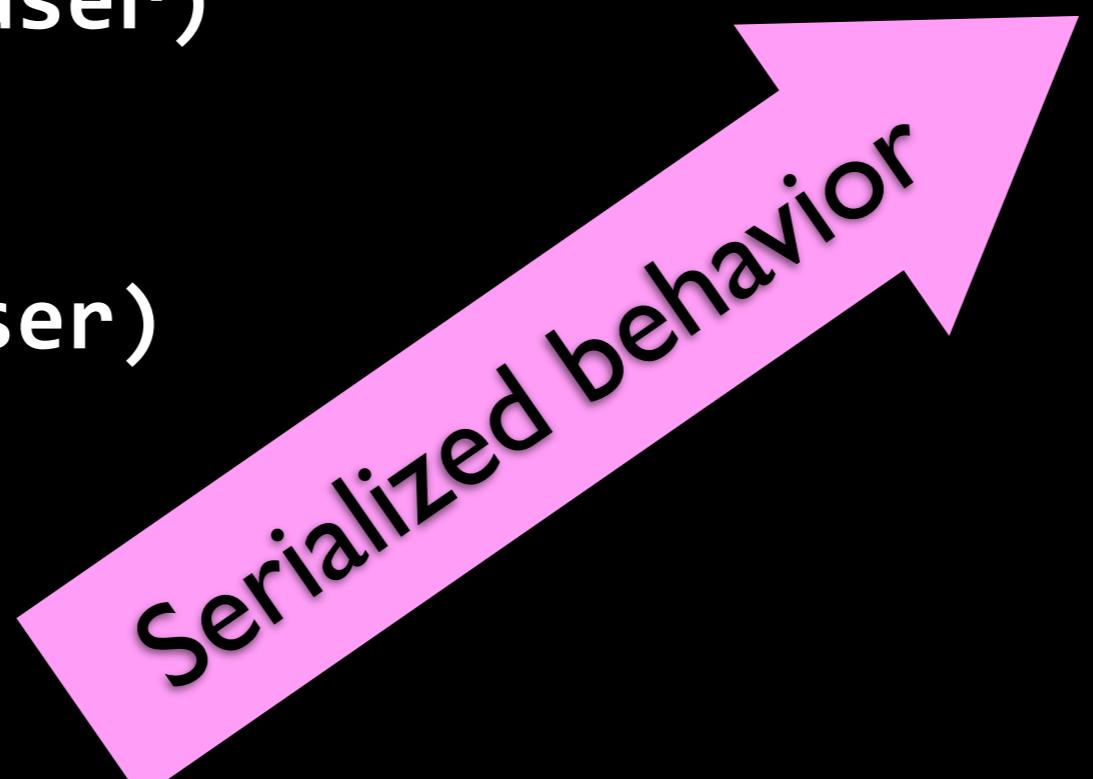
`person.inspect`



`#<{User, Admin} “name”=>“Yuri” ...>`

```
Admin = Meta.new("rights"=>["kill", "ban"]) do
  def kill!(user)
    ...
  end
  def ban!(user)
    ...
  end
end
```

```
Admin = Meta.new("rights"=>["kill", "ban"]) do
  def kill!(user)
    ...
  end
  def ban!(user)
    ...
  end
end
```



Serialized behavior

```
Admin = Meta.new("rights"=>["kill", "ban"]) do
  def kill!(user)
    ...
  end
  def ban!(user)
    ...
  end
end
```



Executable code

The data I need is
somewhere else



~~autoincrement~~

UUID

550e8400-e29b-41d4-a716-446655440000

**1 trillion UUIDs would have to be created every nanosecond
for 10 billion years to exhaust the number of UUIDs.**

Distributed for
offline work
webservices
(REST, SCHMEST)
collaboration

Distributed for
offline work
webservices
(REST, SCHMEST)
collaboration

The image displays three screenshots of web applications arranged horizontally:

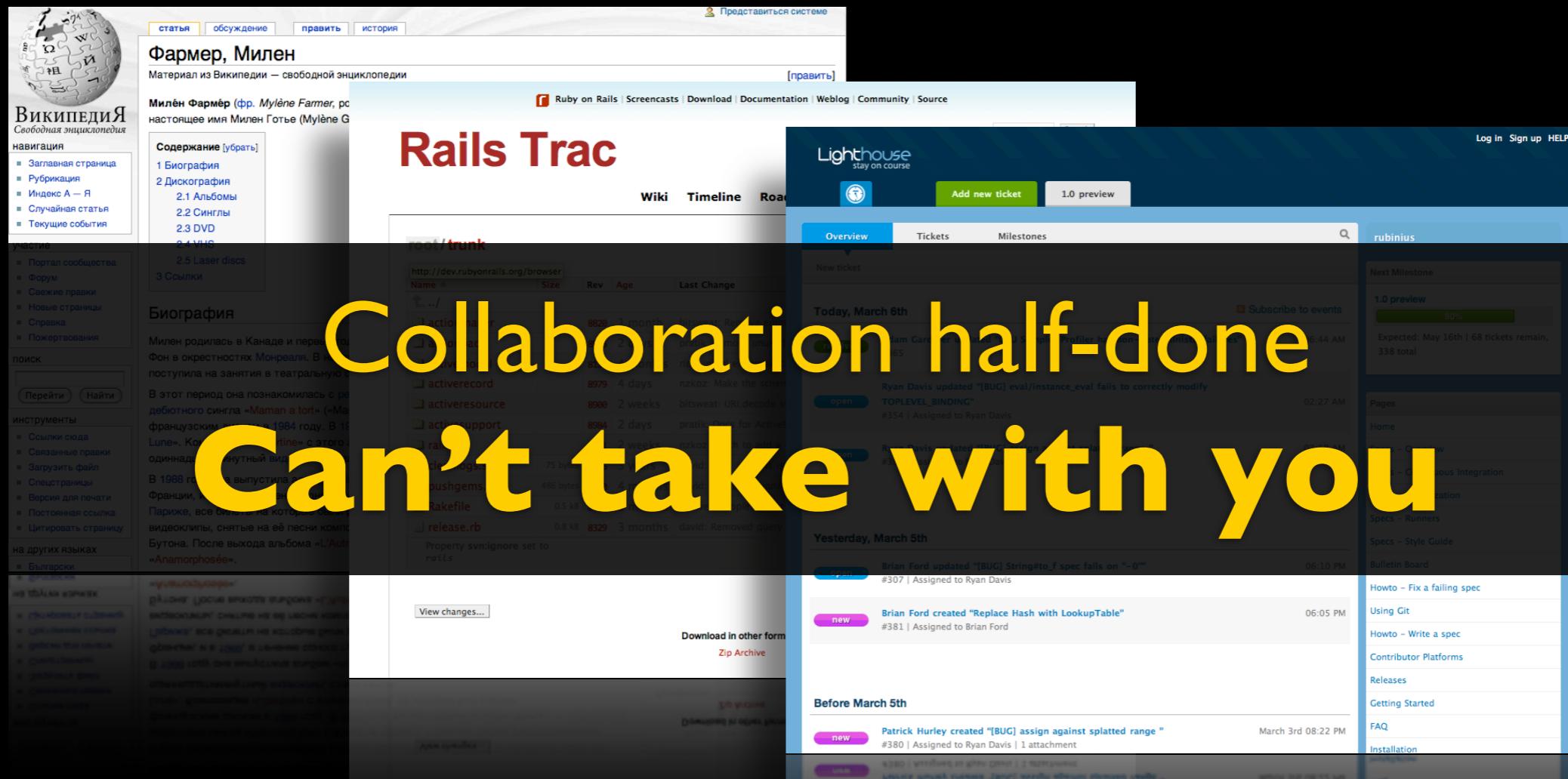
- Wikipedia:** A screenshot of the Russian Wikipedia page for "Фармер, Милен". It shows the article content, sidebar navigation, and search bar.
- Subversion:** A screenshot of the Ruby on Rails Subversion repository browser for the "trunk" branch. It lists files and their details like size, revision, and last change.
- Bug Tracker:** A screenshot of the Lighthouse bug tracking system for the "Rails Trac" project. It shows a timeline of tickets updated on March 6th, 2011, including details like ticket ID, assignee, and description.

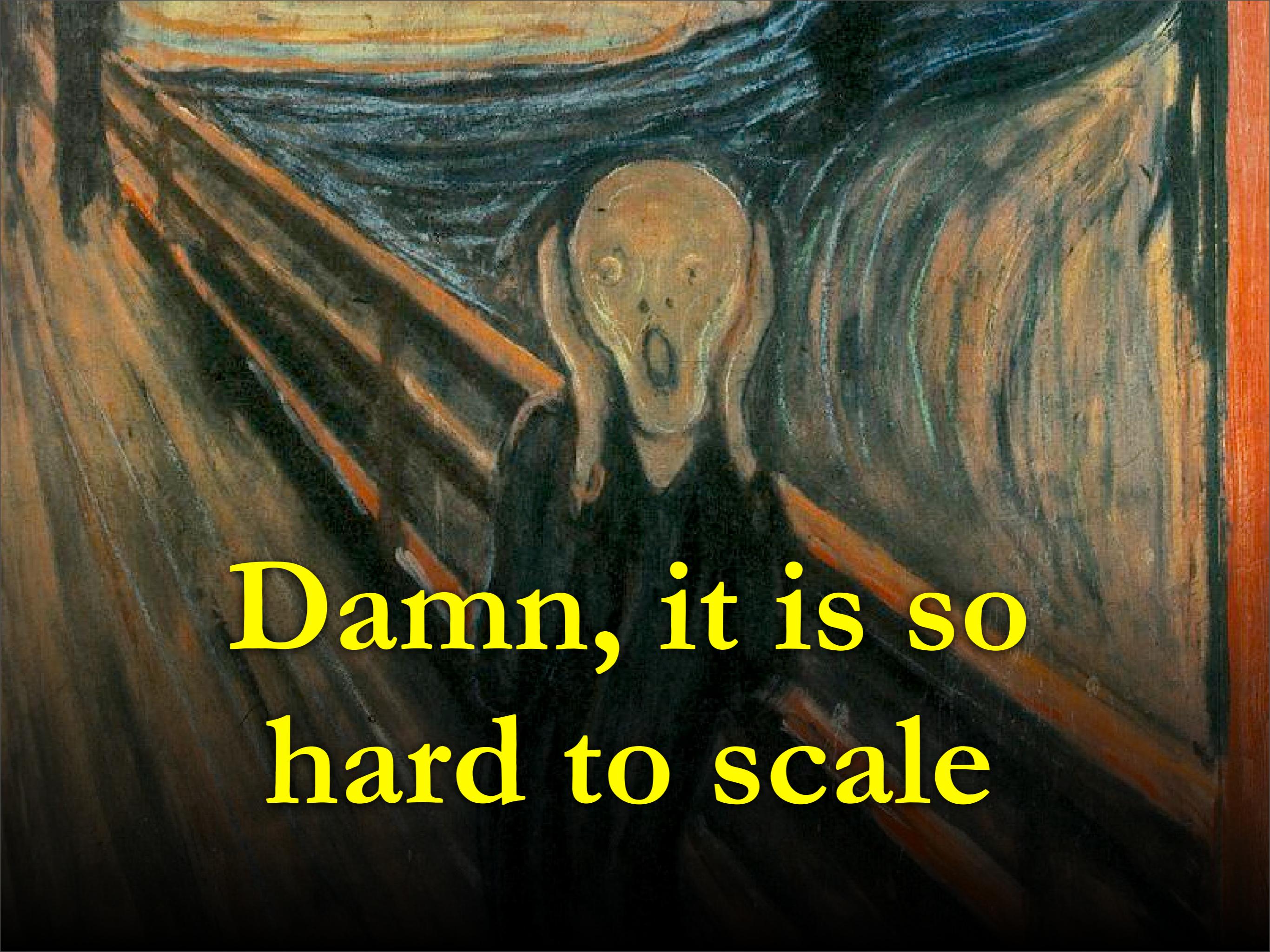
wikipedia, subversion, bugtrackers

Revision control

The image is a collage of three screenshots illustrating revision control:

- Wikipedia Article Screenshot:** A screenshot of a Wikipedia article page for "Фармер, Милен". It shows the "History" tab selected, with a prominent yellow border around it. The page content includes a biography and discography section.
- Rails Trac Repository Screenshot:** A screenshot of a Git commit log for the 'trunk' branch of the Ruby on Rails repository. The log lists various commits with details like author, date, and message. Examples include "Adam Gardner updated 'CPU Sampler Profiler has non-deterministic failures'" and "Brian Ford updated '[BUG] Stringitto_f spec fails on '-0'".
- Lighthouse Ticket System Screenshot:** A screenshot of a ticket tracking interface for a project. It shows a timeline view with tickets for different days. Tickets are color-coded by status (e.g., green for resolved). A sidebar on the right provides navigation links for the project's documentation and resources.





Damn, it is so
hard to scale



Goals

Semistructured data

Decentralization

Universal scalability

Thanks!

```
$ git clone git://gitorious.org/strokedb/mainline.git strokedb
```