

# StrokeDB

Yet Another Database?

*Yuri Rashkovskii & Oleg Andreev*

2008

# Who's talking?

Yurii



Oleg



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.

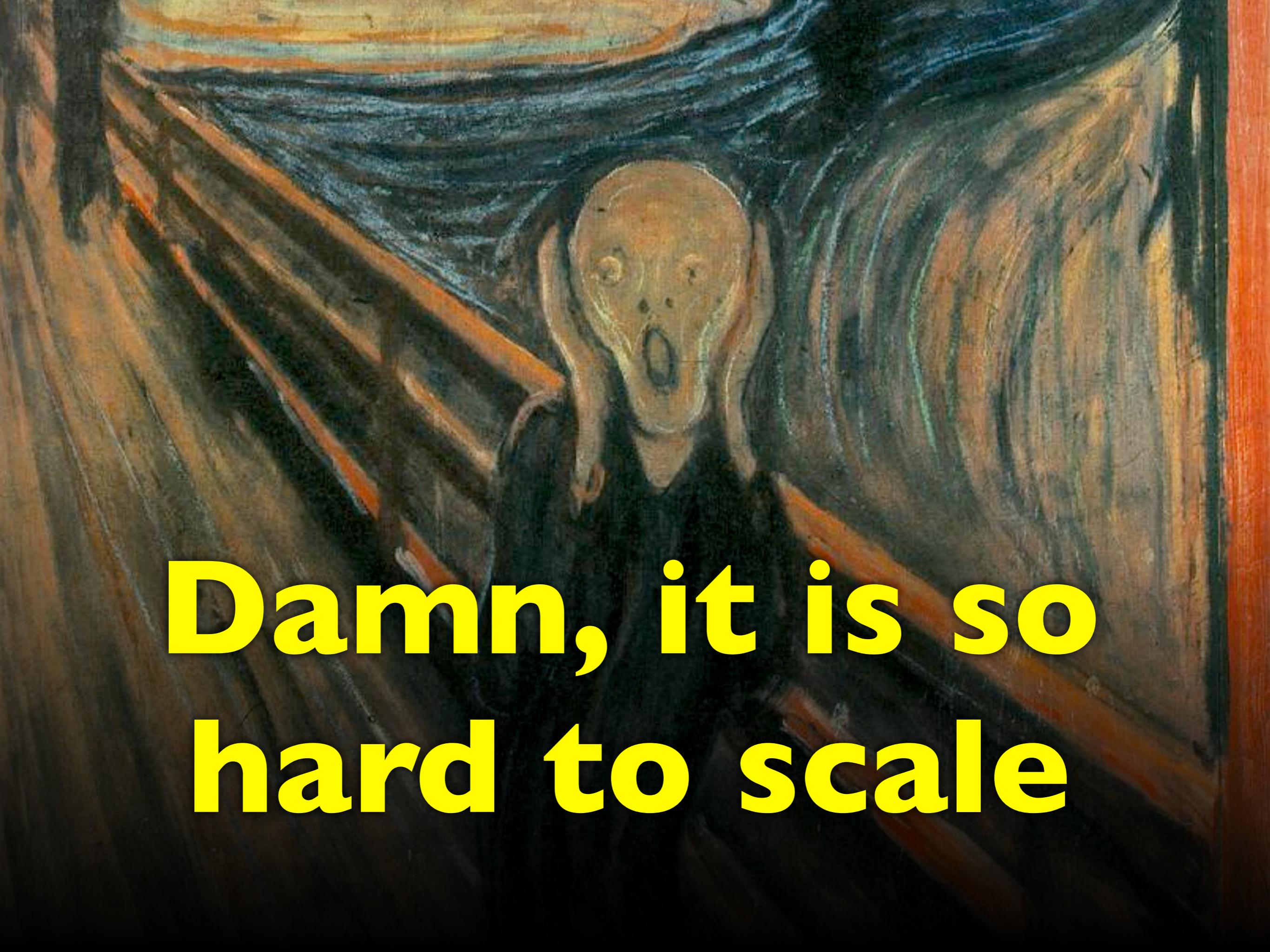
**Trees, maps and  
sortables are hard  
with RDBMS**



We need  
flexible  
data formats

The data I need is  
somewhere else



A close-up photograph of a stack of many colorful, textured books. The spines of the books are visible, showing various patterns and colors, including shades of blue, green, yellow, and orange. The lighting is dramatic, highlighting the texture of the book covers.

Damn, it is so  
hard to scale



**Goals**

Semistructured data

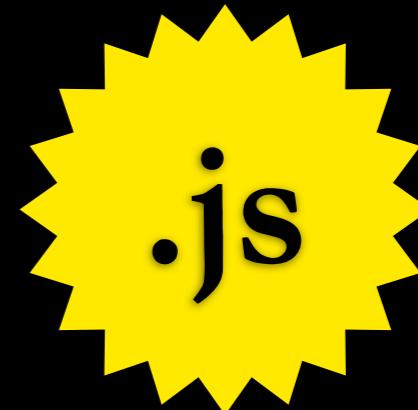
Decentralization

Universal scalability

# document.rb



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



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



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

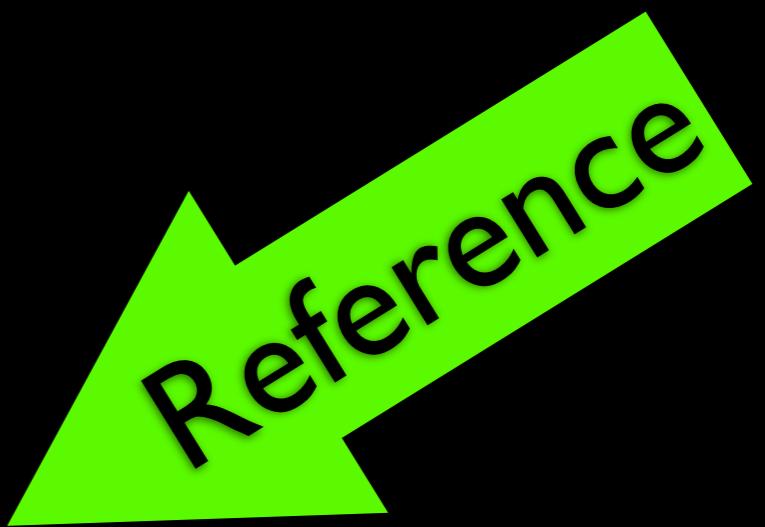


```
{  
  "name" => "Yuri",  
  "height" => 180,  
  "interests" => ["ruby", "lisp"]  
}
```

```
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”=>“Yuri” ...>**

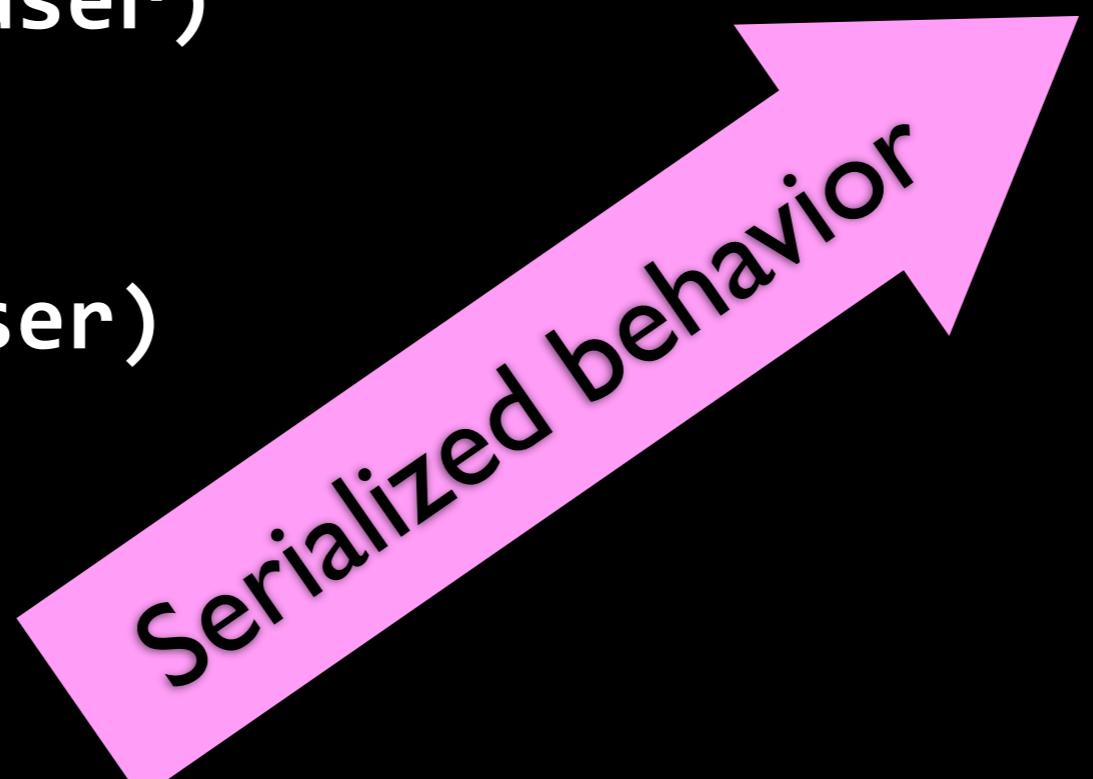
`person.inspect`



`#<{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
```



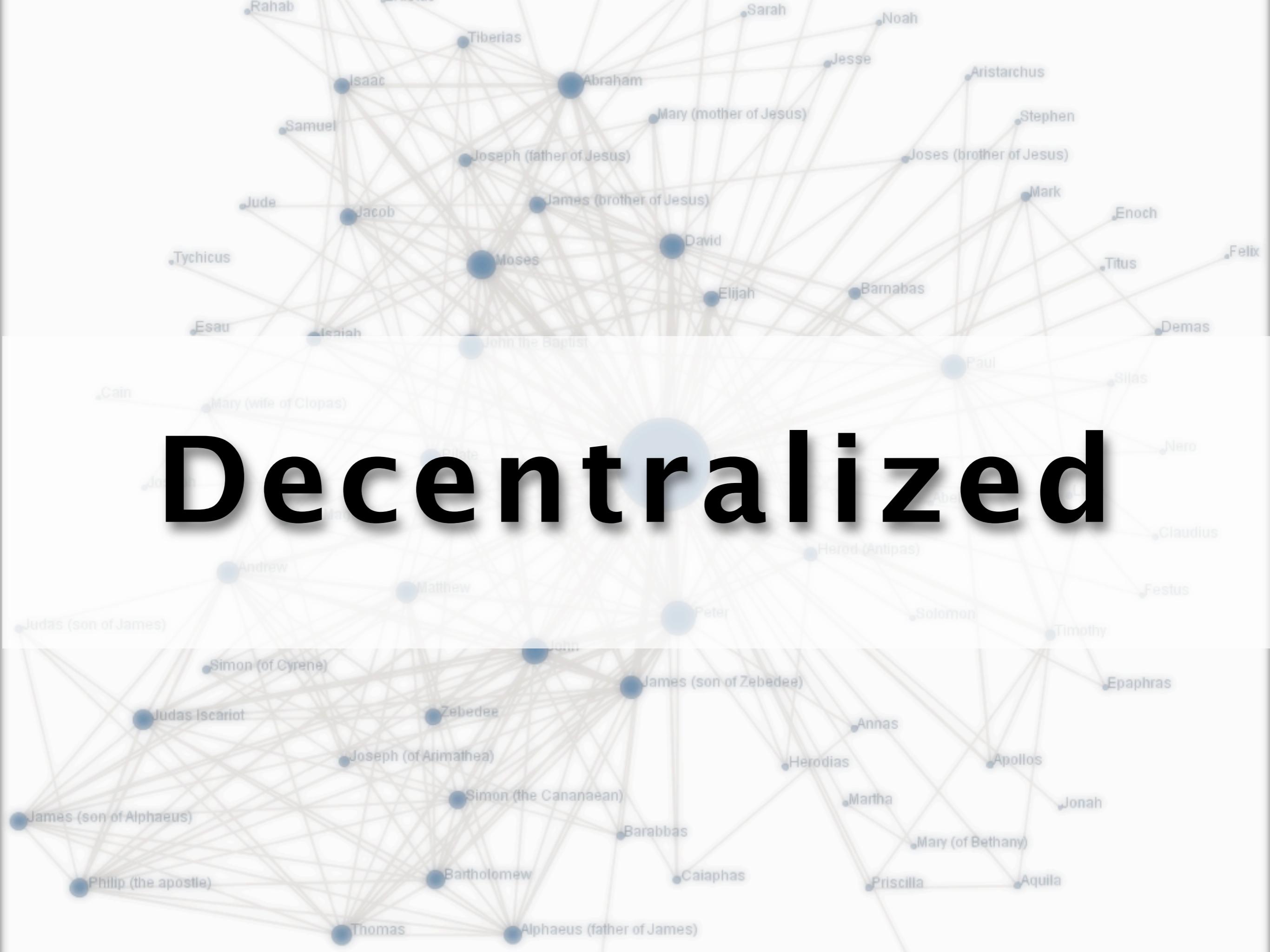
Serialized behavior

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



Executable code

# Decentralized



**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 software interfaces:

- Wikipedia (Russian version):** Shows the article page for "Фармер, Милен". It includes the standard Wikipedia navigation bar at the top, a sidebar with links like "Заглавная страница", "Рубрикация", and "Индекс А – Я", and a detailed content section about Mylène Farmer's biography and discography.
- Subversion (SVN) browser interface:** Shows a list of files and revisions in the "root/trunk" directory. The table includes columns for Name, Size, Rev, Age, and Last Change. Entries include "actionmailer", "actionpack", "activemodel", etc., with revision numbers ranging from 8118 to 8921.
- Lighthouse Bug Tracker:** Shows a timeline of tickets for the "Rails Trac" project. The interface includes sections for "Overview", "Tickets", and "Milestones". Tickets are listed by date (e.g., "Today, March 6th", "Yesterday, March 5th", "Before March 5th"). A sidebar on the right shows the "Next Milestone" progress (80%) and a list of "Pages" related to the project.

# wikipedia, subversion, bugtrackers

# Revision control

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

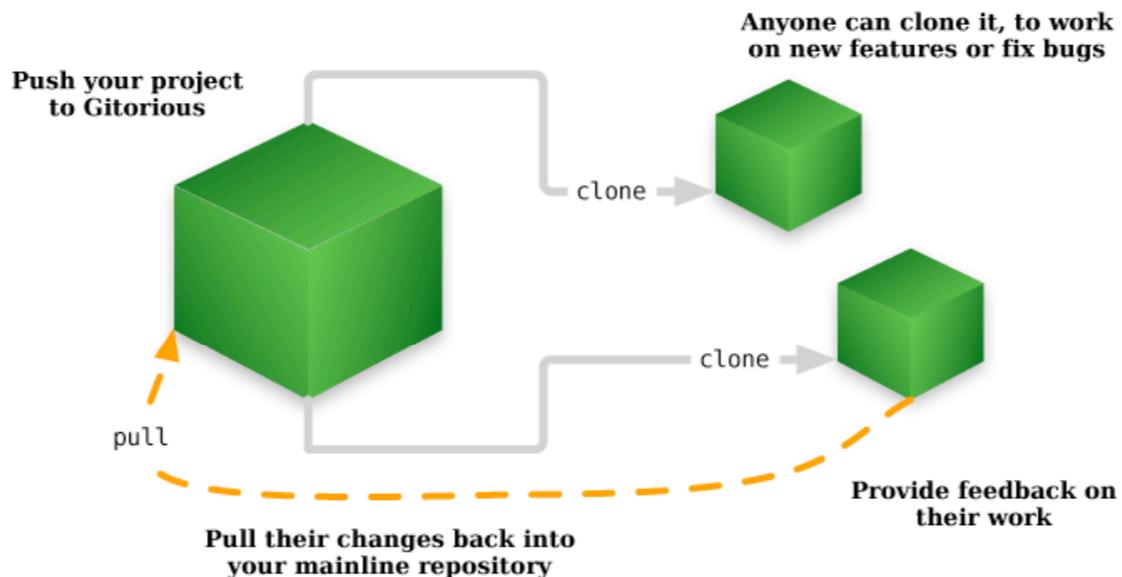
- Wikipedia Edit History:** A screenshot of a Wikipedia article page for "Фармер, Милен". It shows the "History" tab with a list of edits, including author names like "bitsweat", "pratik", and "david", and details such as "Last Change" and "Size".
- Rails Trac Ticket System:** A screenshot of the Rails Trac interface. It shows a list of tickets under the "root/trunk" category. Tickets include "actionmailer", "actionpack", "activemodel", "activerecord", "cleanlogs.sh", "pushgems.rb", "Rakefile", and "release.rb". Each ticket has columns for "Name", "Size", "Rev", "Age", and "Last Change".
- Lighthouse Project Management:** A screenshot of the Lighthouse interface. It shows a timeline view with sections for "Today, March 6th" and "Yesterday, March 5th". Tickets are listed with their status (e.g., "resolved", "new"), assignees (e.g., "Adam Gardner", "Ryan Davis"), and descriptions (e.g., "CPU Sampler Profiler has non-deterministic failures", "Fix a failing spec").

**Collaboration half-done  
Can't take with you**

The image is a composite of three screenshots from open-source projects:

- Wikipedia Page for Mylène Farmer:** Shows the article structure with sections like "Содержание" (Content), "Биография" (Biography), and "Музыкальные альбомы" (Musical albums). It includes a sidebar with community links and a search bar.
- Ruby on Rails GitHub Repository (root/trunk):** Displays a list of files and their details, such as "active\_record.rb" (8979 rev, 4 days ago) and "release.rb" (8329 rev, 3 months ago). It features a "View changes..." button and download options ("Download in other form" and "Zip Archive").
- Lighthouse Ticket Tracking System (Rails Trac):** Shows a timeline of tickets. A prominent ticket from "Ryan Davis" is updated with the note "[BUG] eval/instance\_eval fails to correctly modify TOLEVEL\_BINDING". Other tickets are visible for "Brian Ford" and "Patrick Hurley". The interface includes tabs for "Overview", "Tickets", and "Milestones".

**Gitorious** aims to provide a great way of doing distributed opensource code collaboration



Creating a user account allows you to create your own project or participate in the development of any project.

## Newest projects

[Projeto de Teste](#) Teste de GIT

[pedidos](#) Web application for management of employee requests

[Sputnik](#) Sputnik is a wiki written in Lua. It is also a simple web application for managing change control like applicati...

[test\\_project\\_yuri](#)

[twitterero](#) Bot para el canal #kubuntu-es de la red de irc freenode, su objetivo es publicar mensajes del can...

# Git

...an das Jäger der Leidenschaften sind sie überzeugt, dass es keine bessere Art der Selbstverwirklichung gibt als die sozialen Beziehungen mit anderen Menschen zu pflegen.

...für viele Jäger ist es eine Art der Identität, die sie von anderen unterscheidet.

...die Jäger sind überzeugt, dass es keine bessere Art der Identität gibt als die sozialen Beziehungen mit anderen Menschen zu pflegen.

...dass es für viele Jäger eine Art der Identität ist, die sie von anderen unterscheidet.

...die Jäger sind überzeugt, dass es keine bessere Art der Identität gibt als die sozialen Beziehungen mit anderen Menschen zu pflegen.

...die Jäger sind überzeugt, dass es keine bessere Art der Identität gibt als die sozialen Beziehungen mit anderen Menschen zu pflegen.

# Principles

Clone data

Track revisions

Merge changes

# Possible applications

Backup

Desktop apps to webapps

Next-gen content management

distributed version control is cool, but  
**how's that done?**

Document has a tail of **previous versions**.

Versions form a **tree of revisions**, not a list.

**Reference** is set for a particular version.

**Custom recursive merge** for nested data structures.

distributed version control is cool, but  
**how's that done?**

Document has a tail of **previous versions**.

Versions form a **tree of revisions**, not a list.

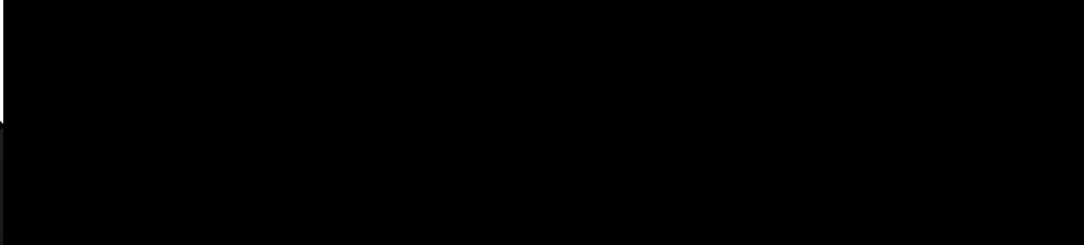
**Reference** is set for a particular version.

**Custom recursive merge** for nested data structures.

Under heavy development



**Scaling**



**scaling**

# Chunkable data

B-trees — No!

Skiplists — Yes!

# Networking

DRb *for poor-man apps*

EventMachine *Python's Twisted in Ruby*

# Data management

## Chunks distribution

## Recovery

## Runtime analysis

## *Disclaimer*

**Scalability features are  
mostly experimental by now.**

developers!  
developers!  
developers!



# For your pleasure

Ruby 1.8.6 and JRuby

script/console

RSpec examples, benchmarks

Hosted on gitorious and github

MIT License

# Near future

Improved indexes

Advanced networking built-in

Rubinius compatibility

Ruby 1.9/2.0 compatibility

The end game  
wutz on ur mind, guyz?!

Web 2.0 → Web 3.11

Users finally have generated content

It's time to do something with it

# All-in-one portals



# Mobility

My data is mine

“Git” meets end-users

REST & XML get sane applications

Linux

Nginx

StrokeDB

Rubinius

# Thanks!

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