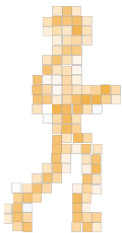


# FAST, FLEXIBLE AND FUN: REVISION CONTROL WITH MERCURIAL

Martin Geisler  
<mg@aragost.com>

aragost Trifork Geek Night in Zurich  
March 24, 2010

Link to presentation: <http://mercurial.ch/>



## ABOUT THE SPEAKER

Martin Geisler:

- ▶ core Mercurial developer:
  - ▶ reviews patches from the community
  - ▶ helps users in our IRC channel
- ▶ PhD in Computer Science from Aarhus University, Denmark
- ▶ now working at aragost Trifork, Switzerland



# OUTLINE

## INTRODUCTION

## USING MERCURIAL

- Workflows

- Branches

- The Underlying Model

- Using History

## COOL EXTENSIONS

- Changing History

- Talking to Other Systems

- Third-Party Tools

## DEMONSTRATION

## WRAPPING UP



# OUTLINE

## INTRODUCTION

### USING MERCURIAL

Workflows

Branches

The Underlying Model

Using History

### COOL EXTENSIONS

Changing History

Talking to Other Systems

Third-Party Tools

### DEMONSTRATION

### WRAPPING UP



# WHAT IS MERCURIAL?

Main features:

- ▶ fast, distributed revision control system
- ▶ robust support for branching **and** merging
- ▶ very flexible and extensible

Strong focus on back- and forwards compatibility:

- ▶ new clients can read/write all old formats on disk
- ▶ old clients can read/write to all new servers

Strong focus on data safety:

- ▶ files are not rewritten, only appended
- ▶ easier to recover from disk crashes



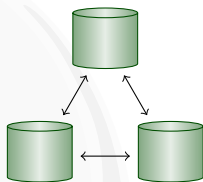
# WHY DISTRIBUTED?

Distributed revision control gives you:

- ▶ offline commits
- ▶ rich set of fast local operations

Derived effects:

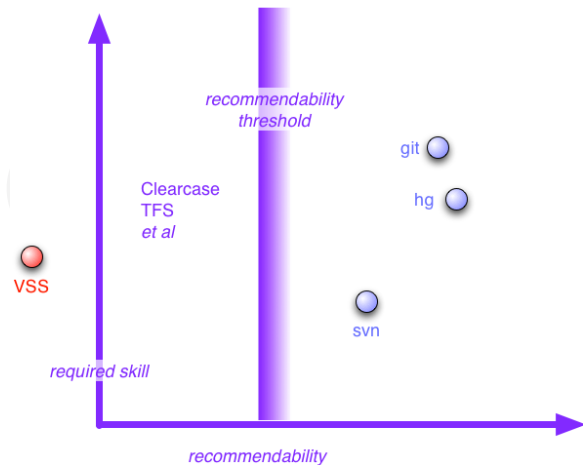
- ▶ fine-grained commits
- ▶ searchable history
- ▶ branching and merging become a natural task (not something to be feared)
- ▶ enables better workflows



# TESTIMONIALS

Martin Fowler, 2010:

<http://martinfowler.com/>



# TESTIMONIALS

Joel Spolsky, 2010:

<http://www.joelonsoftware.com/>

Mercurial is better than Subversion.

It is a better way of working on source code with a team. It is a better way of working on source code by yourself.

It is just better.

Google, 2008:

<http://code.google.com/>

In terms of implementation effort, Mercurial has a clear advantage due to its efficient HTTP transport protocol.

In terms of features, Git is more powerful, but this tends to be offset by it being more complicated to use.



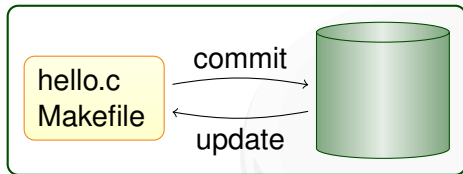
# WHO IS USING IT?

Mercurial is used by:

- ▶ Sun for Java, OpenSolaris, NetBeans, OpenOffice, ...
- ▶ Mozilla for Firefox, Thunderbird, ...
- ▶ Google
- ▶ many more...



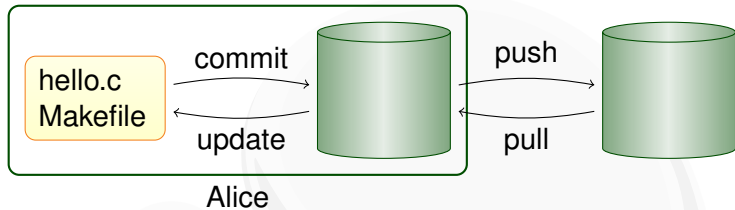
# KEY CONCEPTS



Alice



# KEY CONCEPTS



# KEY MERCURIAL COMMANDS

## Local commands:

- ▶ **hg commit**: save a snapshot into the current repository.
- ▶ **hg update**: checkout revision into working directory.
- ▶ **hg merge**: join different lines of history.

## Network commands:

- ▶ **hg pull**: retrieve changesets from another repository.
- ▶ **hg push**: send your changesets to another repository.



# OUTLINE

## INTRODUCTION

## USING MERCURIAL

Workflows

Branches

The Underlying Model

Using History

## COOL EXTENSIONS

Changing History

Talking to Other Systems

Third-Party Tools

## DEMONSTRATION

## WRAPPING UP



# OUTLINE

## INTRODUCTION

## USING MERCURIAL

### Workflows

Branches

The Underlying Model

Using History

## COOL EXTENSIONS

Changing History

Talking to Other Systems

Third-Party Tools

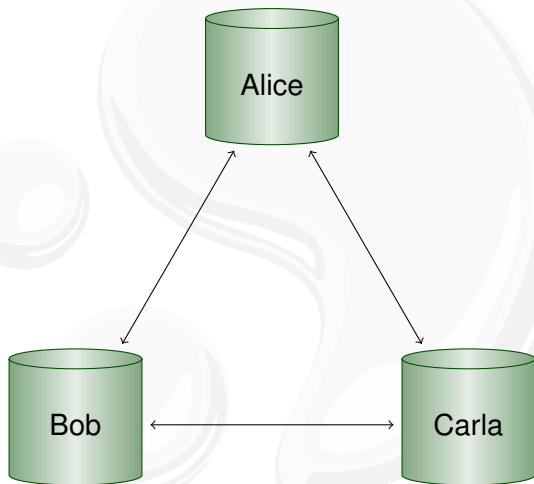
## DEMONSTRATION

## WRAPPING UP



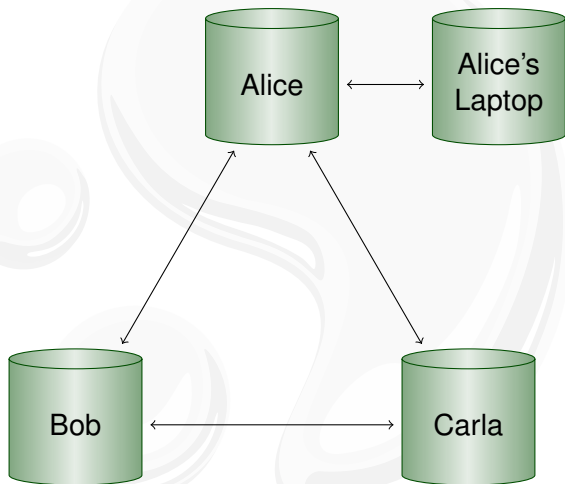
# DISTRIBUTED REVISION CONTROL

Mercurial duplicates the history on many servers:



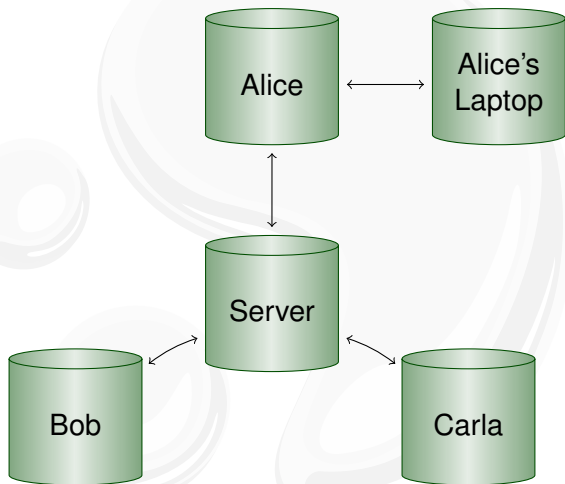
# DISTRIBUTED REVISION CONTROL

Mercurial duplicates the history on many servers:



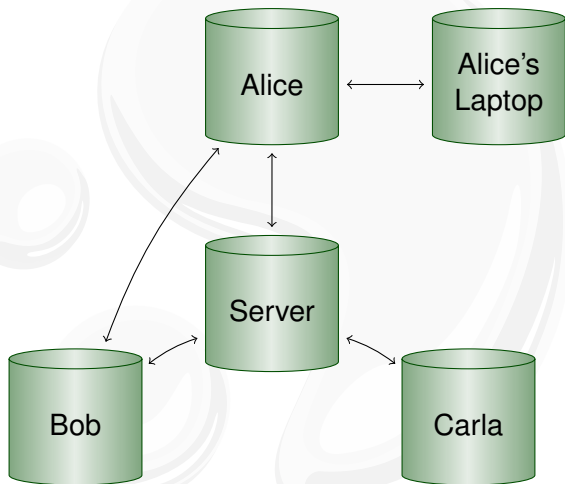
# DISTRIBUTED REVISION CONTROL

Mercurial duplicates the history on many servers:



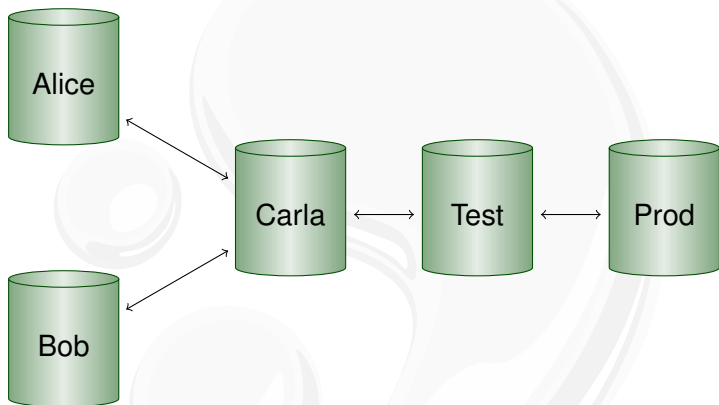
# DISTRIBUTED REVISION CONTROL

Mercurial duplicates the history on many servers:



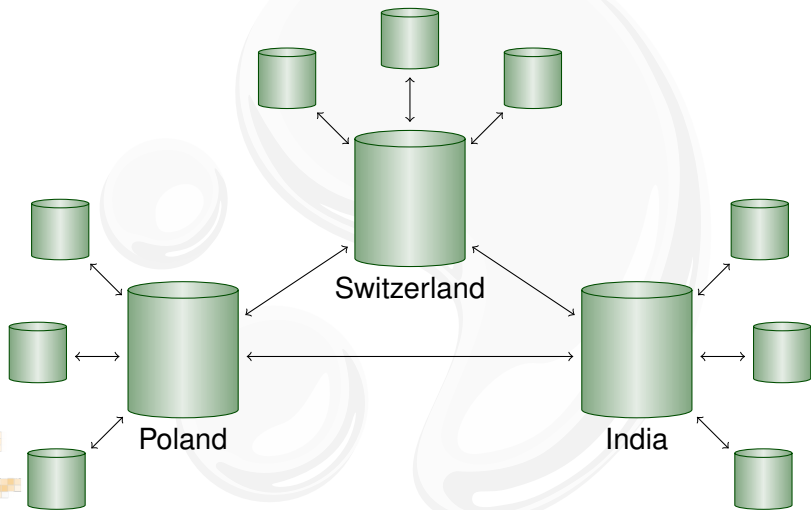
# WORKFLOW IN A TEAM

Mercurial scales from a single team... :



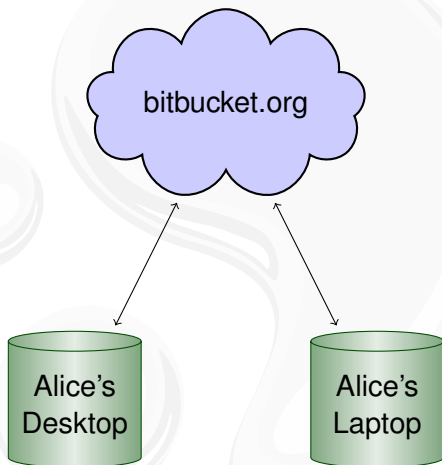
# WORKFLOW BETWEEN COMPANY DIVISIONS

... to enterprise-wide development... :



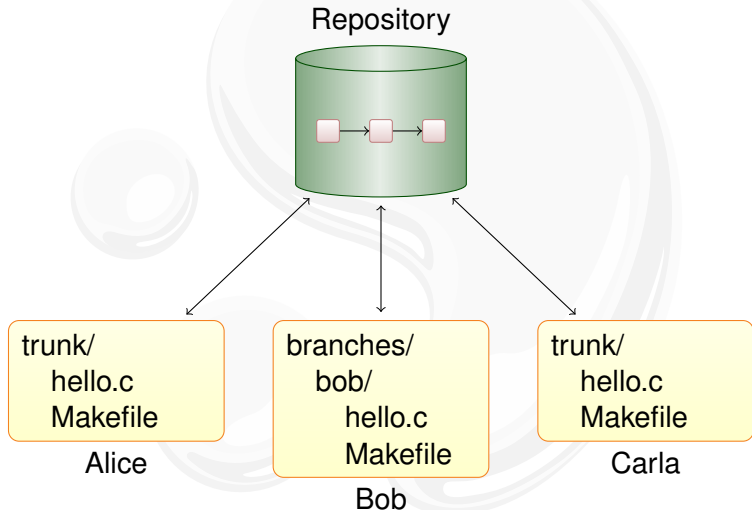
# WORKFLOW BETWEEN TWO COMPUTERS

... to working with yourself:



# CENTRALIZED REVISION CONTROL

Subversion uses a single server:



# OUTLINE

## INTRODUCTION

## USING MERCURIAL

Workflows

**Branches**

The Underlying Model

Using History

## COOL EXTENSIONS

Changing History

Talking to Other Systems

Third-Party Tools

## DEMONSTRATION

## WRAPPING UP



# BRANCHES

A key concept:

- ▶ parallel lines of development
- ▶ used to track releases
- ▶ used to isolate disruptive changes



# BRANCHES

A key concept:

- ▶ parallel lines of development
- ▶ used to track releases
- ▶ used to isolate disruptive changes



# BRANCHES

A key concept:

- ▶ parallel lines of development
- ▶ used to track releases
- ▶ used to isolate disruptive changes



# BRANCHES

A key concept:

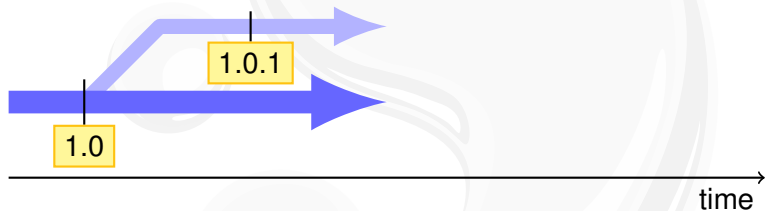
- ▶ parallel lines of development
- ▶ used to track releases
- ▶ used to isolate disruptive changes



# BRANCHES

A key concept:

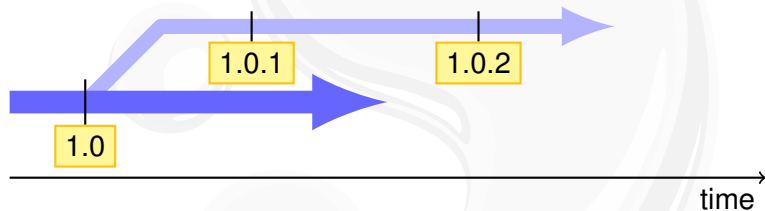
- ▶ parallel lines of development
- ▶ used to track releases
- ▶ used to isolate disruptive changes



# BRANCHES

A key concept:

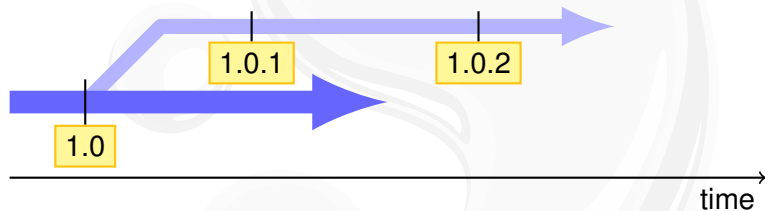
- ▶ parallel lines of development
- ▶ used to track releases
- ▶ used to isolate disruptive changes



# MERGING

The opposite of branching:

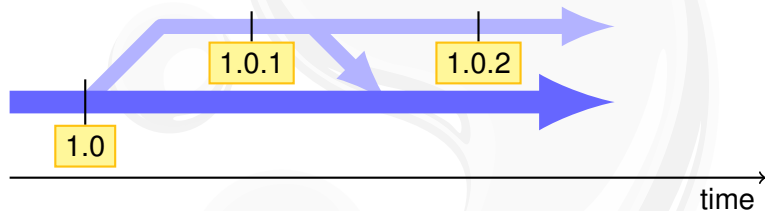
- ▶ combines two branches
- ▶ used to merge back bugfixes
- ▶ used to integrate feature branches



# MERGING

The opposite of branching:

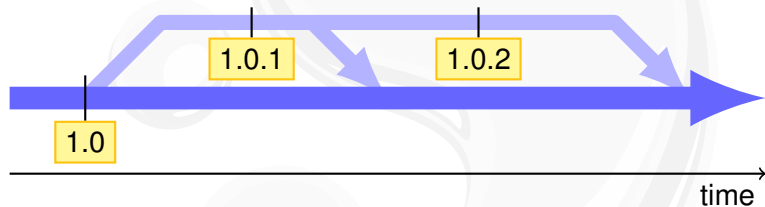
- ▶ combines two branches
- ▶ used to merge back bugfixes
- ▶ used to integrate feature branches



# MERGING

The opposite of branching:

- ▶ combines two branches
- ▶ used to merge back bugfixes
- ▶ used to integrate feature branches



## BRANCHES IN SVN

Subversion knows nothing about branches!

- ▶ but SVN has a cheap copy mechanism
- ▶ used for tags and branches



# BRANCHES IN SVN

Subversion knows nothing about branches!

- ▶ but SVN has a cheap copy mechanism
- ▶ used for tags and branches

r10

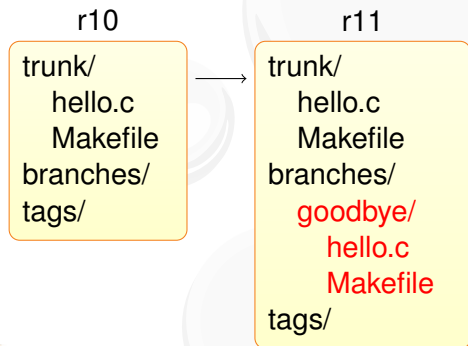
```
trunk/  
  hello.c  
  Makefile  
branches/  
tags/
```



## BRANCHES IN SVN

Subversion knows nothing about branches!

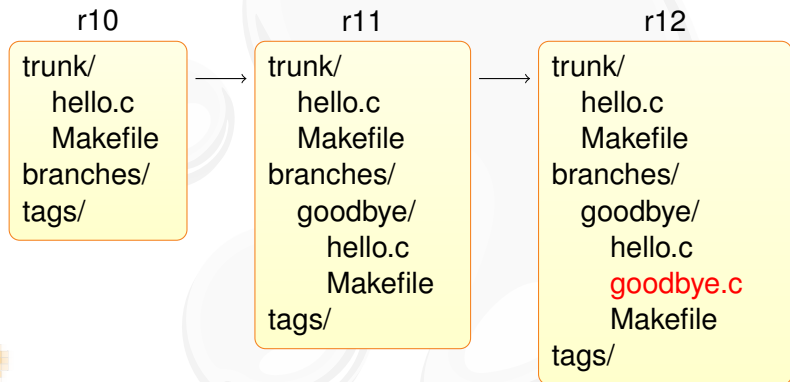
- ▶ but SVN has a cheap copy mechanism
- ▶ used for tags and branches



## BRANCHES IN SVN

Subversion knows nothing about branches!

- ▶ but SVN has a cheap copy mechanism
- ▶ used for tags and branches



# MERGING BRANCHES IN SVN

The support is incomplete and fragile:

- ▶ renamed files are not merged correctly
- ▶ old clients will not update the merge info



## MERGING BRANCHES IN SVN

The support is incomplete and fragile:

- ▶ renamed files are not merged correctly
- ▶ old clients will not update the merge info

The bottom line is that Subversion's merge-tracking feature has an **extremely complex** internal implementation, and the `svn:mergeinfo` property is the only window the user has into the machinery. Because the feature is **relatively new**, a numbers of edge cases and possible unexpected behaviors may pop up.

—*Version Control with Subversion*

(Mercurial has robust built-in support for merging branches.)



# OUTLINE

## INTRODUCTION

## USING MERCURIAL

Workflows

Branches

**The Underlying Model**

Using History

## COOL EXTENSIONS

Changing History

Talking to Other Systems

Third-Party Tools

## DEMONSTRATION

## WRAPPING UP





# OUTLINE

## INTRODUCTION

## USING MERCURIAL

Workflows

Branches

The Underlying Model

Using History

## COOL EXTENSIONS

Changing History

Talking to Other Systems

Third-Party Tools

## DEMONSTRATION

## WRAPPING UP



## BROWSING THE HISTORY OF A FILE

The `hg annotate` command is invaluable:

- ▶ you see when each line was introduced
- ▶ you can quickly jump back to earlier versions



# SEARCHING FILE CONTENT

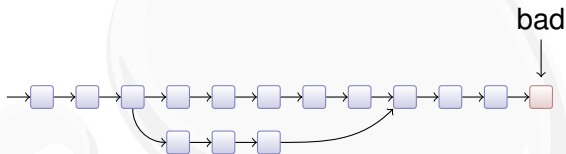
Ever wondered when a function was introduced?

- ▶ `hg grep` can help you!
- ▶ local access makes searching feasible



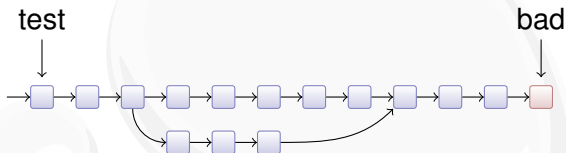
## REVISION GRAPH BISECTION

You've found a bug! When was it first introduced?  
Use `hg bisect` to mark good and bad revisions:



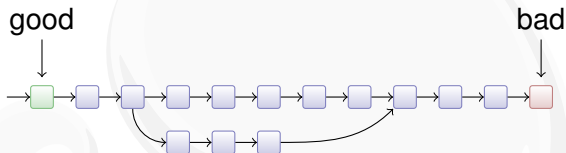
# REVISION GRAPH BISECTION

You've found a bug! When was it first introduced?  
Use `hg bisect` to mark good and bad revisions:



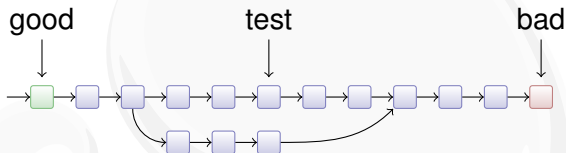
# REVISION GRAPH BISECTION

You've found a bug! When was it first introduced?  
Use `hg bisect` to mark good and bad revisions:



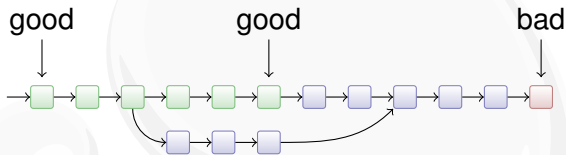
# REVISION GRAPH BISECTION

You've found a bug! When was it first introduced?  
Use `hg bisect` to mark good and bad revisions:



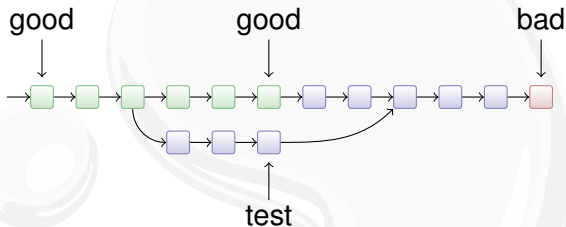
# REVISION GRAPH BISECTION

You've found a bug! When was it first introduced?  
Use `hg bisect` to mark good and bad revisions:



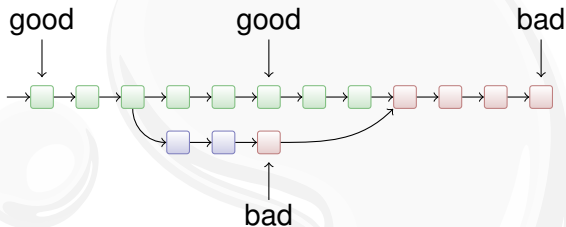
## REVISION GRAPH BISECTION

You've found a bug! When was it first introduced?  
Use `hg bisect` to mark good and bad revisions:



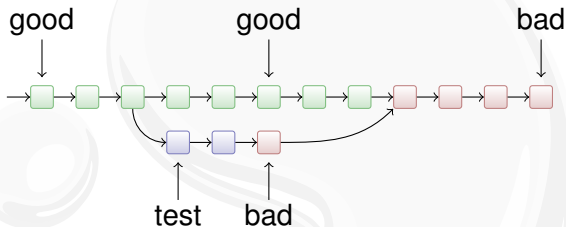
## REVISION GRAPH BISECTION

You've found a bug! When was it first introduced?  
Use `hg bisect` to mark good and bad revisions:



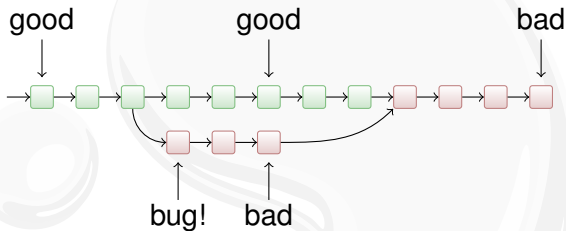
# REVISION GRAPH BISECTION

You've found a bug! When was it first introduced?  
Use `hg bisect` to mark good and bad revisions:



# REVISION GRAPH BISECTION

You've found a bug! When was it first introduced?  
Use `hg bisect` to mark good and bad revisions:



# OUTLINE

## INTRODUCTION

## USING MERCURIAL

Workflows

Branches

The Underlying Model

Using History

## COOL EXTENSIONS

Changing History

Talking to Other Systems

Third-Party Tools

## DEMONSTRATION

## WRAPPING UP



# MERCURIAL IS EXTENSIBLE

You can add new functionality to Mercurial:

- ▶ ships with 30+ extensions
- ▶ wiki lists 75+ extensions
- ▶ extensions can change basically everything
- ▶ helps to keep the core small and focused



# OUTLINE

## INTRODUCTION

## USING MERCURIAL

Workflows

Branches

The Underlying Model

Using History

## COOL EXTENSIONS

Changing History

Talking to Other Systems

Third-Party Tools

## DEMONSTRATION

## WRAPPING UP



# MOVING CHANGESETS AROUND

Tired of all those merges? Use the **rebase** extension!

- ▶ Revision graph:



# MOVING CHANGESETS AROUND

Tired of all those merges? Use the **rebase** extension!

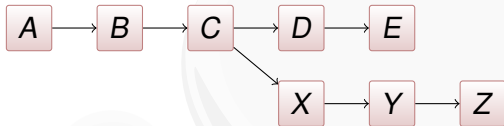
- ▶ Revision graph:



# MOVING CHANGESETS AROUND

Tired of all those merges? Use the **rebase** extension!

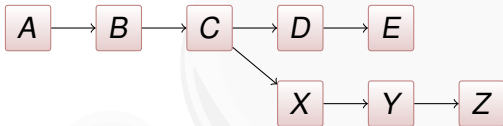
- ▶ Revision graph:



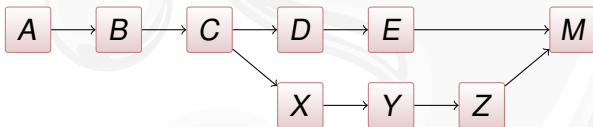
# MOVING CHANGESETS AROUND

Tired of all those merges? Use the **rebase** extension!

- ▶ Revision graph:



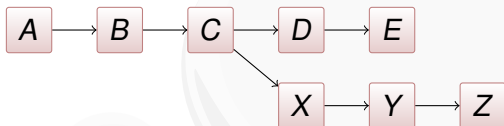
- ▶ Merge:



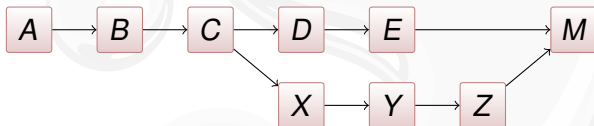
# MOVING CHANGESETS AROUND

Tired of all those merges? Use the **rebase** extension!

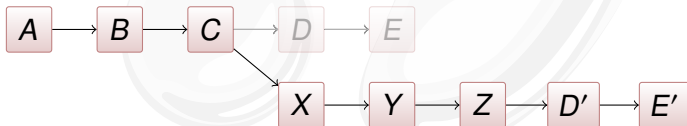
- ▶ Revision graph:



- ▶ Merge:



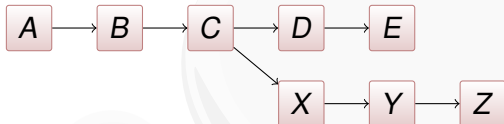
- ▶ Rebase:



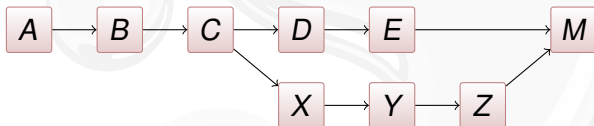
# MOVING CHANGESETS AROUND

Tired of all those merges? Use the **rebase** extension!

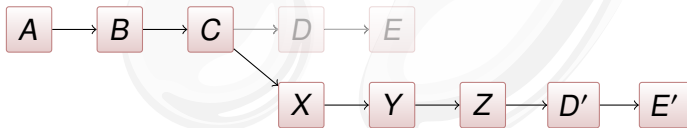
- ▶ Revision graph:



- ▶ Merge:



- ▶ Rebase:



- ▶ Beware: public changes should never be rebased.



# EDITING HISTORY

Inspired by `git rebase -i`, `histedit` lets you

- ▶ reorder changesets:



# EDITING HISTORY

Inspired by `git rebase -i`, `histedit` lets you

- ▶ reorder changesets:



- ▶ fold changesets:



# EDITING HISTORY

Inspired by `git rebase -i`, `histedit` lets you

- ▶ reorder changesets:



- ▶ fold changesets:



- ▶ drop changesets:



# EDITING HISTORY

Inspired by `git rebase -i`, `histedit` lets you

- ▶ reorder changesets:



- ▶ fold changesets:



- ▶ drop changesets:



- ▶ edit changesets:



# OUTLINE

## INTRODUCTION

## USING MERCURIAL

Workflows

Branches

The Underlying Model

Using History

## COOL EXTENSIONS

Changing History

Talking to Other Systems

Third-Party Tools

## DEMONSTRATION

## WRAPPING UP



# MIGRATING HISTORY

The **convert** extension can import history:

- ▶ CVS, SVN, Git, Bazaar, Darcs, ...
- ▶ incremental conversion
- ▶ many options for fiddling with branches, authors, ...



# MIGRATING HISTORY

The **convert** extension can import history:

- ▶ CVS, SVN, Git, Bazaar, Darcs, ...
- ▶ incremental conversion
- ▶ many options for fiddling with branches, authors, ...

Interestingly, **convert** can import from Mercurial:

- ▶ **--filemap** lets you exclude and rename files
- ▶ **--branchmap** lets you rename branches



# INTERFACING WITH SUBVERSION

The **hgsubversion** extension let's you:

- ▶ use **hg clone** on a SVN URL
- ▶ use **hg pull** to convert new SVN revisions
- ▶ use **hg push** to commit changesets to SVN server

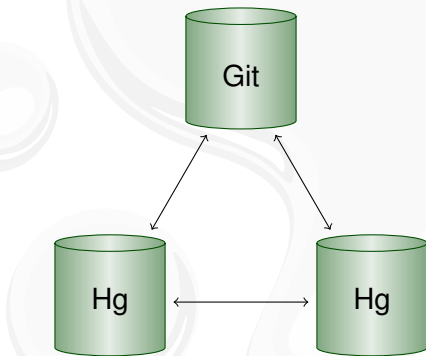
Goal: make **hg** a better Subversion client than **svn**!



# INTERFACING WITH GIT

Need to work on a Git repository? Try **hg-git**!

- ▶ Mercurial extension: you get the nice **hg** command line
- ▶ round-tripping: changeset hashes are preserved



# OUTLINE

## INTRODUCTION

## USING MERCURIAL

Workflows

Branches

The Underlying Model

Using History

## COOL EXTENSIONS

Changing History

Talking to Other Systems

Third-Party Tools

## DEMONSTRATION

## WRAPPING UP



## THIRD-PARTY TOOLS

Tools with Mercurial support:

- ▶ Shell integration: TortoiseHg (Windows, Mac, Linux)
- ▶ IDEs: Eclipse, NetBeans, IntelliJ, Visual Studio, Emacs...
- ▶ Project Support: Trac, JIRA, Maven, Hudson, BuildBot...



# OUTLINE

## INTRODUCTION

## USING MERCURIAL

- Workflows

- Branches

- The Underlying Model

- Using History

## COOL EXTENSIONS

- Changing History

- Talking to Other Systems

- Third-Party Tools

## DEMONSTRATION

## WRAPPING UP





**Live Demo!**



# OUTLINE

## INTRODUCTION

## USING MERCURIAL

- Workflows

- Branches

- The Underlying Model

- Using History

## COOL EXTENSIONS

- Changing History

- Talking to Other Systems

- Third-Party Tools

## DEMONSTRATION

## WRAPPING UP



# MERCURIAL IN A NUTSHELL

Mercurial changes the way you develop:

- ▶ simple yet strong model for **both** branching and merging
- ▶ power tool instead of necessary evil
- ▶ light-weight and snappy



## MORE INFORMATION

- ▶ Mercurial homepage:  
<http://mercurial.selenic.com/>
- ▶ *Mercurial: The Definitive Guide*:  
<http://hgbook.red-bean.com/>
- ▶ Getting Started:  
<http://mercurial.ch/>  
<http://hginit.com/>
- ▶ Some free Mercurial hosting sites:  
<http://bitbucket.org/>  
<http://code.google.com/>  
<http://sourceforge.net/>  
<http://www.codeplex.com/> (Microsoft)



## CONTACT

Please get in touch if you have more questions or have considered using Mercurial in your organization:

- ▶ Email: [mg@aragost.com](mailto:mg@aragost.com)
- ▶ IRC: [mg](#) in [#mercurial](#) on [irc.freenode.net](http://irc.freenode.net)
- ▶ Link to presentation: <http://mercurial.ch/>



## CONTACT

Please get in touch if you have more questions or have considered using Mercurial in your organization:

- ▶ Email: [mg@aragost.com](mailto:mg@aragost.com)
- ▶ IRC: [mg](#) in [#mercurial](#) on [irc.freenode.net](http://irc.freenode.net)
- ▶ Link to presentation: <http://mercurial.ch/>

**Thank you!**

