

Ruby and Adobe Flex



An Introduction to Using
Adobe Flex 2 with Ruby on Rails 1.2.3

Peter Armstrong

August 30, 2007

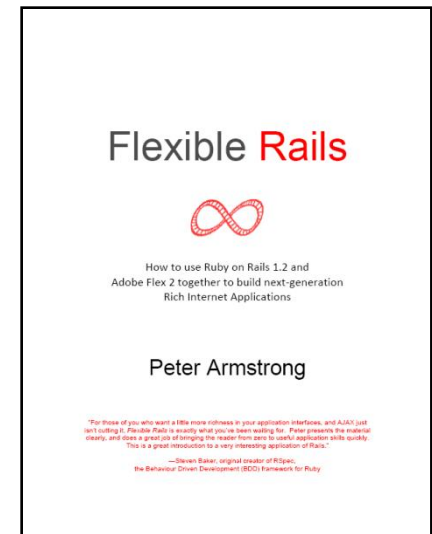
www.flexiblerails.com

Who Am I?

Pay my mortgage! [one slide of Shameless Self-Promotion]

www.flexiblerails.com

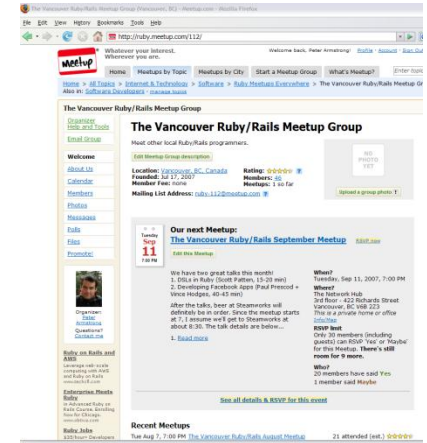
- the basis for this talk
- currently a PDF-only book
- sold on Lulu (currently ranked **#102** all time!)
- **\$20 USD (< \$22 CAD)**
- All book code is MIT-licensed
<http://www.flexiblerails.com/code-samples>
- content-complete “Beta Book”
- 497 pages



[I'm the organizer of this, so I'll plug this too.]

The Vancouver Ruby/Rails Meetup Group

- <http://ruby.meetup.com/112/>
- Monthly meetups (currently every 2nd Tuesday of the month)
- Over 20 people attended the first meetup earlier this month
- 46 members of the group currently
- Next meetup: Tuesday, Sept. 11 with 2 great talks:
 1. DSLs in Ruby (Scott Patten, 15-20 min)
 2. Developing Facebook Apps (Paul Prescod + Vince Hodges, 40-45 min)
- Basically, I'm copying the excellent job that Ross Ladell has done with the Vancouver Flash/Flex Meetup Group which I greatly enjoy belonging to.



Ruby

“Ruby is designed to make programming not only easy but also fun. It allows you to concentrate on the creative side of programming, with less stress.”

–Yukihiro Matsumoto

Rails

Why use Ruby on Rails
with Flex?

Developer Productivity

Ruby on Rails + _____

a) Flex

b) Hybrid: Flex + (HTML
& Ajax)

c) HTML & Ajax

d) All of the above

...in the same controller methods,
thanks to the REST support in Rails

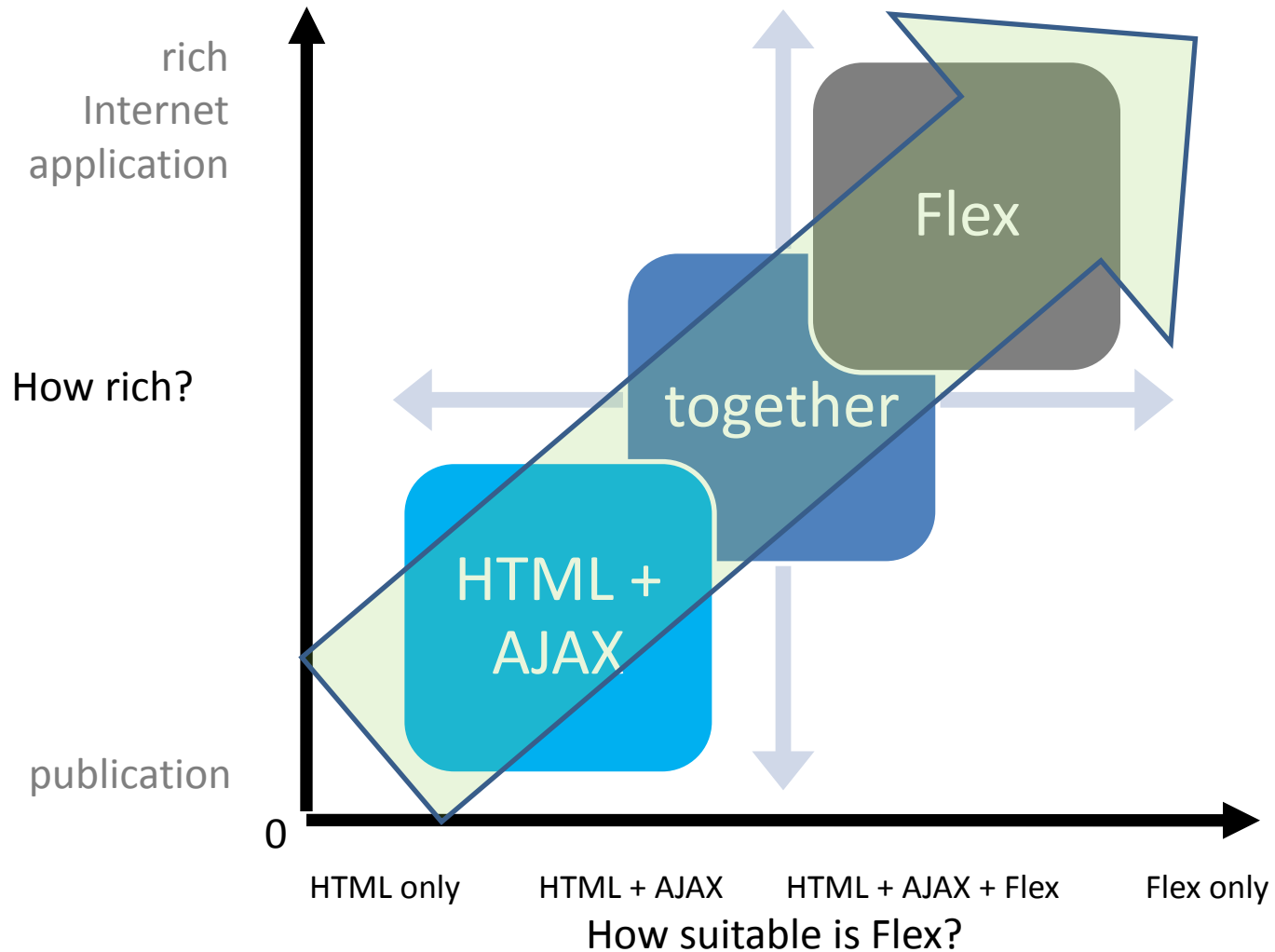
When should you use...

Flex only?

Flex + (HTML & Ajax)?

HTML & Ajax only?

How **rich** do you want (your UI) to be?



Disclaimer: This is MY opinion ONLY; I do NOT speak for my employer or Adobe or anyone else.

No (more) fluff!

How do Flex and
Rails fit together?

How do Flex and Rails fit together?

1. HTTPService

sending:

application/xml,

application/x-www-form-urlencoded

receiving:

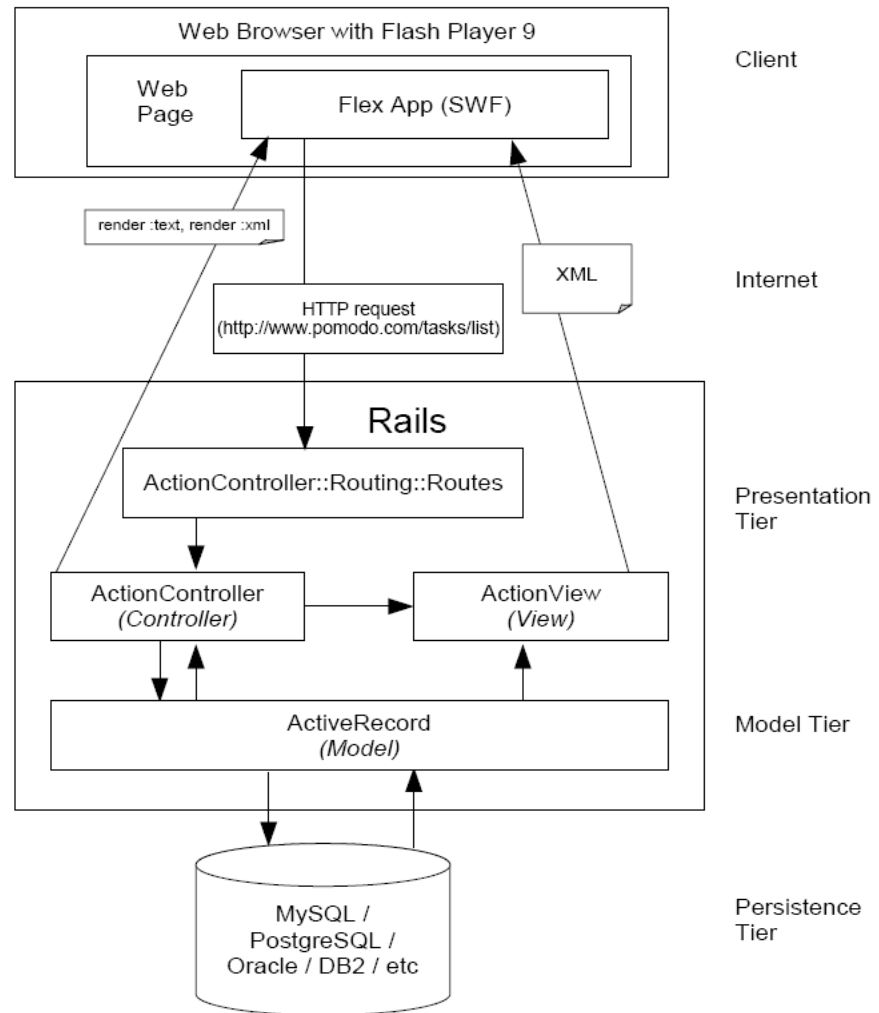
XML, JSON, text

2. AMF

WebORB for Rails

RubyAMF (standalone, Rails plugin)

How Flex and Rails fit together with HTTPService



HTTPService: Flex Sending XML

```
<mx:HTTPService
  id="svcAccountCreate" url="/users.xml"
  contentType="application/xml" resultFormat="e4x"
  method="POST" result="handleAccountCreateResult(event)">
  <mx:request>
    <user>
      <login>{newAccountLoginTI.text}</login>
      <email>{newAccountEmailTI.text}</email>
      <first_name>{newAccountFirstNameTI.text}</first_name>
      <last_name>{newAccountLastNameTI.text}</last_name>
      <password>{newAccountPasswordTI.text}</password>
    <password_confirmation>{newAccountConfirmPasswordTI.text}</password_co
      nfirmation>
    </user>
  </mx:request>
</mx:HTTPService>
```

HTTPService: Flex Sending Name-Value Pairs (like a normal HTTP POST) application/x-www-form-urlencoded

```
private function handleLoginClicked(event:MouseEvent):void {
    svcAccountLogin.send({login: loginTI.text,
        password: passwordTI.text});
}
]]>
</mx:Script>
    <mx:HTTPService
        id="svcAccountLogin"
        url="/sessions.xml"
        resultFormat="e4x"
        method="POST"
        result="handleAccountLoginResult(event)"/>
```

RESTful Rails Controllers

```
# POST /tasks
# POST /tasks.xml
def create
  @task = current_user.tasks.build params[:task]
  respond_to do |format|
    if @task.save
      format.html { redirect_to task_url(@task) }
      format.xml { render :xml => @task.to_xml }
    else
      format.html { render :action => "new" }
      format.xml { render :text => "save failed" }
    end
  end
end
end
```

Relationship between RESTful Rails Controllers*, CRUD, HTTP and SQL

#	Method	Sample URL paths	Pretend HTTP Method	Actual HTTP Method	Corresponding CRUD Method	Corresponding SQL Method
1	index	/tasks /tasks.xml	GET	GET	READ	SELECT
2	show	/tasks/1 /tasks/1.xml	GET	GET	READ	SELECT
3	new	/tasks/new	GET	GET	-	-
4	edit	/tasks/1;edit	GET	GET	READ	SELECT
5	create	/tasks /tasks.xml	POST	POST	CREATE	INSERT
6	update	/tasks/1 /tasks/1.xml	PUT	POST	UPDATE	UPDATE
7	destroy	/tasks/1 /tasks/1.xml	DELETE	POST	DELETE	DELETE

*I didn't say "REST" here, since this is focused on the Rails implementation of REST, not REST architecture in general. See the excellent *RESTful Web Services* for details: <http://www.crummy.com/writing/RESTful-Web-Services/>

DEMOS

Thank you!



<http://www.flexiblerails.com>

<http://ruby.meetup.com/112/>