



Re-graphing The Mental Model of The Rails Router



Vaidehi Joshi

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



Re-graphing The Mental Model of The Rails Router

Vaidehi Joshi

@vaidehijoshi

Rails
Conf
2018

PITTSBURGH

i work on  SKYLIGHT

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



off the happy path

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



abstractions

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



rails router

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



routing basics

@vaidehijoshi



The image shows a screenshot of a video player window. On the left, there's a thumbnail of a woman speaking at a podium. The main content area has a yellow-to-red gradient background. On the left side of the slide, the text "Rails Conf 2018 PITTSBURGH" is displayed in large red font, with "PITTSBURGH" in smaller black font below it. To the right of this text is a graphic of a red building with a white envelope icon on top, labeled "post office" with a black arrow pointing to it. In the bottom right corner of the slide, the handle "@vaidehijoshi" is visible.

Rails Conf 2018
PITTSBURGH

post office

@vaidehijoshi

Rails Conf 2018
PITTSBURGH

The video content displays a diagram illustrating the Rails router's mental model. A large red house icon is shown on the left, with a white envelope icon inside the roof. Below the house are three white square windows. Dotted arrows point from a stack of five smaller red envelope icons on the left towards several smaller red house icons arranged in a cluster on the right. The background of the slide has a yellow-to-red gradient.

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



the router knows where to
send you

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



```
$ rails routes
```

Verb	URI Pattern	Controller#Action
GET	/articles(.:format)	articles#index
POST	/articles(.:format)	articles#create
GET	/articles/new(.:format)	articles#new
GET	/articles/:id/edit(.:format)	articles#edit
GET	/articles/:id(.:format)	articles#show
PATCH	/articles/:id(.:format)	articles#update
PUT	/articles/:id(.:format)	articles#update
DELETE	/articles/:id(.:format)	articles#destroy

@vaidehijoshi



routes.rb

@vaidehijoshi



```
Rails.application.routes.draw do
  root 'welcome#index'

  get 'recipes/:id', to: 'recipes#show'
  resources :articles
end
```

@vaidehijoshi



Rails Conf 2018

PITTSBURGH

\$ rails middleware

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



```
$ rails middleware

use Rack::Sendfile
use ActionDispatch::Static
use ActionDispatch::Executor
use ActiveSupport::Cache::Strategy::LocalCache::Middleware
...
use Rack::Head
use Rack::ConditionalGet
use Rack::ETag
use Rack::TempfileReaper
run ExampleApp::Application.routes
```

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



what's middleware again?

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



what's rack again?

@vaidehijoshi

Rails
Conf
2018

PITTSBURGH

responds to call
(like a proc)

@vaidehijoshi



```
{ [200, {}, ["Hello World"]]
```

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



ProTip™:

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



ProTip™:

controller actions are
just rack endpoints

@vaidehijoshi



```
Rails.application.routes.draw do
  root 'welcome#home'

  root to: proc { [200, {}, ["omg this works!"]] }
end
```



@vaidehijoshi

Rails Conf 2018

PITTSBURGH



```
$ rails middleware

use Rack::Sendfile
use ActionDispatch::Static
use ActionDispatch::Executor
use ActiveSupport::Cache::Strategy::LocalCache::Middleware
...
use Rack::Head
use Rack::ConditionalGet
use Rack::ETag
use Rack::TempfileReaper
run ExampleApp::Application.routes ←
```

@vaidehijoshi



Rails Conf 2018

PITTSBURGH



```
def routes
  # ...
  @routes
end
```

@vaidehijoshi



Rails Conf 2018

PITTSBURGH



```
get 'recipes/:id', to: 'recipes#show'  
post 'articles(.:format)', to: 'articles#create'  
post 'articles/new', to: 'articles#new'  
get 'articles/:id', to: 'articles#new'  
  
# even more routes here...
```

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



```
if request_path =~ /^articles$/
  # go to articles#index
elsif request_path =~ /^recipes$/
  # go to recipes#index
elsif
  # etc etc more looping ahhhhh!
else
  # route not found?
end
```

@vaidehijoshi

Rails Conf 2018

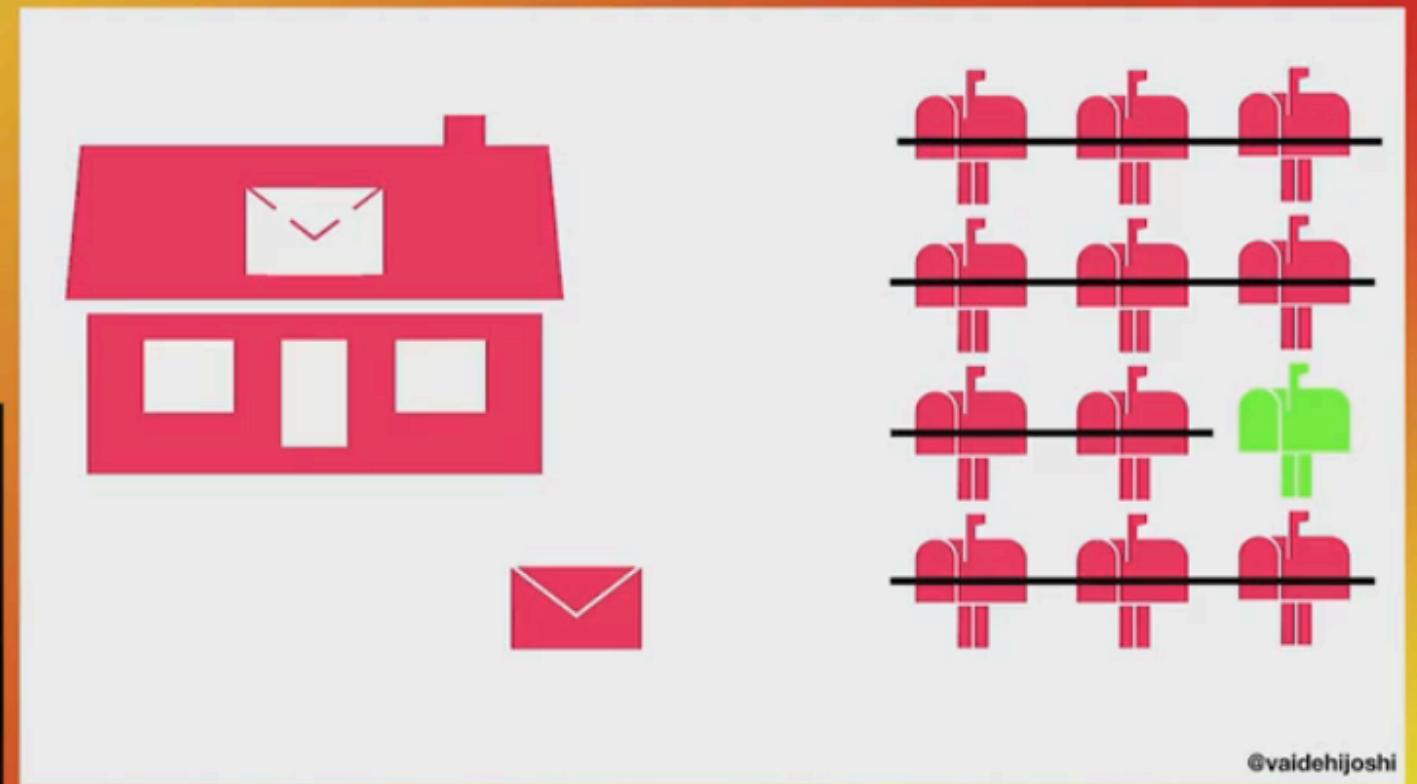
PITTSBURGH



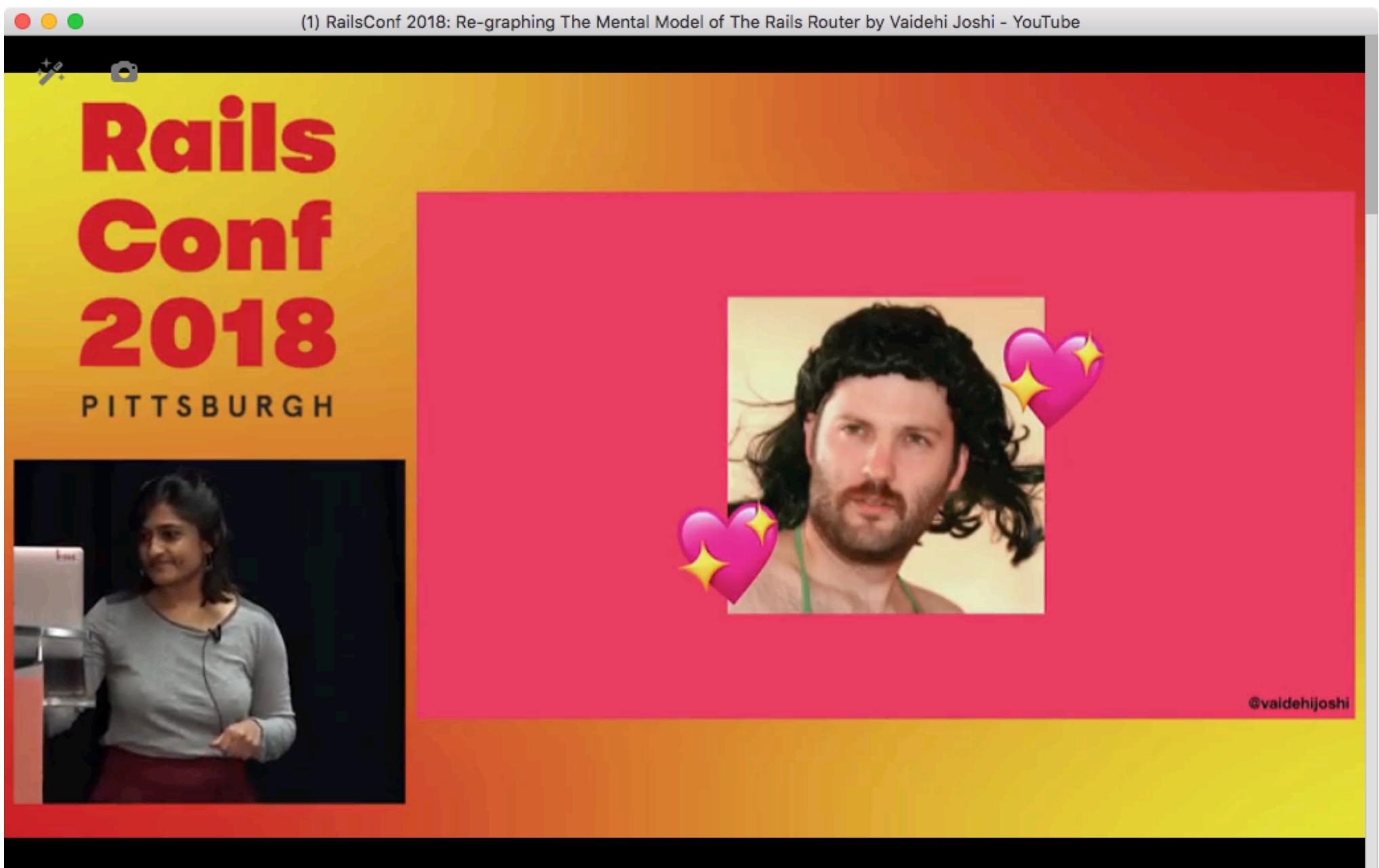
```
if request_path =~ /^articles$/
  # go to articles#index
elsif request_path =~ /^recipes$/
  # go to recipes#index
elsif
  # etc etc more looping ahhhhh!
else
  # route not found?
end
```

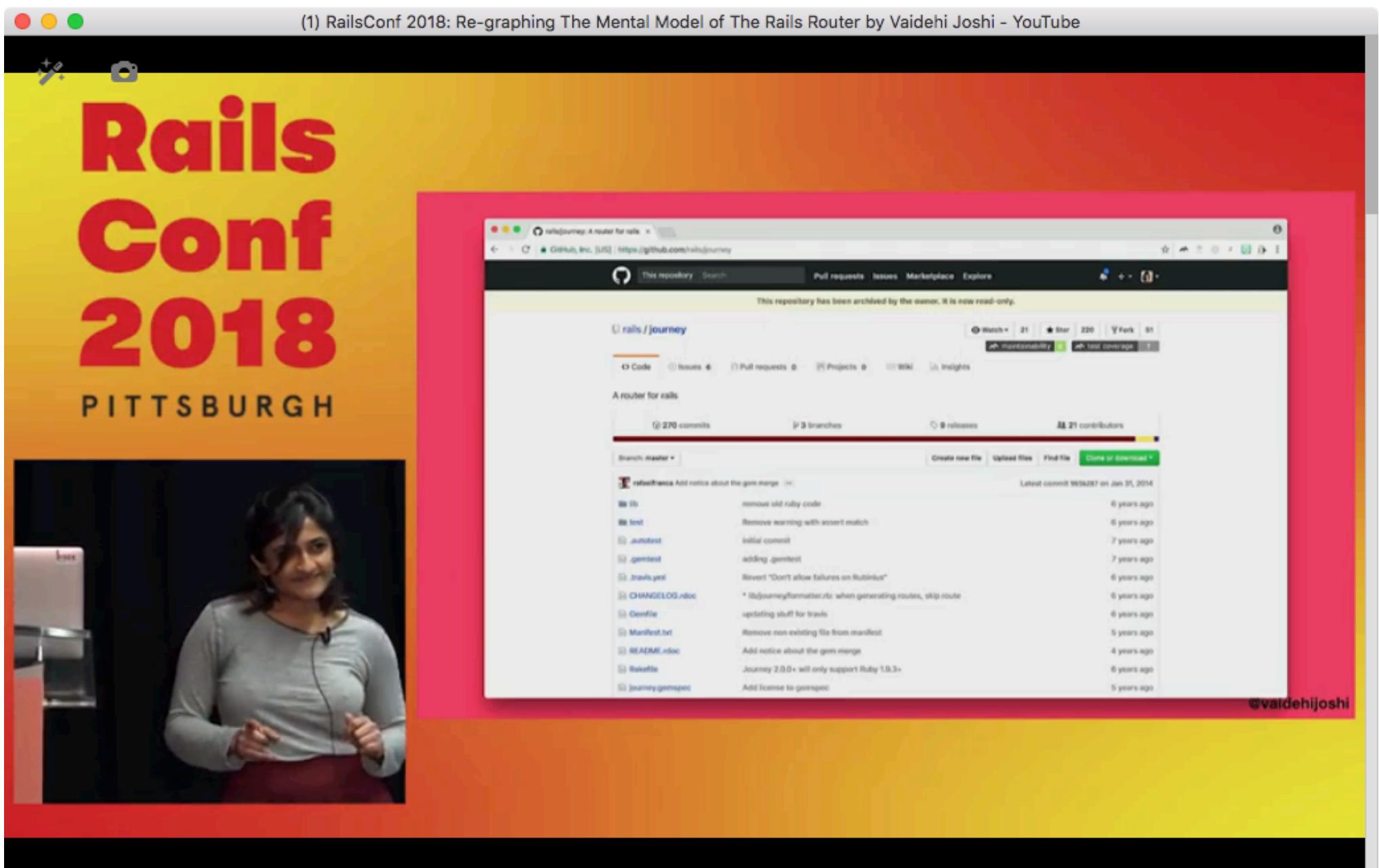
linear, O(n)

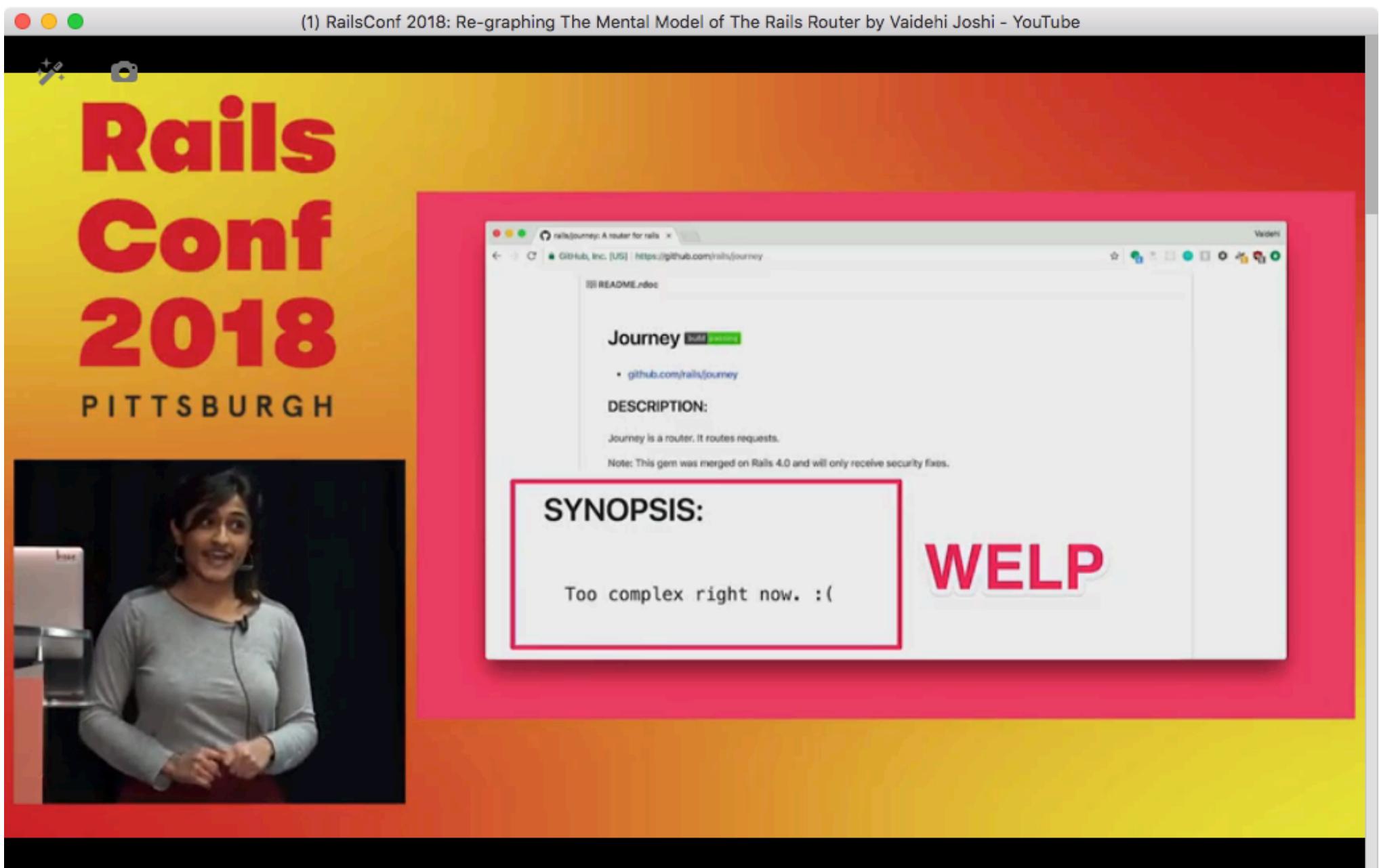
@vaidehijoshi



@vaidehijoshi







The image shows a screenshot of a video player window. On the left, there's a thumbnail of a woman with dark hair, wearing a light grey long-sleeved top and maroon pants, standing and gesturing with her hands. The main area of the video player has a yellow-to-red gradient background. On the left side of this area, the text "Rails Conf 2018 PITTSBURGH" is displayed in large red font, with "PITTSBURGH" in smaller black capital letters below it. To the right of this, a white rectangular box contains the text "Application.routes". A black oval highlights the word "routes". In the bottom right corner of the main slide area, the handle "@vaidehijoshi" is visible. The top of the video player has a standard OS X-style window bar with red, yellow, and green buttons.

Rails Conf 2018

PITTSBURGH



```
@routes ||=  
ActionDispatch::Routing::RouteSet.new
```

@vaidehijoshi



```
#action_dispatch/routing/route_set.rb  
@set ||= Journey::Routes.new
```

@vaidehijoshi

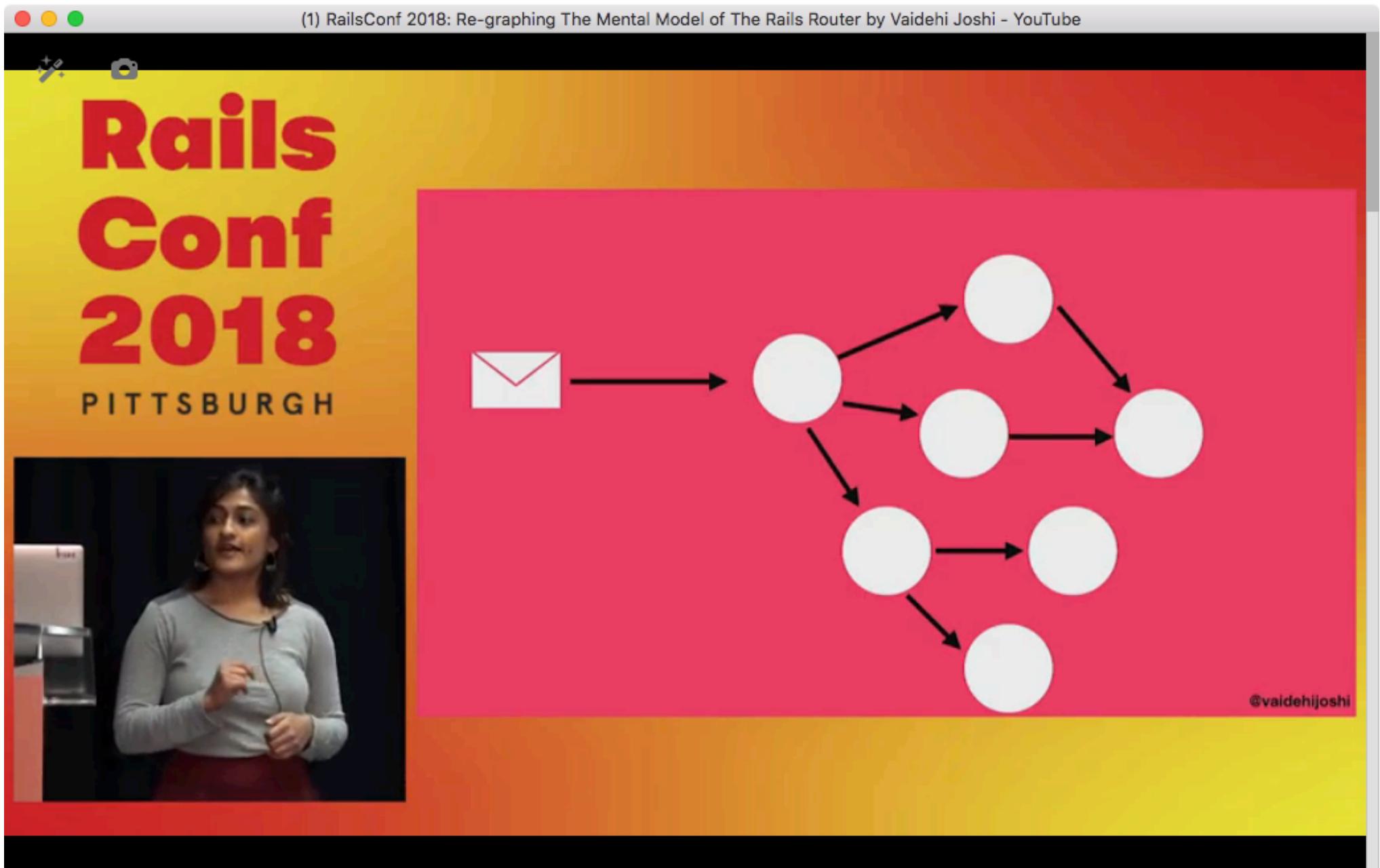
(1) RailsConf 2018: Re-graphing The Mental Model of The Rails Router by Vaidehi Joshi - YouTube

Rails
Conf
2018

PITTSBURGH

directed
edges

@vaidehijoshi





```
Rails.application.routes.draw do
  root 'welcome#home'

  resources :articles
  resources :recipes
  resources :comments
end
```

@vaidehijoshi



```
Rails.application.routes.draw do
  root 'welcome#home'
  resources :articles
  resources :recipes
  resources :comments
end
```

“/recipes/:id”

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



```
$ rails routes
```

Verb	URI Pattern	Controller#Action
GET	/articles(.:format)	articles#index
POST	/articles(.:format)	articles#create
GET	/articles/new(.:format)	articles#new
GET	/articles/:id/edit(.:format)	articles#edit
GET	/articles/:id(.:format)	articles#show
PATCH	/articles/:id(.:format)	articles#update
PUT	/articles/:id(.:format)	articles#update
DELETE	/articles/:id(.:format)	articles#destroy

@vaidehijoshi



```
Rails.application.routes.draw do
  root 'welcome#home'
  resources :articles
  resources :recipes
  resources :comments
end
```

“/recipes/:id”

@vaidehijoshi



```
Rails.application.routes.draw do
  root 'welcome#home'
  resources :articles
  resources :recipes
  resources :comments
end
```

“/recipes/:id”

@vaidehijoshi

Rails
Conf
2018

PITTSBURGH

“/recipes/:id”

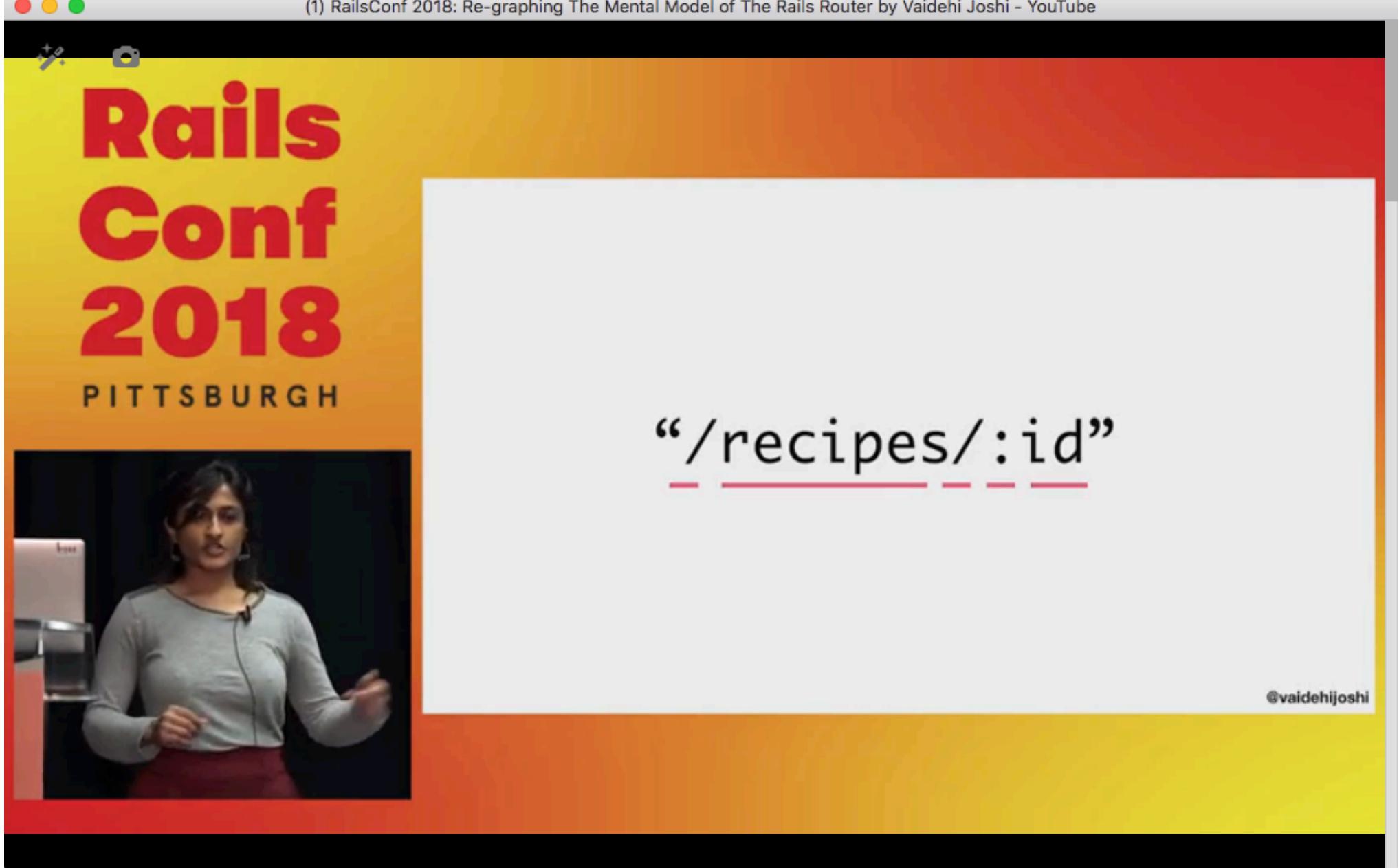
@vaidehijoshi

Rails
Conf
2018

PITTSBURGH

tokenization

@vaidehijoshi





```
$ rails console
> scanner = ActionDispatch::Journey::Scanner.new
> scanner.scan_setup("/recipes/:id")
```



@vaidehijoshi

Rails Conf 2018

PITTSBURGH



```
> scanner.next_token
=> [:SLASH, “/“]
> scanner.next_token
=> [:LITERAL, "recipes"]
> scanner.next_token
=> [:SLASH, “/“]
> scanner.next_token
=> [:SYMBOL, ":id"]
```



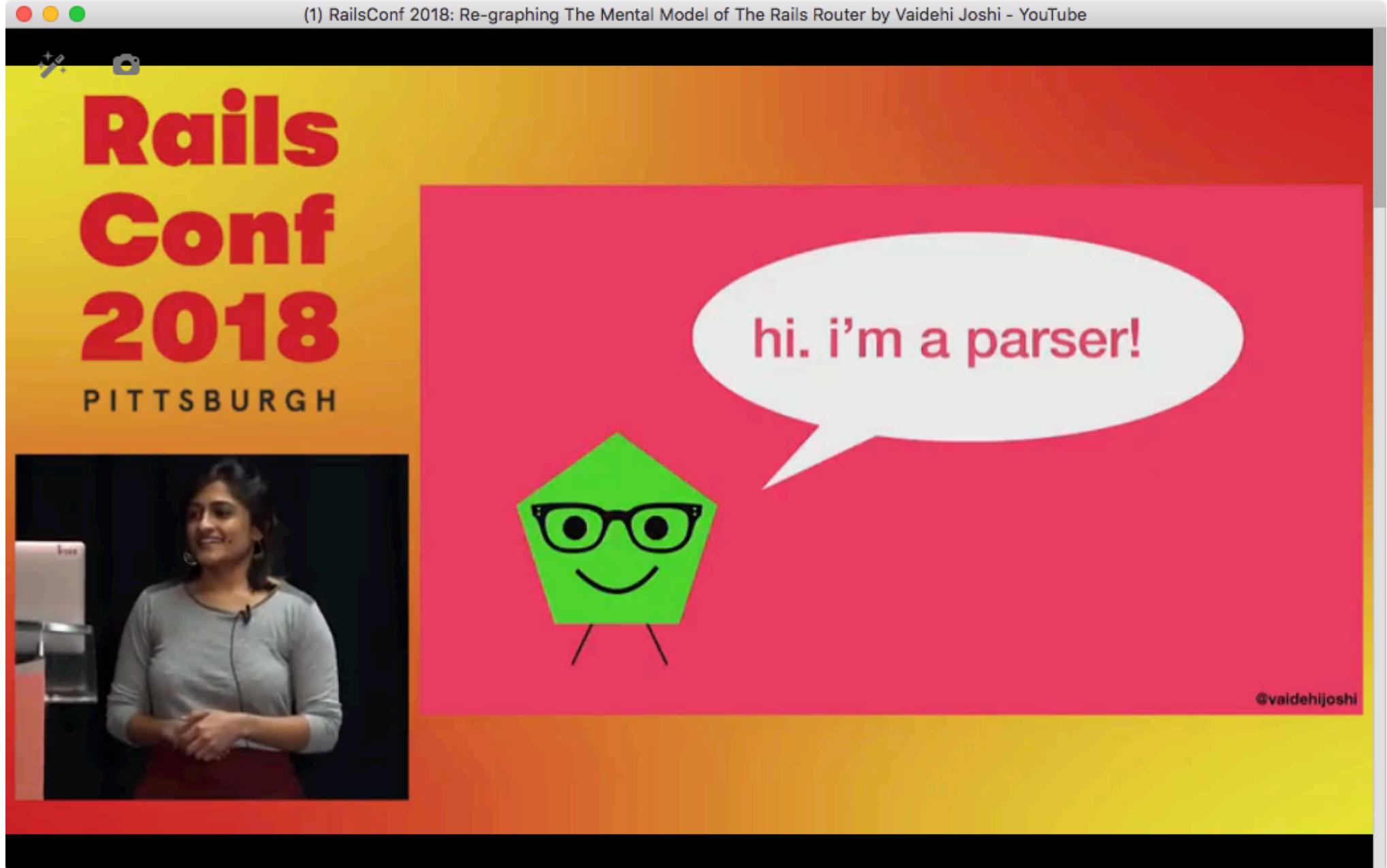
@vaidehijoshi

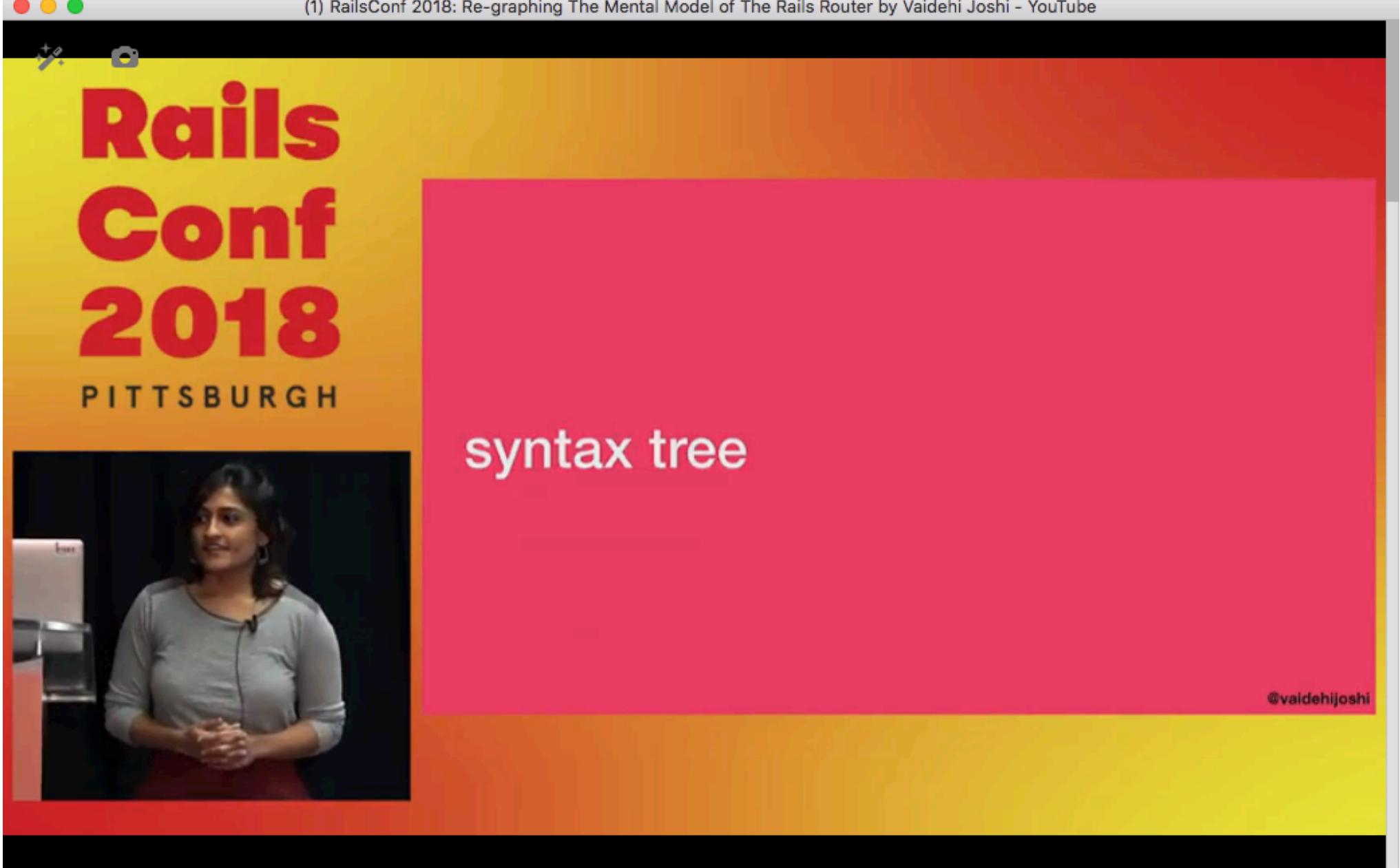
**Rails
Conf
2018**
PITTSBURGH

Tokens I know:

:SLASH, :LITERAL, :SYMBOL,
:LPAREN, :RPAREN, :DOT, :OR,
:STAR.

@vaidehijoshi





(1) RailsConf 2018: Re-graphing The Mental Model of The Rails Router by Vaidehi Joshi - YouTube

The diagram illustrates the components of a Rails route path: "/recipes/:id(.:format)". The path is broken down into tokens: "/", "recipes", "/", ":id", ".", "0", and ":format". Each token is represented by a red square containing its value. Arrows show the flow from left to right between these tokens. A green pentagonal character with glasses is positioned below the first two tokens.

Rails Conf 2018
PITTSBURGH

/recipes/:id(.:format)

Rails Conf 2018 PITTSBURGH

"/recipes/:id(.:format)"

/recipes/:id(.:format)

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



```
$ rails console
> parser = ActionDispatch::Journey::Parser.new
> syntax_tree = parser.parse("/recipes/:id(.:format)")
```



@vaidehijoshi

Rails Conf 2018

PITTSBURGH



intermediate step

@vaidehijoshi

Rails
Conf
2018

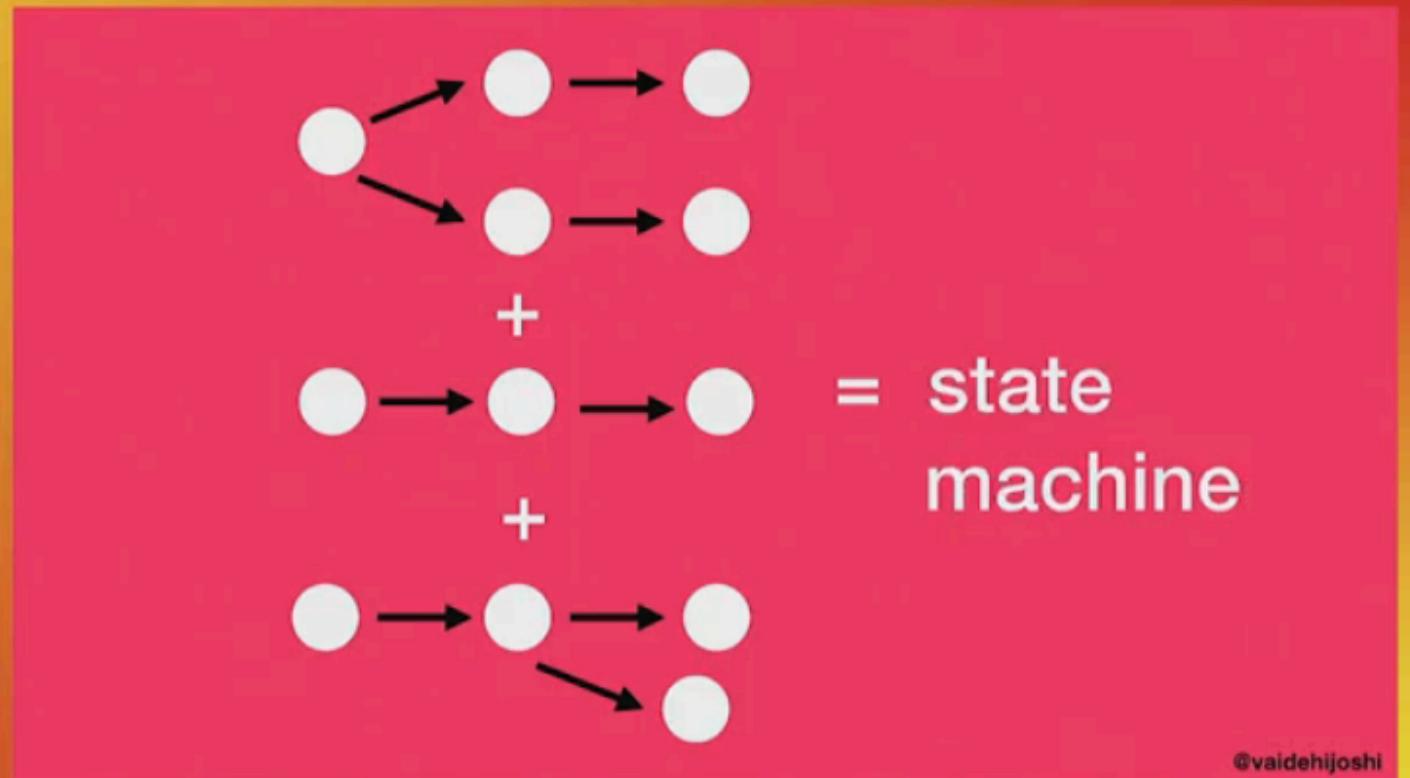
PITTSBURGH

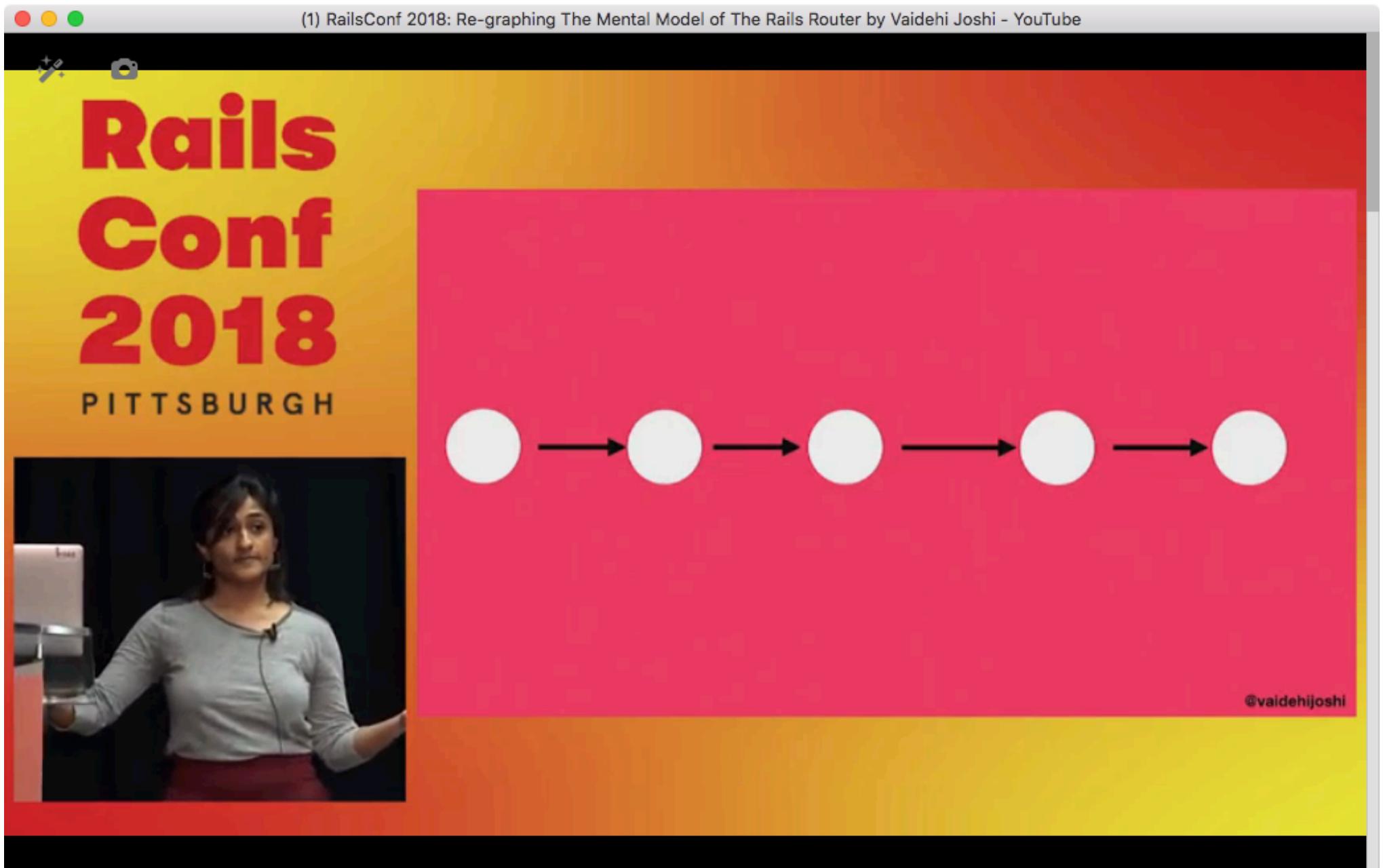
+

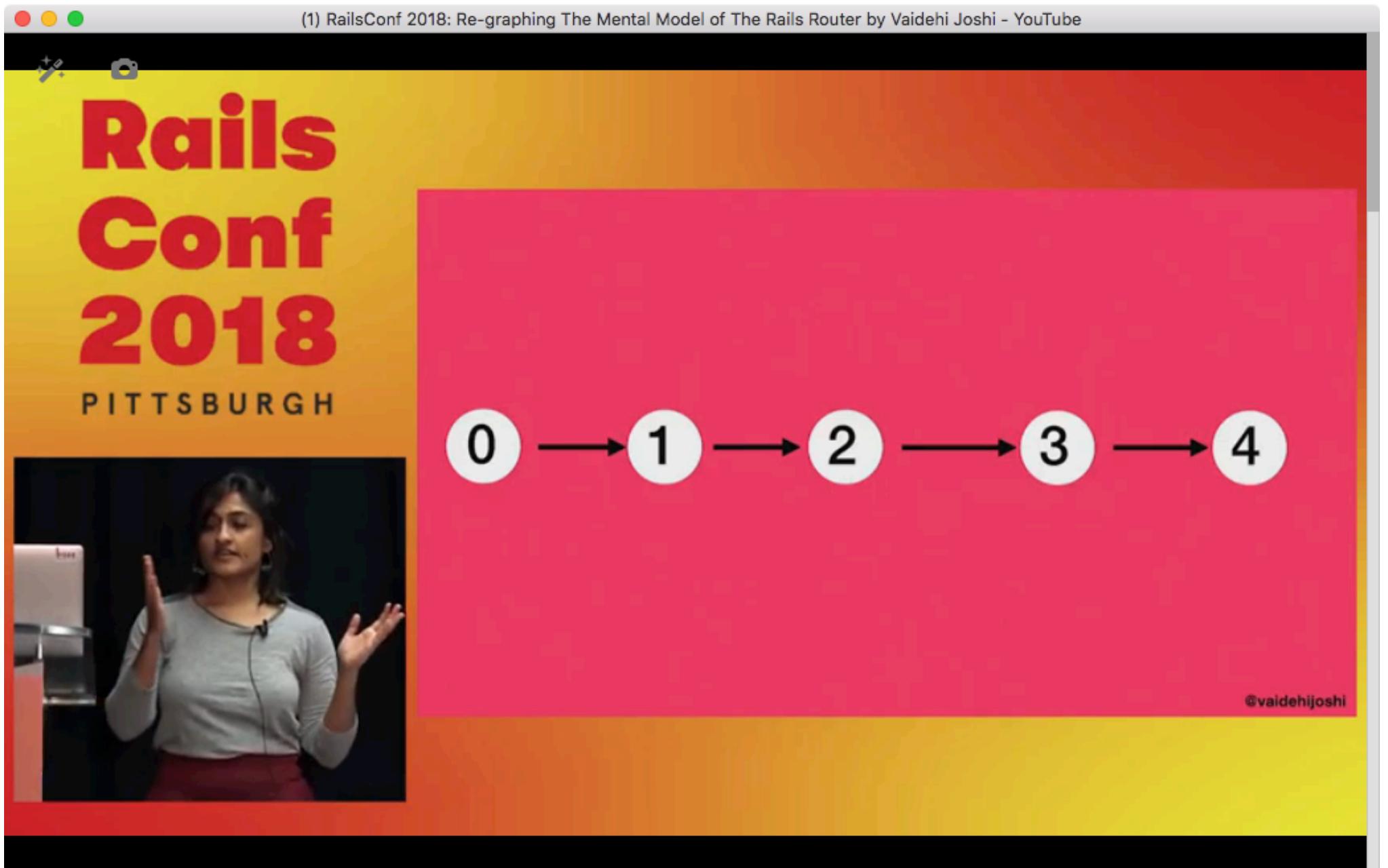
+

= generalized
transition
graph
(GTG)

@vaidehijoshi







(1) RailsConf 2018: Re-graphing The Mental Model of The Rails Router by Vaidehi Joshi - YouTube

Rails Conf 2018
PITTSBURGH

Vaidehi Joshi

Diagram illustrating a state transition graph:

```
graph LR; 0((0)) --> 1((1)); 1 --> 2((2)); 2 --> 3((3)); 2 --> 4((4)); 4 --> 5((5)); 3 --> "..."; 4 --> "...";
```

Annotations on the diagram:

- State 2 has an outgoing arrow labeled *articles*.
- States 3 and 4 have outgoing arrows labeled *...*.
- A regular expression pattern is shown below the diagram: $(?-mix:[^./?]+)$
- The source code credit is @vaidehijoshi.

(1) RailsConf 2018: Re-graphing The Mental Model of The Rails Router by Vaidehi Joshi - YouTube

The image shows a video player interface. On the left, there's a thumbnail of Vaidehi Joshi speaking at a podium. The main content area has a yellow-to-red gradient background. At the top, it says "Rails Conf 2018 PITTSBURGH". Below that is a diagram illustrating a routing path. The path starts at node 0 (green), goes to node 1 (green), then to node 2 (white). From node 2, arrows point to node 3 (white) via a slash character, to node 4 (white) via the word "articles", and to node 5 (white) via a dot character. Node 5 points to node 6 (white) via an ellipsis (...). Below the diagram, the URL "/articles.json" is shown in green. At the bottom right, there's some handwritten-style text: "(?-mix:[^./?]+)" and the handle "@vaidehijoshi".

(1) RailsConf 2018: Re-graphing The Mental Model of The Rails Router by Vaidehi Joshi - YouTube

Rails Conf 2018
PITTSBURGH

/articles.json

```
graph LR; 0((0)) --> 1((1)); 1 --> 2((2)); 2 -- articles --> 4((4)); 2 -- / --> 3((3)); 2 -- . --> 4; 4 --> 5((5)); 3 --> ...([...]); 5 -- "(?-mix:[^./?]+)" --> ...;
```

@vaidehijoshi

(1) RailsConf 2018: Re-graphing The Mental Model of The Rails Router by Vaidehi Joshi - YouTube

Rails Conf 2018
PITTSBURGH

/articles.json

```
graph LR; 0((0)) --> 1((1)); 1 --> 2((2)); 2 -- articles --> 4((4)); 2 -- / --> 3((3)); 2 -- . --> 4; 4 --> 5((5)); 5 -- checked --> end(( ));
```

(?*-mix:[^./?]+)*

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



```
get 'recipes/:id', to: 'recipes#show'  
get 'articles(.:format)', to: 'articles#index'  
post 'articles/new', to: 'articles#new'  
get 'articles/:id', to: 'articles#new'  
  
# even more routes here...
```

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



```
get 'recipes/:id', to: 'recipes#show'  
get 'articles(.:format)', to: 'articles#index'  
post 'articles/new', to: 'articles#new'  
get 'articles/:id', to: 'articles#new'  
  
# even more routes here...
```

@vaidehijoshi

(1) RailsConf 2018: Re-graphing The Mental Model of The Rails Router by Vaidehi Joshi - YouTube

Rails Conf 2018
PITTSBURGH

/articles.json

0 → 1 → 2

2 → 3 → ...

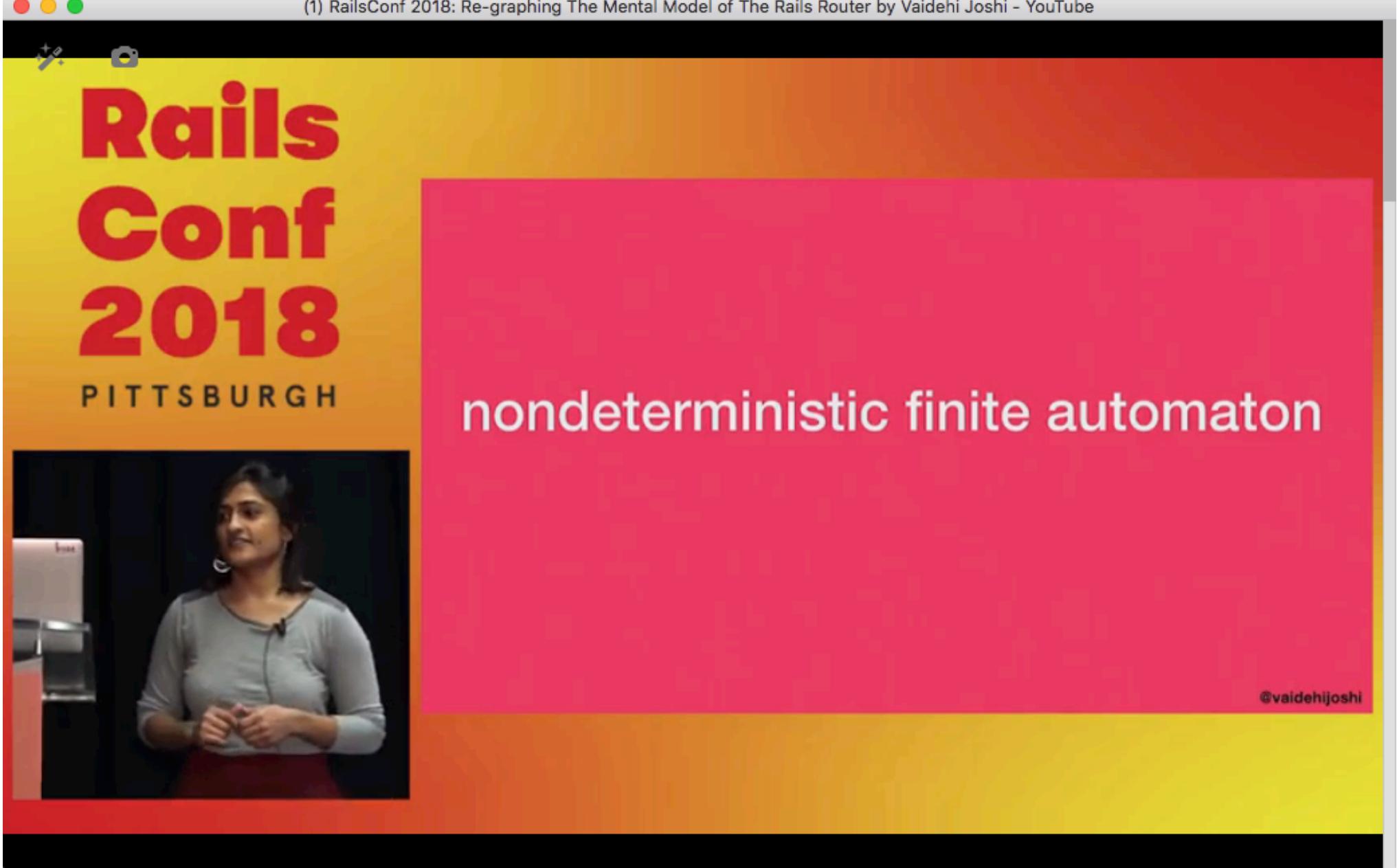
2 → 4 → 5 → ...

articles

/?-mix:[^./?]+)

@vaidehijoshi

The diagram illustrates a router graph for the URL '/articles.json'. It starts with a root node (0) which has an outgoing edge to node (1). Node (1) has an outgoing edge to node (2). From node (2), there are two outgoing edges: one to node (3) and another to node (4). Node (3) has an outgoing edge to an ellipsis (...). Node (4) has an outgoing edge to node (5), which in turn has an outgoing edge to another ellipsis (...). A small robot icon is positioned near node (0). The path from the root to '/articles.json' is labeled 'articles'. Below the graph, the regular expression pattern '?-mix:[^./?]+)' is shown, which matches the trailing part of the URL. The source code '@vaidehijoshi' is also visible.



Rails Conf 2018

PITTSBURGH



nondeterministic finite automaton
(NFA)

just a fancy state machine!

@vaidehijoshi

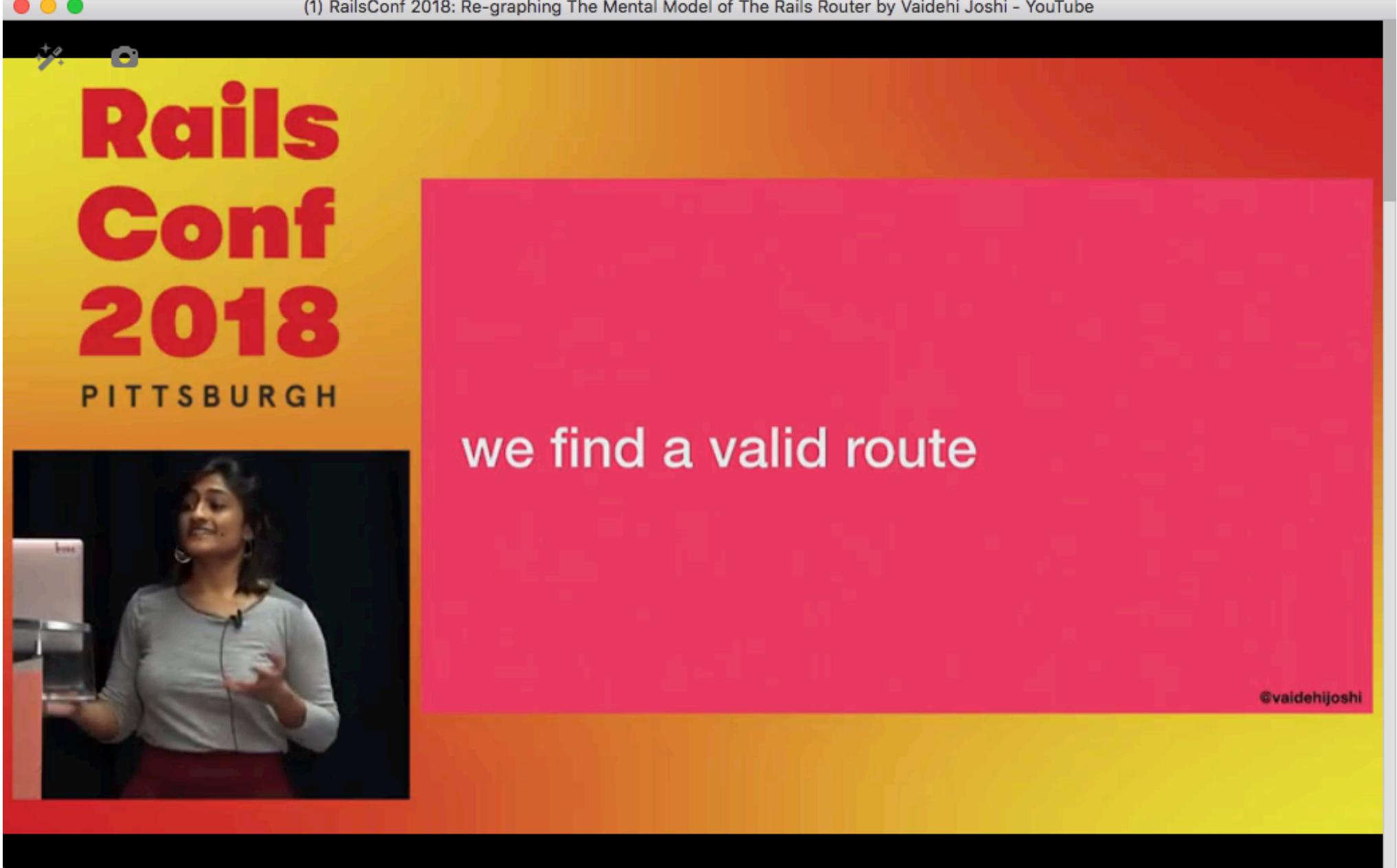
Rails Conf 2018

PITTSBURGH



two possible outcomes

@vaidehijoshi

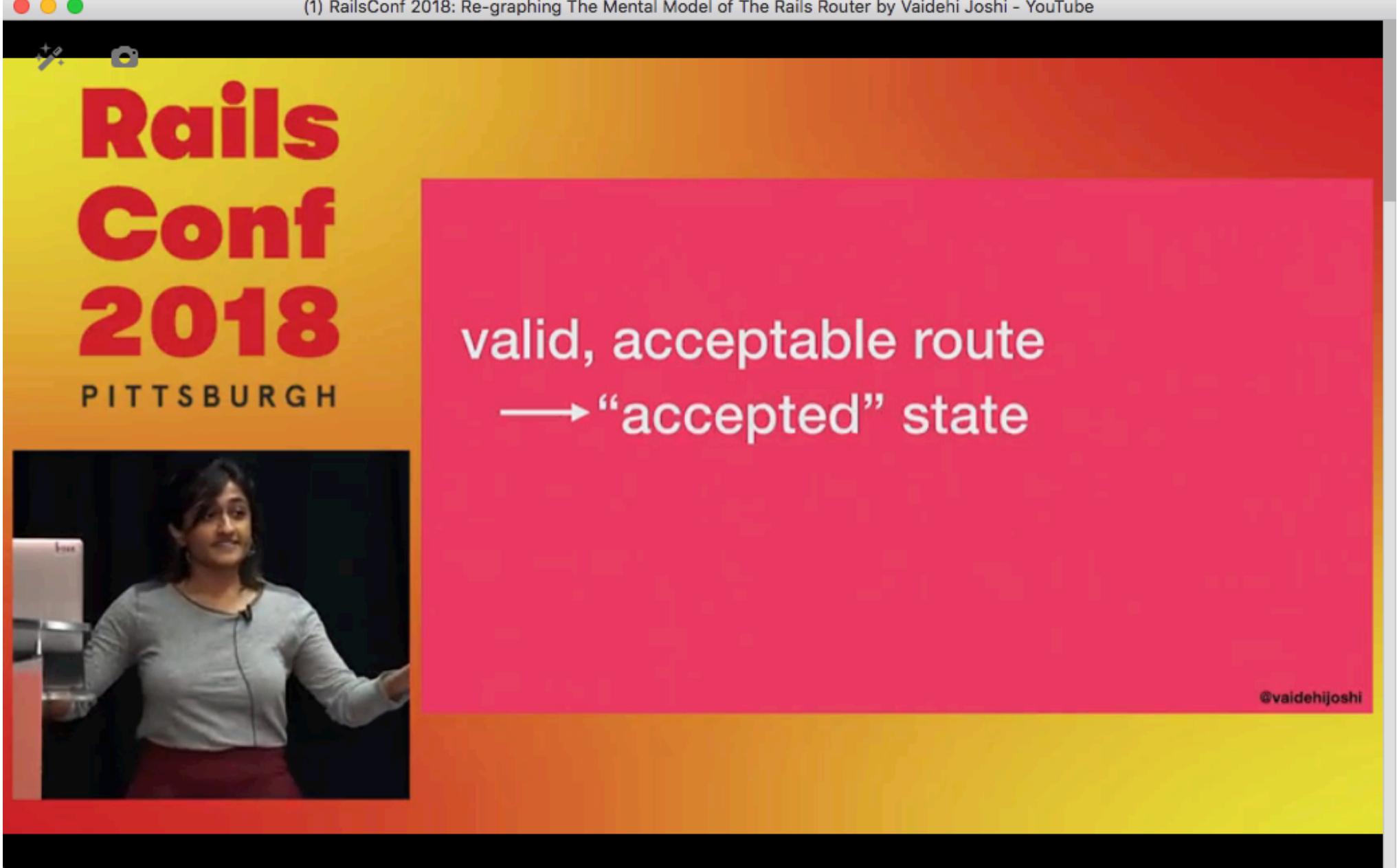


Rails Conf 2018
PITTSBURGH

we find a valid route

@vaidehijoshi

A screenshot of a video player window. On the left, there's a thumbnail image of a woman with dark hair, wearing a grey top and red pants, standing on a stage and holding a small white object. The main video frame is mostly red and contains the text "we find a valid route". The top bar of the video player shows the title "(1) RailsConf 2018: Re-graphing The Mental Model of The Rails Router by Vaidehi Joshi - YouTube". The bottom right corner of the video frame has the handle "@vaidehijoshi". The overall background of the slide is yellow and orange.



Rails
Conf
2018

PITTSBURGH

valid, acceptable route
→ “accepted” state

@vaidehijoshi

A screenshot of a video player window. On the left, there's a thumbnail image of a woman with dark hair, wearing a grey long-sleeved top and maroon pants, standing and gesturing with her hands. To the right of the thumbnail is a large red rectangular area containing white text. The text reads "valid, acceptable route" on the first line and "→ “accepted” state" on the second line, with an arrow pointing from the first phrase to the second. In the bottom right corner of the red area, there is a small white text "@vaidehijoshi". The overall background of the video player has a yellow-to-orange gradient.

The image shows a video player interface. On the left, a video frame displays Vaidehi Joshi, a woman with dark hair, smiling and gesturing while speaking. She is wearing a light grey long-sleeved shirt. The background of the video frame is dark. To the right of the video frame is a large yellow and orange graphic banner for "Rails Conf 2018 PITTSBURGH". The text "Rails Conf 2018" is in large red letters, and "PITTSBURGH" is in smaller black letters below it. At the bottom of the banner is a smaller video frame showing a close-up of a person's hands working on a laptop keyboard.

The main content area of the slide shows a screenshot of a web application titled "Routes FSM with NFA simulation". The application has a search bar at the top with placeholder text "Type a route in to the box and click 'simulate'." It includes buttons for "simulate" and "reset". Below the search bar, there is a list of "Some fun routes to try: /articles /articles/1/edit.json /articles/new, /articles/edit".

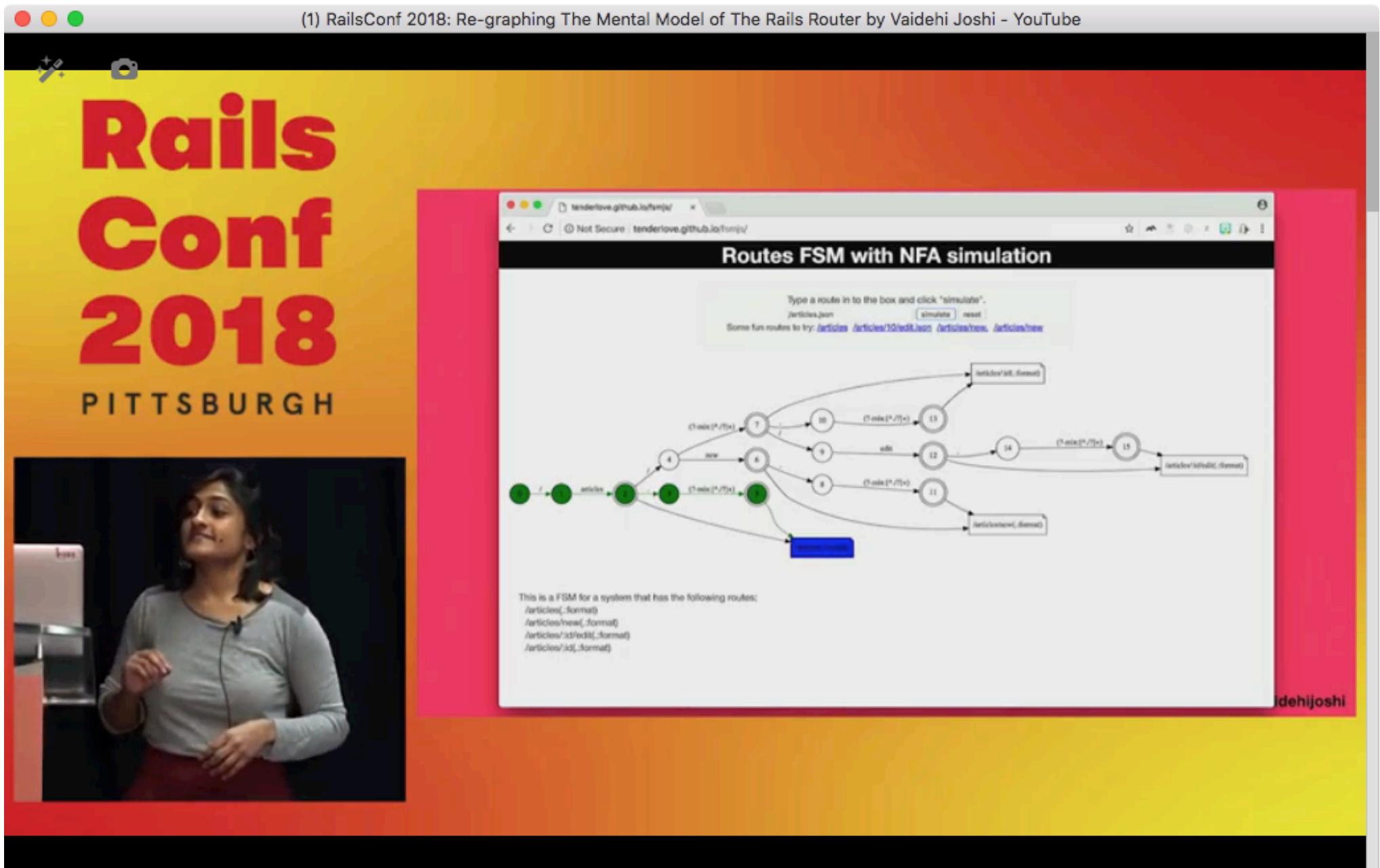
The central part of the slide is a Finite State Machine (FSM) diagram. The states are numbered 0 through 15. Transitions are labeled with route patterns:

- From state 0 to state 1: articles
- From state 1 to state 2: articles
- From state 2 to state 3: articles
- From state 3 to state 4: articles(/:id)
- From state 4 to state 5: articles(/:id)
- From state 5 to state 6: new
- From state 6 to state 7: articles(/:id/:format)
- From state 7 to state 8: articles(/:id/:format)
- From state 8 to state 9: edit
- From state 9 to state 10: articles(/:id/:format)
- From state 10 to state 11: articles(/:id/:format)
- From state 11 to state 12: edit
- From state 12 to state 13: articles(/:id/:format)
- From state 13 to state 14: articles(/:id/:format)
- From state 14 to state 15: articles(/:id/:format)
- From state 15 to state 16: articles(/:id/:format)
- From state 16 to state 17: articles(/:id/:format)

States 16 and 17 are labeled "articles(/:id,:format)".

At the bottom of the slide, there is a note: "This is a FSM for a system that has the following routes:
/articles(:format)
/articles/new(:format)
/articles/:id/edit(:format)
/articles/:id(:format)".

In the bottom right corner of the slide, there is a signature or handle: "dehijoshi".



Rails Conf 2018

PITTSBURGH



invalid, unacceptable route

@vaidehijoshi

Rails
Conf
2018
PITTSBURGH

invalid, unacceptable route
→ “rejected” state

@vaidehijoshi

The image shows a video player interface. On the left, a woman with dark hair, wearing a grey long-sleeved shirt, is speaking. She is standing in front of a black background with a small portion of a silver laptop visible on the left. On the right, there is a screenshot of a web browser window titled "Routes FSM with NFA simulation". The browser address bar shows "tenderlove.github.io/rails/" and the page title is "Routes FSM with NFA simulation". The page contains instructions: "Type a route in to the box and click "simulate". /not-a-real-route simulate reset Some fun routes to try: /articles /articles/10/edition /articles/one /articles/new". Below these are two boxes: one for "simulate" and one for "reset". A Non-deterministic Finite State Machine (NFA) diagram is displayed, showing states represented by circles and transitions represented by arrows. States include 1 (initial), 2 (red), 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, and several final states: "articles(.format)", "articles/new(.format)", "articles/:id(.format)", and "articles/:id/edit(.format)". Transitions are labeled with route patterns like "/articles(.format)" and "/articles/:id(.format)". At the bottom of the diagram, it says "This is a FSM for a system that has the following routes:" followed by the four final state examples. In the bottom right corner of the browser window, there is a watermark that reads "jidehijoshi".

(1) RailsConf 2018: Re-graphing The Mental Model of The Rails Router by Vaidehi Joshi - YouTube

Rails Conf 2018
PITTSBURGH

Routing Error

No route matches [GET] "/not-a-real-route"

Rails.root: /Users/vaidehijoshi/Code/example-app

[Application Trace](#) | [Framework Trace](#) | [Full Trace](#)

Routes

Routes match in priority from top to bottom

Helper	HTTP Verb	Path
Path / Url		Path Match
root_path	GET	/
articles_path	GET	/articles{.:format}
	POST	/articles{.:format}
new_article_path	GET	/articles/new{.:format}

vaidehijoshi



Rails Conf 2018

PITTSBURGH

```
$ rails console  
> nfa = Rails.application.routes.router.visualizer
```

@vaidehijoshi

A video frame showing Vaidehi Joshi, a woman with dark hair, smiling and gesturing with her hands while speaking. She is wearing a light grey zip-up top and maroon pants. The background is dark, suggesting a stage or conference setting.

Rails
Conf
2018

PITTSBURGH

well
this has been quite a
~journey~

@vaidehijoshi

A large yellow and orange gradient bar runs across the bottom of the slide.

Rails
Conf
2018

PITTSBURGH

tokenization

scanning

trees & graphs

automatons

🤯

@vaidehijoshi

Rails Conf 2018

PITTSBURGH



you don't need to think about abstractions every day.

@vaidehijoshi

Rails Conf 2018

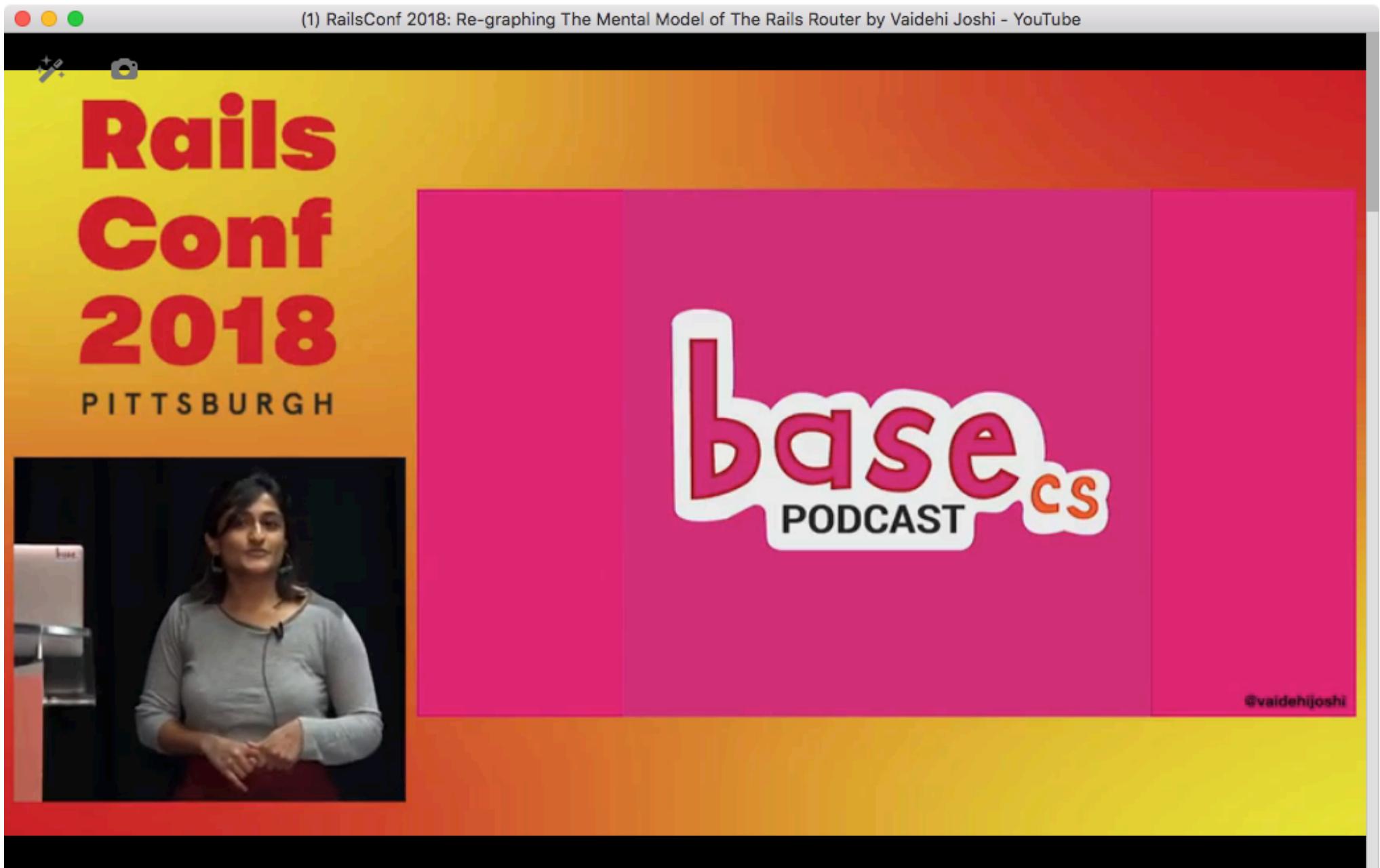
PITTSBURGH



but if you find yourself wanting or
need to learn them, don't be afraid!

@vaidehijoshi





Rails Conf 2018

PITTSBURGH



basecs.org ✨

@vaidehijoshi

