Single-sign-on with OAuth 2

Emily

Meet Emely Enduser

198.51.100.

She has an iPhone app

Emily

203.0.113.100

and a browser

198.51.100.

192.0.2.

Beta

She uses the websites
Alpha and Beta

(OAuth Clients)

Emily

203.0.113.100

192.0.2.

Alpha

198.51.100.

192.0.2.

Beta

beta.dev

Emily

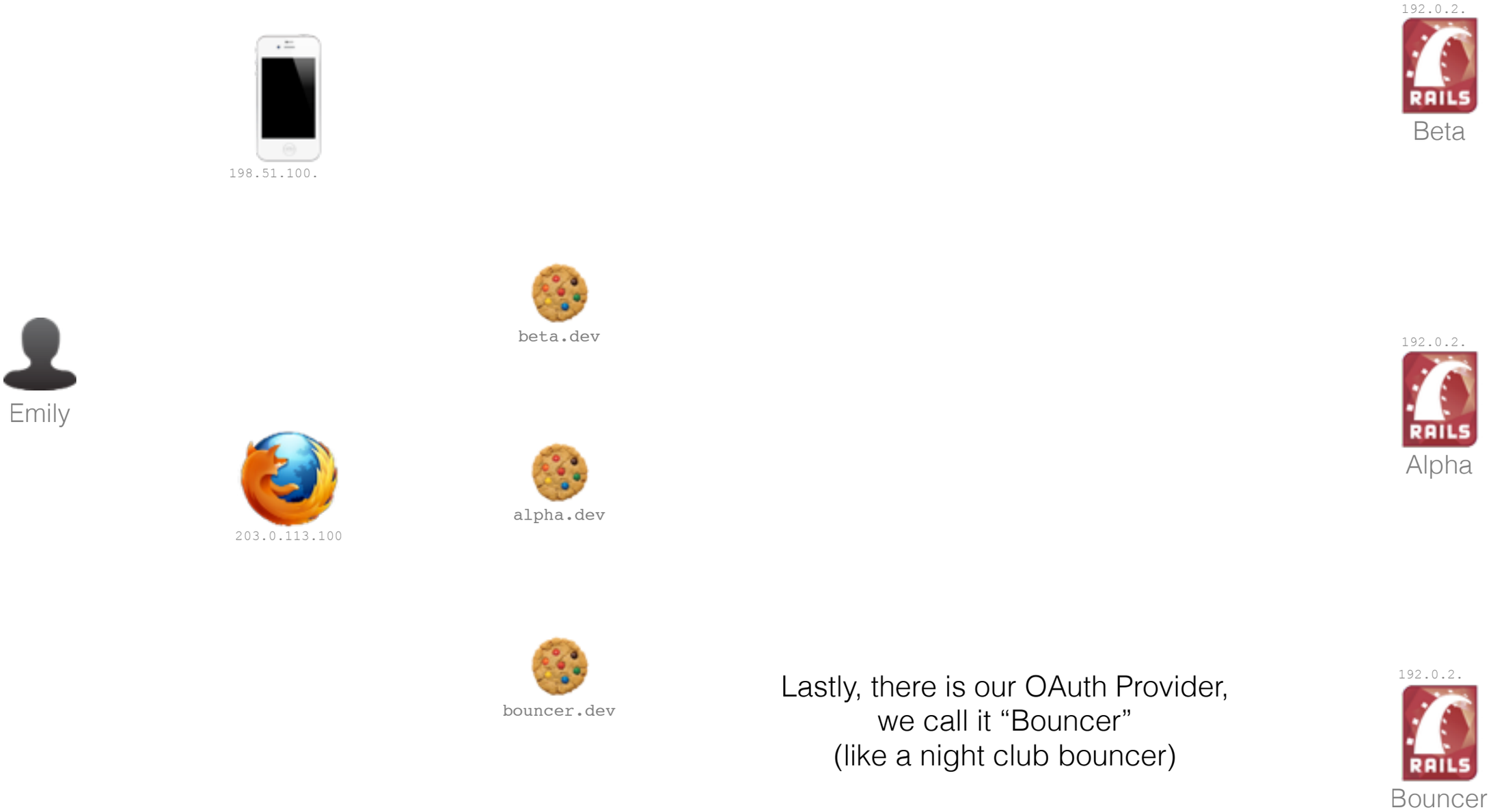
The browser has a lot of cookies, one for each domain

192.0.2.

Alpha

203.0.113.100

alpha.dev

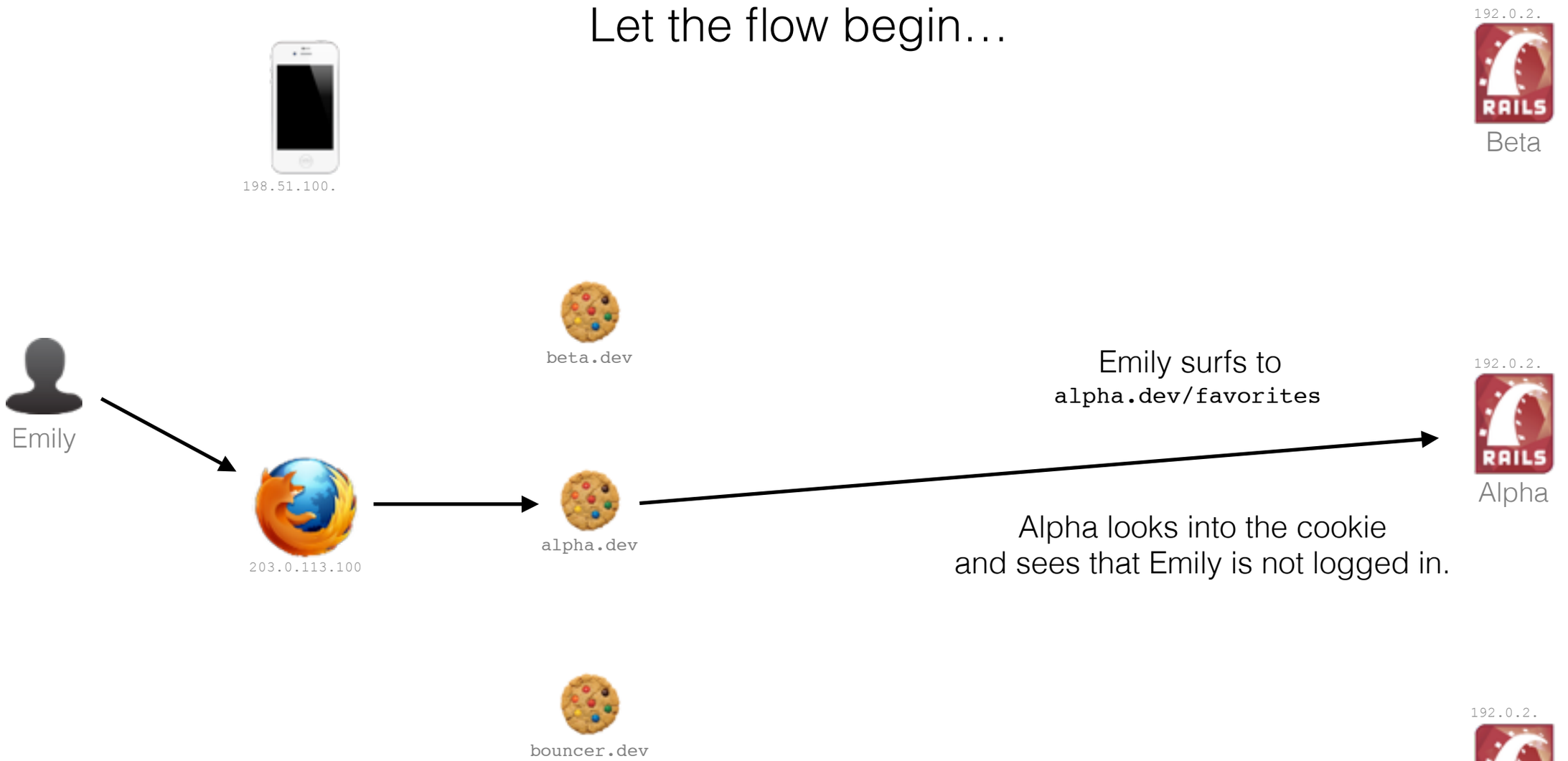


Lastly, there is our OAuth Provider,
we call it “Bouncer”
(like a night club bouncer)

Part 1

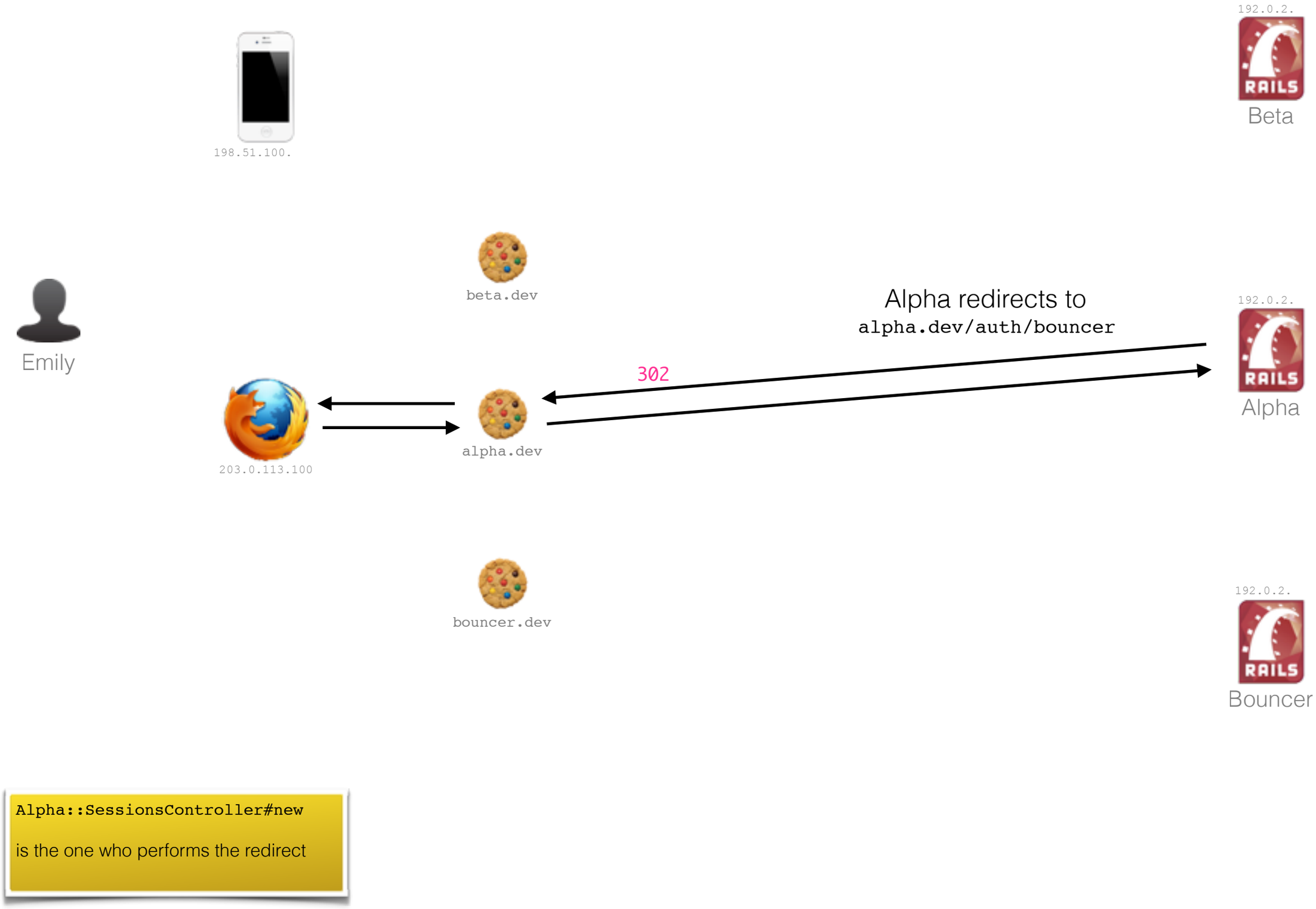
Logging in for the first time

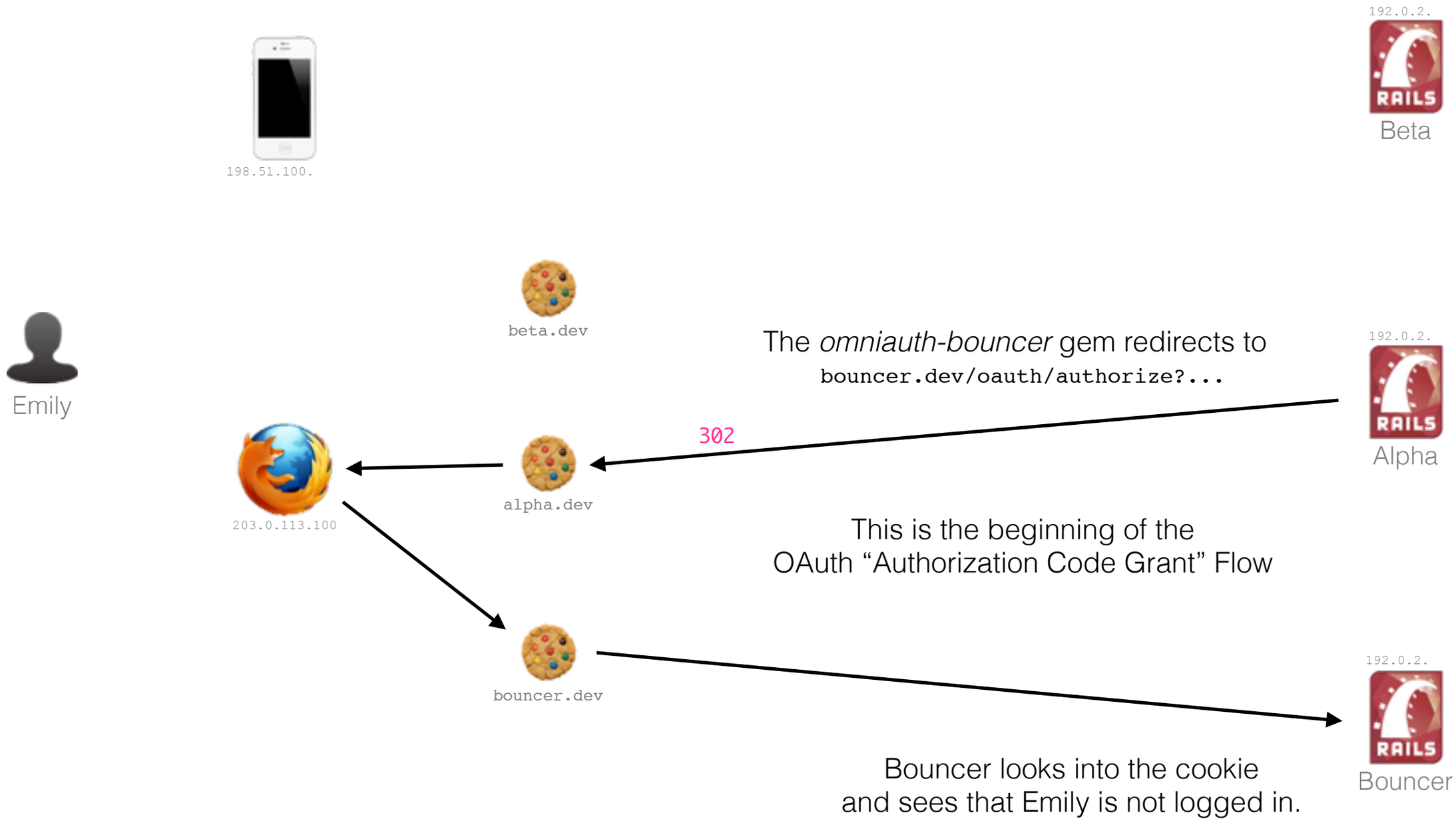
Let the flow begin…



Let's assume that `/favorites` is an endpoint that requires authentication and authorization.

`Alpha::ApplicationController#require_authorization` sees that there is no `session[:passport_id]`


198.51.100.
192.0.2.
RAILS
Beta
beta.dev
Emily
Alpha redirects to
alpha.dev/auth/bouncer
192.0.2.
RAILS
Alpha
302
203.0.113.100
alpha.dev
bouncer.dev
192.0.2.
RAILS
Bouncer
Alpha::SessionsController#new
is the one who performs the redirect

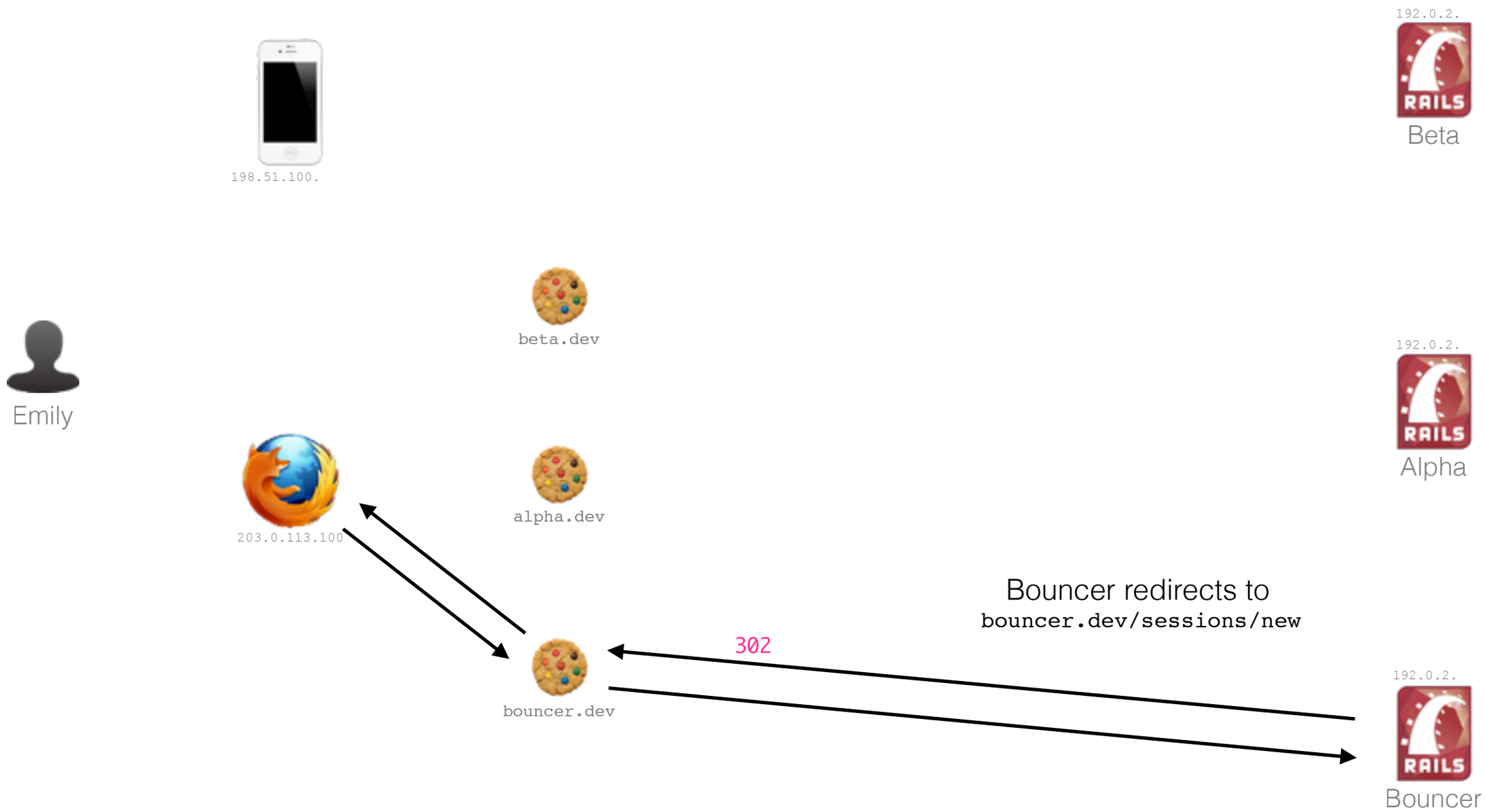


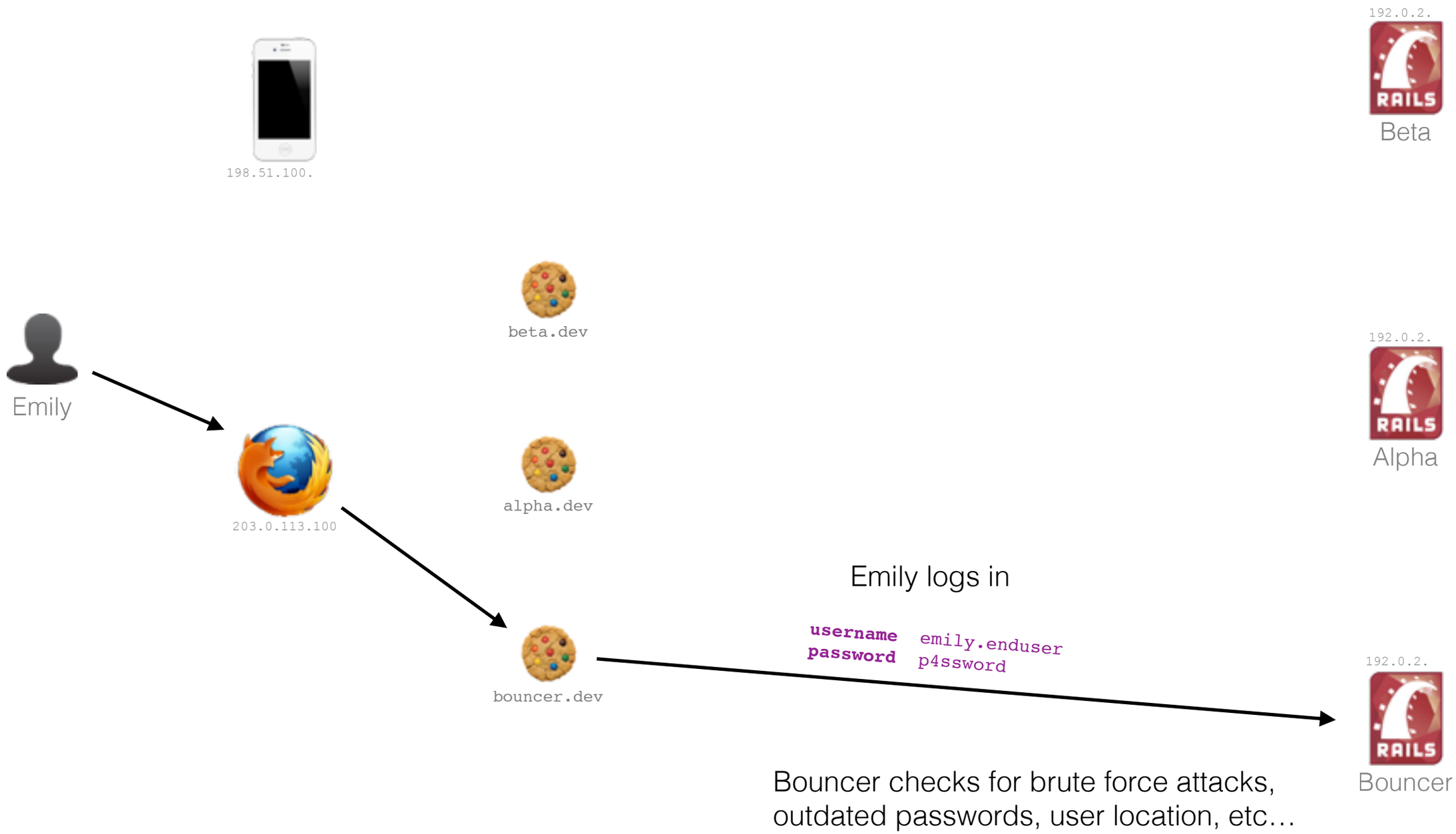
That omniauth-bouncer gem is a middleware defined like this:

```
require 'omniauth-oauth2'

module OmniAuth
  module Strategies
   class Bouncer < OmniAuth::Strategies::OAuth2
     # ...
   end
  end
end
```

`Bouncer::ApplicationController#require_authorization`
sees that there is no
`session[:passport_id]`

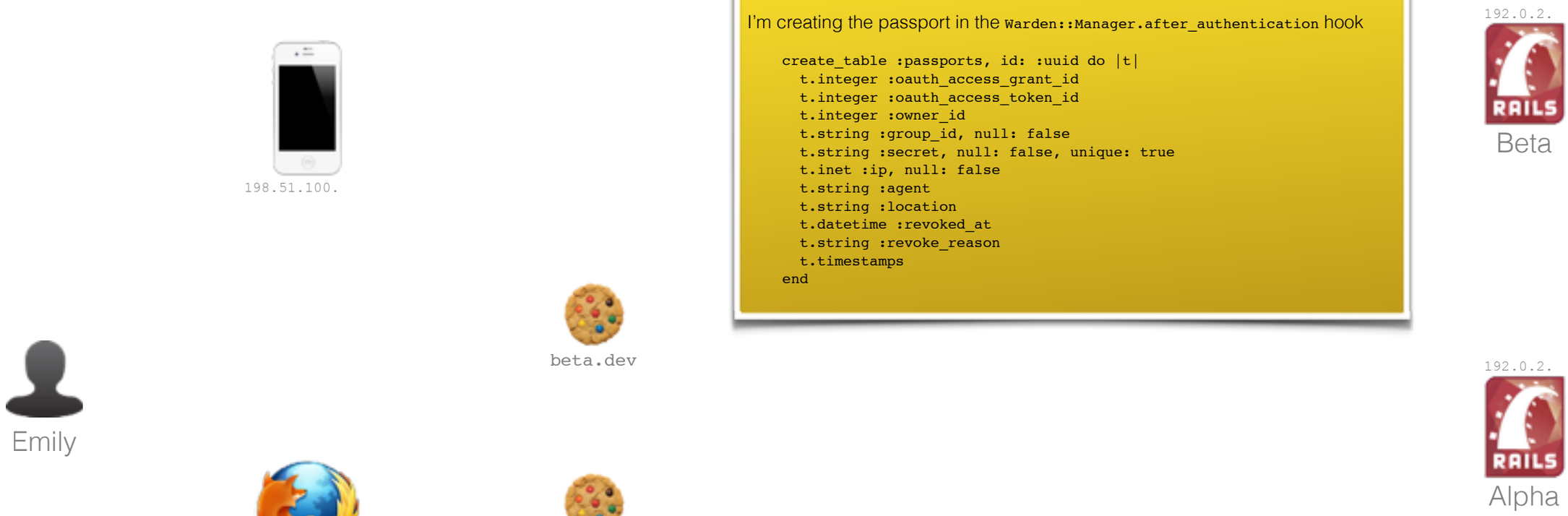

198.51.100.
192.0.2.
Beta
beta.dev
Emily
192.0.2.
Alpha
alpha.dev
203.0.113.100
Bouncer redirects to
bouncer.dev/sessions/new
302
bouncer.dev
192.0.2.
Bouncer


192.0.2.
Beta
198.51.100.
beta.dev
Emily
192.0.2.
Alpha
203.0.113.100
alpha.dev
Emily logs in
username emily.enduser
password p4ssword
192.0.2.
bouncer.dev
Bouncer
Bouncer checks for brute force attacks,
outdated passwords, user location, etc…

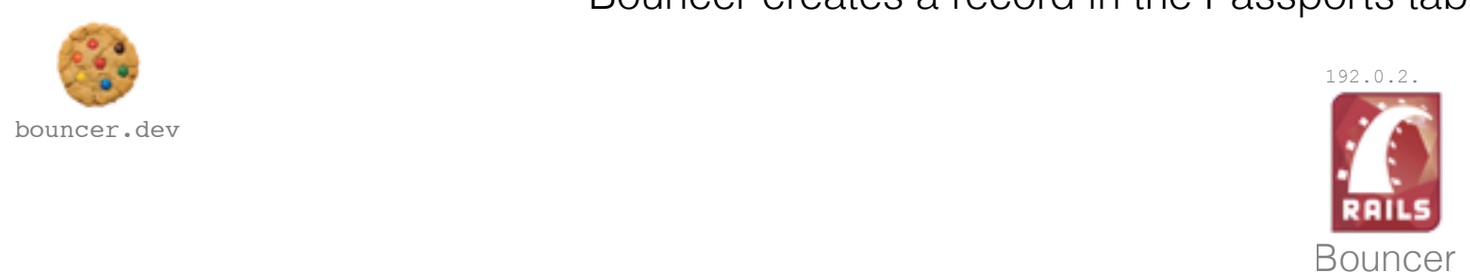
The passports table represents sessions (to avoid ambiguity we give it the name “passports”). One row in the table will correspond to one cookie.

I’m creating the passport in the `Warden::Manager.after_authentication` hook

```
create_table :passports, id: :uuid do |t|
  t.integer :oauth_access_grant_id
  t.integer :oauth_access_token_id
  t.integer :owner_id
  t.string :group_id, null: false
  t.string :secret, null: false, unique: true
  t.inet :ip, null: false
  t.string :agent
  t.string :location
  t.datetime :revoked_at
  t.string :revoke_reason
  t.timestamps
end
```



Bouncer creates a record in the Passports table



Bouncer’s DB

passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	555	-

Random uuid

Emilys user ID

Random uuid

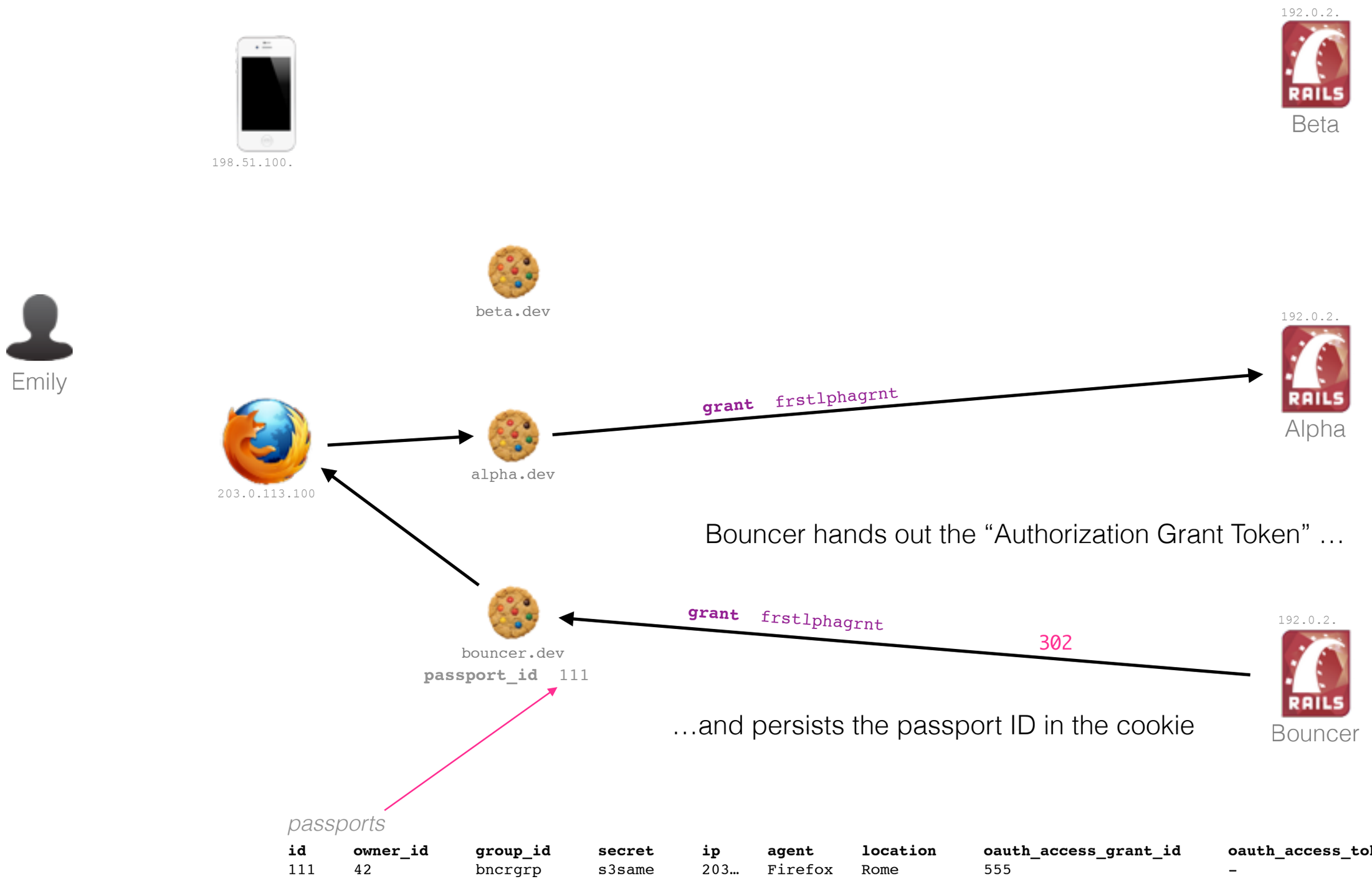
Random string

Browser info

(The table below is internally handled by the doorkeeper gem which just created a grant.)

oauth_access_grants

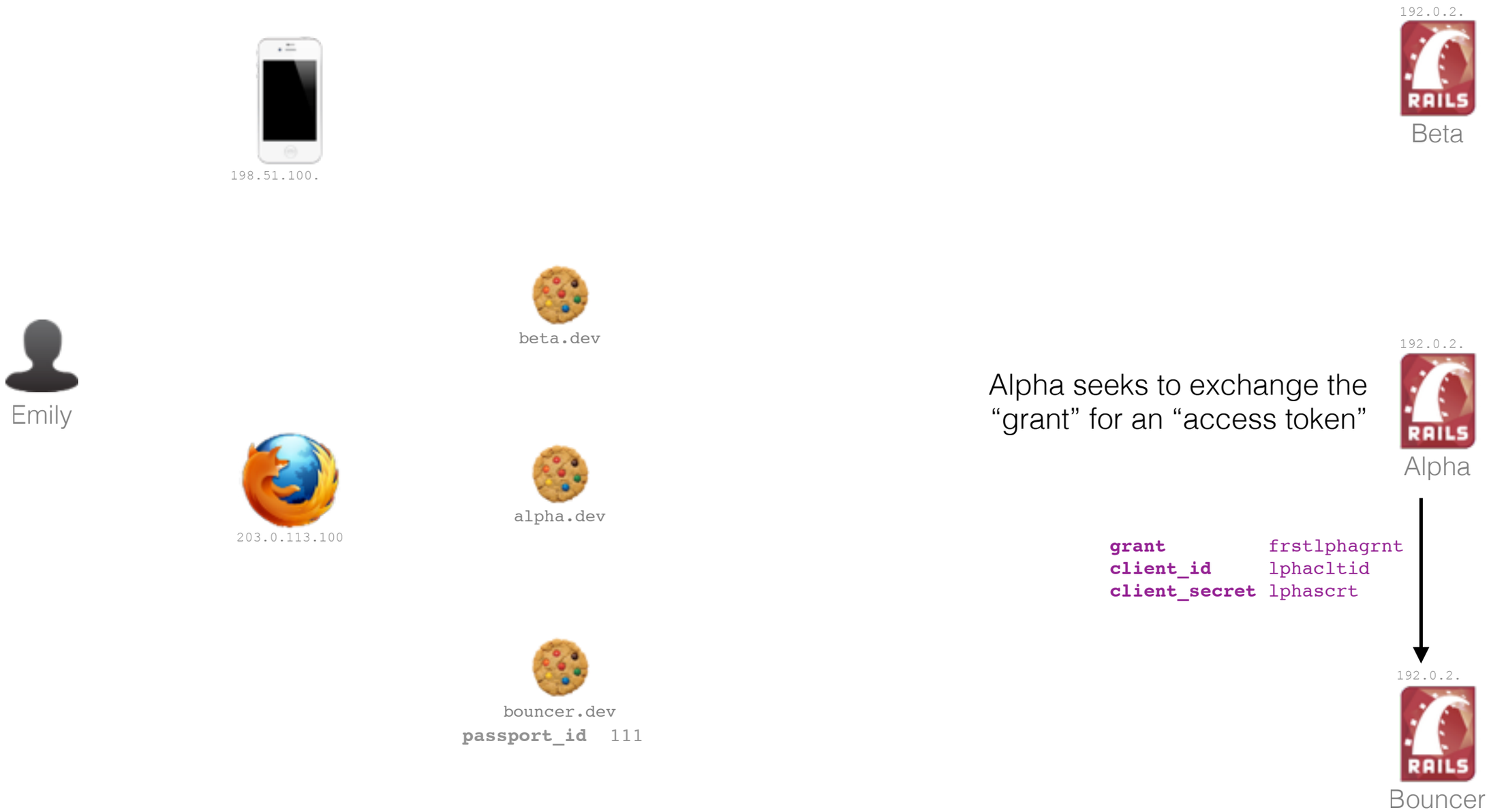
id	resource_owner_id	application_id	token
555	42	2	frstlphagrnt



The grant token is created and returned by the Doorkeeper middleware. I’m actually using a middleware myself to catch it before it leaves and attach it to the Passport I created earlier.

oauth_access_grants

id	resource_owner_id	application_id	token
555	42	2	frstlphagrnt



Alpha seeks to exchange the "grant" for an "access token"

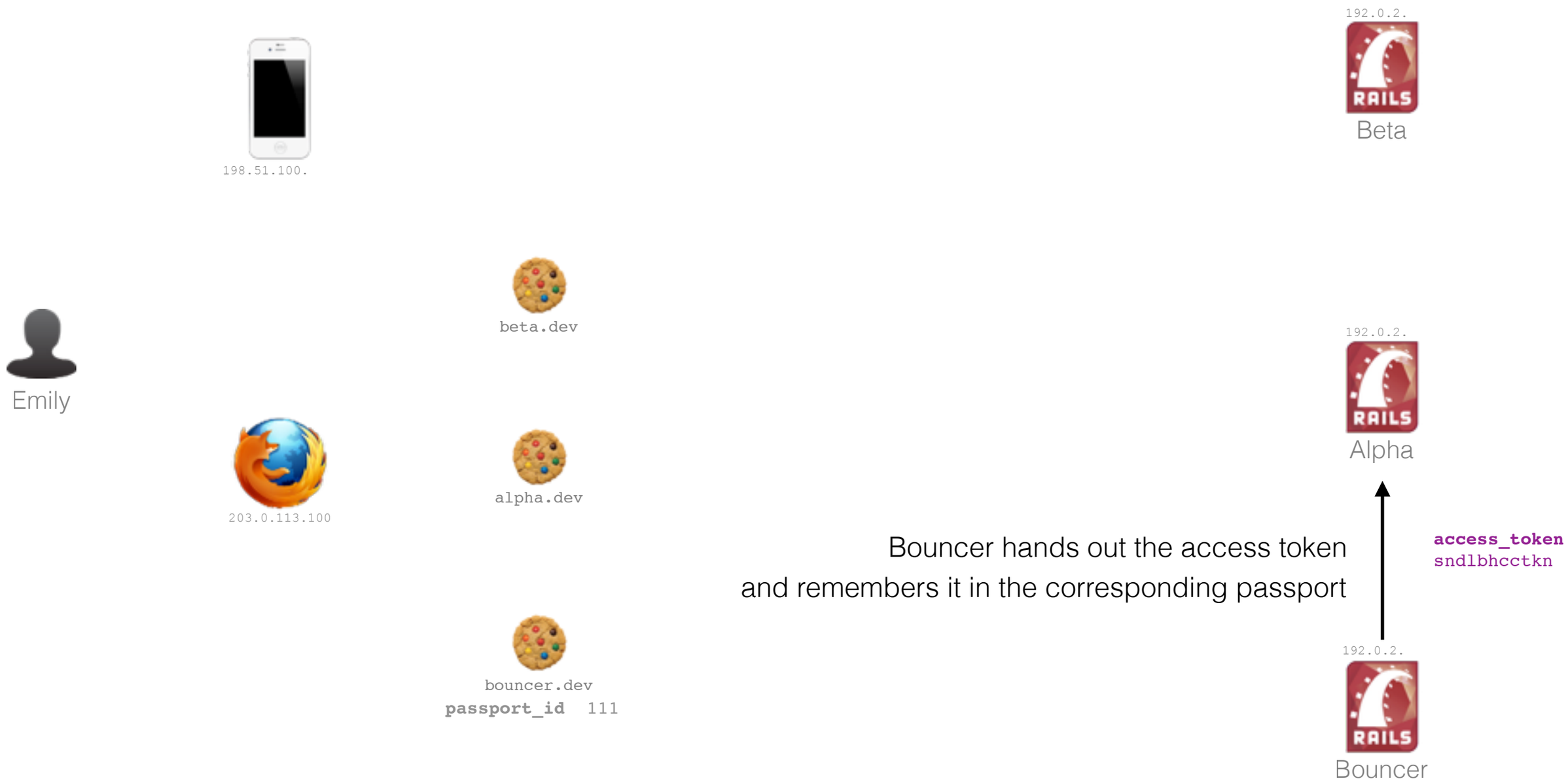
passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	555	-

This is till part of the OAuth 2 "Authorization Code Grant" Flow. The omniauth middleware handles these requests out-of-the-box.

oauth_access_grants

id	resource_owner_id	application_id	token
555	42	2	frstlphagrnt



1. Find the passport by grant_id

2. Remember the access token

passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	555	666

Doorkeeper-internal table

oauth_access_tokens

id	resource_owner_id	application_id	token
666	42	2	sndlbhcctkn

oauth_access_grants

id	resource_owner_id	application_id	token
555	42	2	frstlphagrnt

Doorkeeper handles the access token creation in the middleware. So here again I use a middleware myself to catch the outgoing access token and attach it to the passport. Note that there is no session cookie available at this point, which is why we need to lookup the passport by the grant token.

The logic for this API call resides in the omniauth-bouncer gem

```
module OmniAuth
  module Strategies
   class Bouncer < OmniAuth::Strategies::OAuth2

      def raw_info
        params = { ip: request.ip, agent: request.user_agent }
        @raw_info ||= access_token.post('/api/v1/passports/register', params: params).parsed
      end
    end
  end
end
```

198.51.100.

192.0.2.

Beta

beta.dev

Emily

192.0.2.

Alpha

203.0.113.100

alpha.dev

Alpha performs an API call to

bouncer.dev/api/v1/passports/register

access_token	sndlbhcctkn
ip	203.0.113.100
agent	Firefox

bouncer.dev

passport_id 111

Alpha also informs about Emily's IP and the Browser

192.0.2.

Bouncer

passports

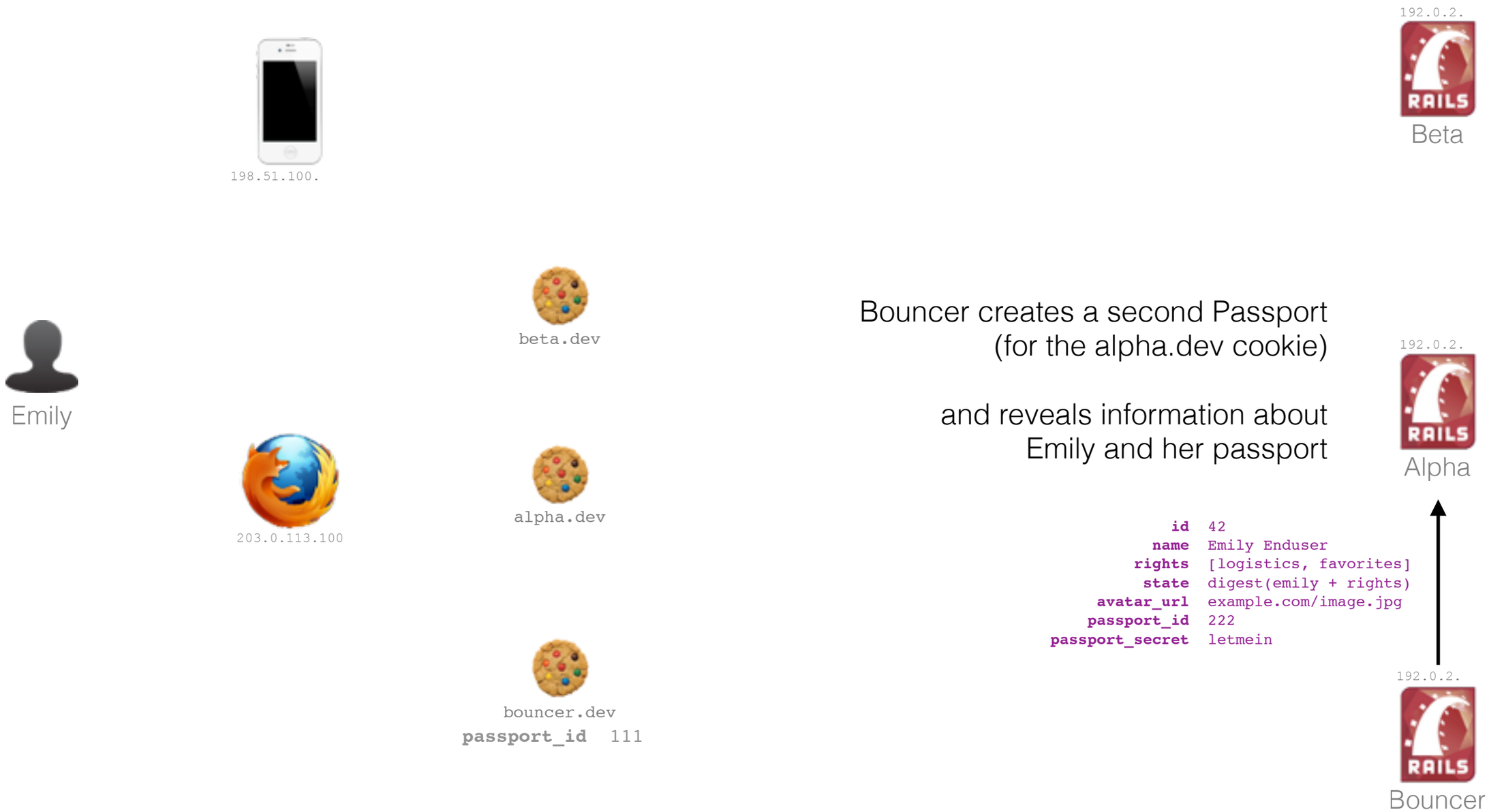
id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	555	666

oauth_access_tokens

id	resource_owner_id	application_id	token
666	42	2	sndlbhcctkn

oauth_access_grants

id	resource_owner_id	application_id	token
555	42	2	frstlphagrnt



Bouncer creates a second Passport
(for the alpha.dev cookie)

and reveals information about
Emily and her passport

passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	555	666
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666

Random uuid

The new passport has
the same group ID

Random String

Extracted the from
params sent by Alpha

oauth_access_tokens

id	resource_owner_id	application_id	token
666	42	2	sndlbhcctkn

oauth_access_grants

id	resource_owner_id	application_id	token
555	42	2	frstlphagrnt

This happens somewhere in `Alpha::SessionsController.oauth_success`

```
session.merge! request.env['omniauth.auth'].info.to_hash
```

198.51.100.

192.0.2.

Beta

beta.dev

Emily

Alpha saves this information in the cookie

192.0.2.

Alpha

203.0.113.100

alpha.dev

user_id	42
name	Emily Enduser
rights	[logistics, favorites]
state	mlrghtsdgst
avatar_url	example.com/image.jpg
passport_id	222
passport_secret	letmein

Login successful!

bouncer.dev

passport_id 111

192.0.2.

Bouncer

passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	555	666
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666

These tables here are now uninteresting for us

oauth_access_tokens

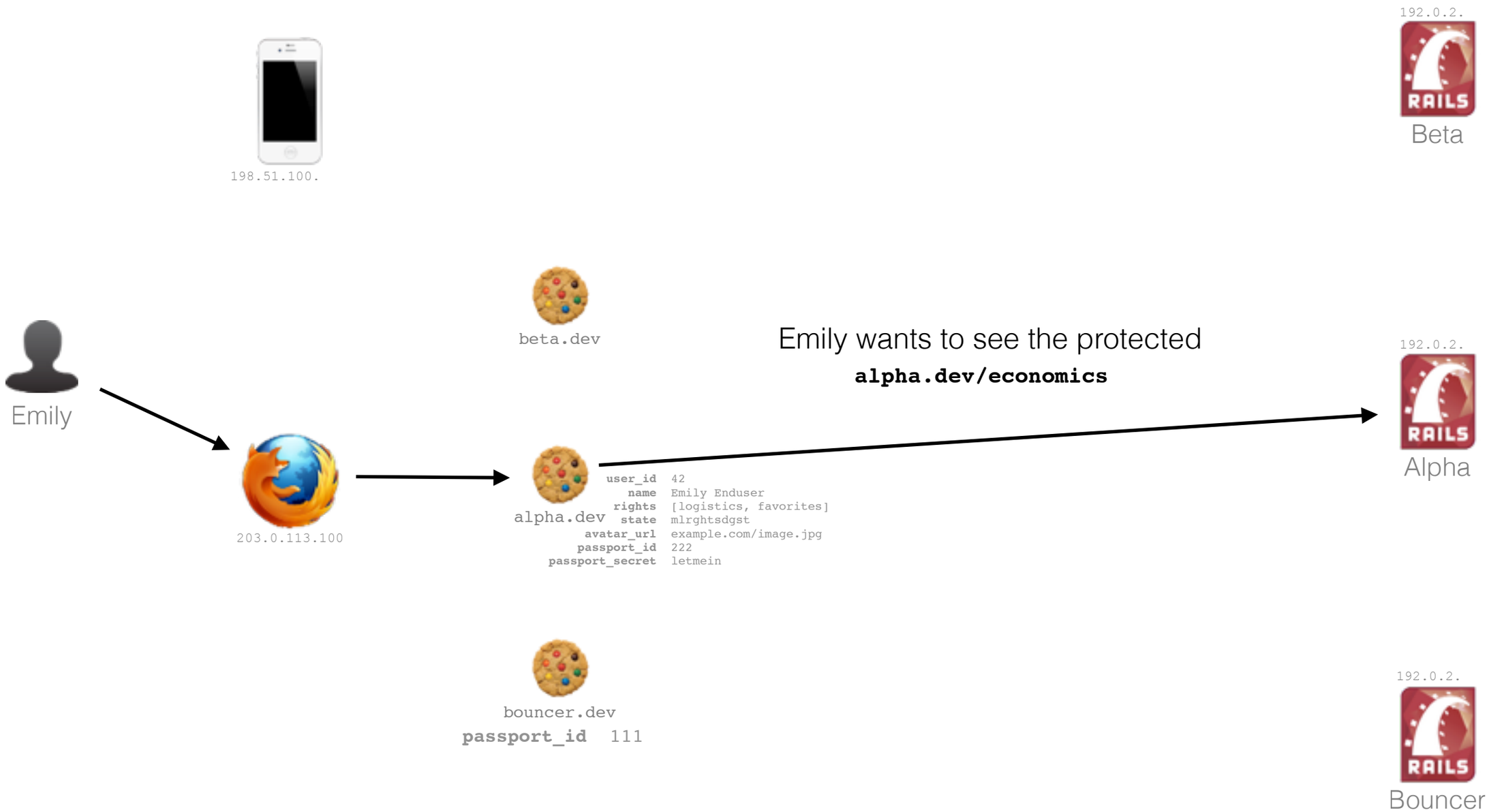
id	resource_owner_id	application_id	token
666	42	2	sndlbhcctkn

oauth_access_grants

id	resource_owner_id	application_id	token
555	42	2	frstlphagrnt

Part 2

Verifying authentication and authorization



passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	555	666
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666

Because this introduces a single point of failure, you might want to have a 1 second timeout on this request and if it fails due to a server error, assume that the user still has the same rights as before.

PS: Give the “signature” gem a try for signing the request.

passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	555	666
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666

E.g. Emily already logged out, changed her password, or revoked her session in her profile, or Emily was kicked out by an administrator.

198.51.100.

192.0.2.

Beta

beta.dev

Emily

192.0.2.

Alpha

203.0.113.100

alpha.dev

user_id	42
name	Emily Enduser
rights	[logistics, favorites]
state	mlrghtsdgst
avatar_url	example.com/image.jpg
passport_id	222
passport_secret	letmein

If the passport_secret is invalid

401

Report the problem

bouncer.dev

passport_id 111

192.0.2.

Bouncer

passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	555	666
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666

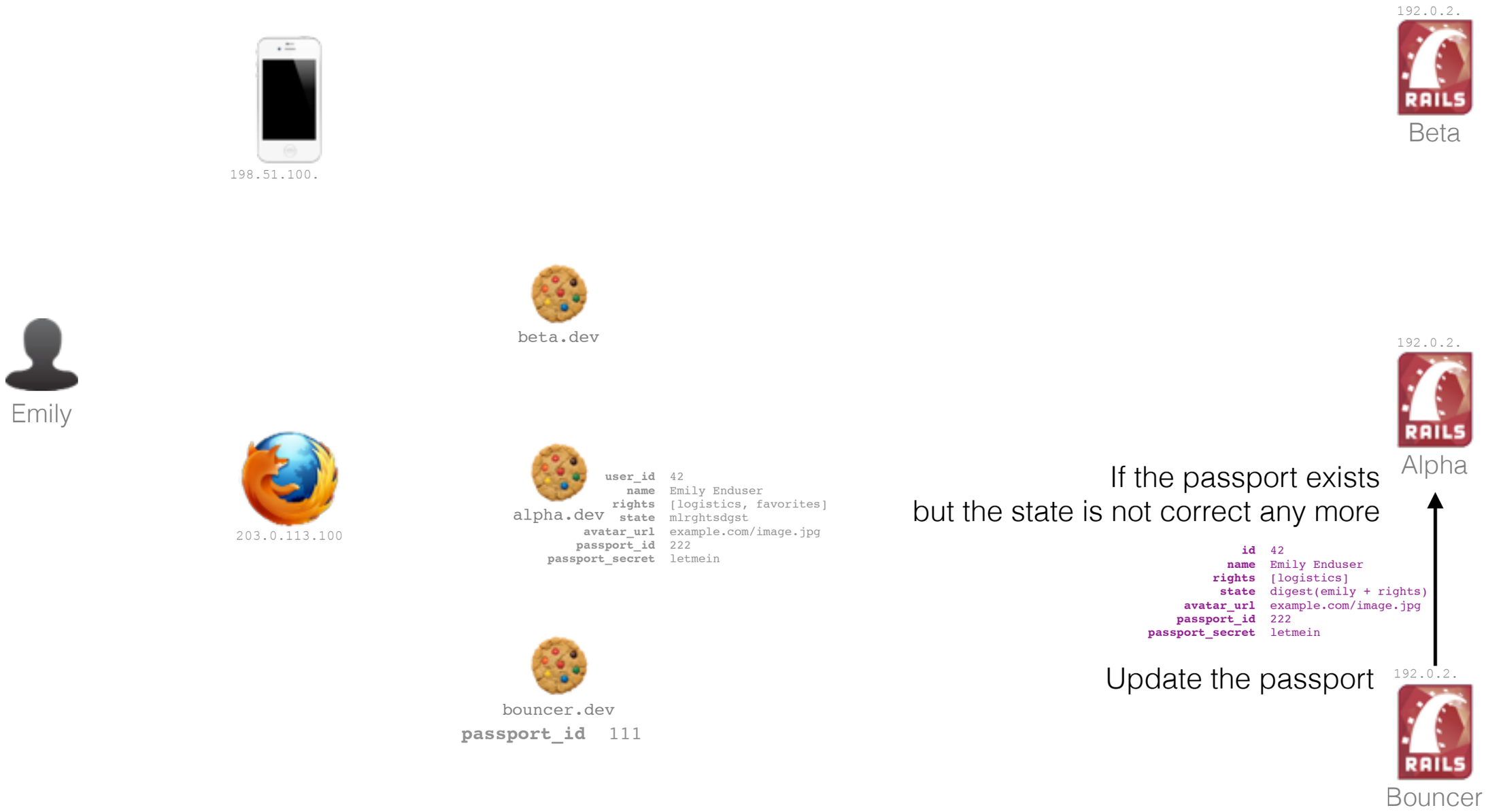
Authentication - “The passport exists and is valid”

Authorization - “Making sure the rights are still the same as before”

passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	555	666
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666

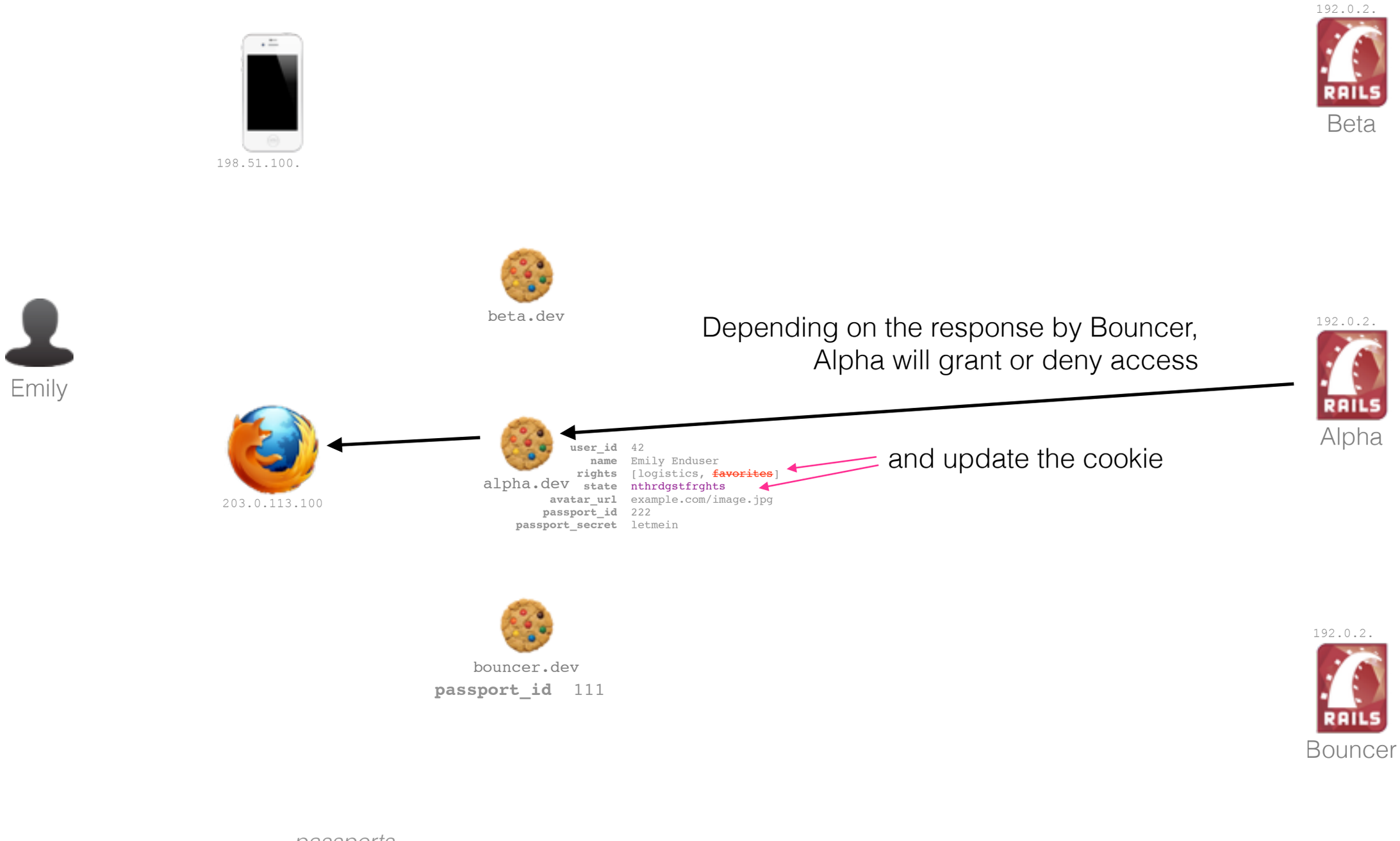
At this point you would also update IP, agent, and location of Passport 222



passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	555	666
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666

At this point you would also update IP, agent, and location of Passport 222

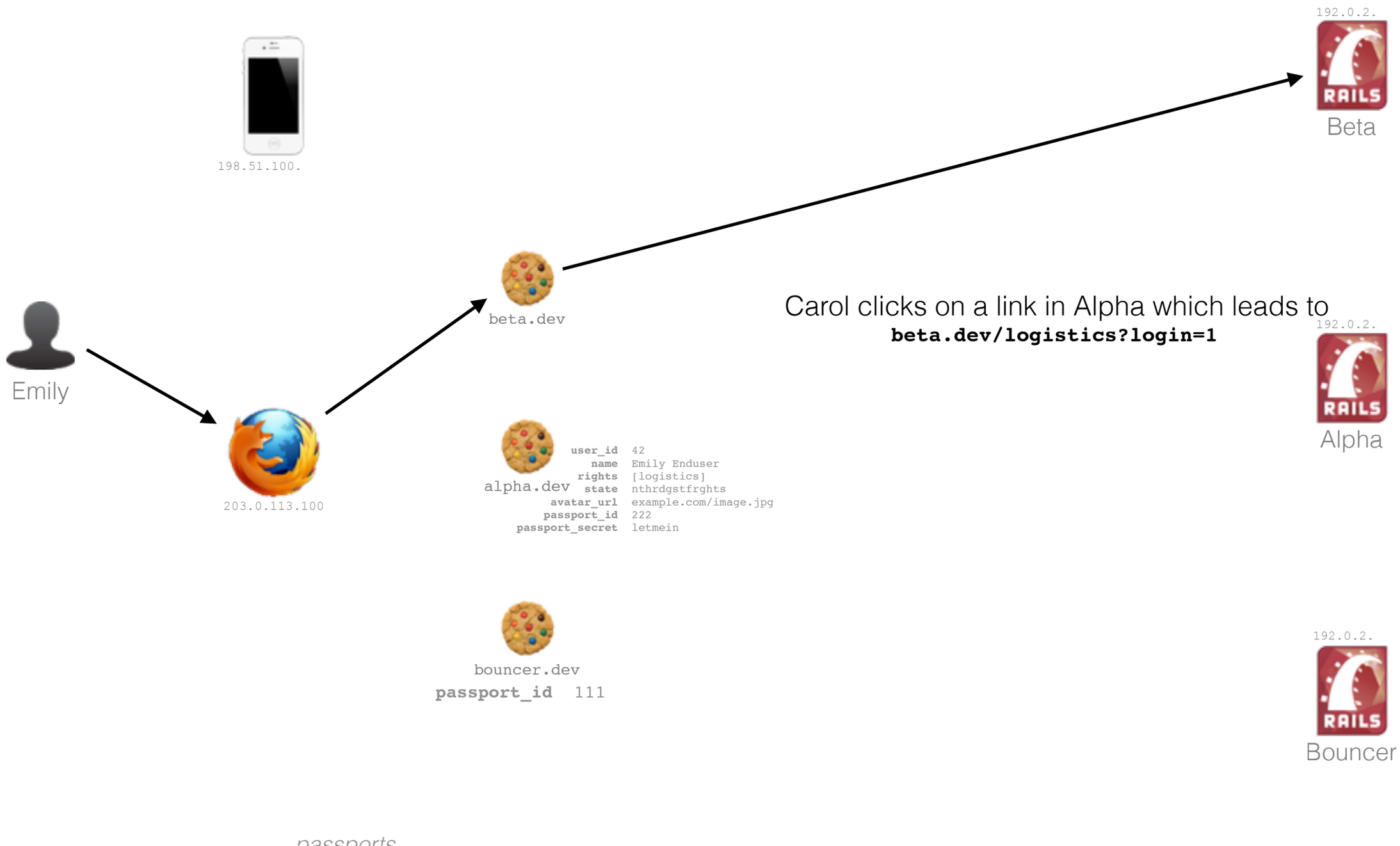


passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	555	666
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666

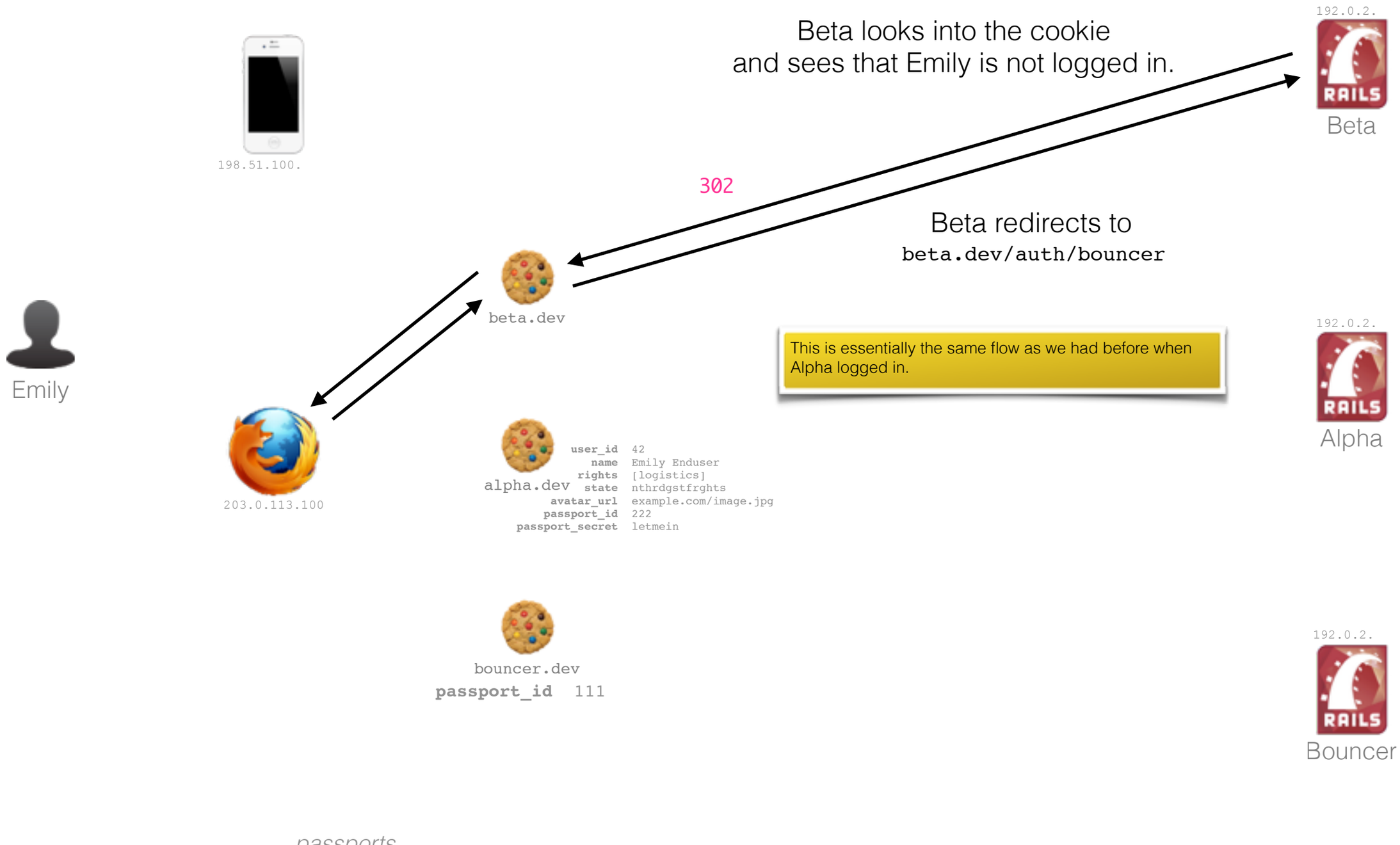
Part 3

Another client comes into play



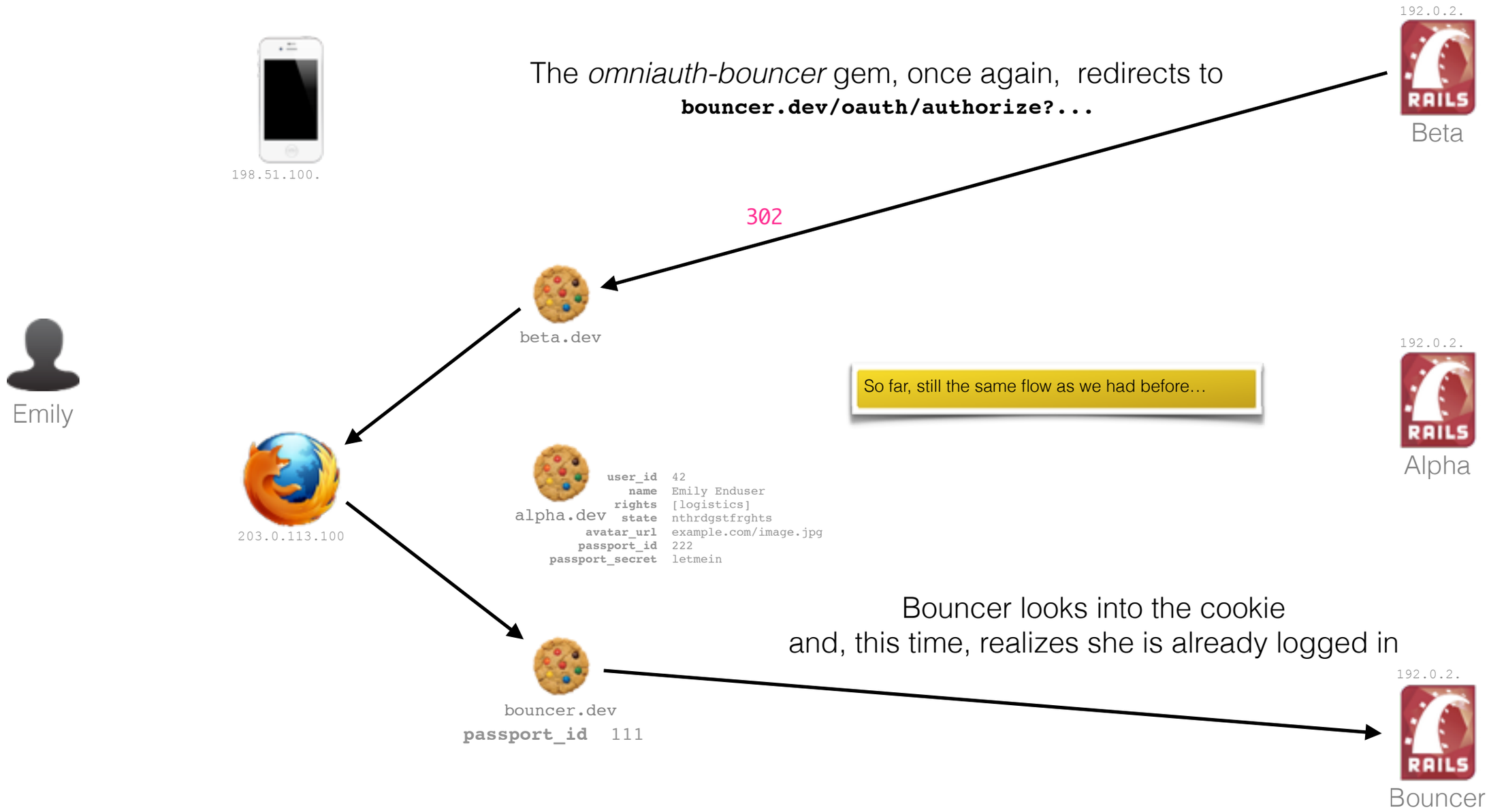
passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	555	666
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666



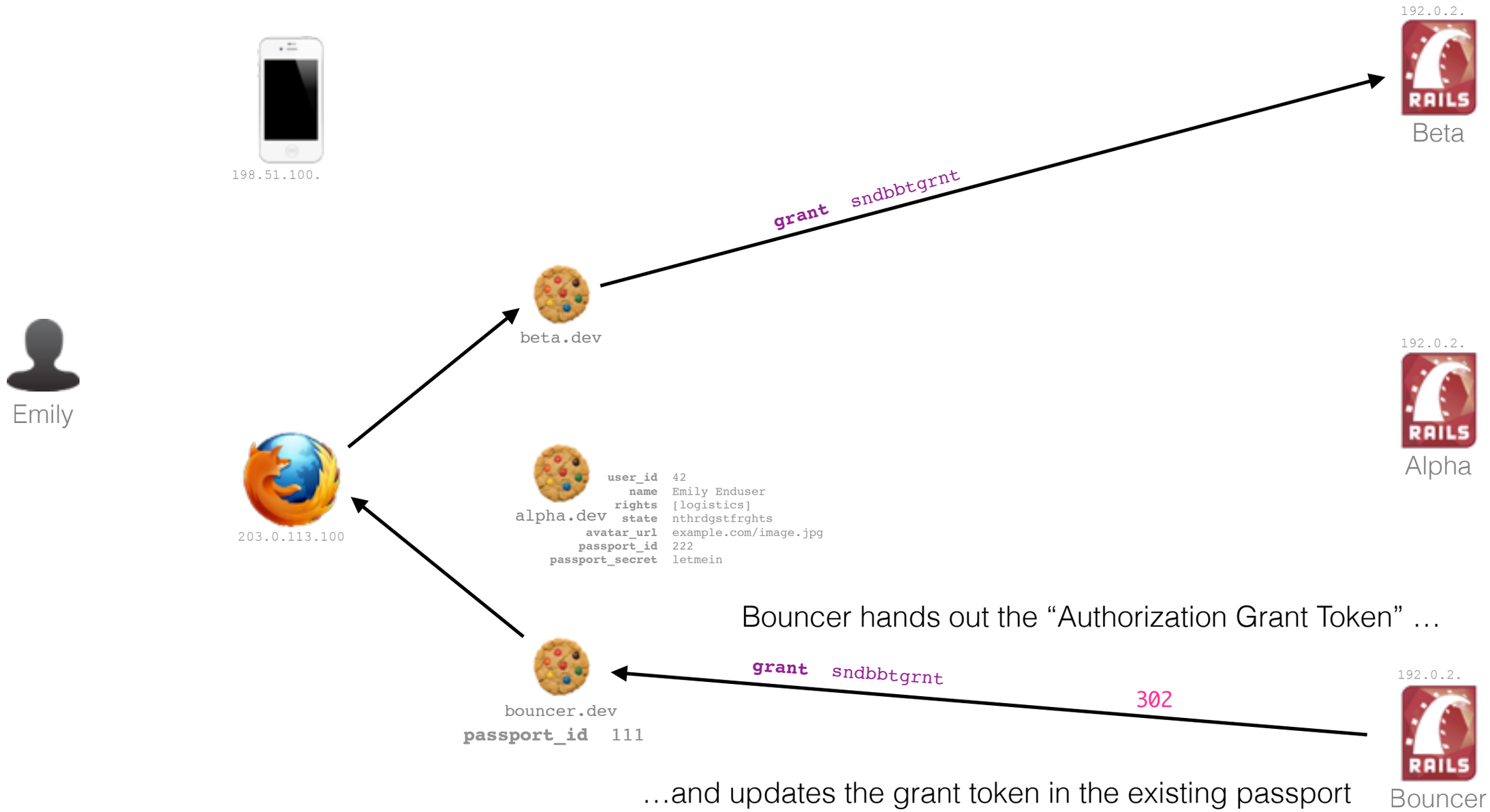
passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	555	666
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666



passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	555	666
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666



passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	777	666
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666

The passport is the one we find in the `bounder.dev` cookie, that is, with ID 111

oauth_access_grants

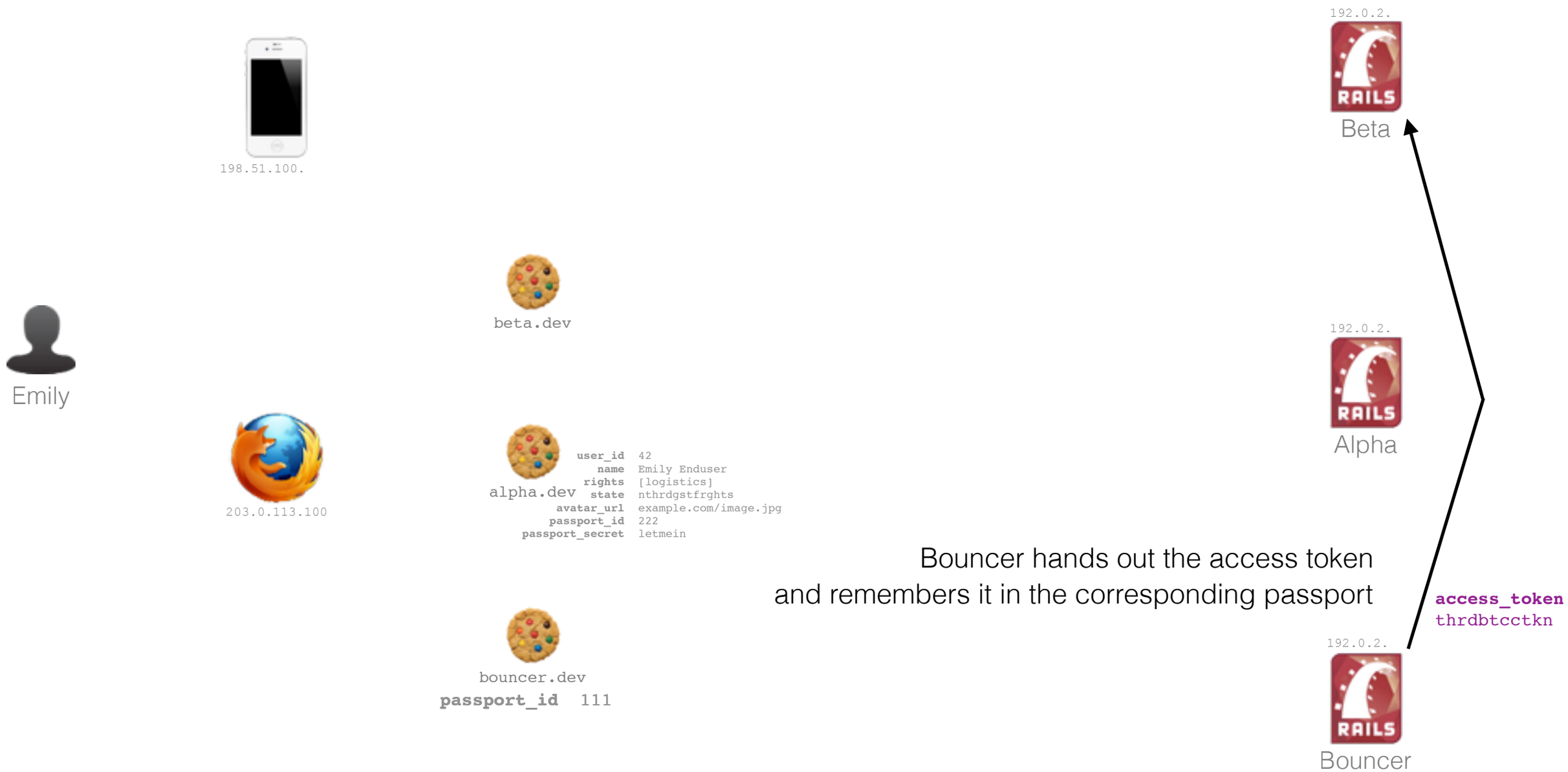
id	resource_owner_id	application_id	token
777	42	3	sndbbtgrnt

passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	777	666
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666

oauth_access_grants

id	resource_owner_id	application_id	token
777	42	3	sndbbtgrnt



1. Find the passport by grant_id

2. Update the access token

passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	777	999
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666

oauth_access_tokens

id	resource_owner_id	application_id	token
999	42	3	thrdbtcctkn

oauth_access_grants

id	resource_owner_id	application_id	token
777	42	3	sndbbtgrnt

Beta performs an API call to
bouncer.dev/api/v1/passports/register


198.51.100.


Emily


203.0.113.100


beta.dev


alpha.dev

user_id	42
name	Emily Enduser
rights	[logistics]
state	nthrdgstfrghts
avatar_url	example.com/image.jpg
passport_id	222
passport_secret	letmein


bouncer.dev
passport_id 111

192.0.2.

Beta

access_token	thrdbtcctkn
ip	203.0.113.100
agent	Firefox

192.0.2.

Alpha

192.0.2.

Bouncer

passports

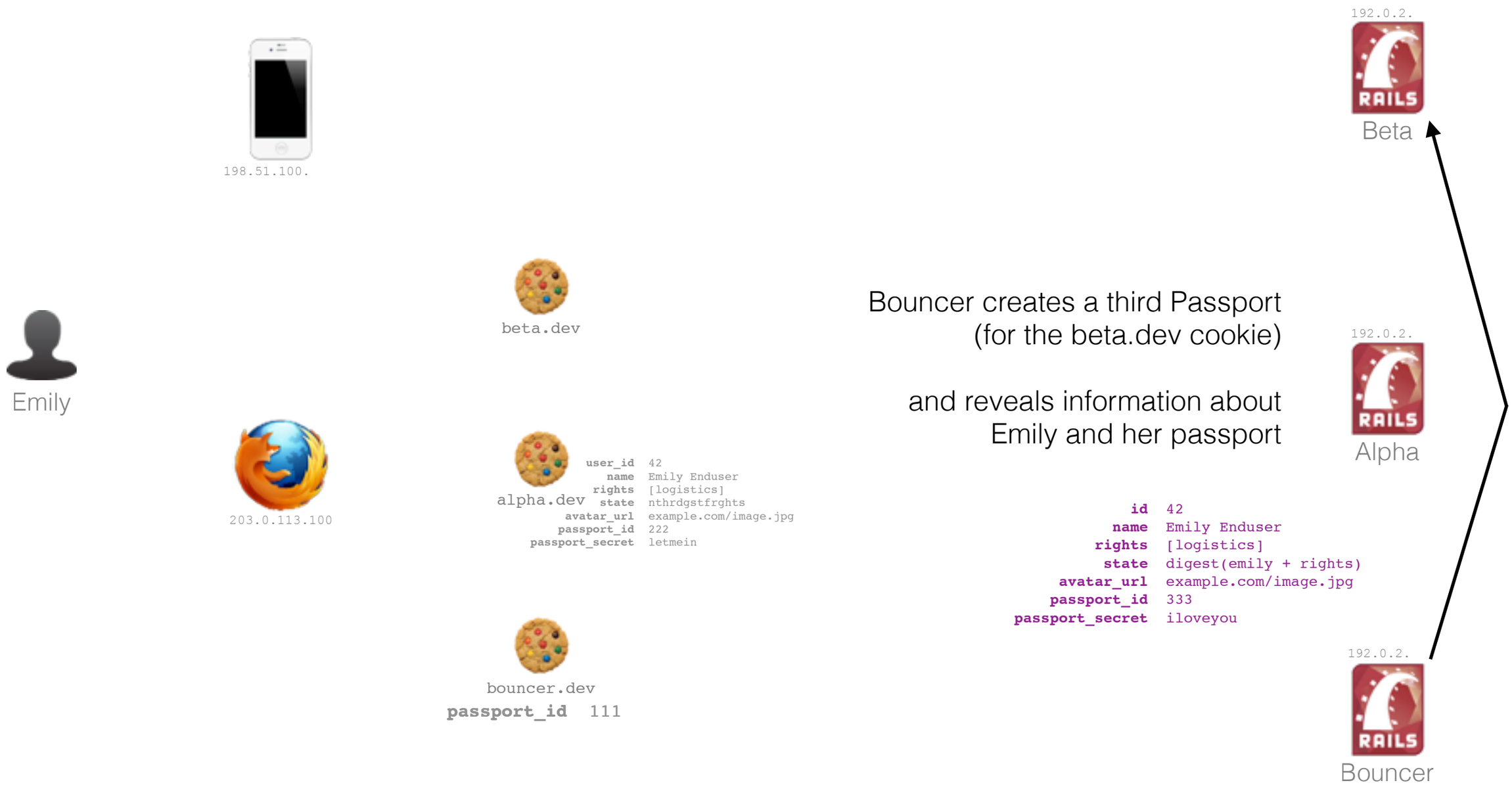
id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	777	999
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666

oauth_access_tokens

id	resource_owner_id	application_id	token
999	42	3	thrdbtcctkn

oauth_access_grants

id	resource_owner_id	application_id	token
777	42	3	sndbbtgrnt



passports

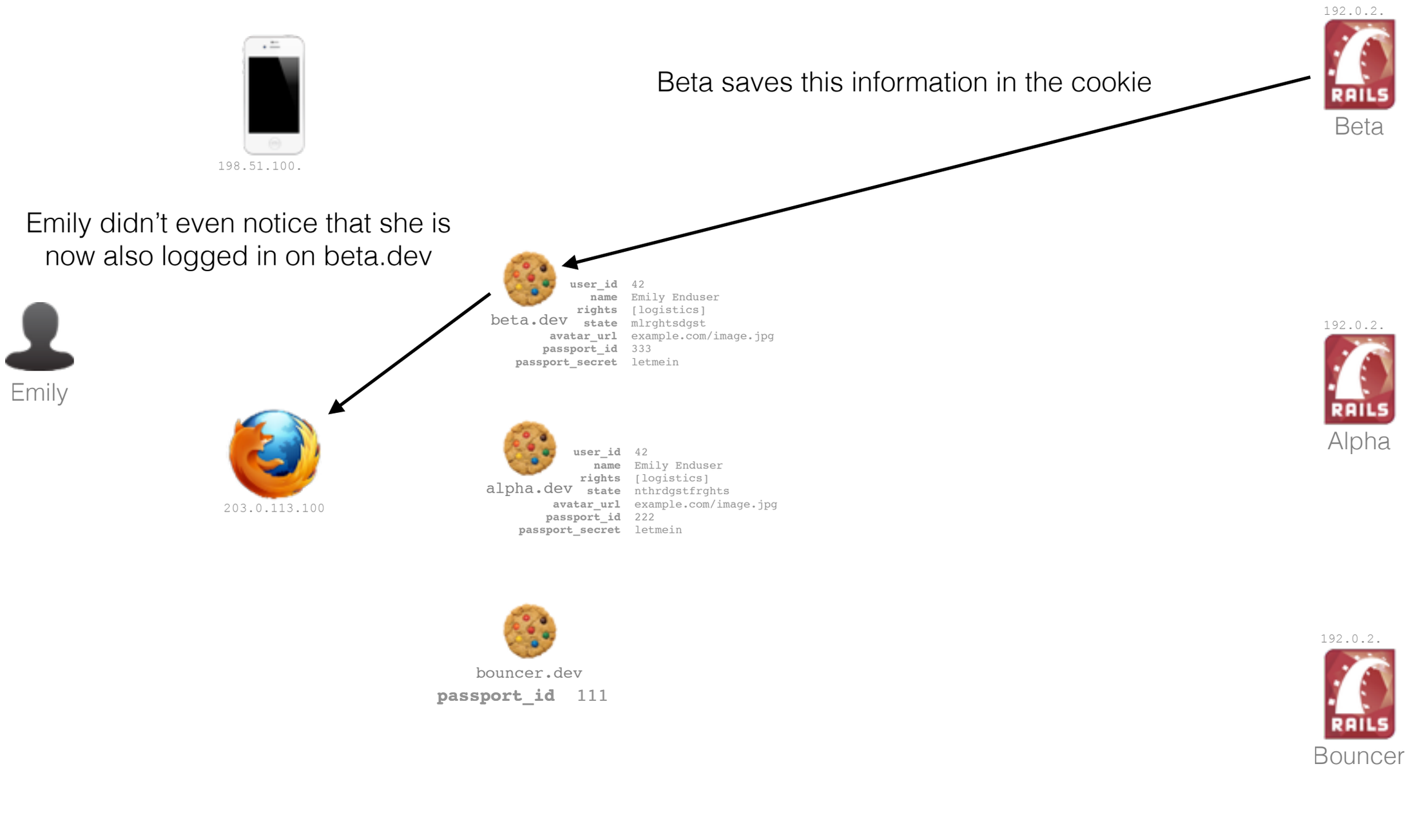
id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	777	999
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666
333	42	bncrgrp	iloveyou	203…	Firefox	Rome	-	999

Random uuid

The new passport has the same group ID

Random String

Extracted the from params sent by Alpha



passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	777	999
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666
333	42	bncrgrp	iloveyou	203…	Firefox	Rome	-	999

Part 4
Resource Owner Password Credentials Grant
(aka iPhone)

Note that there are no cookies involved at all in this flow.

iPhone requests an Access Token

198.51.100.

Emily

203.0.113.100

client_id	phnclntid
client_secret	phnclntscrt
username	emily.enduser
password	p4ssword

beta.dev

user_id	42
name	Emily Enduser
rights	[logistics]
state	mlrghtsdgst
avatar_url	example.com/image.jpg
passport_id	333
passport_secret	letmein

alpha.dev

user_id	42
name	Emily Enduser
rights	[logistics]
state	nthrdgstfrghts
avatar_url	example.com/image.jpg
passport_id	222
passport_secret	letmein

bouncer.dev

passport_id 111

192.0.2.

Beta

192.0.2.

Alpha

192.0.2.

Bouncer

passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	777	999
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666
333	42	bncrgrp	iloveyou	203…	Firefox	Rome	-	999

Bouncer knows which kind of flow this is, because we can look at `request.params['grant_type'] == 'password'` in the `Warden::Manager.after_authentication` hook.

198.51.100.

192.0.2.

Beta

Emily

beta.dev

user_id	42
name	Emily Enduser
rights	[logistics]
state	mlrghtsdgst
avatar_url	example.com/image.jpg
passport_id	333
passport_secret	letmein

access token frthcctkn

192.0.2.

Alpha

203.0.113.100

alpha.dev

user_id	42
name	Emily Enduser
rights	[logistics]
state	nthrdgstfrghts
avatar_url	example.com/image.jpg
passport_id	222
passport_secret	letmein

bouncer.dev

passport_id 111

Bouncer realizes that this is not an *Authorization Code Grant Flow* but a *Resource Owner Password Credentials Grant,* generates a new Passport and hands out the token

192.0.2.

Bouncer

passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	777	999
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666
333	42	bncrgrp	iloveyou	203…	Firefox	Rome	-	999
444	42	snglgrp	qwerty	198…	NativeApp	London	-	888

Random uuid

Random String

Random String

Extracted from the request (*not* params)

oauth_access_tokens

id	resource_owner_id	application_id	token
888	42	4	frthcctkn

iPhone performs an API call to
bouncer.dev/api/v1/passports/register

198.51.100.

access_token thrdbtcctkn

192.0.2.

Beta

Emily

beta.dev

user_id	42
name	Emily Enduser
rights	[logistics]
state	mlrghtsdgst
avatar_url	example.com/image.jpg
passport_id	333
passport_secret	letmein

192.0.2.

Alpha

203.0.113.100

alpha.dev

user_id	42
name	Emily Enduser
rights	[logistics]
state	nthrdgstfrghts
avatar_url	example.com/image.jpg
passport_id	222
passport_secret	letmein

bouncer.dev
passport_id 111

192.0.2.

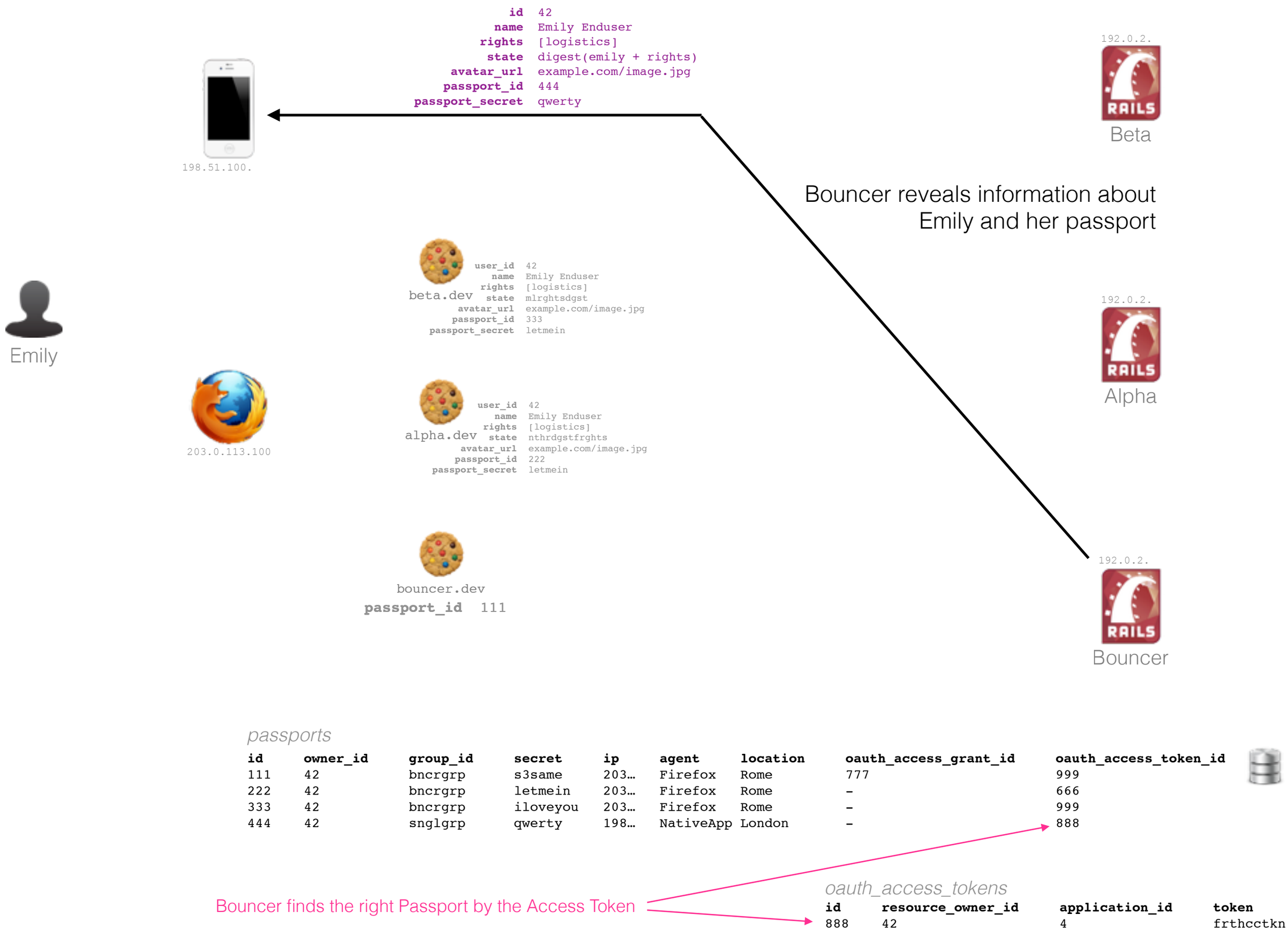
Bouncer

passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	777	999
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666
333	42	bncrgrp	iloveyou	203…	Firefox	Rome	-	999
444	42	snglgrp	qwerty	198…	NativeApp	London	-	888

oauth_access_tokens

id	resource_owner_id	application_id	token
888	42	4	frthcctkn

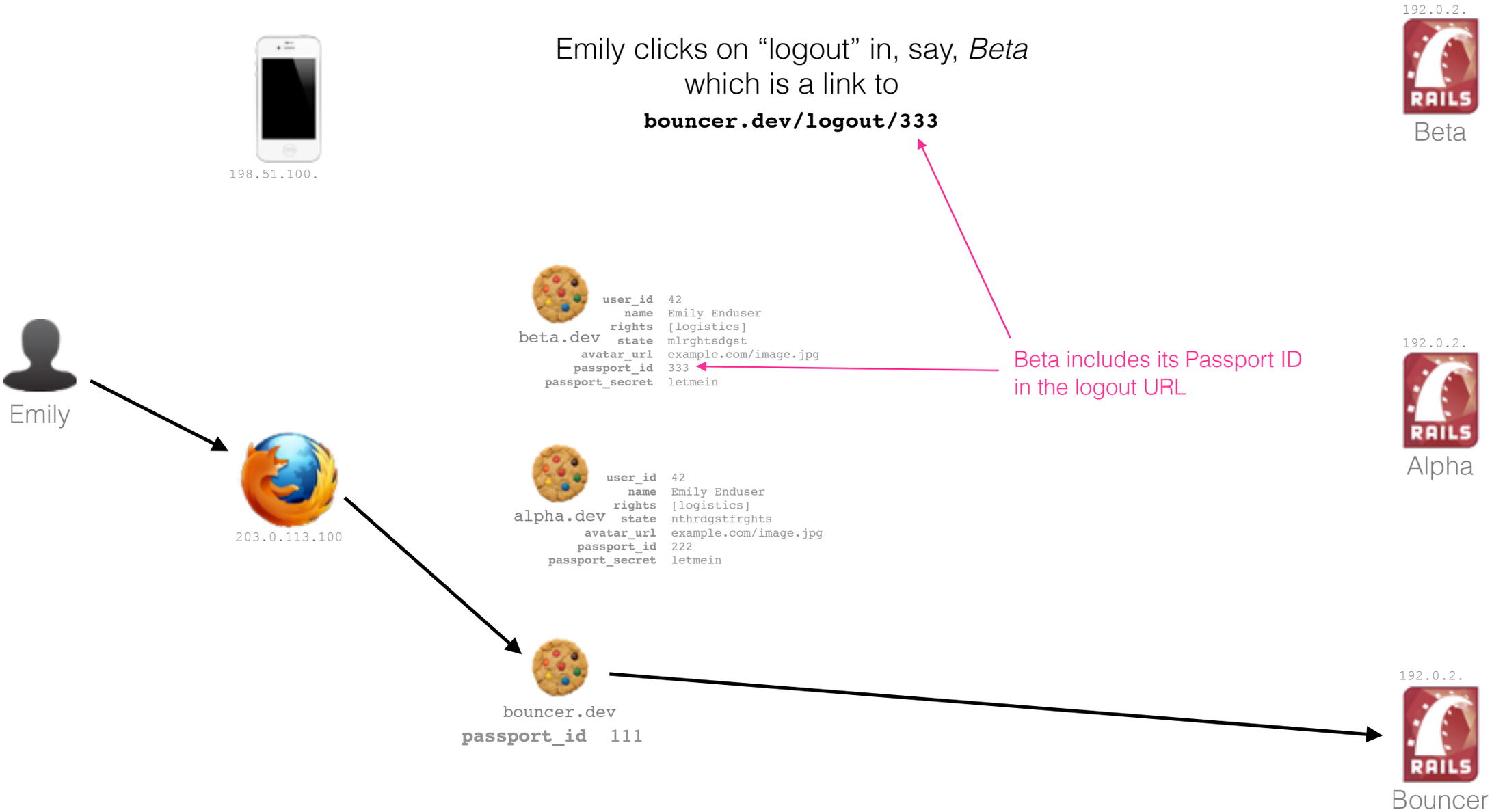

id 42
name Emily Enduser
rights [logistics]
state digest(emily + rights)
avatar_url example.com/image.jpg
passport_id 444
passport_secret qwerty
198.51.100.
192.0.2.
RAILS
Beta
Bouncer reveals information about Emily and her passport
beta.dev
user_id 42
name Emily Enduser
rights [logistics]
state mlrghtsdgst
avatar_url example.com/image.jpg
passport_id 333
passport_secret letmein
Emily
192.0.2.
RAILS
Alpha
203.0.113.100
alpha.dev
user_id 42
name Emily Enduser
rights [logistics]
state nthrdgstfrghts
avatar_url example.com/image.jpg
passport_id 222
passport_secret letmein
bouncer.dev
passport_id 111
192.0.2.
RAILS
Bouncer
passports
id owner_id group_id secret ip agent location oauth_access_grant_id oauth_access_token_id
111 42 bncrgrp s3same 203… Firefox Rome 777 999
222 42 bncrgrp letmein 203… Firefox Rome - 666
333 42 bncrgrp iloveyou 203… Firefox Rome - 999
444 42 snglgrp qwerty 198… NativeApp London - 888
Bouncer finds the right Passport by the Access Token
oauth_access_tokens
id resource_owner_id application_id token
888 42 4 frthcctkn

Part 5

Single sign out

Emily clicks on “logout” in, say, *Beta*
which is a link to

`bouncer.dev/logout/333`



passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	777	999
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666
333	42	bncrgrp	iloveyou	203…	Firefox	Rome	-	999
444	42	snglgrp	qwerty	198…	NativeApp	London	-	888

By revoking I mean setting Passport#revoked_at in the DB.
A session.reset will delete the bouncer.dev cookie.

198.51.100.

192.0.2.

Beta

Emily

203.0.113.100

beta.dev

user_id	42
name	Emily Enduser
rights	[logistics]
state	mlrghtsdgst
avatar_url	example.com/image.jpg
passport_id	333
passport_secret	letmein

192.0.2.

Alpha

alpha.dev

user_id	42
name	Emily Enduser
rights	[logistics]
state	nthrdgstfrghts
avatar_url	example.com/image.jpg
passport_id	222
passport_secret	letmein

bouncer.dev

passport_id 111

Bouncer finds its passport
and revokes every passport in the same group.

192.0.2.

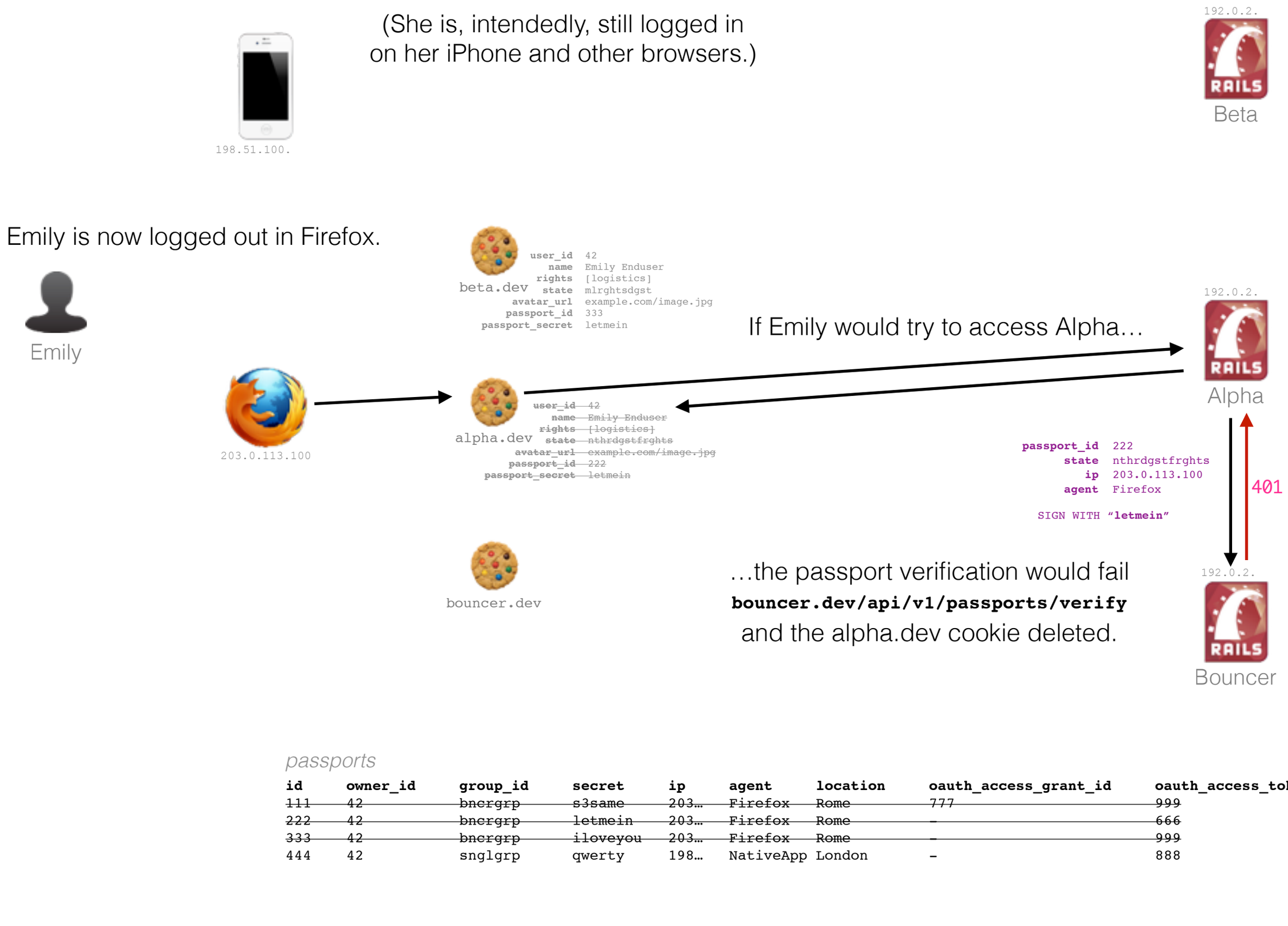
Bouncer

Finding the group_id by
Bouncer's Passport

passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
111	42	bncrgrp	s3same	203…	Firefox	Rome	777	999
222	42	bncrgrp	letmein	203…	Firefox	Rome	-	666
333	42	bncrgrp	iloveyou	203…	Firefox	Rome	-	999
444	42	snglgrp	qwerty	198…	NativeApp	London	-	888

Because the `bouncer.dev` cookie *might be* deleted,
find also Beta's Passport from the URL `bouncer.dev/logout/333`



passports

id	owner_id	group_id	secret	ip	agent	location	oauth_access_grant_id	oauth_access_token_id
~~111~~	~~42~~	~~bncrgrp~~	~~s3same~~	~~203…~~	~~Firefox~~	~~Rome~~	~~777~~	~~999~~
~~222~~	~~42~~	~~bncrgrp~~	~~letmein~~	~~203…~~	~~Firefox~~	~~Rome~~	~~–~~	~~666~~
~~333~~	~~42~~	~~bncrgrp~~	~~iloveyou~~	~~203…~~	~~Firefox~~	~~Rome~~	~~–~~	~~999~~
444	42	snglgrp	qwerty	198…	NativeApp	London	–	888