



intentionally left blank

getting django to play with old friends



getting django
to play with
old friends

or foes



Lynn Root

River Bar, 2013

Red Hat | @roguelynn | roguelynn.com

Lynn Root

freeipa.org

Lynn Root

freeipa.org

IPA != India Pale Ale

Lynn Root

freeipa.org

IPA == Identity, Policy, Audit

Lynn Root

freeipa.org

IPA == Identity, Policy, ~~Audit~~

Alpha

Playbill

ELI5: Kerberos

Setting up Custom User

External Authentication

External Permissions

rogue.ly/kerberos



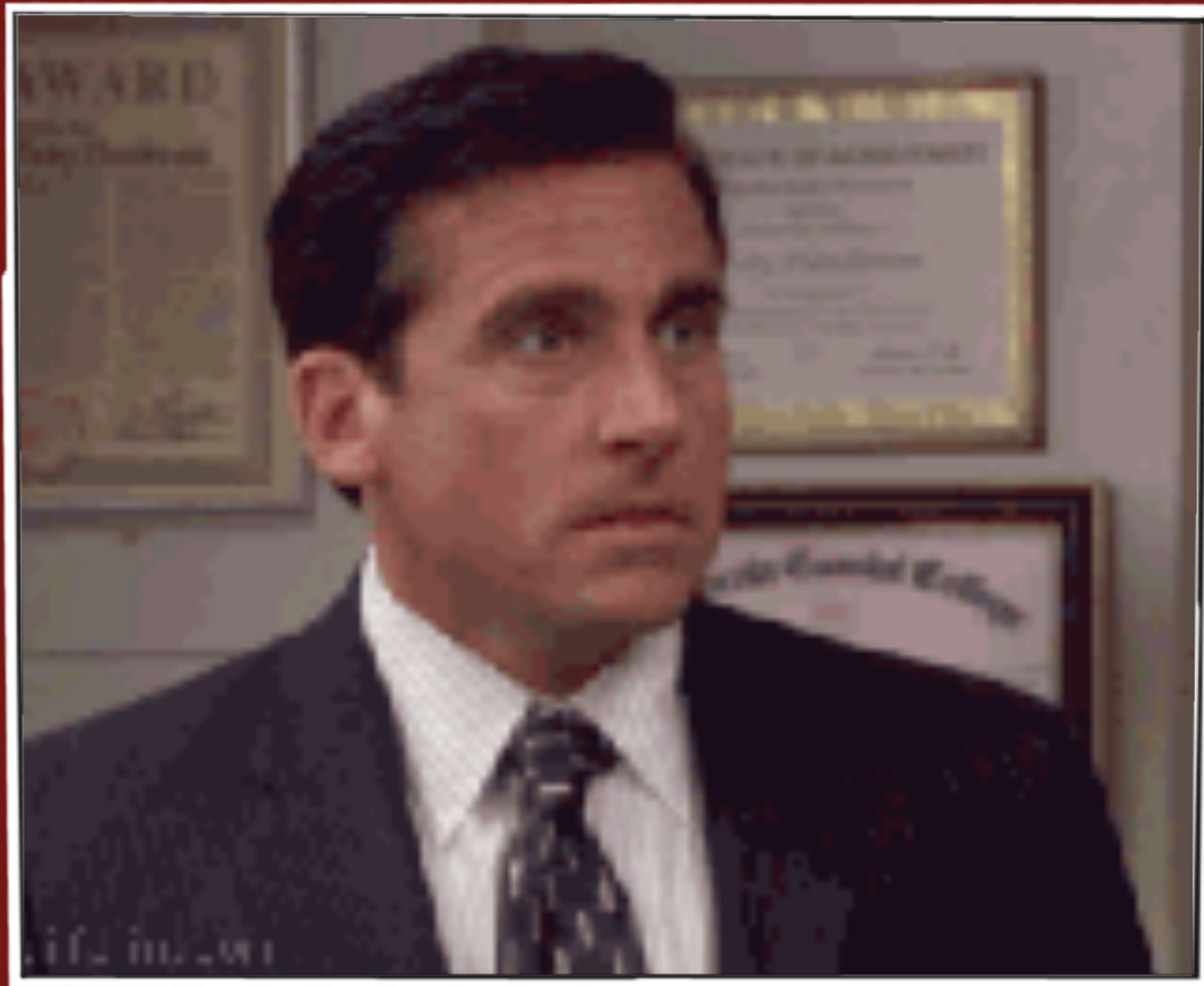
Problem:

Make an internal web app.

Question:
Can I use Postgres
for auth?

Crap.

I have to use single sign-on.



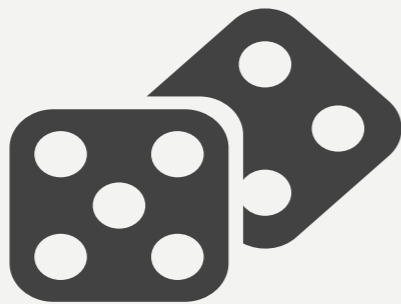
ELI-5: Kerberos



5-year-old you



who wants to play games



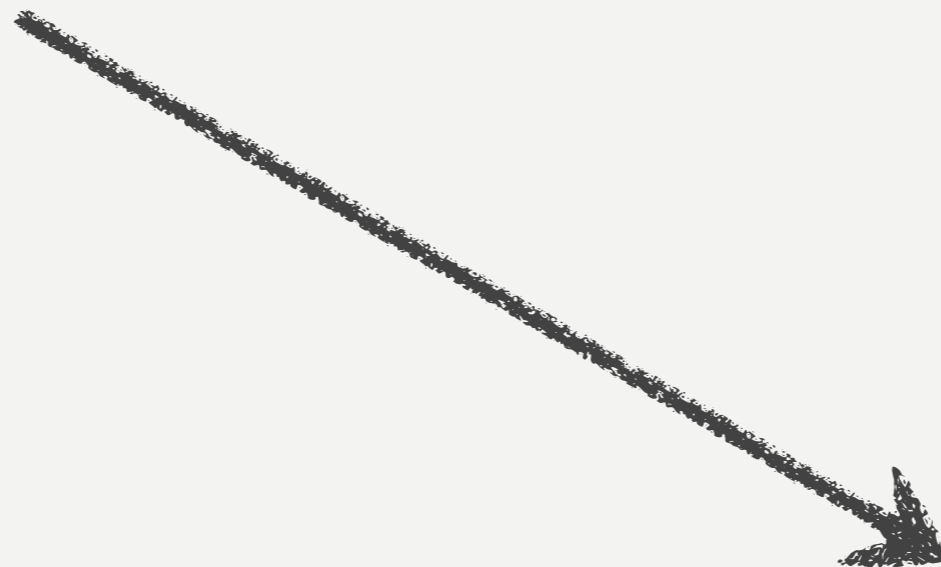


but you must ask Dad for \$\$





so you can buy a ticket to play





so Dad gives you a few bucks





and you go to the clerk
to buy a ticket to play



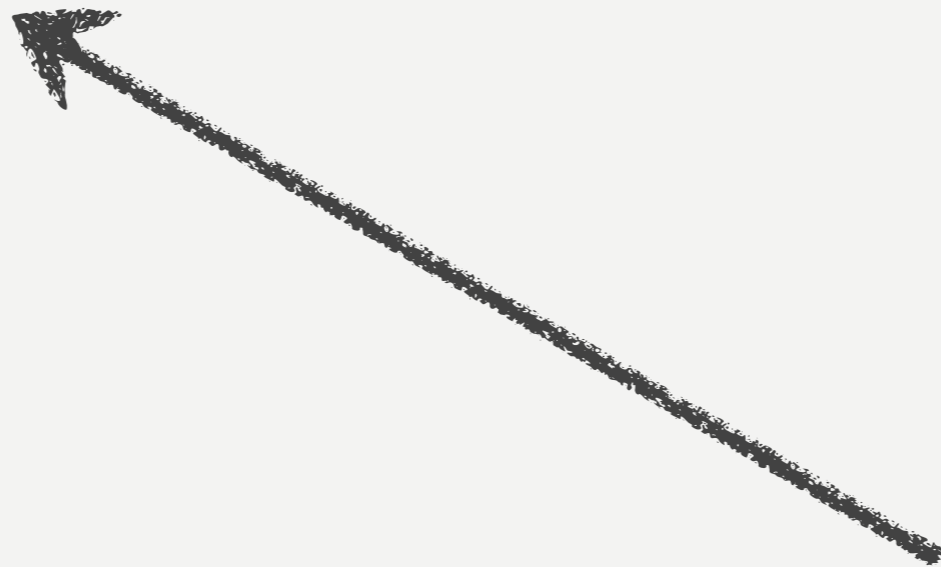


your money is good, so the
clerk gives you a ticket
for the game



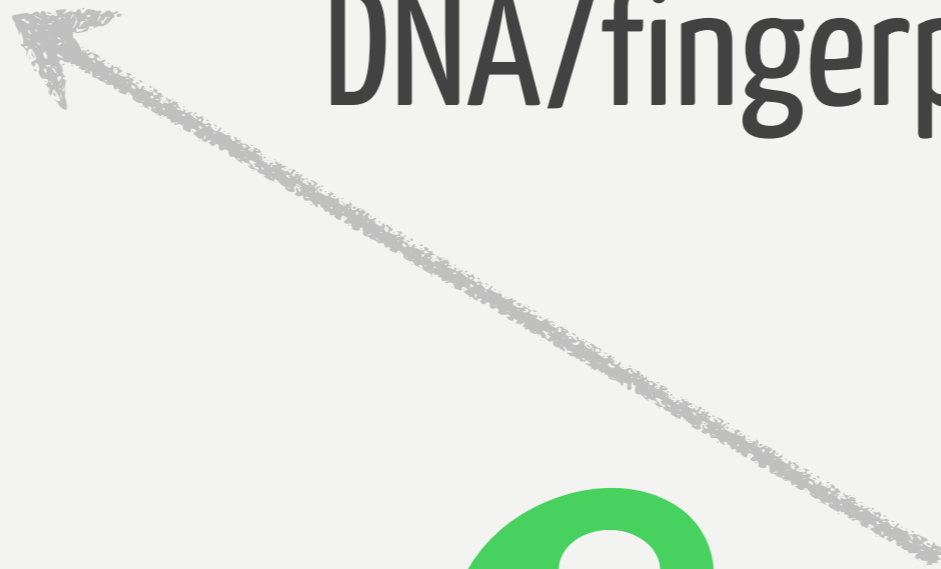


a ticket that only *you* can use



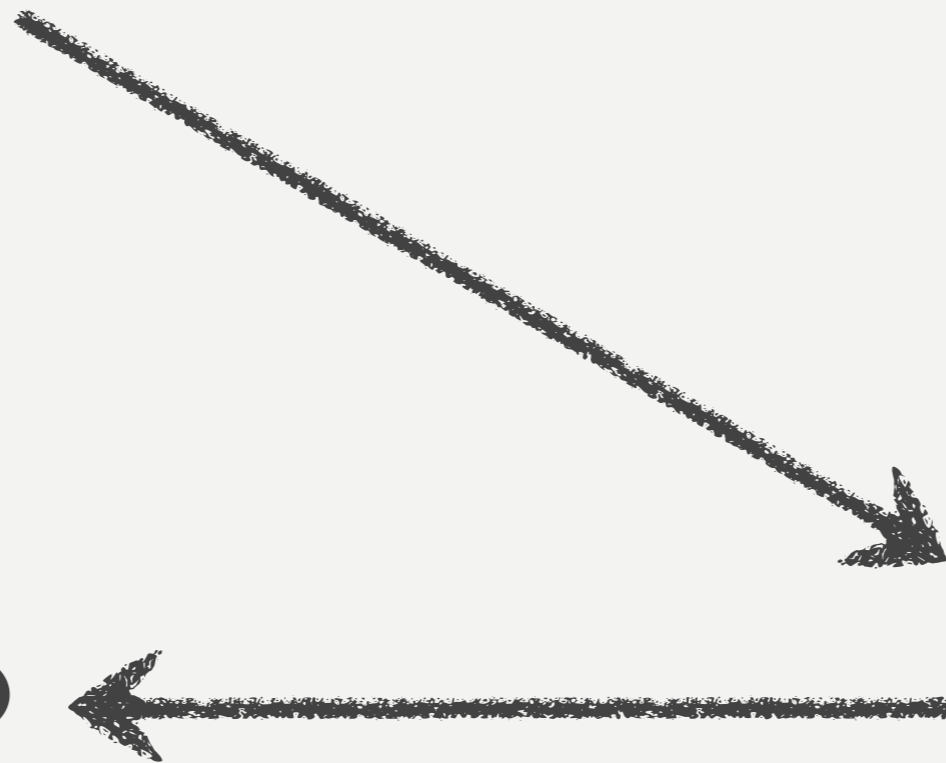
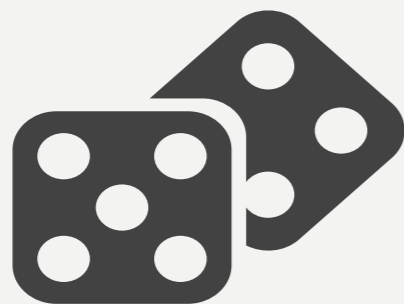


because ticket
as some super secret special
DNA/fingerprint tech



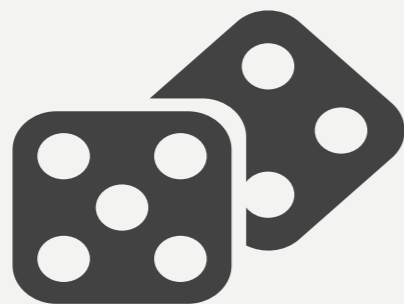
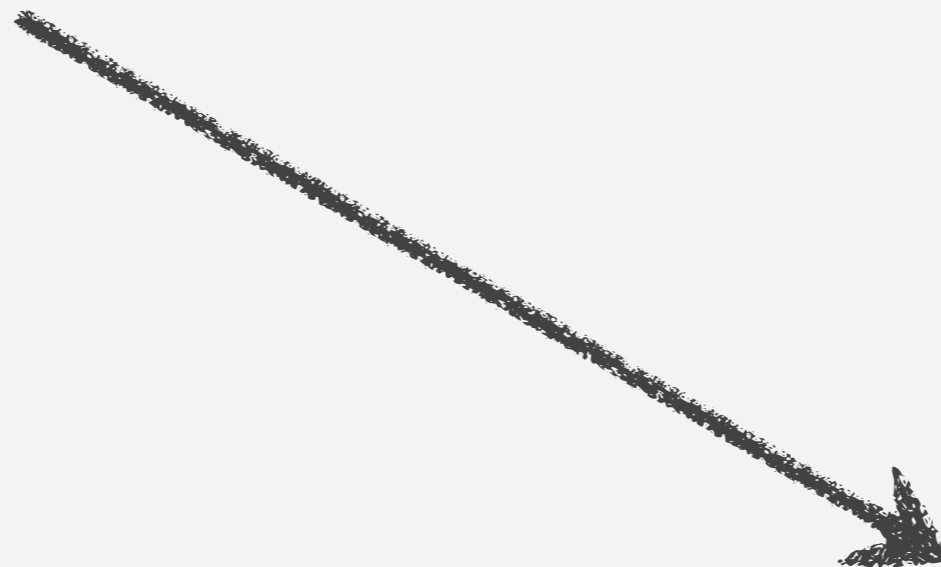


now you can play!





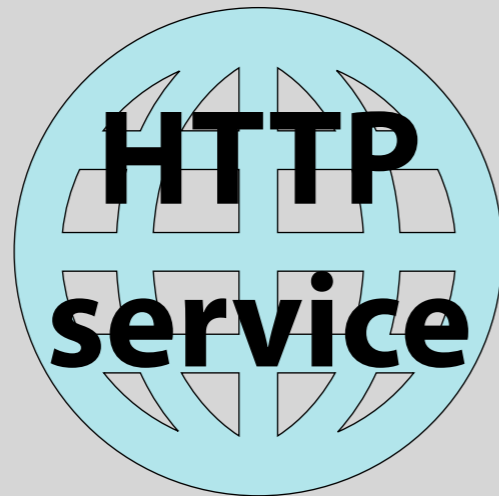
(until your money runs out)



ELI-25: Kerberos

- protocol for authentication
- based on tickets and keytabs
- avoids sending passwords over the internetz
- symmetric-key crypto

KDC Clients

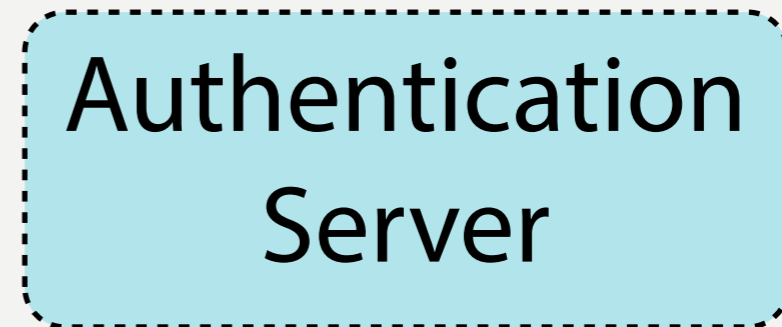


Key Distribution Center

Authentication
Server

Ticket Granting
Server

Kerberos Realm



plaintext request

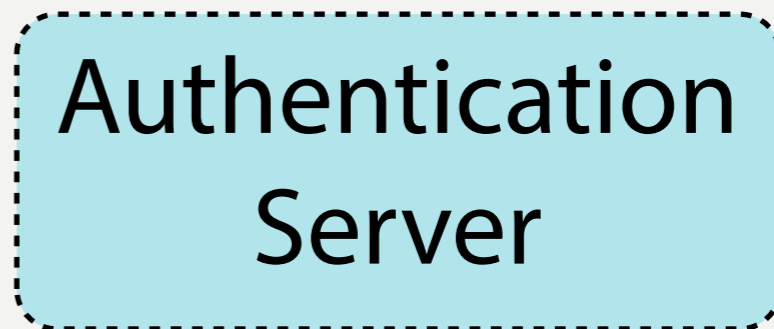
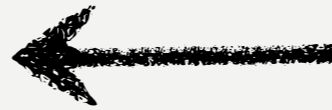
your ID, Ticket Granting Server ID,
IP address, lifetime



Authentication
Server



user ID
lookup in KDC



Ticket Granting Server Session Key

HTTP service's ID, timestamp, lifetime, TGS Session Key



Ticket Granting Ticket

your ID, HTTP service ID, IP address, timestamp, lifetime, and the TGS Session Key



Ticket Granting Server Session Key

🔒 Your Secret Key



Ticket Granting Ticket

🔒 Ticket Granting Server Secret Key



plaintext request

HTTP Service ID and lifetime



Ticket Granting
Server



Authenticator

your ID and timestamp



Ticket Granting Ticket

your ID, HTTP service ID, IP address,
timestamp, lifetime, and the TGS Session Key



Ticket Granting
Server



HTTP service
lookup in KDC



plaintext request



Ticket Granting Server



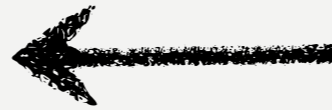
Authenticator

Ticket Granting Server Session Key



Ticket Granting Ticket

Ticket Granting Server Secret Key



Ticket Granting
Server



HTTP Service Session Key
your client ID and timestamp



Ticket for HTTP Service
your ID, HTTP service ID, IP address,
timestamp, lifetime, and the TGS Session Key



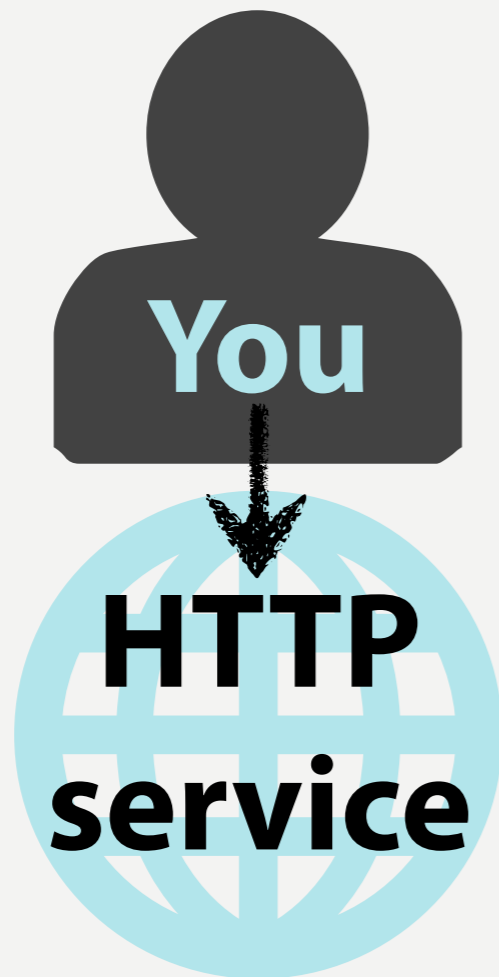
HTTP Service Session Key

🔓 Ticket Granting Server Session Key



Ticket for HTTP Service

🔒 HTTP Service Secret Key



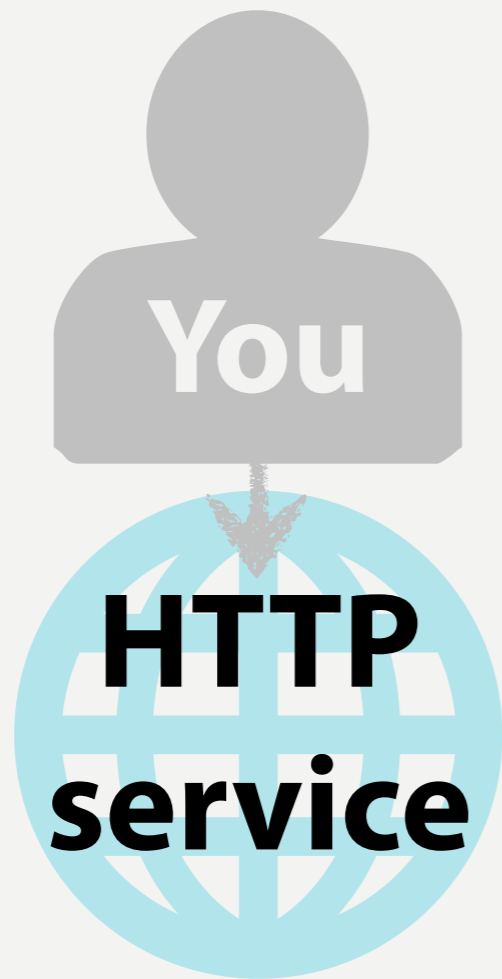
Ticket for HTTP Service

your ID, HTTP service ID, IP address,
timestamp, lifetime,
and the TGS Session Key



Authenticator

your client ID and timestamp



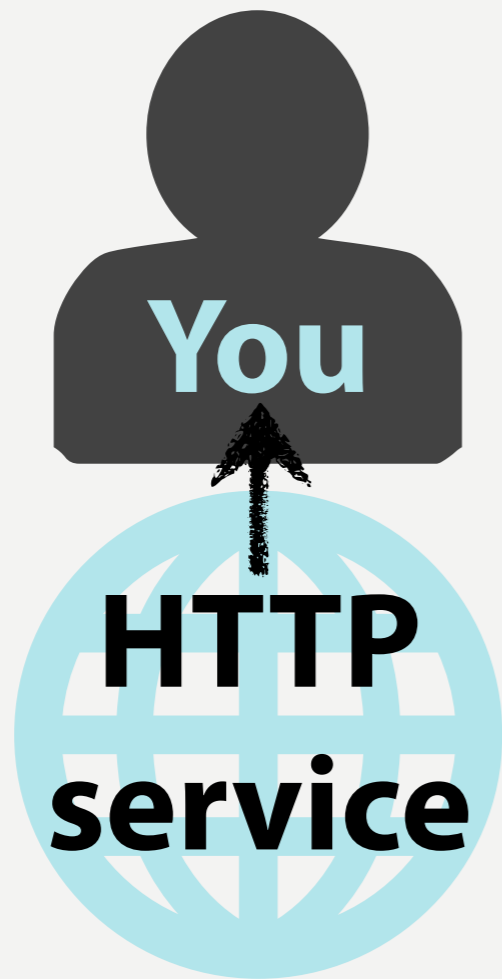
Ticket for HTTP Service

 **HTTP Service Secret Key**



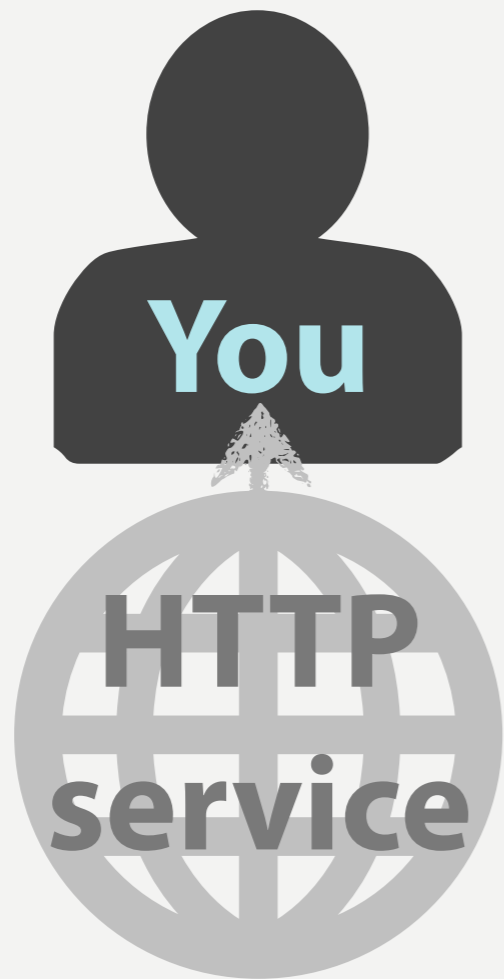
Authenticator

 **HTTP Service Session Key**



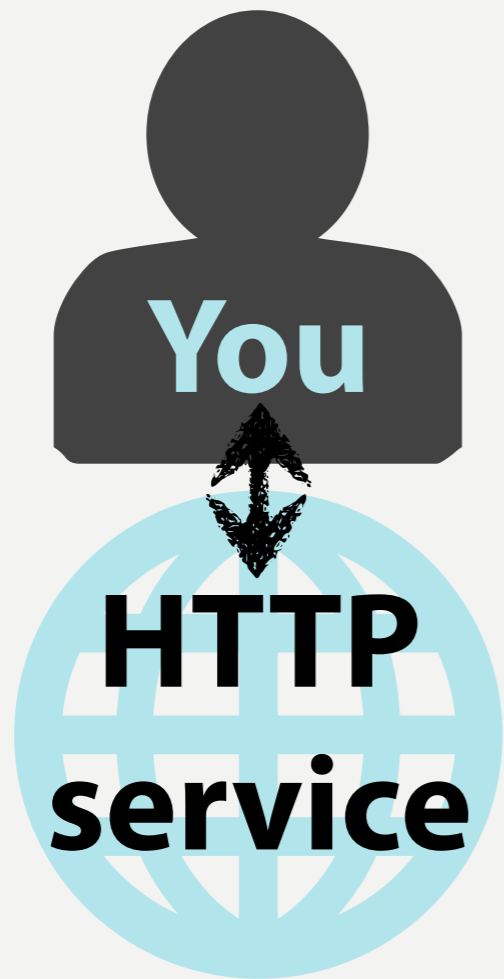
Authenticator

HTTP service ID and timestamp



Authenticator

 **HTTP Service Session Key**



how can Django play with
Kerberos?

leveraging
new User models

+

antiquated authentication

ZOMG

new user modelzz !1!!

what does that mean?

allows custom
user identifiers

```
./manage.py startapp  
synegizerApp
```

user model:


```
# synergizerApp/models.py
from django.contrib.auth.models import AbstractBaseUser

class KerbUser(AbstractBaseUser):
    username = models.CharField(max_length=254, ...)
    first_name = models.CharField(...)
    last_name = models.CharField(...)
    email = models.EmailField(...)

    synergy_level = models.IntegerField()
    is_team_player = models.BooleanField(default=False)

    USERNAME_FIELD = 'username'
    REQUIRED_FIELDS = ['email', 'synergy_level']
```

user manager:

```
# synergizerApp/models.py
```

```
from django.contrib.auth.models import (  
    AbstractBaseUser, BaseUserManager)
```

```
class KerbUserManager(BaseUserManager):  
    def create_user(self, email, synergy_level,  
                    password=None):  
        user = self.model(email=email,  
                           synergy_level=synergy_level)  
  
        # <--snip-->  
        return user  
  
    def create_superuser(self, email, synergy_level,  
                          password):  
        user = self.create_user(email, synergy_level,  
                                 password=password)  
        user.is_team_player = True  
        user.save()  
        return user
```

```
# synergizerApp/models.py

from django.contrib.auth.models import (
    AbstractBaseUser, BaseUserManager)

...

class KerbUser(AbstractBaseUser):
    # <--snip-->

    objects = KerbUserManager()
```

settings.py

```
# settings.py
```

```
AUTH_USER_MODEL = 'synergizerApp.KerbUser'
```

```
MIDDLEWARE_CLASSES = (
```

```
...
```

```
'django.contrib.auth.middleware.AuthenticationMiddleware',
```

```
'django.contrib.auth.middleware.RemoteUserMiddleware',
```

```
...
```

```
)
```

```
AUTHENTICATION_BACKENDS = (
```

```
'django.contrib.auth.backends.RemoteUserBackends',
```

```
)
```

team player!

pointyhairedboss@
STRATEGERY.COM

pointyhairedboss@
STRATEGERY.COM

Within Django App

```
# synergizerApp/krb5.py
```

```
from django.contrib.auth.backends import (  
    RemoteUserBackend)
```

```
class Krb5RemoteUserBackend(RemoteUserBackend):  
    def clean_username(self, username):  
        # remove @REALM from username  
        return username.split("@")[0]
```

client-centric!!1!

```
# settings.py

AUTH_USER_MODEL = 'synergizerApp.KerbUser'

MIDDLEWARE_CLASSES = (
    ...
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.auth.middleware.RemoteUserMiddleware',
    ...
)

AUTHENTICATION_BACKENDS = (
    'synergizerApp.krb5.Krb5RemoteUserBackend',
)
```

a streamlining team player!

Within Apache

```
# /etc/httpd/conf.d/remote_user.conf

<Location />
    AuthType Kerberos

    # <--snip-->
    KrbLocalUserMapping On
    # <--snip-->
</Location>
```

Accessing the user

HTTP Request

HTTP Response

Some Middleware

Some more Middleware

AuthenticationMiddleware

Even more Middleware

Django view functions



HTTP Request

HTTP Response

Some Middleware

Some more Middleware

RemoteUserMiddleware

Even more Middleware

Django view functions



Accessing user

```
user = request.META["REMOTE_USER"]
```

Accessing user

```
user = request.META["REMOTE_USER"]
```



```
pointyhairedboss
```


how do I
Apache?

environment:
Kerberos + Apache

```
# /etc/httpd/conf.d/remote_user.conf
```

```
LoadModule auth_kerb_module modules/mod_auth_kerb.so
```

```
<Location />
```

```
AuthName "RiverBarKerberos"
```

```
AuthType Kerberos
```

```
KrbMethodNegotiate On
```

```
KrbMethodK5Passwd Off
```

```
KrbLocalUserMapping On
```

```
KrbServiceName HTTP/riverbar.rootcloud.com
```

```
KrbAuthRealms ROOTCLOUD.COM
```

```
Krb5KeyTab /etc/http.keytab
```

```
Require valid-user
```

```
Order Deny,Allow
```

```
Deny from all
```

```
Satisfy any
```

```
</Location>
```

```
# /etc/httpd/conf.d/remote_user.conf
```

```
LoadModule auth_kerb_module modules/mod_auth_kerb.so
```

```
<Location />
```

```
AuthName "RiverBarKerberos"
```

```
AuthType Kerberos
```

```
KrbMethodNegotiate On
```

```
KrbMethodK5Passwd Off
```

```
KrbLocalUserMapping On
```

```
KrbServiceName HTTP/riverbar.rootcloud.com
```

```
KrbAuthRealms ROOTCLOUD.COM
```

```
Krb5KeyTab /etc/http.keytab
```

```
Require valid-user
```

```
Order Deny,Allow
```

```
Deny from all
```

```
Satisfy any
```

```
</Location>
```



```
# /etc/httpd/conf.d/remote_user.conf
```

```
LoadModule auth_kerb_module modules/mod_auth_kerb.so
```

```
<Location />
```



```
AuthName "RiverBarKerberos"
```

```
AuthType Kerberos
```

```
KrbMethodNegotiate On
```

```
KrbMethodK5Passwd Off
```

```
KrbLocalUserMapping On
```

```
KrbServiceName HTTP/riverbar.rootcloud.com
```

```
KrbAuthRealms ROOTCLOUD.COM
```

```
Krb5KeyTab /etc/http.keytab
```

```
Require valid-user
```

```
Order Deny,Allow
```

```
Deny from all
```

```
Satisfy any
```

```
</Location>
```

mod_auth_kerb

Enrolled host + service

chown apache

```
# /etc/httpd/conf.d/remote_user.conf
```

```
LoadModule auth_kerb_module modules/mod_auth_kerb.so
```

```
<Location />
```

```
AuthName "RiverBarKerberos"
```

```
AuthType Kerberos
```

```
KrbMethodNegotiate On
```

```
KrbMethodK5Passwd Off
```



```
KrbLocalUserMapping On
```

```
KrbServiceName HTTP/riverbar.rootcloud.com
```

```
KrbAuthRealms ROOTCLOUD.COM
```

```
Krb5KeyTab /etc/http.keytab
```

```
Require valid-user
```

```
Order Deny,Allow
```

```
Deny from all
```

```
Satisfy any
```

```
</Location>
```

```
# /etc/httpd/conf.d/remote_user.conf
```

```
LoadModule auth_kerb_module modules/mod_auth_kerb.so
```

```
<Location />
```

```
AuthName "RiverBarKerberos"
```

```
AuthType Kerberos
```

```
KrbMethodNegotiate On
```

```
KrbMethodK5Passwd Off
```

```
KrbLocalUserMapping On
```



```
KrbServiceName HTTP/riverbar.rootcloud.com
```

```
KrbAuthRealms ROOTCLOUD.COM
```

```
Krb5KeyTab /etc/http.keytab
```

```
Require valid-user
```

```
Order Deny,Allow
```

```
Deny from all
```

```
Satisfy any
```

```
</Location>
```



```
# /etc/httpd/conf.d/remote_user.conf
```

```
LoadModule auth_kerb_module modules/mod_auth_kerb.so
```

```
<Location />
```

```
AuthName "RiverBarKerberos"
```

```
AuthType Kerberos
```

```
KrbMethodNegotiate On
```

```
KrbMethodK5Passwd Off
```

```
KrbLocalUserMapping On
```

```
KrbServiceName HTTP/riverbar.rootcloud.com
```

```
KrbAuthRealms ROOTCLOUD.COM
```

```
Krb5KeyTab /etc/http.keytab
```

```
Require valid-user
```

```
Order Deny,Allow
```

```
Deny from all
```

```
Satisfy any
```

```
</Location>
```



does it **negotiate**?

cURL
requests.py
browsers

```
$ curl --negotiate  
-u : $FQDN
```

```
[vagrant@client]# kinit roguelynn
```

```
Password for roguelynn@ROOTCLOUD.COM:
```

```
[vagrant@client]# klist
```

```
Ticket cache: DIR:./run/user/1000/krb5cc/tktkPmrOq
```

```
Default principal: roguelynn@ROOTCLOUD.COM
```

Valid starting	Expires
06/16/2013 02:02:21	06/17/2013 02:02:18

```
Service principal  
krbtgt/ROOTCLOUD.COM@ROOTCLOUD.COM
```

```
[vagrant@client]# kinit roguelynn  
Password for roguelynn@ROOTCLOUD.COM:  
[vagrant@client]# klist
```



```
Ticket cache: DIR::/run/user/1000/krb5cc/tktkPmrOq  
Default principal: roguelynn@ROOTCLOUD.COM
```

```
Valid starting          Expires  
06/16/2013 02:02:21    06/17/2013 02:02:18
```

```
Service principal  
krbtgt/ROOTCLOUD.COM@ROOTCLOUD.COM
```

```
[vagrant@client]# kinit roguelynn
```

```
Password for roguelynn@ROOTCLOUD.COM:
```

```
[vagrant@client]# klist
```



```
Ticket cache: DIR::/run/user/1000/krb5cc/tkTkPmrOq
```

```
Default principal: roguelynn@ROOTCLOUD.COM
```

Valid starting	Expires
06/16/2013 02:02:21	06/17/2013 02:02:18

```
Service principal  
krbtgt/ROOTCLOUD.COM@ROOTCLOUD.COM
```

```
[vagrant@client]# kinit roguelynn
```

```
Password for roguelynn@ROOTCLOUD.COM:
```

```
[vagrant@client]# klist
```

```
Ticket cache: DIR: :/run/user/1000/krb5cc/tkTkPmrOq
```

```
Default principal: roguelynn@ROOTCLOUD.COM
```

```
Valid starting
```

```
Expires
```

```
06/16/2013 02:02:21
```

```
06/17/2013 02:02:18
```

```
Service principal
```

```
krbtgt/ROOTCLOUD.COM@ROOTCLOUD.COM
```



```
[vagrant@client]# kinit roguelynn
Password for roguelynn@ROOTCLOUD.COM:
[vagrant@client]# curl -I --negotiate -u : \
    https://synergizeapp.strategery.com
```

```
HTTP/1.1 401 Unauthorized
Date: Wed, 15 May 2013 09:10:18 GMT
Server: Apache/2.4.4 (Fedora)
WWW-Authenticate: Negotiate
Content-type text/html; charset=iso-8859-1
```

```
HTTP/1.1 200
Date: Wed, 15 May 2013 09:10:18 GMT
Server: Apache/2.4.4 (Fedora)
WWW-Authenticate: Negotiate sOmE_RanDom_T0k3n
```

```
[vagrant@client]# kinit roguelynn
Password for roguelynn@ROOTCLOUD.COM:
[vagrant@client]# curl -I --negotiate -u :
  https://synergizeapp.strategery.com
```





```
HTTP/1.1 401 Unauthorized
Date: Wed, 15 May 2013 09:10:18 GMT
Server: Apache/2.4.4 (Fedora)
WWW-Authenticate: Negotiate
Content-type text/html; charset=iso-8859-1
```

```
HTTP/1.1 200
Date: Wed, 15 May 2013 09:10:18 GMT
Server: Apache/2.4.4 (Fedora)
WWW-Authenticate: Negotiate sOmE_RanDom_T0k3n
```

ticket cache

```
[vagrant@client]# kinit roguelynn
Password for roguelynn@ROOTCLOUD.COM:
[vagrant@client]# curl -I --negotiate -u : \
    https://synergizeapp.strategery.com
```

```
HTTP/1.1 401 Unauthorized 
Date: Wed, 15 May 2013 09:10:18 GMT
Server: Apache/2.4.4 (Fedora)
WWW-Authenticate: Negotiate
Content-type text/html; charset=iso-8859-1
```

```
HTTP/1.1 200 
Date: Wed, 15 May 2013 09:10:18 GMT
Server: Apache/2.4.4 (Fedora)
WWW-Authenticate: Negotiate sOmE_RanDom_T0k3n
```

two responses

```
[vagrant@client]# klist
```

```
Ticket cache: DIR:./run/user/1000/krb5cc/tktkPmrOq  
Default principal: roguelynn@ROOTCLOUD.COM
```

```
Valid starting          Expires  
06/16/2013 02:02:21    06/17/2013 02:02:18  
06/16/2013 02:02:21    06/17/2013 02:02:18
```

```
Service principal  
krbtgt/ROOTCLOUD.COM@ROOTCLOUD.COM
```

```
HTTP/riverbar.rootcloud.com@ROOTCLOUD.COM
```



ticket for HTTP service

cURL
requests.py
browsers

authentication
VS
authorization

authentication
VS
authorization

accessing permissions

accessing permissions

- Connecting to LDAP
- Calling system commands
- PAM stack + Apache modules:
 - mod_auth_kerb
 - mod_authnz_external

LDAP

is **user** a **member** of
“admins”, or
“team players”, or
“movers and shakers”, ...

getxx system calls: SSSD

- e.g. `getent group $USER`
- conveniently accesses group information stored in IPA LDAP
- Python wrapper: `getent` package

Apache modules

- `mod_auth_kerb` for authentication
- `mod_authnz_external` + `pwauth` to defer to PAM for group lookup
- `pwauth` will need a bit of a hack to accept Kerb auth

your own kerberos
test environment:
rogue.ly/kerberos

word of caution:

**do NOT install IPA
on your local machine!**

In review

- custom user model with SSO
- integrate Django apps with SSO
- access permissions for your app

Crap.

Now I'm the point person
for this.



@roguelynn

rogue.ly/kerberos



Background image:

<http://www.animalhi.com/Mammals/elephants/>

[abstract_flying_elephants_circus_simplistic_simple_1920x1080_wallpaper_29579](http://www.animalhi.com/Mammals/elephants/abstract_flying_elephants_circus_simplistic_simple_1920x1080_wallpaper_29579)