OpenID and Single Sign On

Sergey Kolosov
Wargaming.net

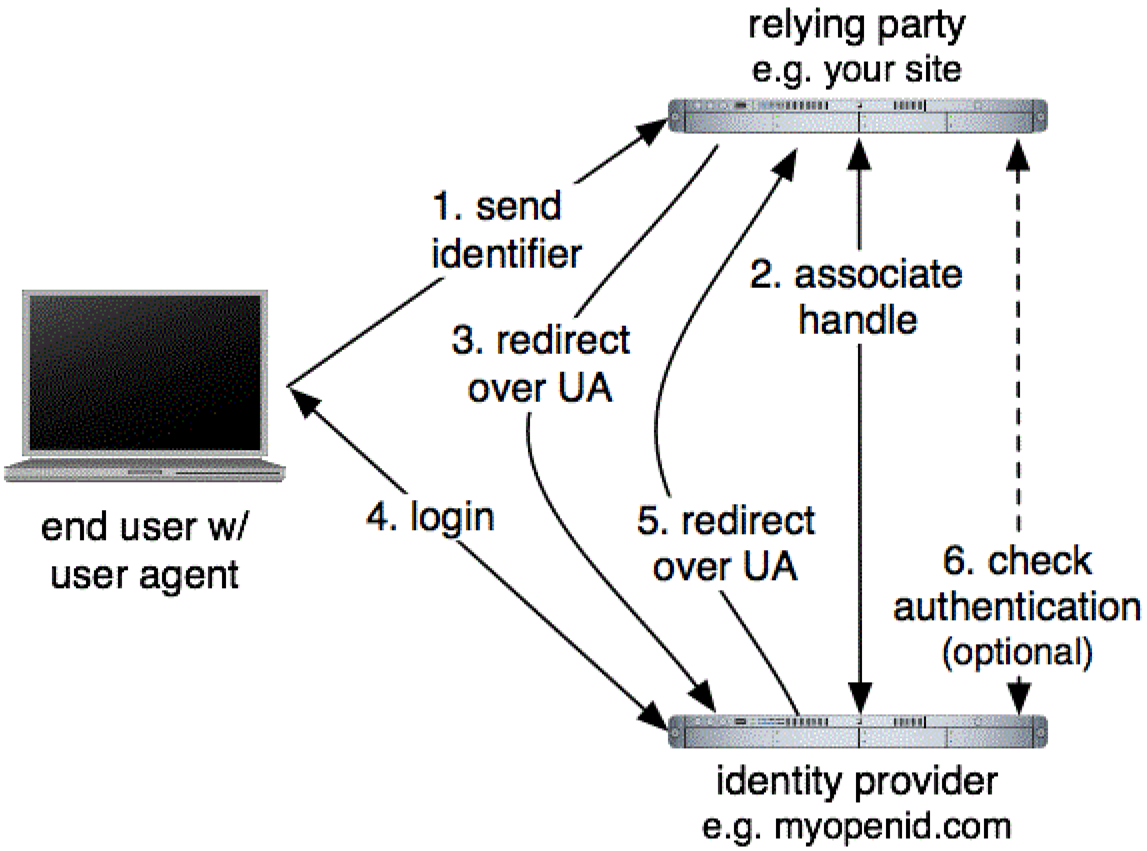
OpenID: facts

- An open standard
- An authentication protocol
- Single identity – multiple sites

OpenID: roles

- End-user (or a User-Agent)
- OpenID Provider (OP)
- Relying Party (RP)

OpenID: flow



courtesy of flickr.com/people/keepthebyte (CC BY-NC-SA 2.0)

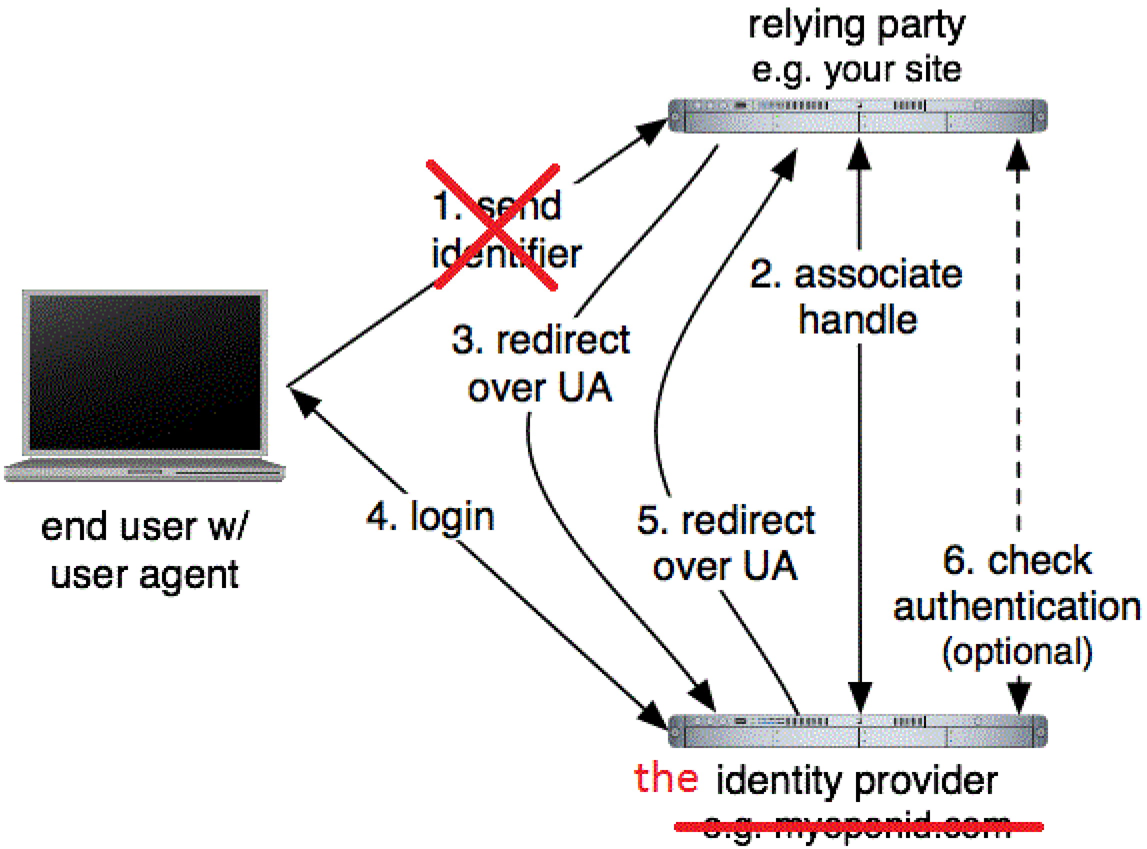
OpenID: extensions

- OpenID specification doesn’t cover any kind of information exchange
- OpenID 2.0 introduced extensions
- Official extension specifications:
  - Attribute Exchange (AX)
  - Simple Registration Extension (SREG)
  - Provider Authentication Policy Extension (PAPE)

Make it bend to your will

EPISODE1:
OPENID FOR A CLOSED ECOSYSTEM

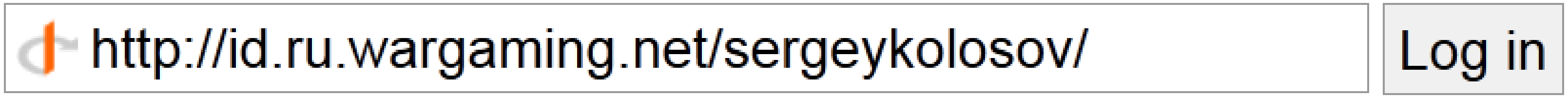
OpenID: flow*



courtesy of flickr.com/people/keepthebyte (CC BY-NC-SA 2.0)

OpenID: flow differences — RP

Log in to an **external** resource



http://id.ru.wargaming.net/sergeykolosov/

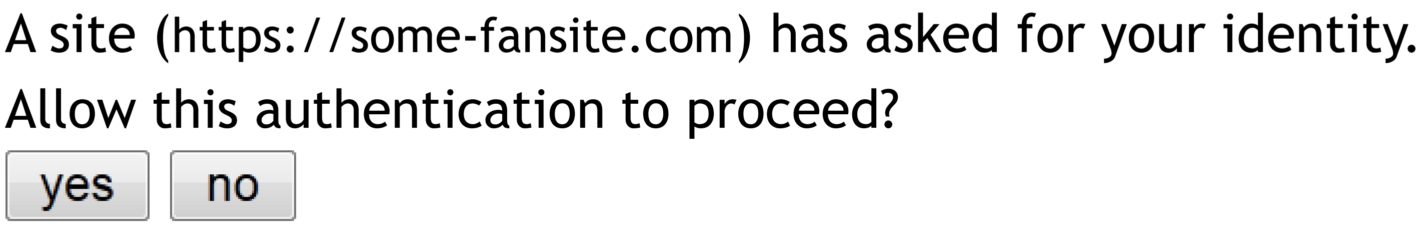
Log in to an **internal** resource *

Just

```
* settings.OPENID_SERVER_URL = "http://id.ru.wargaming.net/"
```

OpenID: flow differences — OP

Log in to an **external** resource



A site (https://some-fansite.com) has asked for your identity.

Allow this authentication to proceed?

Log in to an **internal** resource *

(just redirect)

```
* settings.OPENID_TRUSTED_ROOTS = ("http://ru.wargaming.net/",)
```

That simple?
Not really.*

* spice it up with a yummy bit of high-load

No HTTP-requests allowed inside web-server workers*

* background workers to the rescue!

OpenID: a flavour of high-load

- Make all requests from RP to OP inside the background workers
- Be sure OpenID implementations were not expected to operate this way
- Be ready to have a hard time implementing client-side code to handle this
  - good old “Same origin policy” and friends

Me want COOKIE!

EPISODE2: ADDING SINGLE SIGN ON

OpenID indeed
is a single sign-on system*

* except it is quite not enough

OpenID as a web-based SSO

- It works
- Still, a user have to tolerate one of this:
  - A click on a “Login” button
  - Waiting for AJAX requests to complete (≈ for UI to be ready to use)

Seamless user experience
is what you want*

* and what you can succeed in, while you're inside your own ecosystem

What if we try authenticating to all services beforehand?*

* brilliant idea!

OpenID as an SSO: seamless UX

- Authenticate everywhere on login at OP
- Hide it: IFRAMEs, pixels, etc.
- Cookies are you best friends:
  - share cookie between domains (wisely)
  - number of requests is proportional to a number of second-level domains

Wrapping it up

OpenID gives you
a basis for a web SSO.

If you control both client and server,
you can make it awesome.

Questions?

- Twitter: @m17russia
- E-mail: m17.admin@gmail.com