SOCIAL LOGIN WITH SPRING-SOCIAL

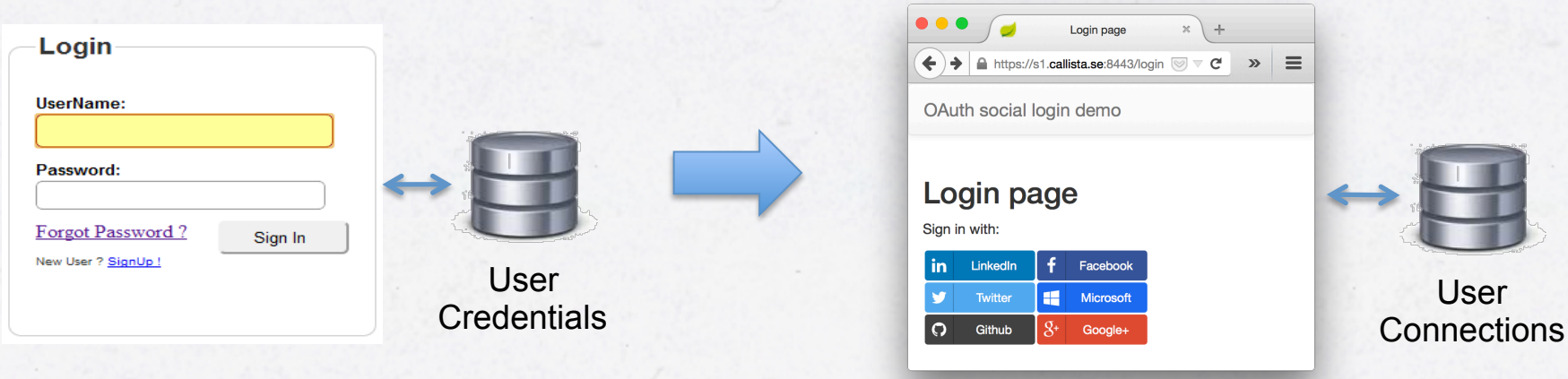
MAGNUS LARSSON

2015.02.04 | CALLISTAENTERPRISE.SE

CALLISTA
— ENTERPRISE —

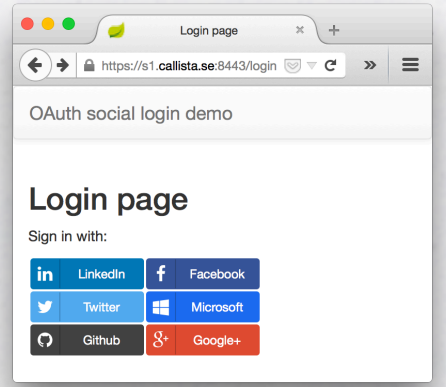
WHAT IS SOCIAL LOGIN?

- Replace self registration of user accounts
  with authentication based on social networks accounts
  + **Users:** No registration process or new password to remember
  + **Developer:** Delegate user registration and authentication
    to trusted social networks

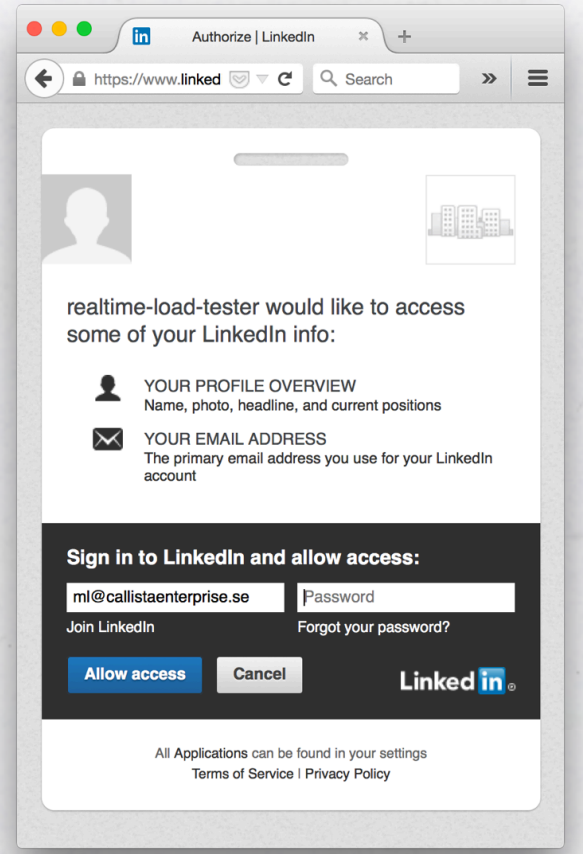


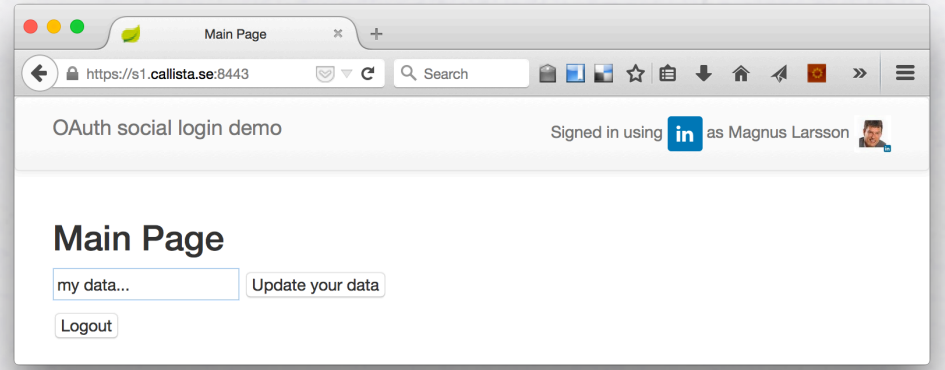
CALLISTA
— ENTERPRISE —

WHAT IS SOCIAL LOGIN?



OAuth





CALLISTA
— ENTERPRISE —

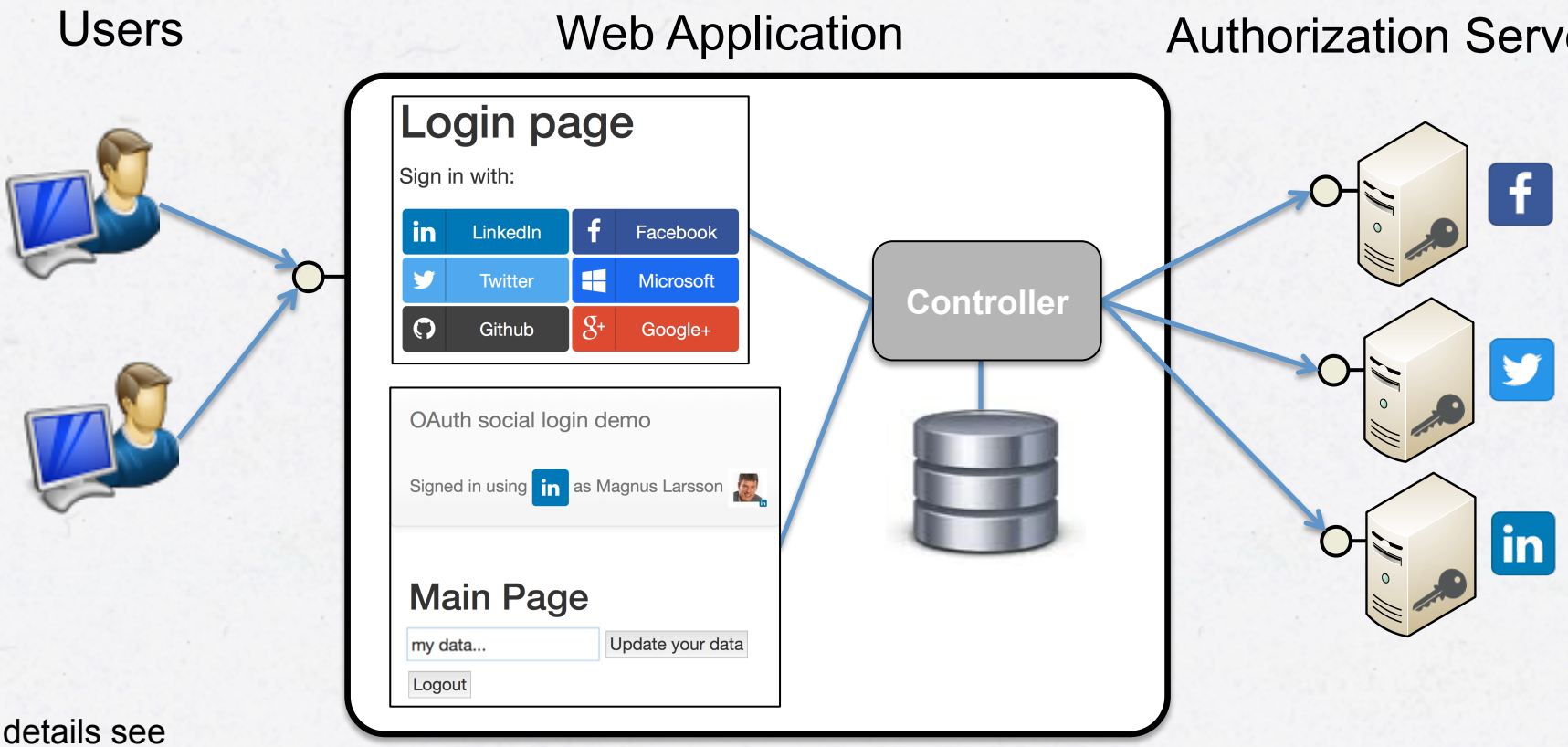
WHAT IS SPRING SOCIAL?

- A spring.io project
- spring-social
  - *"Connect your Spring application with Software-as-a-Service (SaaS) API providers such as Facebook, Twitter, and LinkedIn."*
  - Simplifies
    - » Using OAuth, v1.0 and v2.0
    - » Connecting local user accounts to OAuth provider accounts
      - ▸ Automatic persistence of connections
    - » Setting up a secure environment using **`spring-security`**
    - » Defining Java bindings for OAuth providers APIs

WHAT IS SPRING SOCIAL?

- spring-social, cont…
  - Extensible
    - » Supports Twitter, Facebook, LinkedIn and GitHub in the core
    - » ≈ 30 community projects
      - ▸ Not only for social providers, e.g. Salesforce

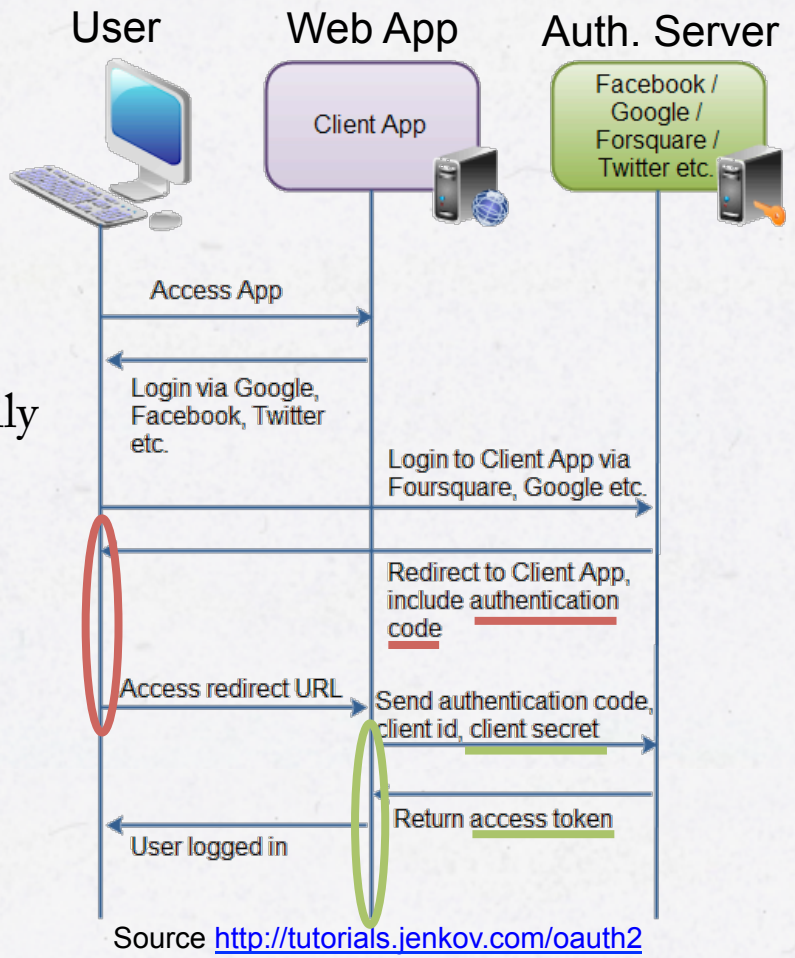
SAMPLE APPLICATION



For details see
http://callistaenterprise.se/blogg/teknik/2014/09/02/adding-social-login-on-a-website-using-spring-social/

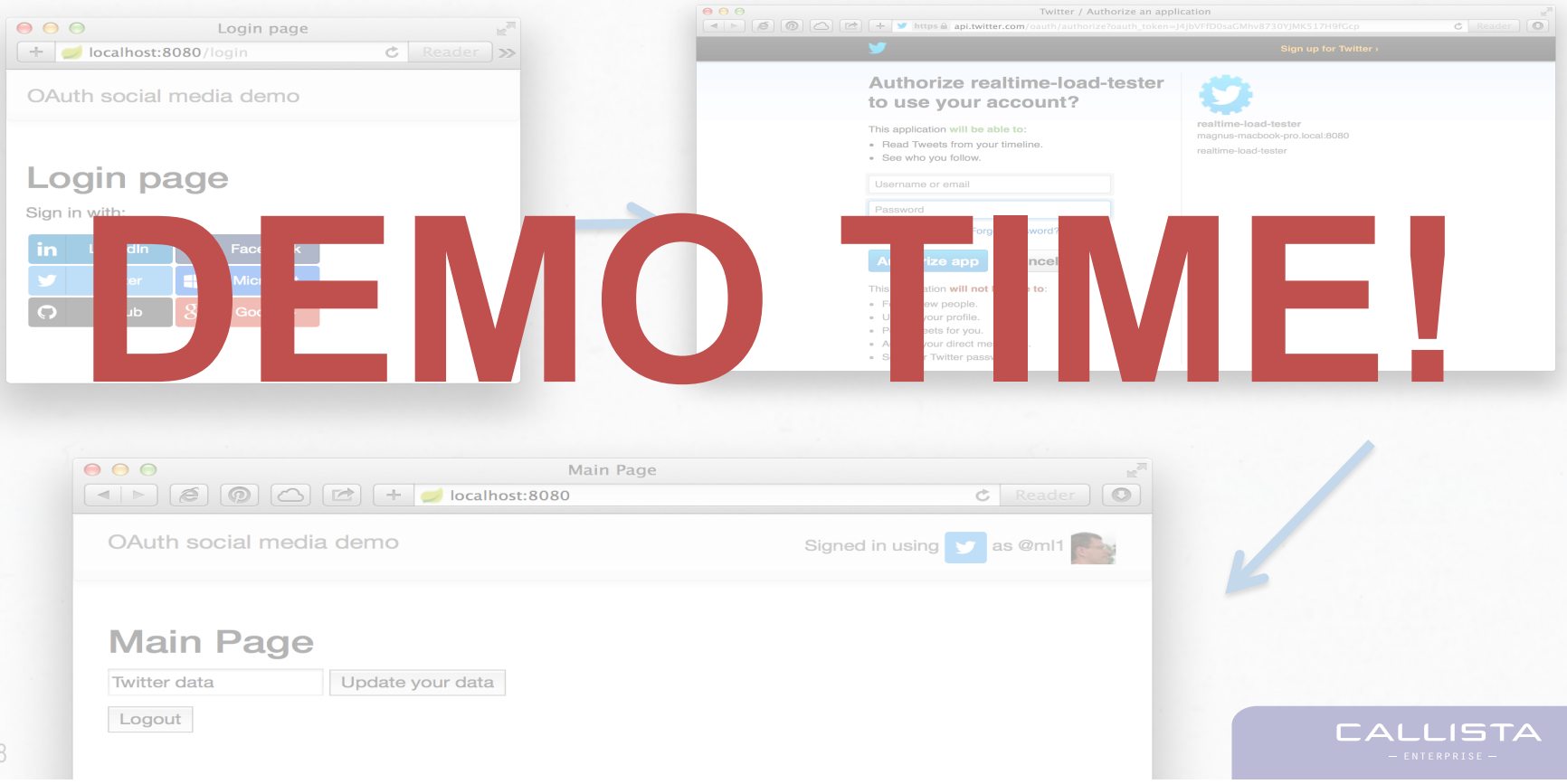
SAMPLE APPLICATION

- Uses
  - *Implicit Login*
    - » Local user account created automatically
  - *Authorization Code Grant Flow*
    - » Only *authentication code* visible in web browser (≈ a one-time password)
    - » *Access token* kept safe in the web app



Source http://tutorials.jenkov.com/oauth2

DEMO



CALLISTA
— ENTERPRISE —

SECURITY - OAUTH

- Use *Authorization Code Grant Flow* in secure environments
  - I.e. where a client secret and access/refresh token can be kept secret
  - E.g. Web Apps
- Use *Implicit flow* in non-secure environments
  - E.g. in Javascript applications or mobile solutions without a Web App
  - Not as secure as *Authorization Code Grant Flow…*
- Hold on to your *client secret*, *access tokens* and *refresh tokens* (if any)
- Keep the life short for *access tokens*

SECURITY - WEB

- The OWASP Top 10 security threats!
  - HTTPS
    - » This is the fundament, ensure correct usage!
    - » Use TLS, correct libraries, cipher suites…
  - Injection
    - » Always use parameterized API's
  - XSS
    - » Escape untrusted data
    - » Use HttpOnly Cookies
  - CSRF
    - » Set a unique token in a hidden field per request

SECURITY - WEB

- Address the OWASP Top 10 security threats, cont…
  - SpoofedMe attacks
    - » Don't use email address as an account identifier
  - Insecure Direct Object References
    - » Use indirect UUID alike references

- Hard to handle?
  - Use web framework that makes the heavy-lifting
  - E.g. use **`spring-social`** together with **`spring-security`** and **`spring-mvc`**

SUMMARY

- Use social login to
  - Relieve your users from registration and new passwords
  - Delegate user registration and authentication to OAuth providers you trust
- Use **`spring-social`** to simplify the work
  - Use **`spring-security`** to handle many of the security aspects
- For details see http://callistaenterprise.se/blogg/teknik/2014/09/02/adding-social-login-on-a-website-using-spring-social/
- New blogs and presentations are published on Twitter (@callistaent) and RSS (http://callistaenterprise.se/feed.xml)

QUESTIONS?

E-mail: magnus.larsson@callistaenterprise.se