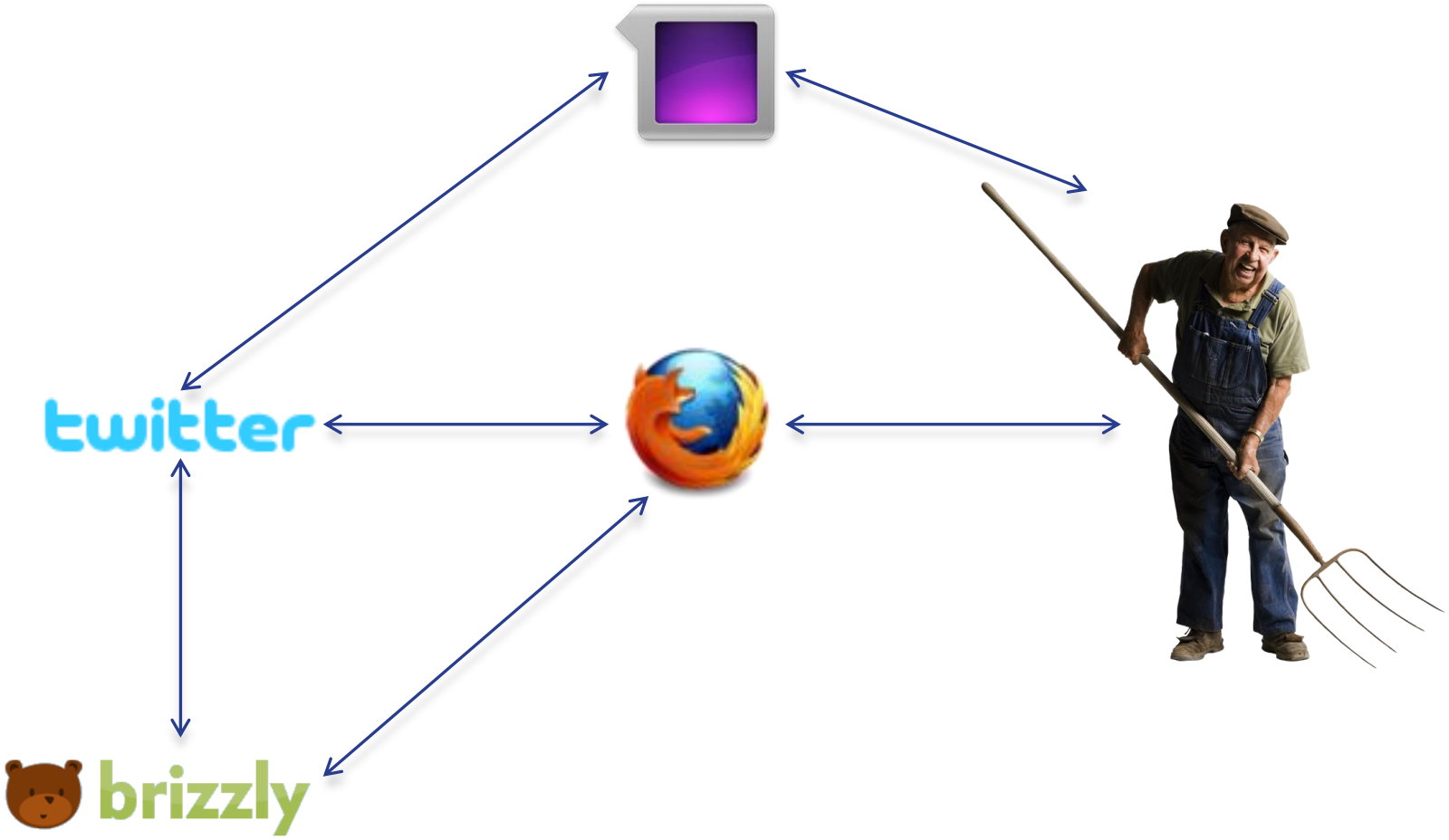
*OAuth

Martin Matula (martin.matula@oracle.com)

28.2.2011, Setkání pražského CZJUG

Agenda

- Úvod – webové služby a autentizace
- Co je OAuth
- Ukázka OAuth klienta a serveru
- OAuth.next stručně
- Nabídka práce, Q&A, občerstvení


twitter
brizzly

* Mnoho hojně používaných webových aplikací zveřejňuje API pro vývojáře
  * Twitter, Facebook, Google Apps, atd.
* V drtivé většině populárních webů převažuje jednoduché HTTP API (v ideálním případě RESTful) nad SOAP WS

Webové služby a REST

Klasická autentizace

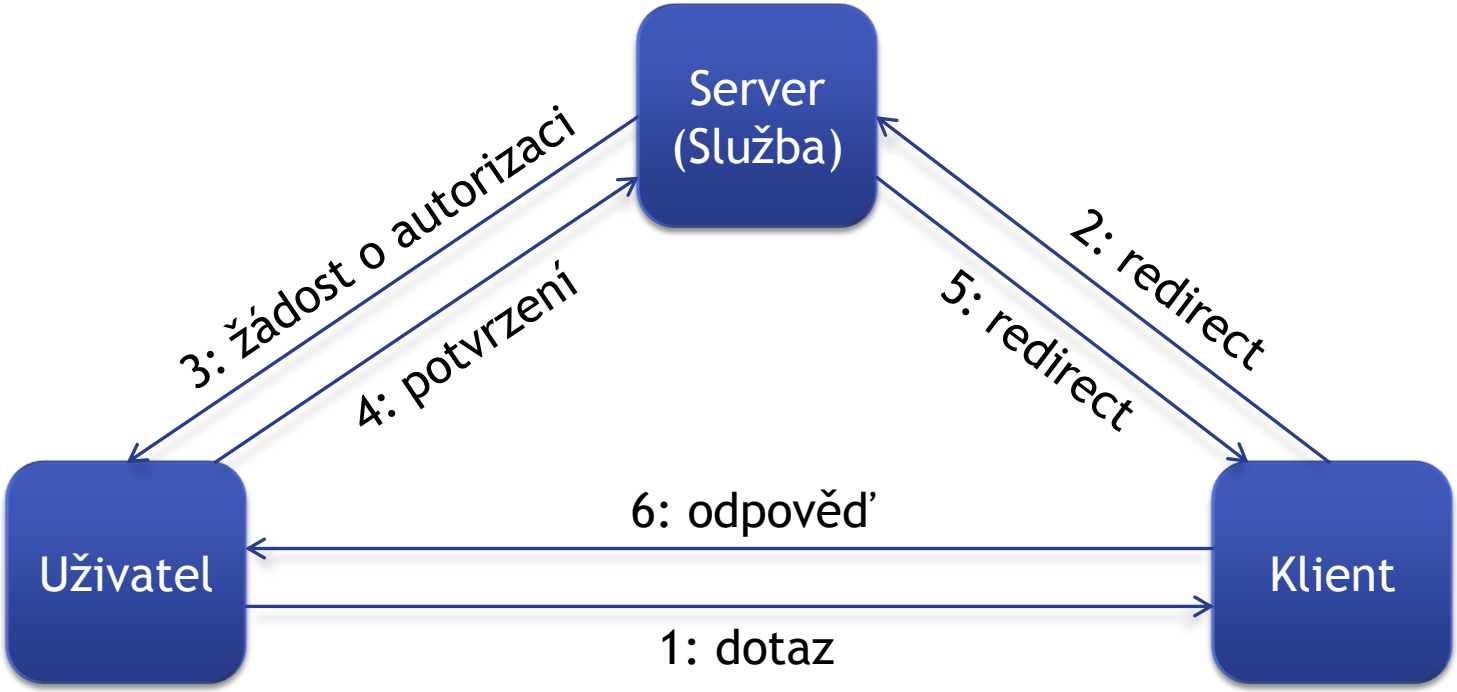
- Autentizuje se koncový uživatel svými credentials
  - Jméno a heslo
    (HTTP Basic Auth., HTTP Digest Auth.)
  - Certifikát
- Vyžaduje vysokou míru důvěry koncového uživatele vůči klientské aplikaci
- Funguje dobře pro webové aplikace – uživatelé důvěřují webovým prohlížečům

Autentizace a webové služby

- Sdílení uživatelských credentials s klientskou aplikací je nežádoucí
  - Poskytuje plnou autoritu klientské aplikaci
  - Nelze spolehlivě odebrat přístup pro daného klienta
- Vznikla řada řešení od různých firem
  - Amazon Signature, Google AuthSub, Yahoo BBAuth, Microsoft Shared Key, …
- OAuth se stává v této oblasti standardem

OAuth 1.0

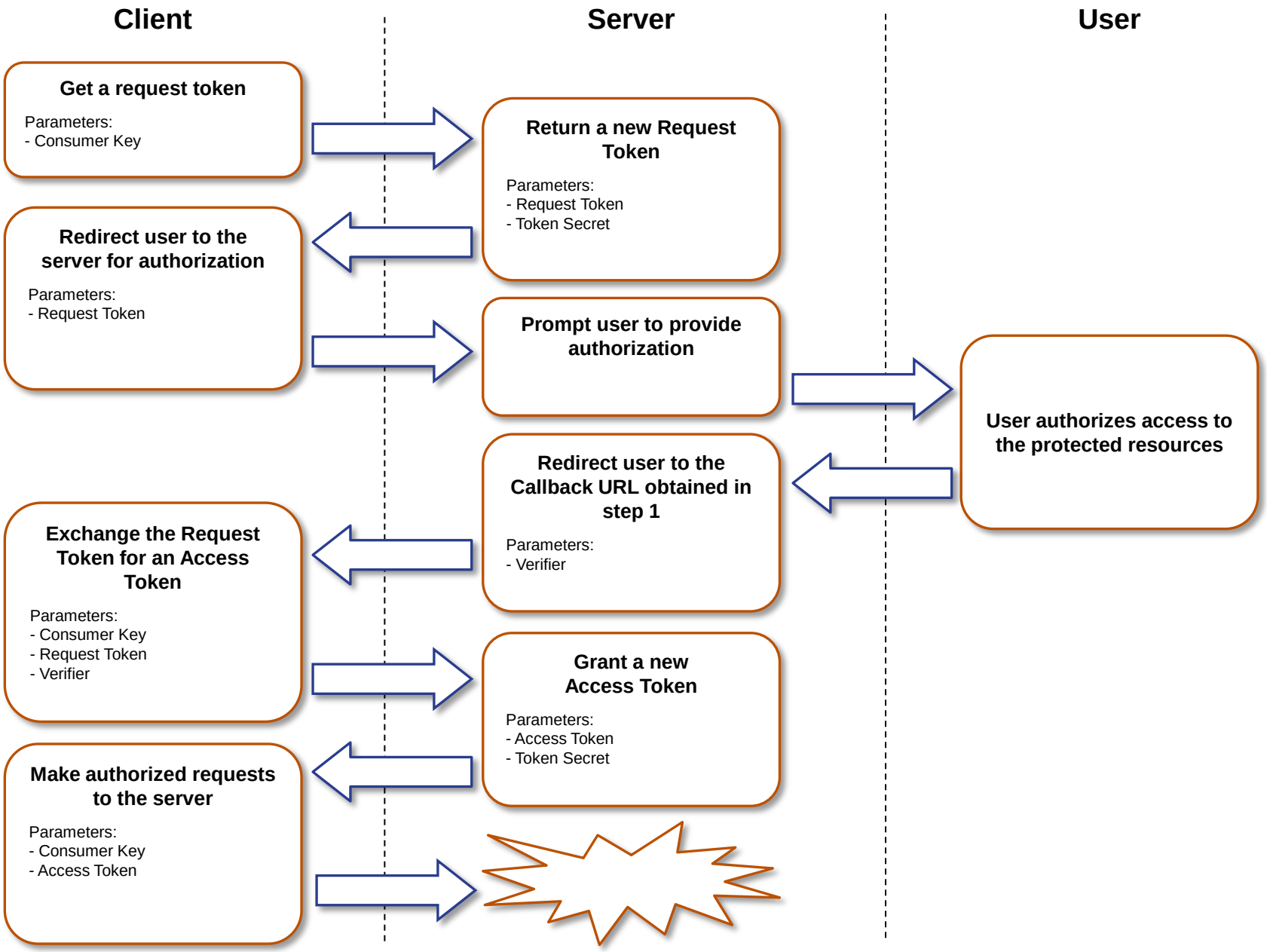
- An **open protocol** to allow **secure API authorization** in a **simple** and **standard** method from desktop and web applications
- Komunitní verze – OAuth Core 1.0 a 1.0a
- Později nahrazeny RFC 5849 – IETF OAuth 1.0
- Umožňuje aplikacím přístup k chráněným prostředkům ve jménu uživatele bez potřeby znalosti uživatelských credentials



*Autorizační “flow”

* Používá klientské credentials pro autentizaci klientské aplikace
  * Consumer Key, Consumer Secret
* Přístupový token pro ověření autorizace
  * Access Token, Token Secret
* “Secret” část se neposílá po drátě – používá se pouze k podpisu zpráv
  * Nevyžaduje použití SSL k ochraně credentials

OAuth 1.0 v detailu


Client
Server
User
Get a request token
Parameters:
- Consumer Key
Return a new Request Token
Parameters:
- Request Token
- Token Secret
Redirect user to the server for authorization
Parameters:
- Request Token
Prompt user to provide authorization
User authorizes access to the protected resources
Redirect user to the Callback URL obtained in step 1
Parameters:
- Verifier
Exchange the Request Token for an Access Token
Parameters:
- Consumer Key
- Request Token
- Verifier
Grant a new Access Token
Parameters:
- Access Token
- Token Secret
Make authorized requests to the server
Parameters:
- Consumer Key
- Access Token

OAuth a Jersey

- Jersey (http://jersey.java.net)
  - Framework pro vývoj RESTových služeb v Javě
- Implementace standardu JAX-RS
- Řada dalších “value-add” funkcí
  - Klientské API, podpora WADL, MVC, atd.
- Podpora OAuth 1.0 na klientské i serverové straně

Jersey OAuth klient

- Implementován jako klientský filtr
- Automaticky podepisuje odchozí zprávy dle nastavených klientských credentials a přístupového tokenu
- Výrazně zjednodušuje implementaci autorizačního flow
- Maven modul oauth-client
  - GroupId: com.sun.jersey.contribs.jersey-oauth

Jersey OAuth server

- Implementuje filtr na straně serveru, který verifikuje OAuth hlavičky v příchozích zprávách
- Vydávání request a access tokenů
- Správa tokenů a klientů je specifická pro každou aplikaci – zasouvá se implementací SPI
  - Modul obsahuje default implementaci pro testovací účely
- Maven modul: oauth-server
  - GroupId: com.sun.jersey.contribs.jersey-oauth

*Rozšířenost OAuth 1.0

* OAuth 1.0 má své problémy
  * Správná implementace podpisů je netriviální
  * Úzké propojení mezi primární a autorizační funkcionalitou služby
  * Zbytečné zatížení na straně serveru
  * Obtížnější implementace newebových klientů
* OAuth WRAP
* OAuth 2.0

* Budoucnost OAuth

OAuth WRAP stručně

- WRAP = Web Request Authentication Protocol
- Google, Yahoo a Microsoft
- Pokus o řešení problémů s OAuth 1.0
- Přináší podporu více profilů/use-casů
  - Desktopové aplikace, JavaScript klienti
  - Web profile flow stejné jako u OAuth 1.0
- Místo podpisů používá SSL
- Vstup pro OAuth 2.0

OAuth 2.0 stručně

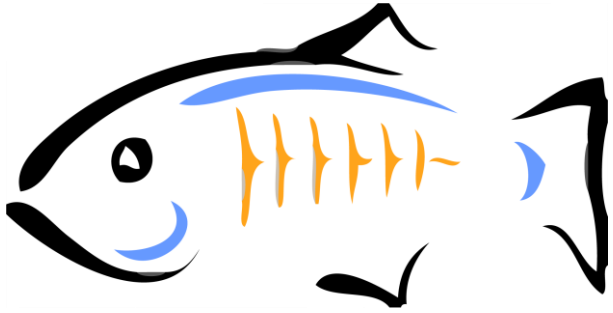
- Momentálně ve vývoji jako standard v rámci OAuth WG v IETF (nyní draft #13)
- Nekompatibilní s OAuth 1.0
- Jednoduchý
  - Odbourána nutnost implementace podpisů,
  - Zjednodušené autorizační flow pro různé typy klientů
- Lepši výkon a škálovatelnost
  - Zjednodušený state-management během autorizace
  - Oddělení autorizačních a resource serverů
- Formalizovaná rozšířitelnost
  - Registr typů tokenů, registr OAuth parametrů, …

* OAuth 1.0 – Otevřený standard
* Podpora množství firem
* Funguje na základě klientských credentials, přístupových tokenů a podpisů
  * Uživatel nemusí odkrývat své credentials
  * Přístup lze kterémukoliv klientovi odejmout
* Nevyžaduje SSL
* V Javě se implementace výrazně zjednodušuje použitím OAuth podpory v Jersey
* Dalším vylepšením bude OAuth 2.0, který ale momentálně není hotový

Závěr

* Užitečné odkazy
  * OAuth – http://oauth.net
  * Jersey – http://jersey.java.net

* Pojďte k nám pracovat!
* CV posílejte na martin.matula@oracle.com



* Dotazy?