Browser Support for the Open Authorization (OAuth) Protocol

http://www.w3.org/2011/identity-ws/papers/idbrowser2011_submission_32.pdf

Hannes Tschofenig, Barry Leiba,
Blaine Cook, Rob Van Eijk

Agenda

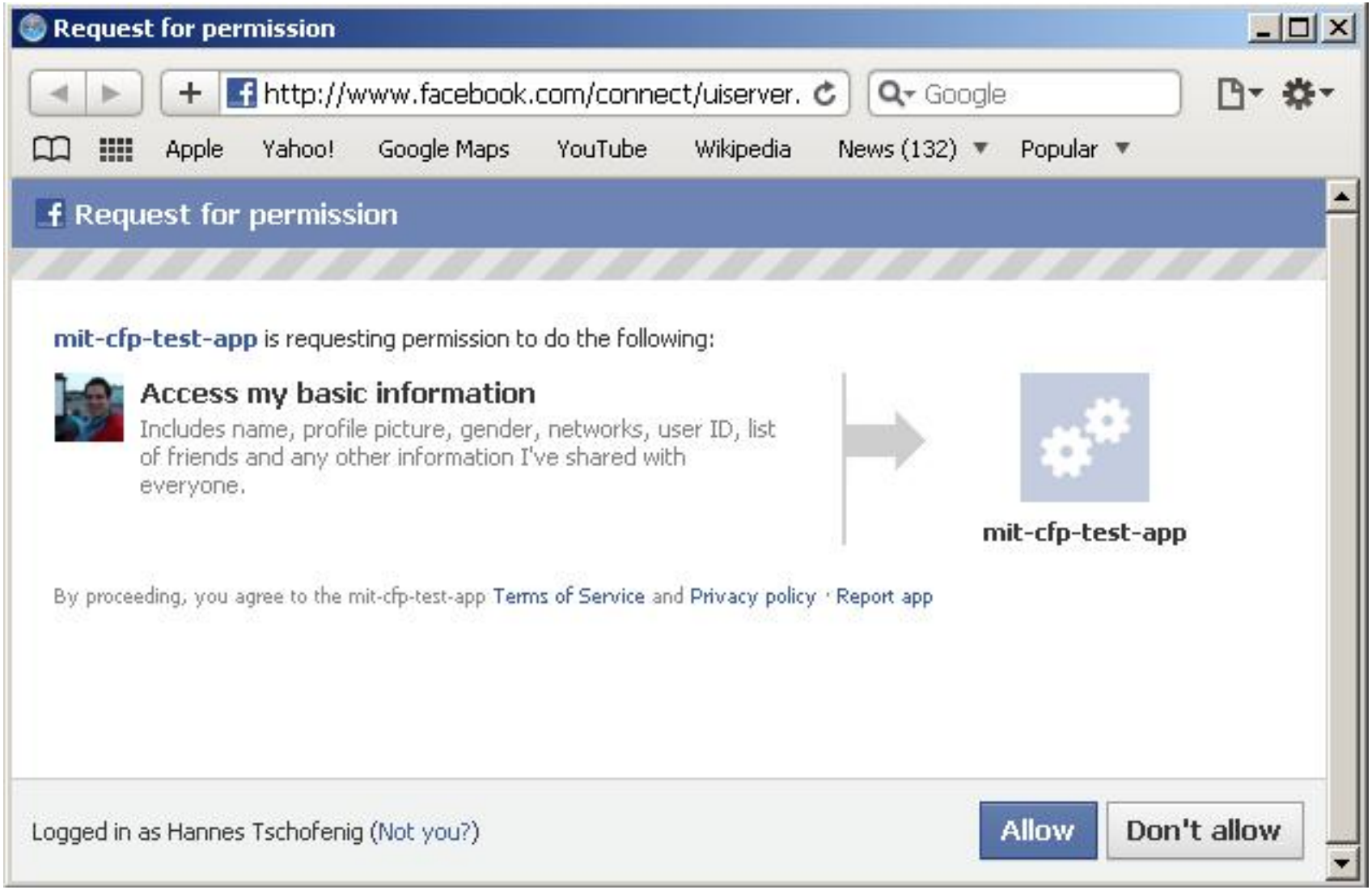
- Authentication Mechanisms
- JavaScript Crypto Library Support
- Authorization Interface
- Moving Crypto Into the Browser

Authentication Mechanisms

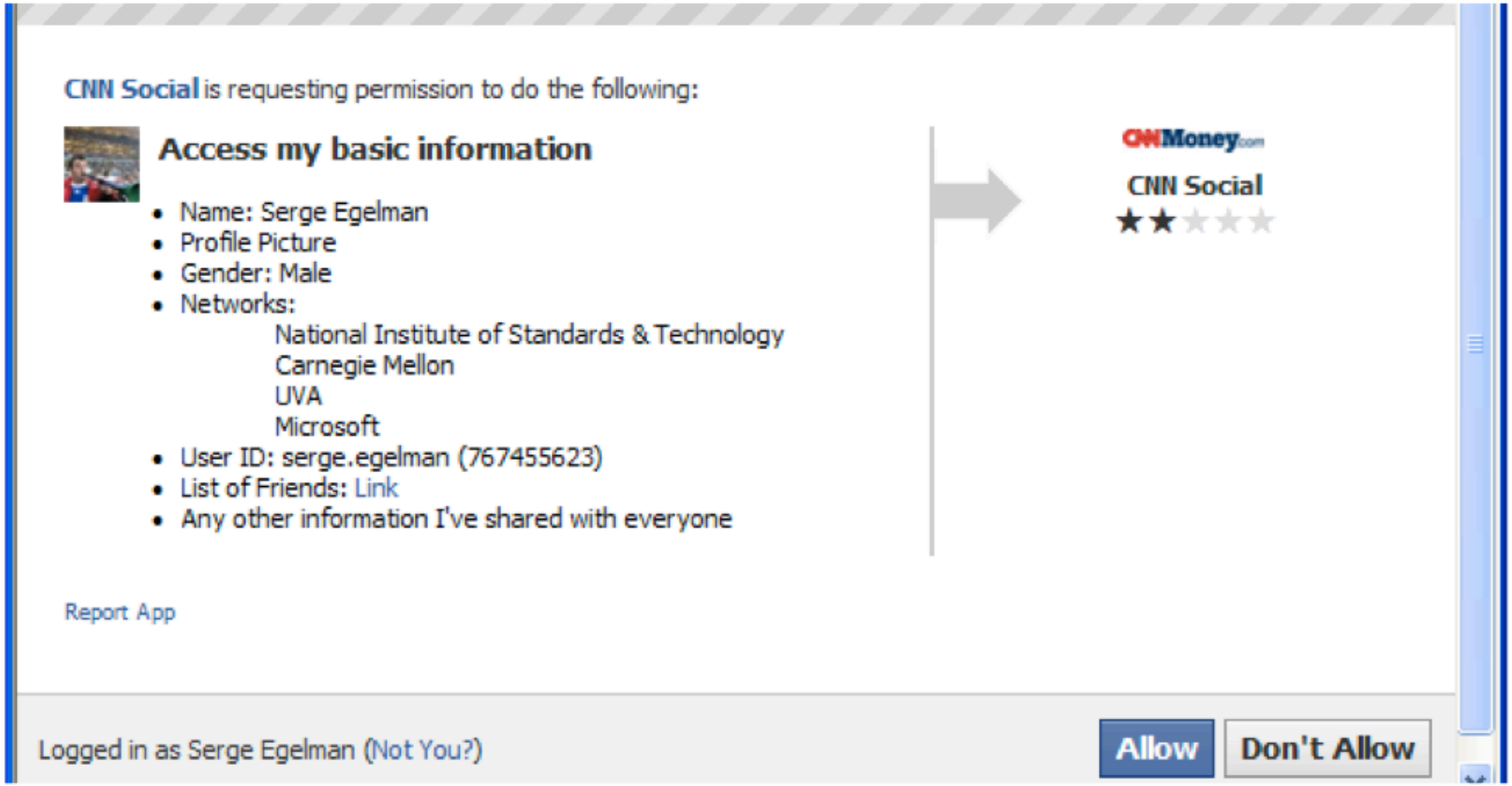
- Many identity management protocols treat the authentication exchange out of scope
  - So does OAuth
- Problems:
  - Authentication credentials being used are of poor quality (low entropy secrets)
  - Enrollment of strong credentials not implemented in browsers (e.g. enrollment of OTP mechanisms)
  - Authentication mechanisms being used on the Web today are pretty weak.
  - There is no authentication framework that allows for easily exchangeable authentication methods.
  - Other problem (but not related to lack of standardization): weak identity proofing
- Examples to look at: GSS-API, SASL, EAP, PSKC

What mechanisms should browser support?

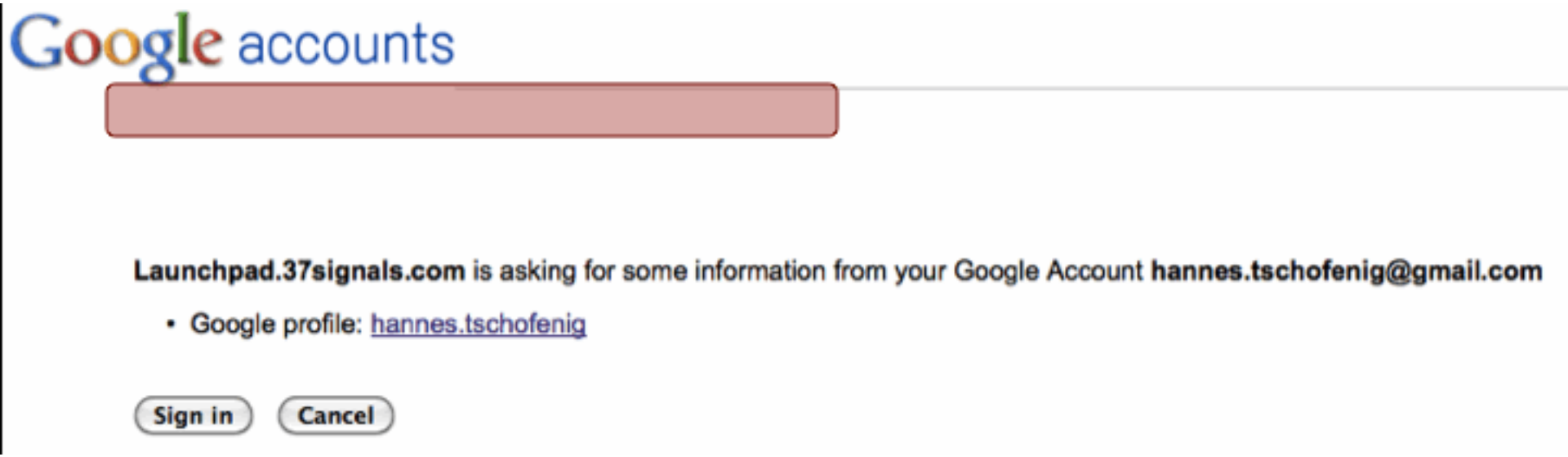
Authorization Interface



Authorization Interface, cont.



Authorization Interface, cont.



What guidance can we provide to developers for a consistent permission dialog experience?

What is the role of the browser in this exchange?

JavaScript Crypto Library Support

- JavaScript is an essential language for the Web eco-system. Increasingly popular also on the server-side.
- Problem: No standardized APIs for access to cryptographic functions and key storage.
- One consequence: Cryptographic material cannot be kept confidential with JavaScript-based clients.
- Example: https://developer.mozilla.org/en/JavaScript_crypto

Wouldn't be useful to standardize a JavaScript crypto API?

Moving Crypto Into the Browser

- OAuth protocol design is impacted by the capabilities offered by browsers.
- The security considerations capture this quite well:
  http://datatracker.ietf.org/doc/draft-ietf-oauth-v2/
  http://tools.ietf.org/html/draft-lodderstedt-oauth-security
- For example, http://datatracker.ietf.org/doc/draft-ietf-oauth-v2-http-mac/ suggests a MAC based authentication mechanism that can also be used to implement a more secure cookie/HTTP state management mechanism.

What functionality can we put into the browser platform?