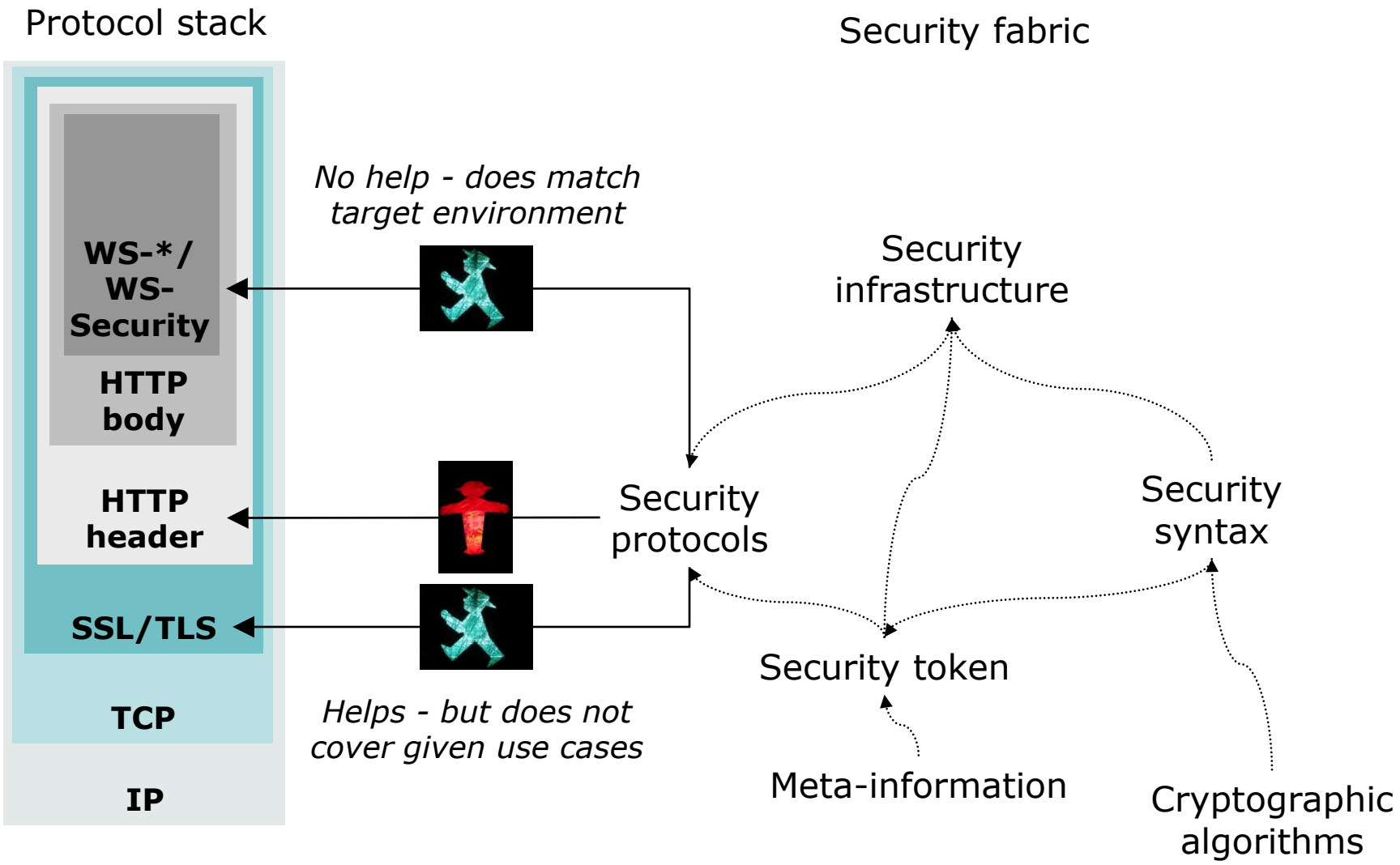
New Trends in Web Security

Munich - June 22, 2012

Oliver Pfaff

Security-Enabling the HTTP Stack - *Until 2010*



Driving Forces

- Constrained clients:
  - Smart phones/tablets accessing via mobile networks
  - Promote native clients: talk HTTP, serve single users – but are no classical Web browsers

- API economy:
  - Content aggregation via mash-ups/composite applications, Web APIs exposing lightweight interfaces, RESTful Web services – no WS-*
  - Promote application clients: talk HTTP, serve multiple users

- Cloud:
  - Procure IT from the network: applications (SaaS), software (PaaS) or hardware infrastructure (IaaS)
  - For many organizations "owning iron" is a snail's pace approach. Holds for the server side (XaaS) and the client side (BYOD)

- Disillusion:
  - Some things did not fly such as PKI to the end-user:
    - Not handy: people do not understand PKI – even IT pro's struggle
    - Compromises: Comodo/DigiNotar/StartTLS CAs, DuQu, Stuxnet
    - Ramifications of lemon markets (Nobel prize-awarded theory) apply

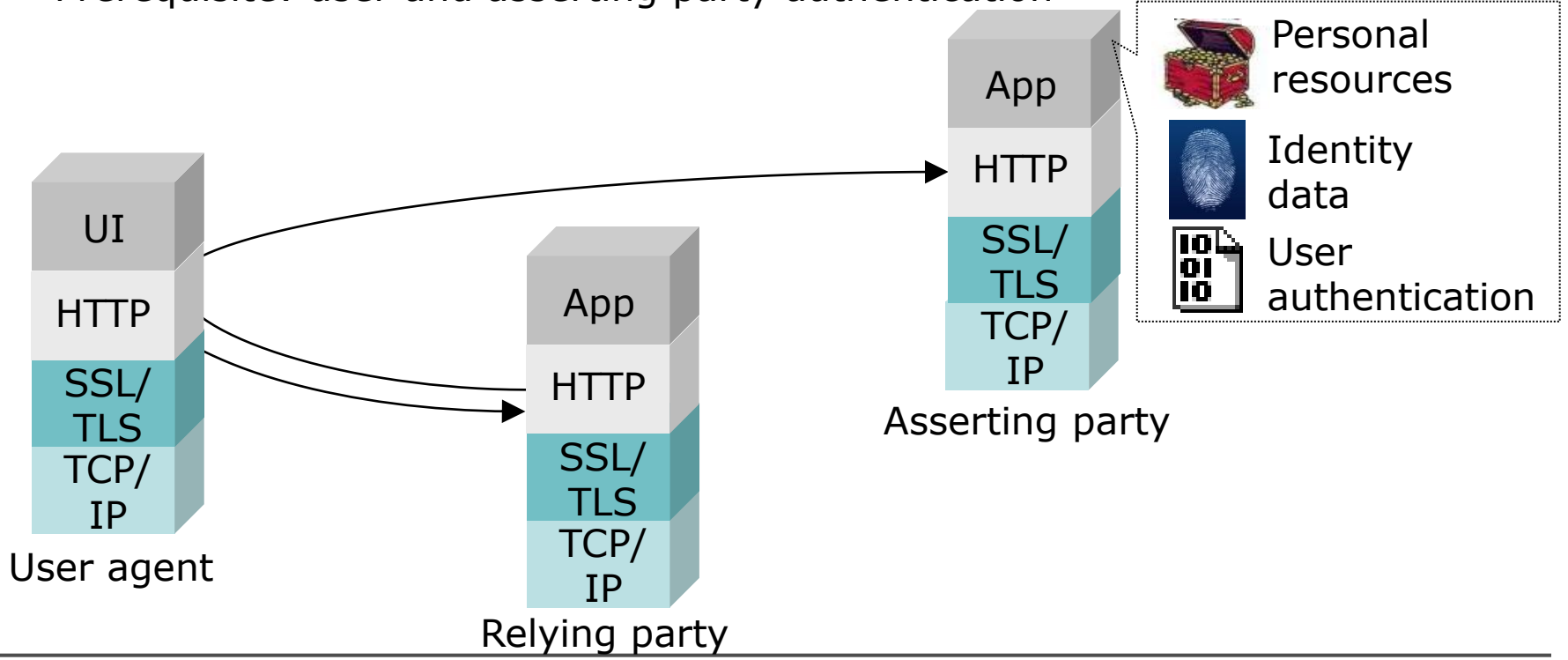
...Their Constraints/Needs/Use Cases

- Constrained clients:
  - Interactions: server-side, no client-side redirections
  - Compact representations: new formats for security objects, no WS-*
- API economy:
  - Authenticate API clients: new authentication schemes for HTTP
  - Manage access to personal resources: new authorization protocols
  - Move identity data (for self): on-boarding of individual users
- Cloud:
  - Externalize user authentication: provide seamless access (i.e. SSO)
  - Manage identity data (for any): user on-boarding in bulk-style
  - Manage authorization: govern access control for subscriber resources
- Disillusion:
  - Alternate entity authentication schemes: stronger than username/static password, less awkward than public key certificate and private key
  - Supply meta-information: express to relying parties how authentication and identity creation was done

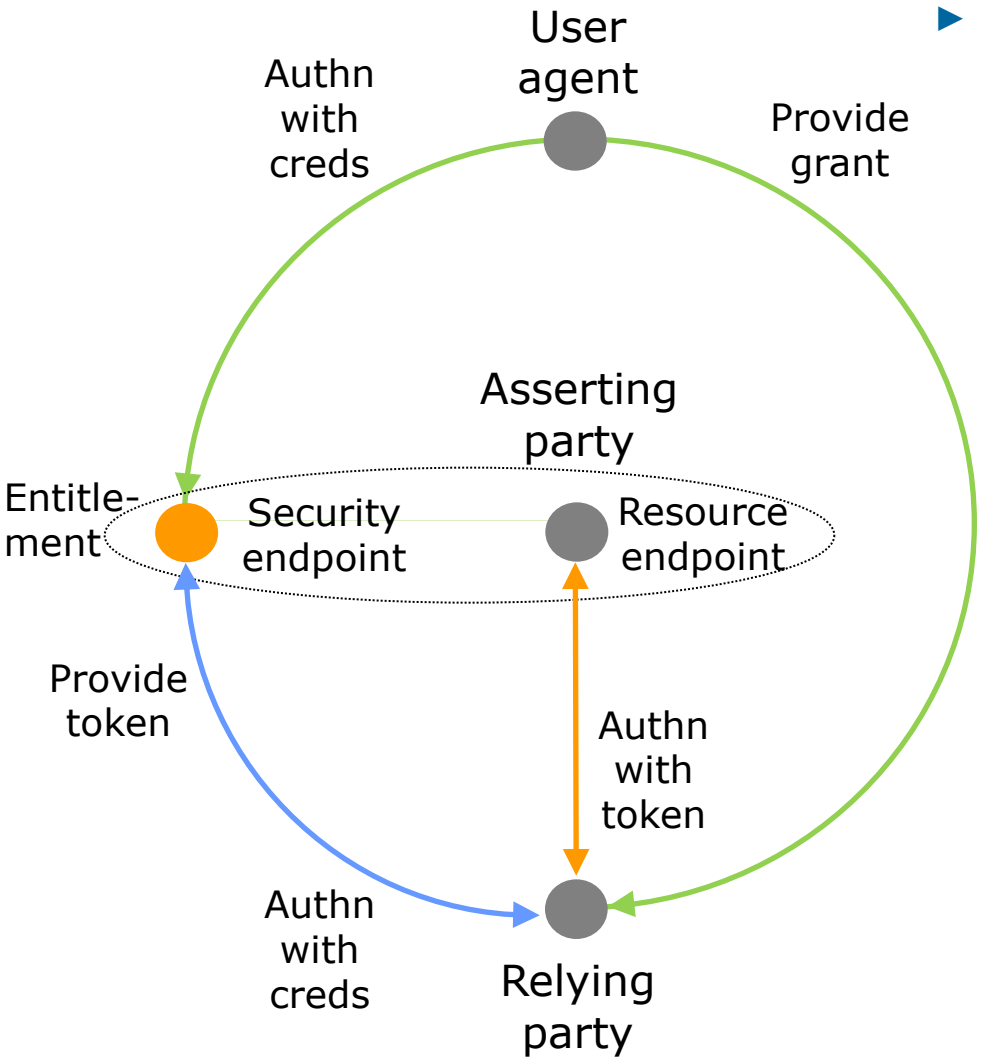
3-party scenarios

Use Cases Requiring 3-Party Exchanges

- Manage access to personal resources
  - Prerequisites: user and asserting/relying party authentication
- Move identity data (for self)
  - Prerequisites: user and asserting/relying party authentication
- Externalize user authentication
  - Prerequisite: user and asserting party authentication



3-Party Exchange Pattern – Functional Requirements

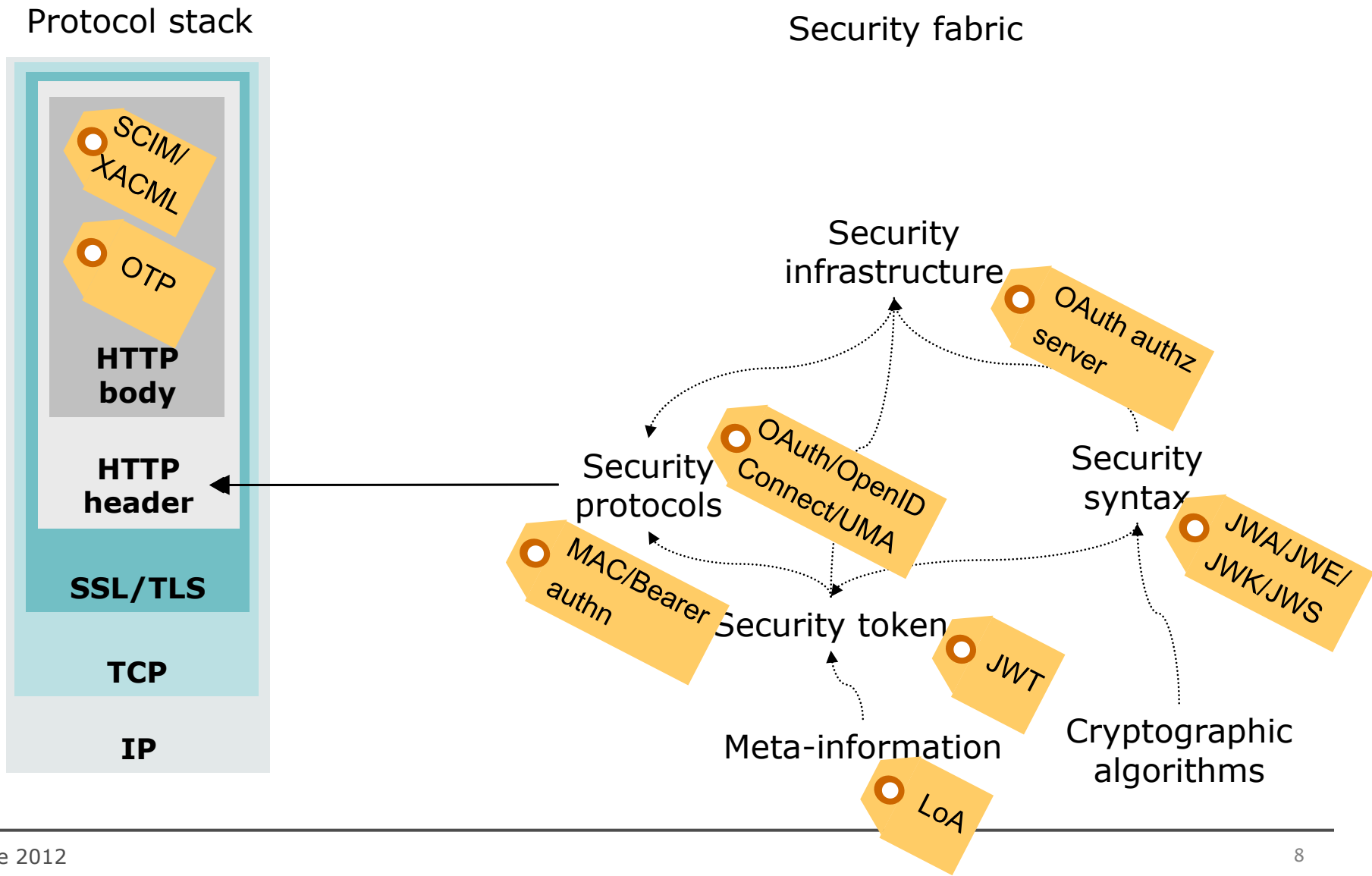


- Facilitate 3-party overlay:
  - User agent to asserting party - security endpoint:
    - Entitle relying party after authenticating
    - UI-style: entitlement dialogue with arbitrary Web user authentication
  - Relying party to asserting party - security endpoint:
    - Obtain *security token* after authenticating
    - API-style: new protocols with arbitrary Web client authentication
  - Relying party to asserting party – resource endpoint:
    - Obtain resource access after authenticating
    - API-style: new authentication protocols (token-based)

Identifying the New Entrants

- Constrained clients:
  - Interactions: N.a.
  - Compact representations: JWA, JWE, JWK, JWS (IETF jose WG) and JWT (IETF individual submission)
- API economy:
  - Authenticate API clients: HTTP `Bearer` and `MAC` authentication (IETF oauth WG)
  - Manage access to personal resources: OAuth (IETF oauth WG), UMA (Kantara)
  - Move identity data (for self): OpenID Connect (OpenID)
- Cloud:
  - Externalize user authentication: OpenID Connect (OpenID)
  - Manage identity data (for any): SCIM (IETF WG candidate)
  - Manage authorization: XACML 3.0 administration and delegation profile (OASIS)
- Disillusion:
  - Alternate authentication schemes: TOTP (RFC 6238), HOTP (RFC 4226), callbacks (custom)
  - Supply meta-information: assurance levels (NIST SP800-63, ITU-T X.1254 | ISO/IEC 29115, Kantara IAF)

Security-Enabling the HTTP Stack - *From ca. 2012*



Is It Real?

- 4-party security protocol examples:
  - UMA: http://kantarainitiative.org/confluence/display/uma/UMA+Implementations
- 3-party security protocol examples:
  - OpenID Connect:
    - Relying party: http://www8322u.sakura.ne.jp/oidconnect/
    - Asserting party: http://oauthssodemo.appspot.com/step/1
  - OAuth:
    - Relying party: https://twitter.com/#!/who_to_follow/import/
    - Asserting party: https://accounts.google.com/OAuthAuthorizeToken
    - Relying party: Import LinkedIn profile at Slideshare account
    - Asserting party: https://www.linkedin.com/uas/oauth/authorize

Relation to Open Source (Random Picks, Not Implying Endorsement)

- 4-party security protocols:
  - UMA: n.a.
- 3-party security protocols:
  - OpenID Connect: mitreid-connect (Java)
  - OAuth: Apache Amber, Google OAuth Client Library (Java), Scribe (Java), Spring Security OAuth (Java), Xotan (Java), DotNetOpenAuth, OAuth library for .NET (.NET), DevDefined OAuth (C#) and others
- 2-party security protocols:
  - HTTP `Bearer` authentication: cf. OAuth
  - HTTP `MAC` authentication: OAuth Signpost (Java, RFC 5849), cf. OAuth
- Security infrastructures:
  - OAuth authorization server: cf. OAuth
- Security tokens:
  - JWT: Nimbus JWT (Java), cf. OAuth
- Meta-information syntaxes:
  - LoA: OpenID Connect
- Security syntaxes:
  - JWA/E/K/S: Nimbus JWT (Java), cf. OAuth

Conclusions

- It is amazing what is happening right now – security-wise as well as IAM-wise
- The current innovation is triggered by use cases from the Internet IAM camp. In particular, it addresses needs related to Web 2.0 as well as social networks
- This does not imply that the emerging mechanisms are limited to these domains:
  - Other industries have matching use cases e.g. user-managed access to medical data to-be-shared among healthcare providers (ECRs – Electronic Case Records)
  - Their resolution delivers security mechanisms that can be (re-)used in other use cases
  - Security functionality for 3-party Web exchanges presents a main focus. Such 3-party exchanges also apply in other industries – probably with some other details but likely requiring similar patterns and approaches.
- The evolution of specifications, implementation of toolkits (many open source) and supply of services on the Internet happens in parallel
- This innovation in Web security is still ongoing and not yet concluded

More Details

- Pfaff, O.: New Trends in Web Security

Background

- Fielding, R.: Architectural Styles and the Design of Network-based Software Architectures. PhD Thesis. University of California, Irvine, 2000.
- Gutmann, P.: PKI: Lemon Markets and Lemonade. RSA Security Conference 2011.
- Jones, M.: The Emerging JSON-Based Identity Protocol Suite. W3C Workshop on Identity in the Browser, 2011.
- Machulak, M.P. et al.: User-Managed Access to Web Resources. Proceedings of the 6th ACM Workshop on Digital Identity Management, 2010.
- Mash-up directory: http://www.programmableweb.com/mashups/directory
- Pautasso, C.; Zimmermann, O.; Leymann, F.: RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision. Proc. of the 17th International World Wide Web Conference, Bejing, 2008.
- Prins, J.R.: DigiNotar Certificate Authority breach "Operation Black Tulip". Interim Report: Investigation DigiNotar Certificate Authority Environment, 2011.
- Rabin, J.; McCathieNevile, C. (eds.): Mobile Web Best Practices 1.0. W3C Recommendation 2008.
- Rutkowski, M. (ed.): Identity in the Cloud Use Cases Version 1.0. OASIS Committee Note, 2012.
- Web API directory: http://www.programmableweb.com/apis/directory
- Yegge, S.: Stevey's Google Platforms Rant. 2011

Author

Oliver Pfaff
Mail: oliver.frank.pfaff@gmail.com