#### User-Managed Access UMA Work Group

#### @UMAWG tinyurl.com/umawg | tinyurl.com/umafaq IIW 16, May 2013



### The "data price" for online service is too high: typing...

- Provisioning by hand
- Provisioning by value
- Oversharing
- Lying!

Name	
Street Address	
City	
State	Enter Text
Zip/Postal	
Province	
Country	Enter Text 🗸
Phone	
Email	
Preferred Communication	Postal Mail Phone E-mail

# The "data price" for online service is too high: connecting...

#### Lwitter



#### An application would like to connect to your account

The application **KanyeAnalysis**™ by **imma-let-u-finish** would like the ability to **access and update** your data on Twitter. This application also plans to **murder all of your children**.

#### Allow KanyeAnalysis<sup>™</sup> to murder your children?



- Meaningless consent to unfavorable terms
- Painful, inconsistent, and messy access management
- Oblivious oversharing

## The "data price" for online service is too high: private URLs...



This video is unlisted. Only those with the link can see it. Learn more

- Handy but insecure
- Unsuitable for really sensitive data

### Most data "sharing" today is back-channel and unconsented



Privacy is about context, control, choice and respect – so UMA enables a "digital footprint control console"

- Web 2.0 access control is inconsistent and unsophisticated
- To share with others, you have to list them literally
- You have to keep rebuilding your "circles" in new apps
- You can't advertise content without giving it away
- You can't get a global view of who accessed what

- You can **unify** access control under a single app
- Your access policies can test for **claims** like "over 18"
- You can **reuse** the same policies with multiple sites
- You can control access to stuff with **public** URLs
- You can manage and **revoke** access from one place

# UMA turns online sharing into a privacy-by-design solution



# UMA turns online sharing into a privacy-by-design solution



# UMA turns online sharing into a privacy-by-design solution



### UMA is a profile of OAuth, with bits added for interop and scale









### Key use cases

http://kantarainitiative.org/confluence/display/uma/Case+Studies

- Subscribing to a friend's personal cloud
- Sharing accessibility attributes (''GPII'')
- E-transcript sharing (''HEAR'')
- Patient-centric health data access
- Enterprise ''access management 2.0''



### Key implementations

http://kantarainitiative.org/confluence/display/uma/UMA+Implementations

- SMARTAM.net (running authorization service from Cloud Identity UK)
- Puma (Python libraries for RS- and client-enabling web apps) from ditto
- Fraunhofer AISEC opensource implementation in Java
- Gluu OX open-source implementation for Access Management 2.0 use cases









### Next steps

- Work on optimization opportunities when UMA and OpenID Connect are used together
- Issue "Implementor's Draft"
- Continue to work with AXN, Scalable Privacy, and others in ''trusted identities in cyberspace'' ecosystem
- Profile UMA for higher ed, accessibility attribute sharing, healthcare use cases
- We welcome your involvement and contributions
  - Become an UMAnitarian!
  - Follow @UMAWG on Twitter and UserManagedAccess on FB

### Questions? Thank you

@UMAWG tinyurl.com/umawg | tinyurl.com/umafaq IIW 16, May 2013



#### UMA phase 1: protecting a resource (rev 07b)

### Phase I: protect a resource



Section references are from http://docs.kantarainitiative.org/uma/draft-uma-core.html dated 6 Jan 2013

Token terminology:

\* PAT = protection API token

Binding obligations terminology, as shown in notes over entities representing obligated parties (see http://docs.kantarainitiative.org/uma/draft-uma-trust.html):

- \* Subject = Individual or Non-Person Entity
- \* Authorizing Party = Subject acting as resource owner
- \* AS Operator = Subject operating authorization server endpoint
- \* RS Operator = Subject operating resource server endpoint



www.websequencediagrams.com

### Phases 2 and 3: get authorization and access resource I of 3



UMA phases 2 and 3: getting authorization and accessing a resource





# Spec call tree for the UMA profile of OAuth



