

# MDQ and mdq-server

Ian A. Young, 2016-07-20

# What is MDQ?

- MetaData Query protocol
  - on-demand acquisition of per-entity metadata by an end entity from an oracle
  - *not* a general purpose search protocol
  - *not* intended for MetaData eXchange (MDX)

# MDQ Specification

- Initially implemented in Shibboleth SP
  - spec couldn't contradict that implementation
- Special handling required for SAML 2 artifacts ("{ sha1 }" wrinkle)
- Simplified by layering:
  - base layer understands only arbitrary IDs
  - SAML profile understands { sha1 }
  - other profiles could use IDs in other ways
- May be a pair of RFCs one day.

# MDQ Servers

- Server:
  - mdq-server
  - pyff ?
  - by design, any web server (modulo issues with encoding, { sha1 } IDs)
  - by design, any web proxy or cache

# MDQ clients

- Shibboleth SP
- Shibboleth IdP
- simpleSAMLphp
- anything which can make use of on-demand metadata, and can generate the URL from the entityID dynamically
- trivially (but not helpfully) anything which can be statically configured

# mdq-server

- Implementation of an MDQ server, based on:
  - Java
  - Spring Framework
  - Spring Boot
  - Shibboleth MDA (for customisation)

# mdq-server Layer 1

- Lower layer: ItemCollectionLibrary
  - Operates on a timer.
  - Responsible for acquiring collection of individual metadata documents and indexing it.
  - Uses an MDA pipeline to collect items.
  - Indexes based on ItemID and ItemTag
  - Each such identifier can be queried to result in a list of items.
  - Includes mechanisms for higher layer cache invalidation.

# mdq-server Layer 2

- Middle layer: MetadataService
  - Is queried for a rendered document for an ID.
  - Responsible for querying ItemCollectionLibrary and handling cache invalidation.
  - Uses an MDA pipeline to render the resulting item collection (e.g., including XML DSig).
  - Caches Results for each ID.
  - Results can have multiple Representations for different compression methods, each with different ETag values.



# mdq-server Layer 3

- Upper layer: EntitiesController
  - Spring MVC component.
  - Responsible for querying MetadataService and handling content negotiation.

# mdq-server Deployment

- Simple to deploy the Spring Boot executable JAR file.
- More flexible to deploy within a Docker container.
- Staging system deploys three containers.
- Production system has 2GB, one container:
  - Refreshes input aggregate every hour