Life Sciences Identifiers. Finally?

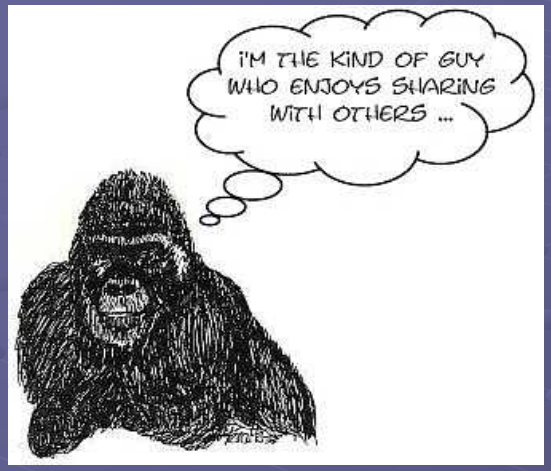
Presented by: Martin Senger
senger@ebi.ac.uk

Identifier? The names of pharaohs…

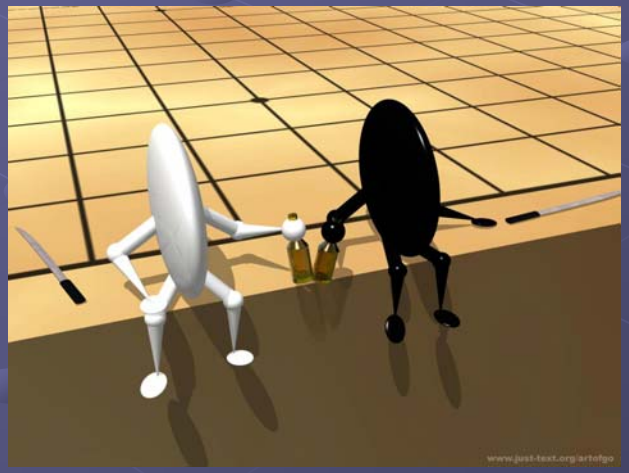
http://www.touregypt.net/featurestories/names.htm

- "…were important from the earliest times through the end of ancient Egyptian history, frequently offering clues to their personality, the period in which they lived and particularly, the gods that they most worshipped…"
- "…At times, some of the naming techniques of the ancient Egyptians could also lead to considerable confusion. This is obvious among some kings, who had a number of different names, but at times also changed their names, particularly when they inherited or otherwise ascended to the throne of Egypt. Furthermore, some individuals seem to possibly have had different names in different parts of Egypt…"

Identifiers – more practically



http://www.gorillaspeak.com/images/sharing.jpg



http://www.3dgo.org/sharing.jpg

- Identifiers are here to help sharing and integration in our domains. Fine, but:
  - "The sharing is an attitude"
    - Unless one wants to share data, the common/universal identifiers cannot help much

LSID - The golden rules
(even before we start to introduce LSIDs)

- The data with the same LSID never changes
  - they may cease to exist but the same LSID cannot be reused for anything else
- An LSID is location independent
  - when data moves its LSID stays the same
- LSID is not only syntax but also an API how to get data identified by this LSID
  - …and not only getting data but also metadata

History & Acknowledgement

- I3C started the initiative
  - concentration on early implementation
- IBM implemented it
  - main and first use case: PDB
- OMG standardized it
  - based on a joint submission of IBM, EBI and I3C
- People started to use it

More concretely…

(affiliations are of the time of the contribution)

- IBM
  - Jordi Albornoz
  - Stefan Atev
  - Ray Lee
  - Alister Lewis-Bowen
  - Sean Martin
  - Chetan Murthy
  - Dennis Quan
  - Ben Szekely
  - Alyssa Wolf
- EBI
  - Ugis Sarkans
  - Martin Senger
- Avaki Corporation
  - Philip Werner
  - Josh Apgar
  - Stephanos Bacon
- Millennium Pharmaceuticals, Inc
  - Ted Liefeld
- MIT/Whitehead Institute
  - Brian Gilman

Availability

- Specification
  - http://www.omg.org/cgi-bin/doc?dtc/04-05-01
  - http://www.omg.org/cgi-bin/doc?dtc/04-05-02
  - (recently about 14 [minor] issues corrected; the final available specification will get new document numbers in September 2004)
- Reference implementation (by IBM, for Java and Perl)
  - http://www-124.ibm.com/developerworks/oss/lsid/

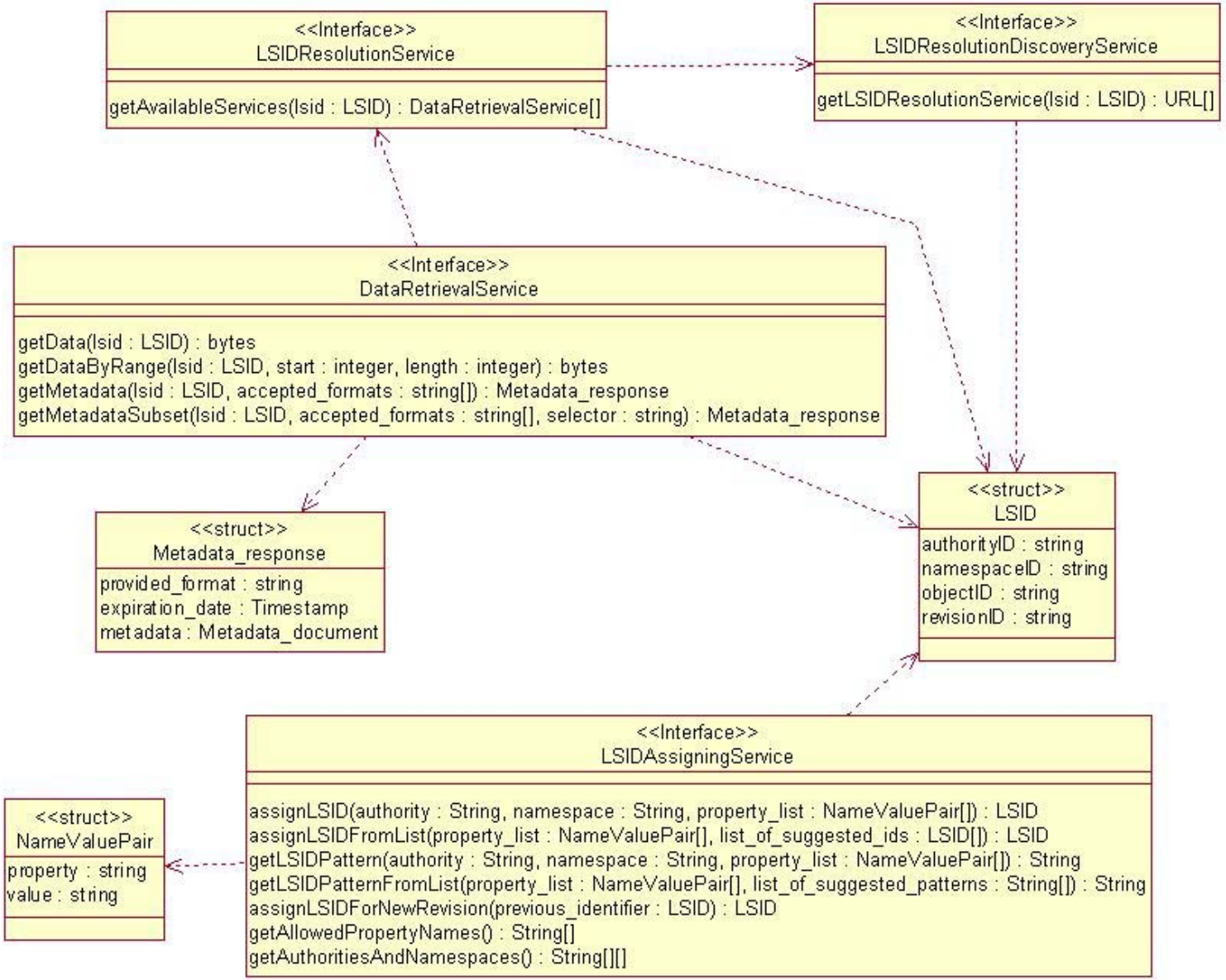
Three basic parts

- LSID Syntax
  - how to name uniquely data entities
- LSID Resolution Service
  - how to get (to) data entity from its LSID
  - *subpart:* how to find the LSID Resolution Service
- LSID Assigning Service
  - how to invent LSIDs for new data entities

LSID Syntax

- Examples
  - URN:LSID::ebi.ac.uk:SWISS-PROT.accession:P34355:3
  - URN:LSID:rcsb.org:PDB:1D4X:22
  - URN:LSID:ncbi.nlm.nih.gov:GenBank.accession:NT_001063:2
- Parts:
  - authority:namespace:object[:revision]
- Notes:
  - An LSID usually represents a piece of data, but it is allowed to have LSIDs representing an abstract entities or concepts
    - If an LSID represents real data, the LSID Resolution service must resolve always the same set of bytes representing such data
    - If an LSID represents an abstract entity the LSID resolution service must always resolve an empty result

LSID API


<<Interface>>
LSIDResolutionService
getAvailableServices(lsid : LSID) : DataRetrievalService[]
<<Interface>>
LSIDResolutionDiscoveryService
getLSIDResolutionService(lsid : LSID) : URL[]
<<Interface>>
DataRetrievalService
getData(lsid : LSID) : bytes
getDataByRange(lsid : LSID, start : integer, length : integer) : bytes
getMetadata(lsid : LSID, accepted_formats : string[]) : Metadata_response
getMetadataSubset(lsid : LSID, accepted_formats : string[], selector : string) : Metadata_response
<<struct>>
Metadata_response
provided_format : string
expiration_date : Timestamp
metadata : Metadata_document
<<struct>>
LSID
authorityID : string
namespaceID : string
objectID : string
revisionID : string
<<Interface>>
LSIDAssigningService
assignLSID(authority : String, namespace : String, property_list : NameValuePair[]) : LSID
assignLSIDFromList(property_list : NameValuePair[], list_of_suggested_ids : LSID[]) : LSID
getLSIDPattern(authority : String, namespace : String, property_list : NameValuePair[]) : String
getLSIDPatternFromList(property_list : NameValuePair[], list_of_suggested_patterns : String[]) : String
assignLSIDForNewRevision(previous_identifier : LSID) : LSID
getAllowedPropertyNames() : String[]
getAuthoritiesAndNamespaces() : String[][]
<<struct>>
NameValuePair
property : string
value : string

What technologies is that API for?

- Pure Java API
- Web Services
  - using SOAP over HTTP
  - using pure HTTP GET
  - using FTP
  - *all of these are real web services APIs – having their own WSDL descriptions*

How to find an appropriate LSID Resolution Service from an LSID?

- LSID Resolution Service is well advertised with the correct endpoint (URL, …)
  - usually the same resolution service works for a collection of data entities from the same repository

or

- If the "authority" field in the LSID is a domain name, a DDDS/DNS resolution service can be used to find the LSID Resolution Service
  - DDDS = Dynamic Delegation Discovery System

or

- Use "LSID Resolution Discovery Service" API
  - getLSIDResolutionServices(LSID)

Major discussion topics: attributes

- How many data attributes to include in an identifier?
  - e.g. should be a data format a part of an identifier
    - http://sequence.org/dna/v00808/fasta
  - finally, only version remained in LSID
    - not always easy to implement it…but it is AGT (“A Good Thing”)
    - treat everything else as “metadata”

Major discussion topics: location

- LSIDs are location independent
  - If we use URLs instead we could use more existing software (browsers, HTTP libraries,…) – so to get to data may be easier
  - But we could not move data and still be sure that they are the same
    - and the software libraries for accessing data are available (e.g. a plug-in into IE so an LSID can be resolve as any other URL using browser's address bar)

Major discussion topics: LSIDs for concepts

- Data returned by an LSID never change – so they must be in a particular format
- But we can use an LSID to identify a "concept" – represented by any data format. Then:
  - a call to *getData()* returns nothing…
  - …and a call to *getMetadata()* may give you LSIDs of various concrete formats for the same data "concept"

Major myths: Metadata are underspecified

- Well, it's not a myth, it's true
  - The metadata format was considered "out of scope" of the LSID specification – because we would never completely agree on them
  - But the specification has methods to find what metadata formats are used by each metadata provider
  - And, the format of metadata is not "deux ex machina" anyway – unless we agreed on ontologies of metadata predicates
    - And (paraphrasing Phillip Lord): "The ontologies can save the world only if the word agrees on sharing them"

Major Myths: I need to change my database schema to use LSIDs

- No, you don't (unless you want)
  - LSID is also an implementation of a software layer (called "Resolution service") that can map your DB records to the LSIDs
  - A difficult part is to keep the same LSIDs if your data are changing (versions)
    - But this is a general database problem, not a new one introduced by LSIDs
    - And it is AGT – to have the same identifier for the same data, for ever

Major Myths: I cannot get the latest data using the same LSID

- Yes, you can – using metadata
  - an LSID can identify “a concept”
    - …meaning: it does not return any real data
  - the same LSID may return metadata that can include another LSID pointing to the “latest” data
    - …and this LSID can change every time you have a new version
    - …there are already several projects doing this, using predicate “latest”, so I assume that a client-side library will soon appear to help the others

Major Myths: LSIDs are opaque

- Well, it's not a myth, it's true
  - The clients should always work with the whole LSID, and not to assume anything about data before they get the data
  - But service providers can "hide" into LSID useful information that can help to map it back to their databases
    - the API for LSID Assigning Service helps here
  - And if a service provider wants to tell more about the data, there are always metadata

Major Myths: LSIDs are domain specific

- It's a typical myth
  - LSID are general…
  - …but are named "Life Sciences"
    - historically, it was their first name
    - practically, it was easier to standardize

Finally, who is using LSIDs

- Examples of the projects and sites I am aware of:
  - myGrid (Taverna workbench, workflow repository,…)
  - BioMoby
  - Broad Institute, Cambridge, MA
  - several universities (Toronto, Vermont, Tufts, Harvard – for astronomy, Wisconsin)
  - The Genome Database (GDB)
  - IBM is adding supports for LSIDs in their products (e.g. InsightLink Annotation)
  - …