Local Prototype for DAM-LR

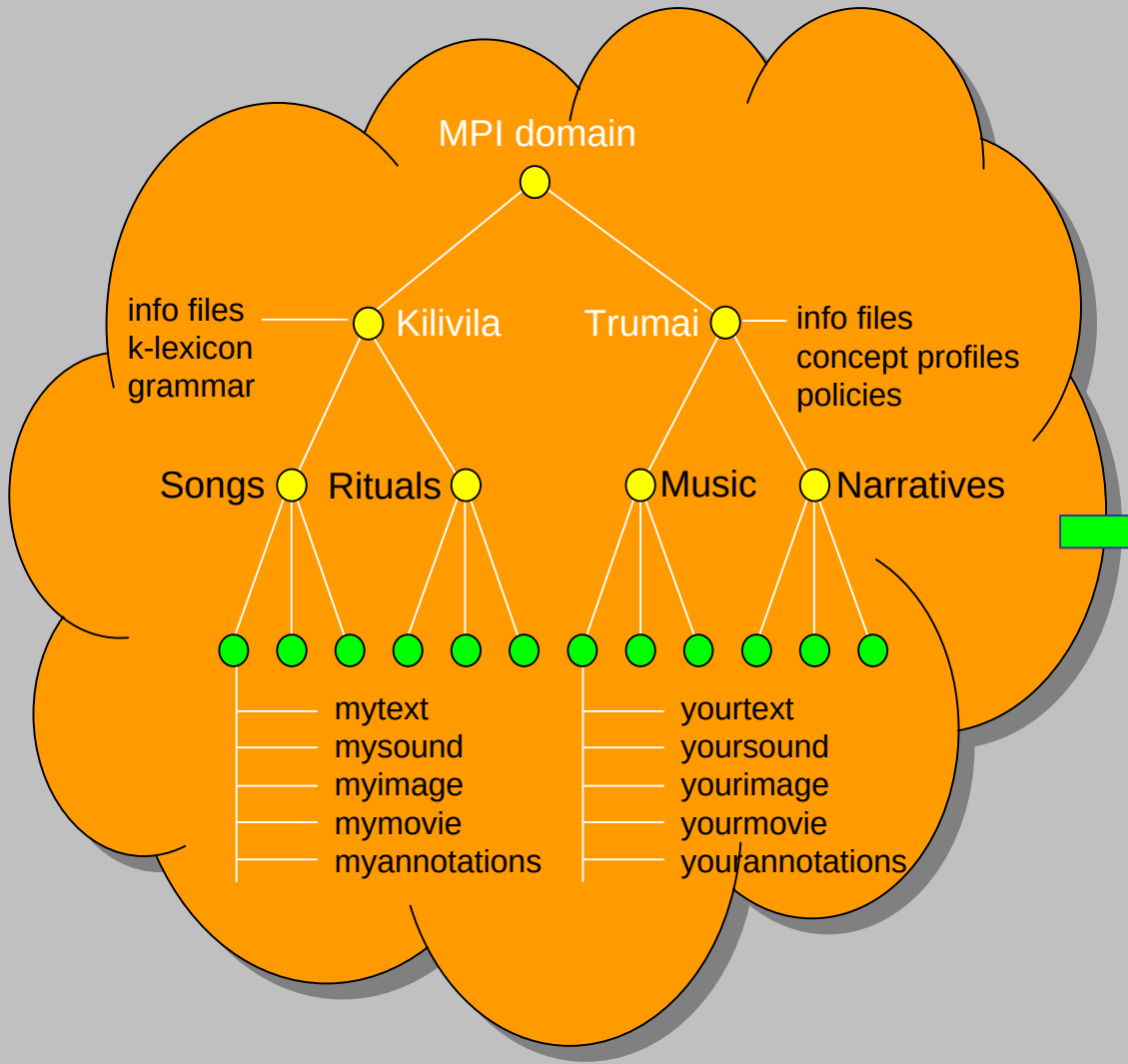
- only reference -

MPI Team
(Daan, Freddy, Peter)

A typical LR Archive

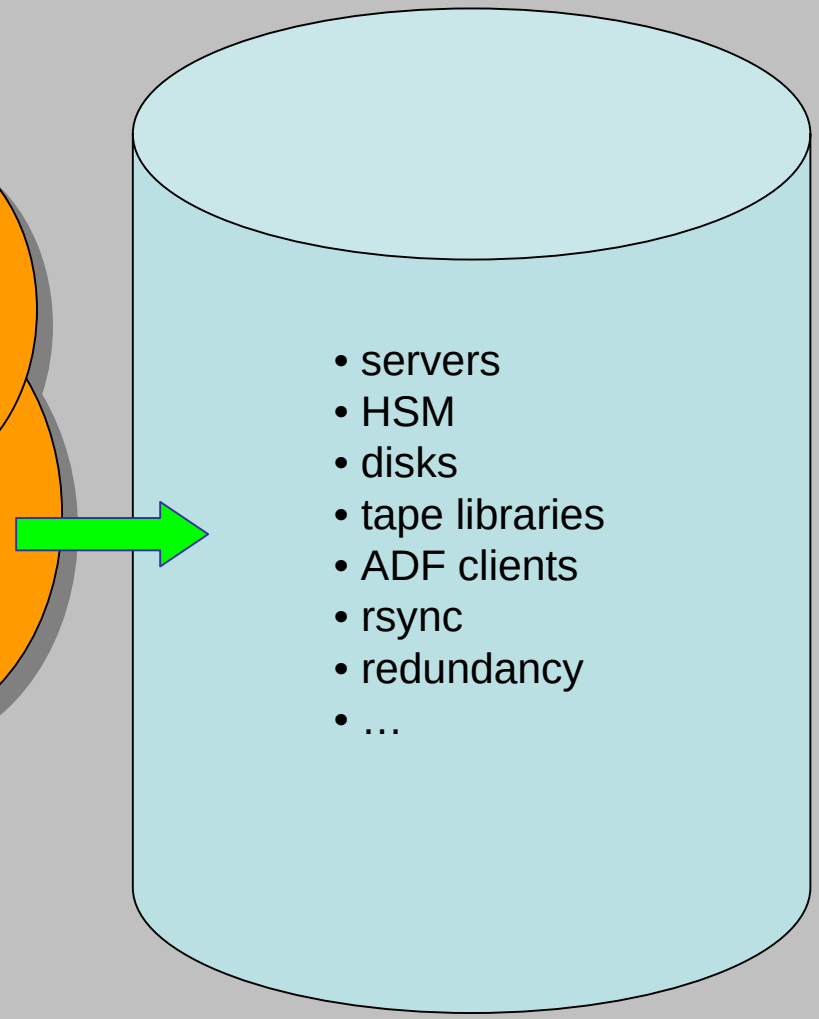
Virtual IMDI Organization

Users, Archive Managers

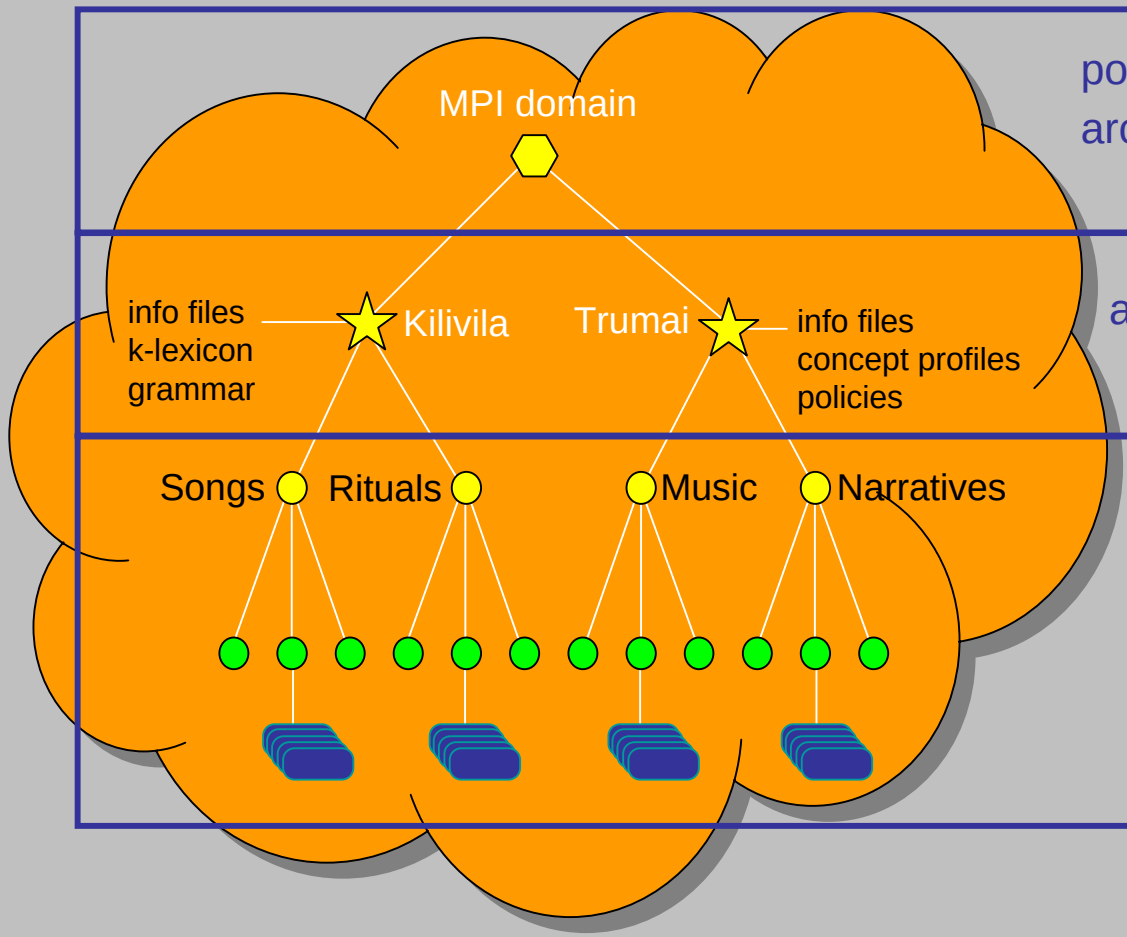


Physical Organization

System/Archive Managers



Typical IMDI Organization



portal root (indexes, servlets, …)

archive root (management, org authority, A&A systems, …)

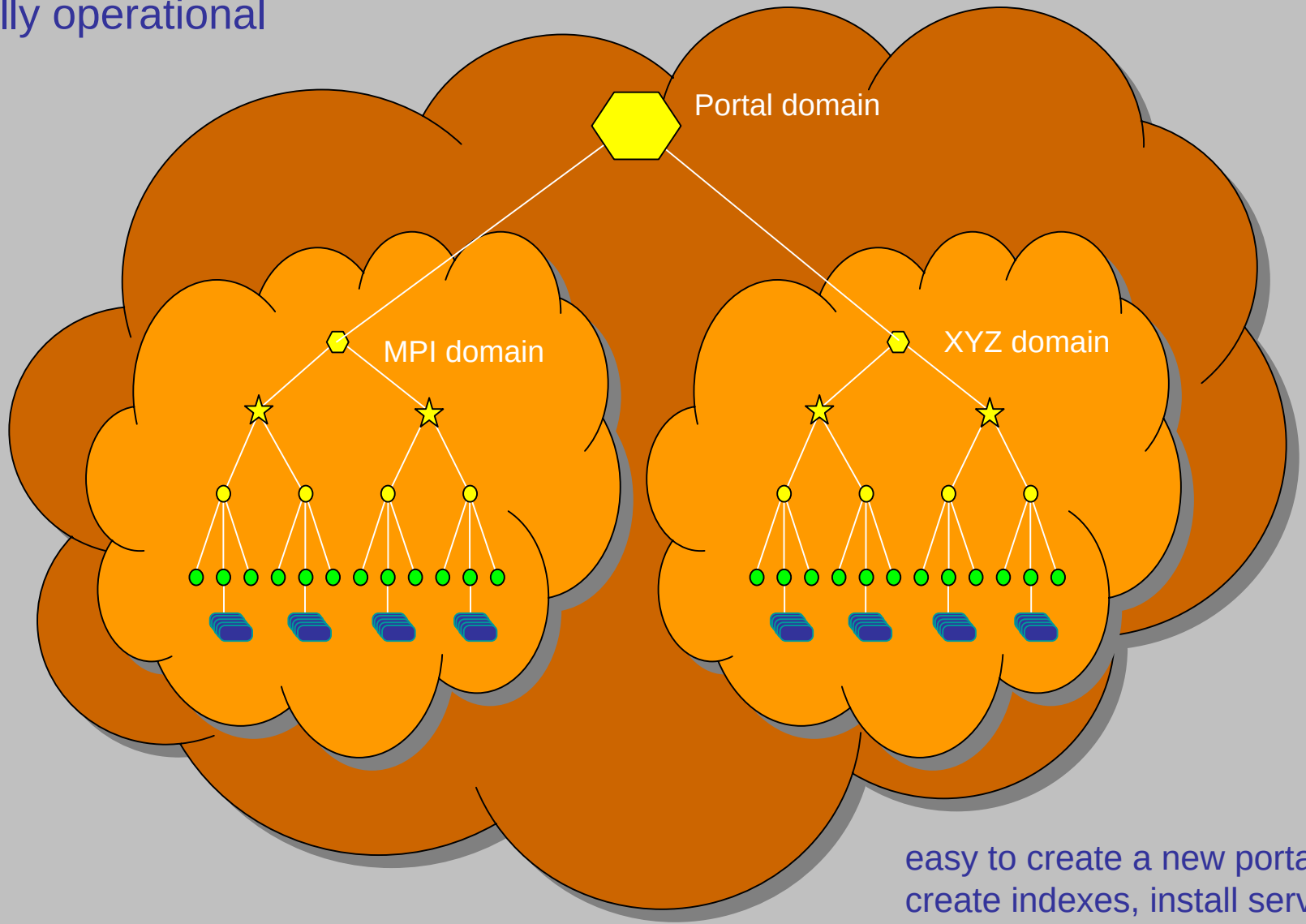
authority root (policies, rights, formats, structure, ling encoding, …)

arbitrary corpus structure according to linguistic insights (flat and/or deep)

resource bundles

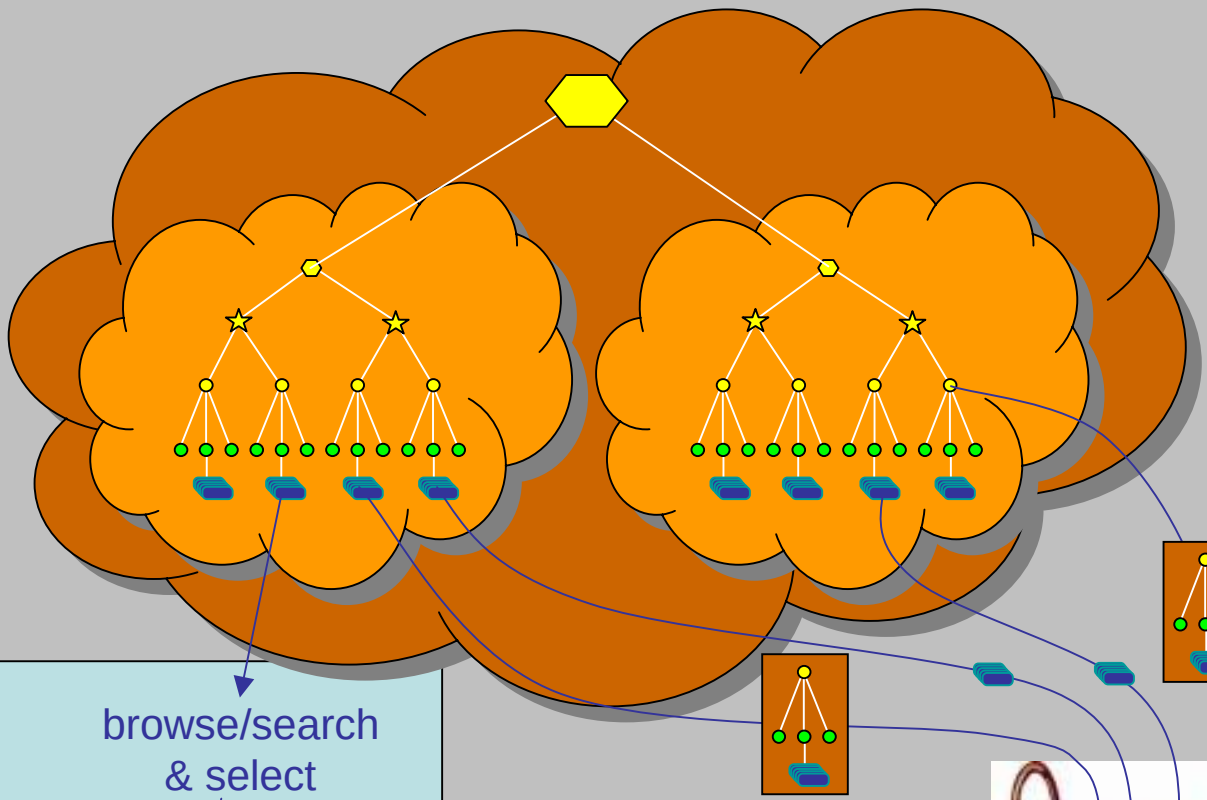
Joint IMDI Domain

fully operational

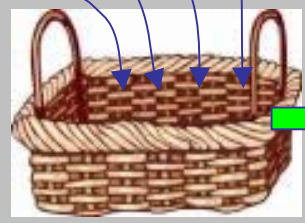


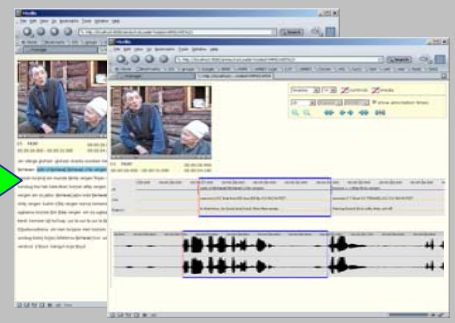
easy to create a new portal root
create indexes, install servlets, …

Private Domain - DELAMAN Vision



- user selects resources from everywhere
- this is his/her private virtual temporary domain
- can store it as profile
- invoked by session
  - one authentication step
  - one authorization step
- then search (MD + content)
- visualization/listening/…
- comparison
- …





browse/search & select

authenticate authorize

play

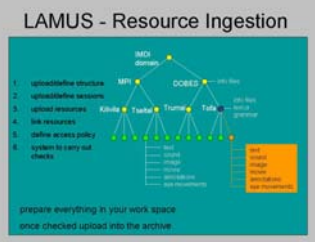
web-browser & plug-ins

complex selection and utilization framework currently in development incl. ontology support (3 Max-Planck-Institutes)

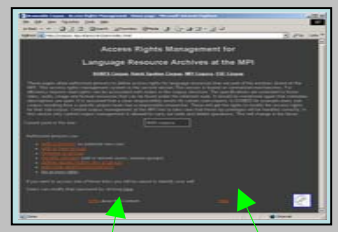
Local Prototype

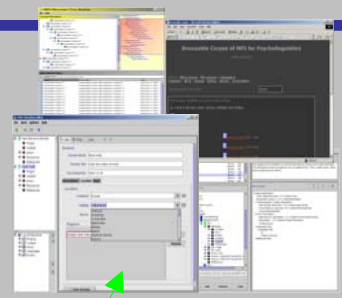
done
in progress
to start

LAMUS V1 is ready

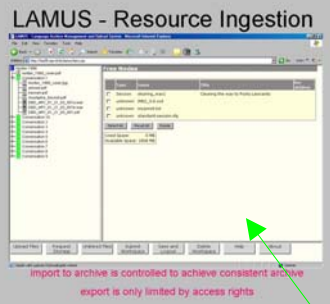


AMS is ready





IMDI infrastructure is ready



Archive Utility Layer

- User Authentication
- Access Management
- Metadata Tools

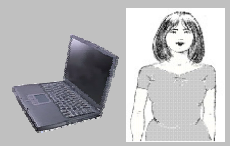
Data Ingestion& Management

Creator

LAMUS is the Archive Gatekeeper for new resources and updates

The Archive

- Domain of Registered Primary and Secondary Resources
- Domain of Descriptive Metadata

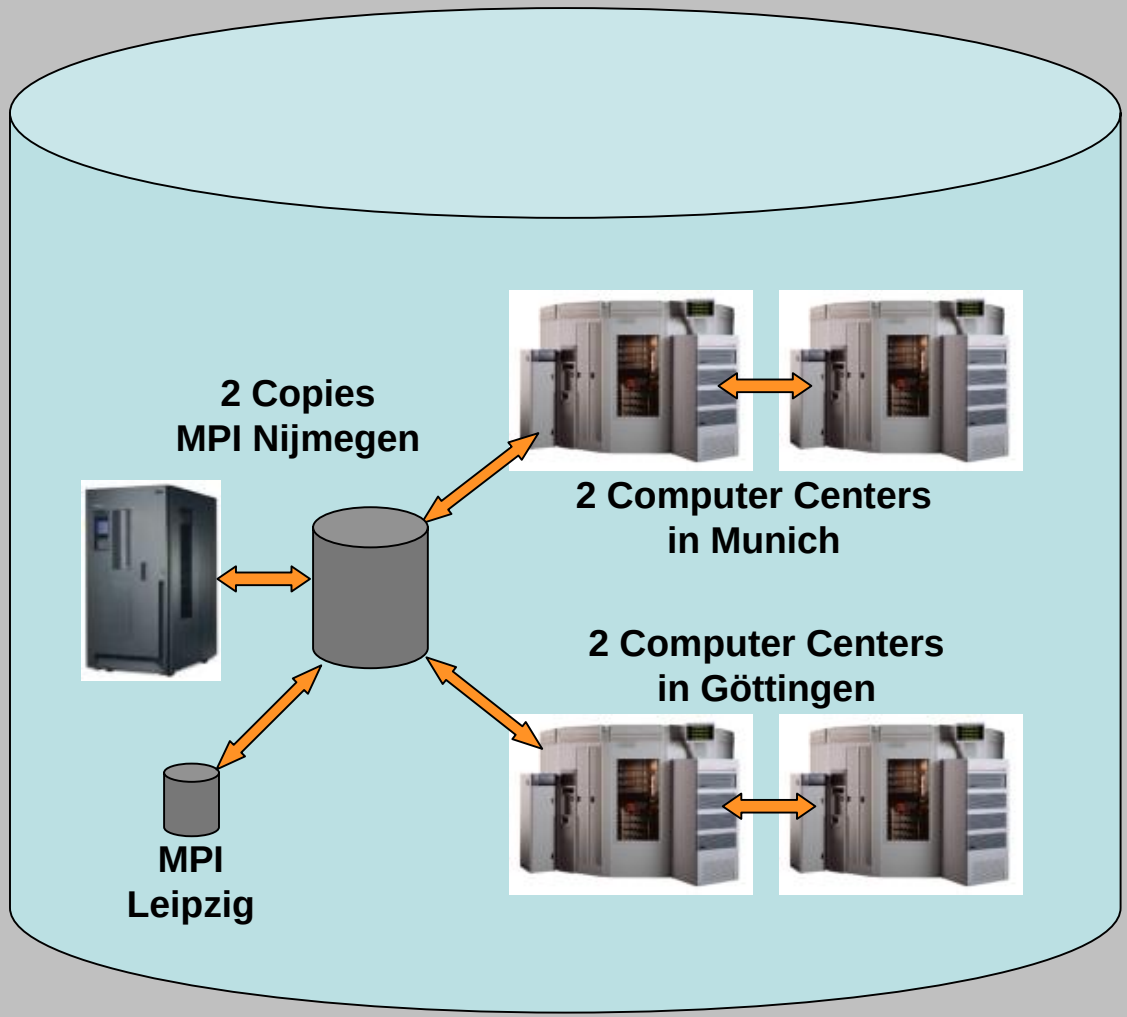


User

Important:
all resources available in standard formats and directly accessible including IMDI descriptions using HTTP

What is missing?

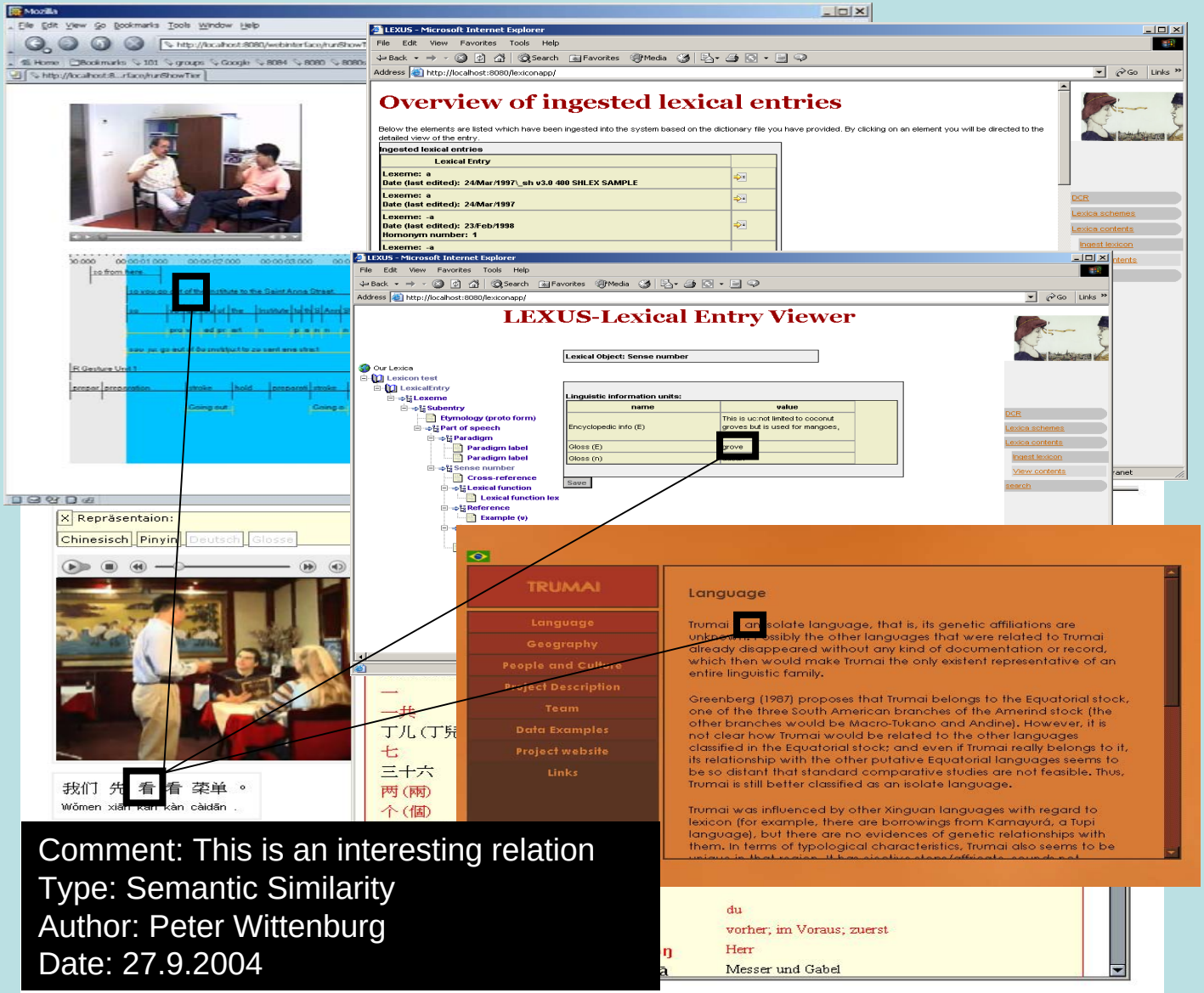
Physical Organization ok



System and Archive Management Operations not yet transparent to archive organization

- **URIDs are missing**
- **MD point to URLs**
- **have to introduce URIDs now to make all names unique and addresses stable**
- **Handle System candidate**

Web-based Commentary



another reason to introduce URIDs

need one point where changes are done

moving and copying has to use special tools

LAMUS has to create URIDs

Prototype Requirements I

4.1 Metadata Layer **(almost all done)**

- All archive resources must be described by IMDI metadata.
- All IMDI descriptions must be open and accessible as XML files.
- Structured and unstructured metadata searches within the IMDI domain have to be possible. (right now making it symmetric)
- Searches with the help of search engines such as Google have to be supported.
- The IMDI resources have to be offered as OLAC/DC records according to the OAI PMH protocol to make them searchable via OLAC services providers.

(OLAC service provider has problems since we deliver too many records)

- The IMDI files must be linkable into a linked domain that supports an organization of resources into logical bundles, browsing and management.
- It must be simple to register and integrate new IMDI-based repositories into the IMDI domain.
- It must be easy to setup an IMDI portal.
- If authorized it must be easy for users to access resources via the metadata descriptions.

Prototype Requirements II

4.2 Physical Layer **(except URIDs all done)**

- The physical storage must be transparent to the user, i.e. the user should not have to deal with servers, disks etc.
- The physical location of the resources should be easily modifiable without causing problems for the users. (URIDs are missing)
- The organization of the archive should allow for the copying of whole and sub-parts of the archive to new archives to support long-term preservation and redundant access paths.
- Each archived resource has to be identified by a unique resource identifier (URID). The metadata descriptions have to contain URIDs to refer to the resources (for archive objects). (missing)
- The storage concept must be such that several copies of all resources can be generated automatically and location resolving can be carried out.

(the last point can only be done when URIDs are in place)

- The stored resources have to be in archivable formats and directly accessible for authorized people.

Prototype Requirements III

4.3 Access Management **(almost all done)**

- The access management system must support the definition of policies (declaration of code of conducts, usage, etc) and rights.
- The access management system must support the specification of usages and temporary tickets associated with these usages. (usage declaration to come)
- The access management system must support efficient electronic operations via web interfaces and a delegation mechanism to allow resource owners to define access policies from remote sites.
- Access policy specifications must be based on the metadata layer, i.e. the physical layer is transparent to the definer and the specifications are independent of the physical location of the resources.
- It must be possible to specify domains of authority in the metadata layer.
- Delegation of rights must be supported

Prototype Requirements IV

4.4 Ingesting Resources **(essential work to be done)**

- The possibility to integrate new resources or update existing resources into the language resource archive has to be controlled by an upload system that ensures that its coherence and consistency is guaranteed. (checks to be added)
- The user has to be provided with a workspace mechanism which allows him/her to arrange the data and test its compliance until it is ready for integration.
- The upload system has to be equipped with a configurable list of permitted file types and where possible with format checkers. In particular for complex resource types, dependent types must be indicated, some being required such as a schema. (to be integrated)
- The upload must support the definition and integration of an upload node in the existing archive, archive structure, metadata descriptions and resources. It must support the proper linking of these elements.
- Versioning must be done in the case of integrating new versions. (to come)

Prototype Requirements V

4.5 Archive Management **(work to be done)**

An archive management shell has to be available to carry out typical management activities such as:

- copying and moving data while retaining the correctness of the archive's organizational links (URIDs needed)
- checking the consistency of all links in the archive and modifying them where necessary (programs available)
- checking the format and technical encoding correctness of all resources where possible (some ready, Shoebox is difficult)
- automatically generating additional resource types for presentation purposes such as MP3, MPEG4, etc (in progress)
- creating different types of statistics (programs available)
- the possibility of removing sub-parts of the archive which is the most dangerous operation and which therefore has to be guided (well …)

Local Prototype

Acknowledgements:
local prototype development funded by VWS, INL and MPI

What else?

Did we forget something?