Musings on the DINA Architecture
Perspectives, Ideas, Tasks

Stefan Daume

Swedish Museum of Natural History
Biodiversity Informatics Group

EU BON Workshop - 17. September 2014

Naturhistoriska riksmuseet

EU BON

Architectural perspectives & drivers

- High-level strategic perspective
- Usability and delivery
- Experiences with the Specify relational database model
- Integration of new biodiversity information sources
- Workflows and data lifecycles with regard to large-scale digitization projects

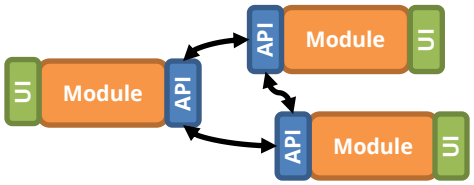
How ambitious is the project and what are the goals in an international context?

- International reach
- Set de-facto standard for collection management
- Organisational structure and technical framework to mobilise new partners

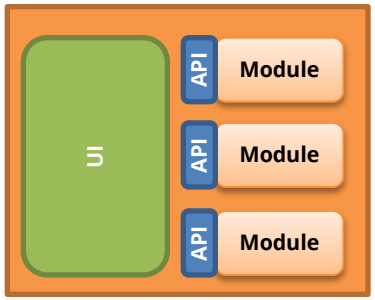
How fragmented and loosely coupled or how compact and coherent should DINA be?

- „System of systems“ with multiple delivery options
- „Integrated platform“ with multiple delivery options

DINA: System of systems



DINA: Platform



Systems integration ⟷ **Assembly** ⟷ *Systems engineering*

Administrators ⟷ **Configuration** ⟷ *End-users*

Heterogeneous ⟷ **Technology stacks** ⟷ *Homogeneous*

What architecture and technological choices guarantee a high-level of data model flexibility and address already identified additional schema requirements?

- Rigorous semantics and formalisation of data models
- Accommodate parallel institutional requirements
- Separate data models from persistence technology constraints

Accession

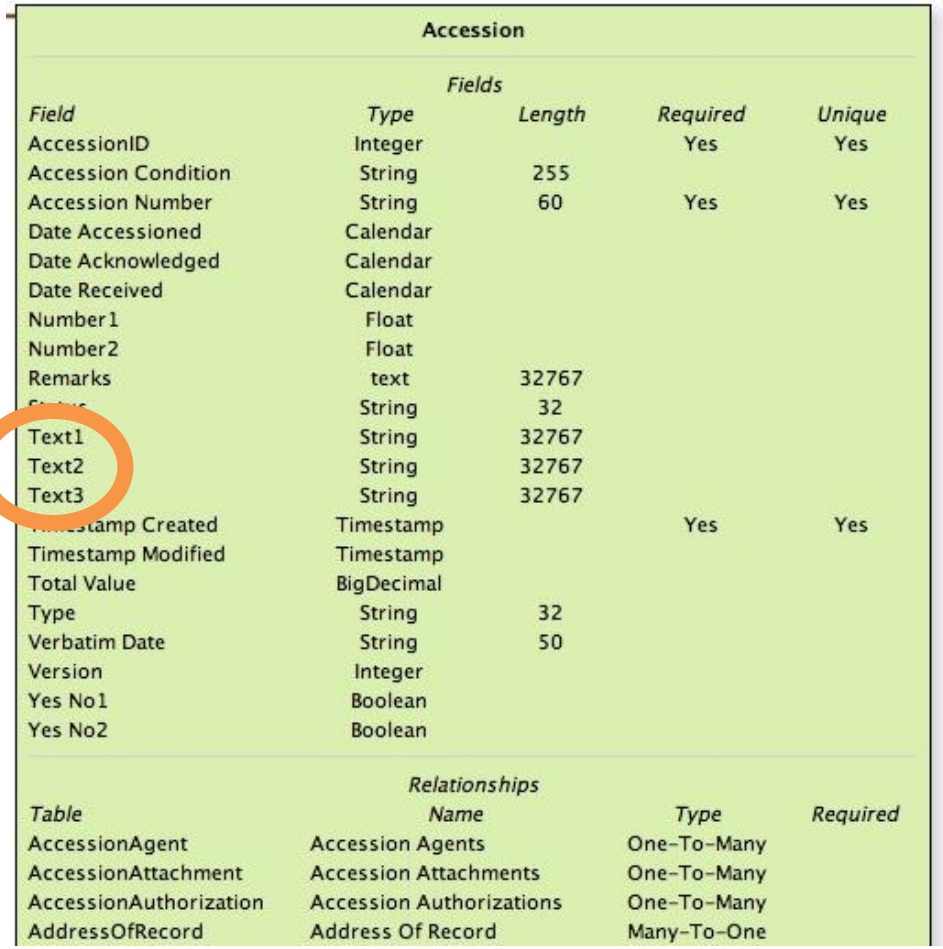
Fields

Field	*Type*	*Length*	*Required*	*Unique*
AccessionID	Integer		Yes	Yes
Accession Condition	String	255		
Accession Number	String	60	Yes	Yes
Date Accessioned	Calendar			
Date Acknowledged	Calendar			
Date Received	Calendar			
Number1	Float			
Number2	Float			
Remarks	text	32767		
[illegible]	String	32		
Text1	String	32767		
Text2	String	32767		
Text3	String	32767		
[illegible]stamp Created	Timestamp		Yes	Yes
Timestamp Modified	Timestamp			
Total Value	BigDecimal			
Type	String	32		
Verbatim Date	String	50		
Version	Integer			
Yes No1	Boolean			
Yes No2	Boolean			

Relationships

Table	*Name*	*Type*	*Required*
AccessionAgent	Accession Agents	One-To-Many	
AccessionAttachment	Accession Attachments	One-To-Many	
AccessionAuthorization	Accession Authorizations	One-To-Many	
AddressOfRecord	Address Of Record	Many-To-One	

Externalising the semantics of the data model without formalising them.



Accession

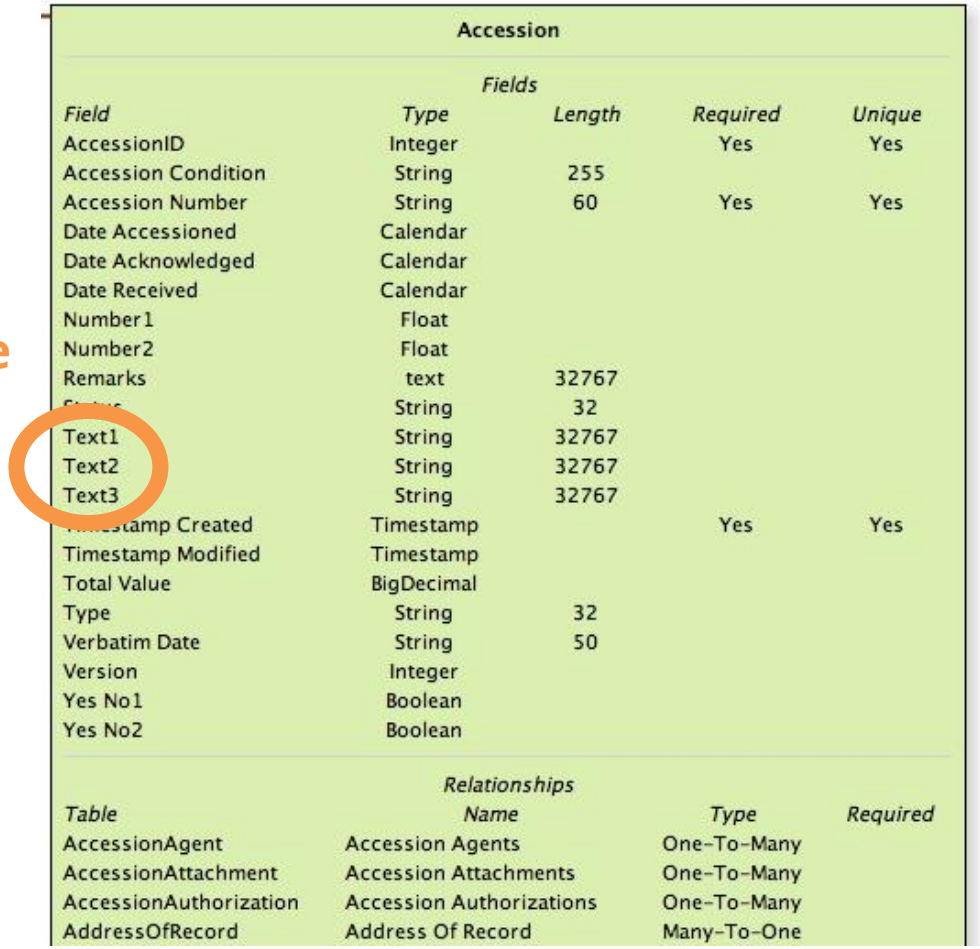
Fields

Field	*Type*	*Length*	*Required*	*Unique*
AccessionID	Integer		Yes	Yes
Accession Condition	String	255		
Accession Number	String	60	Yes	Yes
Date Accessioned	Calendar			
Date Acknowledged	Calendar			
Date Received	Calendar			
Number1	Float			
Number2	Float			
Remarks	text	32767		
[illegible]	String	32		
Text1	String	32767		
Text2	String	32767		
Text3	String	32767		
[illegible]amp Created	Timestamp		Yes	Yes
Timestamp Modified	Timestamp			
Total Value	BigDecimal			
Type	String	32		
Verbatim Date	String	50		
Version	Integer			
Yes No1	Boolean			
Yes No2	Boolean			

Relationships

Table	*Name*	*Type*	*Required*
AccessionAgent	Accession Agents	One-To-Many	
AccessionAttachment	Accession Attachments	One-To-Many	
AccessionAuthorization	Accession Authorizations	One-To-Many	
AddressOfRecord	Address Of Record	Many-To-One	

Externalising the semantics of the data model without formalising them.



Accession

Fields

Field	*Type*	*Length*	*Required*	*Unique*
AccessionID	Integer		Yes	Yes
Accession Condition	String	255		
Accession Number	String	60	Yes	Yes
Date Accessioned	Calendar			
Date Acknowledged	Calendar			
Date Received	Calendar			
Number1	Float			
Number2	Float			
Remarks	text	32767		
[illegible]	String	32		
Text1	String	32767		
Text2	String	32767		
Text3	String	32767		
[illegible]amp Created	Timestamp		Yes	Yes
Timestamp Modified	Timestamp			
Total Value	BigDecimal			
Type	String	32		
Verbatim Date	String	50		
Version	Integer			
Yes No1	Boolean			
Yes No2	Boolean			

Relationships

Table	*Name*	*Type*	*Required*
AccessionAgent	Accession Agents	One-To-Many	
AccessionAttachment	Accession Attachments	One-To-Many	
AccessionAuthorization	Accession Authorizations	One-To-Many	
AddressOfRecord	Address Of Record	Many-To-One	

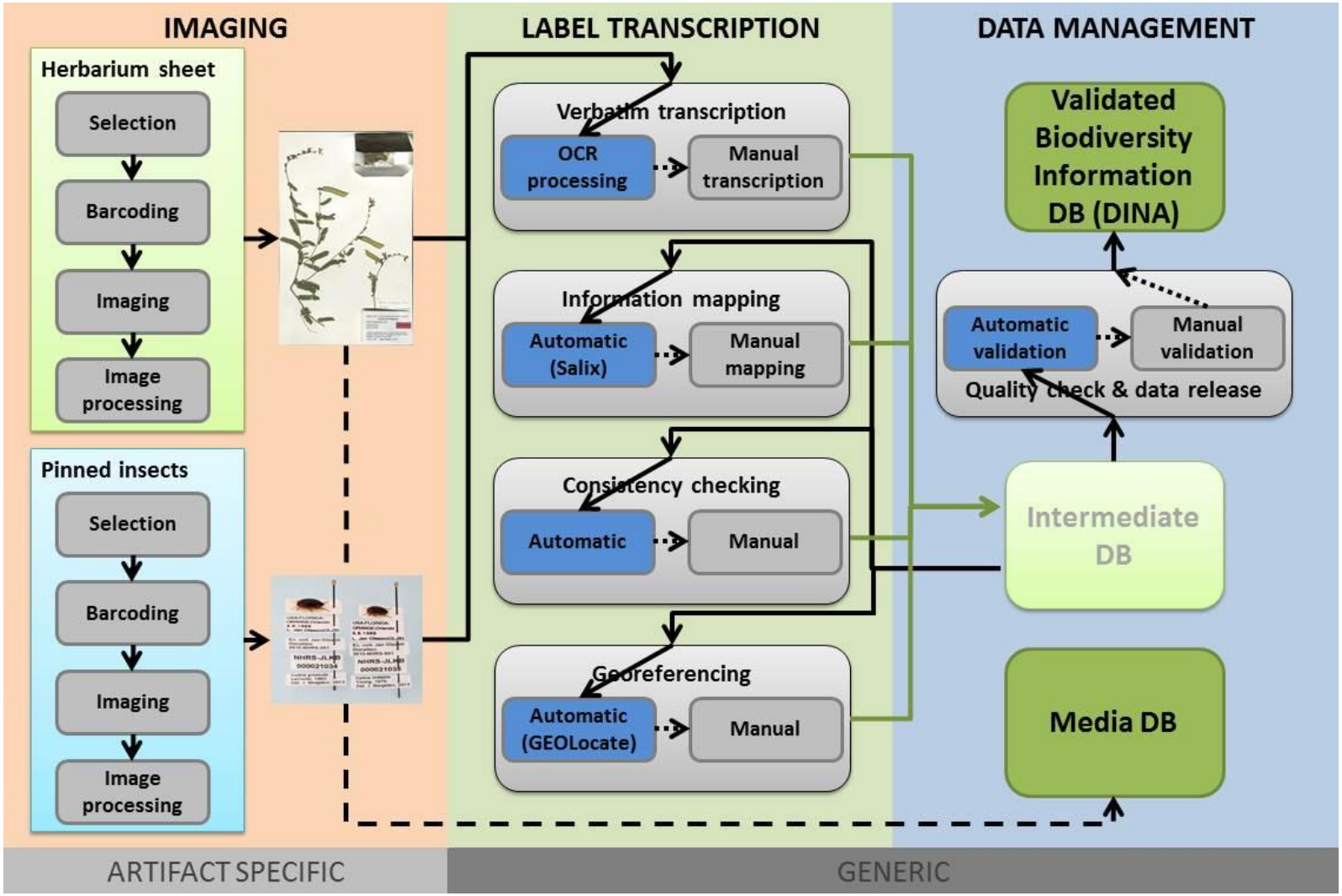
Short-term workaround: mapping tables connecting to external sources/data models.

What architecture and technological choices cover access and use of new or even as yet unidentified data and workflows anticipated for integration to the DINA systems?

- Integrate traditional collection data, observations, literature, DNA, social media, etc
- Avoid „data model lock“ and non-formalised semantics

How can the new DINA system best accommodate the expected massive data flows from digitization projects?

- Collection management lifecycles with variable timelines
- Annotation-based workflows
- Annotations as valuable meta-data for data processing innovations


IMAGING
LABEL TRANSCRIPTION
DATA MANAGEMENT
Herbarium sheet
Selection
Barcoding
Imaging
Image processing
Pinned insects
Selection
Barcoding
Imaging
Image processing
Verbatim transcription
OCR processing
Manual transcription
Information mapping
Automatic (Salix)
Manual mapping
Consistency checking
Automatic
Manual
Georeferencing
Automatic (GEOLocate)
Manual
Validated Biodiversity Information DB (DINA)
Automatic validation
Manual validation
Quality check & data release
Intermediate DB
Media DB
ARTIFACT SPECIFIC
GENERIC

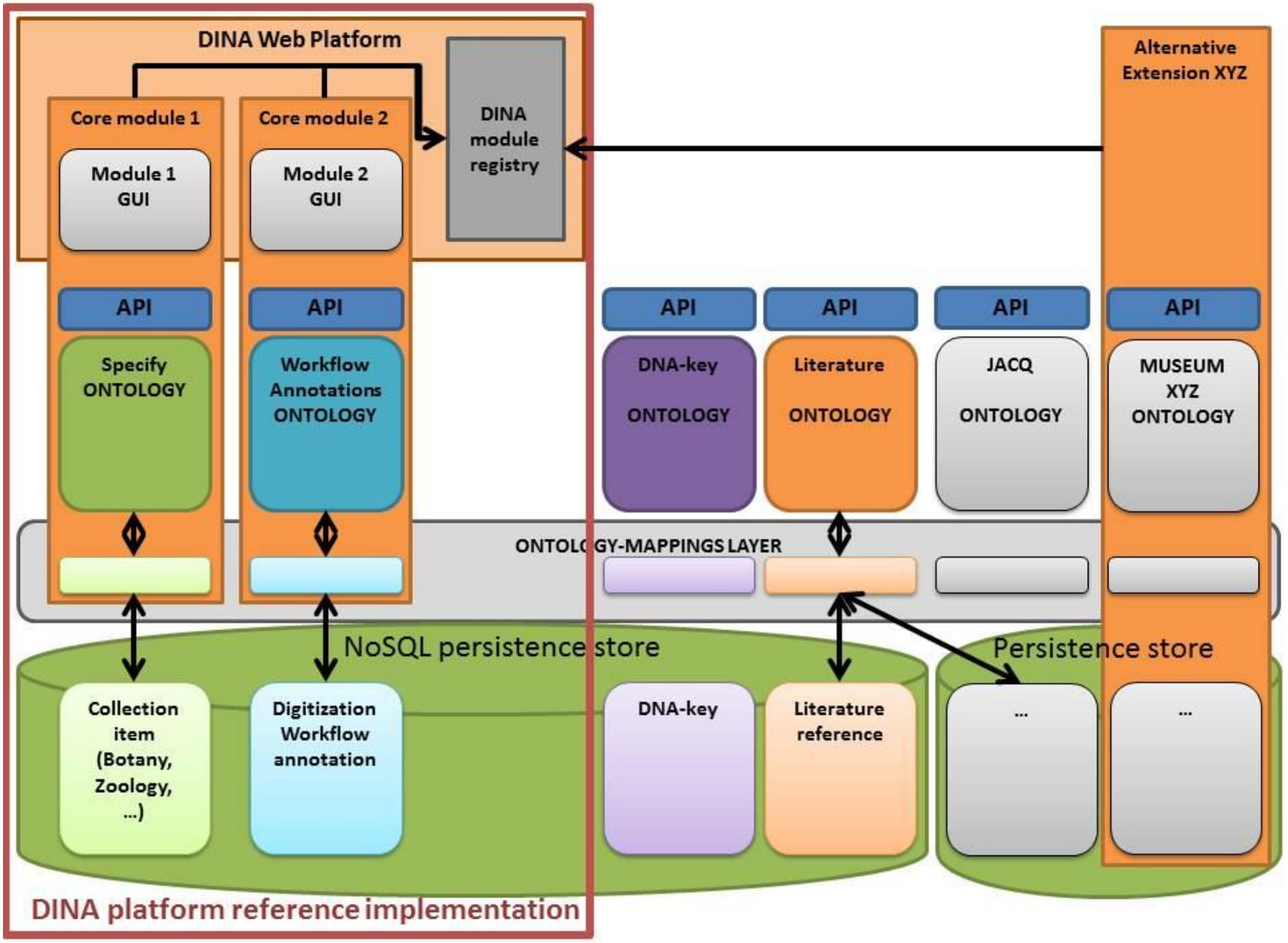
Architecture and technology options

Possible conclusions

- Develop DINA as a well-integrated coherent web-based platform that is customizable and extensible through an extension (“plugin”) model (inspired by the open source platforms such as Eclipse).

Possible conclusions

- Move away from the relational database model and separate the semantics from the persistence by connecting data model ontologies to NoSQL data stores.


DINA Web Platform
Core module 1
Core module 2
DINA module registry
Module 1 GUI
Module 2 GUI
API
API
Specify ONTOLOGY
Workflow Annotations ONTOLOGY
Alternative Extension XYZ
API
API
API
API
DNA-key ONTOLOGY
Literature ONTOLOGY
JACQ ONTOLOGY
MUSEUM XYZ ONTOLOGY
ONTOLOGY-MAPPINGS LAYER
NoSQL persistence store
Persistence store
Collection item (Botany, Zoology, ...)
Digitization Workflow annotation
DNA-key
Literature reference
...
...
DINA platform reference implementation

Caveats

- Ontologies
  - Powerful, but can get complex
  - Flexible, but have to be wrapped in a user-friendly way
- NoSQL
  - Prominent examples, but not yet as mature as RDBs
  - Flexible, but may not be required if data types are primarily static and homogeneous
- Platform models
  - User-friendly, but have to be targeted at the right user groups
  - Clear constraints, but could be perceived as too restrictive

Tasks & Next Steps

DINA „roadmap“ – next steps

- DINA development
  - REST APIs
    - Agree and publish DINA standards
    - Implement and publish for partner modules
  - Module reference UIs
  - Store documentation in shared repo & pull versioned docs into DINA-Wiki

DINA „roadmap“ – next steps

- Collaborative tools & processes
  - Setup of shared Github repo
  - Continuous integration (most likely Jenkins)
  - Evaluation of issue tracking tools (JIRA, Redmine, …)
  - IRC channel(s) – botposted to Wiki
  - Mailing list(s)
  - Monthly Google Hangout

DINA „roadmap“ – next steps

- Practices & Principles
  - Agile processes (tool support TBD)
- Licensing
  - SETF to suggest suitable DINA license(s)

Thanks!