

# Blending and deblending data in the daily routine of a university library



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT







# Blending and Deblending

## 1) Named Entity



## 2) SPARQL Endpoint



```
SELECT ?s WHERE {  
  {  
    ?s rdfs:label "Olea Europaea"@en;  
    a owl:Thing .  
  } UNION {  
    ?altName rdfs:label "Olea Europaea"@en;  
    dbo:wikiPageRedirects ?s .  
  }  
  [...]  
}
```

## 4) Ingest to SOLR index

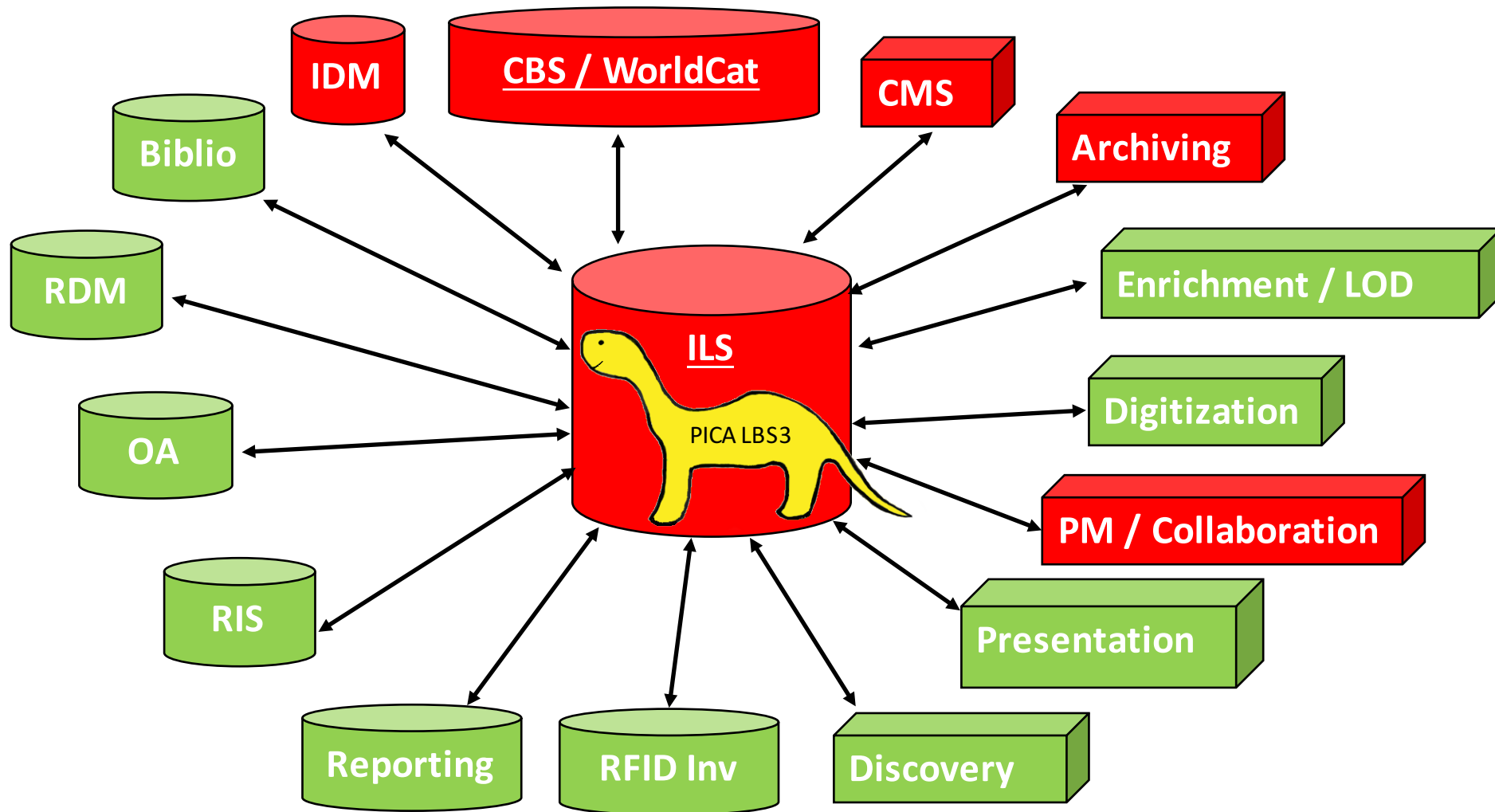


## 3) Blend with attributes from LOD

rdfs:label: Olive  
dbo:family: Oleaceae  
dbo:genus: Olea  
dbo:order: Lamiales  
dbo:abstract: The olive /'blɪv/ or /'ɑːləv/, known by the botanical name *Olea europaea*, meaning "european olive", (syn. *Olea sylvestris*) is a species of small tree in the family Oleaceae, found in much of Africa, the Mediterranean Basin from Portugal to the Levant, [...]  
foaf:depiction:



# Part I: Status Quo of Data (De-)Blending





# By means of what software?

## Commercial

- PICA LBS version 3 and 4
- PICA CBS
- Novell NetIQ
- ExLibris Rosetta
- Atlassian Confluence / JIRA / Crowd
- FirstSpirit CMS
- Visual Library

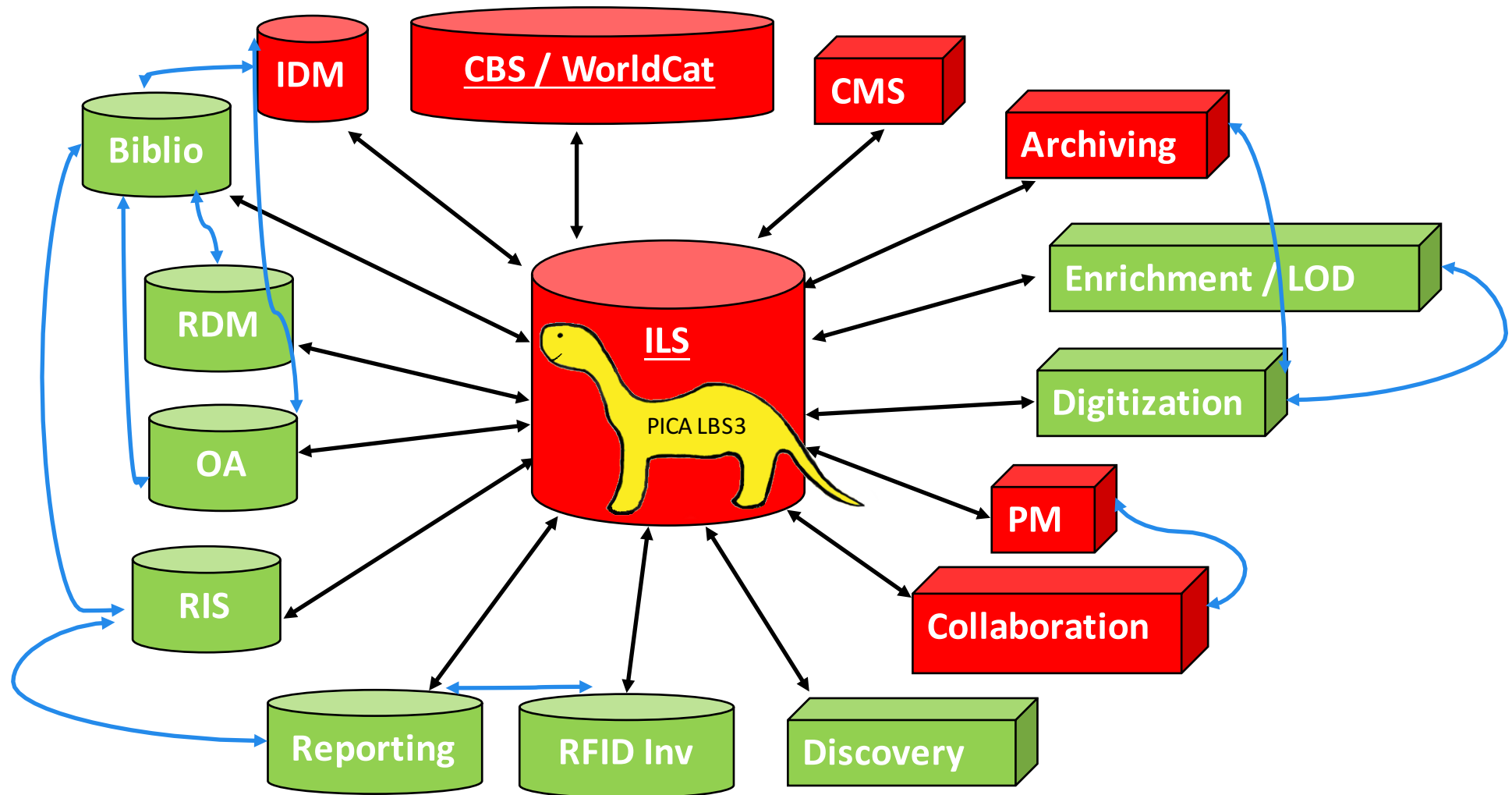
## Open Source

- Eprints
- Dspace
- OJS
- vufind
- MySQL, PostgreSQL, SOLR, Tomcat
- Dwork
- Catmandu
- Open Refine
- Thousands of lines of Perl/PHP/Python/JAVA code

# How integrated are Integrated Library Systems?



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

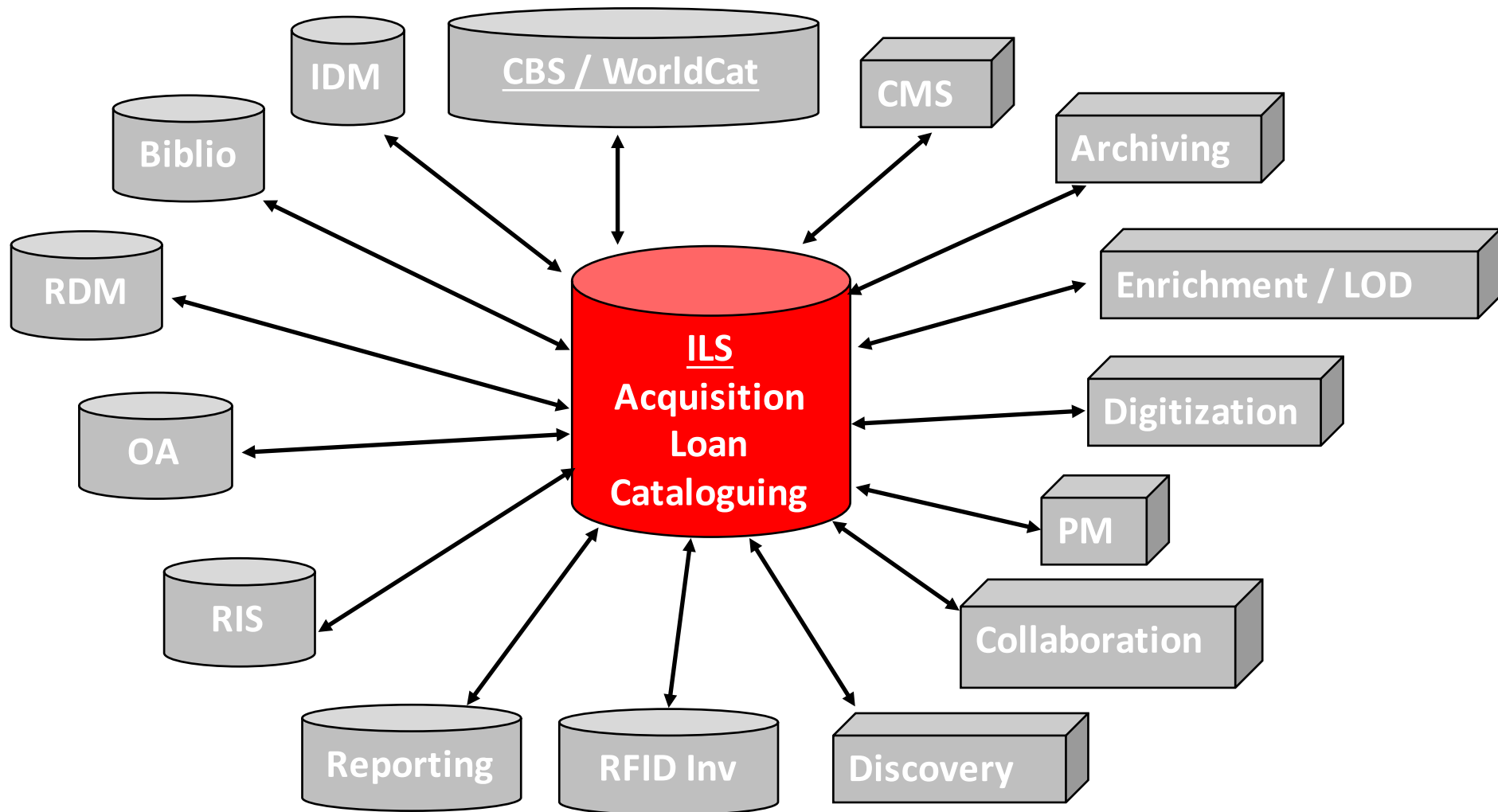


# Where to go?

- Requirements to ILS have changed significantly over the years:
  - Libraries are supposed to offer a lot more digital services than in the past
  - Services have to interconnect to other services on campus, e.g. IDM, Campus Management, Research Information System, Research Data Repository, ...
  - Services should be reusable in other contexts and interoperable
- This is only poorly addressed by commercial ILS products:
  - During the last two years, we had a working group running an assessment of two commercial ILS products. The result was, that none of the systems at their actual state satisfied our (quite basic) requirements.



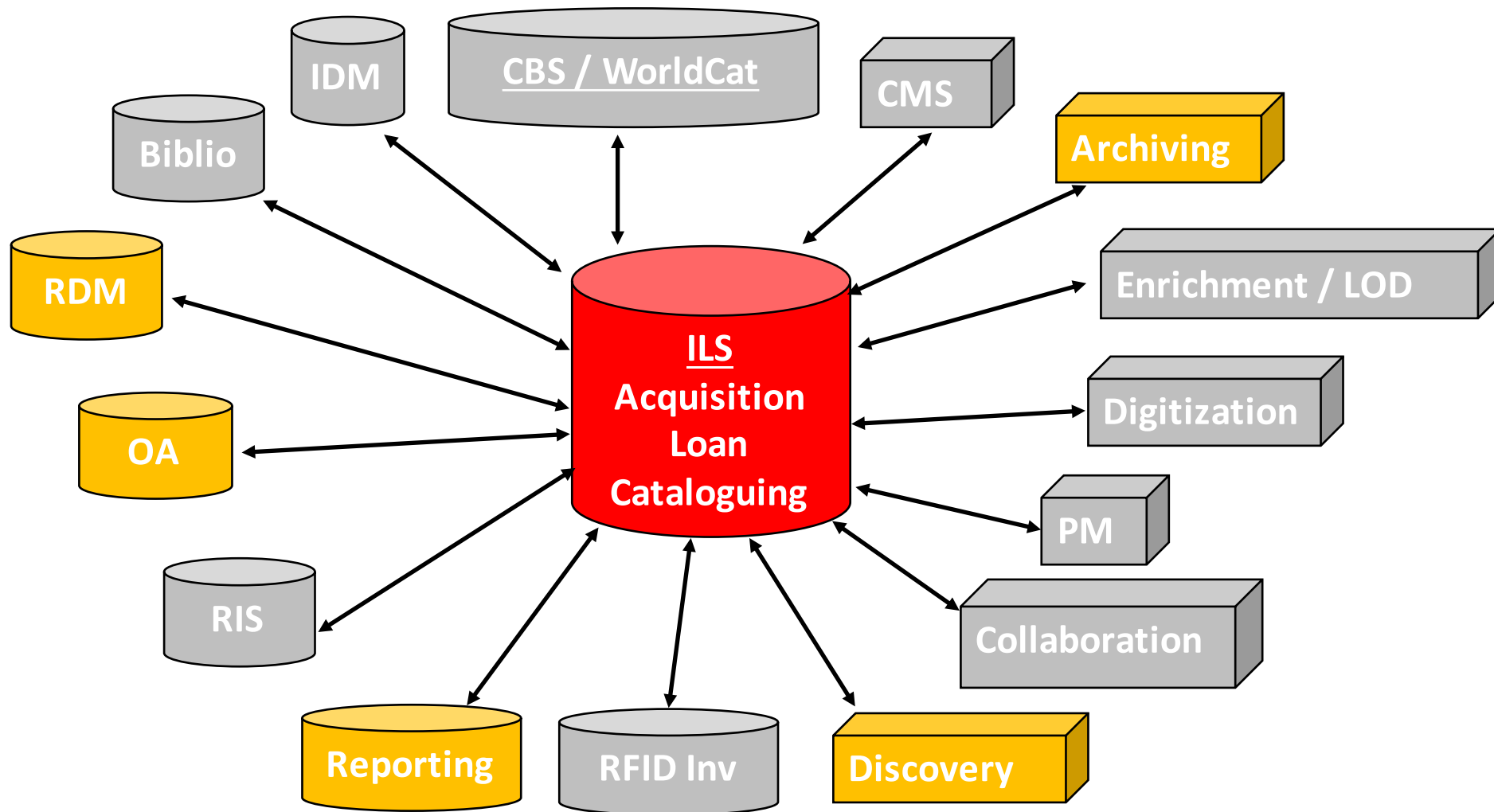
# Conventional Migration: Replace ILS?



# Conventional Migration: Only few modules available, rest has to be connected by us...

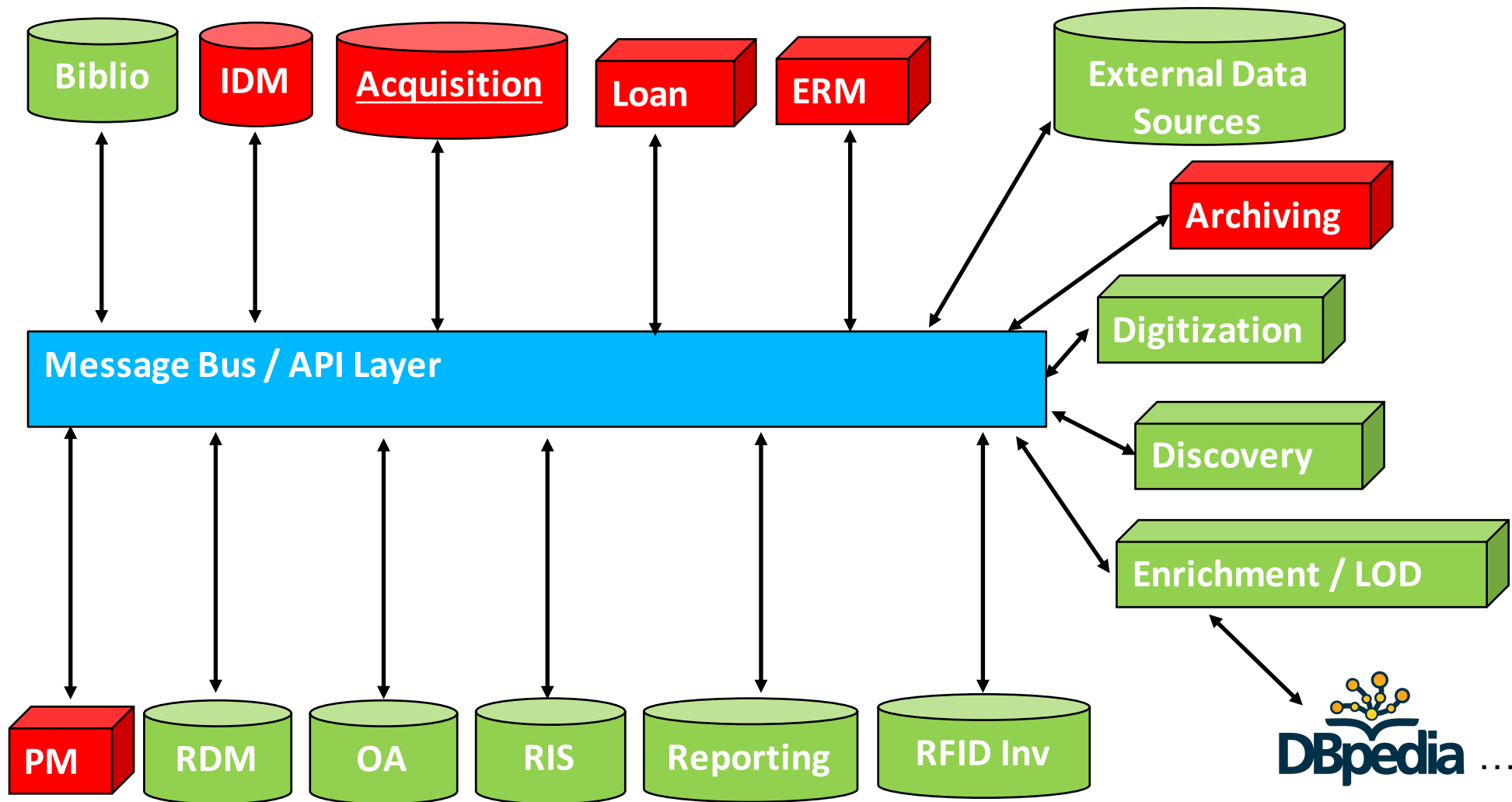


TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

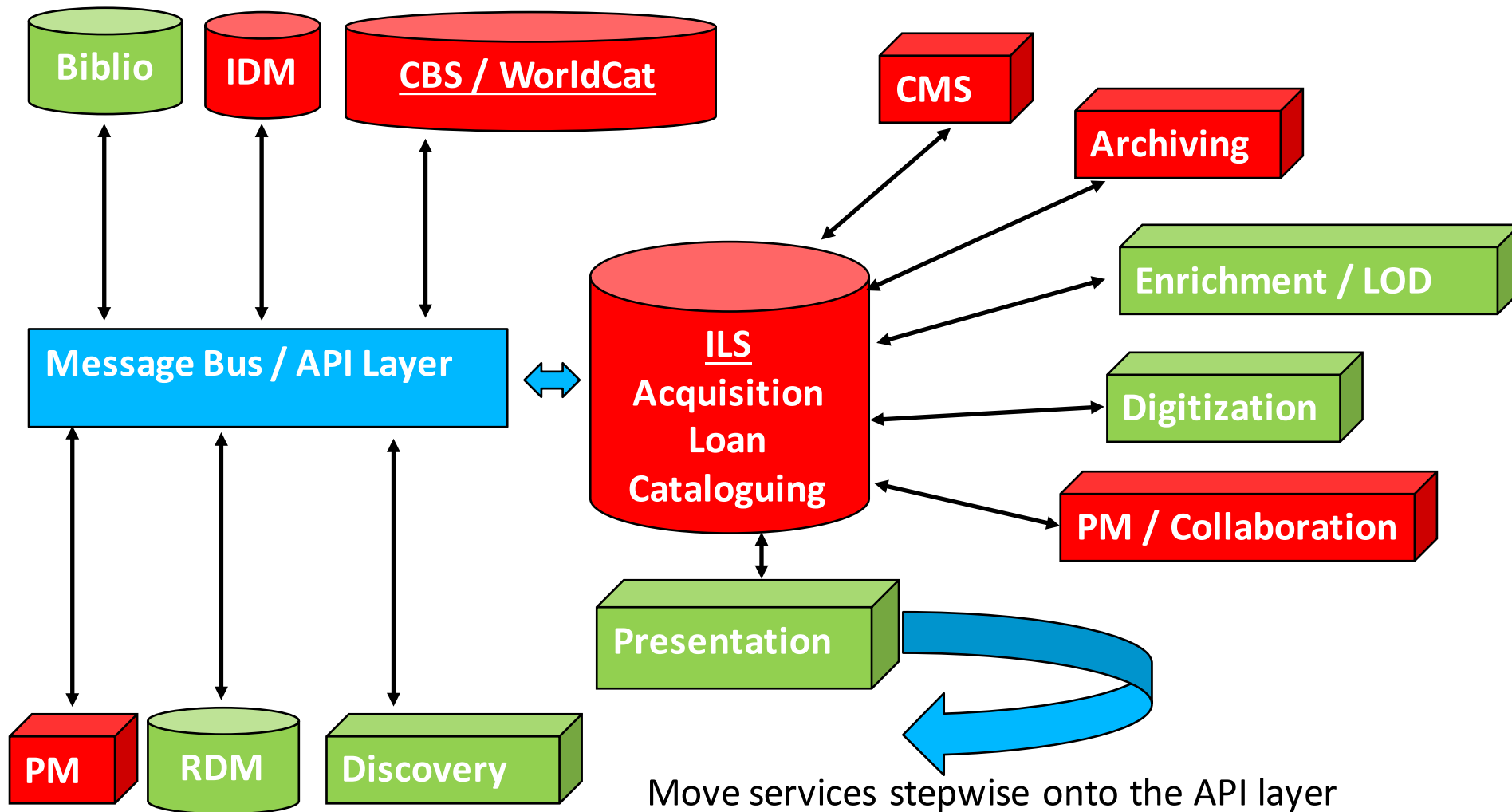




# Alternative Topology: Decentralize, Modulize



# Stepwise Migration of Services



## Part II: Commercial Software

### Benefits of commercial software from a management perspective

- Costs are easy to estimate (are they really?)
- Defined range of functionality is delivered (is it?)
- No trouble with human resources, esp. permanent positions (but also no inhouse knowhow)
- Professional support from the vendor (sometimes)

### Difficulties you may run into

- NDAs: you are not allowed to talk about difficulties with other customers
- Restricted functionality in some areas ( → workarounds necessary)
- Your requirements may not be fully addressed by the vendor ( → workarounds)
- Vendor support cannot / doesn't help you ( → workarounds)
- Product portfolio changes / products are sold or discontinued / new products are not downward compatible / product costs twice the price next year

# Examples from the last two years with commercial software vendors

- **Project A:** Large ILS software evaluation project was cancelled after several months, because the system was not operational to run a large library
- **Project B:** The advertised and offered product was not developed at all, because there was no market for it (after funds from the university have been acquired)
- **Project C:** The project was finished 1.5 years later than planned, because the project manager changed 3 times during the project
- **Project D:** The project is still not finished after 2 years delay because of technical issues
- **Project E:** In his requirements analysis, the vendor states 50 open questions and problems for our specific library, but no solutions. This happened in the pre-sales phase.
- tbc ...





How the customer explained it



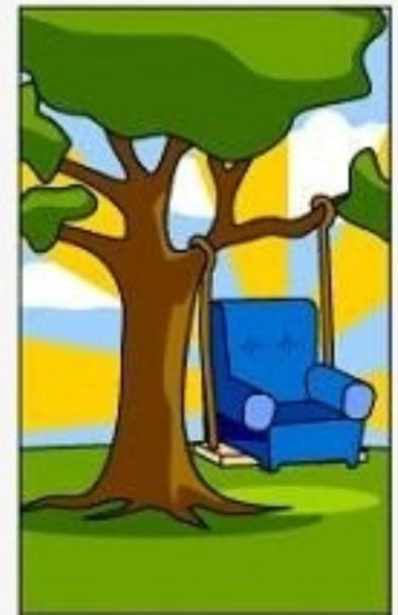
How the Project Leader understood it



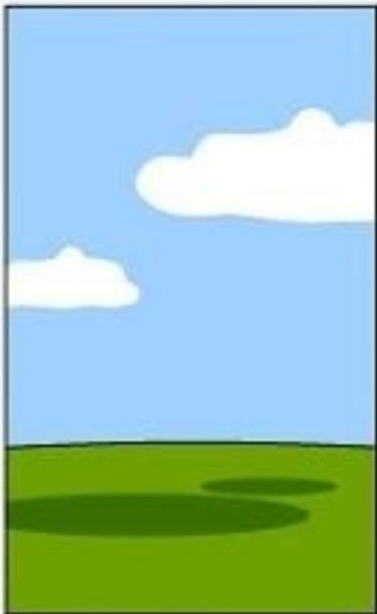
How the System Analyst designed it



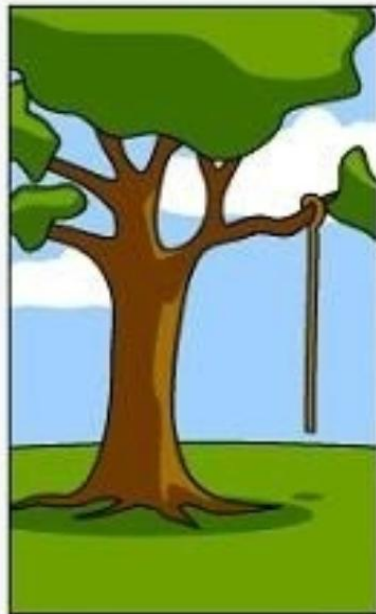
How the Programmer wrote it



How the Business Consultant described it



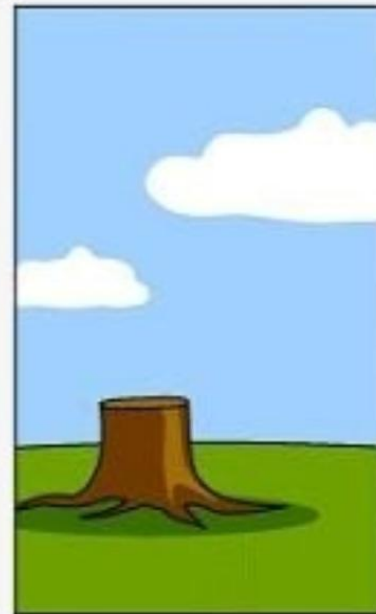
How the project was documented



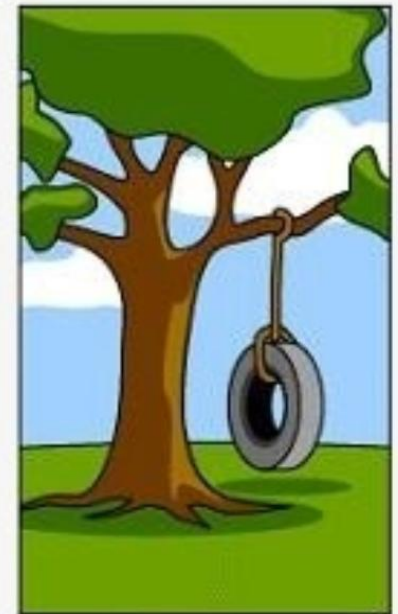
What operations installed



How the customer was billed



How it was supported



What the customer really needed



# Possible Consequences

No warranty that the project will end up successfully. If not, then:

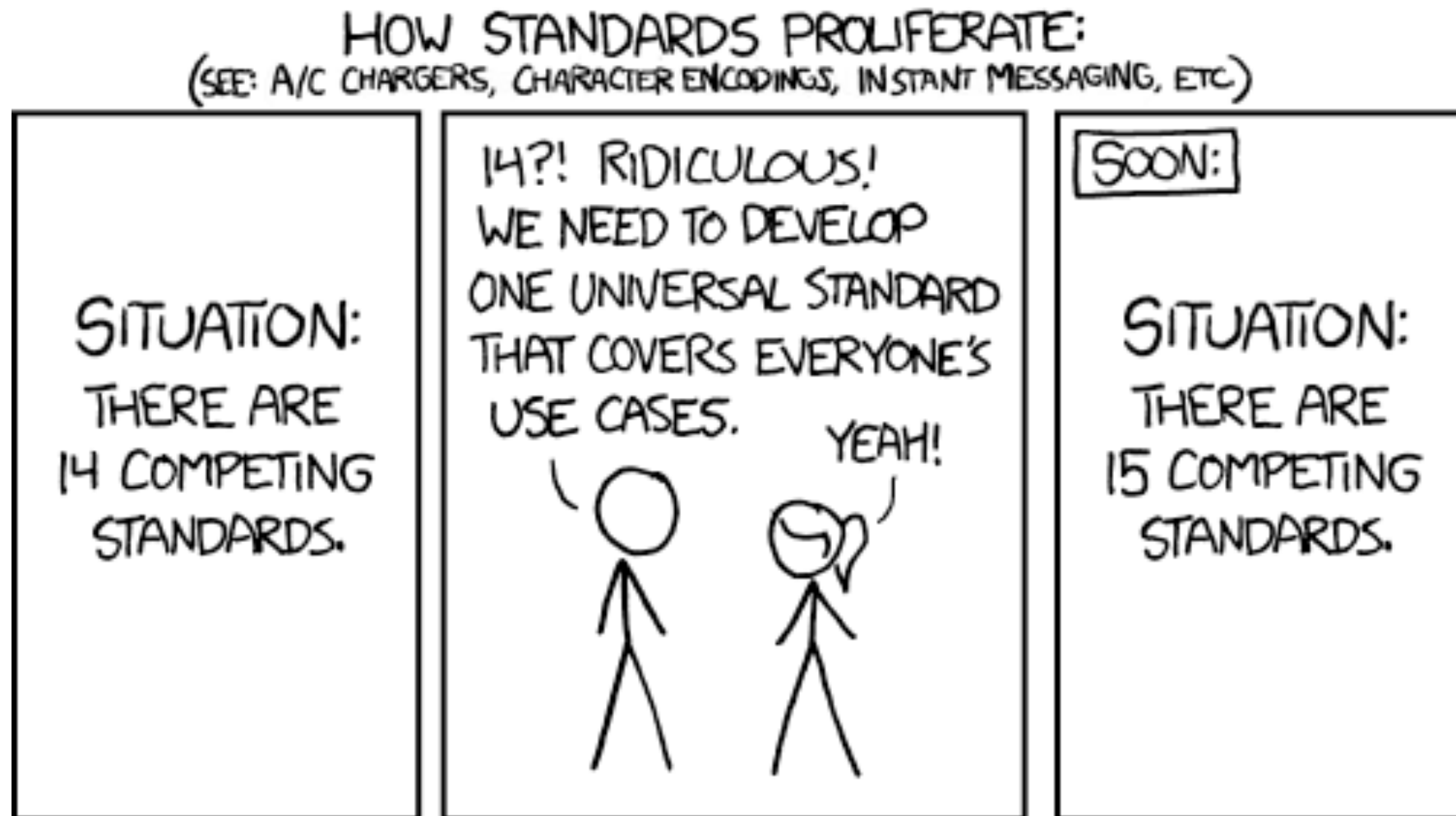
- You may lose time
- You may lose money
- You may lose both
- You may have a piece of software that features only half of the requirements, and you have to improvise the rest by yourself in an environment you are not familiar with
- Solution of problems shifts from the project layer to the management layer. The problem cannot be solved directly
- **All of this doesn't make you happy**







# What about DIY?



<https://xkcd.com/927/>

# DIY Best Practices

- Having questions is naturally a good thing
- Getting involved is also a good thing
- Taking responsibility might be a good thing (could be also seen as a drawback)

## What else?

- Have a look at existing solutions / prevent isolated solutions
- Talk to people who use and / or develop that software in a productive context
- Join / found an open source software project to be not alone
- Consider existing standards
- A community network is important (e.g. national consortia)

**Also no warranty that the project will end up successfully, but you are in charge and you can act.**





# The Future of (De-)Blending?

- Plausible Architecture (API, microservices, ...)
- Blend of community driven project and commercial players looks promising
- Possibility to get involved at different levels and with different amounts of resources
- Manageable risk

Data Structures

**FOLIO**  
*The Future of Libraries is Open*



---

# Our Consequence: Commitment in FOLIO

---



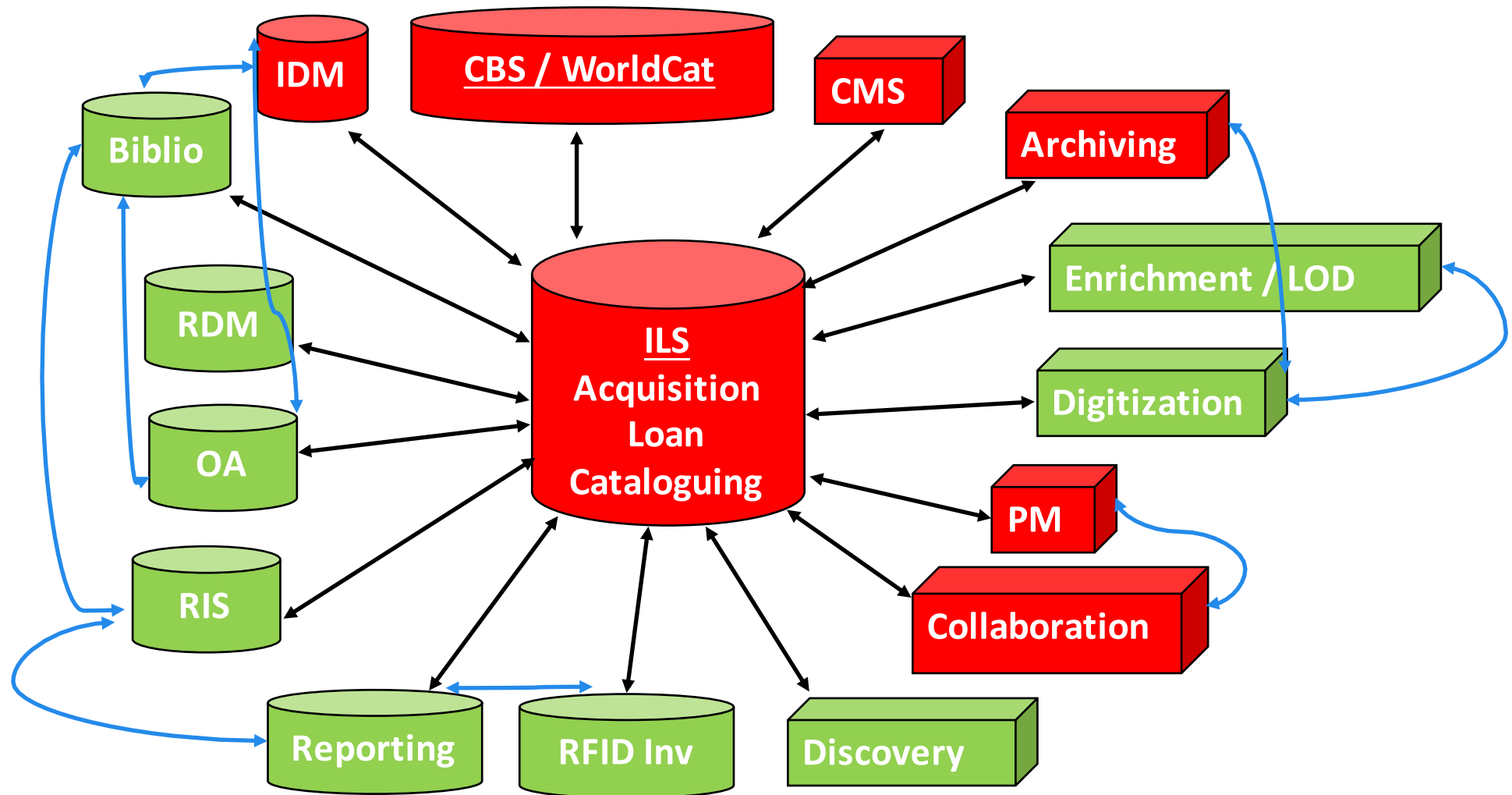
TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- Since March 2018, ULB Darmstadt and another library in Hesse spend around 1 FTE in FOLIO development (ERM project driven by GBV)
- Accession of the Hessian Library Federation to the Open Library Foundation is determined and will take place soon
- Another FTE as project manager / software developer is actually advertised
- All university libraries in Hesse make commitments in Special Interest Groups
- Commitment is fixed for one year, after that evaluated, and hopefully continued ...

# Aspects to convince management

- You have to do development anyways (either workarounds or proper software)
- Competition is good for business: there should be at least three different options to avoid a monopoly
- What about a TCO estimate? (no one does, at least 5-10 years should be standard)
- Attraction of external project funding may be possible, at least temporarily
- In a decentralized architecture, there are more options to get services developed (OSS / commercial modules, DIY, 3<sup>rd</sup> party commissioned work, ...)
- Human resources development can take place from the beginning of the project (requirements analysis, implementation, testing, usage, ...)
- Active Development of digital literacy is crucial for today's libraries' staff
- Stay as long as possible independent from vendor lock-ins in your decision making process

# We already did that in the past!





# Conclusion

- There is no entirely happy solution off the shelf for future library systems
- You have to get involved anyways regardless which way you choose
- Decentralized architectures are more future-proof for scientific digital library business than monolithic systems
- By participating a community project, you strengthen the community and your departments expertise at the same time
- Involvement can be positively used to develop human resources towards more digital literacy around the library staff

**Taking part in the cooking process is  
more attractive than just eating**

