German in 7 Million Shared Objects

Jan Schümmer, intelligent views

Darmstadt, Germany

j.schuemmer@i-views.de

Outline

Background

- intelligent views & K-Infinity
- BI & Duden

K-Infinity as a production environment

- demo

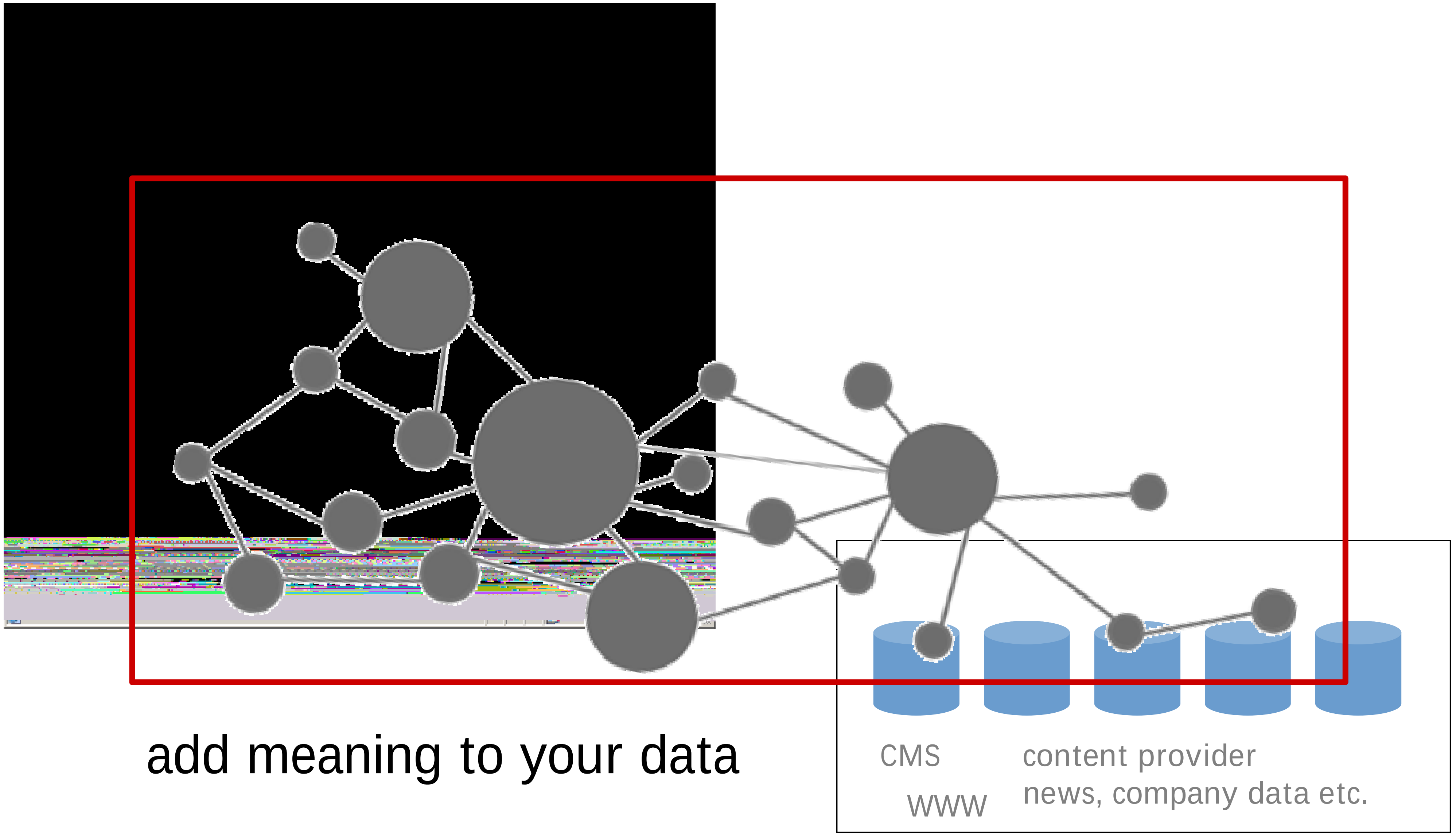
How we did it

- architecture
- scale-up of the standard product
- Improvements in COAST

intelligent views

- located in Darmstadt, Gemany
- founded in 1997
- spin-off enterprise of former GMD – National Research Center for Information Technology, institute IPSI

What we are doing



add meaning to your data

K-Infinity tool suite

Knowledge Builder

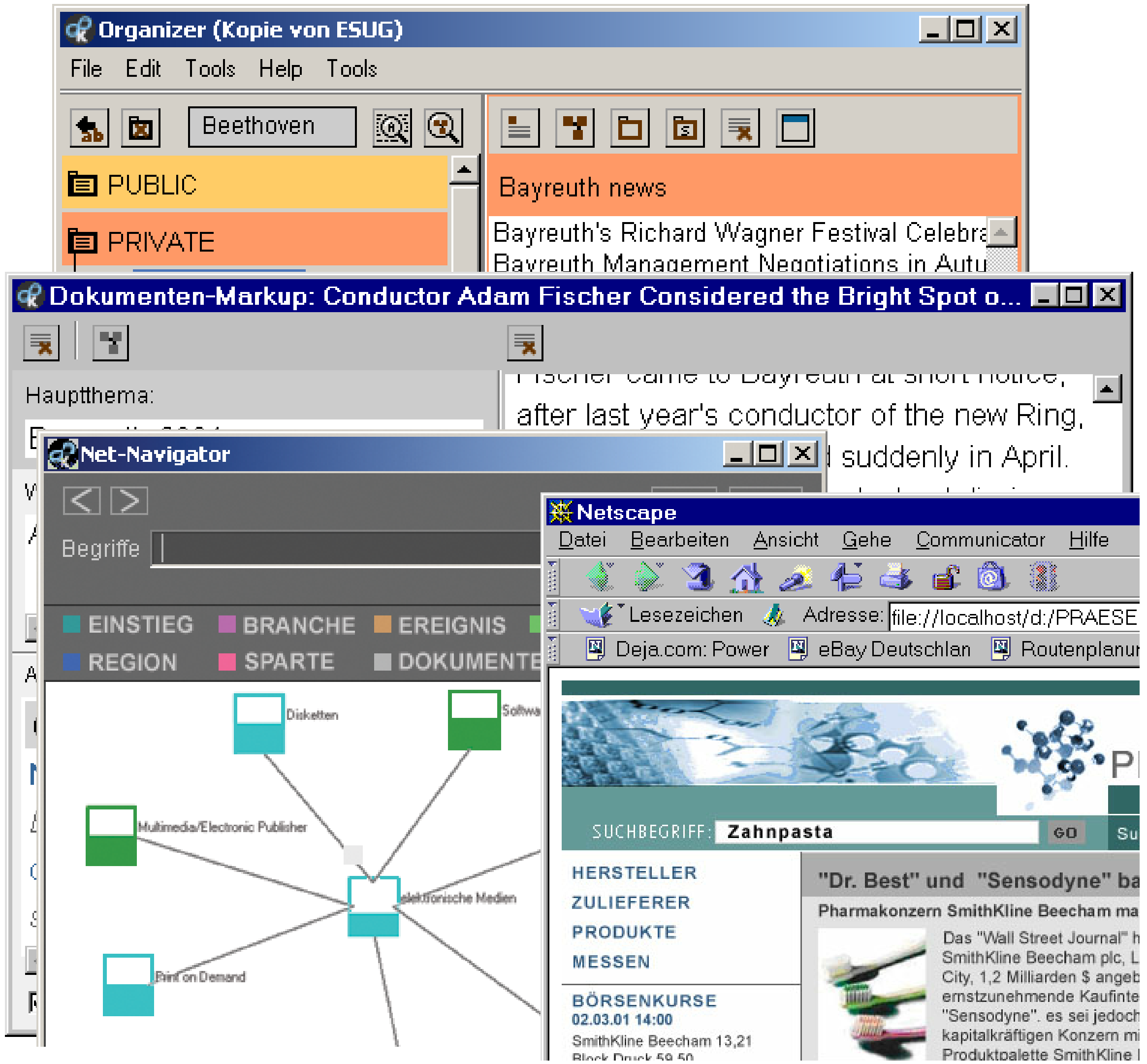
- schema definition

Markup Tool

- link documents to the net

Usage tools (Java)

- Net Navigator
- Knowledge Accelerator
- Web presentation engines



Bayreuth's Richard Wagner Festival Celebra
Bayreuth Management Negotiations in Autu

after last year's conductor of the new Ring,
suddenly in April.

SmithKline Beecham 13,21
Block Druck 59,50

"Dr. Best" und "Sensodyne" ba
Pharmakonzern SmithKline Beecham ma
Das "Wall Street Journal" h
SmithKline Beecham plc, L
City, 1,2 Milliarden $ angeb
ernstzunehmende Kaufinte
"Sensodyne". es sei jedoch
kapitalkräftigen Konzern mi
Produktpalette SmithKline

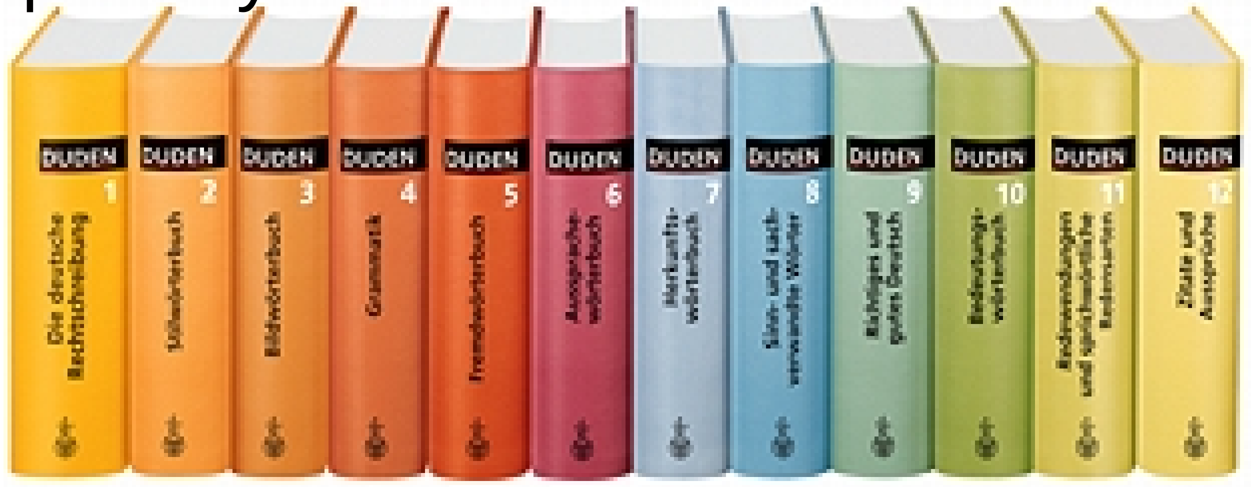
our customer

publisher of the „Duden“ dictionaries, encyclopaedias etc

several „Duden“ products

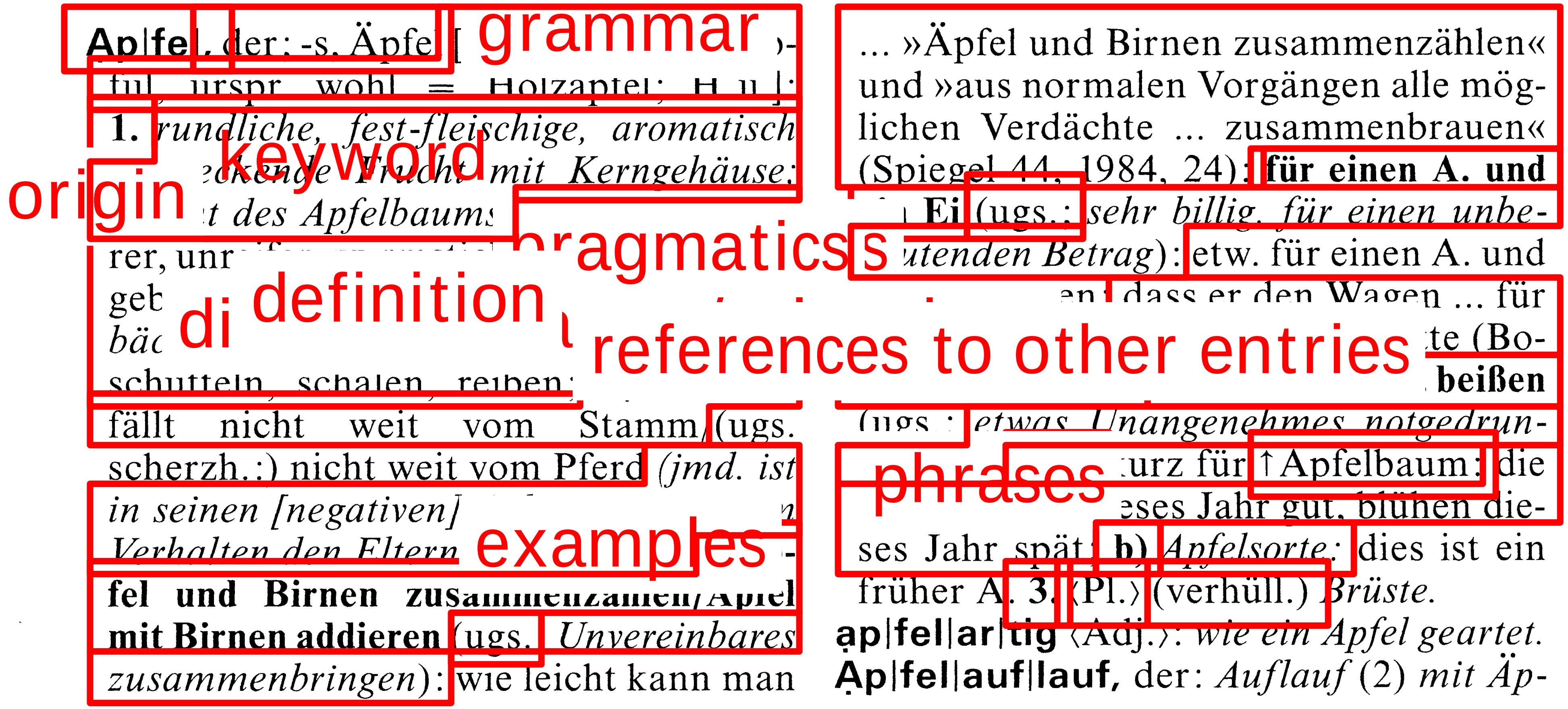
- German language (10 volumes)
- standard dictionary (1 volume)
- specialized dictionaries (etymology, foreign words, sayings, ...)

requirement: single repository for all dictionary publications


DUDEN 1 Die deutsche Rechtschreibung
DUDEN 2 Stilwörterbuch
DUDEN 3 Bildwörterbuch
DUDEN 4 Grammatik
DUDEN 5 Fremdwörterbuch
DUDEN 6 Aussprachewörterbuch
DUDEN 7 Herkunftswörterbuch
DUDEN 8 Sinn- und sachverwandte Wörter
DUDEN 9 Richtiges und gutes Deutsch
DUDEN 10 Bedeutungswörterbuch
DUDEN 11 Redewendungen und sprichwörtliche Redensarten
DUDEN 12 Zitate und Aussprüche

intelligent views

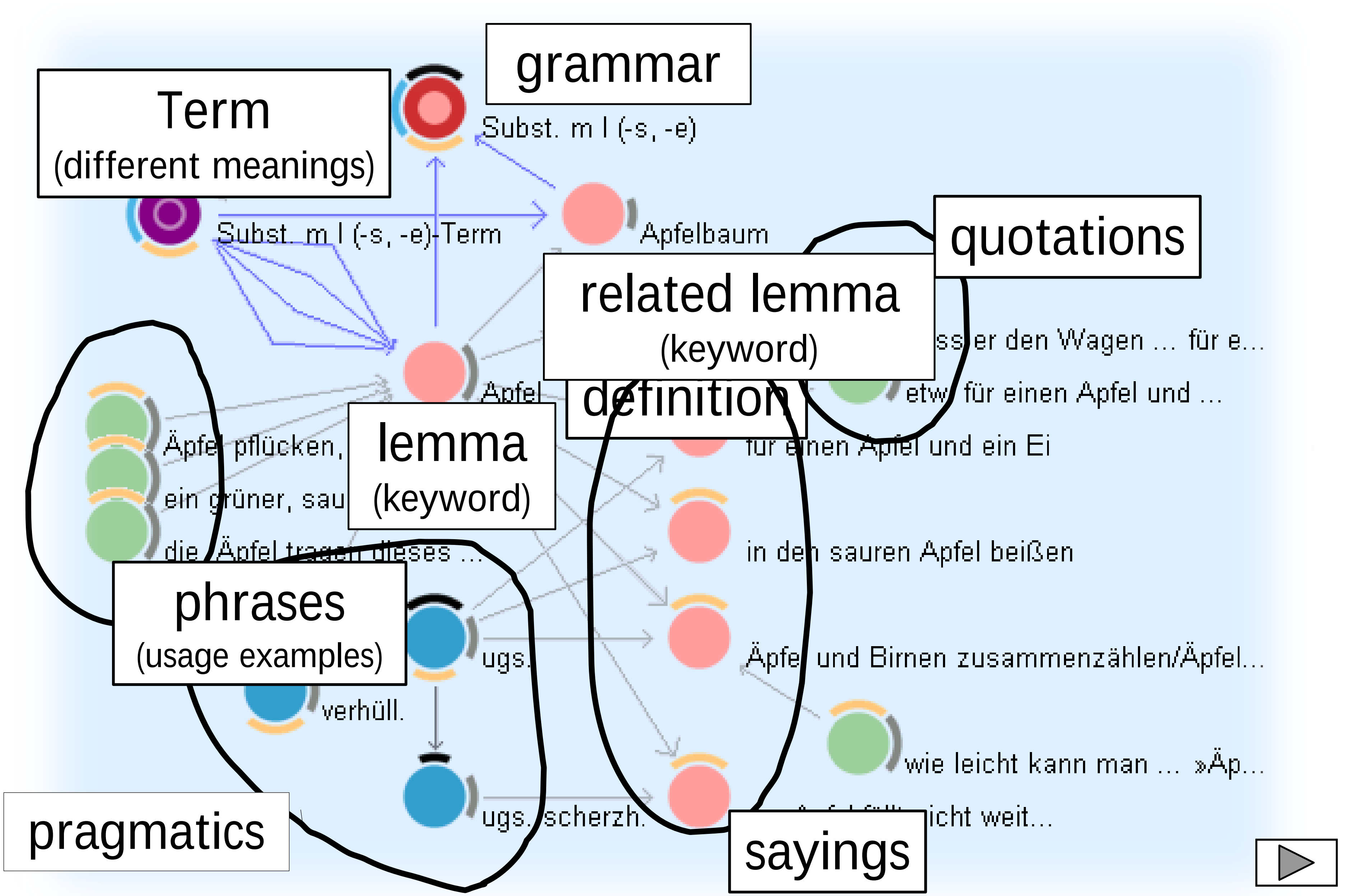
dictionary entries contain ...


grammar
keyword
origin
pragmatics
definition
references to other entries
phrases
examples

Ap|fel, der; -s, Äpfel [
-ful, urspr. wohl = Holzapfel; H. u.]:
1. *rundliche, fest-fleischige, aromatisch*
ckende Frucht mit Kerngehäuse;
t des Apfelbaums
rer, unr
geb
bäc
schütteln, schälen, reiben;
fällt nicht weit vom Stamm (ugs.
scherzh.:) nicht weit vom Pferd *(jmd. ist*
in seinen [negativen]
Verhalten den Eltern
fel und Birnen zusammenzählen/Äpfel
mit Birnen addieren (ugs.: *Unvereinbares*
zusammenbringen): wie leicht kann man

... »Äpfel und Birnen zusammenzählen«
und »aus normalen Vorgängen alle mög-
lichen Verdächte ... zusammenbrauen«
(Spiegel 44, 1984, 24); **für einen A. und**
Ei (ugs.; *sehr billig, für einen unbe-*
utenden Betrag): etw. für einen A. und
en; dass er den Wagen ... für
te (Bo-
beißen
(ugs.: *etwas Unangenehmes notgedrun-*
urz für ↑Apfelbaum: die
eses Jahr gut, blühen die-
ses Jahr spät; **b)** *Apfelsorte:* dies ist ein
früher A. **3.** (Pl.) (verhüll.) *Brüste.*
ap|fel|ar|tig ‹Adj.›: *wie ein Apfel geartet.*
Ap|fel|auf|lauf, der: *Auflauf* (2) *mit Äp-*

knowledge model



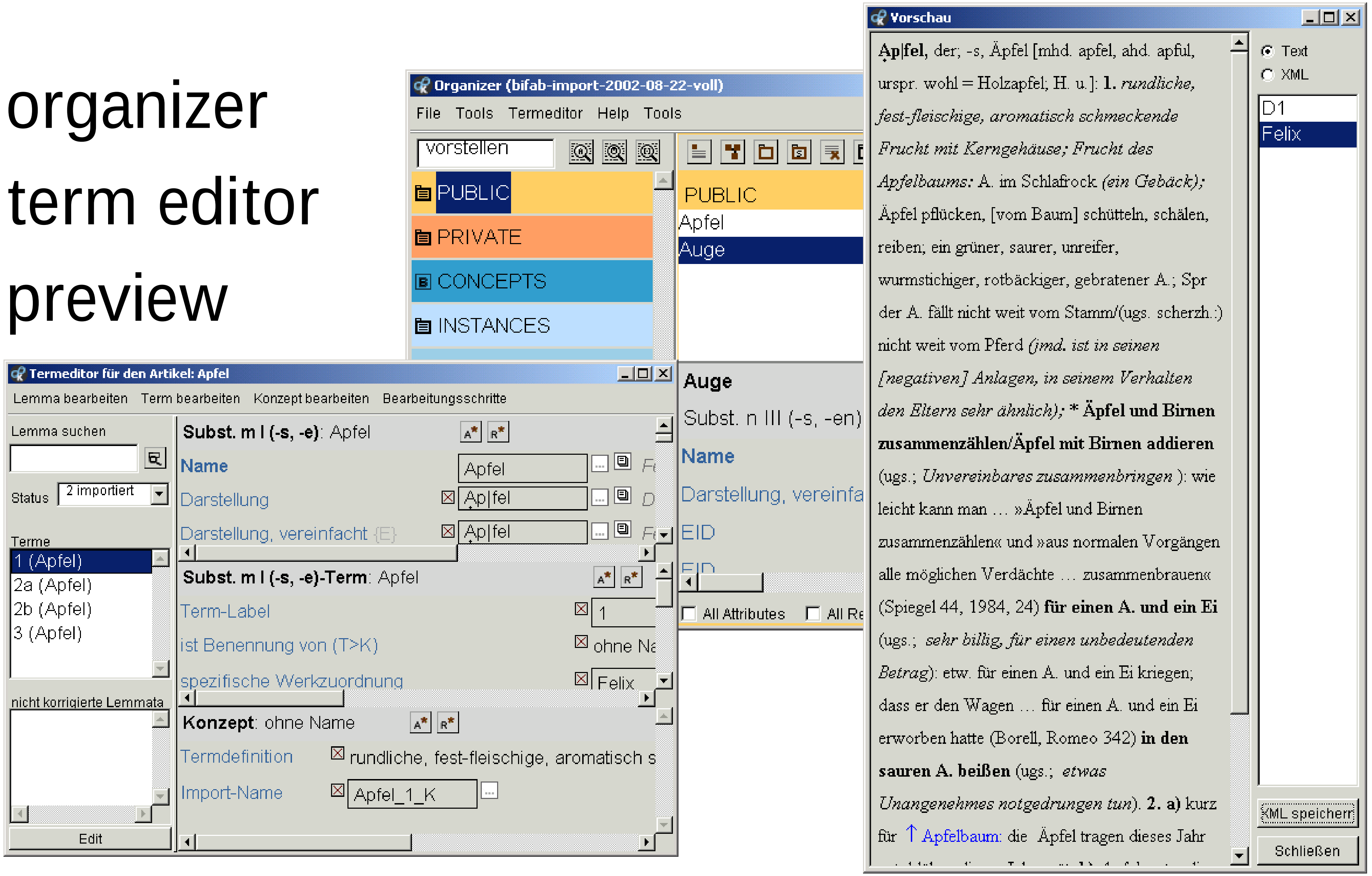
demo

intelligent views

organizer

term editor

preview



Organizer (bifab-import-2002-08-22-voll)

File Tools Termeditor Help Tools

vorstellen

PUBLIC
PRIVATE
CONCEPTS
INSTANCES

PUBLIC
Apfel
Auge

Auge
Subst. n III (-s, -en)
Name
Darstellung, vereinfa
EID
All Attributes All Re

Termeditor für den Artikel: Apfel

Lemma bearbeiten Term bearbeiten Konzept bearbeiten Bearbeitungsschritte

Lemma suchen

Status: 2 importiert

Terme
1 (Apfel)
2a (Apfel)
2b (Apfel)
3 (Apfel)

nicht korrigierte Lemmata

Edit

Subst. m I (-s, -e): Apfel

Name	Apfel
Darstellung	Ap|fel
Darstellung, vereinfacht {E}	Ap|fel

Subst. m I (-s, -e)-Term: Apfel

Term-Label	1
ist Benennung von (T>K)	ohne Na
spezifische Werkzuordnung	Felix

Konzept: ohne Name

Termdefinition	rundliche, fest-fleischige, aromatisch s
Import-Name	Apfel_1_K

Vorschau

Text [✓] XML

D1
Felix

XML speichern
Schließen

Ap|fel, der; -s, Äpfel [mhd. apfel, ahd. apful, urspr. wohl = Holzapfel; H. u.]: **1.** *rundliche, fest-fleischige, aromatisch schmeckende Frucht mit Kerngehäuse; Frucht des Apfelbaums:* A. im Schlafrock *(ein Gebäck);* Äpfel pflücken, [vom Baum] schütteln, schälen, reiben; ein grüner, saurer, unreifer, wurmstichiger, rotbäckiger, gebratener A.; Spr der A. fällt nicht weit vom Stamm/(ugs. scherzh.:) nicht weit vom Pferd *(jmd. ist in seinen [negativen] Anlagen, in seinem Verhalten den Eltern sehr ähnlich);* * **Äpfel und Birnen zusammenzählen/Äpfel mit Birnen addieren** (ugs.; *Unvereinbares zusammenbringen*): wie leicht kann man … »Äpfel und Birnen zusammenzählen« und »aus normalen Vorgängen alle möglichen Verdächte … zusammenbrauen« (Spiegel 44, 1984, 24) **für einen A. und ein Ei** (ugs.; *sehr billig, für einen unbedeutenden Betrag*): etw. für einen A. und ein Ei kriegen; dass er den Wagen … für einen A. und ein Ei erworben hatte (Borell, Romeo 342) **in den sauren A. beißen** (ugs.; *etwas Unangenehmes notgedrungen tun*). **2. a)** kurz für ↑ Apfelbaum: die Äpfel tragen dieses Jahr

need to scale up

211839 keywords (lemmas)

248965 meanings (terms)

± 250000 definitions

234826 quotations and examples

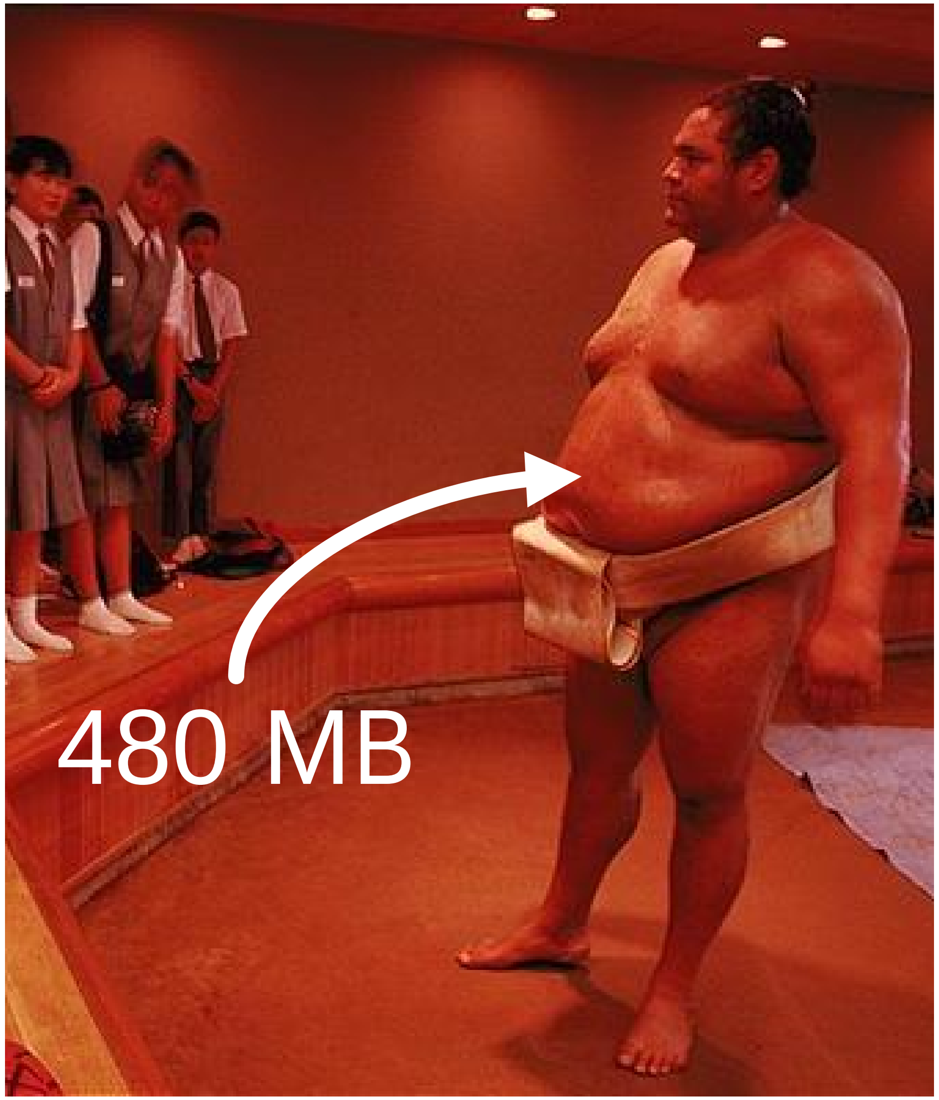
17 average attributes & relations per lemma with terms

4 average attributes & relations per quotation

→ calculates to 5 998 000 shared objects

→ actual: 8 177 364 (incl. Index)

→ far too much for K-Infinity in 2001



what we did

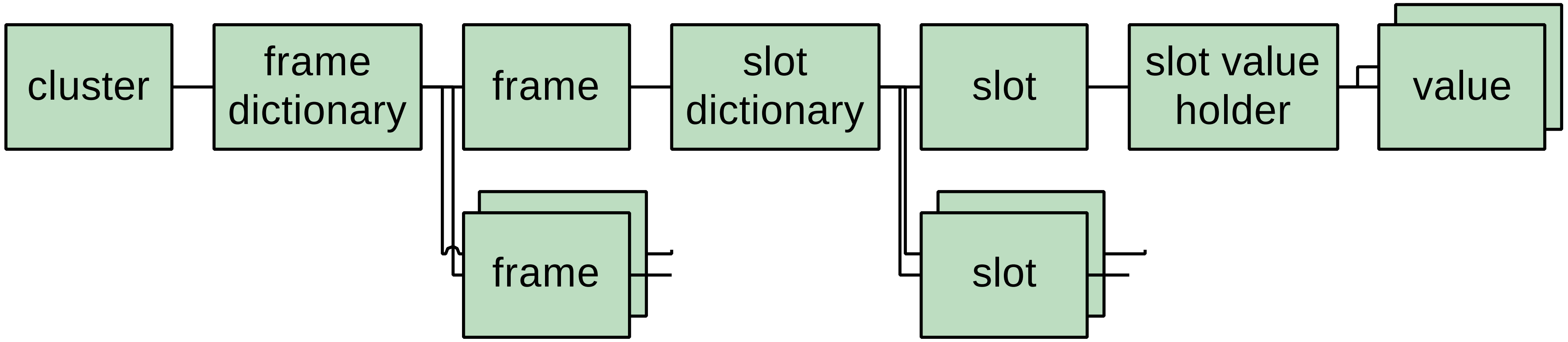
measure first

- there always is a screw to turn
- memory usage and performance are related
- have a look at the allocation profiler when time profiling doesn‘t show results

Slim down: Object loading (1)

only load into memory what you really need

- model objects are organized in clusters
- clusters are the smallest unit that can be requested from the central server (mediator)
- COAST 1.0
  - whole clusters are loaded on demand



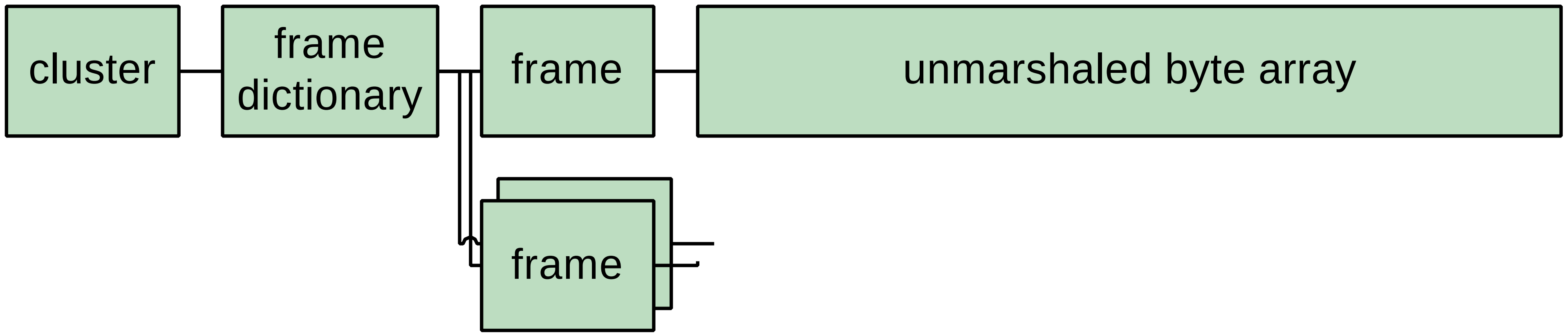
Slim down: Object loading (2)

first commercial project

- delay unmarshaling of structures until they are accessed (lazy unmarshaling)
- keep persistent representation in ByteArrays

→ reduced number of objects to be created during cluster loading

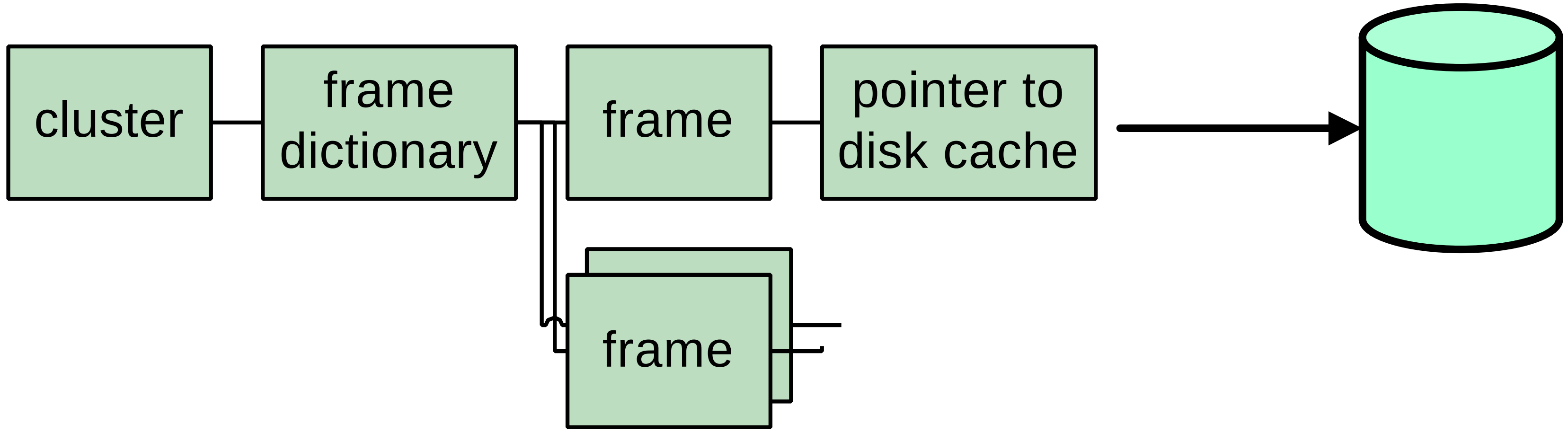
→ untouched objects can be directly written from their ByteArray representation



Slim down: Object loading (3)

latest improvement

- stream clusters to disk cache
- create empty frames for those objects that are not needed at the moment
- lazy unmarshaling/loading from disk cache



object loading: memory footprints

intelligent views

		before		after		
	Class	**instances**	**bytes**	**instances**	**bytes**	**+- byte**
1	ByteArray	275976	22723153	172	96926	-22626227
2	TwoByteString	74554	2655308	74554	2655308	0
3	Array	7710	2597668	7662	2592020	-5648
4	IdentityDictionary	620	2389560	603	2385784	-3776
5	COAST.CatCSFrameLocator	79182	1583640	79182	1583640	0
6	KInfinity.KMarkupAttribute	50759	1421252	50759	1421252	0
14	IdentitySet	14517	555636	14523	555852	216
15	KInfinity.BIFABChangeRecord	16696	467488	16696	467488	0
16	COAST.CatFSSlot	9138	328968	9143	329148	180
17	ByteString	1403	54734	1509	55400	666

Slim down at application level

- shortcuts within the model to avoid loading intermediate objects
- re-clustering to have those objects together that are typically needed together
- add index structures to avoid searching
- in general: Be careful when traversing object structures

Slim down: Programming tricks

avoid large ST Dictionaries and Sets, instead

- use more compact data types (e.g. Array)
- teach WeakDictionary to shrink
- consider using IdentityDictionary instead of Dictionary
- get rid of instance variables
- have a look at memory allocation for temporaries:

```
importantObjects := (myLargeCollection collect:
                                   [ :e | e -> e foo1 ])
           select: [ :assoc | assoc key foo2 ].
^ importantObjects anyElement value
```

Lowest level possible

The VisualWorks VM

- bulk load problems in 2001 were finally solved by reducing object allocation and –footprint (lazy unmarshaling)
- specialised MemoryPolicy
  - thanks to John M. McIntosh and ESUG 2001
- hard problems while importing
  - Out of memory in scavange.c (ine 1622)
    - This was not an endless loop
  - Solved by setting FreeMemoryUperBound to max
    - Thanks to Cincom support and Eliot Miranda

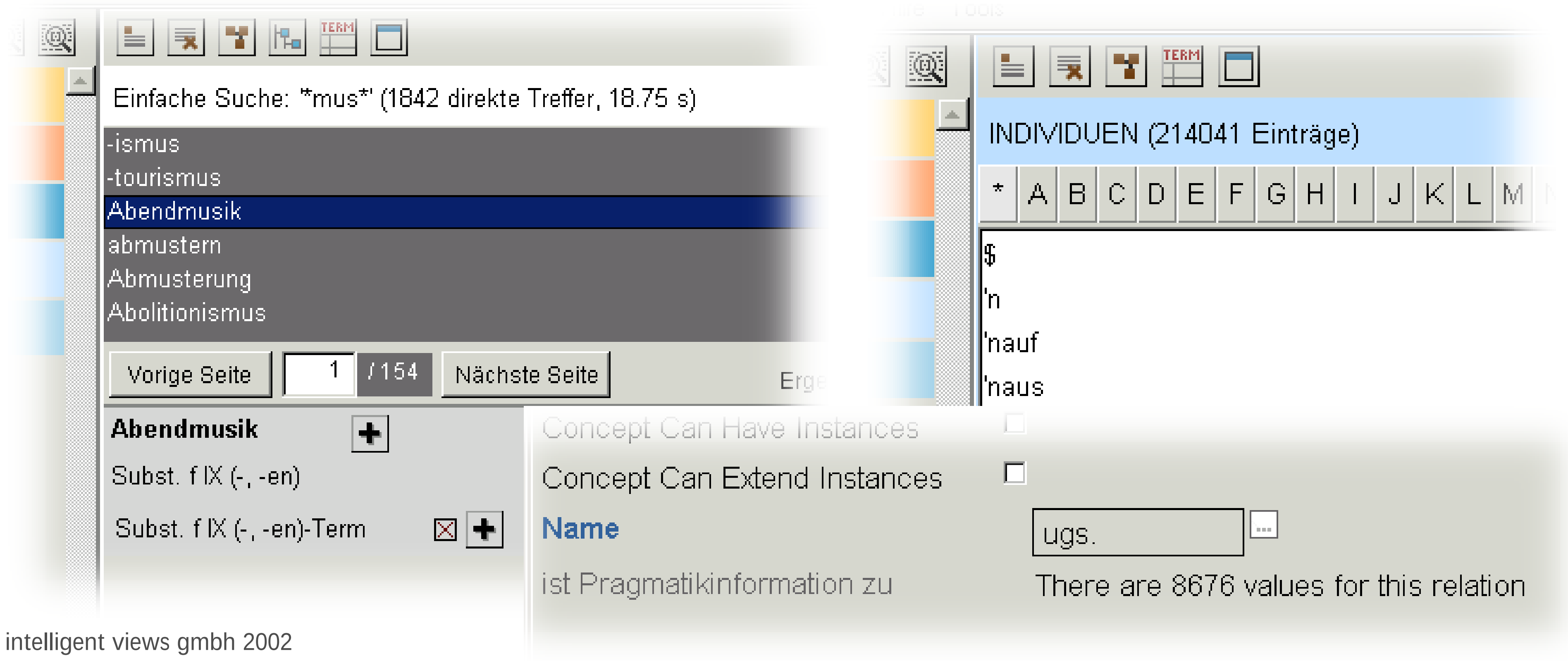
Performance: Get the sumo to dance

performance 1

some user interfaces simply don‘t scale up

- drop down lists
- composite views with many elements

strategy: change the interface without losing functionality



Einfache Suche: "*mus*" (1842 direkte Treffer, 18.75 s)

-ismus
-tourismus
Abendmusik
abmustern
Abmusterung
Abolitionismus

Vorige Seite | 1 | / 154 | Nächste Seite

Abendmusik
Subst. f IX (-, -en)
Subst. f IX (-, -en)-Term

INDIVIDUEN (214041 Einträge)

* A B C D E F G H I J K L M

$
'n
'nauf
'naus

Concept Can Have Instances	☐
Concept Can Extend Instances	☐
Name	ugs.
ist Pragmatikinformation zu	There are 8676 values for this relation

performance 2

even slim sumos get fat when they are fed too much

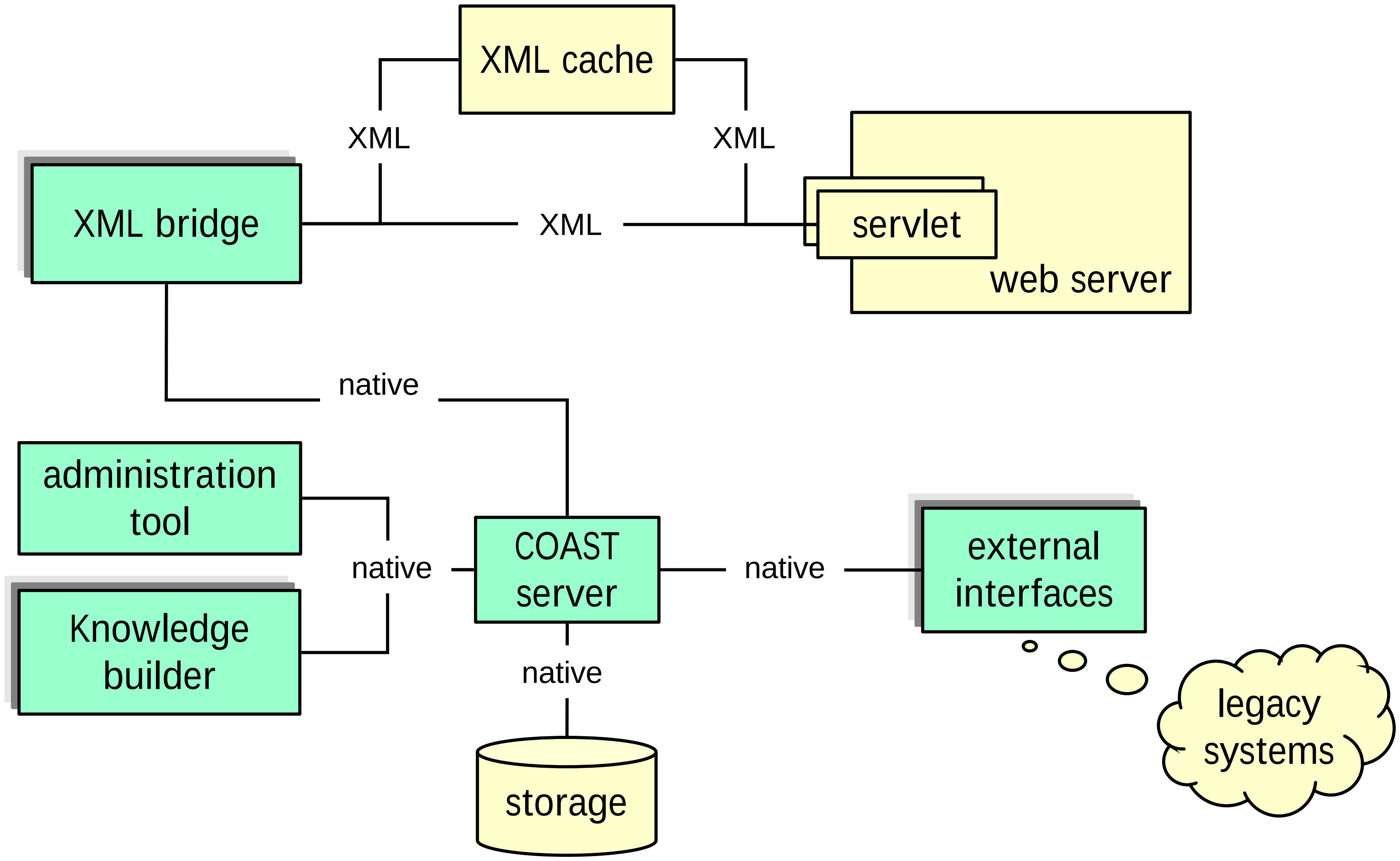
- e.g. structural queries

strategy: let others do the work ;-)

- delegate time/resource consuming tasks to dedicated processes

Behind the Scenes

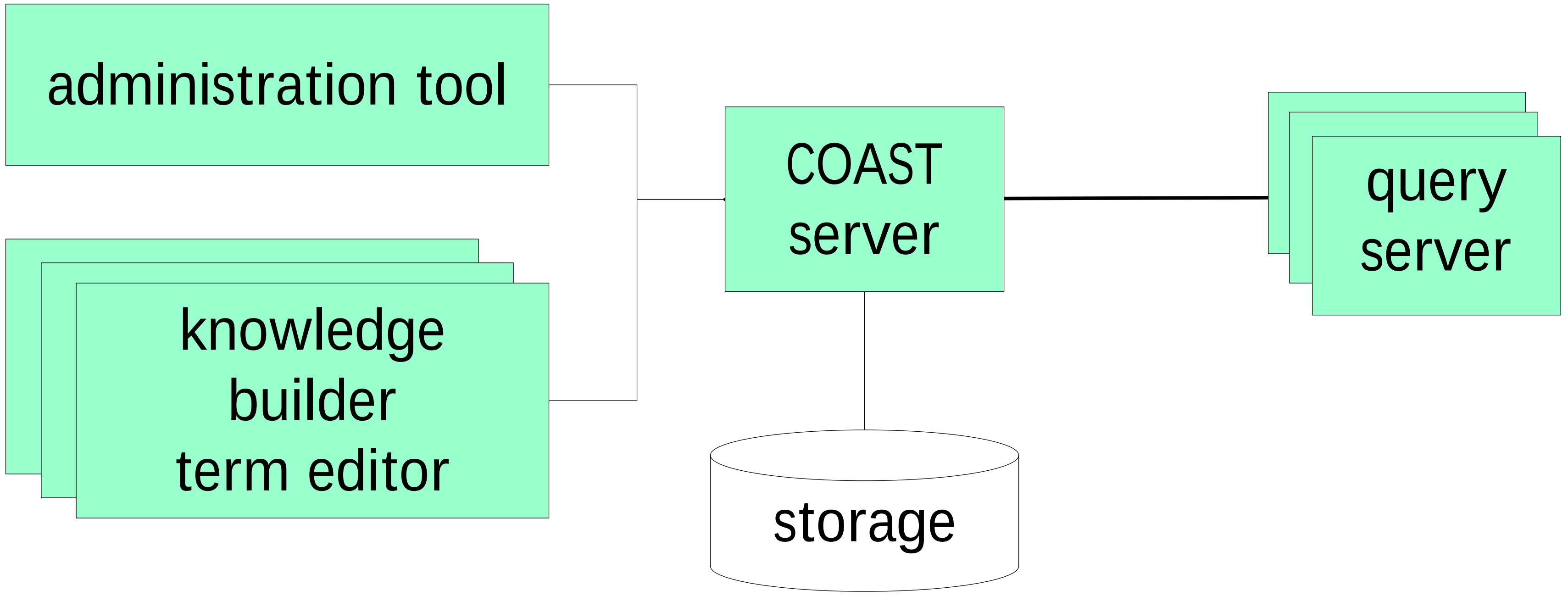
K-Infinity standard architecture



architecture scale-up

fat-client architecture, but time- and/or memory consuming tasks are better performed on on a server

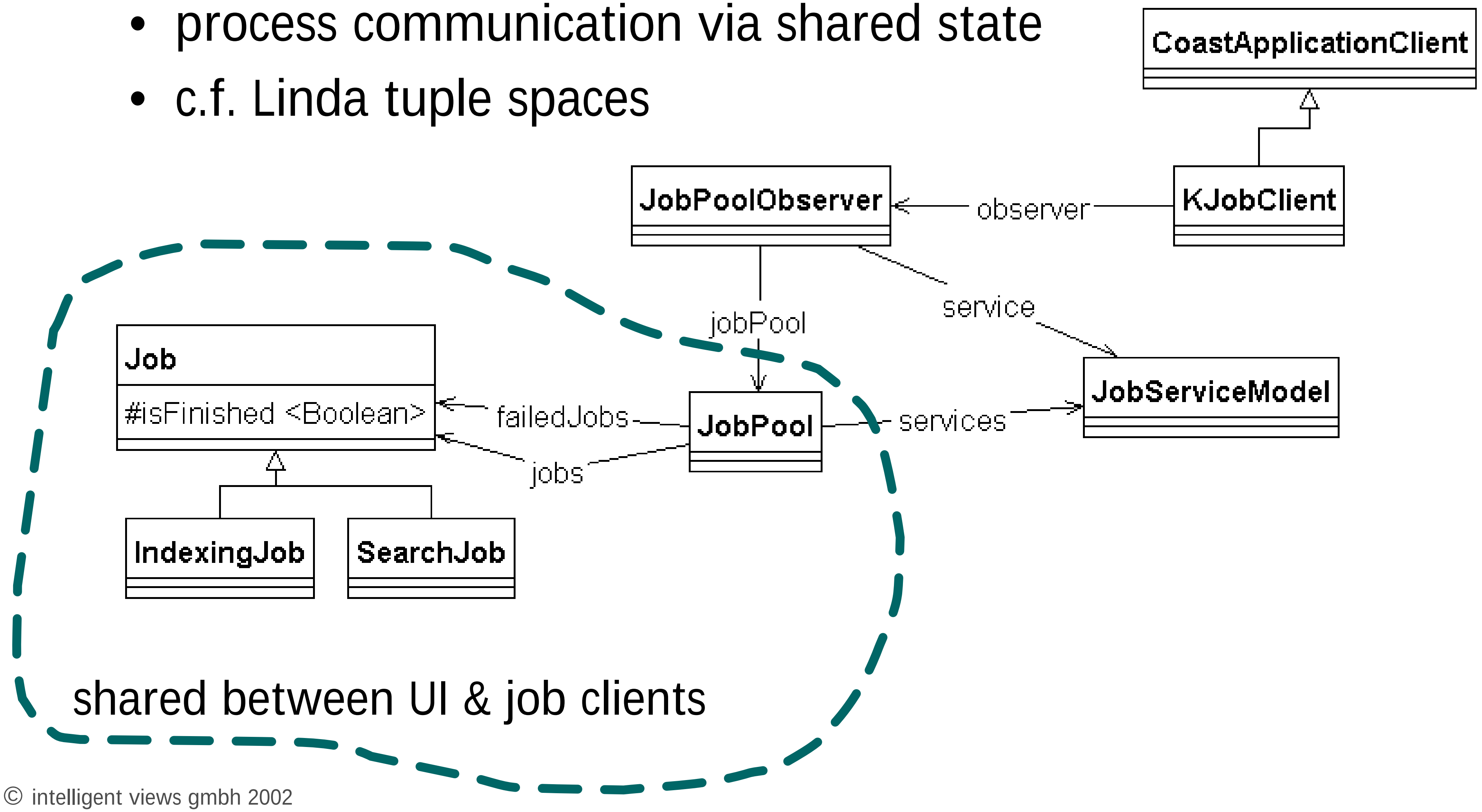
solution: specialised servers for e.g. structural queries, full text index queries



programmer‘s view

Distributed computing with COAST

- process communication via shared state
- c.f. Linda tuple spaces



shared between UI & job clients

programmer‘s view (2)

ui client

- create a (search) job object
- add it to the global job pool
- open a view on the result set

job executor client

- wake up & take job (mark it as #inProgress)

ui client

- Coast view updating visualizes #inProgress

job executor client

- execute job
- write result into some shared object

ui client

- Coast view updating visualizes results

tools & practices

„Take advantage of the tools“

- no magic involved
- allocation profiler
- time profiler
- SystemAnalyzer>>showSystemHistogram

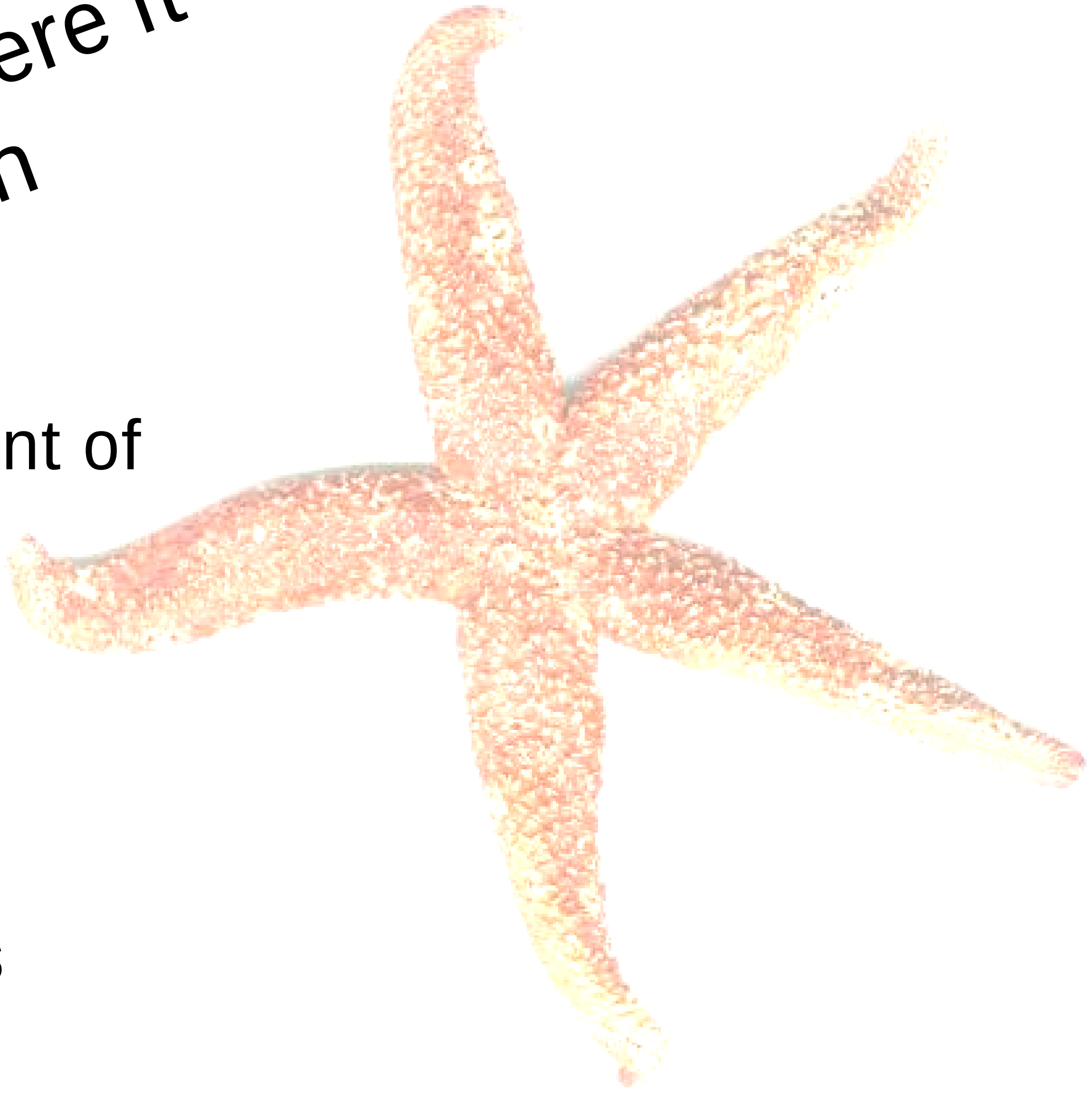
optimisations in underlying layers rather than in the concrete application

- ... application tuning has local effect
- ... low level optimizations let the whole system benefit

get it run – get it right – get it fast

The COAST Framework

Remember where it all began



Support the development of

- object oriented
- synchronous
- interactive, and
- complex

groupware applications

How we use COAST

underlying technology for K-Infinity

- persistence
- transactions & concurrency control
- user interface (automatic view updating)
- modelling (object behavior framework)

COAST evolves towards an OO database

- data volume is still growing
- most features that you would expect from an OODBMS
- we are tuning COAST as well as K-Infinity
  - 12 800 empty transactions/second on this 500 MHz PC

you can use COAST for free

COAST is open source

It is included as a goodie in Cincom‘s VisualWorks 7 release

Feedback, usage experiences, and contributions are highly welcomed

further information

www.i-views.de

www.opencoast.org