LILY

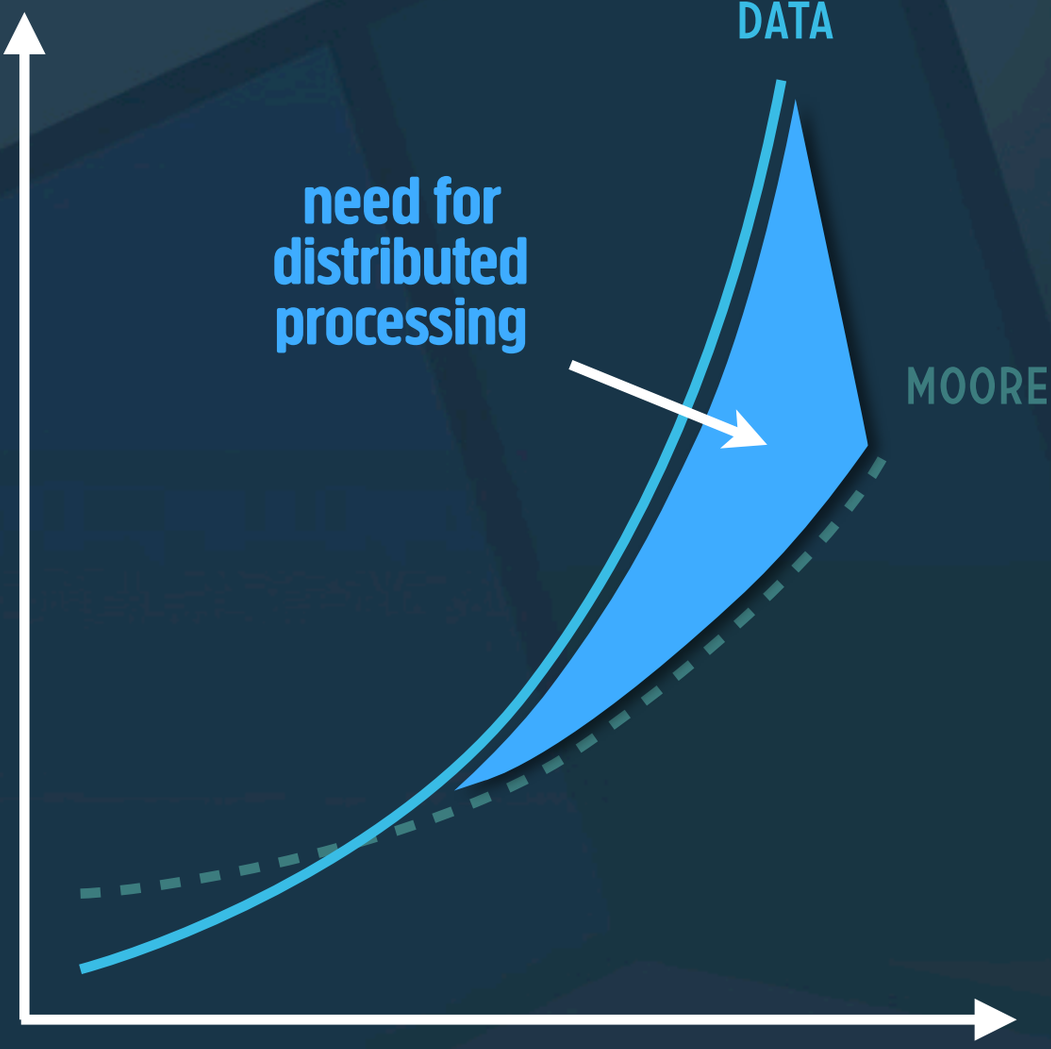
SMART DATA,
AT SCALE
MADE *easy*

FROM CONTENT STORAGE TO SCALING SMART DATA

OUTER THOUGHT

IIC » TECHNOLOGIEPARK 3 » B-9052 ZWIJNAARDE (GENT) » www.outerthought.org

THE PAIN



THE PAIN

- growth of data sets
- smart businesses need to apply analytics to activities
- doing business online means real-time
- talent shortage

SMART DATA, AT SCALE MADE *easy*

LILY

The Real-time Platform built for the Age of Data.

We manage, track and measure your data and users, and do the mat(c)hmaking in-between:

- provide you with business intelligence and analytics
- harvest user profiles and learn their interests
- dynamically engage your users using quality recommendations

WHERE WOULD YOU USE LILY?

- large collections of data
  - content repositories
  - library catalogs
  - (media) asset management
  - product catalogs
  - ‘live’ archives
- large groups of users
  - e-commerce / retail
  - news / media

- ... if you want to use big data, but you need easy.

this is where the magic happens

+

BEYOND CONTENT MANAGEMENT

marketing

revenue

product / service

BEYOND CONTENT MANAGEMENT: DATA + ANALYTICS

recommendations

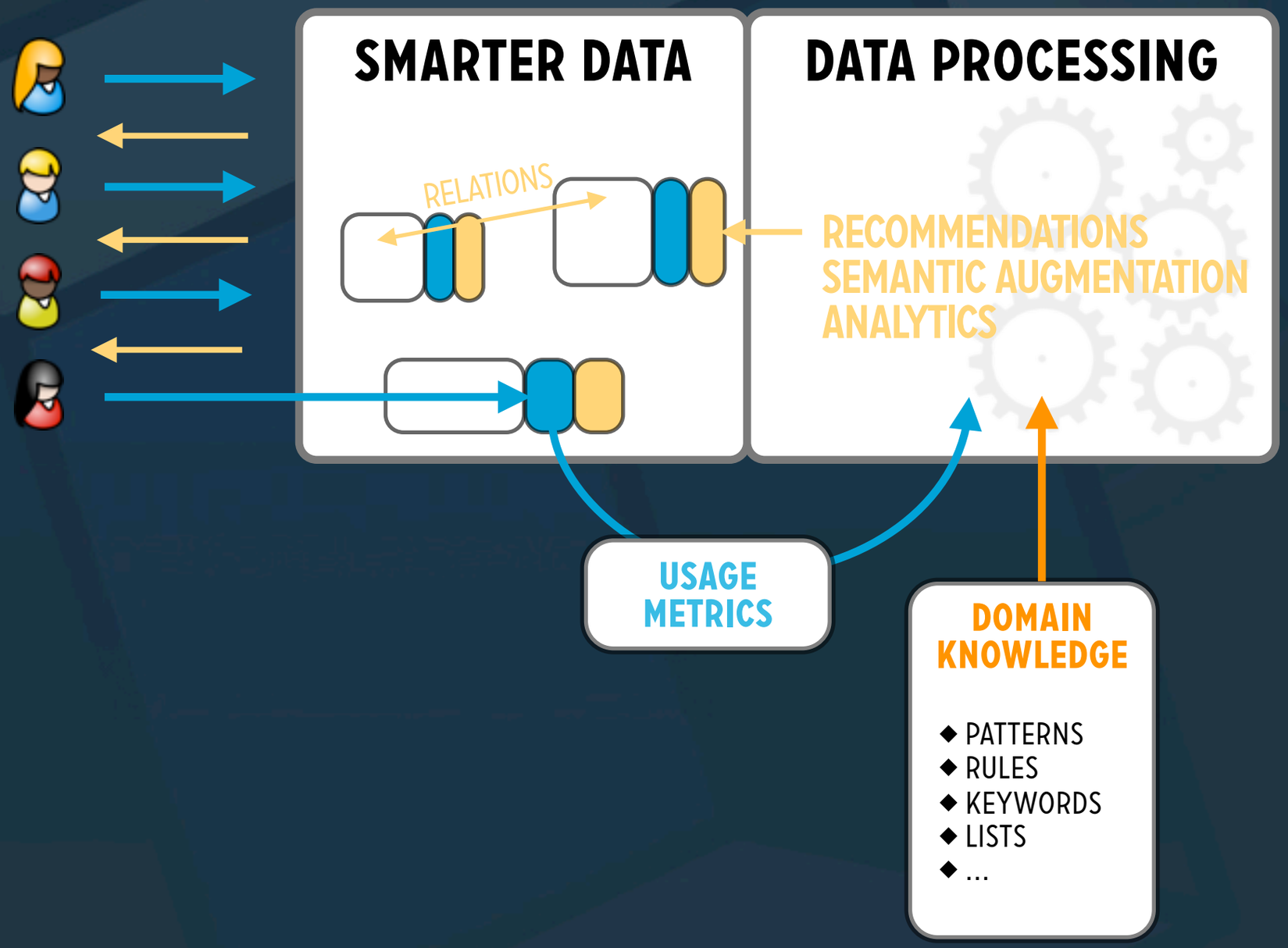
personalised

revenue

product / service

audience data

LILY 2.0: SMART DATA

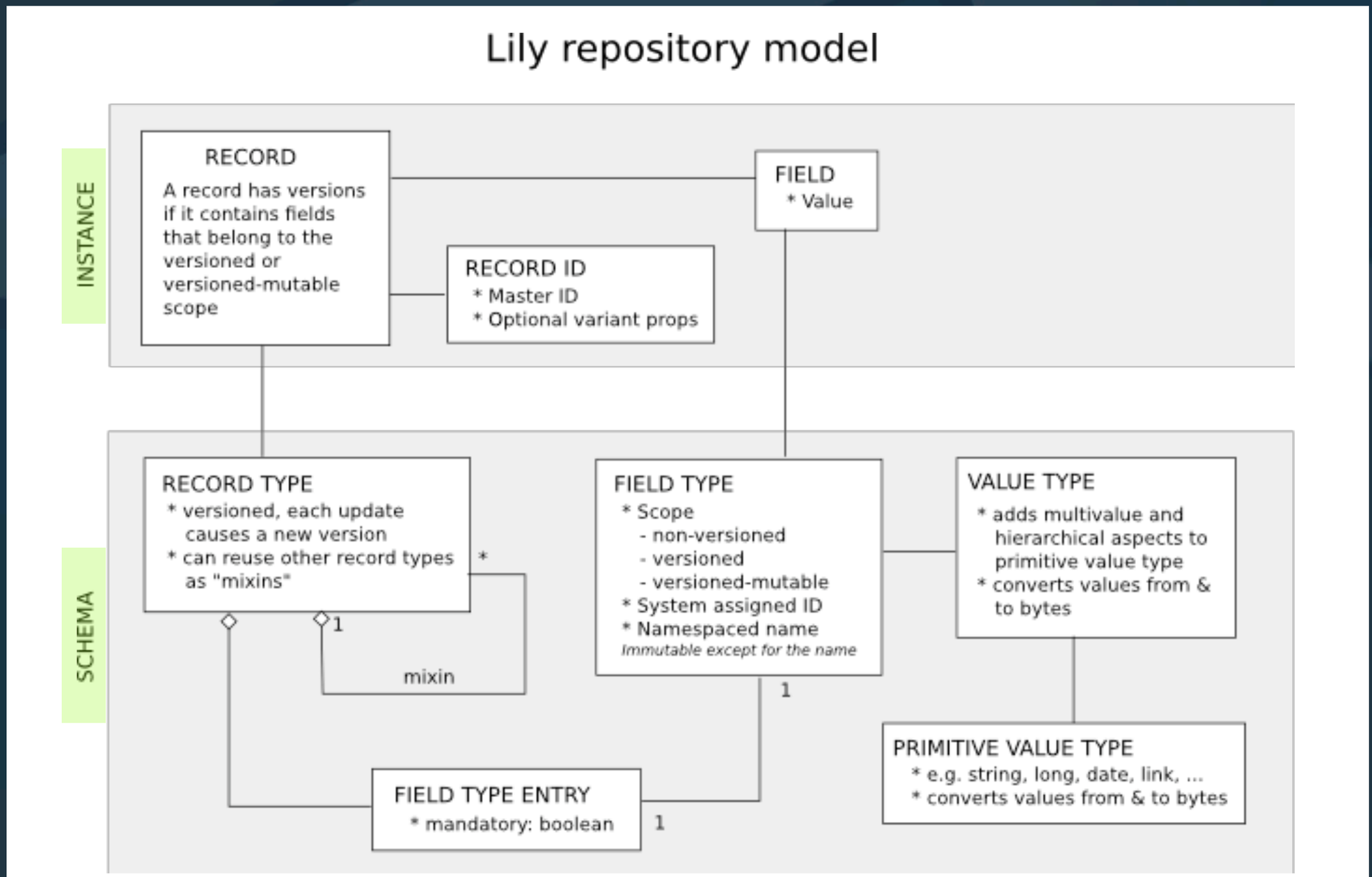


ROADMAP

- **now**: highly-scalable data repository: store, index and search
- **next**: with real-time usage stats gathering and analytics
- **later**: and built-in context- and user-sensitive recommendations

- built on top of Google BigTable / HBase / Solr
  - identical, robust technology in use at Facebook, Twitter, StumbleUpon, Yahoo!
  - scales widely over distributed (cloud) infrastructure

LILY REPOSITORY MODEL

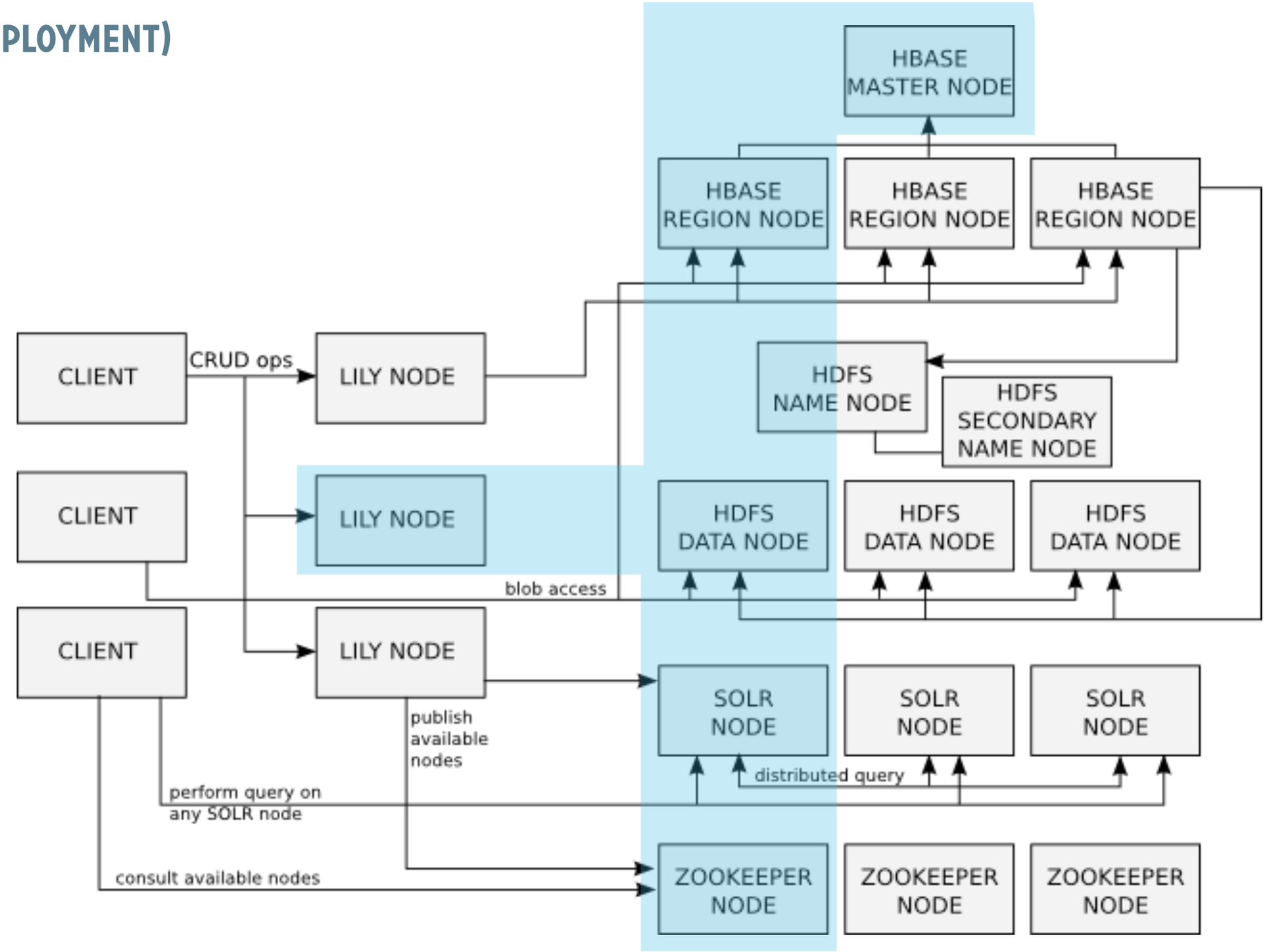


SAMPLE LILY SCHEMA (EXCERPT)

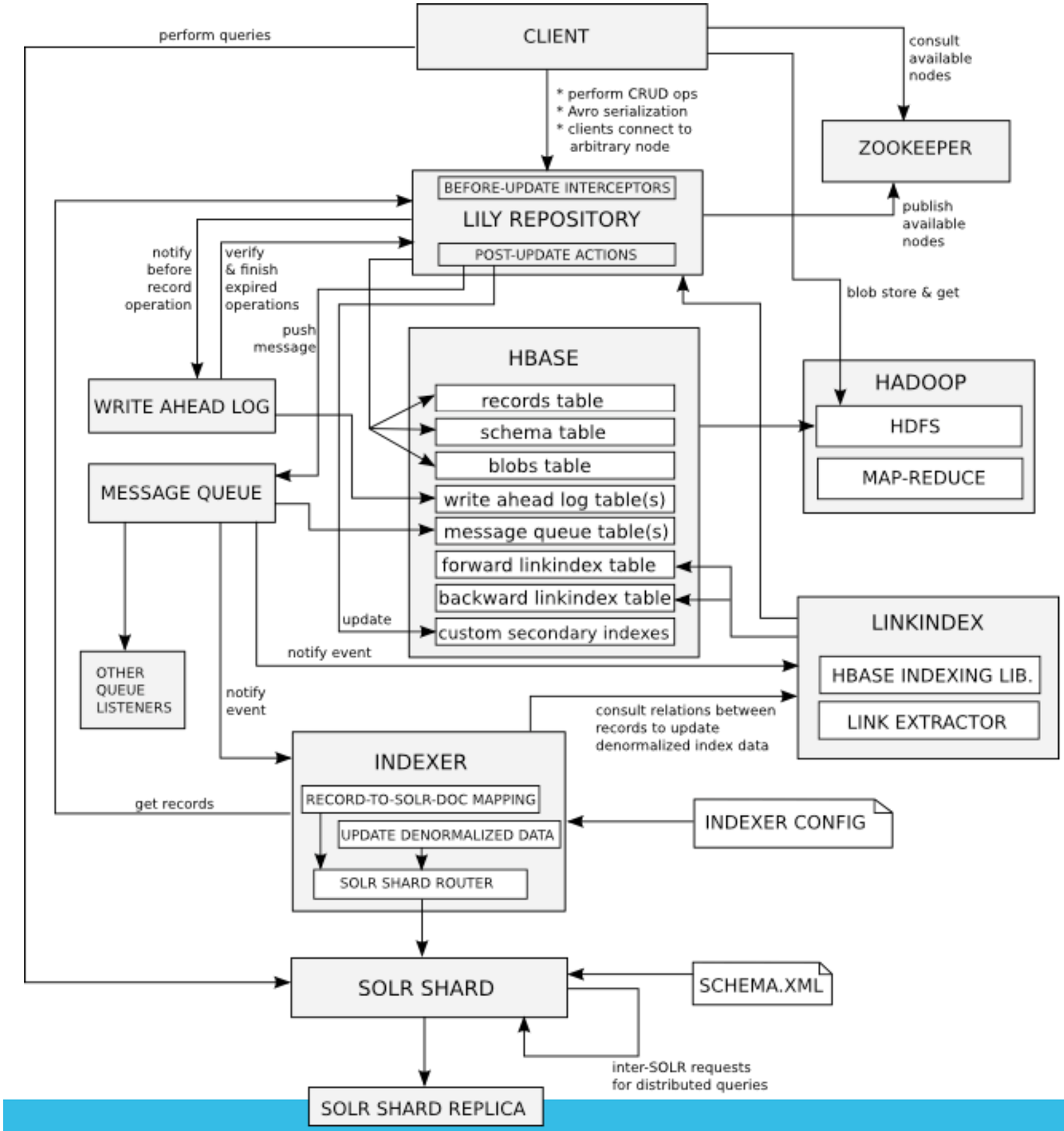
```
namespaces: {
    /* Declaration of namespace prefixes. */
    "org.lilyproject.bookssample": "b",
    "org.lilyproject.vtag": "vtag"
  },
fieldTypes: [
  {
    name: "b$title",
    valueType: { primitive: "STRING" },
    scope: "versioned"
  },
  {
    name: "b$pages",
    valueType: { primitive: "INTEGER" },
    scope: "versioned"
  },
  {
    name: "b$language",
    valueType: { primitive: "STRING" },
    scope: "versioned"
  },
  {
    name: "b$authors",
    valueType: { primitive: "LINK", multiValue: true },
    scope: "versioned"
  },
  {
    name: "b$name",
    valueType: { primitive: "STRING" },
    scope: "versioned"
  },
  {
    name: "b$bio",
    valueType: { primitive: "STRING" },
    scope: "versioned"
  },
  {
    name: "vtag$last",
    valueType: { primitive: "LONG" },
    scope: "non_versioned"
  }
  ],
recordTypes: [
  {
    name: "b$Book",
    fields: [
      {name: "b$title", mandatory: true },
      {name: "b$pages", mandatory: false },
      {name: "b$language", mandatory: false },
      {name: "b$authors", mandatory: false },
      {name: "vtag$last", mandatory: false }
    ]
  },

...
```

LILY ARCHITECTURE

(DEPLOYMENT)



LILY ARCHITECTURE
(COMPONENTS)



HBASE INDEXING & ROWLOG LIBRARY

- building and querying indexes, GAE-style

content table

rowkey	col	col
A	val3	foo6
B	val2	foo7

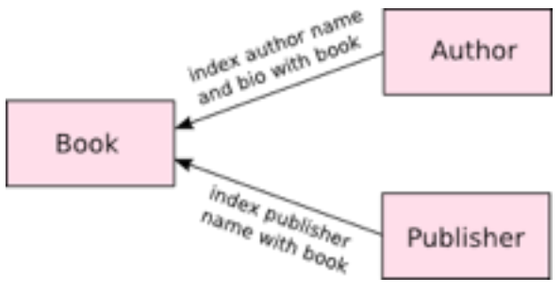
index table A (order)

rowkey	col
val2-B	
val3-A	

- need for sync/async operations
  - updating of secondary indexes (e.g. link tables)
  - feeding of Indexer (= indexes Lily-content into Solr)
- not: transactions
- need for distribution and durability

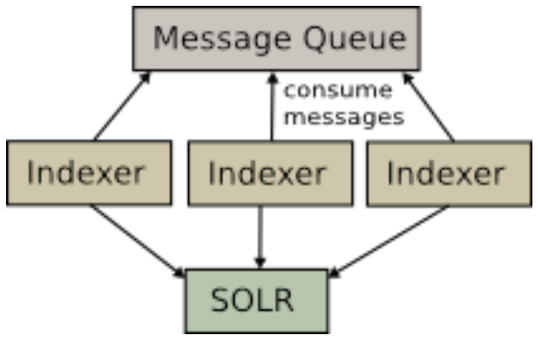
THE LILY INDEXER

denormalization

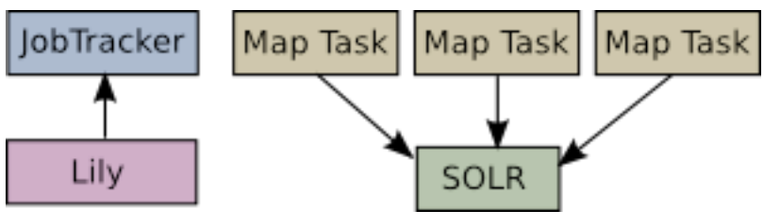


indexing of multiple versions of a record

incremental index updating

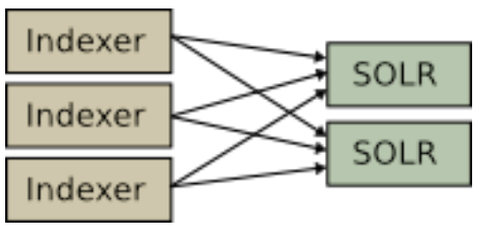


batch index building



blob content extraction

sharding towards multiple SOLR instances

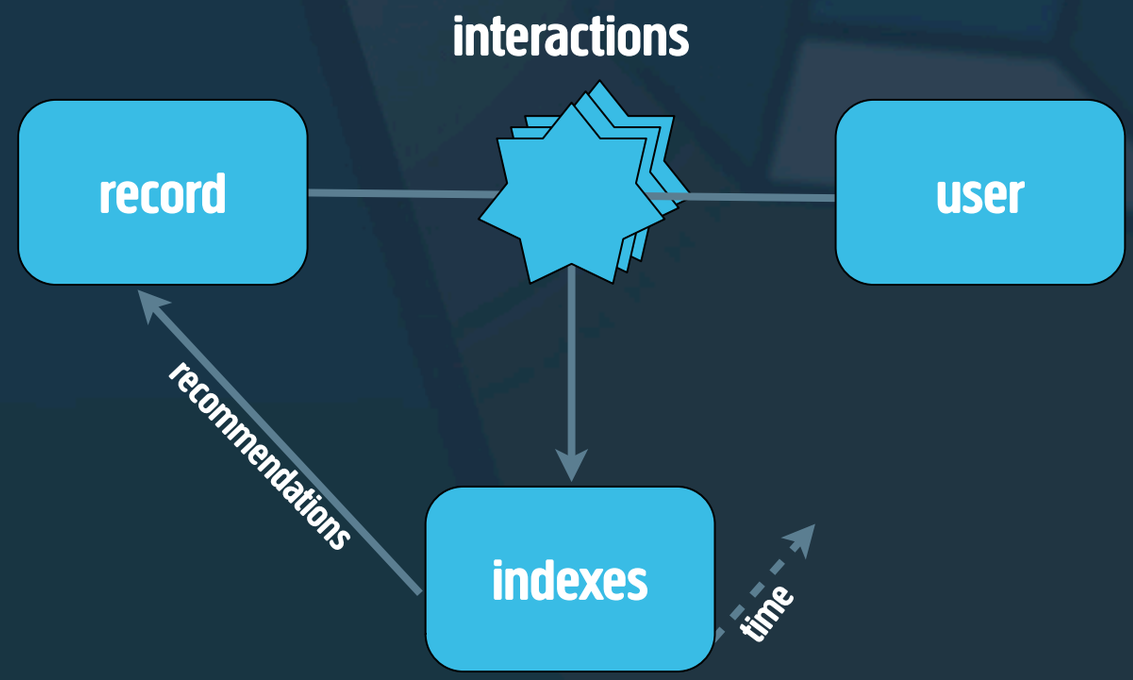


STATUS JUNE 2011

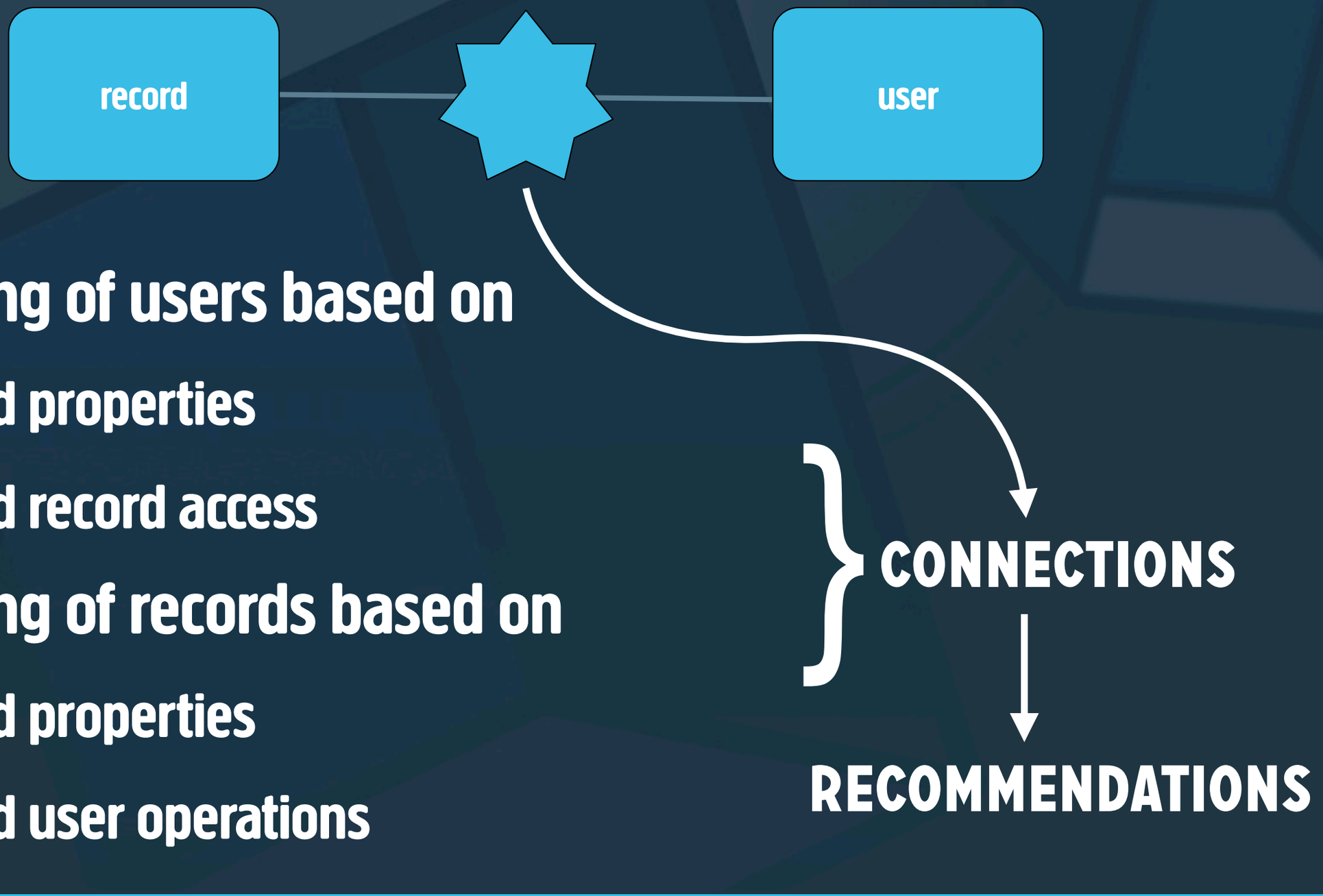
- Lily 1.0.1 released - developing since Q4/09
- some customers - DIY retail / media / news
  - e-commerce platform project
  - Lily as the data (integration) tier
- first contrib: FrogPond (annotated Java <> Lily mapper)
  https://bitbucket.org/calmera/frogpond

NEXT UP: USAGE STATS

- sits in CRUD-path
- tracks users ops against records
  - from both perspectives
  - arbitrary K/V properties: time, location, ...
- automatically builds user profiles (as records)
  - tied to records ops
  - indexed access
  - time dimension: trending



FROM USAGE STATS TO RECOMMENDATIONS 'LIGHT'



- grouping of users based on
  - shared properties
  - shared record access
- grouping of records based on
  - shared properties
  - shared user operations

FULL-ON RECOMMENDATIONS

- look at real-time-capable Mahout algorithms
- pre-index or -calculate as much as possible
  - save as secondary indexes
  - present recommendations as part of record API
- allow user to contribute ‘domain knowledge’ to record processing pipeline
  - pattern detection, keywords, ontologies, ...

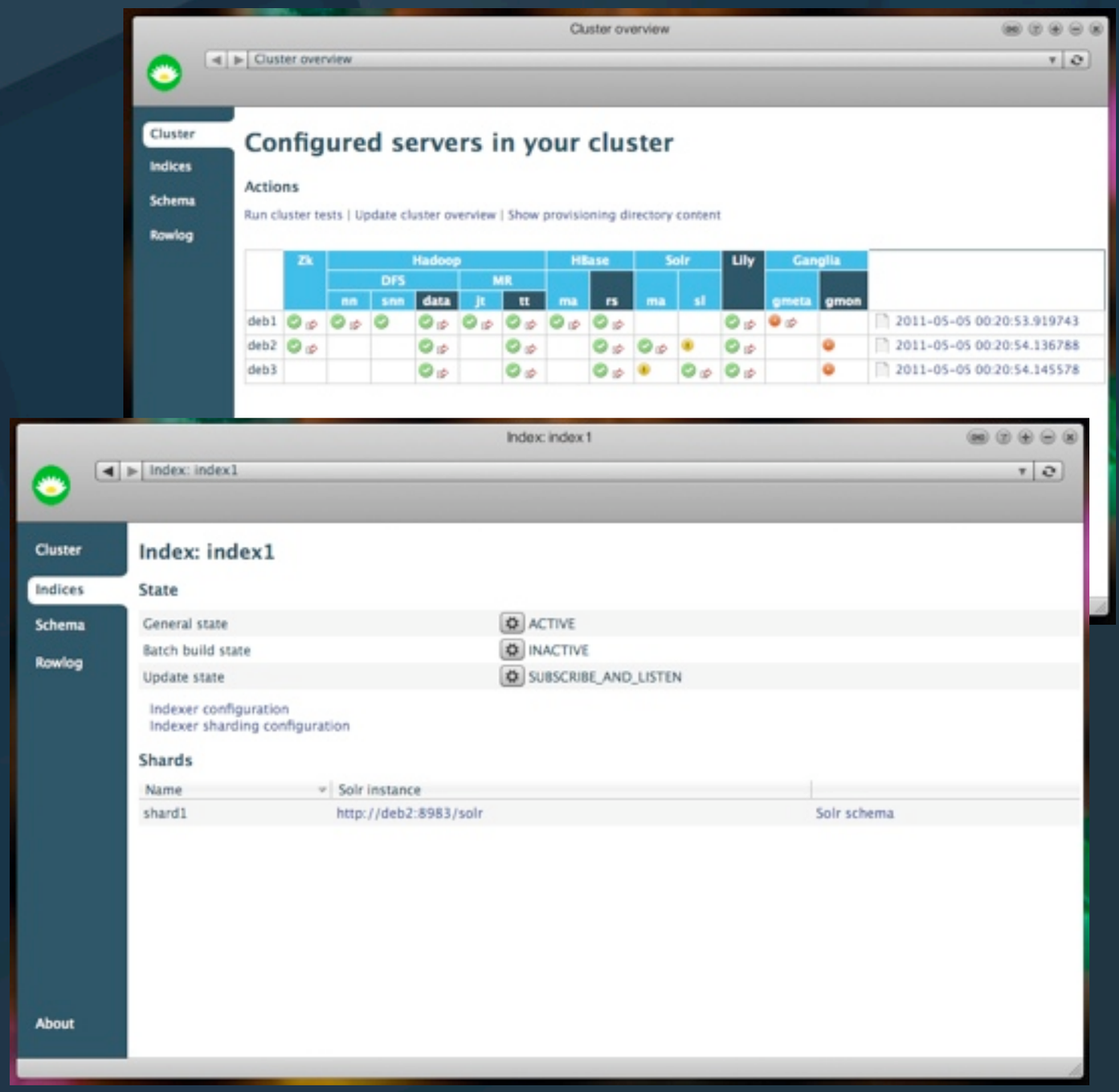
TIMELINE

- » Lily + usage stats — 10/2011
- » Lily + usage stats + light-weight analytics — 12/2011
- » Lily + recommendations ‘light’ — 3/2012
- » Lily 2.0 : full-on recommendations — 6/2012

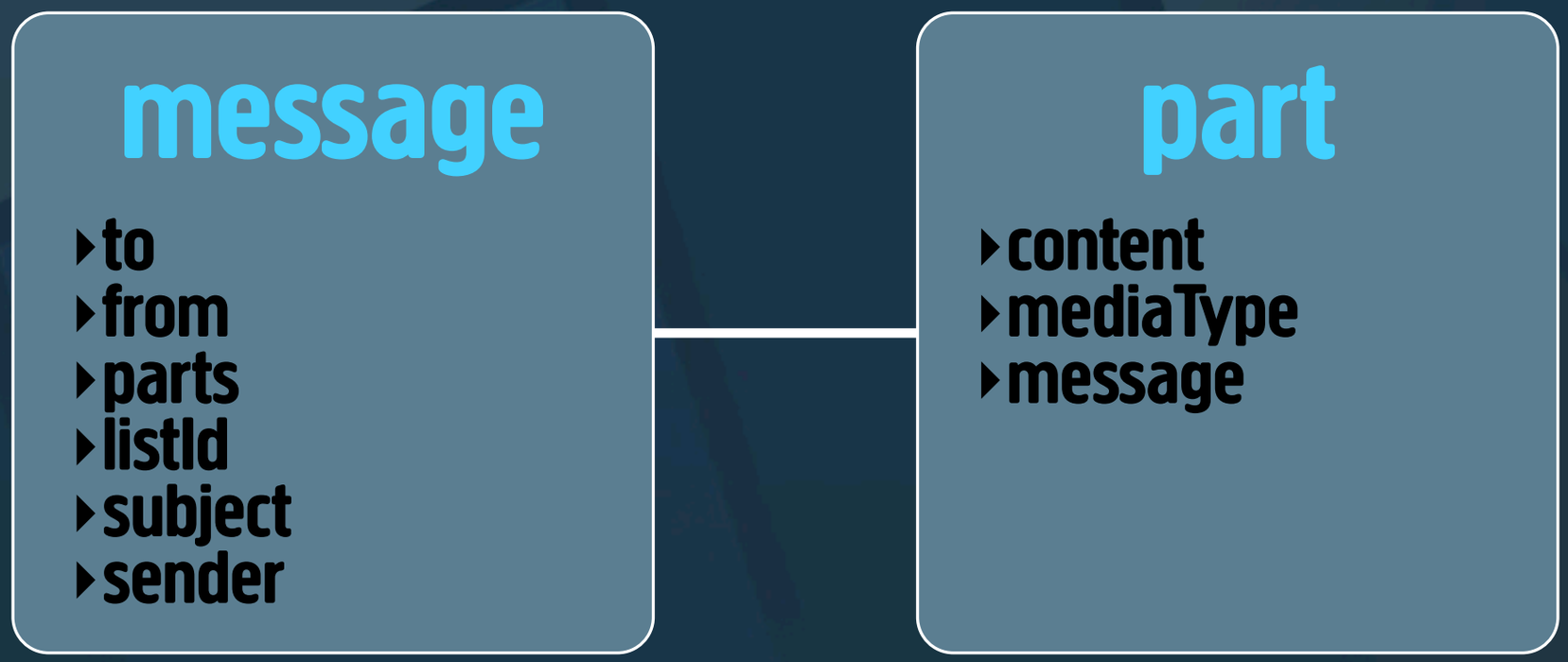
~ THE CONFERENCE OF HIGH SCALABILITY ~
BERLINBUZZWORDS

LILY ENTERPRISE

- adds tools:
  - yum/deb package repo
  - cluster deploy scripts (also EC2)
  - Admin UI
- + enterprise support



DEMO (IF TIME PERMITS)



WHERE?

www.lilyproject.org

THANK YOU !

FOR YOUR ATTENTION

FOR YOUR QUESTIONS

» stevenn@outerthought.org

» @stevenn