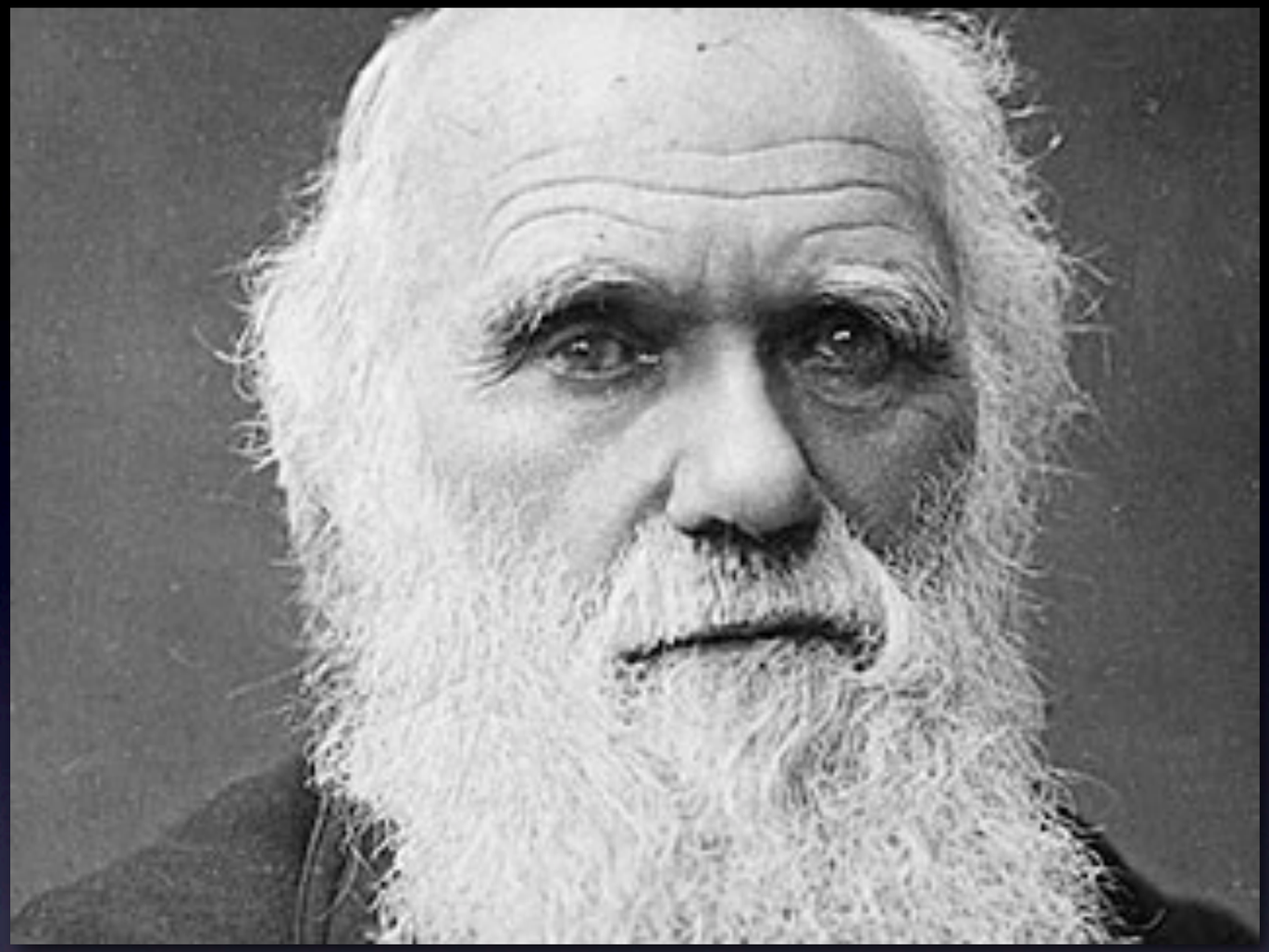
Why I chose **mongodb** for *guardian.co.uk*

Mat Wall
Lead Software Architect, *guardian.co.uk*

guardian.co.uk



"It is not the strongest of the species that survives, nor the most intelligent. It is the one that is most adaptable to change."

guardian.co.uk

Early Period

circa ’95

The “Lash It Together” era

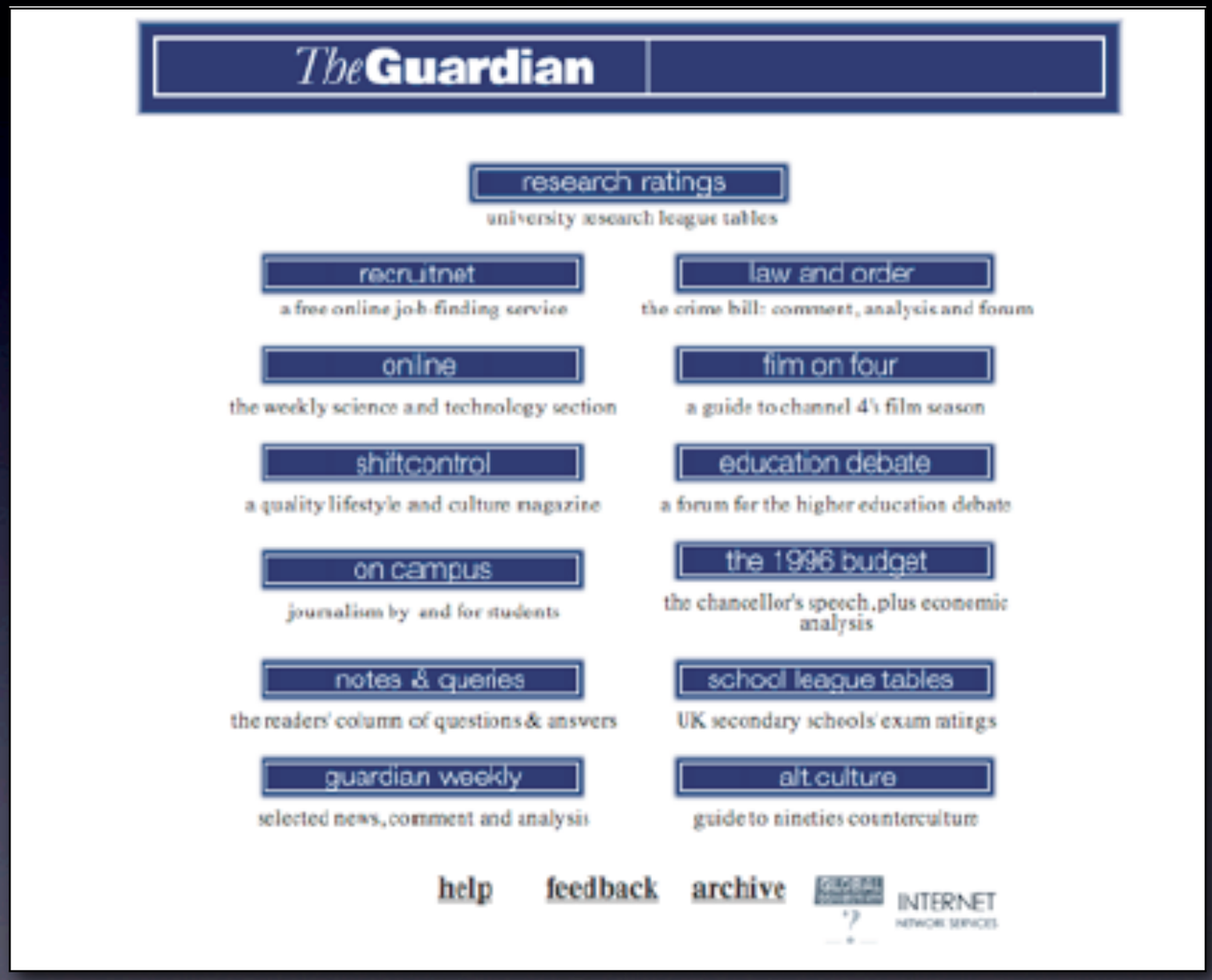
guardian.co.uk

Early Period (95, the “Lash It Together” era)

Perl, CGI, apache

Experimental
Manual processes
Bespoke software

RDBMS, scripts
& static files



guardian.co.uk

Mid Period

circa ’00

The “Vendor CMS” era

guardian.co.uk

Mid Period: 2000s (The “Vendor CMS era”)

Vignette / AOLserver
TCL, Apache, Oracle

Platform for online publishing

Initially scales well with acceleration in delivery of features

guardian.co.uk

Mid Period: 2000s (The "Vendor CMS era")

Surprise! Vendor's CMS doesn't do what we want!

Mish-mash in templates: HTML, JavaScript, TCL, SQL, PL-SQL

No model in app tier, only in RDBMS schema created in Oracle Designer

guardian.co.uk

Mid Period: 2000s (The “Vendor CMS era”)

```
SELECT
        act.first_names,
        act.last_name,
        cas.id,
        cas.template,
        art.body
FROM
        cactor act,
        cas2_scribblings_vw cas,
        carticle art,
        (
        SELECT
                act.first_names,
                act.last_name,
                MAX(art.publication_date) AS pdate
        FROM
                cactor act,
                cas2_scribblings_vw cas,
                carticle art
        WHERE
                art.id = cas.scribbling_id
                [IF {[info exists PUBLICATION]} {AND act.publication = [QUOTE_SQL $PUBLICATION]}]
                AND act.archive_date IS NULL
                AND cas.context_type = [set contexttype]
                AND cas.external_id = act.id
                AND cas.widget_name = 'Article Widget'
                [IF {$CONFIG_TYPE == "Live"} {AND cas.last_live_time IS NOT NULL}]
        GROUP BY
                act.first_names,
                act.last_name
        ) x
WHERE
        art.id = cas.scribbling_id
        [IF {[info exists PUBLICATION]} {AND act.publication = [QUOTE_SQL $PUBLICATION]}]
        AND act.archive_date IS NULL
        AND cas.context_type = [set contexttype]
        AND cas.external_id = act.id
        AND cas.widget_name = 'Article Widget'
        [IF {$CONFIG_TYPE == "Live"} {AND cas.last_live_time IS NOT NULL}]
        -- limit to rows with the max(pdate) selected by x.
        AND NVL(act.first_names, '-') = NVL(x.first_names, '-')
        AND act.last_name = x.last_name
        AND art.publication_date = x.pdate
```

guardian.co.uk

Mid Period: 2000s (The “Vendor CMS era”)

```
<FORM METHOD="post" ACTION="[CURL /redirect]">
<TABLE WIDTH="[SHOW tablewidth]" BORDER="0" CELLPADDING="0" CELLSPACING="0">
<TR>
<TD WIDTH="[SHOW tablewidth]" BGCOLOR="[SHOW tablebgcolor]" HEIGHT="26">

<SELECT NAME="Url" onChange="goUrl(this)">
<OPTION VALUE=" " SELECTED>[
        switch "$PUBLICATION" {
                "Guardian" {set tp "Guardian columnists"}
                "Observer" {set tp "Observer columnists"}
                default {set tp "Columnists"}
        }
        if {$contexttype == 6348} {set tp "Diarists"}
        if {$contexttype == 2065} {set tp "Critics"}
        set tp]
<OPTION VALUE=" ">[INCLUDE LIBNAME "/lib/lib_separator.vgn"]
[
        set out {}
        # Iterate over the reduced list of articles, sorting by columnist name.
        foreach name [lsort -command namesort [array names columns]] {
                set row $columns($name)
                append out [subst {<option value="[CURL [FIELD template $row] [FIELD id $r
st_name $row]\\n}]
        }
        set out
]
</SELECT>

<NOSCRIPT><INPUT TYPE="submit" VALUE="Go"></NOSCRIPT>

</TD>
</TR>
</TABLE>
</FORM>
```

guardian.co.uk

Mid Period: 2000s (The "Vendor CMS era")

After a few years, **very** difficult to extend

Database schema becomes fixed due to dependencies in templates

guardian.co.uk

Mid Period: 2000s (The “Vendor CMS era”)

If you can’t change the system:

guardian.co.uk

Modern Period

circa ’05-09

The “J2EE Monolithic” era

guardian.co.uk

guardian.co.uk

Breaking news: Flights ban costing Tui Travel up to £6m a day

Naval ships sent to rescue volcano-stranded Britons

Last updated three minutes ago

Aircraft carrier and assault ships deployed to boost cross-Channel options amid ash cloud flight ban

- 'We need to bring people home'
- British students stranded in Chinese 'paradise'
- Call for research on climate link to geo-hazards
- **Ferries and Eurostar lay on extra capacity**
- Dan Snow fails in Dunkirk-style mission
- Kenya's farmers losing $1.3m a day
- **Iceland volcanic ash sunsets**

Live election blog: Latest campaigning

Last updated five minutes ago

Andrew Sparrow covers all the general election news and events in the second full week of the campaign

109 comments

Volcano live blog

Latest updates on travel disruption

You review

Blur's new single

Election picks

guardianjobs

Web server

Web server

Web server

App server

App server

App server

Oracle

CMS

Data feeds

guardian.co.uk

Modern java app

Spring / Hibernate

DDD / TDD

Strong model in java

Database abstracted away with ORM

guardian.co.uk

Problems

Each release involves schema upgrade

Schema upgrade = downtime for journalists

guardian.co.uk

Complexity still increasing:

300+ tables,
10,000 lines of hibernate XML config
1,000 domain objects mapped to database
70,000 lines of domain object code
Very tight binding to database

guardian.co.uk

ORM not really masking complexity:

Database has strong influence on domain model: many domain objects made more complex mapping joins in RDBMS

Complex hibernate features used, interceptors, proxies

Complex caching strategy
Lots of optimisations

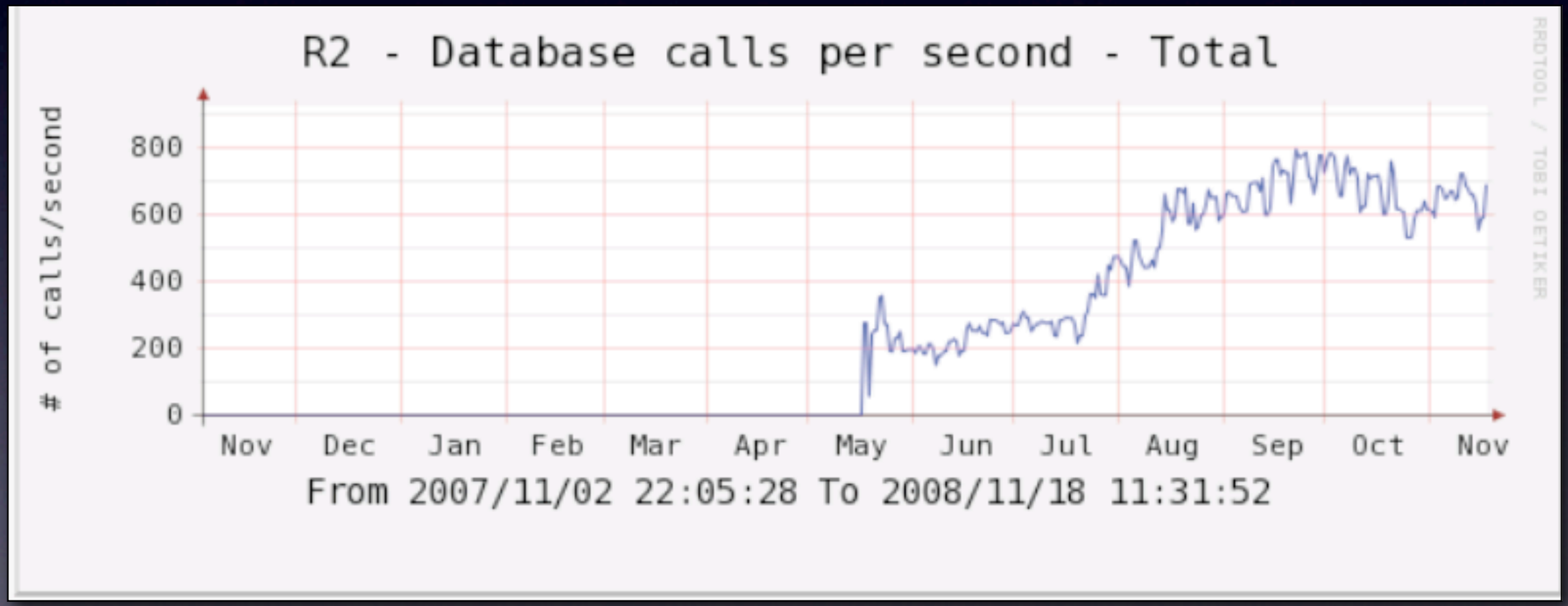
And:

We still hand code complex queries in SQL!

guardian.co.uk

Load becoming an issue

RDBMS difficult to scale



guardian.co.uk

Partial NoSQL

circa ’09-10

The “Sticking Plaster” era

guardian.co.uk

Introduce yet more caching to patch up load problems

Decouple applications from database by building APIs

Power APIs using alternative, more scalable technologies

APIs used to scale out database reads

Writes still go to RDBMs

guardian.co.uk

Core

Web servers

App server

Memcached (20Gb)

rdbms

Solr

M/Q

CMS

Api

Solr/API

Solr/API

Solr/API

Solr/API

Solr/API

Cloud, EC2

guardian.co.uk

Content API

Read API delivered using Apache Solr

Hosted in EC2

Document oriented search engine

Loose schema: records, fields, facets

Scales well for read operations

Related content from Solr

Introduction of memcached

guardian.co.uk

We've solved our load problem (for now)

but

Increased our complexity

guardian.co.uk

We now have 3 models!

RDBMS tables

Java Objects

JSON API

guardian.co.uk

```
response: {
      status: "ok",
      userTier: "free",
      total: 1155993,
      startIndex: 991,
      pageSize: 10,
      currentPage: 100,
      pages: 115600,
      orderBy: "newest",
      results: [
            {
                  id: "world/2010/apr/16/recycling-plutonium-insane-policy",
                  sectionId: "world",
                  sectionName: "World news",
                  webPublicationDate: "2010-04-16T00:00:00+01:00",
                  webTitle: "Letter: Recycling plutonium is an insane policy",
                  webUrl: "http://www.guardian.co.uk/world/2010/apr/16/recycling-plutonium-insane-policy",
                  apiUrl: "http://content.guardianapis.com/world/2010/apr/16/recycling-plutonium-insane-policy"
      },
      {
            id: "world/2010/apr/16/sahil-saeed-kidnap-ringleader-killed",
            sectionId: "world",
            sectionName: "World news",
            webPublicationDate: "2010-04-16T00:00:00+01:00",
            webTitle: "Sahil Saeed kidnap 'ringleader' killed in police shoot-out",
            webUrl: "http://www.guardian.co.uk/world/2010/apr/16/sahil-saeed-kidnap-ringleader-killed",
            apiUrl: "http://content.guardianapis.com/world/2010/apr/16/sahil-saeed-kidnap-ringleader-killed"
      },
```


guardian.co.uk

```
results: [
    {
        id: "world/2010/apr/16/recycling-plutonium-insane-policy",
        sectionId: "world",
        sectionName: "World news",
        webPublicationDate: "2010-04-16T00:00:00+01:00",
        webTitle: "Letter: Recycling plutonium is an insane policy",
        webUrl: "http://www.guardian.co.uk/world/2010/apr/16/recycling-plutonium-insane-policy",
        apiUrl: "http://content.guardianapis.com/world/2010/apr/16/recycling-plutonium-insane-policy",
        fields: {
            headline: "Recycling plutonium is an insane policy",
            shortUrl: "http://gu.com/p/2gb9d"
        },
        tags: [
            {
                id: "world/nuclear-weapons",
                type: "keyword",
                webTitle: "Nuclear weapons",
                webUrl: "http://www.guardian.co.uk/world/nuclear-weapons",
                apiUrl: "http://content.guardianapis.com/world/nuclear-weapons",
                sectionId: "world",
                sectionName: "World news"
            },
            {
                id: "environment/nuclear-waste",
                type: "keyword",
                webTitle: "Nuclear waste",
                webUrl: "http://www.guardian.co.uk/environment/nuclear-waste",
                apiUrl: "http://content.guardianapis.com/environment/nuclear-waste",
                sectionId: "environment",
                sectionName: "Environment"
            },
            {
                id: "environment/nuclearpower",
                type: "keyword",
                webTitle: "Nuclear power",
                webUrl: "http://www.guardian.co.uk/environment/nuclearpower",
                apiUrl: "http://content.guardianapis.com/environment/nuclearpower",
                sectionId: "environment",
                sectionName: "Environment"
            },
```

```
response: {
      status: "ok",
      userTier: "partner",
      total: 1,
      content: {
            id: "world/picture/2010/apr/18/russia",
            sectionId: "world",
            sectionName: "World news",
            webPublicationDate: "2010-04-18T00:00:00+01:00",
            webTitle: "Eyewitness: De-icing a ship in Russia",
            webUrl: "http://www.guardian.co.uk/world/picture/2010/apr/18/russia",
            apiUrl: "http://content.guardianapis.com/world/picture/2010/apr/18/russia",
            mediaAssets: [
                  {
                        type: "picture",
                        rel: "big-picture",
                        index: 1,
                        file: "http://static.guim.co.uk/sys-
                        images/Guardian/Pix/pictures/2010/4/18/1271593143610/Workers-remove-ice-blocks-001.jpg",
                        fields: {
                              source: "Reuters",
                              photographer: "Ilya Naymushin",
                              credit: "Ilya Naymushin/Reuters",
                              height: "768",
                              altText: "Workers remove ice blocks from under a vessel to release its propeller on the
                              Khatanga River, Russia",
                              caption: "Workers remove ice blocks from under a vessel to release its propeller on the
                              Khatanga River in Russia",
                              width: "1024"
                        }
                  },
                  {
                        type: "picture",
                        rel: "body",
                        index: 1,
                        file: "http://static.guim.co.uk/sys-
                        images/Guardian/Pix/pictures/2010/4/18/1271593145807/Workers-remove-ice-blocks-003.jpg",
                        fields: {
                              source: "Reuters",
                              photographer: "Ilya Naymushin",
                              height: "560",
                              altText: "Workers remove ice blocks from under a vessel to release its propeller on the
                              Khatanga River, Russia",
                              caption: "Workers remove ice blocks from under a vessel to release its propeller on the
                              Khatanga River in Russia",
                              width: "780"
                        }
                  }
```

JSON API is very simple

Multiple domain concepts expressed in single document

Can be designed in forwardly extensible way

What if the JSON API was our primary model?

guardian.co.uk

Full NoSQL

in development

The “It’s the future!” era

guardian.co.uk

The first project: Identity

Current login/registration system still in TCL/PL-SQL

3M+ users in relational database

Very complex schema + PL-SQL

New system required

Can we migrate from Oracle to NoSql?

guardian.co.uk

Database selection

Simple keystore. Too simple?

Huge scalability. Do we need it?
Schema design difficult.

Simple to use, can execute similar queries to RDBMs

guardian.co.uk

MongoDB

Document oriented database
Stores parsed JSON documents

Can express complex queries

Can be flexible about consistency

Malleable schema: can easily change at runtime

Can work at both large & small scales

guardian.co.uk

MongoDB concepts

RDBMS	MongoDB
Table	Collection
Row	JSON Document
Index	Index
Join	Embedding & Linking
Partition	Shard

guardian.co.uk

Flexible Schema

```
{
    id: "tone/obituaries",
    type: "tone",
    webTitle: "Obituaries",
    webUrl: "http://www.guardian.co.uk/tone/obituaries",
    apiUrl: "http://content.guardianapis.com/tone/obituaries"
}
```

guardian.co.uk

Flexible Schema

```
{
    id: "tone/obituaries",
    type: "tone",
    webTitle: "Obituaries",
    webUrl: "http://www.guardian.co.uk/tone/obituaries",
    apiUrl: "http://content.guardianapis.com/tone/obituaries"
}
```

```
{
    id: "music/madonna",
    type: "keyword",
    webTitle: "Madonna",
    webUrl: "http://www.guardian.co.uk/music/madonna",
    apiUrl: "http://content.guardianapis.com/music/madonna",
    sectionId: "music",
    sectionName: "Music"
}
```

guardian.co.uk

Flexible Schema

Can easily represent different classes of tag as documents

Both documents can be inserted into same collection

Far simpler than equivalent hibernate mapped subclass configuration

guardian.co.uk

Flexible Schema

Simple to query:

```
db.tags.find({id: "tone/obituaries"})
```

Flexible Schema

Simple to query:

```
db.tags.find({id: "tone/obituaries"})
```

Query operators:

$ne, $nin, $all, $exists, $gt, $lt, $gte ...

```
db.tags.find({"section": {$exists: true}})

db.tags.find({"webTitle": /^Obit*/i})
```

guardian.co.uk

Modifying the schema

```
{
    id: "music/madonna",
    type: "keyword",
    webTitle: "Madonna",
    webUrl: "http://www.guardian.co.uk/music/madonna",
    apiUrl: "http://content.guardianapis.com/music/madonna",
    sectionId: "music",
    sectionName: "Music"
}
```

Modifying the schema

```
db.tags.update(
    {"id": "/music/madonna"},
    {"$push": "references"
        {
            "type": "musicbrainz",
            "id": "musicbrainz/79239441-bfd5-4981-a70c-55c3f15c1287"
        }
})
```

Modifying the schema

```
{
    id: "music/madonna",
    type: "keyword",
    webTitle: "Madonna",
    webUrl: "http://www.guardian.co.uk/music/madonna",
    apiUrl: "http://content.guardianapis.com/music/madonna",
    sectionId: "music",
    sectionName: "Music",
    references: [
    {
        type: "musicbrainz",
        id: "musicbrainz/79239441-bfd5-4981-a70c-55c3f15c1287"
    }
    ]
}
```

Schema upgrades

Schema can be upgraded simply by upgrading the application version

Application must deal with differing document versions

Can become complex over time

guardian.co.uk

Schema upgrades

This can be mitigated by:

Adding a “version” key to each document

Updating the version each time the application modifies a document

Using MapReduce capability to forcibly migrate documents from older versions if required

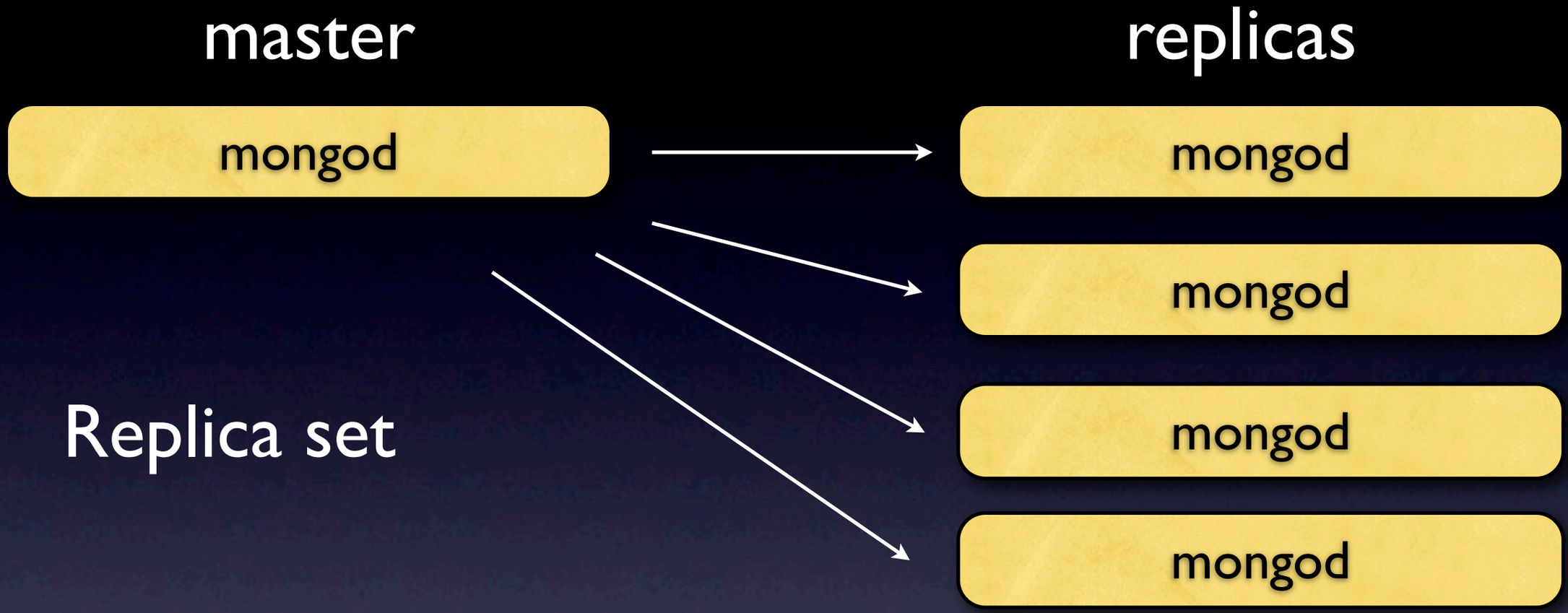
Mongodb architecture

mongod

Single node
Durability only possible in upcoming 1.8 release
(databse fsync from buffer every min)

guardian.co.uk

Mongodb architecture



Can choose to read & write from master for full consistency

Can choose to run reads on slaves to scale reads

guardian.co.uk

Mongodb architecture

Durability achieved (<1.8) via replication

Reads can be scaled out onto replicas
(eventual consistency)

All writes to master

If master fails, new master nominated by election

DB drivers handle most cluster complexity

Mongodb architecture

guardian.co.uk

Mongodb architecture

Writes scaled by sharding

Shards populated by ranges

mongos queries appropriate shard(s)

Shards automatically balanced

Developers (essentially) unaware of shards

Aggregator

consistent (master)

inconsistent (replica)

mongos

shard shard shard shard

replica replica replica replica

replica replica replica replica

guardian.co.uk

Mongodb durability

Relies (pre 1.8) on replication for durability
1.8 features optional journaling & redo logs

Database users need to be cluster aware,
each query can specify:

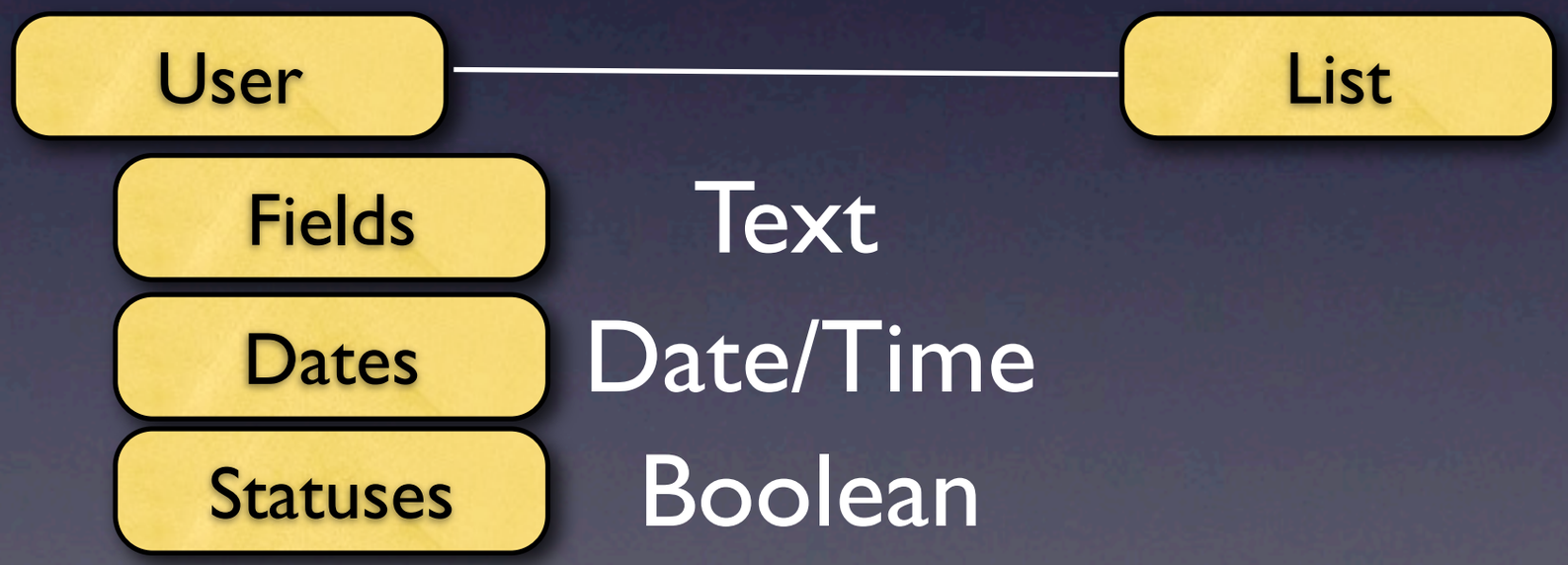
No error checking / write confirmation
Write confirmed on master
Write replicated to N slave servers

guardian.co.uk

Old Idenity system

Hundreds of tables & stored procedures

New Identity model



```
"user": {
    "primaryEmailAddress":"kenneth.lim-1299682798333anon@forwardmail.gutest.gnl",
    "id":"4425770",

    "publicFields": {
        "username":"15b86c59a2"
    },

    "privateFields": {
        "postcode":"N11GU",
        "country":"United Kingdom",
        "status":"waiting_initial_email_validation",
        "legacyPackages":"CRE,RCO,GEXT",
        "legacyProducts":"GEX,CRE,RCO",
        "registrationIp":"192.168.15.15"
    },

    "lists":[
        {
            "listId":"/system/policies/basicCommunity",
            "joinedOn":"2011-03-09T15:00:09Z"
        }
    ],

    "statusFields":{
        "gnmMarketing":true,
        "thirdPartyMarketing":true,
        "userEmailValidated":false
    },

    "dates": {
            "accountCreatedDate":"2011-03-09T14:59:56Z",
            "lastActivityDate":"2011-03-09T15:00:09Z"
        }
    }
}
```

Very simple domain objects

```
case class ListItem(id: String,
                    joined: Option[DateTime] = None)

case class User(primaryEmailAddress: String,
                id: String,
                publicFields: Map[String, String],
                privateFields: Map[String, String],
                lists: List[ListItem],
                statusFields: Map[String, Boolean],
                dates: Map[String, DateTime])
```

Simple, flexible objects
No hibernate session

Very simple domain objects

```
case class ListItem(id: String,
                    joined: Option[DateTime] = None)

case class User(primaryEmailAddress: String,
                id: String,
                publicFields: Map[String, String],
                privateFields: Map[String, String],
                lists: List[ListItem],
                statusFields: Map[String, Boolean],
                dates: Map[String, DateTime])
```

Flexible schema embraced in domain object design

Very simple domain objects

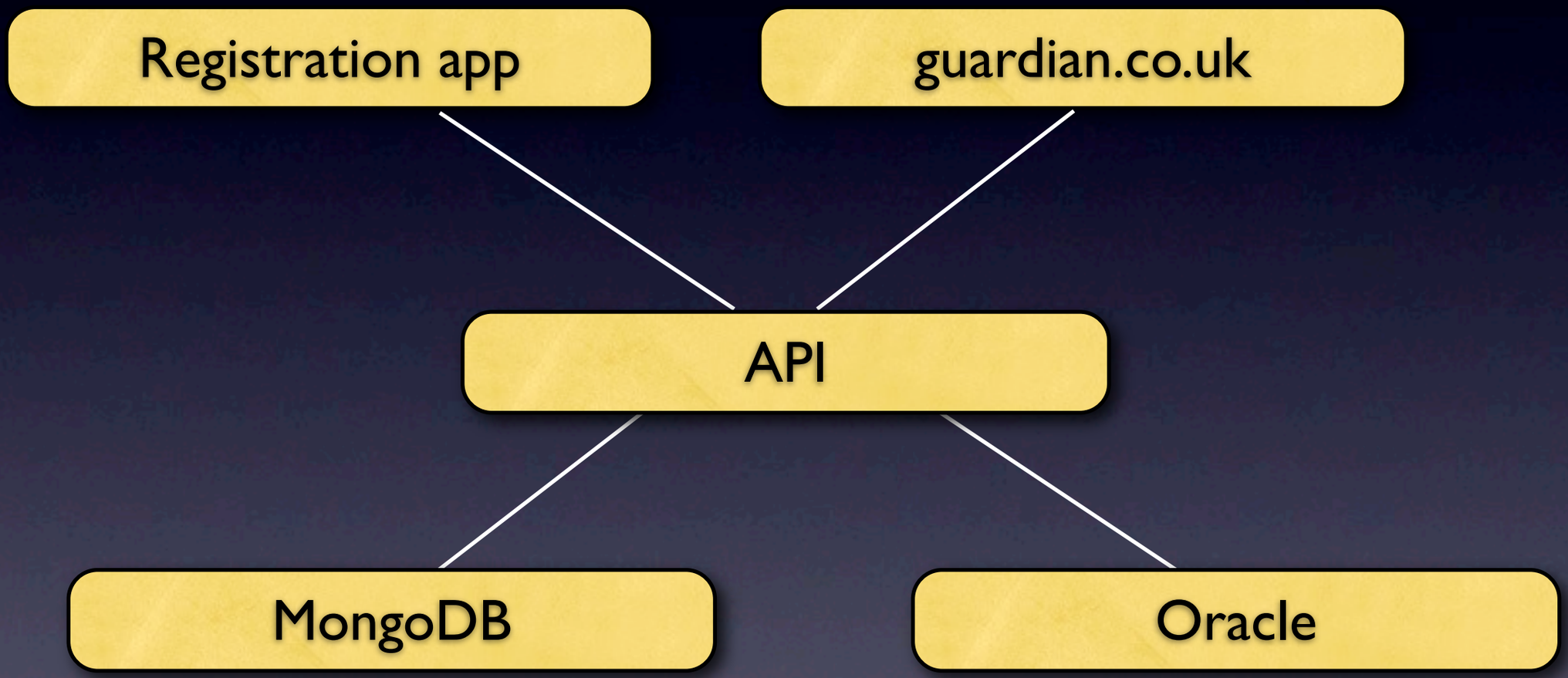
```
case class ListItem(id: String,
                    joined: Option[DateTime] = None)

case class User(primaryEmailAddress: String,
                id: String,
                publicFields: Map[String, String],
                privateFields: Map[String, String],
                lists: List[ListItem],
                statusFields: Map[String, Boolean],
                dates: Map[String, DateTime])
```

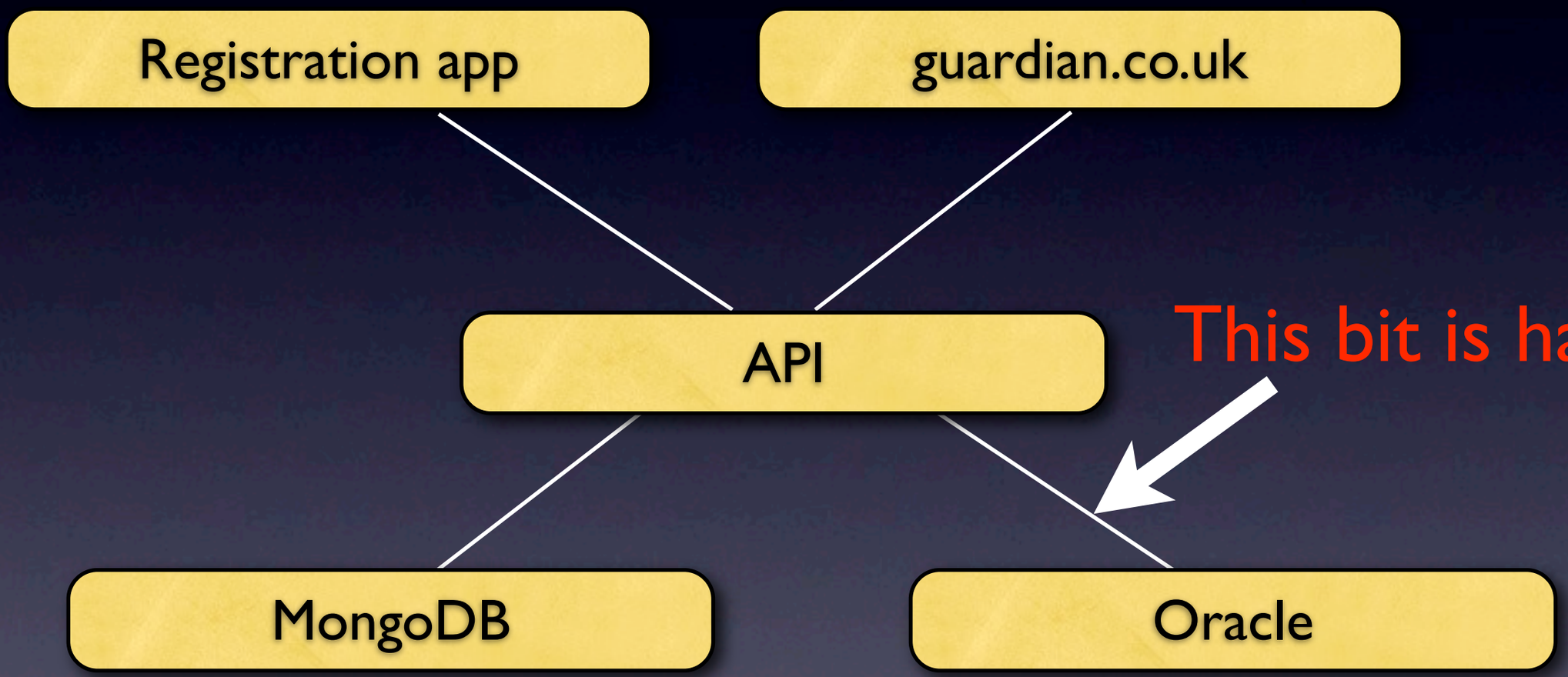
Using casbah scala drivers = significant reduction in LOC vs SQL implementation

guardian.co.uk

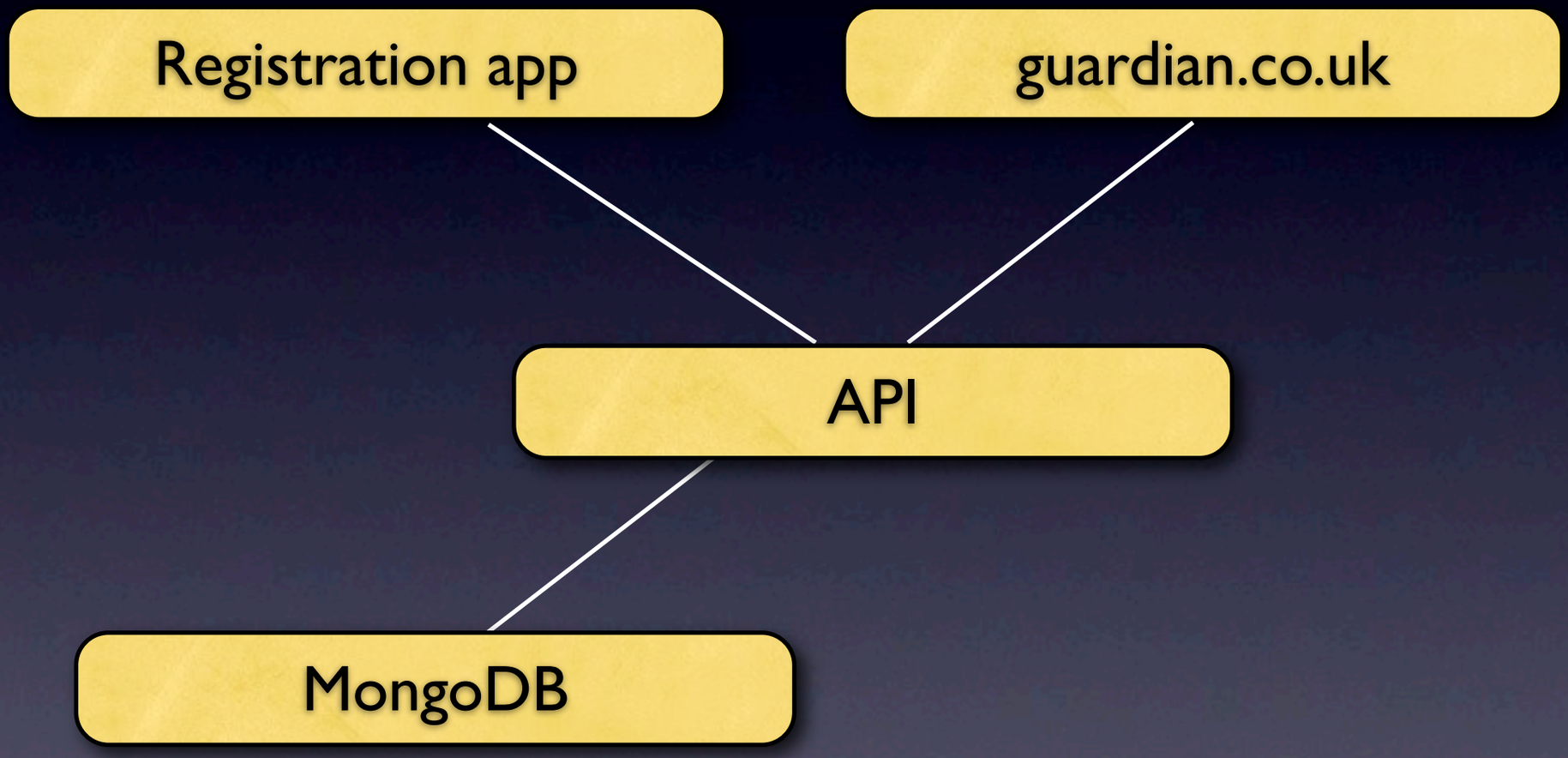
Build API that can support both backends



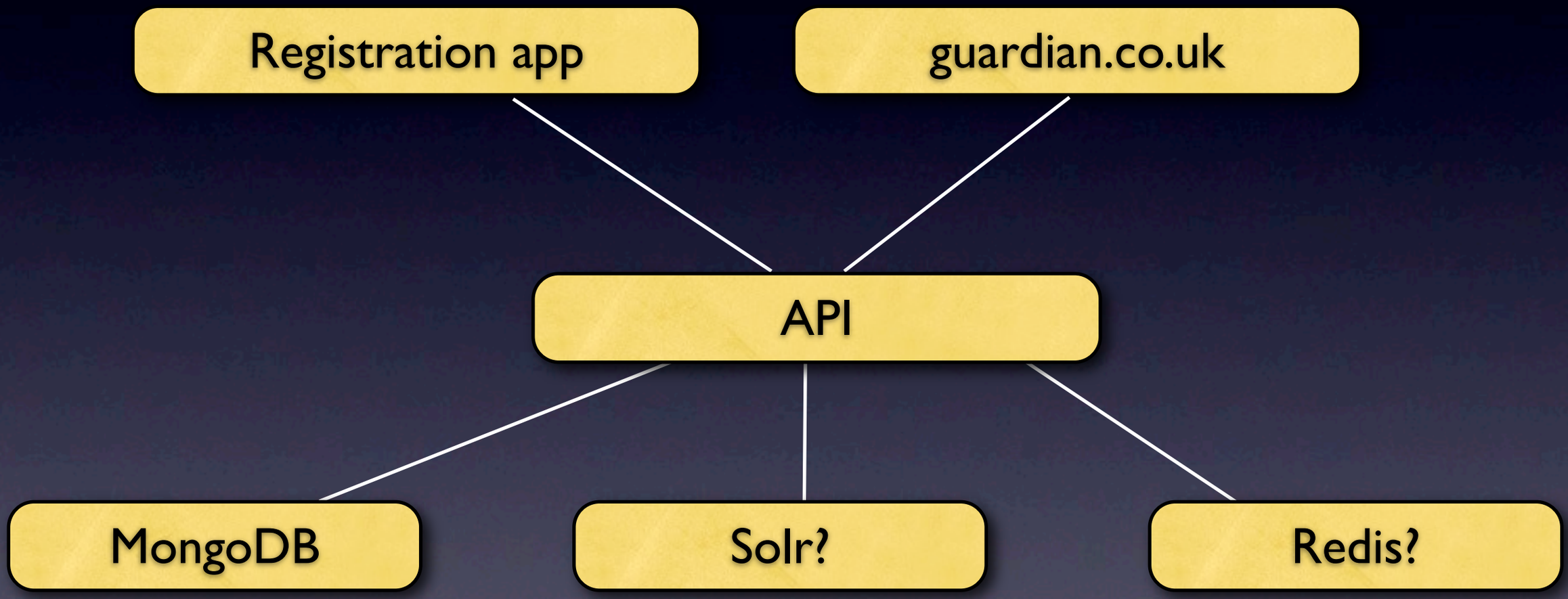
Build API that can support both backends



Migrate using API & decommision



Add new stuff!



MongoDB

Simple, flexible schema with similar query & indexing to RDBMS

Great at small or large scale

Easy for developers to get going

Commercial support available (10Gen)

One day may power all of guardian.co.uk

No transactions / joins: developers must cater for this

Produces a net reduction in lines of code / complexity

guardian.co.uk

Shameless plug

We’re hiring:

http://www.careersatgnl.co.uk

guardian.co.uk