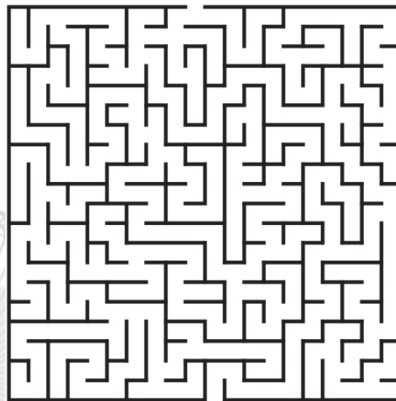




Finding your way out of the Domino maze

Julian Woodward | LDC Via



www.ldcvia.com

Looking for the
perfect answer to
a difficult question...

is like the blind man...
in the dark room...
looking for the black cat...

... that isn't there.

The session is NOT about...

- Persuading anybody to leave Domino
- Persuading anybody to stay on Domino
- Persuading anybody to move to Domino
- An LDC Via sales pitch
- Email migration



The session **is** about...

- Examining a difficult issue
- From all sides
- Casting some light into the darkness
- Sharing some ideas

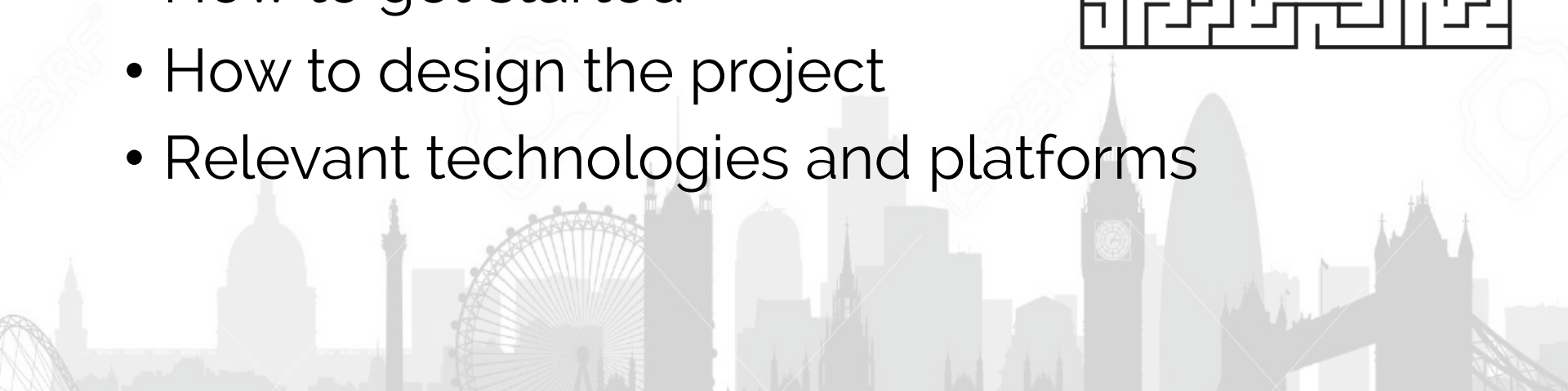
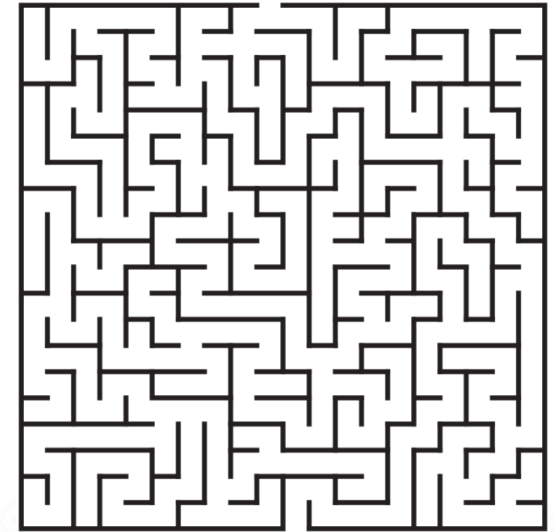


About Me

- UK – London – Oxfordshire
- 80s/90s – C, Assembler, Prolog, Clipper, SQL
- 90s/00s – Notes/Domino (1993), MS Access, Visual Studio, VB, Java, Professional classical singer
- 2005 – Independent / freelance
- 2008 – London Developer Coop -> "LDC"
- 2014+ – LDC Via and KEEP.WORKS

What we will cover

- Where we are now
- Technical considerations
- Migration approaches
- How to get started
- How to design the project
- Relevant technologies and platforms



Where are we now?



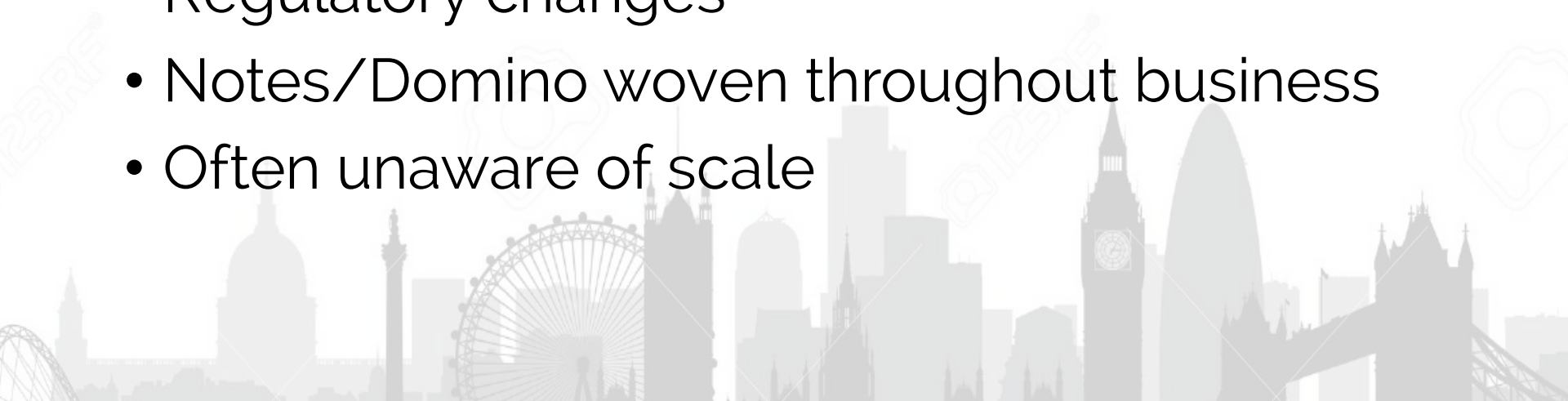
Where are we now?

- Two parts to this question
 - Business
 - Technical



Business context

- Budgetary constraints
- Major business events
- Political considerations
- Regulatory changes
- Notes/Domino woven throughout business
- Often unaware of scale



IT Context

- Trend: Virtualisation -> Cloud -> PaaS/SaaS
- Old platforms (R6? R7?)
- Off-the-shelf vs bespoke software
- Restructuring
- Offshoring
- Personalities



Defining the challenge



What is “migrating”?

- Single application or entire installation
- Notes vs Domino
- “Everything is going to SharePoint”



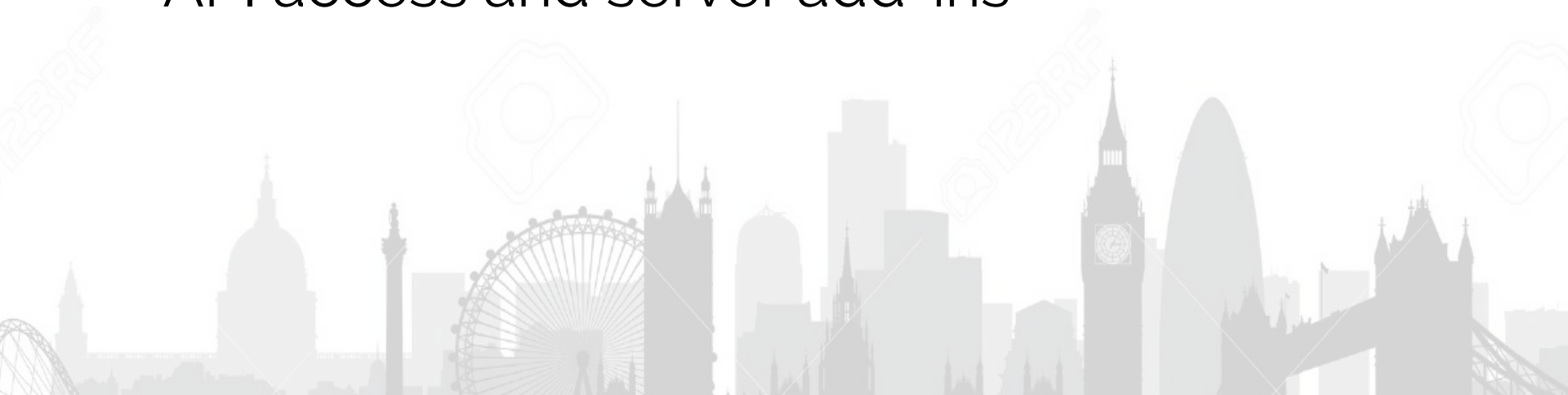
Technical landscape

- Notes applications
- 'Classic' Domino applications
- XPages Domino applications



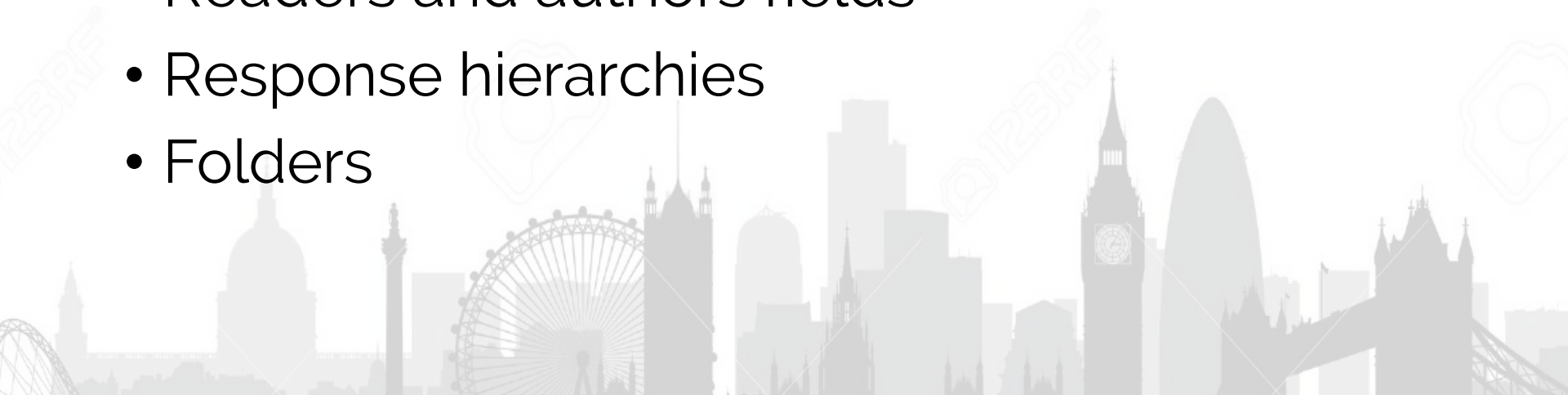
Notes/Domino challenges

- Scheduled agents
- Triggered agents
- Server-side integration
- API access and server add-ins



Notes/Domino challenges

- Mail-out functionality
- Mail-in functionality
- Rich text editor capabilities
- Readers and authors fields
- Response hierarchies
- Folders



Domino challenges

- Classic Domino applications
 - Complex interlinked code base
 - Forgotten Domino 'hacks'
- XPages applications
 - Domino version-specific behaviour
 - Use of OpenNTF projects
 - Code base can be similarly complex

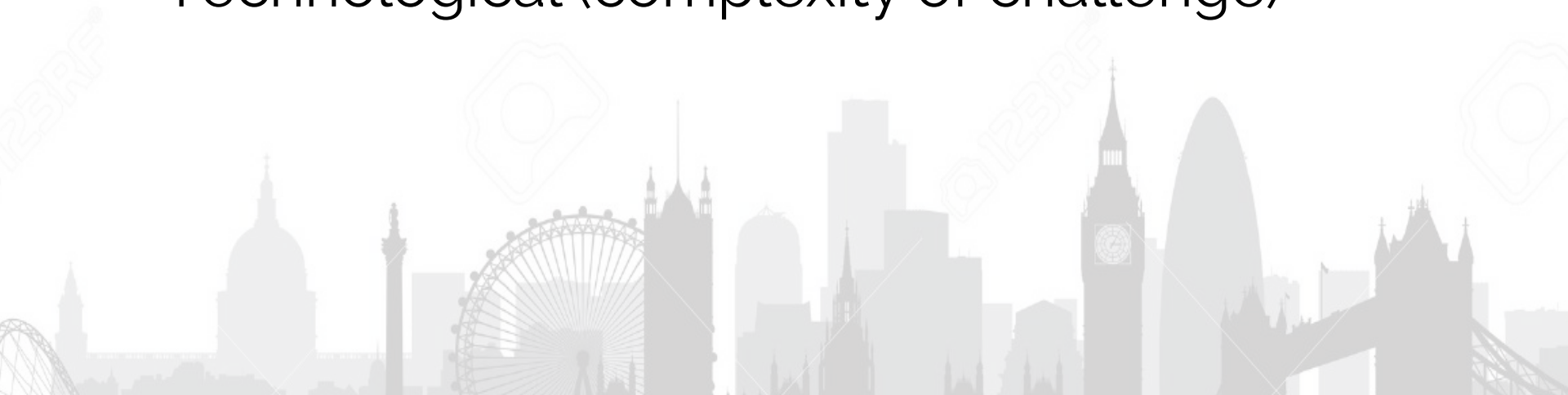


Notes client challenges

- Rich client environment
- Client-side integration
- Doclinks
- Offline use
- User habits and working practices
- Copy/paste between applications/databases
- Copy/paste to/from email

Summary

- A lot of factors to consider
- High-level (business, strategy)
- Environmental (regulation, policy)
- Technological (complexity of challenge)



Where do we go from here?



www.ldcvia.com

Where do we go from here?

Wholesale “big-bang”

or

“best fit” approach



“Big bang” approach

- Normally business-led decision
- Single destination platform
 - Salesforce
 - SharePoint
- Not really a big bang – long-term work



"Best fit" approach

- No single destination platform
- Need to select appropriate route for each application
- An IT-led approach
- A lot more decisions to take
- Possibly a 'better' outcome



Options

- Mothball
- Archive
- Re-skin
- Abstract
- Start fresh
- Application migration
- Data migration

Option 1: Mothball

Make application read-only and leave it running

- PROs
 - Quick
 - No investment
- CONs
 - Still running Domino
 - Application is 'dead'
 - Not a migration



Option 2: Archive

Extract data to static store

- PROs
 - Low cost
 - Quick
 - Getting off Domino
- CONs
 - Application is 'dead'
 - Limited access
 - Not a migration



Option 3: Re-skin

Data stays on Domino but through a new UI

- PROs
 - Easy transition
 - May solve a problem
 - Existing skills(?)
- CONs
 - Still using Domino
 - Significant investment in a temporary solution
 - Not a migration



Option 4: Start fresh

Leave existing data behind and build new application

- PROs
 - Clean
 - Re-scope
 - No data migration
- CONs
 - Still need to mothball or archive
 - Loss of access to data
 - Not a migration



Option 5: Abstract

Data stays on Domino but as data store only

- PROs
 - Good architecture
 - Half-way house
 - No data migration
- CONs
 - Data still on Domino
 - Not a migration



Option 6: App Migration

Machine migration of full application: both data and code

- PROs
 - Quick
 - Comprehensive(?)
 - Move off Domino
- CONs
 - Options limited
 - Complexity of code
 - Code maintainability
 - Data model
 - Completion work



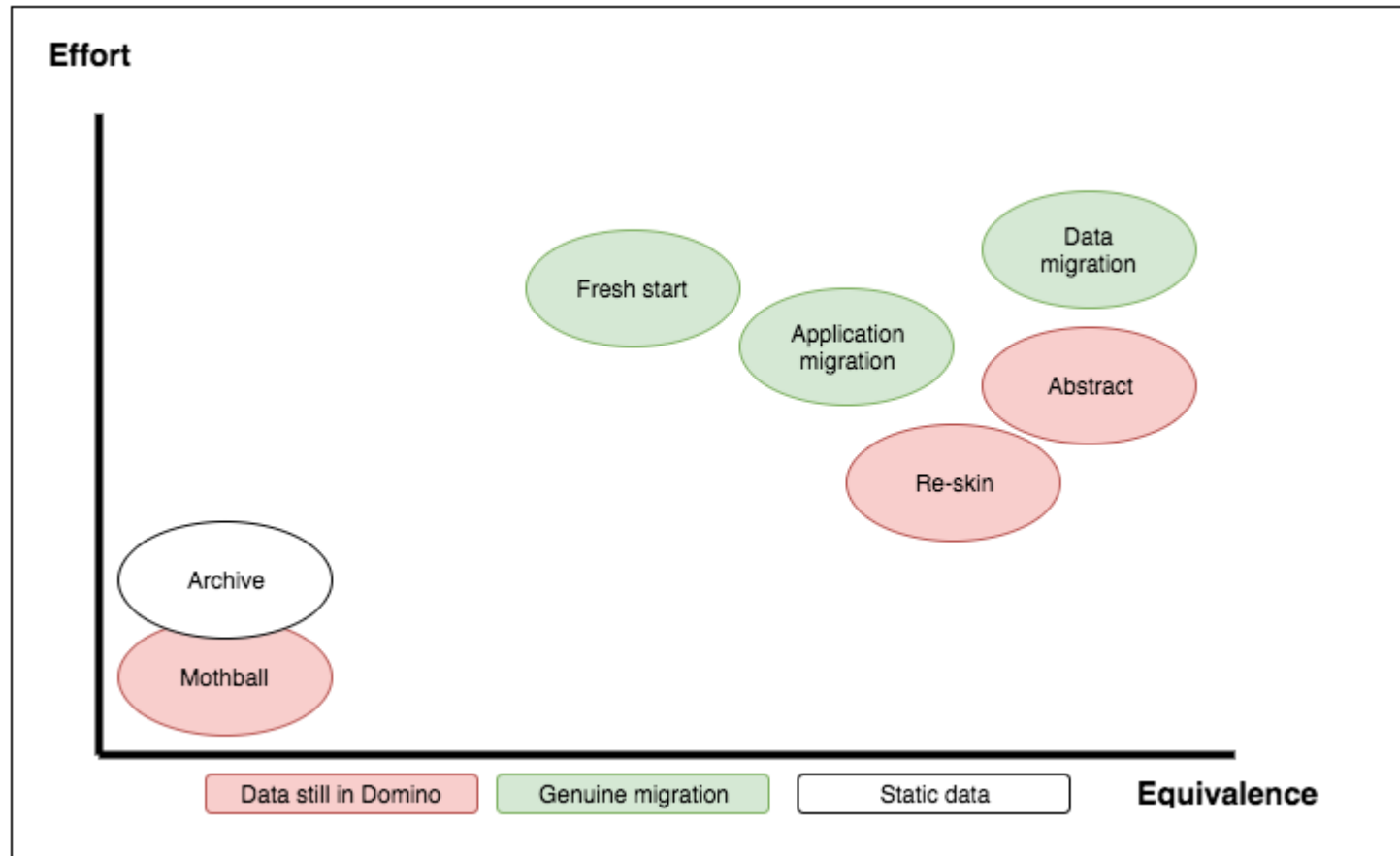
Option 7: Data migration

Migrate data, and rewrite application

- PROs
 - Flexible
 - Good architecture
 - Genuine migration
- CONs
 - Maximum effort
 - How to do the data migration?
 - Challenges with messy Domino data



Options



Data migration

- Database type
 - SQL: Oracle, MySQL, DB2, SQL Server, ...
 - NoSQL: MongoDB, Cloudant, Couch, ...
- Data migration approach
 - Manual – maximum control, maximum work
 - Tool-based – less control, less work
- Data gotchas
 - Multi-value fields, Readers/Authors, response hierarchies, doclinks, dirty data, ...

How do we begin?



How do we begin?

- Take the “eat an elephant” approach
- Get a sense of scale, early
- It's a programme of work ...
 - ... consisting of multiple workstreams ...
 - ... consisting of multiple projects.
- It's just Another Thing To Do
 - Emotion and psychology

Start with 2 vital questions

1. What are the most important applications?
2. What are the most urgent needs?



Question 1: what are the most important applications?

- “Important” could mean:
 - Revenue-generating
 - High-profile
 - Used by C-level
 - Mission-critical
 - Dependency-laden
 - Who knows...



Question 1: what are the most important applications?

- Who knows what's important?
 - Engage with the business
 - Use tools to give 'scientific' answers re apps
 - TeamStudio
 - Panagenda
 - Others(?)
- Remember:
Business priorities are more "important" than IT priorities

Question 2: what are the most urgent needs?

- Often the same as the 'important' ones
 - But: "important" <> "urgent"
- Focus on the "important but not urgent"
- Drivers:
 - Acquisition
 - Restructuring
 - Hard deadlines
 - Managerial 'decisiveness'
 - Political (promises made)

Remember: it's just a project

- Normal project management rules apply
- Engage stakeholders
- Sponsors
- Evangelists
- Build a team
- Clarity of goals, timescales, objectives, CSFs
- Change management

Taking the first steps



Taking the first steps

- Audit the applications
- Categorise the applications
- Ascertain preferred option(s) per application
- Evaluate technologies



Auditing applications

- What you need to know (a suggestion):

Urgent:	Y / N	(reason)
Important:	Y / N	(reason)
Business sponsor:	(name)	
Department/function:	(name)	
Type(s):	Notes / Classic / XPages	
Complexity rating:	(score)	
Deadline:	(date)	(details)
Used offline?:	Y / N	(details)
Related application(s)	(details)	

Auditing applications

- Complexity
 - Size (Gb, document count)
 - Number of forms, views
 - Number of agents, script libraries, subforms, roles, etc
 - Lines of Java/LotusScript
 - Size of an empty template copy (Mb)
 - Age
 - Number of users
 - Geography (users, number of replicas)
- Use tooling to help with this

Categorising applications

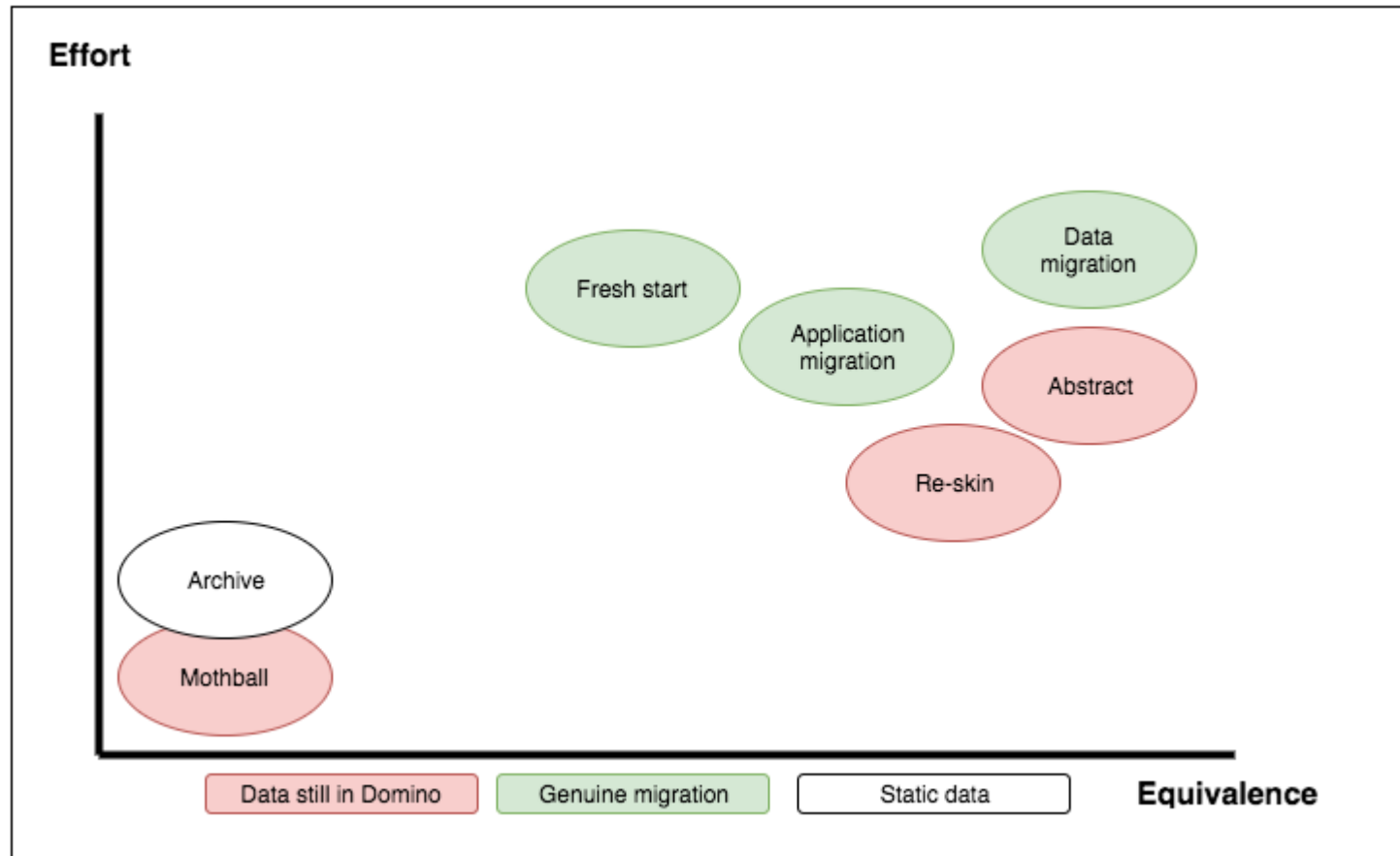
- Analyse top-down by
 - Urgency
 - Importance
 - Department
 - Complexity
 - Deadline
 - User base



Ascertain options

- Outcome, per application
 - Bin / Archive / Merge / Migrate
- Consider
 - Lifespan
 - ROI
- Quick wins?
 - Identify the quick wins, get early success
 - Pay attention to the urgent vs important

Options



Evaluate Technologies

- Languages
- Frameworks
- Databases
- Tools



Languages / Frameworks

- Java
 - Vaadin
- C#
 - .NET MVC
- Javascript
 - Nodejs + Express
 - Angular
 - React
- Be realistic

Relational databases

- Options
 - DB2
 - Oracle
 - SQL Server
 - MySQL
 - PostgreSQL
- Best for some applications
- Cleaning and restructuring
- Data migration likely to be a major project

NoSQL databases

- “Not only SQL”
 - Non-relational
 - Distributed
 - Open-source
 - Horizontally scalable
- Schema-less
- Replicating
- Several types
- Characterised by scalability and flexibility

NoSQL database types

- Key-Value stores
 - Redis, Dynamo, Oracle NoSQL
- Wide-column stores
 - Cassandra, Hadoop Hbase
- Graph databases (based on graph theory)
 - Neo4J, ArangoDB
- Document stores
 - MongoDB, LDC Via, Couchbase, CouchDB, Azure DocumentDB, Cloudbant

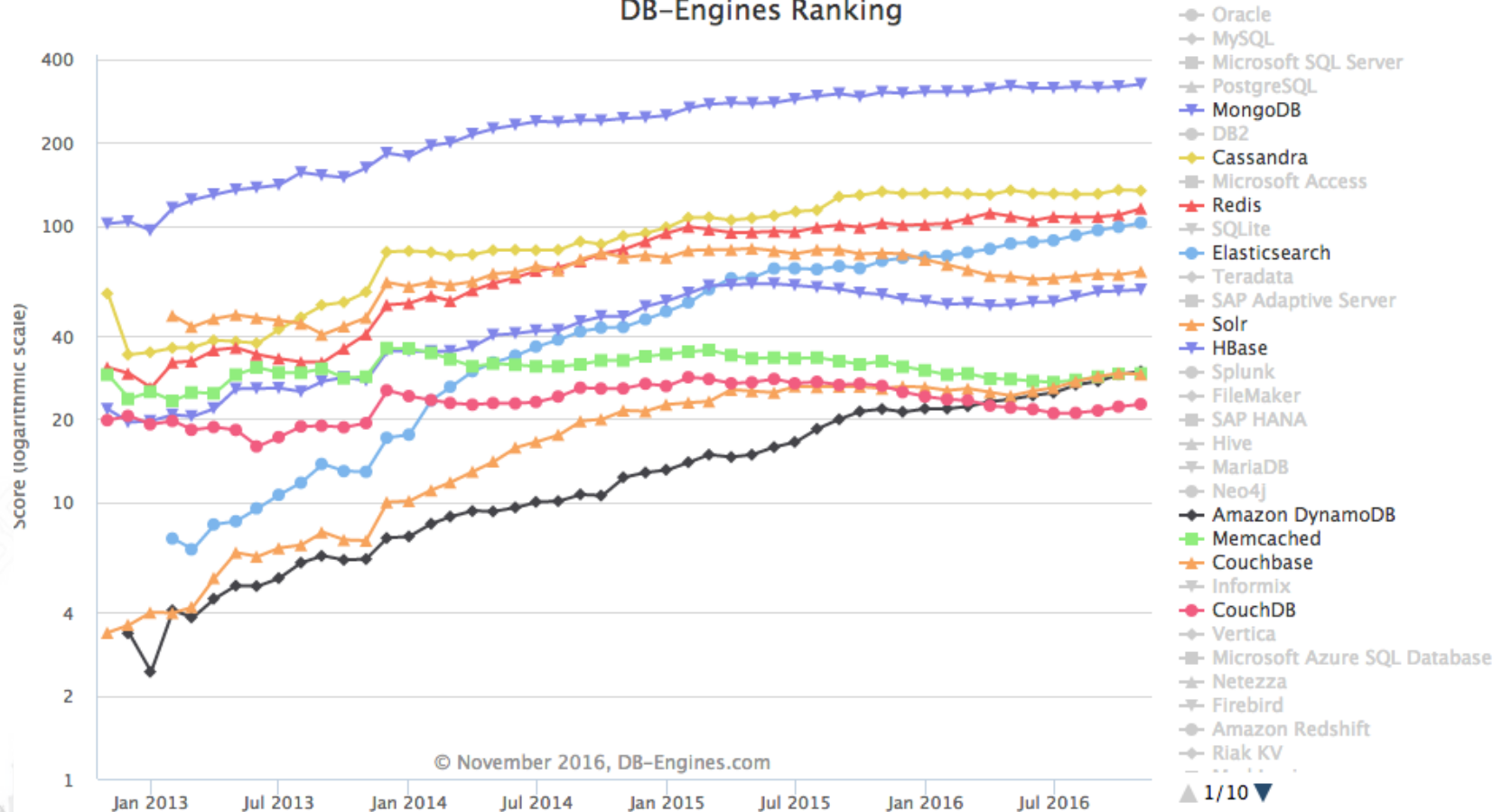
NoSQL Document Store

"Designed for storing, retrieving, and managing document-oriented information, also known as semi-structured data. Document-oriented databases are one of the main categories of NoSQL databases, and the popularity of the term "document-oriented database" has grown with the use of the term NoSQL itself. XML databases are a subclass of document-oriented databases that are optimized to work with XML documents."

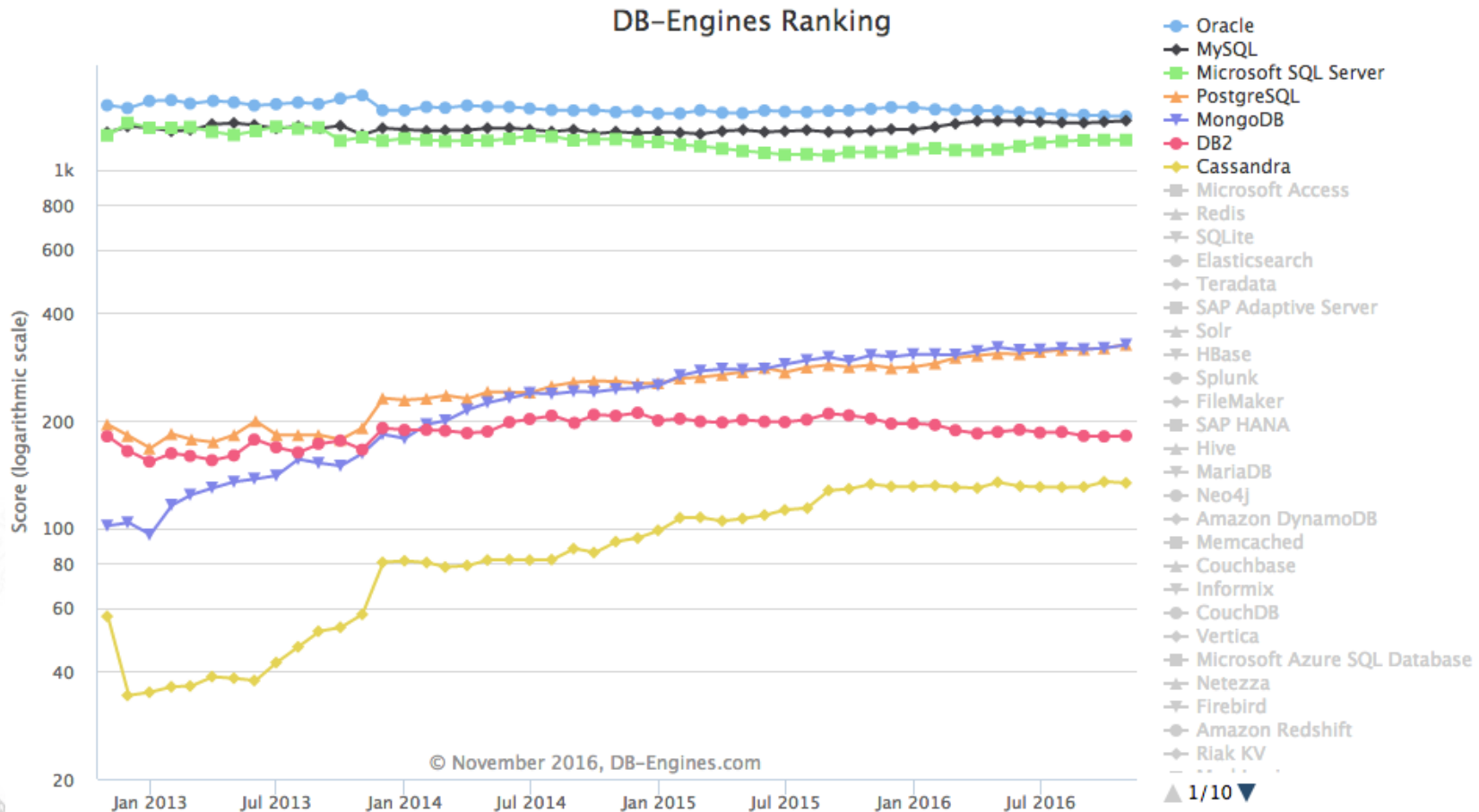
- Document-centric
- Familiar concepts
- Most popular

NoSQL databases

DB-Engines Ranking



NoSQL vs SQL databases



Where does LDC Via fit?



About LDC Via

- NoSQL “Document store” database
- Specifically optimised for Domino migrations
- Cloud-based Platform-as-a-Service
 - Ireland, Switzerland (US soon)
 - On-premises also available
- Simple tiered pricing model
- Built on top of MongoDB

A screenshot of the LDC Via pricing page, showing five pricing tiers: Basic (FREE), Standard (£199 p/m), Pro (£499 p/m), Enterprise (£999 p/m), and Dedicated (£ Contact us). Each tier lists its features, such as document limits, database counts, storage, and support options. The background of the pricing page shows a faint London skyline with the London Eye.

Basic FREE <ul style="list-style-type: none">500 documentsUp to two databases1GB of storageFull-text indexAll basic templatesSupport forums Sign Up	Standard £199 p/m <ul style="list-style-type: none">100,000 documentsUnlimited databases10GB of storageFull-text indexAll basic templatesSupport forumsStandard back-up Sign Up	Pro £499 p/m <ul style="list-style-type: none">500,000 documentsUnlimited databases10GB of storageFull-text indexAll basic templatesSupport forumsOnline supportPro back-up Sign Up
Enterprise £999 p/m <ul style="list-style-type: none">Unlimited documentsUnlimited databases100GB of storageFull-text indexAll basic templatesSupport forumsOnline supportPhone supportEnterprise back-up Sign Up	Dedicated £ Contact us <ul style="list-style-type: none">Unlimited documentsUnlimited databasesUnlimited storageFull-text indexAll basic templatesSupport forumsDedicated serversOnline supportPhone supportTailored back-up Sign Up	

About LDC Via

- Domino-like functionality
 - Documents
 - Response hierarchies
 - Readers/Authors fields
 - Rich text
 - File attachments
 - Doclinks
 - Dirty/inconsistent data

About LDC Via

- Migration tools
 - Web-based tool for web-facing servers
 - Installable tool
- Simplest and quickest data migration
 - No cleaning or remodelling
 - Data is recognisably “the same”
 - You can focus on developing the application

About LDC Via

- RESTful API
 - JSON-based
 - Covers all aspects of LDC Via including user management
- Sample code
- KEEP.WORKS

USER MANAGEMENT
Search Collection Data
DATABASE
All databases
Database Tags
Databases with tag
Delete Database
Get Database Collections
Get Database Details
Set Database Details
Stats (Multiple Databases)
Stats (Single Database)
Stats (Single Database)
DOCUMENT
Delete Document
Get Document Collection
Get Document Location
Get Document
Get Responses
Insert Document (array or no UNID)
Insert Document (with UNID)
Undelete Document
Update Document
DOMAIN SEARCH
Get Results
Get list of domain searches
Start Domain Search
EXPORT
Export to EML
Export to Excel
Export to PDF
METADATA
Get MetaData
Reset MetaData
Update MetaData
USER
Delete Other User Details
Get Other User Details
Get User Details
Update Other User Details
UTILITIES
Get DocLink
Get Organisations
Get list of distinct values
Merge Two Collections

Document - Get Document Collection

1.0.20160824

Given a database, and an array of _unid values, return an array of document objects

POST

`https://eu.ldcvia.com/1.0/documents/:db`

Parameter

Field	Type	Description
arrayofunids	object[]	The list of _unid values that you want to get corresponding document objects for

Request:

```
{
  "docs": [
    {
      "_form": "main",
      "_unid": "BA4B82DE1880988862573C300078738"
    }
  ]
}
```

Success 200

Field	Type	Description
result	object[]	An object array containing response documents

Response:

```
[
  {
    "_id": "5729d4cebb68f05e647caa95",
    "From": "CN=Matt White/O=Exhilarate",
    "AbbreviateFrom": "Matt White/Exhilarate",
    "AltFrom": "CN=Matt White/O=Exhilarate",
    "ThreadId": "MWHE-9N4L4G",
    "MainID": "BEAC980A0192689380257D3800548C38",
    "AbFrom": "Matt White/Exhilarate",
    "Body": "\r\nWhen you migrate your data we also migrate file attachments. We provide a very simple URL structure to access those files in future",
    "NewsletterSubject": "Accessing File Attachments",
    "Subject": "Accessing File Attachments",
    "Categories": "API",
    "WebCategories": "API",
    "Abstract": "When you migrate your data we also migrate file attachments. We provide a very simple URL structure to access those files in future",
    "_href": "http://dev.ldc.com:80/demos/discussion.nsf/api/data/documents/unid/BEAC980A0192689380257D3800548C38",
    "_unid": "BEAC980A0192689380257D3800548C38",
    "_noteid": "1172",
    "_created": "2014-08-18T15:25:32Z",
    "_modified": "2014-08-18T15:25:40Z",
    "_authors": "CN=Matt White/O=Exhilarate",
    "_form": "MainTopic",
    "_readerrrole": "dev-londc-com-demos-discussion-nsf1462359269872-MainTopic-reader",
    "_authorrole": "dev-londc-com-demos-discussion-nsf1462359269872-MainTopic-author",
    "Body_parsed": "When you migrate your data we also migrate file attachments. We provide a very simple URL structure to access those files in future"
  }
]
```

About LDC Via

API example, to read a “collection”

`https://<endpoint>/collections/<database>/<collection>?<options>`

e.g.

`https://eu.ldcvia.com/1.0/collections/my-crm/person?count=30&sortasc=surname`

Pass authenticated credentials with each request: an apikey or a session cookie.

About LDC Via

- Data browser
- Standard templates
 - Discussion, TeamRoom, Document Library
 - Email (read-only)
- Export to Excel, PDF, EML
- LDC Via Lens: point-and-click simple apps

Summary

- Different meanings of “migration”
- Big-bang vs best-fit
- Analysis, planning, and proper project management
- Change management
- Lots of decisions to take
- NoSQL now mainstream
- It's an opportunity...



Useful Links

- Webinar recordings: "Introducing LDC Via" and "Developing with LDC Via"
 - <https://www.youtube.com> and search for "LDC Via"
- <http://nosql-database.org>
- <http://db-engines.com>
- <https://code.visualstudio.com>
- <https://vaadin.com>
- <https://nodejs.org>
- <https://github.com/reactjs>
- <https://angularjs.org>
- <http://ldcvia.com>
- <http://api.ldcvia.com>

Contact Me

- @woowar
- julian@ldcvia.com
- Download slides
<http://blog.ldcvia.com>

