

Data Engine Thinking

*FastChangeCo's journey
towards a fully flexible data solution*

Imprint

Data Engine Thinking

By Roelant Vos and Dirk Lerner

Published by

Vos & Lerner Data Engine Thinking GbR

Astrid-Lindgren-Straße 4

64331 Weiterstadt

Germany

<https://dataenginethinking.com>

info@dataenginethinking.com

Layout, typesetting, cover

Romi Klockau | [linkedin.com/in/romi-klockau](https://www.linkedin.com/in/romi-klockau)

Graphics and illustration

Matthias Seifert | www.matthias-seifert.com

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means —electronic, mechanical, photocopying, recording, or any information storage and retrieval system— without written permission from the publisher, except for brief quotations in reviews.

All trade and product names mentioned in this book are trademarks, registered trademarks, or service marks of their respective owners and should be treated accordingly.

Printing history

First edition, 2025

Copyright © 2025 by Roelant Vos and Dirk Lerner

ISBN 978-3-9827180-0-2

Print softcover edition

ISBN 978-3-9827180-1-9

Kindle Replica-eBook edition

ISBN 978-3-9827180-2-6

PDF edition

ISBN 978-3-9827180-3-3

Print hardcover edition

ISBN 978-3-9827180-4-0

Print hardcover limited edition

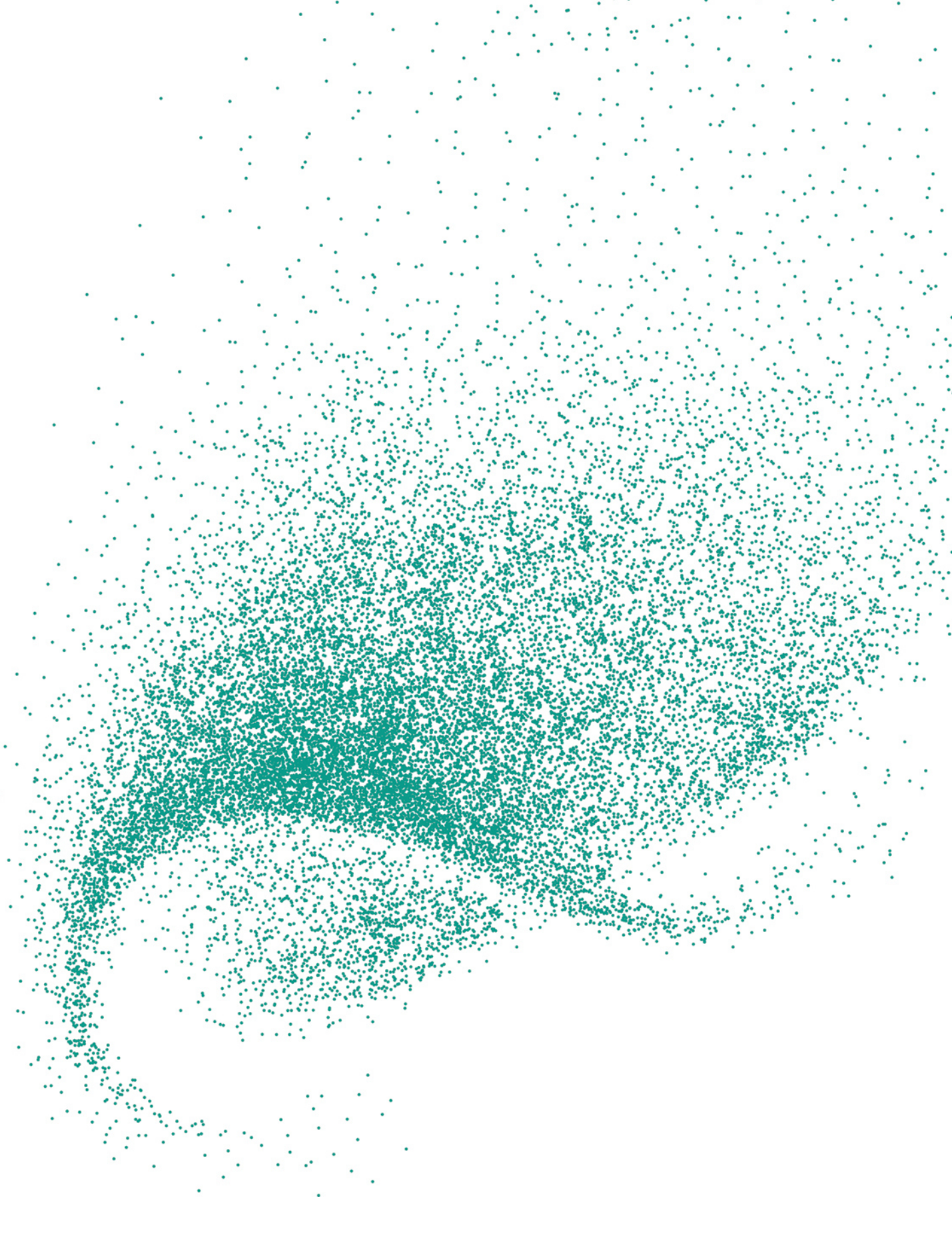
Exclusion of liability

The authors and publisher have taken great care in preparing this book, but they make no express or implied warranties regarding its content. They assume no responsibility for errors, omissions, or any incidental or consequential damages arising from the use of the information or programs contained herein.

The techniques, ideas, and recommendations in this book, including any associated code, are provided in good faith based on the authors' expertise and experiences. However, technology and individual circumstances vary, and the applicability of the provided solutions should be carefully assessed.

Readers are encouraged to use their own judgment and discretion when applying the concepts presented. While every effort has been made to ensure accuracy, the authors do not guarantee the effectiveness or suitability of these approaches for all scenarios.

The authors disclaim any liability for unintended consequences, errors, or omissions that may result from applying the book's content. By using the information in this book, readers acknowledge and accept this disclaimer, recognizing the importance of individual responsibility in its application.



Contents

00 Introduction	2
Introducing FastChangeCo	2
FastChangeCo disclaimer	2
The breadcrumbs of human activity	3
01 FastChangeCo—the world of data	6
Data silos	7
What is wrong with our data?	9
The need for a data initiative	11
Starting with the data initiative	12
Vision and strategy	14
02 the world of data	20
Fundamental assumptions when dealing with data	20
A machine designed for change	21
The role of a data solution	22
Solutions for ongoing data interpretation	23
A paradox of complexity	24
..... Separation of concerns	25
..... Data solution virtualization	26
..... Engine Thinking	27
..... Design patterns	28
..... Design metadata	28
..... Persistent staging	29
..... Information model	31
..... Code generation templates	32
..... Data logistics process control	33
..... Version control	34
..... Refactoring	35
..... Notification	37
..... Deployment automation	38
..... Testing	39
..... Business logic	41
..... Optimizer	41

03 FastChangeCo—Understanding the landscape ..	46
The birth of the DMCE	47
Celebrations	47
The data landscape draft	48
..... Learning from past mistakes	48
..... Moving business logic downstream	49
..... Marbles, data, and hubs	50
..... A first draft	51
Aligning the data landscape to the vision and strategy	52
..... The initial investigation	52
..... Moving forward	54
Assess data?	56
..... Nothing is as it seems	57
..... Through knowledge one attains humanity	59
..... Separation of concerns	60
..... Exploring reality	63
Preparing to start the engine	64
..... Bringing Jeff on board	65
..... The DMCE engine	66
..... Marketing	67
Introducing the Data Quadrant Model	68
04 The Data Quadrant Model	70
Supply and demand	71
Development style	73
The Data Quadrant Model	74
..... Data quadrant I	75
..... Data quadrant II	76
..... Data quadrant III	77
..... Data quadrant IV	78
Applications of the Data Quadrant Model	79
..... Data production	79
..... Accountability	80
..... Governance	82
..... People and skills	84
..... Technology	85
..... Data modeling	85
..... Development	87
05 FastChangeCo—Finalizing the data landscape ...	90
The Data Quadrant Model	90
An analogy	91
The four data quadrants	92
..... Data Quadrant I—the quadrant of the order	93
..... Data Quadrant II—the quadrant of the context	94
..... Data Quadrant III—the quadrant of unordered	96
..... Data Quadrant IV—the quadrant of innovation	97
Maturity of data management	99
Research the solution architecture	100



06 Data solution architecture 102

Purposes of a data solution	102
..... Making data accessible	102
..... Integrating data	104
..... Mapping the data to the information model	104
Architecture objectives and principles	105
..... Establish trust	105
..... Be unbreakable	106
..... Enable flexibility in orchestration	106
..... Be an institutional memory	107
..... Respond to changing business requirements	107
..... Refactor without impacting consumers	108
..... Manage data quality and interpretation	108
..... Implement a modular design	108
Architecture overview	109
..... Defining a data solution architecture	110
..... Design patterns	112
..... Solution patterns	112
..... Pattern structure	113
..... An overview of the data solution design	113
..... The staging layer	118
..... Areas of the staging layer	119
..... Landing area	119
..... Persistent staging area	119
..... Objectives and requirements for the staging layer ..	120
..... Agreeing on data interfaces	121
..... Data interface schema	122
..... Data interfacing mechanisms	123
..... Incorporating business logic in interfaces ..	129
..... Determination of the data delta	129
..... Recording the data origins	131
..... Setting the inscription timestamp	132
..... Logging the record order	132
..... Setting the source timestamp	133
..... Evaluating logical deletes	134
..... Streamlining data types	135
..... The integration layer	136
..... Areas of the integration layer	137
..... Base integration area	138
..... Derived integration area	138
..... The main integration layer entity types	139
..... Core business concept	139
..... Identifying core business concepts	140
..... Abstraction—subtyping and supertyping ..	141
..... The business key	141

..... Natural business relationship	145
..... Granularity	146
..... Role playing	148
..... Recursive structures and hierarchies	151
..... Context entity	153
..... Objectives and requirements for the integration layer ..	154
..... Mapping data to the information model	155
..... Defining core business concepts	156
..... Distributing surrogate keys	157
..... Managing relationships	157
..... Managing history	158
..... Prepare the assertion and state timelines	160
..... Handling changes in data structures	162
..... Implementing organization-wide business logic ..	162
..... The presentation layer	163
..... Areas of the presentation layer	164
..... Performance area	164
..... Access area	165
..... Objectives and requirements	
..... for the presentation layer	165
..... Transform data to the delivery format	165
..... Granularity	167
..... Implementing specific business logic	167
..... Calculation of timelines and 'date math'	168
..... Displaying data changes over time	171
..... Supporting frameworks	172
..... Data logistics control	173
..... Orchestration	183
..... Testing	184
..... Applying the testing framework for monitoring	185
..... Monitoring	186
..... Example architectures	187
..... The reference	188
..... The traditional	188
..... The better safe than sorry	189
..... The judger	190
..... The go for broke	190
..... The minimalist	191
..... The hedger	192
..... The virtual deliver	192
..... The all-in virtual	193
..... The Lambda	194
..... The Kappa	195
Closing notes on the solution architecture	196
..... Fundamental principles for data logistics	196
..... About referential integrity	198
..... Exception handling	199
..... The semantic layer	201

07 FastChangeCo—Setting up the project 204

Enforcing referential integrity?	205
Benefits of the Audit Trail Id	206
MVP or PoC?	208
Defining an MVP.....	209
What is the best use case?	209
The 'Wearables' use case	210
The 'ReCol' use case	211
Deciding on a use case	213
Migrating an existing report	213
Decide for a solution architecture	214
Data hub, or not?	215
Log data hub	215
Raw data hub	215
Information data hub	215
Presentation data hubs	215
Innovation and research data hubs	216
Missing data hubs	217
Either, or both?	217
How to resolve architecture dependencies?	218
Decoupling source systems and staging layer	218
Decoupling the integration layer	219
Decoupling the presentation layer	221
Decoupling decision	222
The need for automation	222
How to data model	223
Leveraging data models	223
The information model	223
The physical data model	224
Formalizing the department	225
Movers and shakers	227
Researching how to get the data in	230

08 Metadata, automation, and code generation 232

The DNA of a data solution	232
Automating the delivery of data logistics processes	233
A schema for data solution automation	234
Being implementation agnostic	235
Introducing data objects, items and mappings	237
Schema definition	238
Data objects, items and queries	239
Supporting objects	242
Mapping objects	245
Extensions	248
Data object mapping lists	249
Lineage and graphs	250
Storage considerations	253
Compilation and code generation	254
Styles of code generation	254
Ways of code generation	256
Templating engines	256
Alternatives to templating engines	261
Using APIs and SDKs	261
Domain-Specific Languages	261
Version control	264
Git, in a nutshell	264
What code to version?	265
Versioning the data model	265
Data solution versioning	266
Deployment automation	267
Builds for a data solution	268
Environments	268
Production	268
Development	268
Integration	269
Environment considerations	269
The deployment way of working	271
Branching strategy	271
Main branch	272
Development branch	273
Integration branch	273
Feature branches	273
The development process	274
Creating a new feature branch	275
What needs to be in a change?	276
Pull requests	277
Continuous integration	278
Releases	279
What happens when deployment fails?	279

..... Automating deployment	280	09 Getting the data in	296
..... Pausing the data logistics processes	281	Staging layer implementation	296
..... Generating and deploying the data structures	281	 Checksums	297
..... Generating and deploying the data logistics	283	 How to create a checksum	298
..... Starting the data logistics processes	283	 Hash algorithm	300
..... System- and regression testing	284	 Compute and storage	300
..... Production release	284	 Collision	301
..... Further considerations	285	 Is stronger also better?	302
..... Refreshing development	285	 Other considerations	302
..... Hot fixes	286	 Comparing columns individually	302
Data solution virtualization	286	 String conversion	303
..... Reducing barriers	287	 NULL handling	303
..... Layers of views	287	 Padding	303
..... Not the same as data virtualization	288	 Sanding	304
..... Performance considerations	289	 Landing area	304
..... Are layers still needed?	290	 Landing area data object structure	305
..... Managing change and backwards compatibility	290	 Inscription timestamp	306
..... Data integration technology for a virtualized solution	293	 Inscription record id	307
		 Source timestamp	307
		 Change data indicator	310
		 Audit Trail Id	311
		 Descriptive columns	312
		 Mapping exceptions	312
		 Data type alignment	313
		 String padding	313
		 Data type conversion defaults and precision	313
		 Using Unicode	314
		 Exceptions	314
		 Optional columns	315
		 Landing area design pattern	316
		 Considerations	317
		 What columns do I really need?	317
		 Selecting the delta mechanism	318
		 Restartability and recovering from failure	319
		 Timezones	319
		 Applying sort order	319
		 Managing data type interpretation from text files	320
		 Loading data delta directly into the PSA	321
		 Avoiding data quality checks in the staging layer	321
		 Ensuring completeness	321
		 Automation	322
		 Design metadata example	322
		 Solution pattern example	324
		 Persistent staging area	325
		 Persistent staging area data object structure	325
		 Inscription timestamp	326
		 Inscription record id	326
		 Source timestamp	327
		 Change data indicator	327
		 Audit Trail Id	327
		 Optional columns	327

09 Getting the data in

..... Persistent staging area design pattern	328	 Considerations	373
..... Considerations	328	 Degenerate columns	373
..... Defining a primary key	328	 Deciding how to implement transactions	373
..... Reusing or recalculating checksums	329	 Same-As Links	376
..... Physical storage	329	 Filtering	377
..... Schema drift	329	 NULL handling for Links	378
..... Batching inscription timestamps	331	 Automation	379
..... Automation	331	 Design metadata example	379
..... Design metadata example	331	 Solution pattern example	381
..... Solution pattern example	333	 Contextual data	382
..... Staging layer orchestration	336	 Satellite data object structure	383
Integration layer implementation — back room	338	 Surrogate key	383
..... A brief introduction to Data Vault	339	 Inscription timestamp	384
..... What is Data Vault?	339	 Inscription record id	385
..... Data Vault table types	339	 State timestamp	385
..... Unified decomposition	340	 Audit Trail Id	386
..... Core business concepts	342	 Descriptive columns	386
..... Hub data object structure	342	 Optional columns	386
..... Hub surrogate key	343	 Source timestamp	386
..... Business key column(s)	344	 Record source indicator	387
..... Audit Trail Id	345	 Change data indicator	387
..... Optional columns	345	 Checksum	387
..... Hub design pattern	346	 Satellite design pattern	387
..... Considerations	347	 Considerations	388
..... Complex business keys	347	 Redundancy, compacting, and change merging	388
..... Composite keys	347	 Redundant data	389
..... Concatenated keys	348	 Column scope	91
..... Pivoting	350	 Reasons for applying record compacting	392
..... Transformations and hard-coded values	351	 Where to apply record compacting	393
..... Storing the business key in Satellites	352	 'On the fly' compacting	394
..... Our business key pattern	357	 Post process compacting	402
..... Surrogate key approaches	357	 Compacting and the engine	404
..... Sequence value keys	357	 How many context entities do we need?	404
..... Hash value keys	359	 Satellite splitting	404
..... Natural business Keys	361	 Adding or removing columns	
..... Final thoughts on surrogate key approaches	362	 in an existing Satellite	406
..... Zero keys	363	 Managing logically deleted records	406
..... Automation	364	 Struggling with time—while getting the data in	410
..... Design metadata example	365	 Clock ticks	411
..... Solution pattern example	366	 The Allen Relationships	412
..... Solution pattern example—virtual	367	 Open or closed?	413
..... Natural business relationships	368	 Mapping the state timestamp	415
..... Link data object structure	368	 Receiving and integrating data	
..... Link surrogate key	368	 from multiple time zones	416
..... Hub surrogate keys	370	 Insert only, or not?	417
..... Audit Trail Id	372	 End-dating	418
..... Optional columns	372		
..... Link design pattern	372		

..... Modeling time in Satellites	422
..... Nontemporal	422
..... Unitemporal	423
..... Visualizing (uni)temporal data	424
..... Late-arriving transactions	428
..... Merging unitemporal Satellites into a bitemporal data set	432
..... Replaying history	434
..... Combining the data	437
..... Looking in detail	441
..... Time travel	444
..... Tackling bitemporal complexities upfront	446
..... Viewing late-arriving transaction as bitemporal	447
..... Bitemporal	448
..... Receiving a complete time period	449
..... Receiving an incomplete time period	456
..... Interpreting Closed-Closed time periods	458
..... Where to implement temporal design?	460
..... Current record indicators	461
..... Multi-variance	461
..... Adding a multi-variant key to a Satellite	464
..... Adding a separate object for each multi-variant instance	465
..... Creating a dependent Hub	466
..... Introducing a peg-legged Link	467
..... What to look out for when designing multi-variance	468
..... How to select a solution approach	468
..... Exception bitmap	468
..... Satellites for Links	471
..... Link-Satellites, when, and how?	472
..... Driving keys and relationship end-dating	477
..... Driving keys and insert-only	480
..... NULL handling for Satellites	484
..... Combining multiple data sources in a Satellite	484
..... Automation	486
..... Design metadata example	486
..... Solution pattern example	488
..... Other integration layer considerations	499
..... Managing reference data	499
..... Adding reference data	500
..... Unconnected Hubs	500
..... CBC, or not?	501
..... Orchestration	502
..... Integrity processing, for a reason	503
..... Batch design, dependencies, and integrity	506

..... Continuous parallel processing	508
..... State machine	508
..... Positioning the PSA	509
..... Batch-based PSA design	510
..... Creating load windows	513
..... Interpreting load windows	515
..... Eventual consistency	517
..... Deriving the consistency timestamp	517
..... Moving closer to the Engine	522
..... Race conditions	523
..... Joining objects in the PSA	525
..... Customer, Category, Category Group	525
..... Working with alternate business keys	530
..... Testing	533
..... Essential integration layer controls	533
..... Uniqueness	533
..... Completeness in temporality	535
..... Referential integrity	537
..... Notification	538
..... Back room and front room	539

10 FastChangeCo—getting the data in 542

MVP—Increasing time to value	542
..... The road to automation	543
..... Minimizing environments	543
..... Deployment on time	545
Data modeling for the MVP	545
..... Information model	547
..... Data Vault data model	549
..... Managing metadata	549
Building the staging layer	550
..... Landing area	550
..... Analyzing the source data objects	551
..... Analyzing columns	551
..... Persistent staging area	551
..... Pre-calculation of business keys hash values	552
..... Duplicate data records	553
Building the integration layer	555
..... Data logistics processes	556
..... Deciding for a timeline	556
..... Reject tables	559
Bringing the MVP to production	560

11 Getting the data out 564

Integration layer implementation — front room	564
Why have a derived area when you have a PSA?	565
Where to apply business logic	565
Raw Vault, Business Vault	566
Translating intent to code	567
The business rule bitmap	567
Documentation of business rules	568
Documentation of implementation rules	569
Implementation	569
Considerations	570
Implementing a business rule as a view	570
Parallel processing	570
What timestamps to pick when creating new data?	571
User managed landing objects	571
Example — reference data	572
Example — indexing	574
Presentation layer implementation	576
Struggling with time — while getting the data out	576
Timelines and snapshots	577
Timelines	577
Joining two time-variant objects	577
Combining multiple time-variant objects	580
Compacting in the presentation layer	582
Combining data with different granularity	584
Filtering	587
Frequency compacting	587
Continuous gap compacting	589
Temporal data selection	591
Snapshots	593
Combining data	593
Granularity and redundancy	593
Fidelity	595
Filtering	596
Temporal data selection	597
Performance area	597
Objects in the performance area	598
Performance area data object structure	598
Timestamp	600
Hub surrogate key(s)	600
Audit Trail Id	601
Satellite pointer(s)	601
Optional columns	601
Key	601
Descriptive columns	602
Performance area design pattern	602
Considerations	603
Compacting, again?	603

Snapshot maintenance	603
Bridges	604
Do we need both Hub and Satellite keys?	605
Automation	606
Design metadata example	607
Solution pattern example	611
Access area	615
Access area data object structure	615
Dimension key(s)	616
Audit Trail Id	617
Descriptive columns	618
Optional columns	618
Checksums	618
From and before timestamps	618
Access area design pattern	619
Considerations	619
Types of history	619
Conditional updates	620
Applying remaining business logic updates	620
Automation	621
Design metadata examples	621
Considering performance area objects to support the access area	640
Performance objects in virtual solutions	641
Bringing it all together	641

12 FastChangeCo — Getting the data out	648
The MVP's dimensional model	648
Building the access layer	649
.... Data validation	649
.... Documenting business rules	650
.... Designing the RelCol data mart	654
.... Dimension identifiers	655
.... Performance optimization	656
The new report	658
13 Completing the engine	660
Connecting the information model	660
Beyond DevOps	665
Starting the engine	665
14 FastChangeCo — The Resurrection	668
Prototyping the engine	669
Team triumph	670
Looking forward	672
15 Data Engine Thinking — taking the next step	674



Data Engine Thinking *— taking the next step*

Ready to put Data Engine Thinking into action?

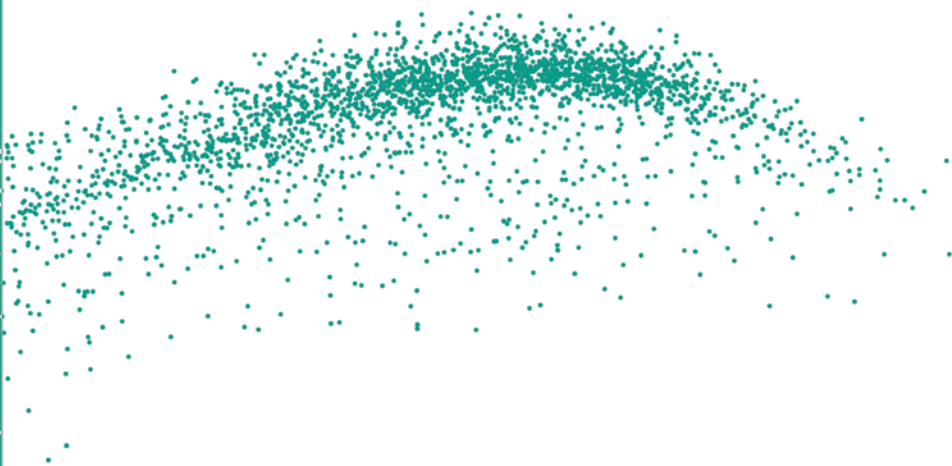
You've explored the principles — now it's time to bring them to life in your organization. Whether you're just getting started or ready to scale, we're here to help you go further, faster.

As your journey continues, we invite you to connect on <https://dataenginethinking.com>.

Our website has the latest information about the services we provide, including:

- Our training schedule
- Our coaching opportunities, including individual and team-based coaching
- Talent mentoring
- Assessments and reviews
- Consultancy

Let's unlock the full potential of your data together.



Data Engine Thinking

dataenginethinking.com

Data Engine Thinking covers the end-to-end methodology to deliver a data solution that is truly designed to adapt to progressive understanding - and ultimately meet the business' needs.

- Design and implement a solution that is designed for change
- Solve real-world problems encountered when working with data
- Fully automate your delivery

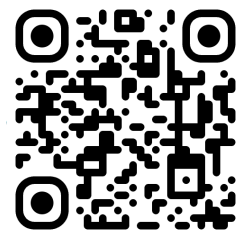
*Roelant Vos &
Dirk Lerner*

The Authors

Having worked as a consultant, trainer, software vendor, and decision maker in the corporate world over the years, **Roelant** has observed data management from many different points of view.

 roelantvos.com

Get your
copy here



Dirk is an experienced independent consultant and managing director of TEDAMOH. He is considered a global expert on BI architectures, data modeling and temporal data.

 dirklerner.com

