Deelnemerslijst

Ontwerpen van een Nieuwe Data Architectuur
11 en 12 november 2025

BEDRIJF	NAAM DEELNEMER		FUNCTIE
Bizzin	Jan Nanne	de Jong	it-information-management-consultant
Certis Belchim	Niels	Rennen	information-manager
Certis Belchim	Steven	Van der Ven	systems-designer
Dienst Toeslagen	Sebina	Rosbergen-Heida	data architect
Equibis bv	Koen	Van Waeyenbergh	bi-specialist
Gemeente Westland	van der	Hoek	data architect
Greenchoice B.V	Didy	Bos	it-manager
Greenchoice B.V.	Jeroen	Kneppers	data engineer
Greenchoice BV	Misja	Wilders	data architect
Greenchoice BV	Wim	Peters	datawarehouse-specialist
IND	Bianca	Schouten	data engineer
Nipro	Joris	Verbeiren	data architect
Rail & OV	Bert-Jan	Steerneman	other

Evaluatieformulier
Ontwerpen van een nieuwe Data Architectuur
11 en 12 november 2025

Naam: ____________________________ Bedrijf: ____

Wat is in een cijfer uitgedrukt uw beoordeling van (s.v.p. aankruisen):

		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1.	Dit seminar overall:	O	O	O	O	O	O	O	O	O	O

		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
2.	De presentatie van de spreker:										
	- Inhoudelijk :	O	O	O	O	O	O	O	O	O	O
	- Presentatie:	O	O	O	O	O	O	O	O	O	O

3. Voldeed het programma aan uw **verwachtingen**? **[ ] Ja [ ] Gedeeltelijk [ ] Nee, want**

____

4. Zou u dit seminar aanbevelen bij collega's of binnen uw netwerk? **[ ] Ja [ ] Nee, want**

____

5. Welke onderwerpen heeft u gemist of vond u onderbelicht?

____

6. Welke onderwerpen vond u overbodig of kregen te veel aandacht?

____

7. Welke overige op- of aanmerkingen of suggesties heeft u nog?

____

8. Organisatie en accommodatie. Hoe waardeert u de:

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Organisatie van de dag	O	O	O	O	O	O	O	O	O	O
Kwaliteit zaal, beeld en geluid	O	O	O	O	O	O	O	O	O	O
Bereikbaarheid / ligging	O	O	O	O	O	O	O	O	O	O

9. Hoe bent u hier gekomen? **[ ] Auto [ ] Openbaar Vervoer [ ] Anders,** ___________

10. Zou u een korte aanbeveling willen schrijven voor dit event, dat wij als testimonial kunnen tonen op onze website? Zo ja, schrijf deze dan hieronder:

____

____

____

Naam: ____

Bedrijf: ______________________________ Functie:____________________________________

S.v.p. inleveren bij de organisatie of mailen naar seminars@adeptevents.nl

Welcome

Adept Events

WHO WE ARE

AdeptEvents

DW&BI SUMMIT

BI-Platform.

RELEASE·

Werner Schoots

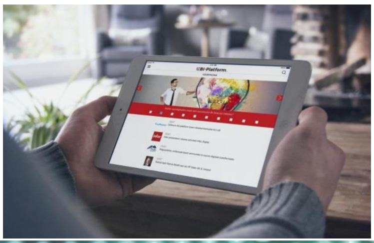
Founder Adept Events

BI·Platform.

- Launched in 2008 as online spin-off from Database Magazine (DB/M)
- Topics: Business Intelligence, Data Warehousing, Analytics, Data Management

News	*Job board*	*Selected Whitepapers*	*Events*
Articles	*Blogs*	*Video interviews*	*Cases*

- We welcome your input: redactie@biplatform.nl



www.biplatform.nl & weekly newsletter

@BIplatform on & YouTube

Download the BI-Platform App

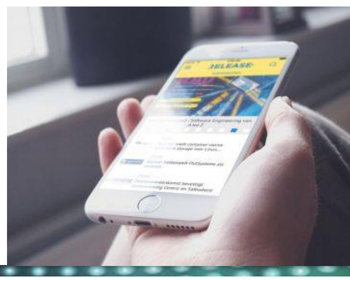
Join our LinkedIn Discussion Group

RELEASE·

- Launched in 1996 as Software Development spin-off from Database Magazine
- Topics: Software Engineering – Analysis, Design, Development, Testing and Deployment

News	*Job board*	*Selected Whitepapers*	*Events*
Articles	*Blogs*	*Video interviews*	*Cases*

- We welcome your input: redactie@release.nl



www.release.nl & weekly newsletter

@Release_nl on Twitter & YouTube

Download the Release App

Join our LinkedIn Discussion Group

SEMINARS

All seminars and workshops are organised twice a year

Alec Sharp	Business-oriented Data Modelling Masterclass Working with Business Processes Masterclass Concept Modelling for Business Analysts The Data-Process Connection (virtual half day session)
Panos Alexopoulos	Knowledge Graphs – pragmatic approach and best practices
Rick van der Lans	Ontwerpen van een Nieuwe Data Architectuur
Mathias Vercauteren	Data Governance Sprint
Nigel Turner	A Data Strategy for Becoming Data Driven Tackling Data Quality Problems (virtual half day session)
Chris Bradley / Winfried Etzel	Data Management Fundamentals
Lawrence Corr	Agile Data Warehouse Design & Dimensional Modeling
Christian Gijsels	Generatieve-AI in Business Analyse Cursus Sparx Enterprise Architect 16
Multiple speakers	*Data Warehousing & BI Summit – Yearly conference in March/April*

IN-HOUSE

All seminars and workshops can be organized in-company.
With local speakers and international speakers!

Please contact Werner Schoots

☎+31 (0)172 742680 ✉ seminars@adeptevents.nl

Guidelines for Designing New Data Architectures

CONSULTANCY

Rick F. van der Lans
Industry analyst
Email rick@r20.nl
Twitter @rick_vanderlans
www.r20.nl

AdeptEvents

Rick F. van der Lans

Rick F. van der Lans is a highly-respected independent analyst, consultant, author, and internationally acclaimed lecturer specializing in data warehousing, business intelligence, big data, and database technology. He is managing director of R20/Consultancy BV.

He has presented countless seminars, webinars, and keynotes at industry-leading conferences. Rick helps clients worldwide to design their data warehouse, big data, and business intelligence architectures and solutions and assists them with selecting the right products. He has been influential in introducing the new logical data warehouse architecture worldwide which helps organizations to develop more agile business intelligence systems.

He is the author of several books on computing, including his new *Data Virtualization: Selected Writings* and *Data Virtualization for Business Intelligence Systems*. Some of these books are available in different languages. Books such as the popular *Introduction to SQL* is available in English, Dutch, Italian, Chinese, and German and is sold world wide. He also authored numerous whitepapers for vendors.

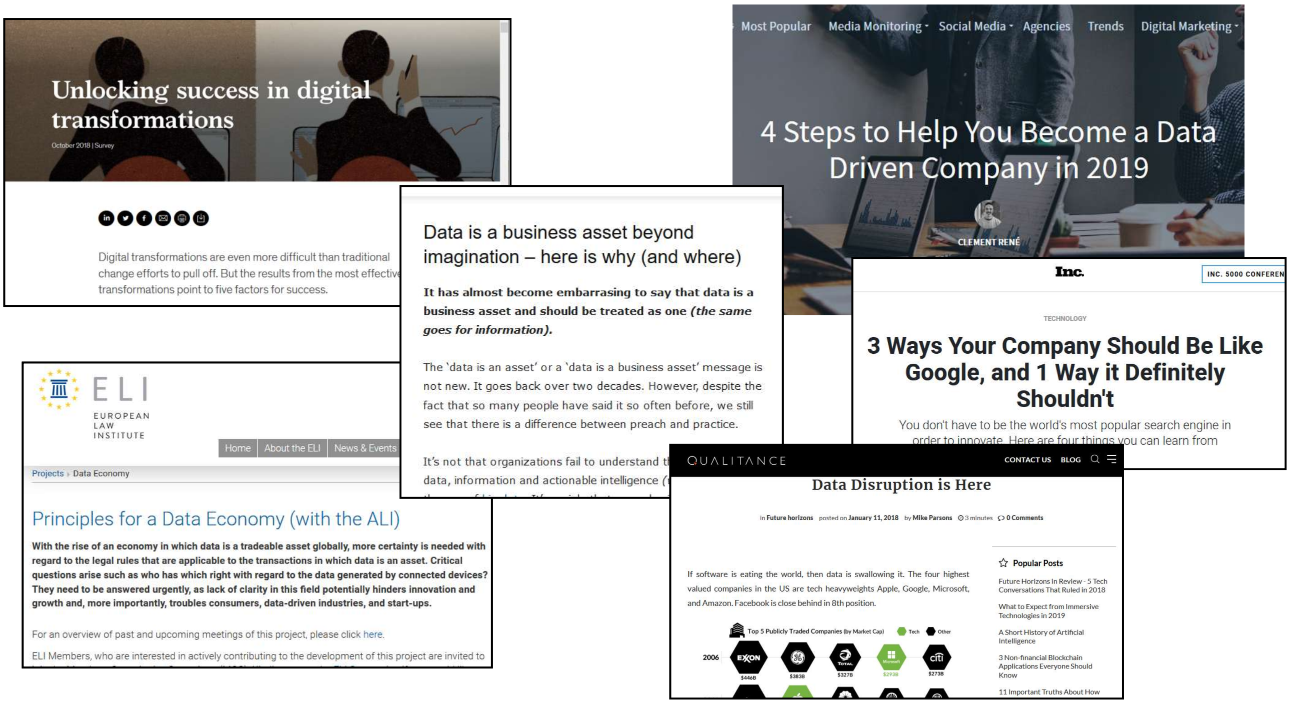
In 2018 he was selected the sixth most influential BI analyst worldwide by onalytica.com.

You can get in touch with Rick van der Lans via:
Email: rick@r20.nl
Website: www.r20.nl
LinkedIn: http://www.linkedin.com/pub/rick-van-der-lans/9/207/223

Agenda and Subjects

1. Introduction: Why a New Data Architecture?
2. Introduction to Data Architectures
3. Steps 1-3: Setting the Stage
4. Step 4: Analyze New Technologies for Data Storage, Processing, and Analytics
5. Step 5: Architectural Design Principles
6. Step 6: Reference Data Architectures
7. Step 7-8: Designing New Data Architectures
8. Steps 9-10: Final Steps
9. Closing Remarks

Part 1:
Introduction: Why a New Data Architecture?



Examples of 'Doing More' with Data

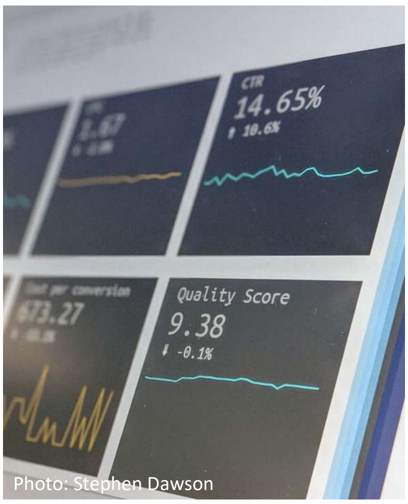


Photo: Stephen Dawson

- Enabling self-service reporting, dashboards, and for analytics to work with (near) real-time data
- Combining internal with external data to enrich analytical capabilities
- Accelerating AI/machine learning initiatives to discover new patterns or trends in the data
- Simplifying deployment of IoT technology to generate more data on the machines and business processes
- Offering edge analytics in which real-time data is analyzed continuously and near the place where the data is produced by the sensor or business process


"DATA IS THE NEW OIL."
Data is the new oil!
OIL
DATA
Data is the new oil
DATA
DATA
DATA
this is good news, right?
data is the new oil
we need to find it, extract it, refine it, distribute it and monetize it.
David Buckingham
DATA
is the new oil


BIG DAT

Data hasn't changed,
it's just more of the same

Data *consumption* has changed

Self-service BI
Embedded BI
Supplier- and Customer-driven BI
Applied AI in Text, Image, Video Analysis
Edge Analytics
Data Marketplace
Data Science
Automated decisions
…


Raising the Bar for ICT systems
& BAR


How Good is our Track Record?
7
6
5
Photo: Samuelk Blanck

Software Development Hall of *Shame*

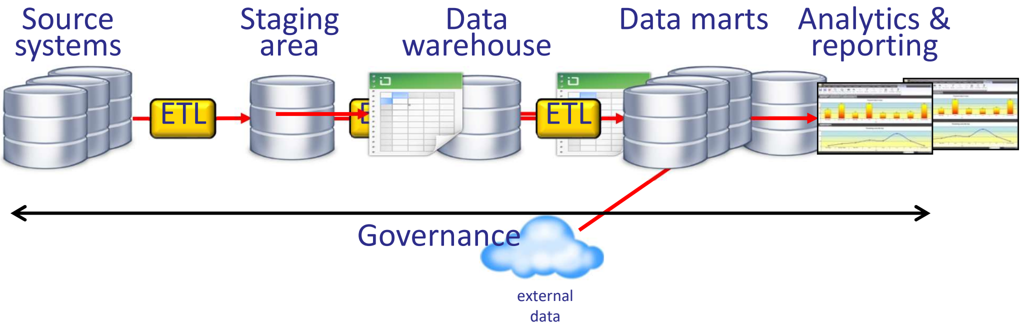
YEAR	COMPANY	OUTCOME (COSTS IN US $)
2005	Hudson Bay Co. [Canada]	Problems with inventory system contribute to $33.3 million* loss.
2004–05	UK Inland Revenue	Software errors contribute to $3.45 billion* tax-credit overpayment.
2004	Avis Europe PLC [UK]	Enterprise resource planning (ERP) system canceled after $54.5 million† is spent.
2004	Ford Motor Co.	Purchasing system abandoned after deployment costing approximately $400 million.
2004	J Sainsbury PLC [UK]	Supply-chain management system abandoned after deployment costing $527 million.†
2004	Hewlett-Packard Co.	Problems with ERP system contribute to $160 million loss.
2003–04	AT&T Wireless	Customer relations management (CRM) upgrade problems lead to revenue loss of $100 million.
2002	McDonald's Corp.	The Innovate information-purchasing system canceled after $170 million is spent.
2002	Sydney Water Corp. [Australia]	Billing system canceled after $33.2 million† is spent.
2002	CIGNA Corp.	Problems with CRM system contribute to $445 million loss.
2001	Nike Inc.	Problems with supply-chain management system contribute to $100 million loss.
2001	Kmart Corp.	Supply-chain management system canceled after $130 million is spent.
2000	Washington, D.C.	City payroll system abandoned after deployment costing $25 million.
1999	United Way	Administrative processing system canceled after $12 million is spent.
1999	State of Mississippi	Tax system canceled after $11.2 million is spent; state receives $185 million damages.
1999	Hershey Foods Corp.	Problems with ERP system contribute to $151 million loss.
1998	Snap-on Inc.	Problems with order-entry system contribute to revenue loss of $50 million.
1997	U.S. Internal Revenue Service	Tax modernization effort canceled after $4 billion is spent.
1997	State of Washington	Department of Motor Vehicle (DMV) system canceled after $40 million is spent.
1997	Oxford Health Plans Inc.	Billing and claims system problems contribute to quarterly loss; stock plummets, leading to $3.4 billion loss in corporate value.

Source IEEE Spectrum: September 2005

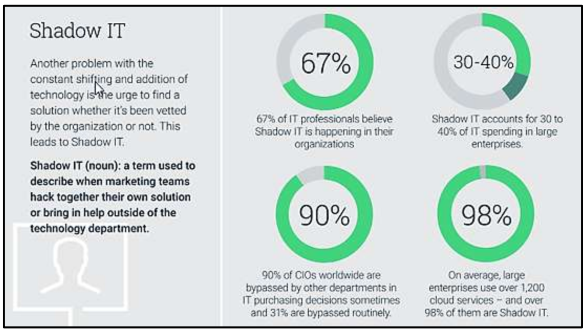
Enkele Mislukte ICT-Projecten Bij NL Overheid

- Minister Carola Schouten (Landbouw, CU) [besluit April 2019] te stoppen met de vernieuwing van de ICT-systemen bij de Nederlandse Voedsel- en Warenautoriteit. Na 65 miljoen euro te hebben uitgegeven, bleek er maar weinig te werken.
- De Belastingdienst, dat sinds 2005 probeerde een systeem te bouwen dat alle transacties van de fiscus zou verwerken. Na negen jaar en 203 miljoen euro gaven ze het op.
- Defensie, waar ze sinds 2002 bouwden aan 'Speer'. Na volgens eigen zeggen 433 miljoen euro te hebben uitgegeven – de Algemene Rekenkamer kwam op 900 miljoen euro uit – gaf het ministerie het in 2015 op. Speer was nog lang niet af, en werkte niet zoals bedoeld.
- Het nieuwe bevolkingsregister BRP. Daarvan werd de ontwikkeling in 2017 stopgezet, na tien jaar bouwen en 100 miljoen aan uitgaven.
- Digitalisering van de rechtspraak, die in april 2018 na zes jaar en ruim 200 miljoen euro (oorspronkelijk werden de kosten op 7 miljoen euro geschat) werd stopgezet.

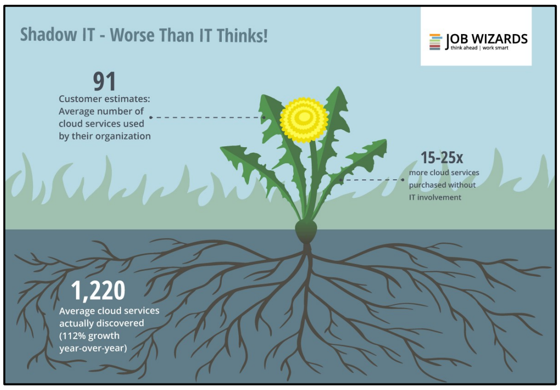
Is This Really the Entire Data Architecture?



Shadow IT

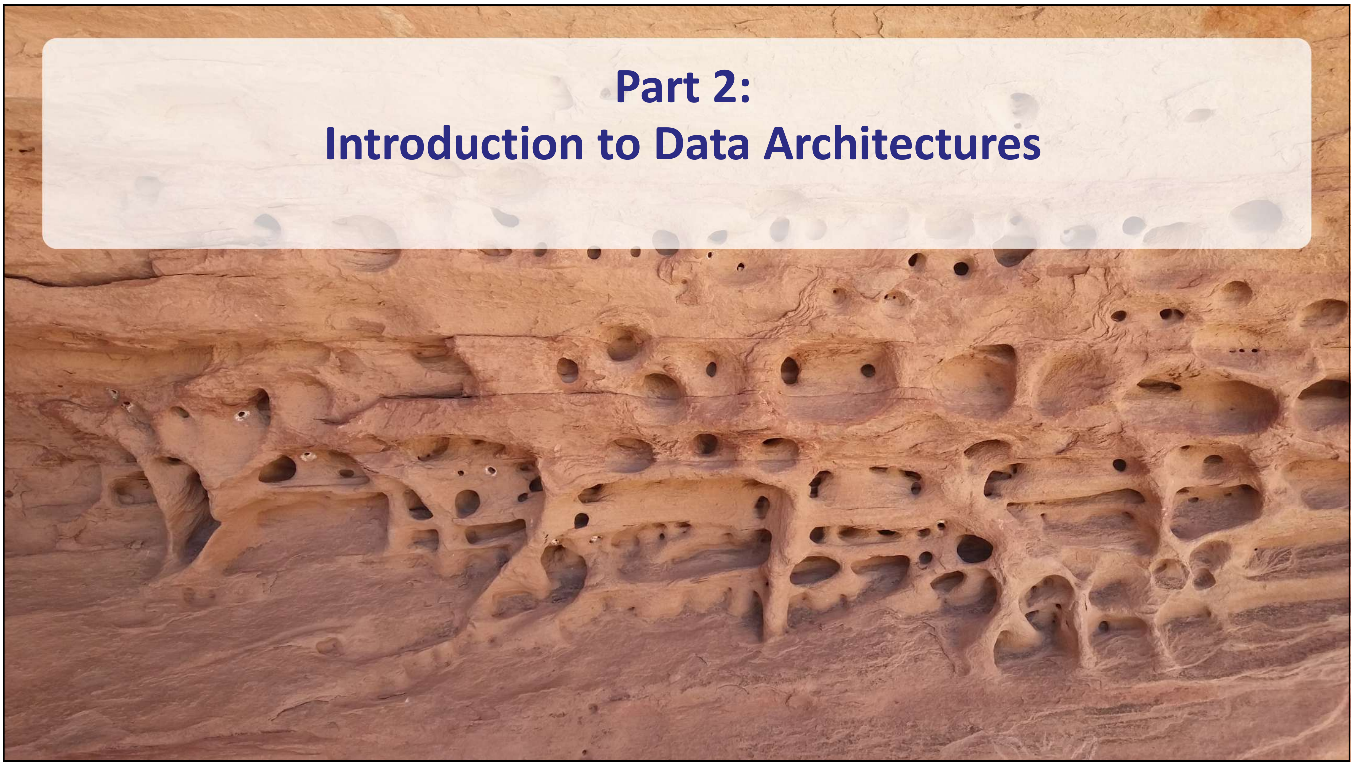


Source: https://www.emailvendorselection.com/building-a-consolidated-tech-stack/



Source: https://job-wizards.com/en/shadow-it-the-hidden-menace-for-every-company/

Part 2:
Introduction to Data Architectures



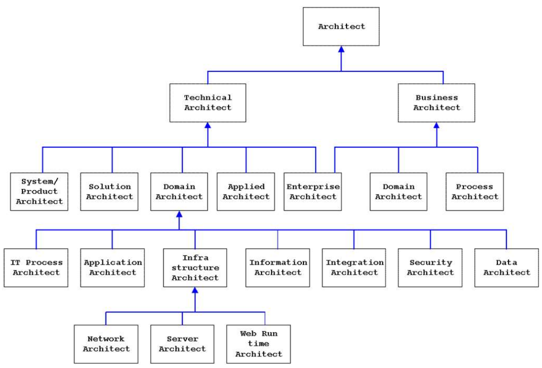
What is a Data Architecture?

- Wikipedia: A *data architecture* is composed of models, policies, rules or standards that govern which data is collected, and how it is stored, arranged, integrated, and put to use in data systems and in organizations.
- Examples of data architectures:
  - Data warehouse architecture
  - Data streaming architecture
  - Transactional system

Data Architects versus Solutions Architects

Data Architects	Solutions Architects
… focus on how information moves across the system from one application to another	… look at the overall technological environment of the company
… collaborate with clients to determine the specifications of the project, as well as the business goals that will align with the collected data	… meet with their clients and establish their specific technology needs based on their business objectives
… design the data model for the organization; where to store the customer data, how to retrieve the data; who can read the data	… has a more technical point of view. Do we select a cloud solution, or on premise? What will the network look like? How will everything be connected without failures?

So Many Different Types of Architects



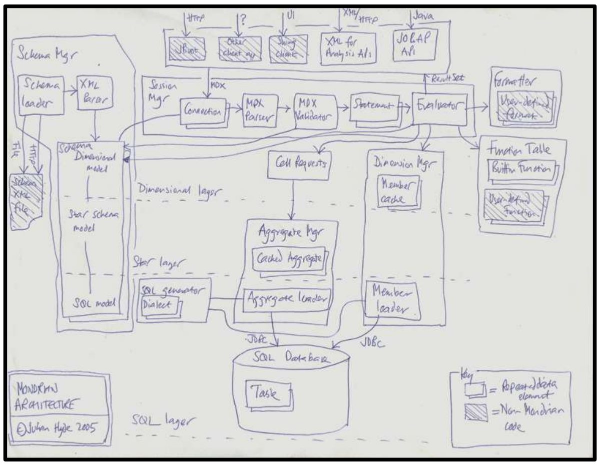
Paul Catalin Tomoiu : In [the] IT world, an architect is a person with enough knowledge to find a high level solution to a challenge in an IT environment.

The challenge type, defines the nature of the architecture role.

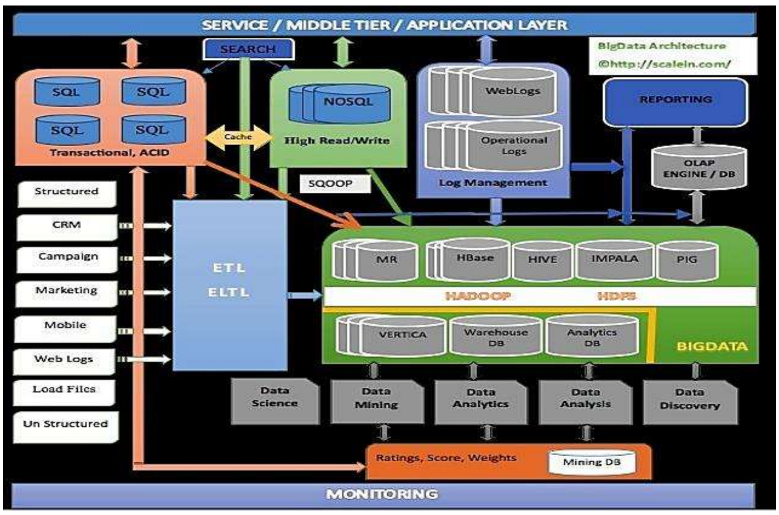
We can speak about an Enterprise Architecture, a Business Architecture, Data Architecture, Solution Architecture, IT Technical Architecture, Application Architecture, Software Architecture, Security Architecture, etc.

Source: https://blog.prabasiva.com/2008/08/21/different-types-of-architects/

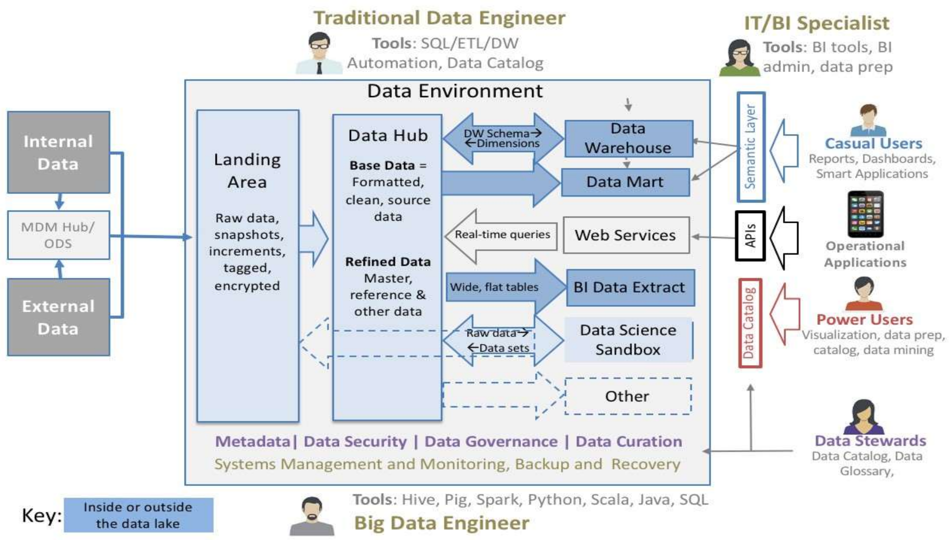
The Birth of a Data Architecture



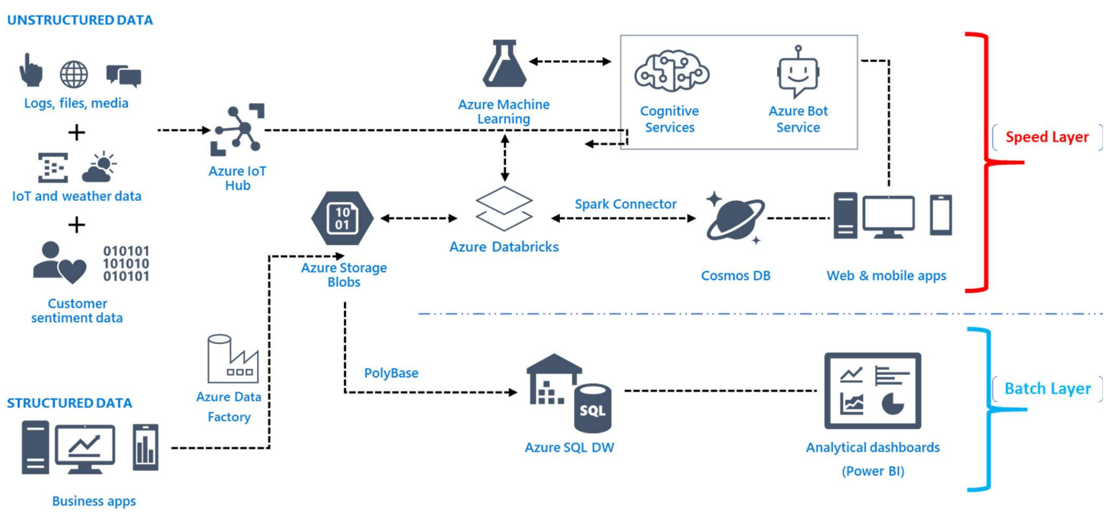
Example Data Architecture (1)



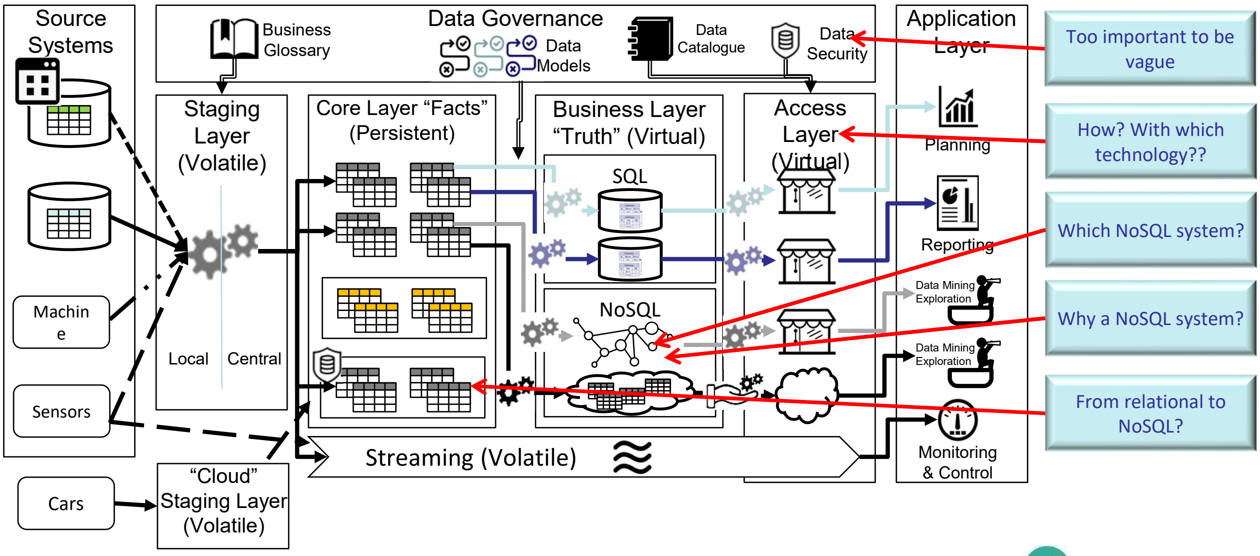
Example Data Architecture (2)



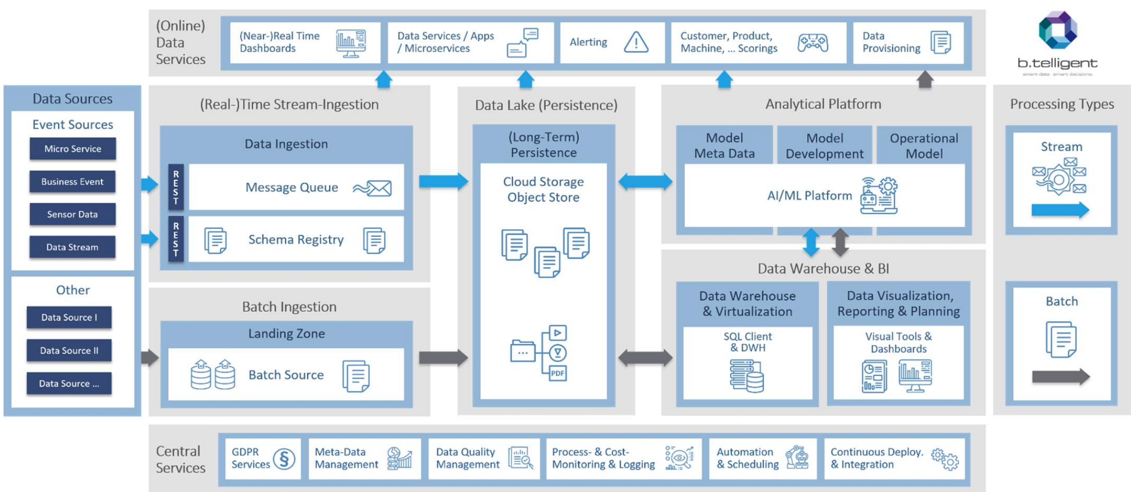
Example Data Architecture (3)



Example Data Architecture (4)

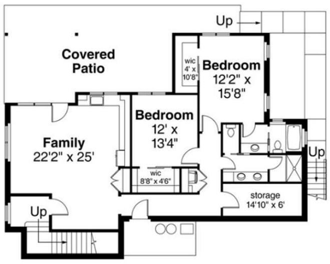


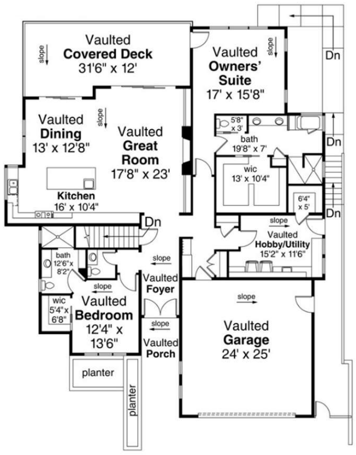
Example Data Architecture (5)



Source: https://www.btelligent.com

Architecture of a House





Architecture of a House (IT Style)

Electricity | Internet | Water drainage | Water supply

Community rooms
Living room
Dining room

Sleeping rooms
Parents room
Children's rooms

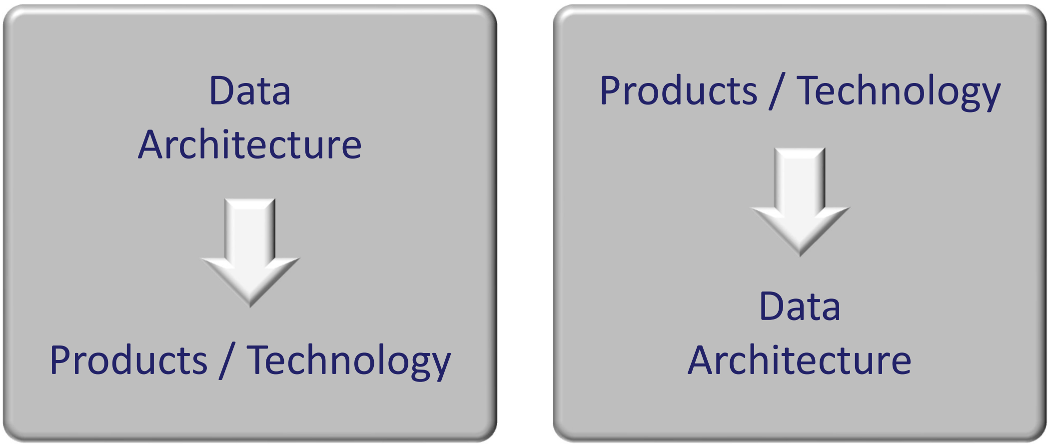
Wet rooms
Restrooms
Bathroom
Kitchen

Storage rooms
Garage
Walk-in closet
Basement

Relaxation rooms
Private film theater
Front garden
Back garden

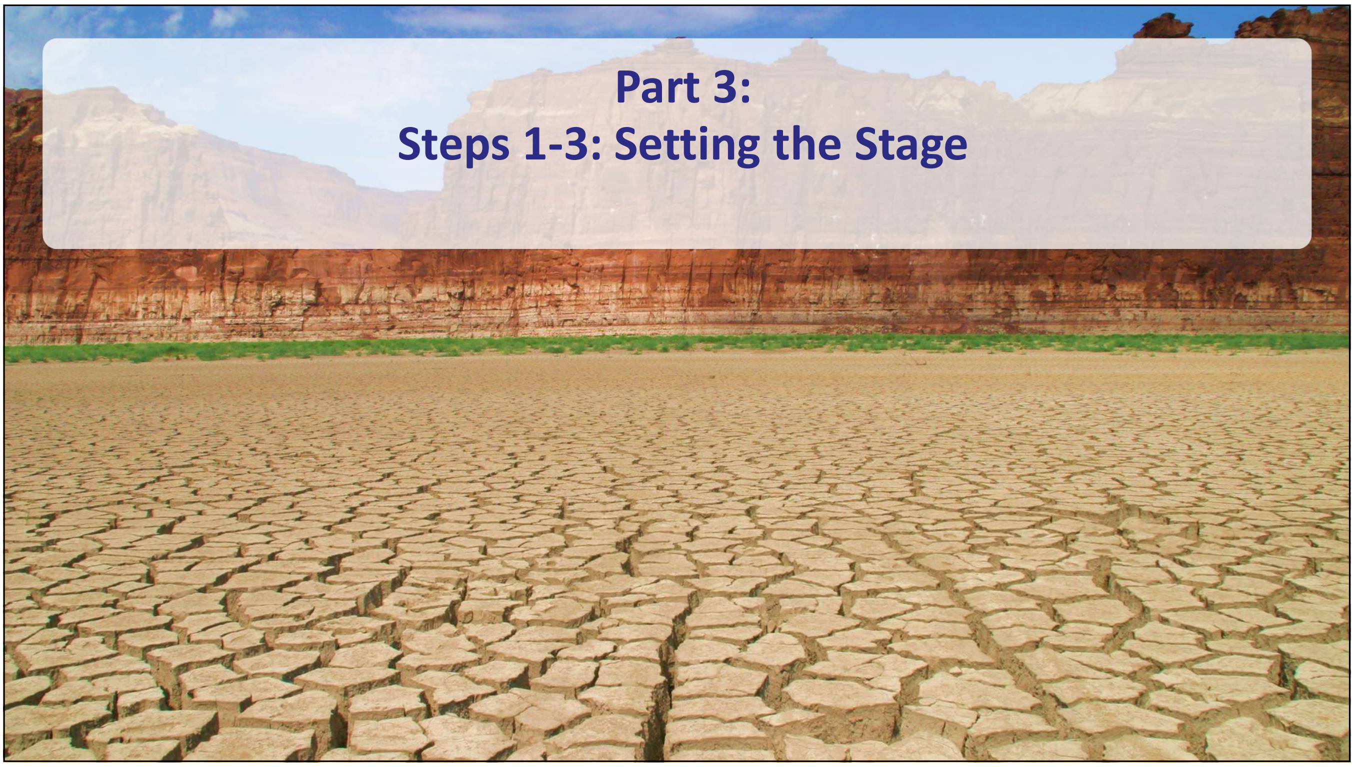
Connector rooms
Stairs
Hallway
Corridors

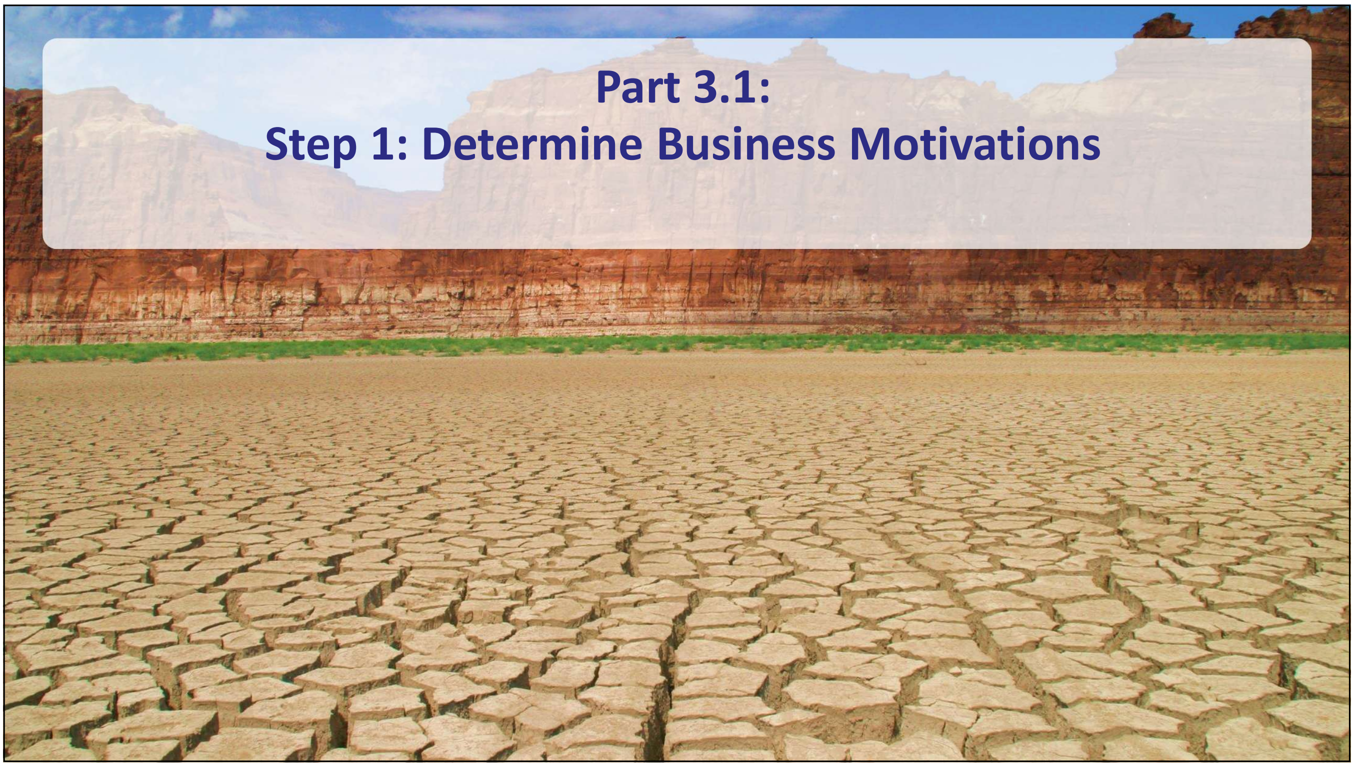
What Comes First?



Roadmap for Designing Data Architectures

1. Determine business motivations
2. Determine new requirements
3. Analyze the existing environment
4. Study new products and technologies
5. Define architectural design principles
6. Select a reference data architecture
7. Design the new data architecture
8. Determine the Implementation approach
9. Select new products and technologies
10. Introduce the data architecture within the organization


Part 3:
Steps 1-3: Setting the Stage


Part 3.1:
Step 1: Determine Business Motivations

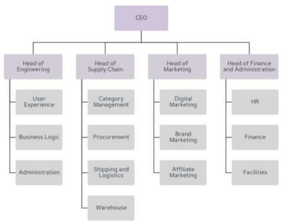
Poor Examples of Business Motivations

- Change insights and requirements
- Deployment of self-service BI
- Optimization of existing data architecture
- The platform on which the current BI system is hosted externally is old and needs to be replaced
- Move to the cloud
- Data science is not very well supported by current data warehouse environment
- We want to do more with the data we have, but it's hard to get to it

Proper Business Motivations

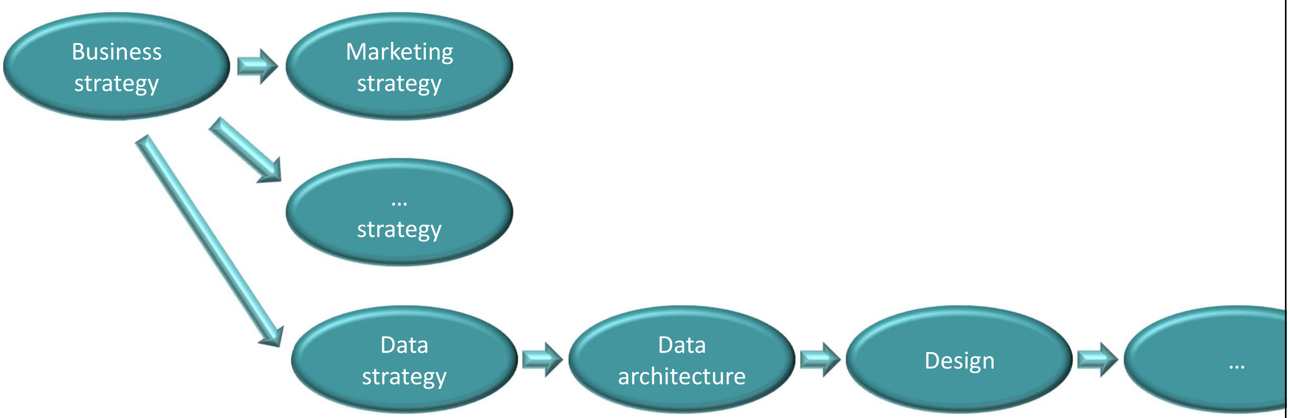
- Competitive improvement
  - Improving reaction speed to customer requests
- Support for customer journey and valuestream
- New business model
  - Allow customers real-time access to data
- Lower costs of specific business processes to improve margin
- Organization under threat
  - New competitor
- Comply with new laws and regulations
  - E.g. GDPR, CCPA, PSD2

Business Strategy and Data Strategy

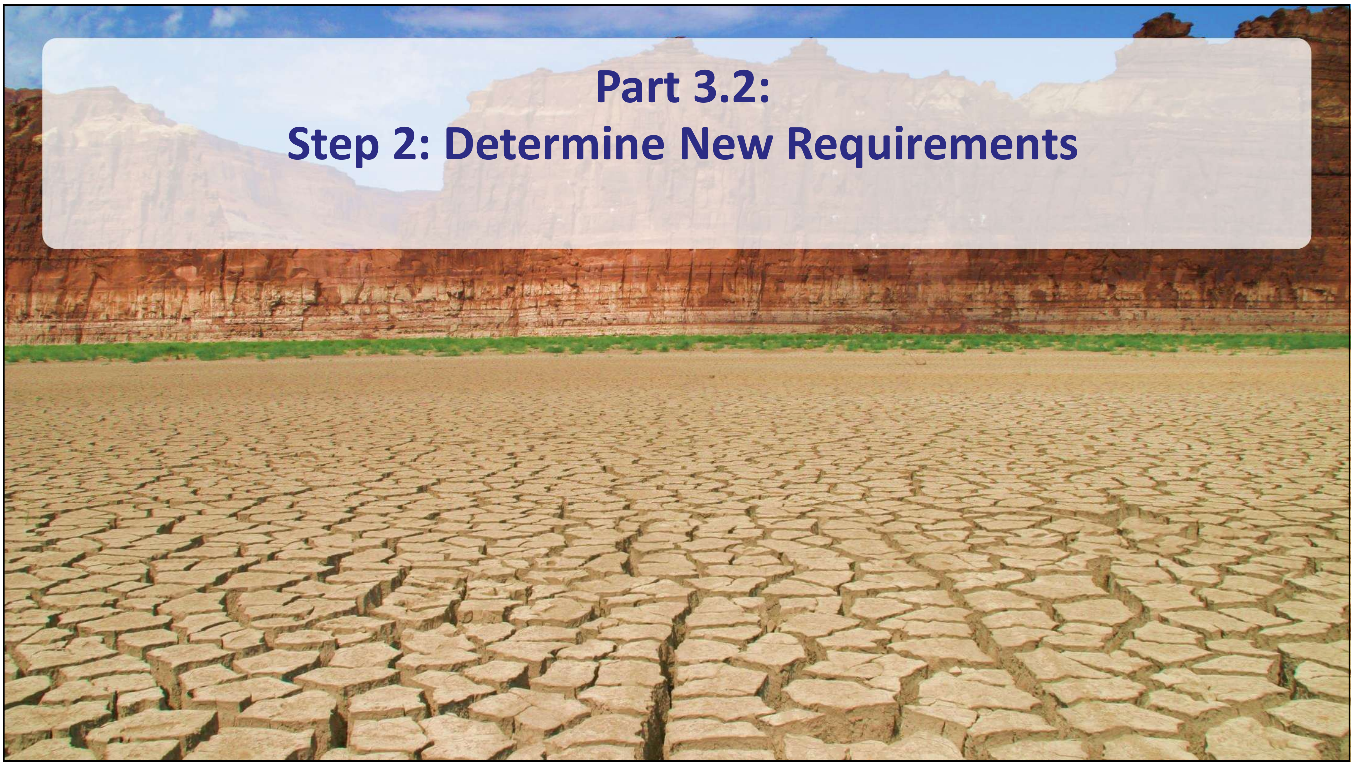


- Business Strategy
  - The challenges of top executives
    - New regulations, competitors, …
  - The main concerns for current business processes
  - Future business developments
    - New business domains
- Data Strategy
  - New data architecture has to “fit” the data strategy
  - New demands for data delivery

From Strategy to Data Architecture and Onwards



Part 3.2:
Step 2: Determine New Requirements



Determine New Requirements (1)


Photo: Markus Spiske

- New analytical functionality
- Lower latency for reports
- More users
- More access to metadata
- Migration to cloud platform
- More data
- More transparency of architecture
- Better security
- Deployment of data science
- …

Determine New Requirements (2)

Photo: Markus Spiske

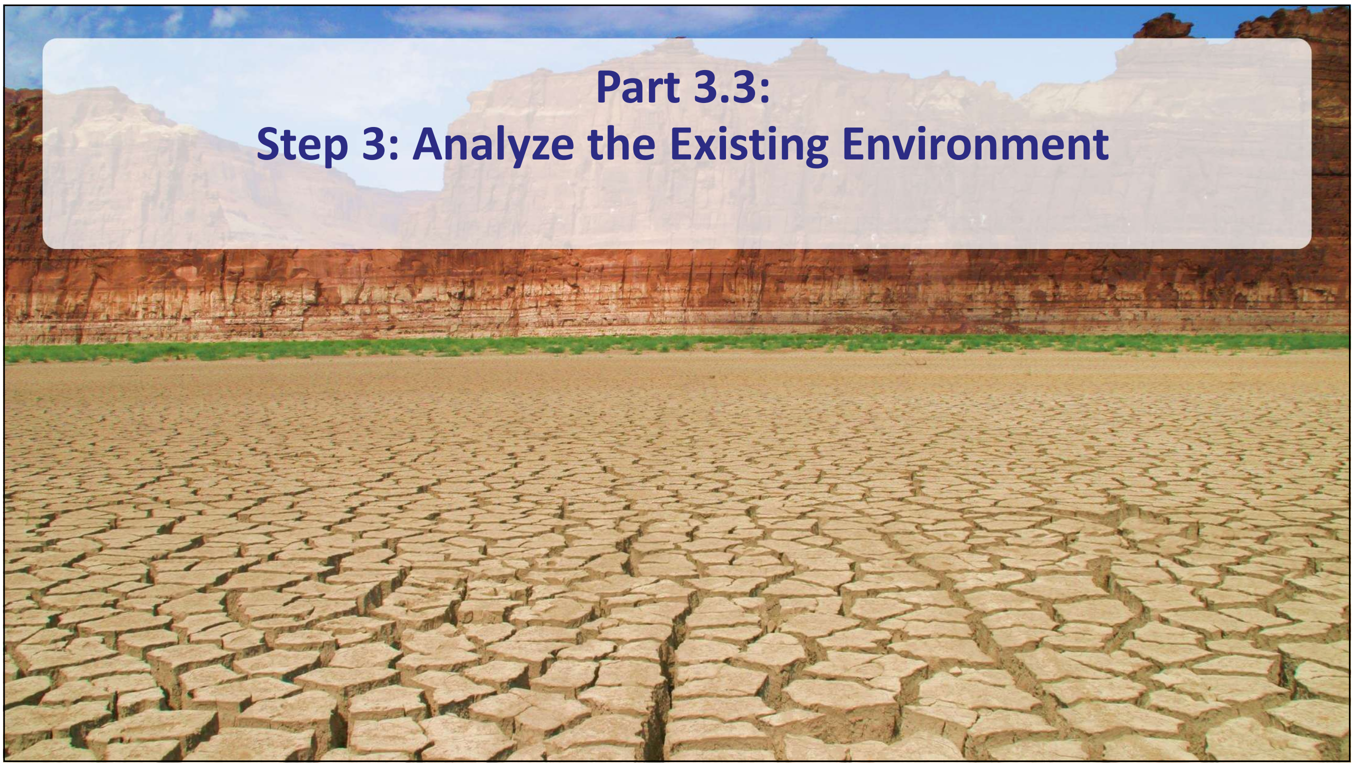
- Reconstruction of processes and decisions
- Data-shop to discover and describe data
- Migrate/redevelop source systems
- Minimize data copies
- Sharing data with other organizations
- Making data AI-ready
- …

Determine Constraints

Photo: Martin Sanchez

- Laws and regulations
  - GDPR, CCPA, PSD2, …
- Budget restrictions
- Software limitations
  - One-stop shopping, open source preferred, company preferences, …
- Hardware limitations
  - No easy processing, memory or storage scalability, …
- Current legacy systems
  - Mainframe-based, proprietary applications, plain old, out-of-date/obsolete development environments
- Internal ICT skills

Part 3.3:
Step 3: Analyze the Existing Environment



Determine Current ICT Bottlenecks

Photo: Iwona Castiello dAntonio

- Performance
- Report latency
- Productivity - backlog
- Functionality
- Costs too high
- Business – ICT cooperation
- Non-professional IT organization
- Not IT savvy
- …

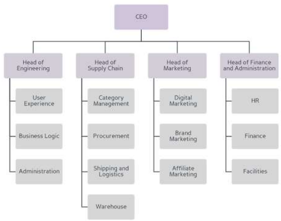
Analyze Existing Applications

- Data producers
  - Can we access the database directly or through an API?
  - Current workload?
  - Homemade or application?
- Data transformers and transporters
  - Home-made or professional (e.g. ETL, bus, data virtualization)?
  - Implementation style?
- Data Consumers
  - Homemade?
  - Internal or external?

Analyze Technology and Products in Use

- Selected products and versions
- Selected (cloud) platforms
- License costs
- Infrastructure
- Potential migration challenges

Determine the Culture of the IT Organization



- Traditional?
- Risk evasive?
- No experience with modern technologies?
- Cynical towards new developments?

Determine IT Maturity Level of Organization (1)

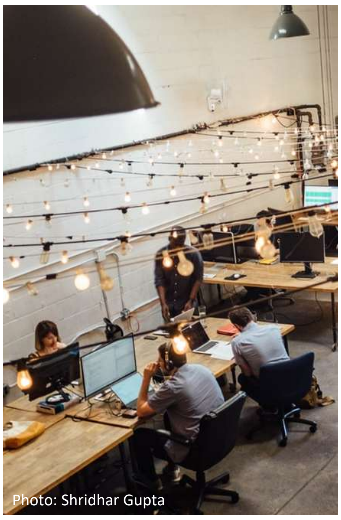
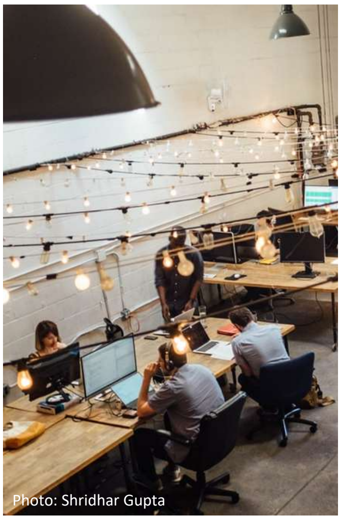


Photo: Shridhar Gupta

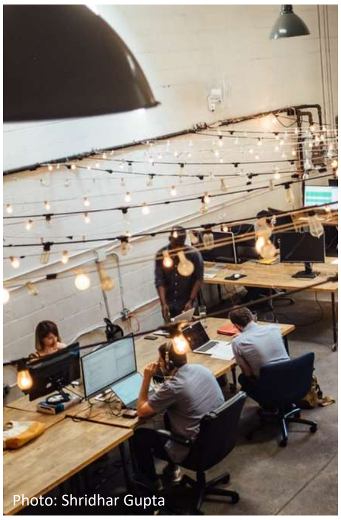
- Data processing checks
  - Is data primarily stored to support business processes and to conform to reporting regulations?
  - Can DBAs see the data?
  - Are ETL processes started manually?
  - Is ETL crash automatically fixed?
  - Are data processing specifications scattered across all modules?
  - Is metadata available and kept up to date?
  - Are “old” reports reproducible?

Determine IT Maturity Level of Organization (2)



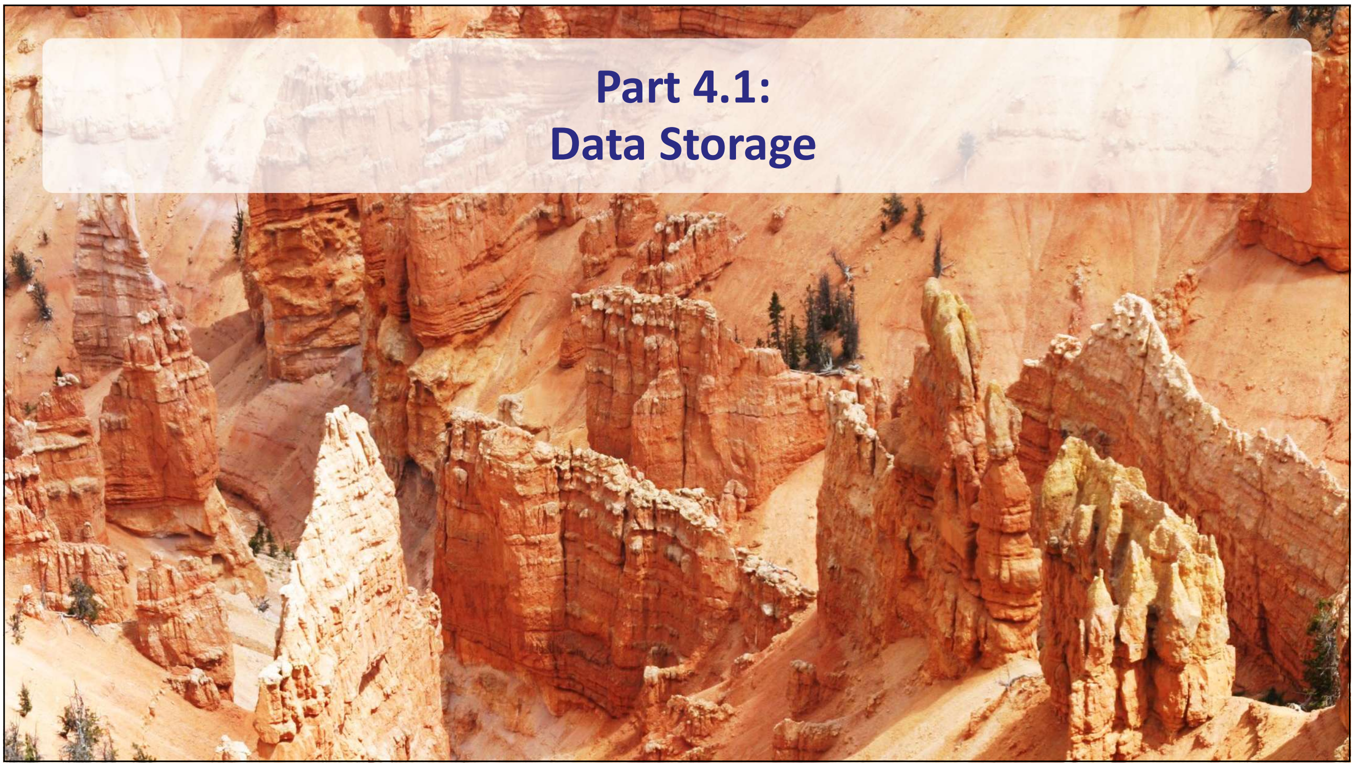
- Data consumption
  - Do reports primarily show what has happened within business processes?
  - High data latency?
  - Do they use predictive analytics to optimize business processes and decision-making processes?
- Data management
  - Ownership of data assigned?
  - Is there focus on data quality?
  - Are there procedures in place to fix incorrect data?

Determine IT Maturity Level of Organization (3)



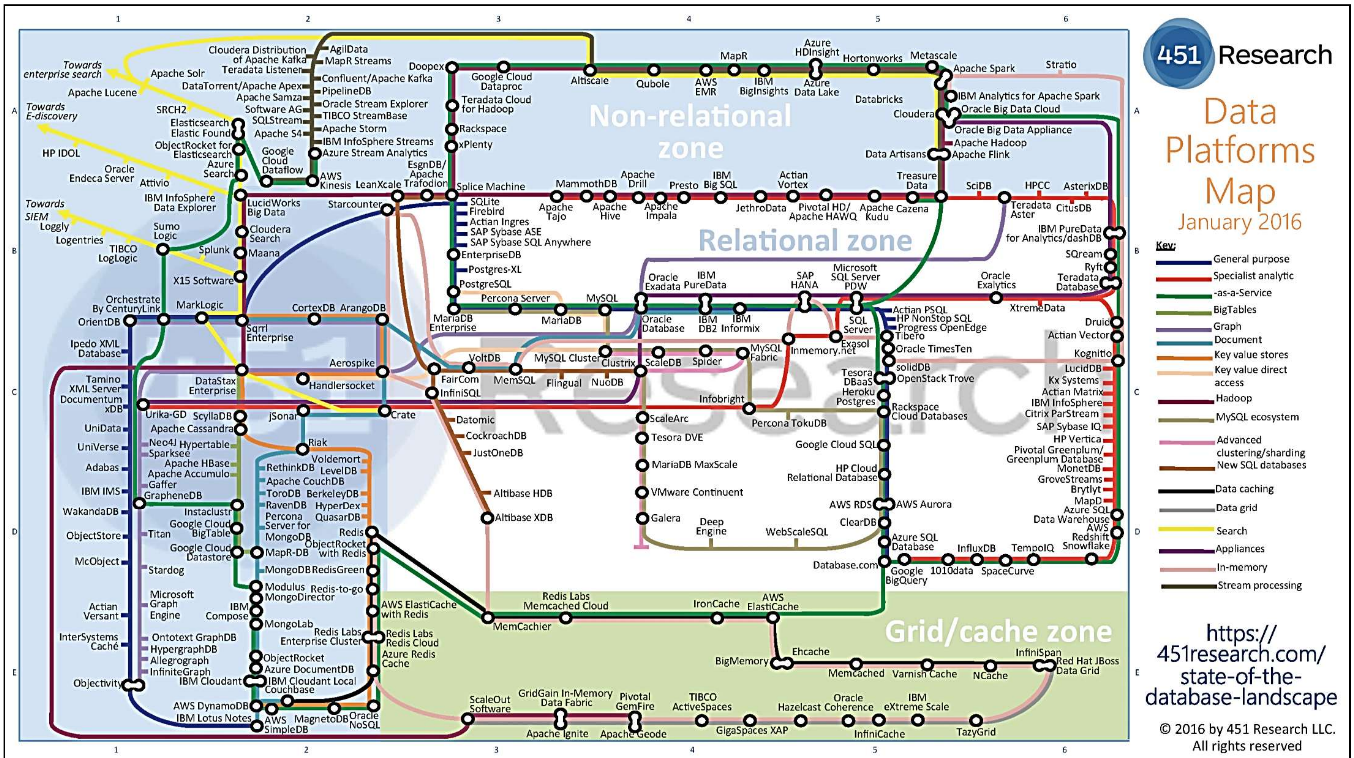
- ICT skills
  - All development outsourced?
  - Many tool-jockeys?
  - Performance anxiety?
  - Minimal knowledge of new technologies?


Part 4:
Step 4: Analyze New Technologies for
Data Storage, Processing, and Analytics

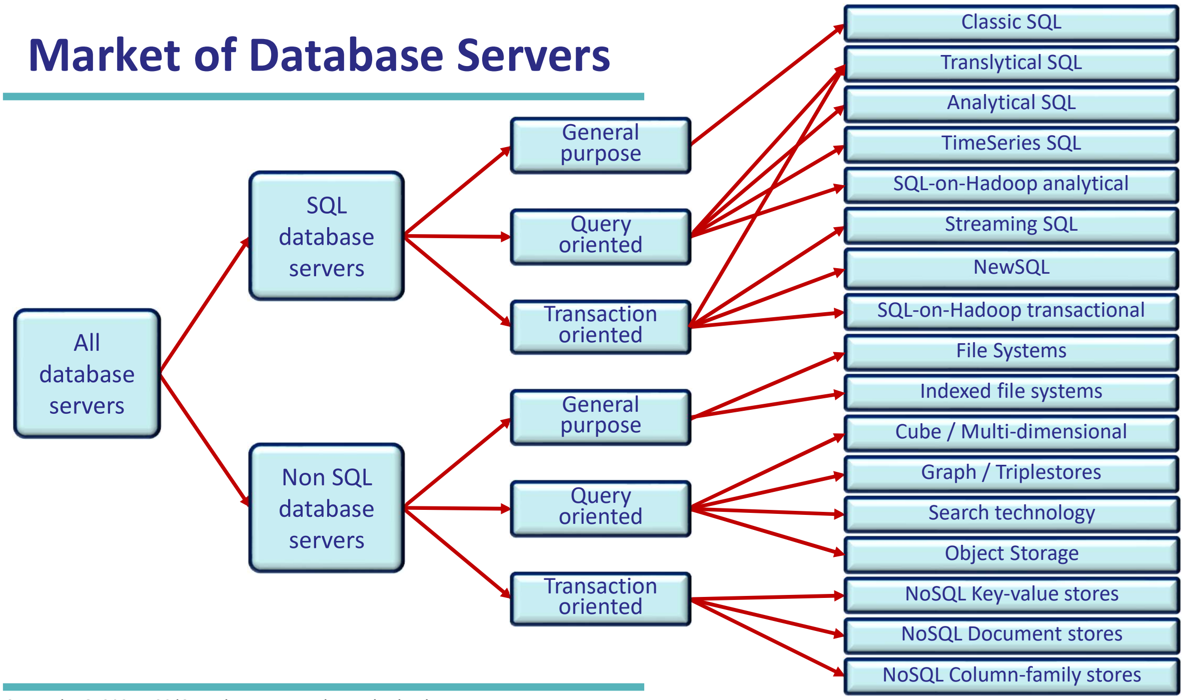

Part 4.1:
Data Storage

Trends in Database Market

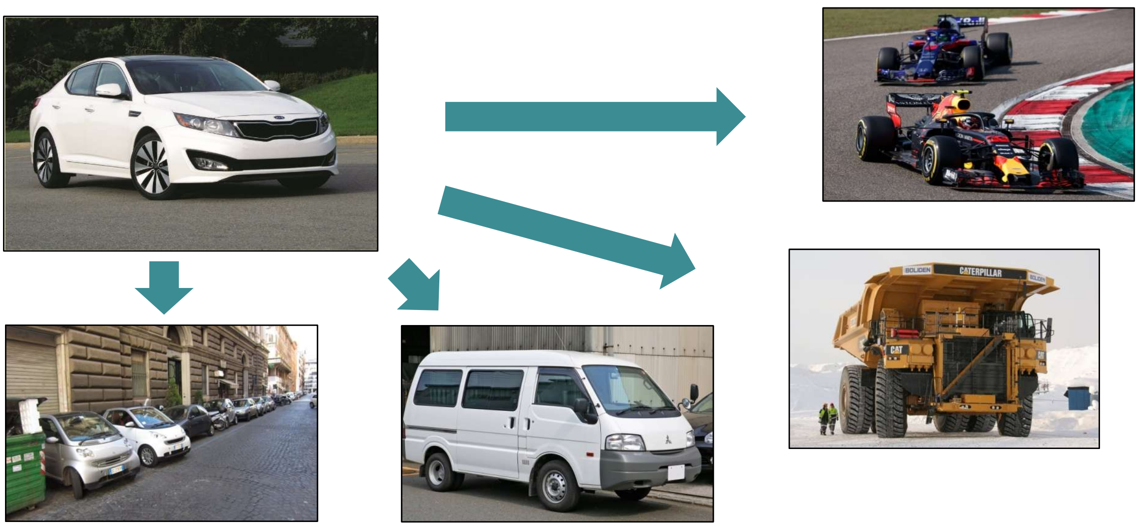
- More data, more queries, more transactions, more concurrent users, more complex queries, …
- Coming and going of products
- Less standards
  - Less portability
- No interchangeable products
- Specialization of products
  - Limited use cases



Market of Database Servers



Specialization of Cars



Database Technology has Changed

Generic Database Solutions

↓

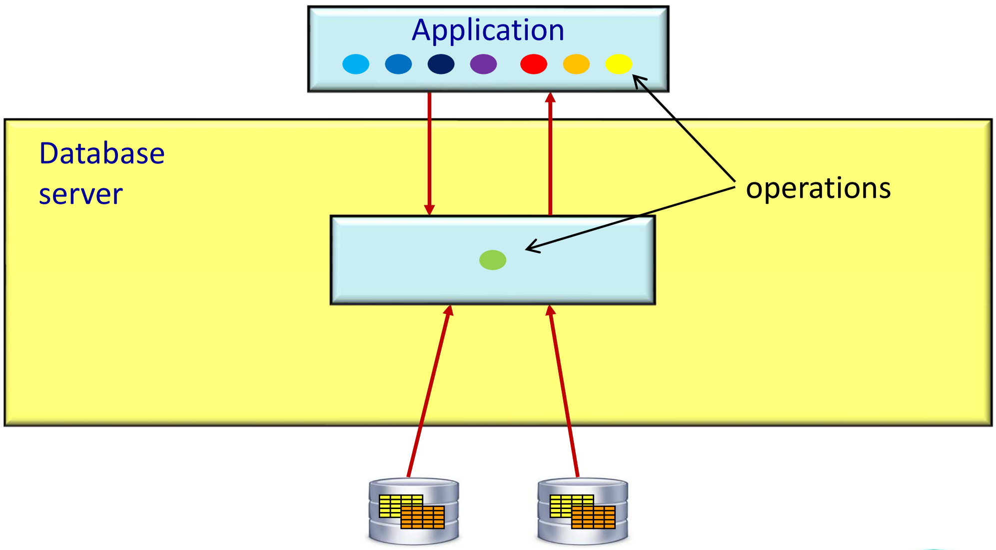
Specialized Database Solutions

Use Case Agnostic

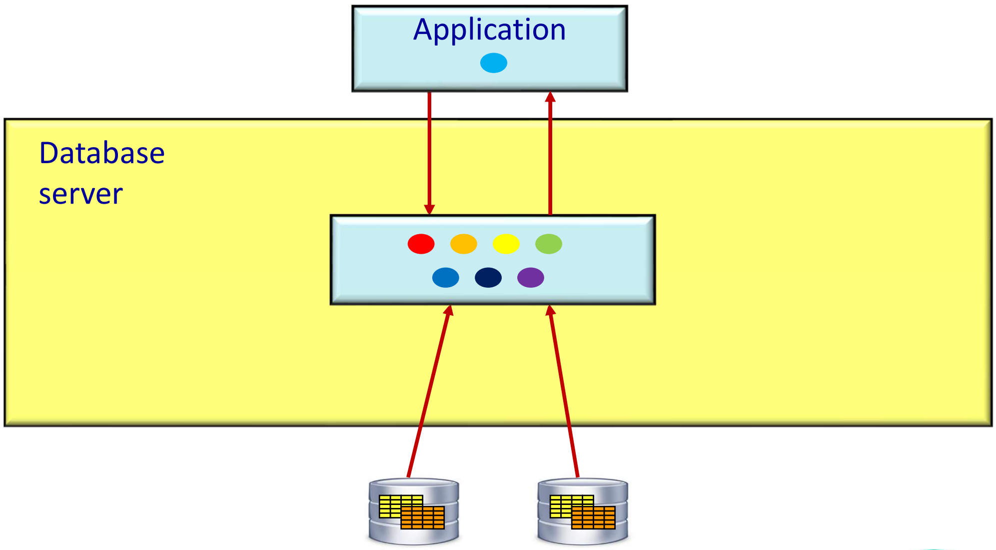
↓

Use Case Specific

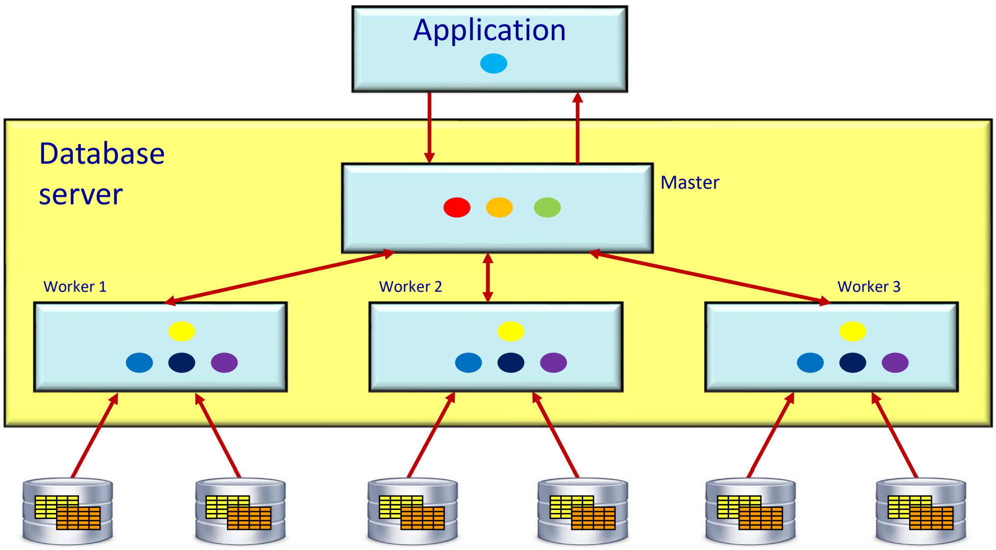
Application-based Analytics



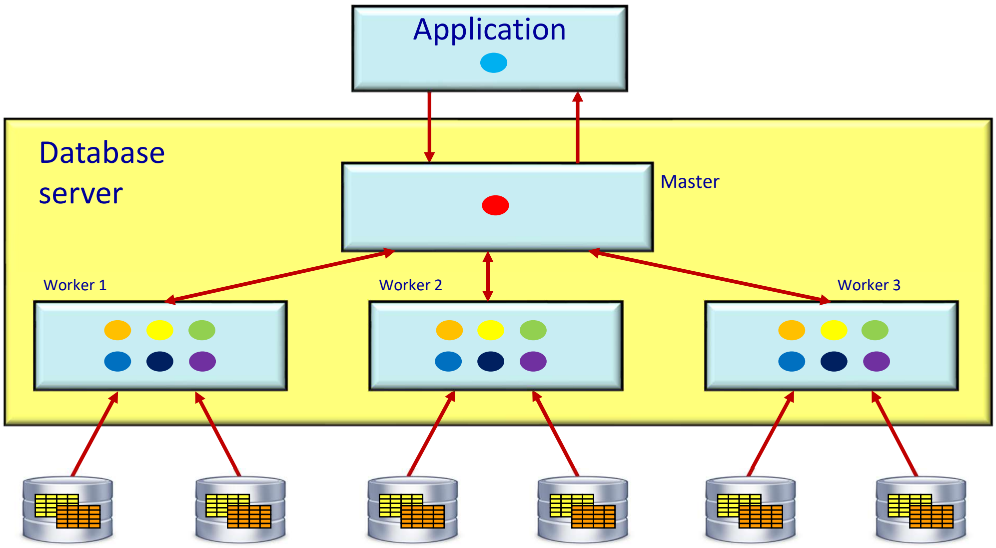
In-Database Analytics



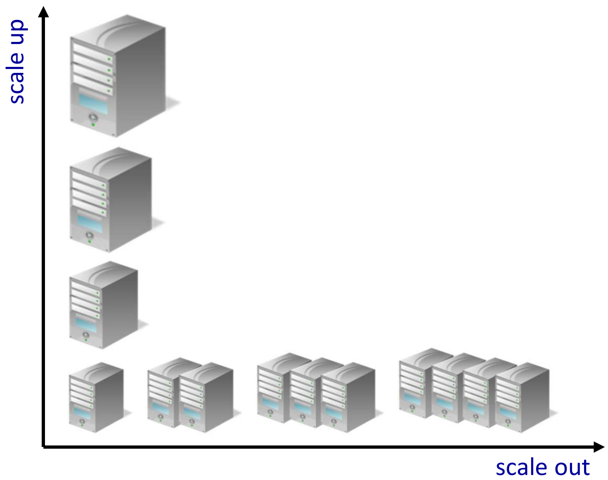
Partial Parallel Analytics



Full Parallel Analytics

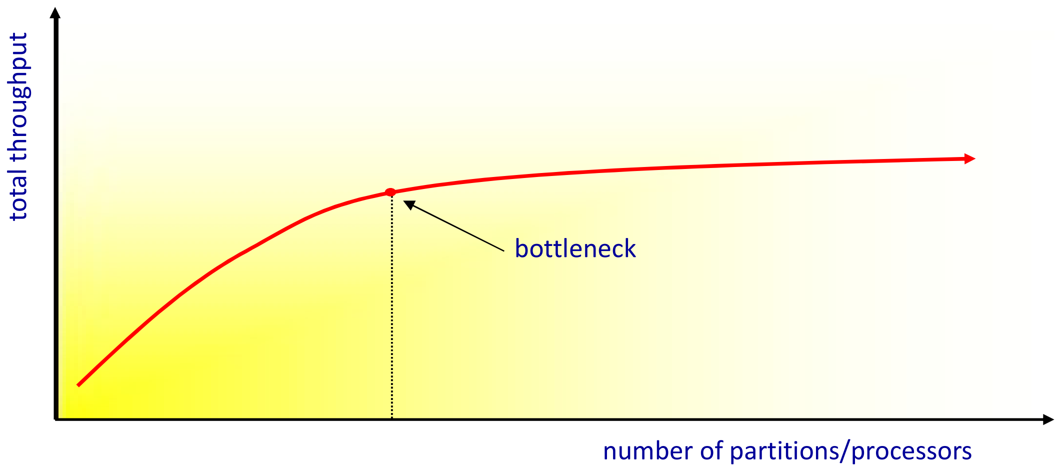


Scale Up versus Scale Out



- Scale up (vertical scaling) means adding more resources to one node in a system
- Scale out (horizontal scaling) means adding more nodes to a system
  - Continuous availability/redundancy
  - Cost/performance flexibility
  - Contiguous upgrades
  - Geographical distribution

Effect of Partitions on Query Response



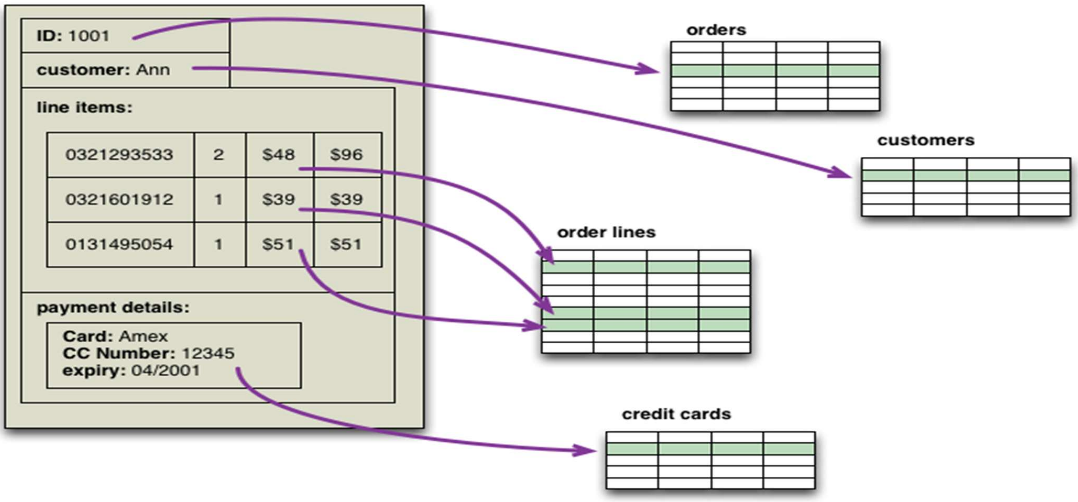
NoSQL Database Servers

Apache HBase
mongoDB
redis
Cassandra
Google BigTable
Azure Cosmos DB
CouchDB relax
riak
LevelDB
RAVENDB
UpscaleDB
Couchbase
HYPERTABLE
memCached

Tricks to Improve Performance

- Aggregate data model
  - To remove the impedance mismatch
- Design architecture to scale-out
  - Sharding
- Reduce functionality (security, query power, data integrity, …)
- Lower consistency
- Give developers full control over internal processing
- "Push down" complex operations

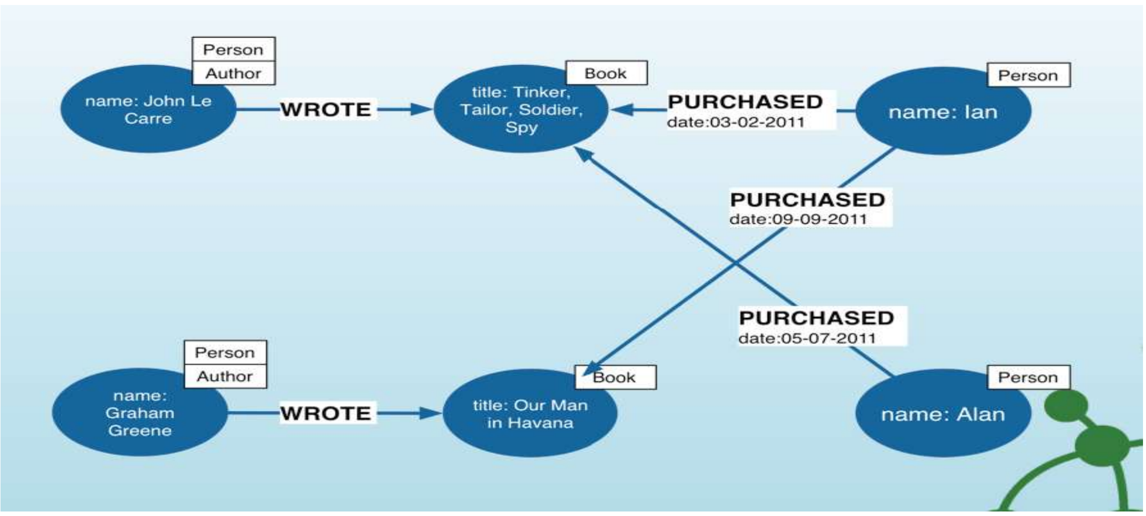
NoSQL: Aggregate Data Model



Typical NoSQL Use Cases

- Transactional
- Big transactional workload
- Single record/document transactions
- Massive data ingestion
- Simple reporting – point queries
- Dynamic data structures
- Complex data structures
- "Narrow" data model

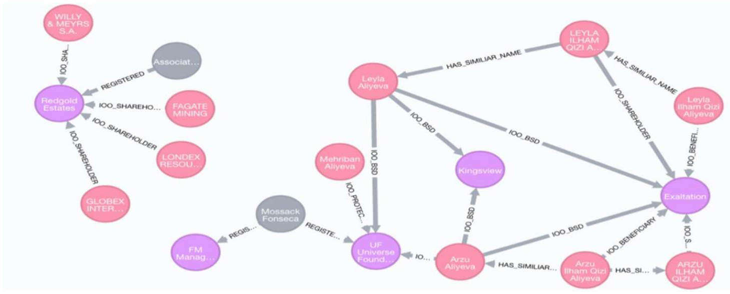
Graph Database Servers



Source: M. Hunger, Neo Technology, Data Modeling with Neo4j, Aug 2013

Application Areas Graph Databases

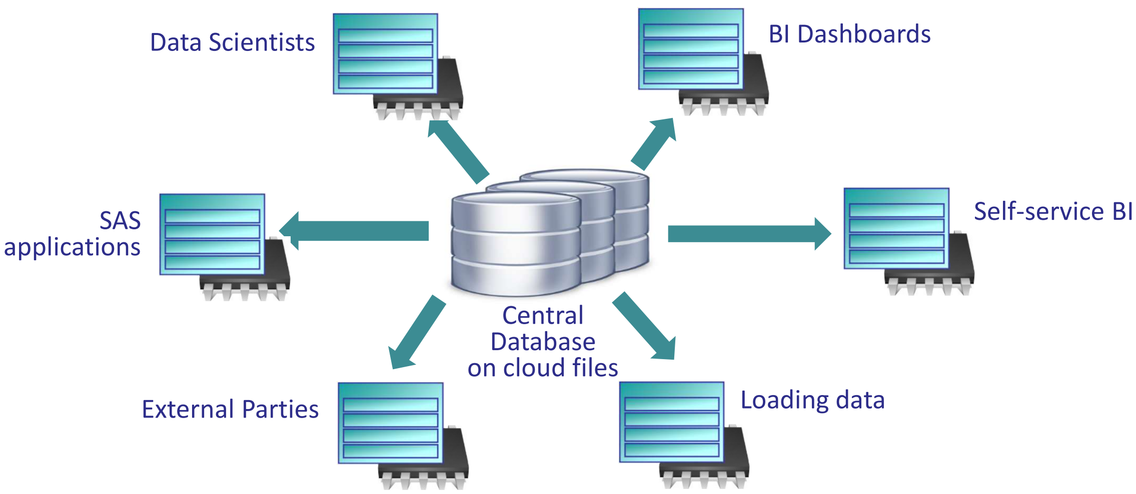
- Social network analysis
- Network impact analysis
- Optimal route determination
- Internet retail recommendations
- Logistics
- Fraud analysis
- Securities and debts
- “Panama papers”
- And many more



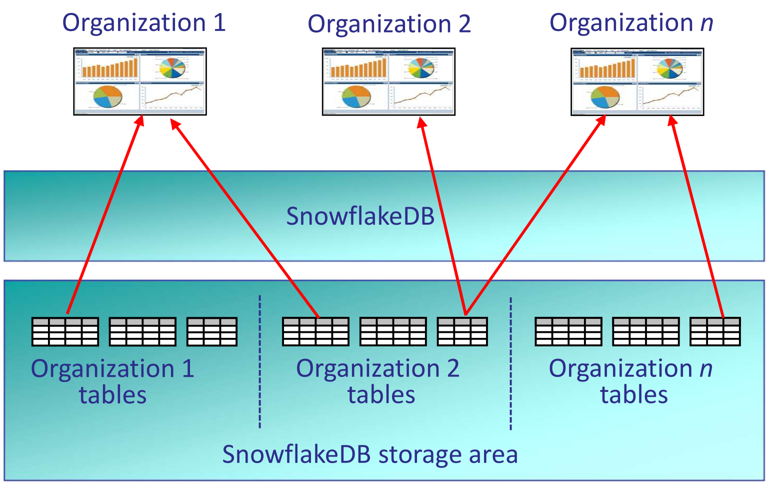
Analytical SQL Database Servers

Examples	
1010Data	Microsoft Azure Synapse
Amazon Redshift and Athena	OmniSci (MapD)
Apache HAWQ	Oracle Database In-Memory
BlazingSQL	SAP HANA
CitusDB (PostgreSQL)	SAP Sybase IQ
ClickHouse SQL	SnowflakeDB
Databricks Delta Lake	Splice Machine
Edge Intelligence	SQream
Exasol	Starburst (Trino formerly Presto)
Google BigQuery	Teradata Vantage
Greenplum	XTremeData dbX
IBM DB2 Warehouse on Cloud	Several SQL-on-Hadoop engines
Ignite InfoBright DB	Vertica
Kinetica	And many others …
Kognitio WX2	

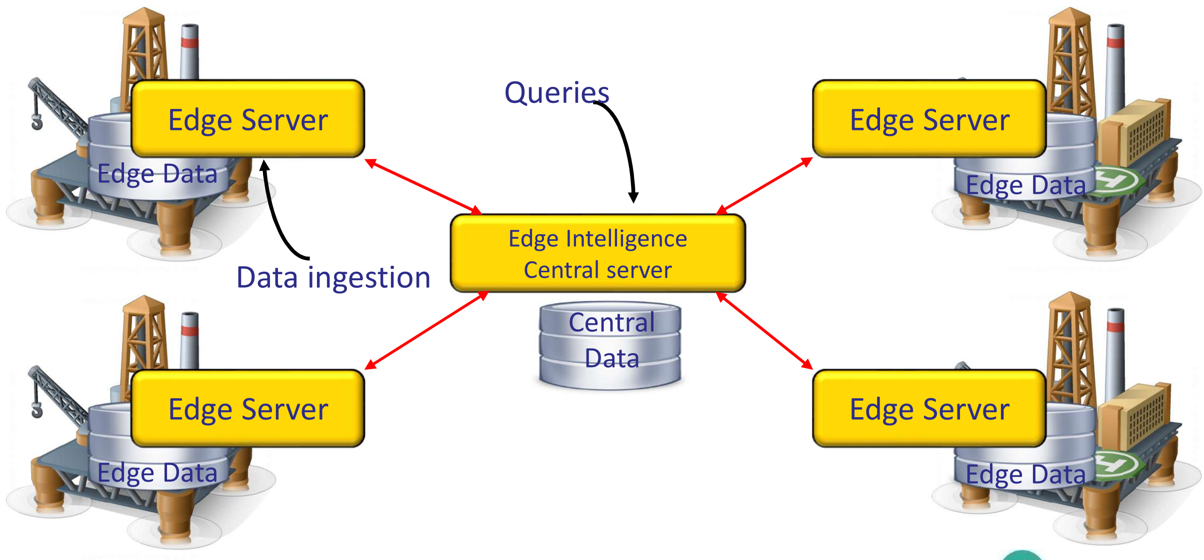
Example 1: Snowflake



Example 1: Snowflake



Example 2: Edge Intelligence



Example 2: Edge Intelligence

Non-replicated Data at Edge 1

Store_id	Customer	Datetime	...
1	Metheny	2017-11-01 12:00:08	...
1	Johnson	2017-11-01 12:10:18	...
1	Young	2017-11-01 12:12:33	...
1	Morrison	2017-11-01 12:50:09	...
1	Harris	2017-11-01 12:55:45	...

Replicated Data at Edge 1

Customer	...
Metheny	...
Johnson	...
Young	...
Morrison	...
Harris	...
Mitchell	...
Stills	...
Dylan	...
Brown	...

Non-replicated Data at Edge 2

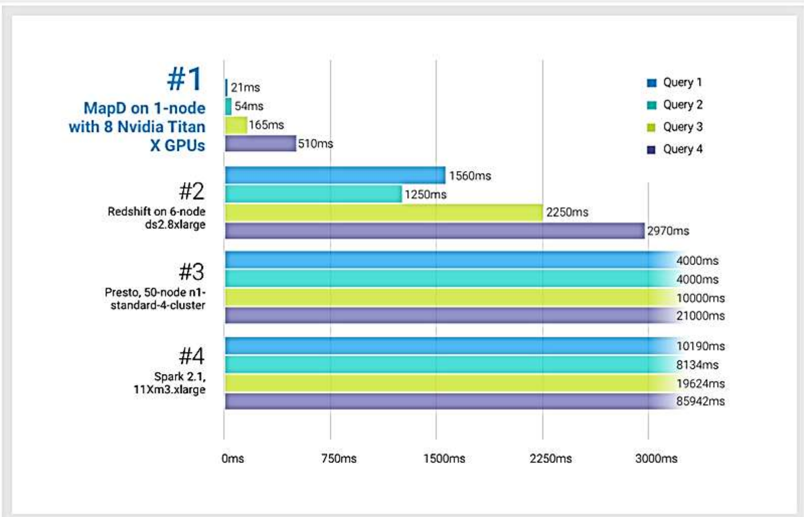
Store_id	Customer	Datetime	...
2	Metheny	2017-11-01 12:01:32	...
2	Mitchell	2017-11-01 12:05:42	...
2	Stills	2017-11-01 12:11:39	...
2	Dylan	2017-11-01 12:12:30	...
2	Brown	2017-11-01 12:40:19	...

Replicated Data at Edge 2

Customer	...
Metheny	...
Johnson	...
Young	...
Morrison	...
Harris	...
Mitchell	...
Stills	...
Dylan	...
Brown	...

NVIDIA TITAN V: GPU With More Than 5,000 Cores

Comparison of Analytical SQL Database Servers

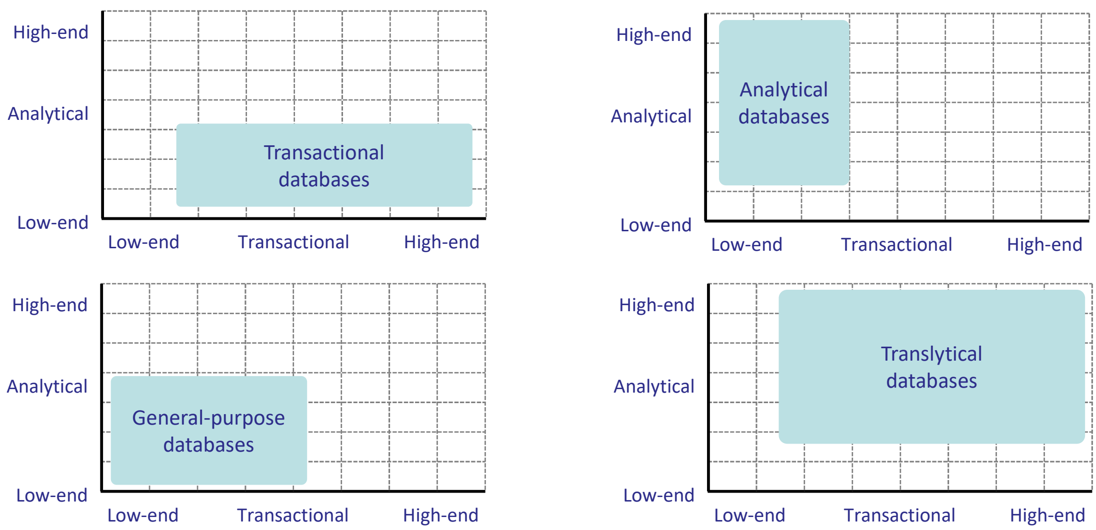


- Products: BlazingSQL, Kinetica, HeavyDB (OmniSciDB, MapD), SQream
- They make use of the parallel power of GPU's
- Long-term data persistency is not their core business

Transactional SQL Database Servers

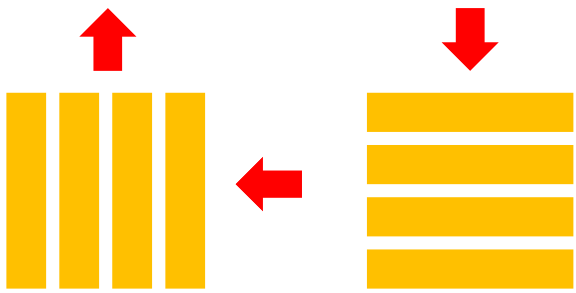
- Examples: Clustrix, DataBricks Delta lake, SingleStore (MemSQL), Splice Machine, Pivotal GemFire XD (SQLFire), VoltDB, and YugabyteDB
- NewSQL is not a new SQL dialect
  - The internal architectures are different from classic SQL database servers
- High scalability with respect to transactions
- Full-blown SQL - high level of data independence
- ACID-compliant = 100% consistency
- Exploitation of low-cost clusters

Four Categories of SQL Databases

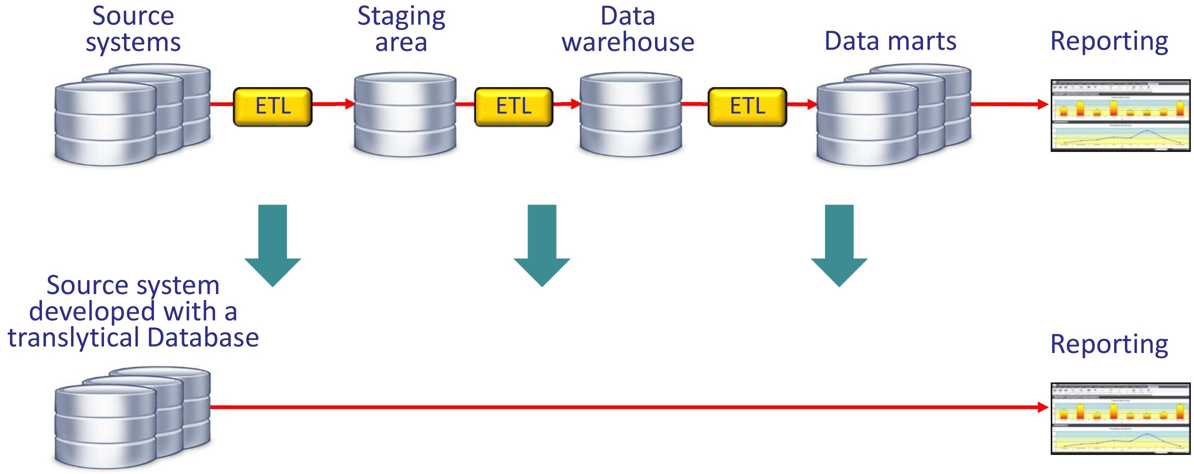


Example: SingleStore (translytical)

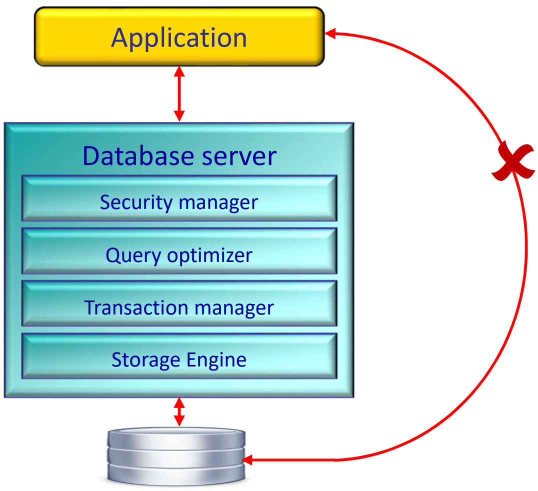
ID	Name	Initials	Date Entered	City	State
12345	Young	N	Aug 4, 2008	San Francisco	CA
23324	Stills	S	Sep 10, 2009	New Orleans	LA
57657	Furay	R	Oct 16, 2010	Yellow Springs	OH
65461	Palmer	B	Nov 22, 2011	Boston	MA
…	…	…	…	…	…



Zero Data-Latency Architectures

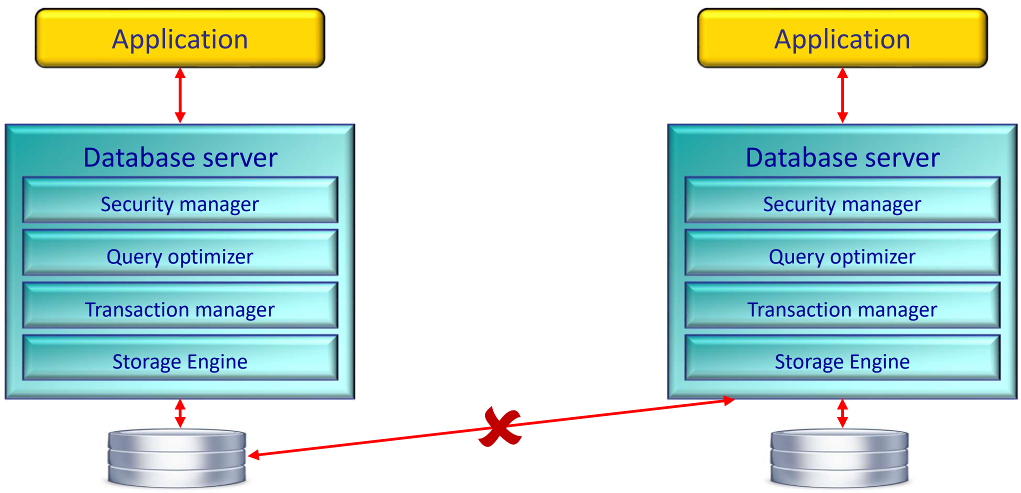


Most Database Servers Use Proprietary Files

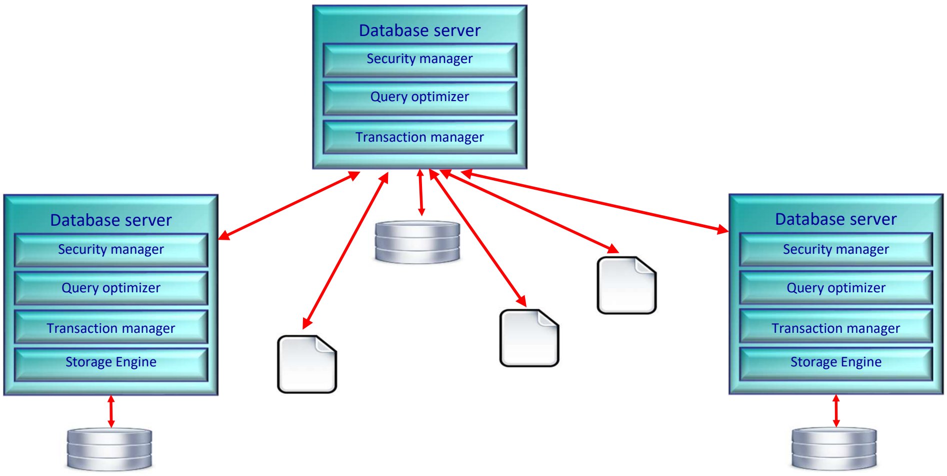


- In many database servers a proprietary storage format is used
- Data can only be accessed via database server
- Data needs to be copied for other database servers

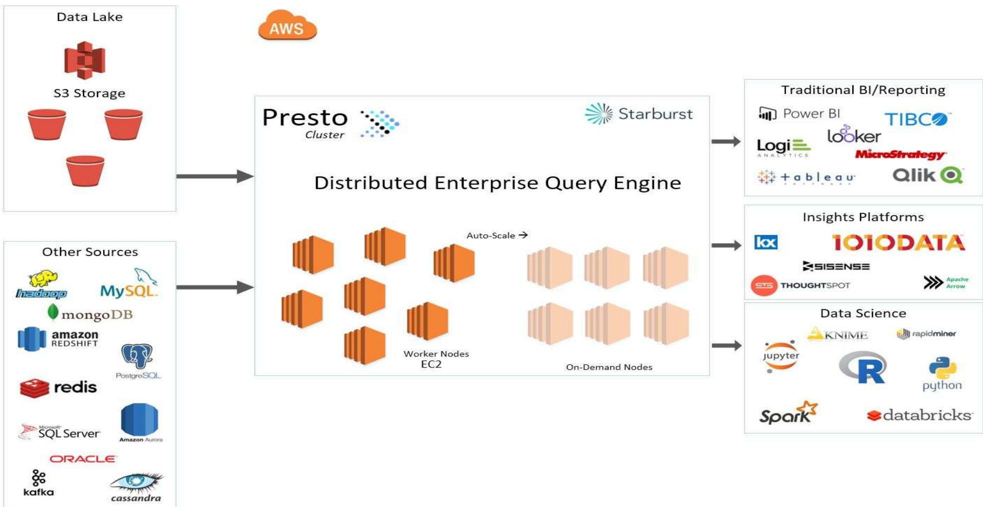
Database Servers Can't Share Data



Accessing "External" Data



Example: Starburst (based on Trino)

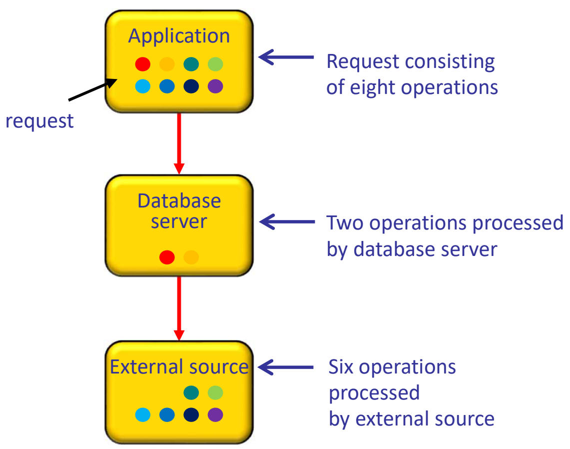


Example: Amazon Athena

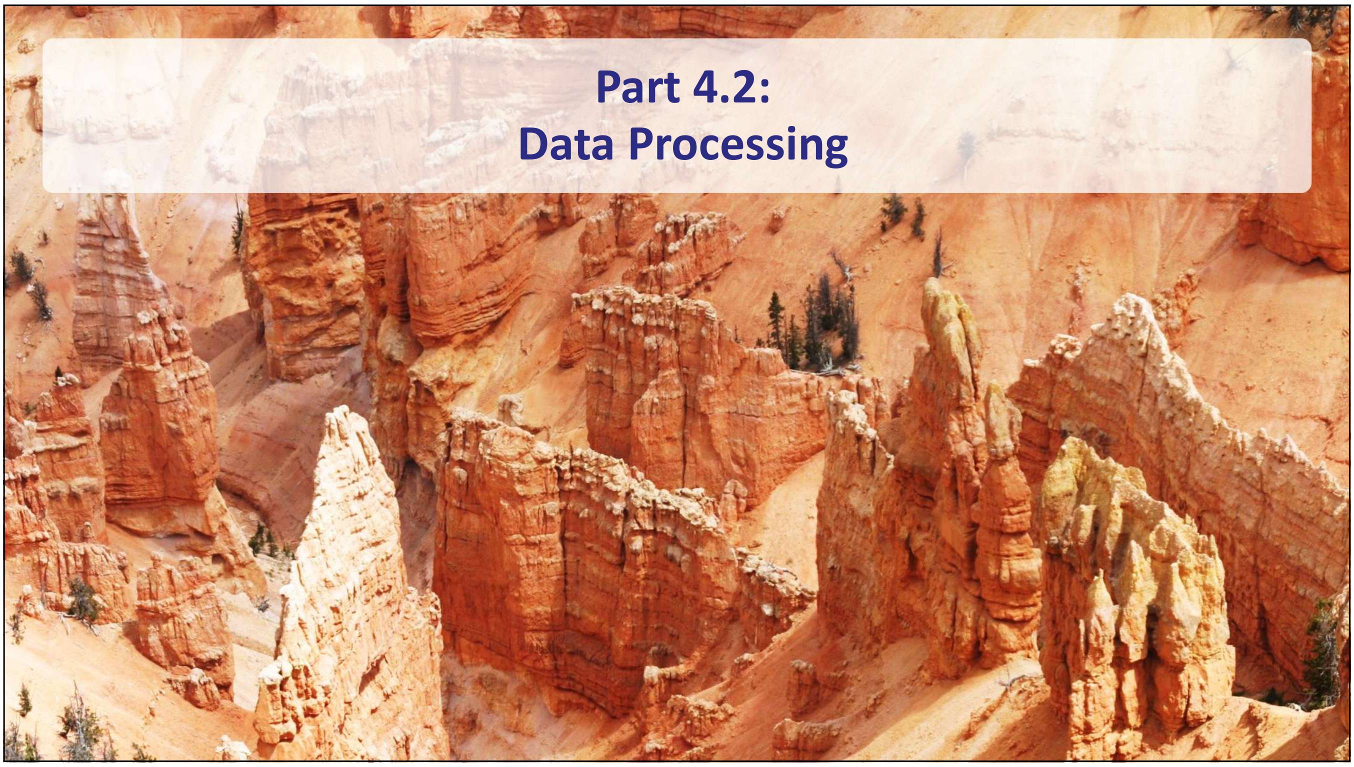
```
CREATE EXTERNAL TABLE employee
(
  ID string,
  NAME string,
  AGE string,
  GEN string,
  CREATE_DATE bigint,
  PROCESS_NAME string,
  UPDATE_DATE bigint
)
STORED AS AVRO
LOCATION 's3://my-bucket/staging/employees'
TBLPROPERTIES (
'avro.schema.literal'='
{
    "type" : "record",
    "name" : "AutoGeneratedSchema",
    "doc" : "Sqoop import of QueryResult",
    "fields" : [ {
      "name" : "ID",
      "type" : [ "null", "string" ],
      "default" : null,
      "columnName" : "ID",
```

```
}, {
  "name" : "NAME",
  "type" : [ "null", "string" ],
  "default" : null,
  "columnName" : "NAME",
  "sqlType" : "12"
}, {
  "name" : "AGE",
  "type" : [ "null", "string" ],
  "default" : null,
  "columnName" : "AGE",
  "sqlType" : "2"
}, {
  "name" : "GEN",
  "type" : [ "null", "string" ],
  "default" : null,
  "columnName" : "GEN",
  "sqlType" : "12"
}, {
  "name" : "CREATE_DATE",
  "type" : [ "null", "long" ],
  "default" : null,
  "columnName" : "CREATE_DATE",
```

Accessing External Data and Query Pushdown

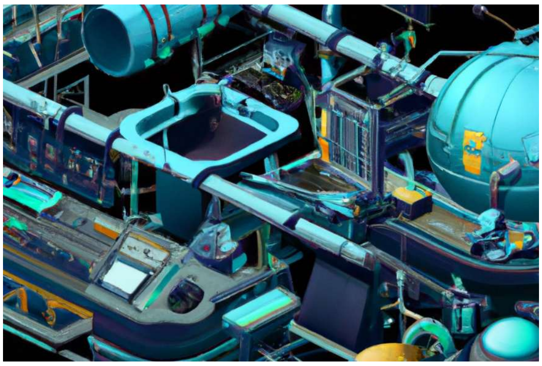


- Push processing to the external source
- Minimize network traffic
- Exploit the full power of the external source
- Optimize distributed joins
- Deal with datatype differences
- From structure-less data to structure-rich data



Part 4.2: Data Processing

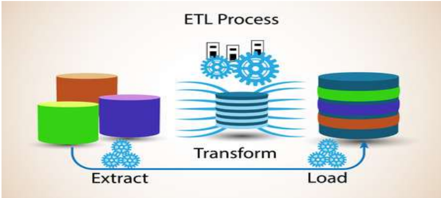
Categories for Data Processing



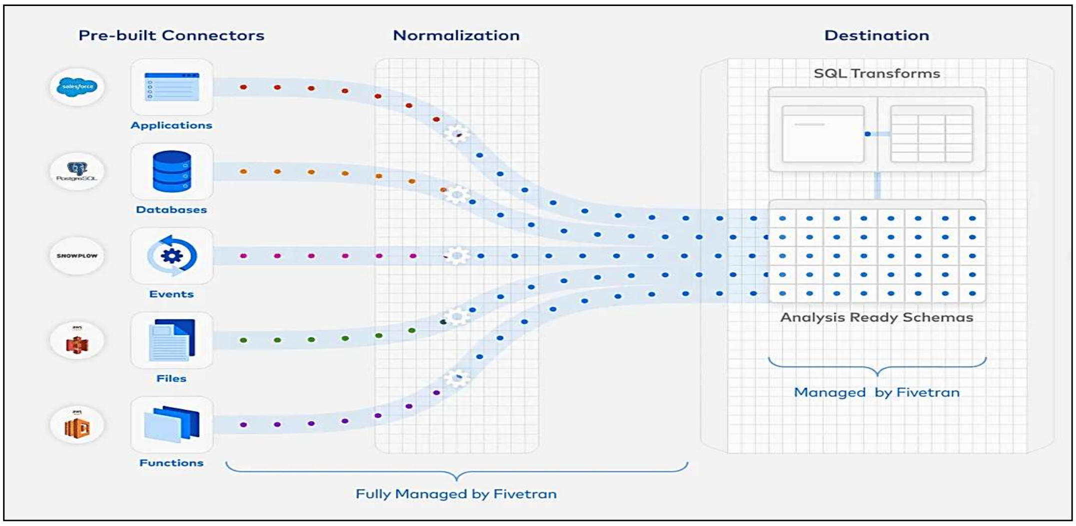
- ETL (Extract Transform Load)
- Data Replication (Change Data Capture)
- ESB (Enterprise Service Bus)
- Data Virtualization

ETL = Extract Transform Load

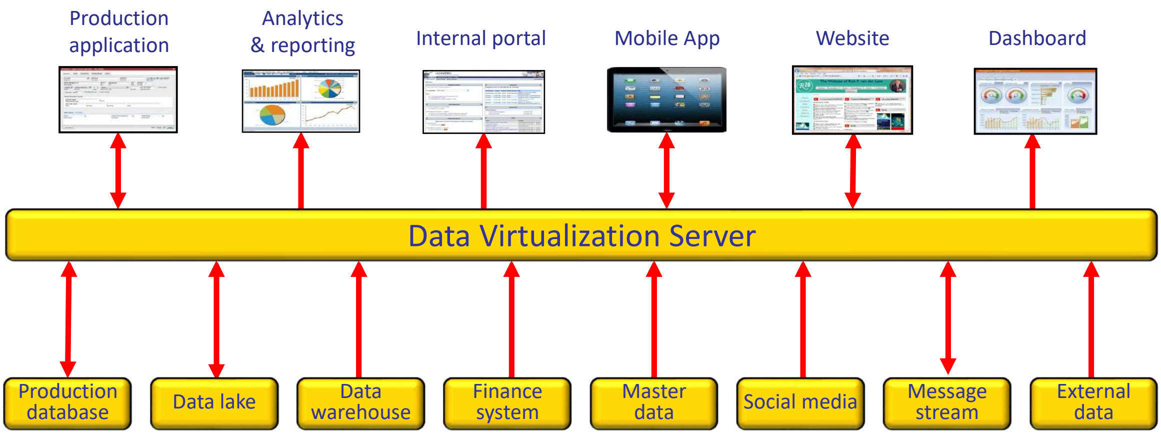
- Transforming of data structures
  - To a data structure suitable for reporting and analysis
- Cleansing of data
- Integration of data from production systems
- Transforming data
  - Filtering, aggregating, projecting, joining, splitting, …
- Scheduling the ETL process
  - Batch-oriented
- Managing the ETL process



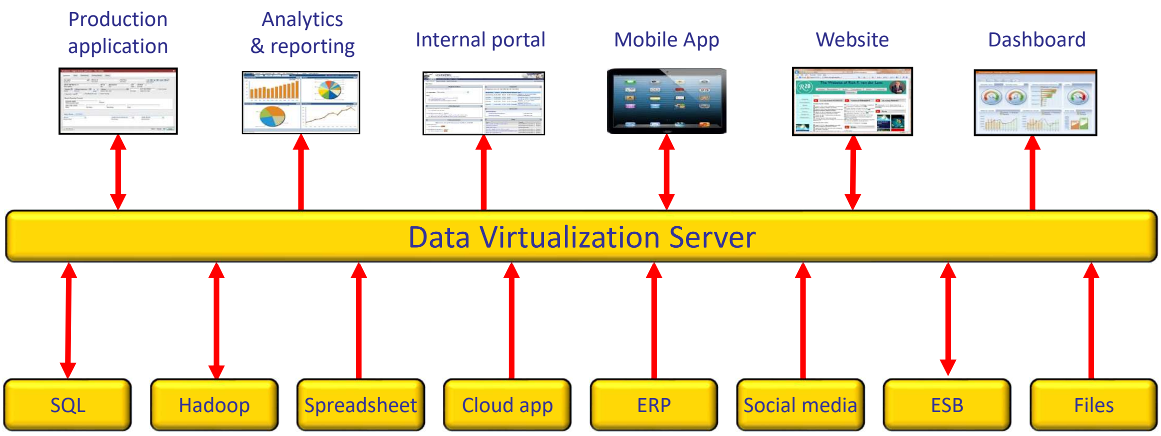
Example: Fivetran



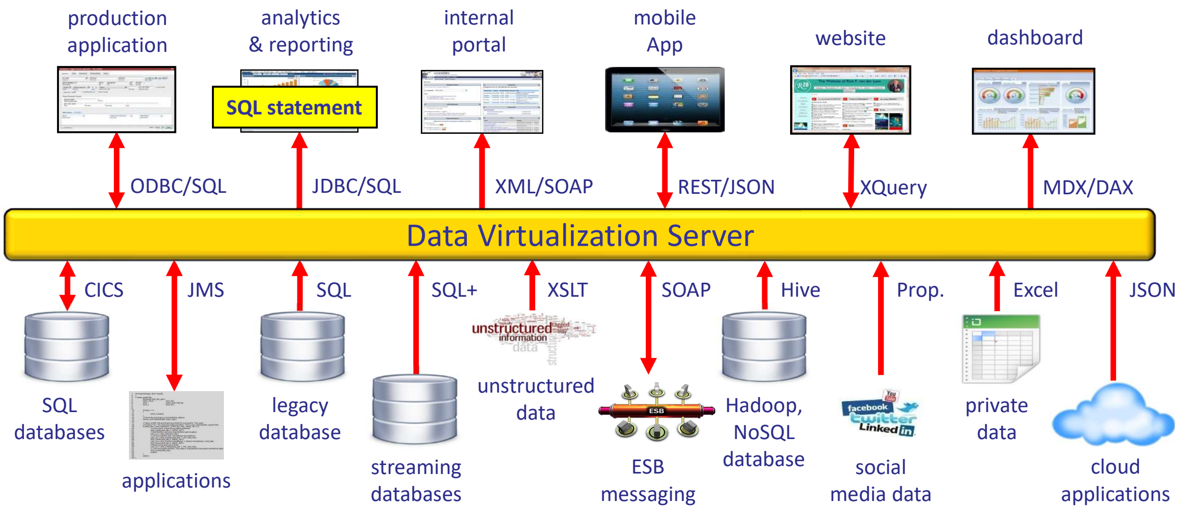
Data Virtualization Overview (1)



Data Virtualization Overview (2)



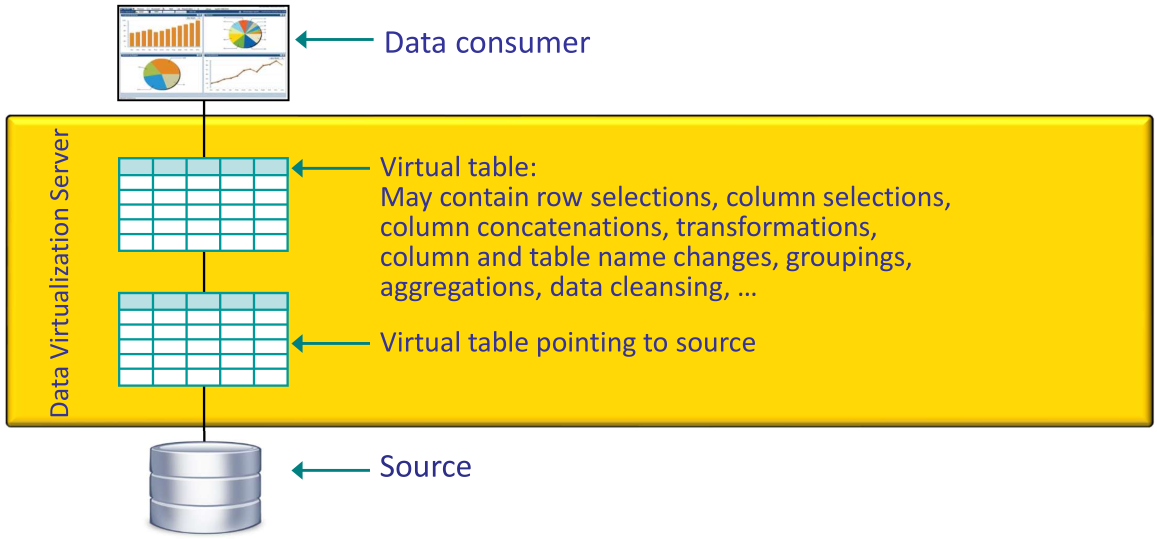
Data Virtualization Overview (3)



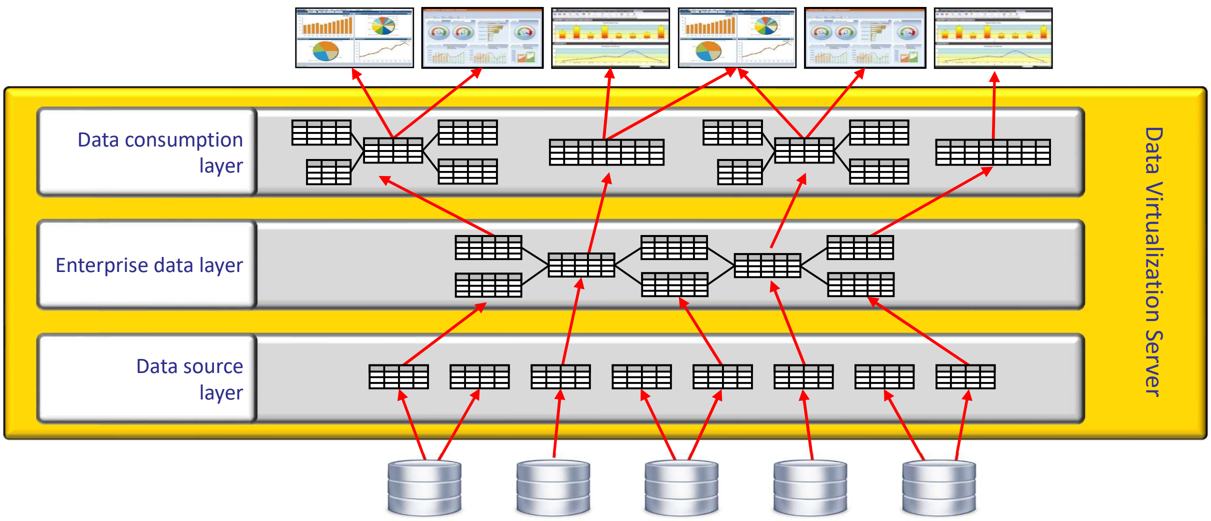
The Market of Data Virtualization Servers

- AtScale
- DataVirtuality (Pipes, Pipes Prof, LDW)
- Denodo Platform
- Dremio
- Fraxses
- IBM InfoSphere Federation Server & IBM Data Virtualization Manager for z/OS (formerly Rocket Data Virtualization)
- Red Hat JBoss Data Virtualization (Teiid) ??
- Stone Bond Enterprise Enabler Virtuoso
- TIBCOData Virtualization (formerly Cisco & Composite)
- Zetaris
- And many more …

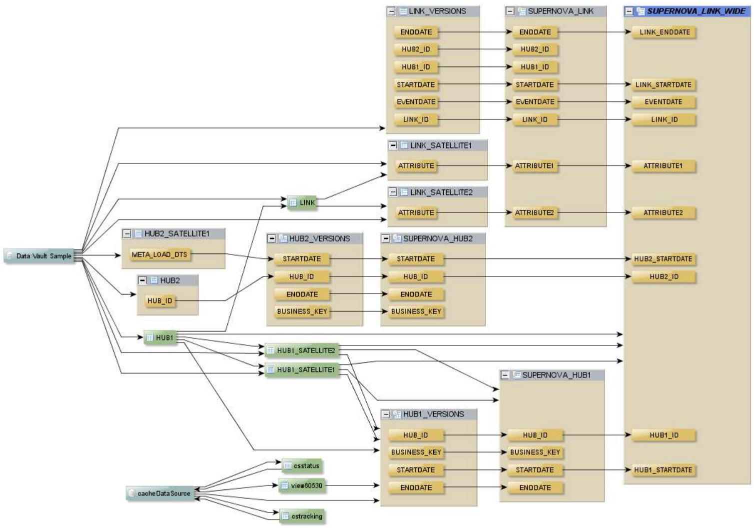
Developing Virtual Tables



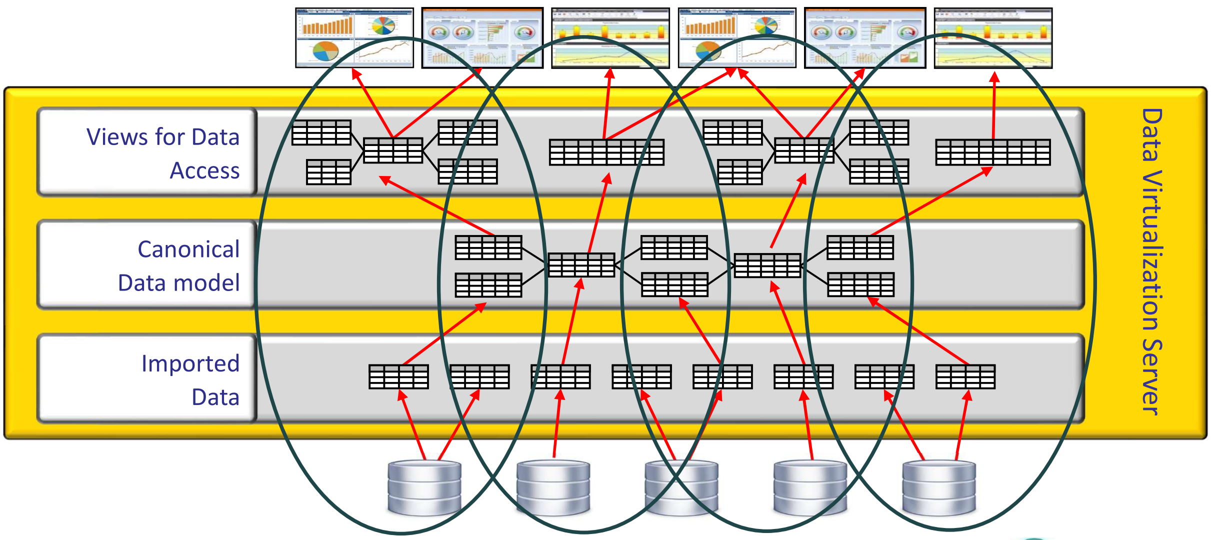
Layers of Virtual Tables



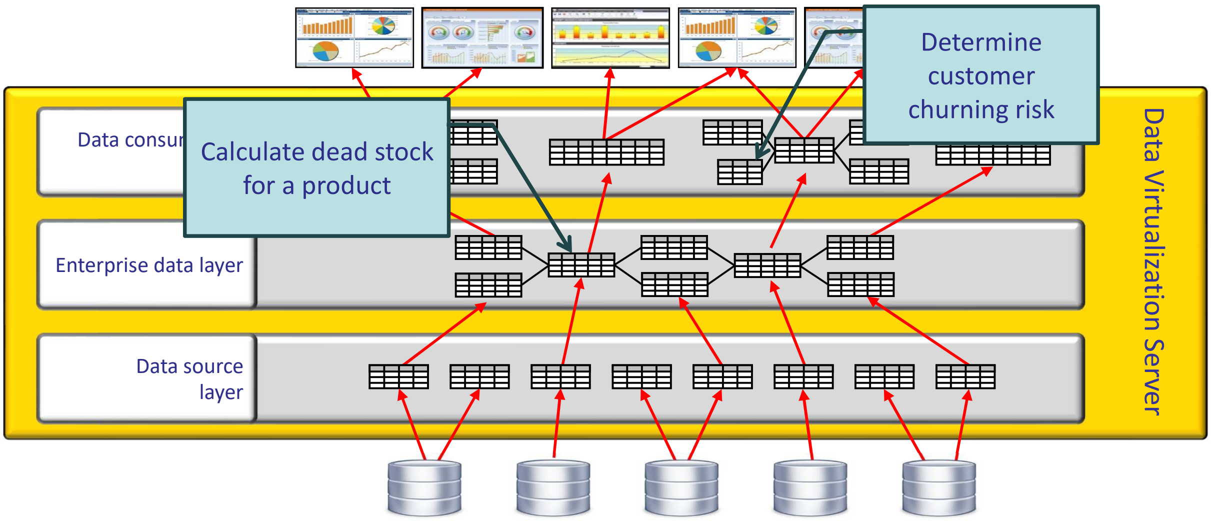
Lineage and Impact Analysis



Evolutionary Development Approach



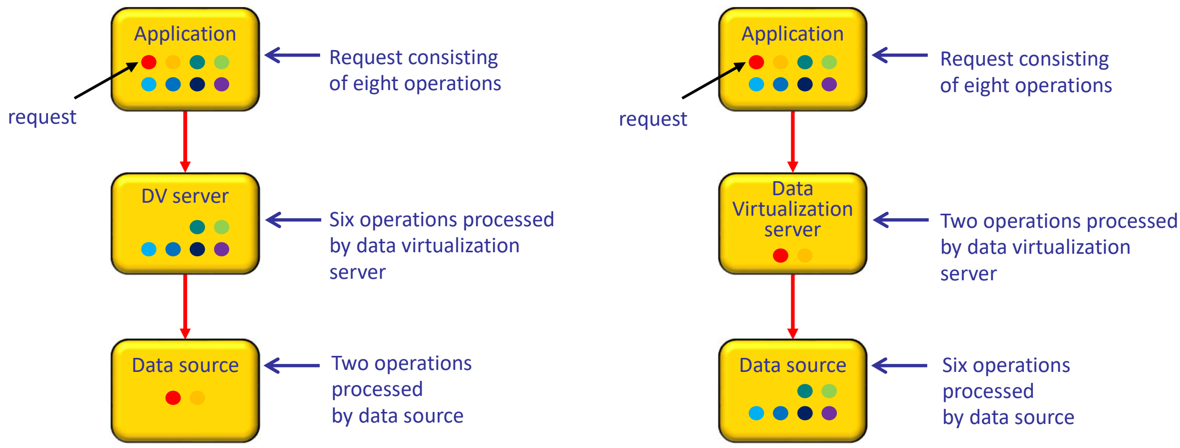
Improved Productity Through Sharing



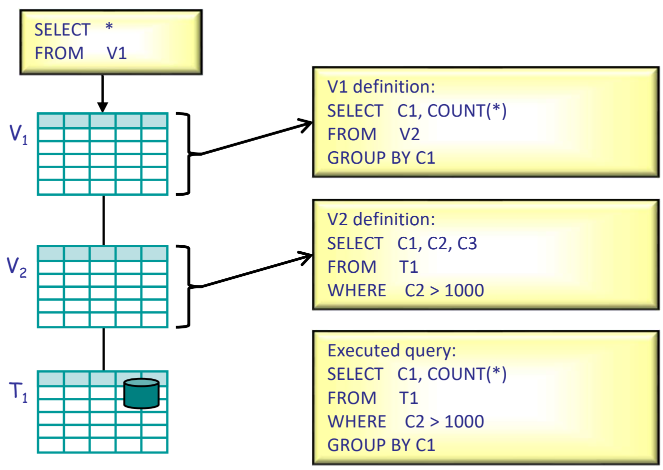
Performance Improving Features

- Easy-to-optimize queries
- Environment setup
- Query optimization
- Parallel processing and parallel pushdown
- Caching virtual tables
- The network
- Efficient drivers and connectors

Improved Performance Through Query Pushdown



Many Levels and One Query



Push Down Query Processing

Data Virtualization Server	(1) Incoming Query:	(3) Executed Query:

	`SELECT C2, CONCAT(C5, C6)`
	`FROM   Virtual table`
	`WHERE  C1 = > 1000`
	`AND    C4 BETWEEN 10 AND 20`
		`SELECT C2, CONCAT(C5, C6)`
		`FROM   Result`

Data Source	(2) Executed Query:

	`SELECT C2, C5, C6`
	`FROM   Table`
	`WHERE  C1 = > 1000`
	`AND    C4 BETWEEN 10 AND 20`

Accessing Files

Data Virtualization Server	(1) Incoming Query:	(3) Executed Query:

	`SELECT C2, CONCAT(C5, C6)`
	`FROM   Virtual table`
	`WHERE  C1 = > 1000`
	`AND    C4 BETWEEN 10 AND 20`
		`SELECT C2, CONCAT(C5, C6)`
		`FROM   Result`
		`WHERE  C1 = > 1000`
		`AND    C4 BETWEEN 10 AND 20`

Data Source	(2) Executed Query:
	`SELECT C1, C2, C4, C5, C6` `FROM   File`

Inferred Filters

Data Virtualization Server

(1) Incoming Query:

```
SELECT T1.C1, T1.C2, T2.C2
FROM   T1, T2
WHERE  T1.C1 = T2.C1
AND    T1.C1 => 1000
AND    T2.C2 BETWEEN 10 AND 20
```

(3) Executed Query:

```
SELECT T1.C1, T1.C2, T2.C2 FROM
T1, T2
WHERE  T1.C1 = T2.C1
```

Data Source

(2a) Executed Query:

```
SELECT C1, C2
FROM   T1
WHERE  C1 => 1000
```

Data Source

(2b) Executed Query:

```
SELECT C1, C2
FROM   T2
WHERE  T2.C1 => 1000
AND    T2.C2 BETWEEN 10
       AND 20
```


Join of Data Sources with Query Injection

Data Virtualization Server

(1) Incoming Query:

```
SELECT Tbig.C1, Tbig.C2
FROM   Tsmall, Tbig
WHERE  Tsmall.C1 = Tbig.C1
```

(4) Executed Query:

```
SELECT *
FROM   Result
```

Data Source

(2) Executed Query:

```
SELECT C1
FROM   Tsmall
```

Data Source

(3) Executed Query:

```
SELECT C1, C2
FROM   Tbig
WHERE  C1 in
       (v1, v2, v3, …)
```

Join of Data Sources with Ship Join

Data Virtualization Server

(1) Incoming Query:

```
SELECT T1.C1, T1.C2, T2. C2
FROM   Tbig1, Tbig2
WHERE  Tbig1.C1 = Tbig2.C1
```

(4) Executed Query:

```
SELECT *
FROM   Result
```

Data Source

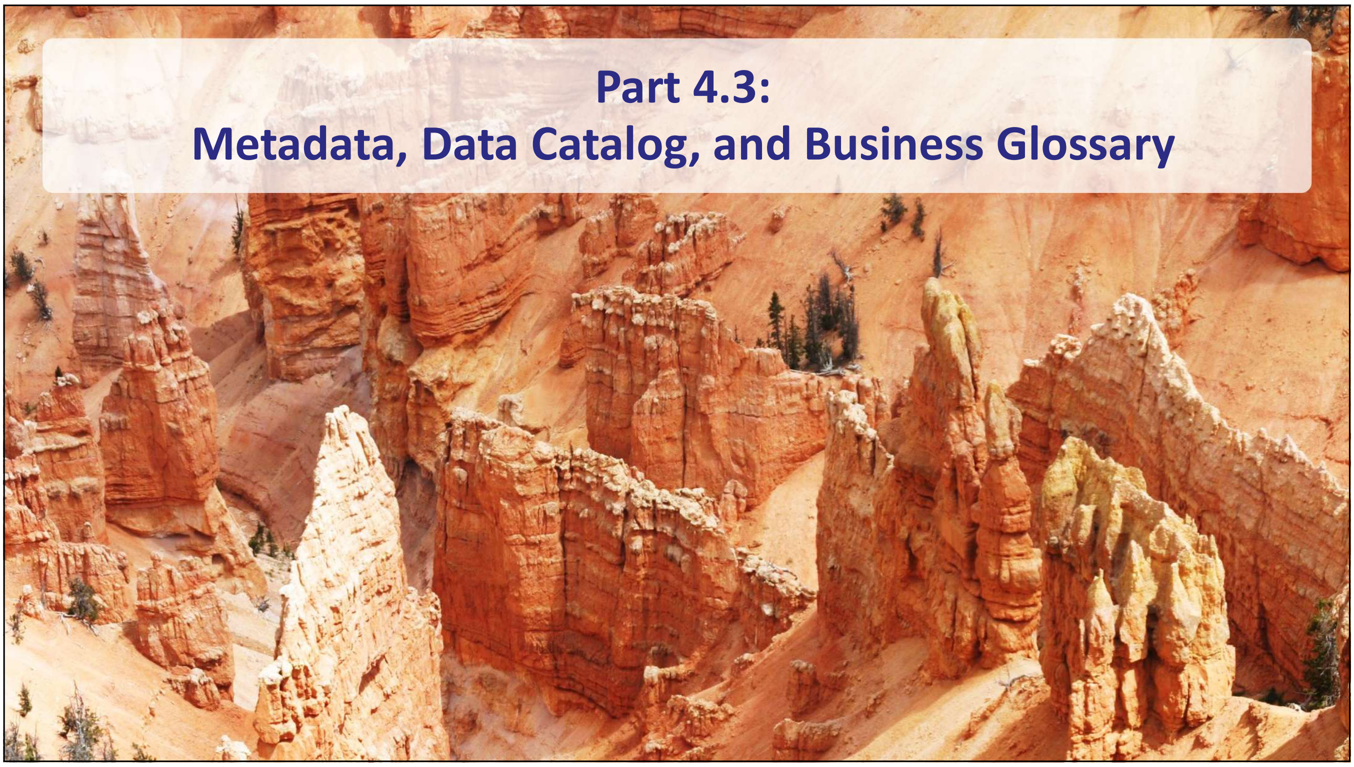
(2) Executed Query:

```
SELECT C1, C2
FROM   Tbig1
```

Data Source

(3) Executed Query:

```
CREATE TEMP TABLE TEMP;
INSERT INTO TEMP
SELECT *
FROM   Tbig1;
SELECT TEMP.C1, TEMP.C2,
       Tbig2.C2
FROM   TEMP, Tbig2
WHERE  TEMP.C1 = Tbig2.C1;
DROP TEMPORARY TABLE TEMP;
```

Part 4.3: Metadata, Data Catalog, and Business Glossary

Types of Metadata

Metadata on Data

- Textual definition
- Description
- Annotations by users and IT specialists
- Data lineage including transformations
- Retention information
- Qualifications: trustworthiness, completeness, data quality, …
- Value descriptions
- Original or masked/anonymized
- Ontology
- Owner and support
- …

Metadata on Metadata

- Retention information on metadata
- History of metadata
- Qualifications: trustworthiness, completeness, data quality, …
- Metadata value descriptions
- Original or masked/anonymized metadata
- Owner and support
- …

Metadata in 1976

DE DATA DICTIONARY/DIRECTORY (DD/D)

door L. Delport

1.1 **Wat is een DD/D? (Data Dictionary/Directory) (gegevenskataloog)**
Zeer algemeen kunnen we een DD/D als volgt bepalen:
Een DD/D is een katalogus die de omschrijving bevat van alle informatie-elementen die in een bedrijf bestaan.' Deze bepaling laat natuurlijk de weg open

1.2 **Waarom hebben we een DD/D nodig?**
Centralisatie, het vermijden van dubbele gegevens en het opzoeken en vinden van informatie langs allerlei wegen en door middel van allerlei sleutels zijn technieken eigen aan M.I.S. (7) en systemen van gegevensbanken.

Informatie jaargang 18 nr. 7/8 pag. 430 t/m 491 Amsterdam juli/augustus 1976

A Target Audience Shift of Metadata

IT Specialists

- Developers
- DBAs
- Designers
- Testers
- …

→

Business Users

- Self-service BI users
- Power users
- Dashboard users
- Data scientists
- Auditors
- …

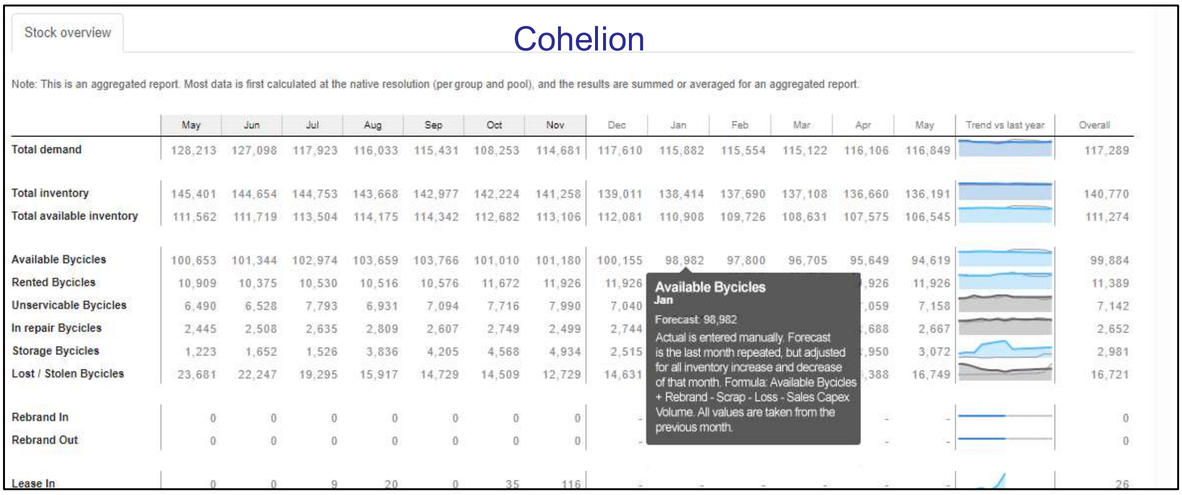
Numerous Solutions that Manage Metadata

- Home made metadata systems
- Professional data catalogs and business glossaries: Alation, Apache Atlas, Collibra, Informatica, TIBCO EBX, …
- Scraping and linking: ASG, Manta, SQLdep (Collibra), …
- Data warehouse automation: Astera, Attunity Compose, BiGenius, TimeXtender, WhereScape, …
- BI tools: semantic layers
- ETL tools
- Data profiling tools
- Data quality tools
- Data virtualization servers: Data Virtuality, Denodo, Fraxses, Tibco DV, …
- And many more …

Study Metadata Needs of Data Consumers (1)

- Metadata on which objects are required by which user
  - Metadata on data elements, reports/dashboards, data science models, business rules, …
- How do they want to access metadata?
  - Instant metadata (integrated within BI dashboard)
  - Via a service interface
  - Search interface on metadata
  - Do they need an ontology?
- Need for adding personal annotations to metadata
  - Annotation on tables and columns
  - Annotations on individual data values
  - Annotations on aggregated/derived values

Instant Metadata

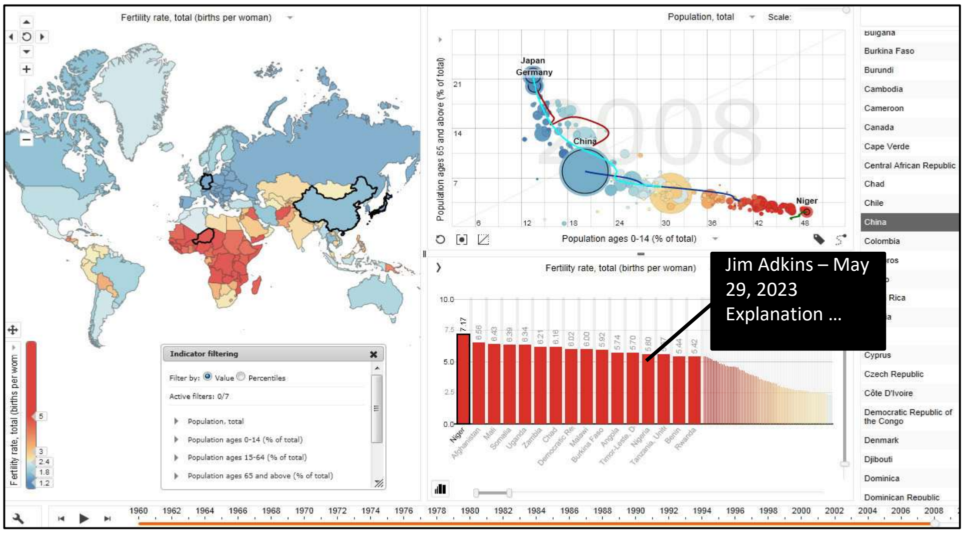


	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Trend vs last year	Overall
Total demand	128,213	127,098	117,923	116,033	115,431	108,253	114,681	117,610	115,882	115,554	115,122	116,106	116,849		117,289
Total inventory	145,401	144,654	144,753	143,668	142,977	142,224	141,258	139,011	138,414	137,690	137,108	136,660	136,191		140,770
Total available inventory	111,562	111,719	113,504	114,175	114,342	112,682	113,106	112,081	110,908	109,726	108,631	107,575	106,545		111,274
Available Bycicles	100,653	101,344	102,974	103,659	103,766	101,010	101,180	100,155	98,982	97,800	96,705	95,649	94,619		99,884
Rented Bycicles	10,909	10,375	10,530	10,516	10,576	11,672	11,926	11,926				,926	11,926		11,389
Unservicable Bycicles	6,490	6,528	7,793	6,931	7,094	7,716	7,990	7,040				,059	7,158		7,142
In repair Bycicles	2,445	2,508	2,635	2,809	2,607	2,749	2,499	2,744				,688	2,667		2,652
Storage Bycicles	1,223	1,652	1,526	3,836	4,205	4,568	4,934	2,515				,950	3,072		2,981
Lost / Stolen Bycicles	23,681	22,247	19,295	15,917	14,729	14,509	12,729	14,631				,388	16,749		16,721
Rebrand In	0	0	0	0	0	0	0	-				-	-		0
Rebrand Out	0	0	0	0	0	0	0	-				-	-		0
Lease In	0	0	9	20	0	35	116	-	-	-	-	-	-		26

Study Metadata Needs of Data Consumers (1)

- Metadata on which objects are required by which user
  - Metadata on data elements, reports/dashboards, data science models, business rules, calculations, retention specifications, …
- How do they want to access metadata?
  - Instant metadata (integrated within BI dashboard)
  - Via a service interface
  - Search interface on metadata
  - Do they need an ontology?
- Need for adding personal annotations to metadata
  - Annotation on tables and columns
  - Annotations on individual data values
  - Annotations on aggregated/derived values

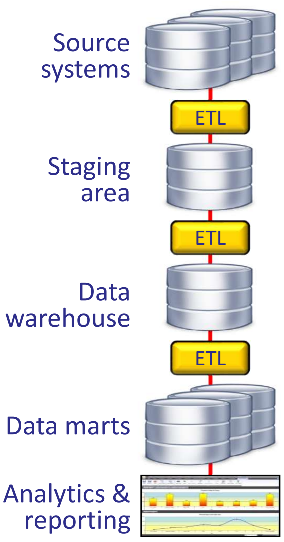
Adding Annotations on Data Points



Study Metadata Needs of Data Consumers (2)

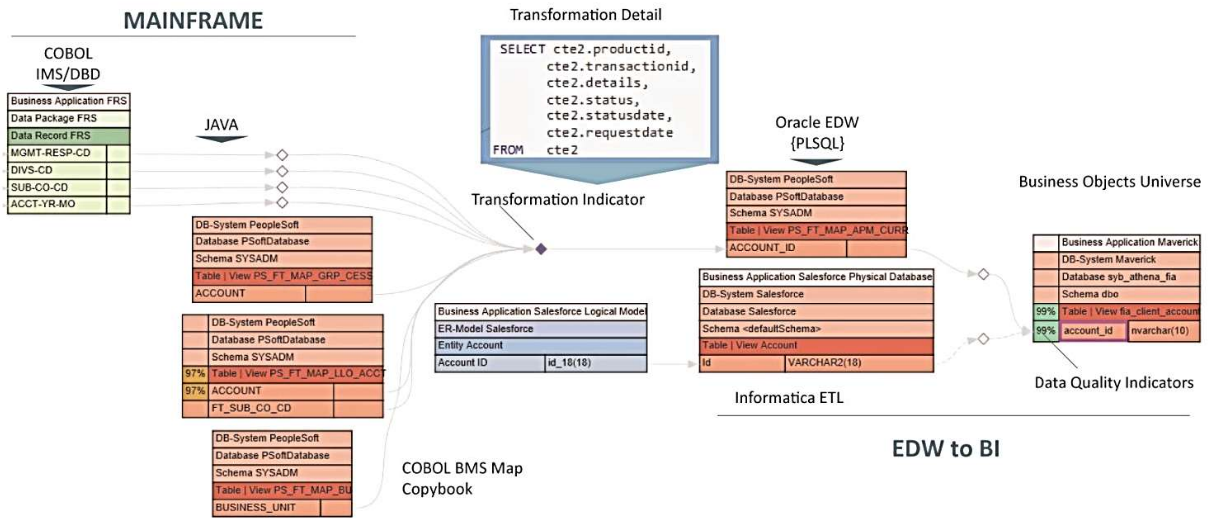
- Metadata tagging
  - Retention period, real or anonymized, responsible data steward, trustworthiness, completeness, data quality, …
  - Taggings added by business users
- Versioning of metadata
- Automatic notifications
  - a retention period of frequently used data is about to expire
  - a definition has changed
  - a piece of legislation is about to change
- Fuzzy boundaries between metadata and master data
  - Are all the state codes metadata or master data?

Metadata is Dispersed

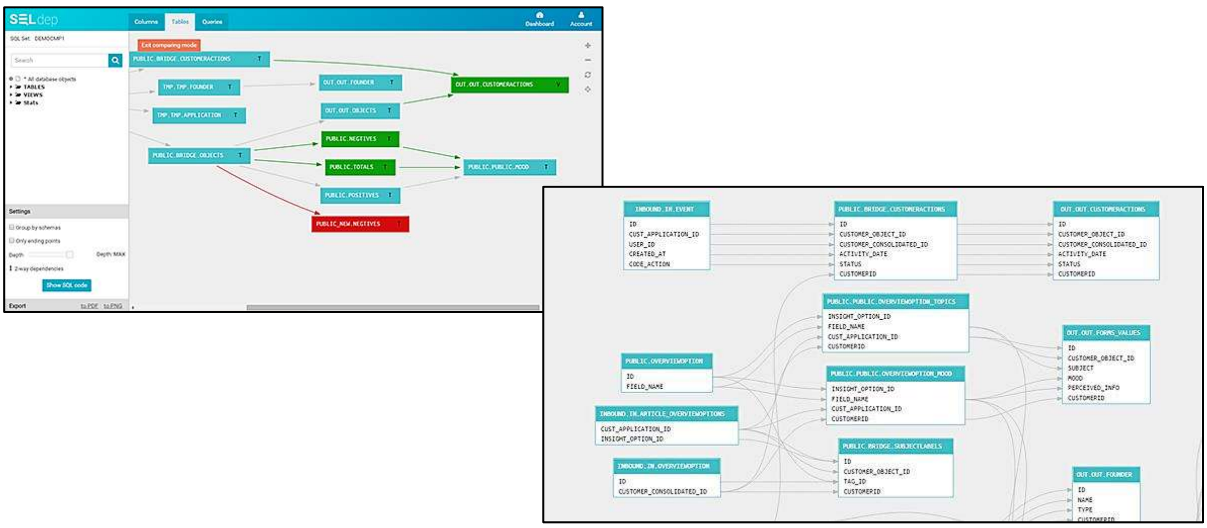


- Metadata dispersed across many systems
  - In database servers (system tables)
  - In integration tools
  - In documentation
  - In reporting tools (semantic layer)
  - In spreadsheets
  - In application code
  - And many more …
- Most is technical and not business metadata
- Not integrated – no clear relationships between metadata elements

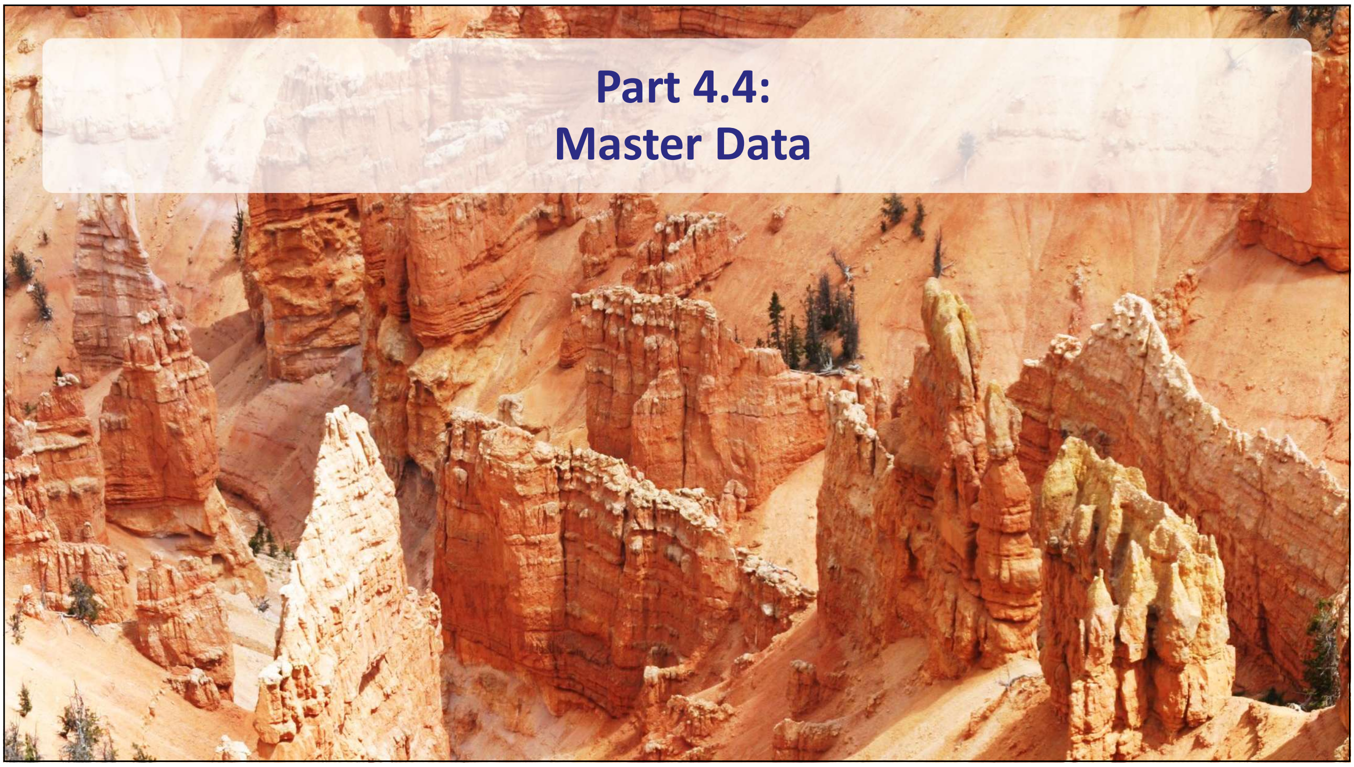
Example: Lineage Through Scraping by ASG



Example: Lineage Through Scraping by SQLdep (Collibra)



Part 4.4: Master Data



Master Data Management: Two Customer Tables

Customer table in **Sales** System

ID	Name	Initials	Date Entered	City	State
12345	Young	N	Aug 4, 2008	San Francisco	CA
23324	Stills	S	Sep 10, 2009	New Orleans	LA
57657	Furay	R	Oct 16, 2010	Yellow Springs	OH
65461	Palmer	B	Nov 22, 2011	Boston	MA
…	…	…	…	…	…

Customer table in **Finance** System

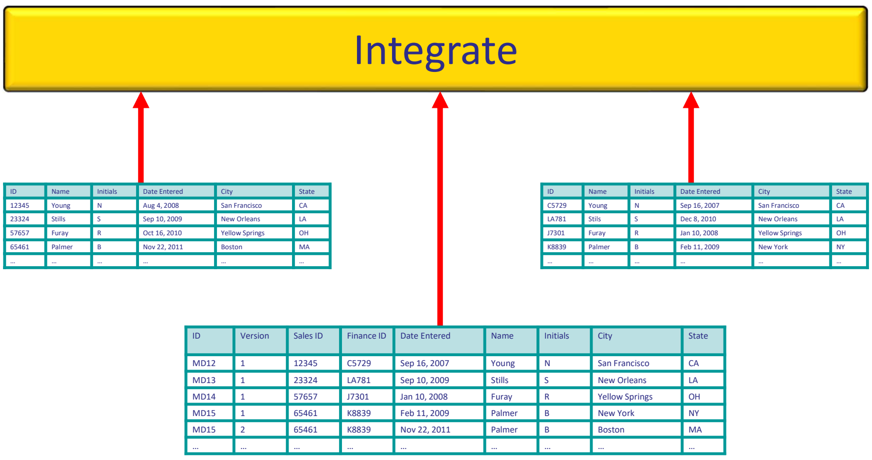
ID	Name	Initials	Date Entered	City	State
C5729	Young	N	Sep 16, 2007	San Francisco	CA
LA781	Stils	S	Dec 8, 2010	New Orleans	LA
J7301	Furay	R	Jan 10, 2008	Yellow Springs	OH
K8839	Palmer	B	Feb 11, 2009	New York	NY
…	…	…	…	…	…

The Master Customer Table

Master Customer table

ID	Version	Sales ID	Finance ID	Date Entered	Name	Initials	City	State
MD12	1	12345	C5729	Sep 16, 2007	Young	N	San Francisco	CA
MD13	1	23324	LA781	Sep 10, 2009	Stills	S	New Orleans	LA
MD14	1	57657	J7301	Jan 10, 2008	Furay	R	Yellow Springs	OH
MD15	1	65461	K8839	Feb 11, 2009	Palmer	B	New York	NY
MD15	2	65461	K8839	Nov 22, 2011	Palmer	B	Boston	MA
...	...	...	...	...	...	...	...	...

Joining Needs Master Data

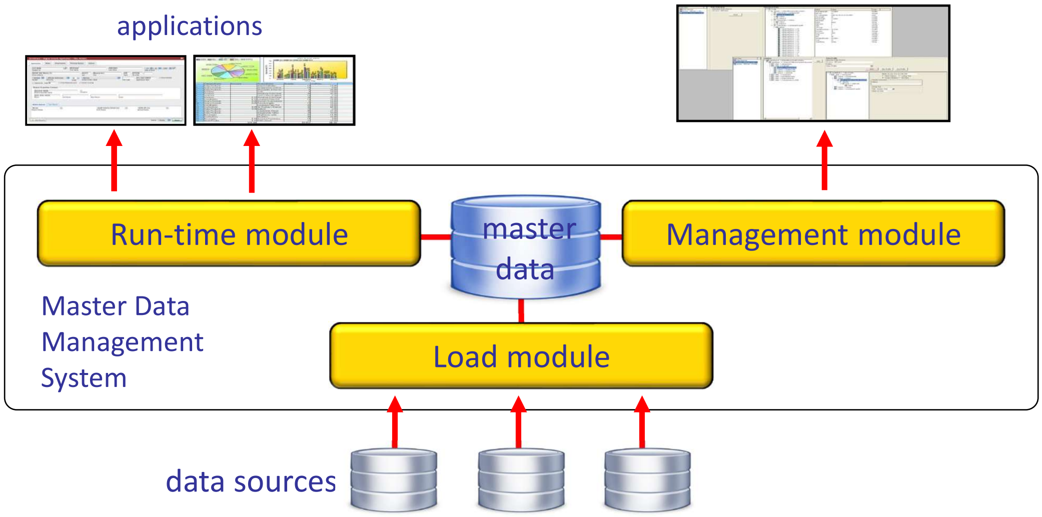


ID	Name	Initials	Date Entered	City	State
12345	Young	N	Aug 4, 2008	San Francisco	CA
23324	Stills	S	Sep 10, 2009	New Orleans	LA
57657	Furay	R	Oct 16, 2010	Yellow Springs	OH
65461	Palmer	B	Nov 22, 2011	Boston	MA
...	...	...	...	...	...

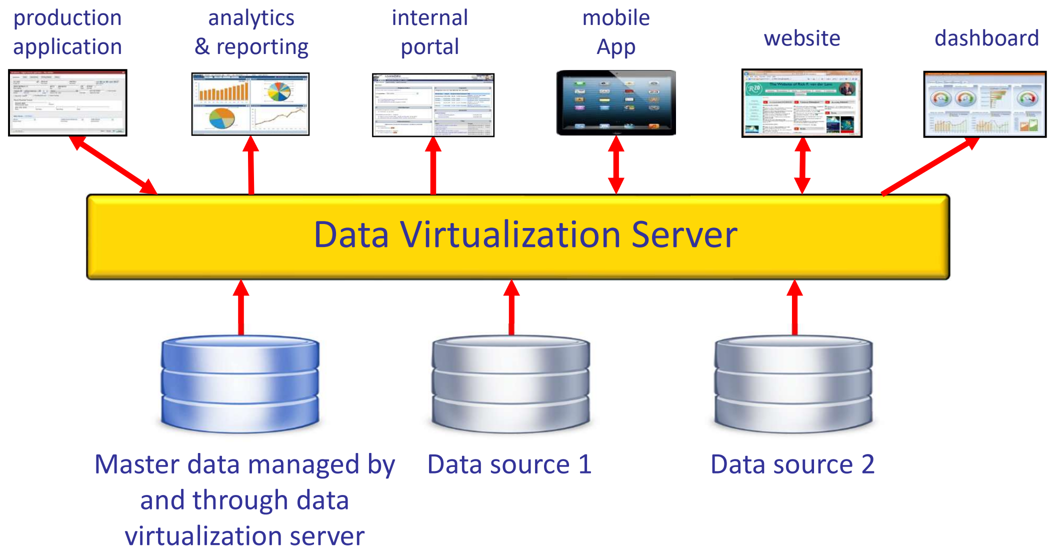
ID	Name	Initials	Date Entered	City	State
C5729	Young	N	Sep 16, 2007	San Francisco	CA
LA781	Stils	S	Dec 8, 2010	New Orleans	LA
J7301	Furay	R	Jan 10, 2008	Yellow Springs	OH
K8839	Palmer	B	Feb 11, 2009	New York	NY
...	...	...	...	...	...

ID	Version	Sales ID	Finance ID	Date Entered	Name	Initials	City	State
MD12	1	12345	C5729	Sep 16, 2007	Young	N	San Francisco	CA
MD13	1	23324	LA781	Sep 10, 2009	Stills	S	New Orleans	LA
MD14	1	57657	J7301	Jan 10, 2008	Furay	R	Yellow Springs	OH
MD15	1	65461	K8839	Feb 11, 2009	Palmer	B	New York	NY
MD15	2	65461	K8839	Nov 22, 2011	Palmer	B	Boston	MA
...	...	...	...	...	...	...	...	...

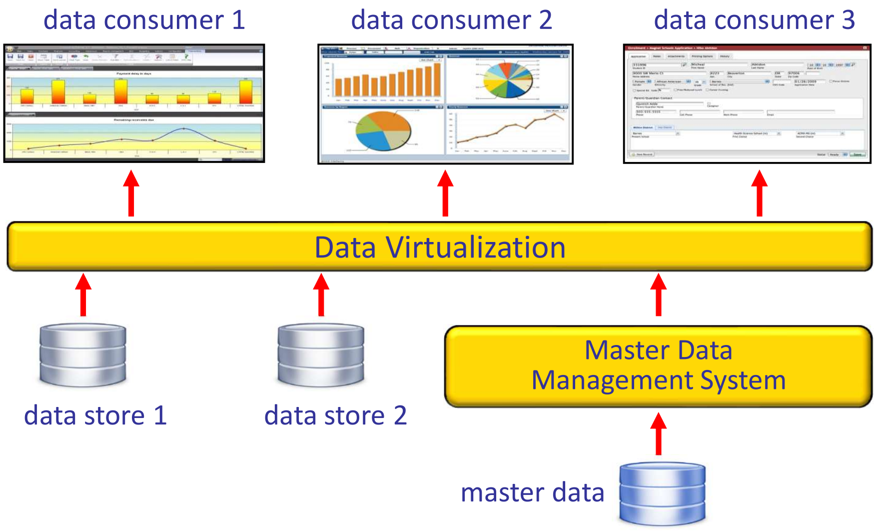
Overall Architecture of an MDM System



Lightweight Solution for Master Data



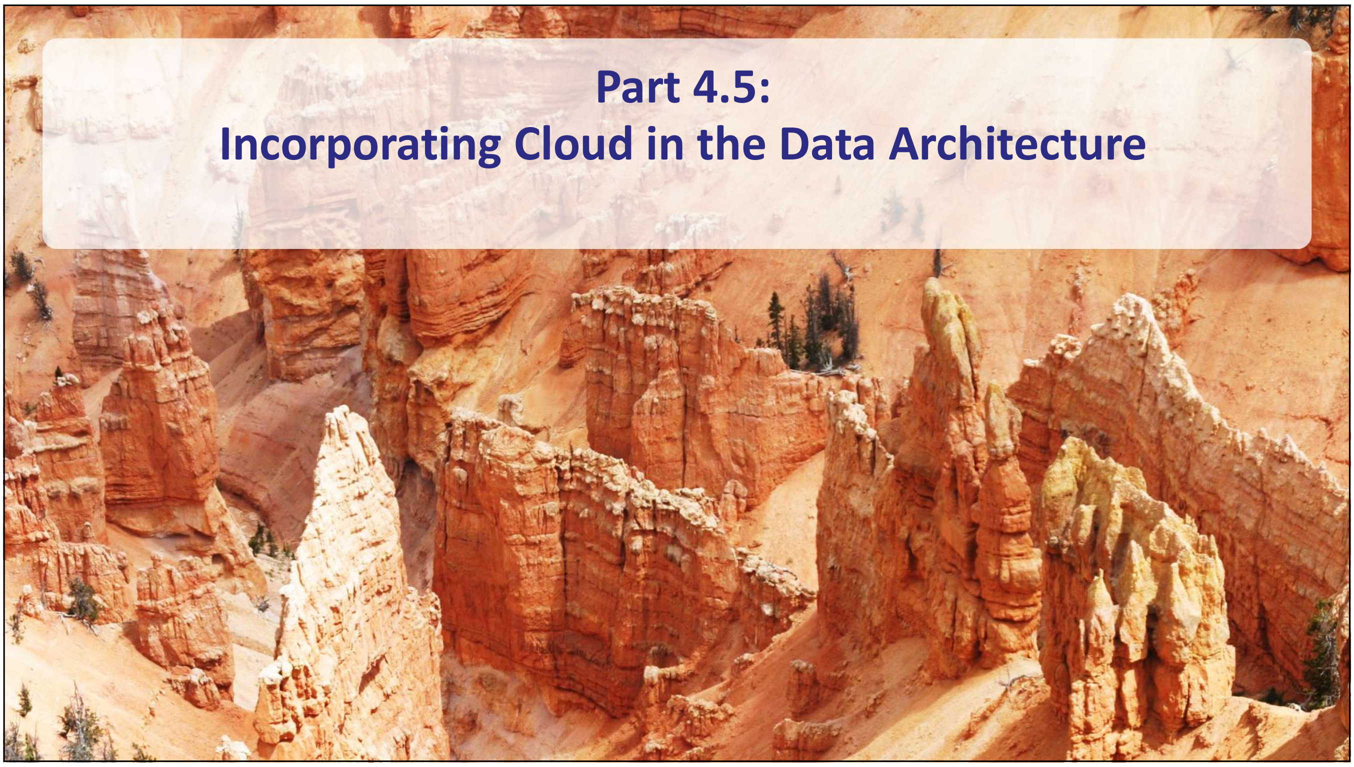
MDM as Source for Data Virtualization



Example: Cohelion (Master Data-driven BI)

Accounts Payable	30,809	35,972	37,370	37,668	37,346	33,360	34,141	42,505
Profit - Loss Accounts								
Gross profit	79,150	56,450	61,467	77,134				60,400
EBIT	11,327	-732	-1,247	11,101				-374
Sales Income	201,736	167,789	170,810	222,655	199,000	185,254	7,600	131,000
Rental Income	0	0	0	0	0	0	0	0
Other income	0	0	0	0	0	0	0	0
Expense Accounts								
Office Expense	812	681	790	532	960	372	1,010	455

Actual	**185,254.00**
Manual	20,248.00
SalesForce - Feed	30,855.00
SAP - Feed	134,151.00


Part 4.5:
Incorporating Cloud in the Data Architecture


Cloud Platforms are Becoming the New Mainframes

Mainframe = Lock In

- Proprietary operating systems
- Proprietary system management software
- Proprietary database servers
- Proprietary security systems
- Proprietary development environments
- Proprietary JCLs
- Proprietary …

Cloud Platform = Lock In?

- Proprietary operating systems
- Proprietary management software
- Proprietary database servers
  - E.g. Amazon: RDS, RedShift (SQL), S3, …
- Proprietary security systems
- Proprietary development environments
  - E.g. Microsoft Azure: Reporting Services, Analytics services, Data Management Services, …
- Proprietary …

Data Storage Technologies Available on Cloud Platforms

Cloud Platform	Data Storage Technology
Amazon AWS	Aurora DocumentDB DynamoDB Elasticache for Redis RDS Redshift S3 Timestream
Google	BigQuery Cloud Bigtable Cloud Firestore Cloud Spanner Cloud SQL
Microsoft Azure	Cache for Redis Cosmos DB Data Lake SQL Database Synapse Analytics

Stay Cloud Platform Independent (IT sovereignty)

=

Design to Migrate

Watch Out For Egress Costs!

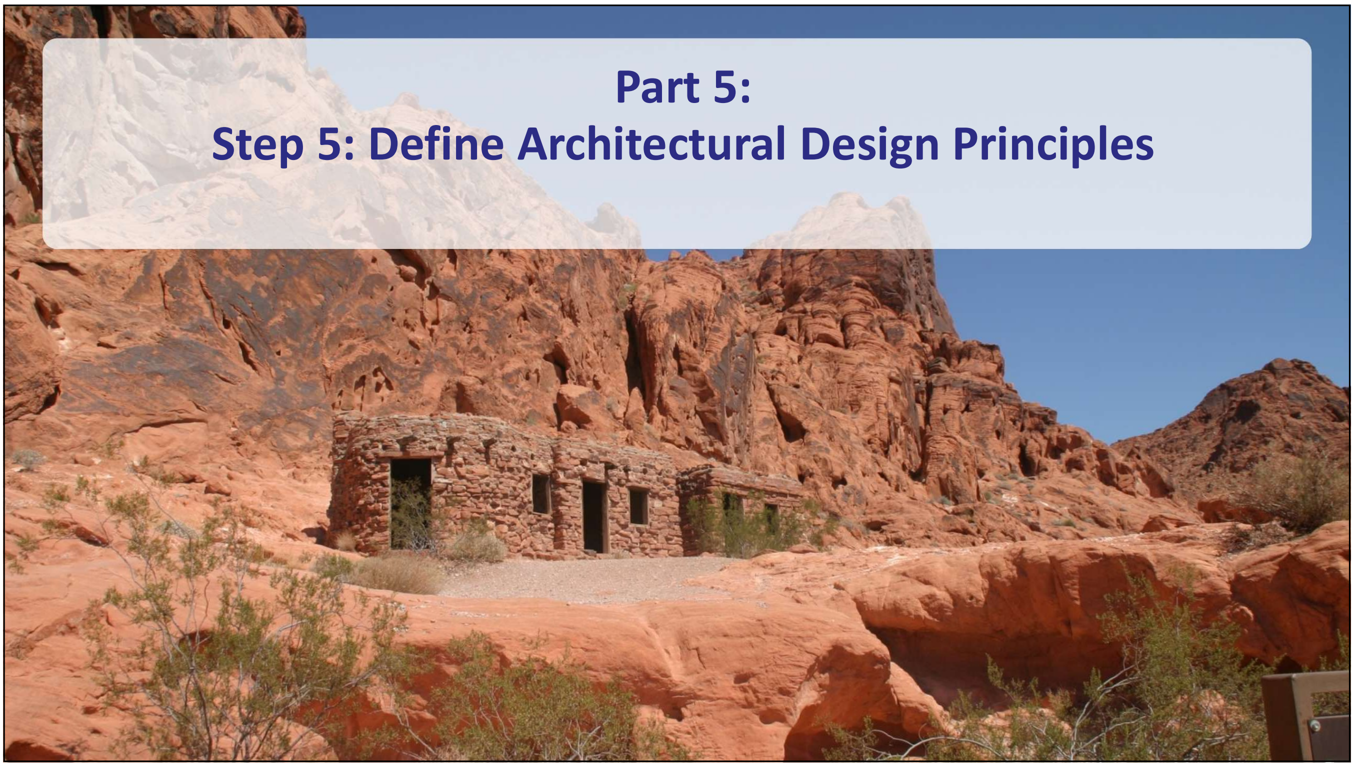
Public Cloud	Typical Data Egress Charge (per GB)	Cost to move 10 TB, *per month*	Discounted Data Egress Charge (per GB) for 100 TB	Cost to move 100 TB, *per month*
Azure	$0.08	$800	$0.07	$7,000
AWS	$0.02	$200	$0.02	$2,000
Google Cloud Platform	$0.11	$1100	$0.08	$8,000
Oracle	Free up to 10TB	free	$0.0085-$0.050 depending on geography	$850 to $5,000

Source: https://www.factioninc.com/blog/it-challenges/egress-charges-how-to-prevent-costs/

Cloud Platform Fees

- Fees can have influence on data architecture
- Example:
  - SnowflakeDB: pay for data usage (queries)
    - Store more derived data
  - Exasol: pay for environment size (queries for free)
    - Work with views in stead of physical data marts
- How well can the technology exploit the cloud platform?
  - E.g. cloud is endless MPP, what about the database server?
- Pushing processing into the cloud, close to where data is produced

Adopt New Technologies, But Don't Stick to Old Ideas

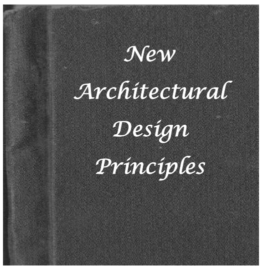


Part 5: Step 5: Define Architectural Design Principles

Forget Old Architectural Design Principles

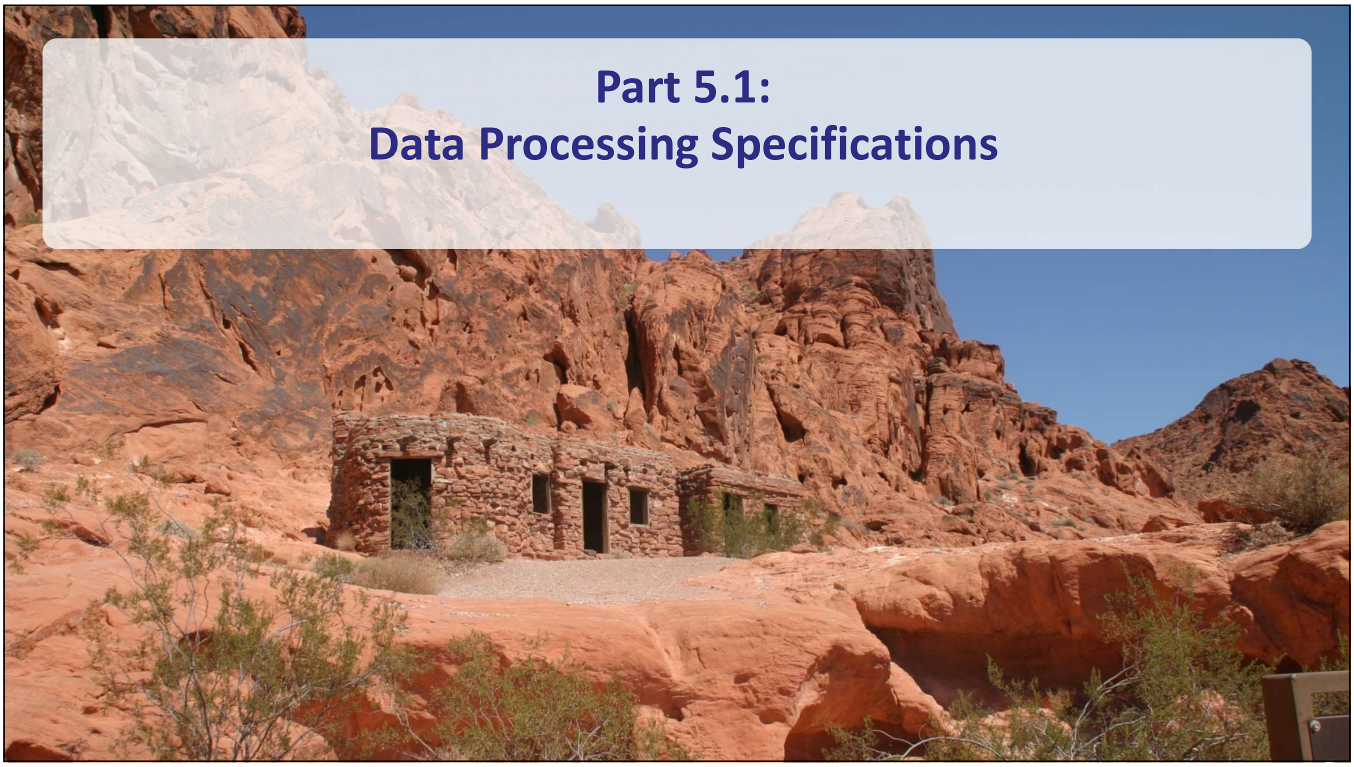
- No reporting on the production database
  - Reporting and transaction workloads clash
- *Physical* data marts are needed to improve reporting performance
- Data marts need a star schema design to speed up analytical queries
- ETL is used to transform data
  - Batch oriented
- When SQL databases are used
  - Indexes are required to improve query performance
  - Use locking for concurrency management
  - Not ideal for MPP
  - Need constant tuning by DBA
- ...

Examples of Architectural Design Principles

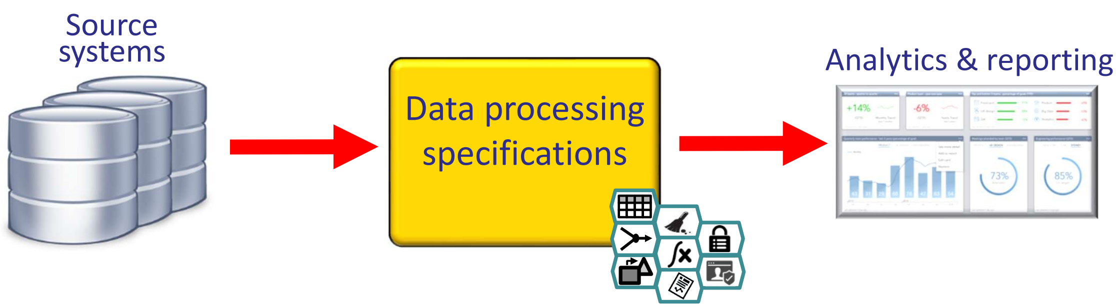


- Centralized and active data processing specifications
  - Searchable definitions and descriptions for technical and business users
  - Lineage and impact analysis
- One universal architecture for all forms of data consumption
  - Standard reporting, self-service BI, apps, data science, …
- Data storage and access technology agnostic
  - Hadoop, SQL, cubes, …
  - Abstraction
- Push the processing to the data, not the data to the processing
  - Decentralized data production
  - Edge analytics
  - Hyper-decentralized data production and storage
- Generator-driven
- ...

Part 5.1: Data Processing Specifications



The Data Processing Specifications

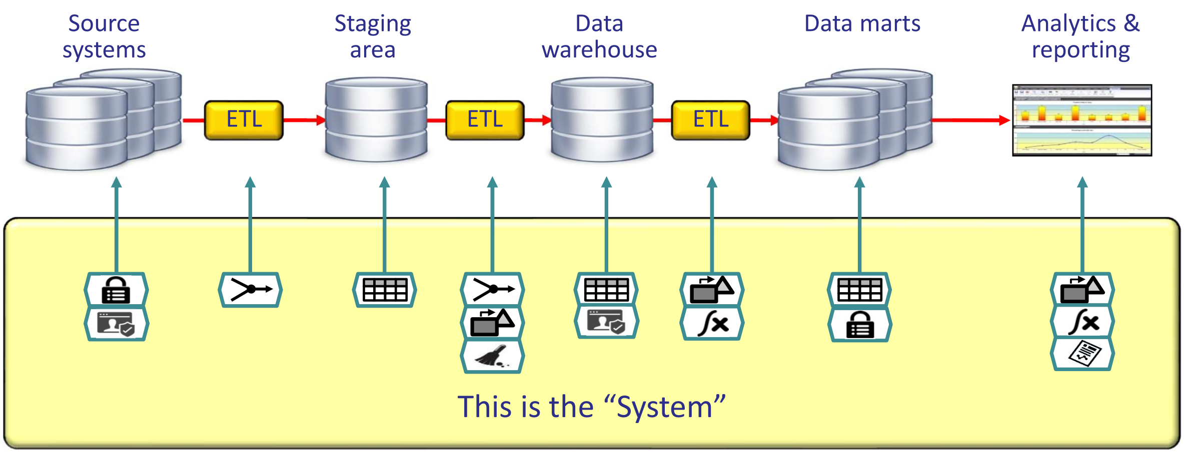


- Data structure specifications
- Integration specifications
- Transformation specifications
- Data security specifications
- Data cleansing specifications
- Analytical specifications
- Visualization specifications
- Data privacy specifications

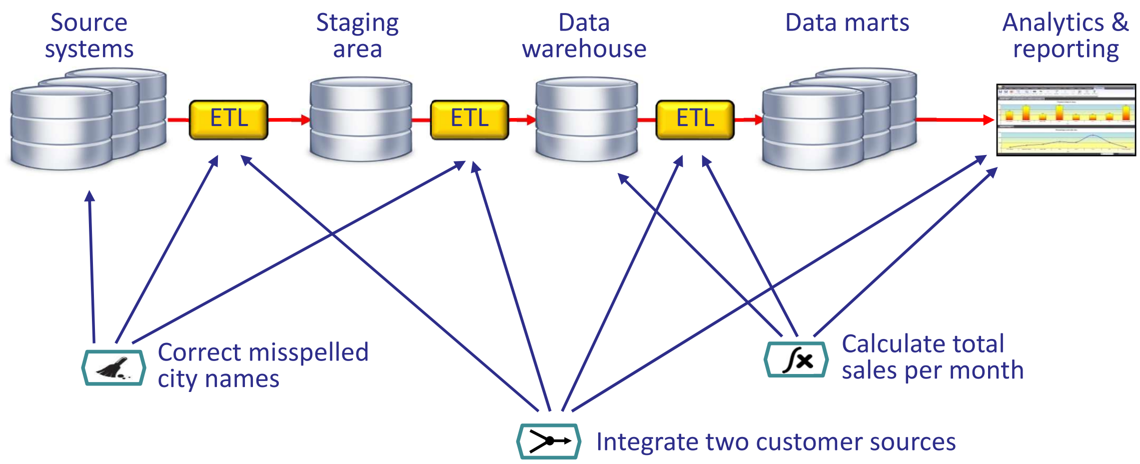
Examples of Data Processing Specifications

- Data value transformations
- Data structure transformations
- Aggregations
- Filters
- Calculations
- Integrations
- Technical corrections
- Functional corrections
- Anonymizations
- Authorizations and authentications
- Historizations
- Metadata-related specifications
- …

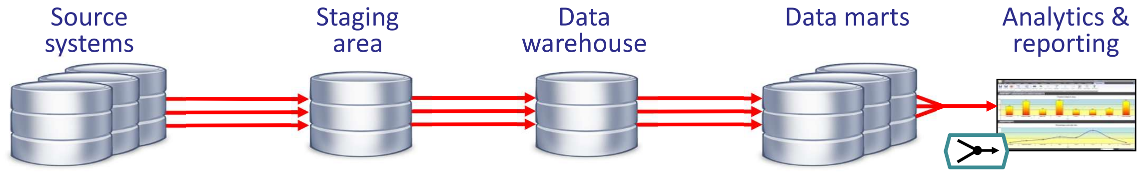
Data Processing Specifications



Where to Implement Data Processing Specifications?

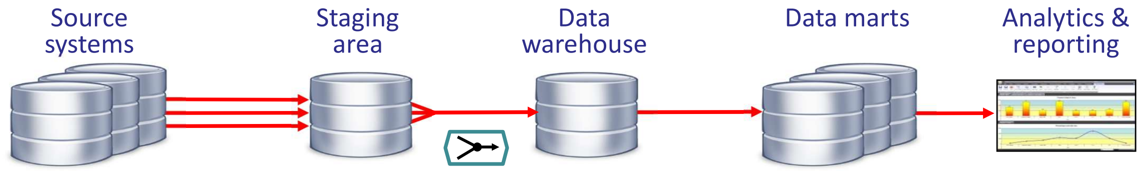


Example: Integration and Aggregation (1)



- Slow processing of integration logic in reports
- Complex queries in reports
- No sharing of integration logic across reports and tools
- Potential errors and inconsistencies in reports
- Fast copying and lower data latency
- Data structure of source database determines all data structures
- Use of original raw data possible
- What about integration errors?

Example: Integration and Aggregation (2)

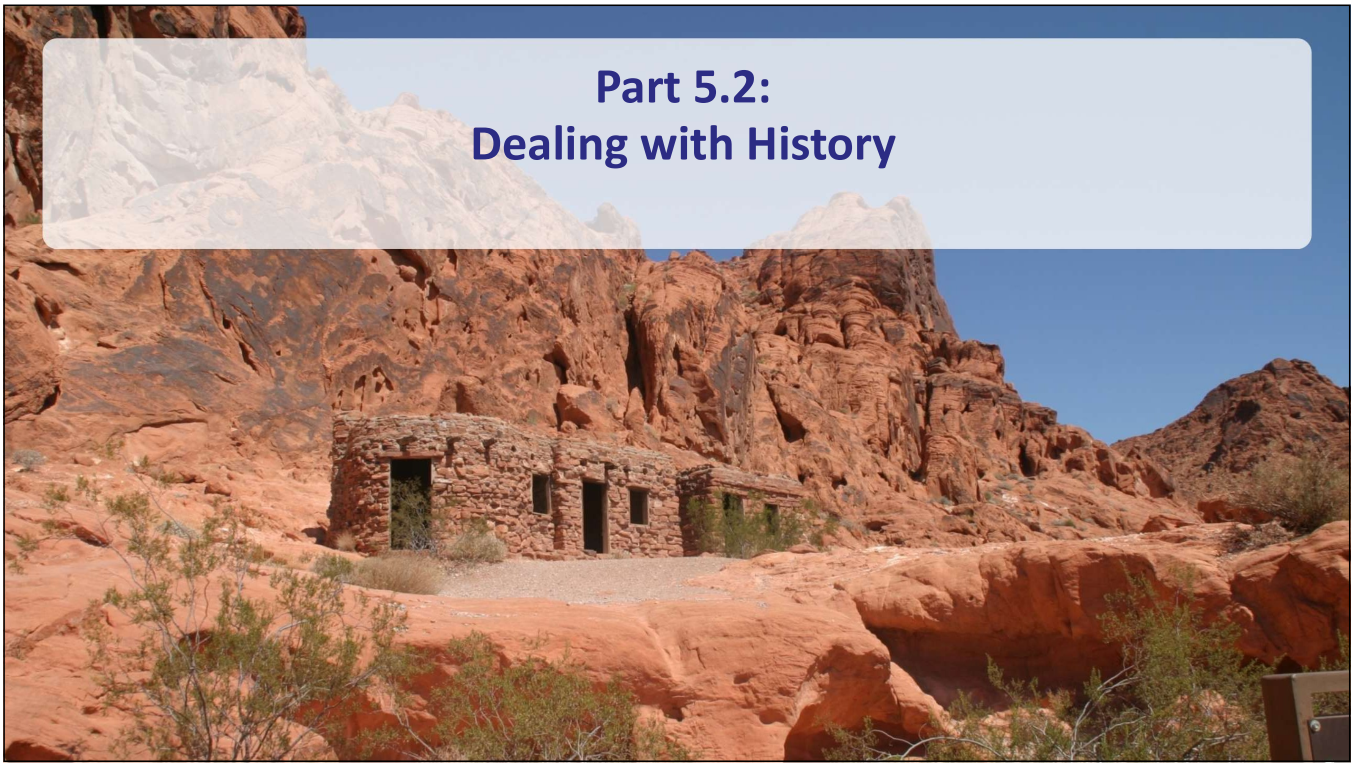


- Fast processing of integration logic in reports
- Simpler queries in reports
- Sharing of integration logic across reports and tools
- Potential errors and inconsistencies in ETL
- Slower copying and higher data latency
- Data structure of source database does not determine all data structures
- Use of original raw data possible
- Integration errors easier to fix

Implementing the Data Processing Specifications

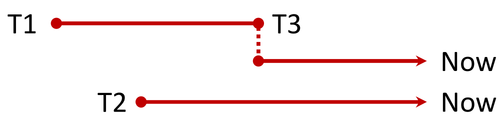
- Full history
- Permanent surrogate keys
- No cleansing
- No integration
- No deletes
- Data warehouse becomes "data lakish"

Part 5.2:
Dealing with History



Modeling History: Simple History for Updates

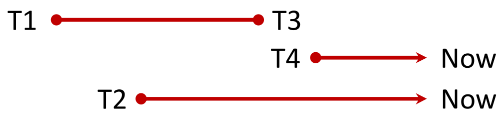
Cid	City	Start	End
C1	London	T1	T3
C1	Leicester	T3	Now
C2	Paris	T2	Now



- No gaps in history
- Only one value for an object on a specific datetime
- Supports following queries:
  - What is the current value – Where End = Now
  - What was the value on a specific datetime - Where date between Start and End

Modeling History: Simple History for Updates with Gaps

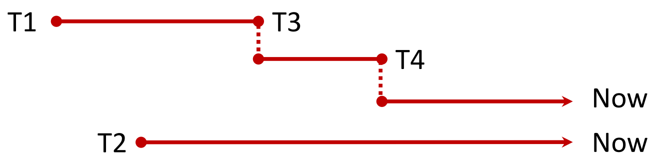
Cid	City	Start	End
C1	London	T1	T3
C1	Leicester	T4	Now
C2	Paris	T2	Now



- No gaps in history
- Only one value for an object on a specific datetime
- Supports following queries:
  - What is the current value – Where End = Now
  - What was the value on a specific datetime - Where date between Start and End - may return no values

Modeling History: Simple History for Updates Without Gaps

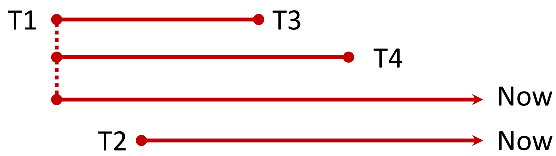
Cid	City	Start	End
C1	London	T1	T3
C1	Gap	T3	T4
C1	Leicester	T4	Now
C2	Paris	T2	Now



- No gaps in history
- Only one value for an object on a specific datetime
- Supports following queries:
  - What is the current value – Where End = Now
  - What was the value on a specific datetime - Where date between Start and End – always returns a value; sometimes nothing

Modeling History: Corrections

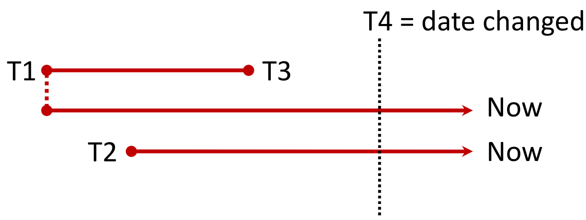
Cid	City	Start	End
C1	Londdon	T1	T3
C1	Londn	T1	T4
C1	London	T1	Now
C2	Paris	T2	Now



- No gaps in history
- Multiple values for an object on a specific datetime
- Supports following queries:
  - What is the current value – Where End = Now
  - What was the value on a specific datetime – Get the oldest where date between Start and End

Modeling History: Delayed Corrections

Cid	City	Start	End	Changed
C1	Londdon	T1	T3	T4
C1	London	T1	Now	
C2	Paris	T2	Now	



- Extra column required
- No gaps in history
- Multiple values for an object on a specific datetime
- Supports following queries:
  - What is the current value – Where End = Now
  - What was the value on a specific datetime – Get the oldest where date between Start and End

Modeling History: Logging Updates and Corrections

Cid	City	Start	End	Changed	Insert id	Change id
C1	London	T1	T2		Insert1	Update1
C1	Leicester	T2	Now		Insert2	
C2	Paris	T3	T4		Insert3	Update2
C2	Lyon	T4	T5		Insert4	Delete1

Change id	Who	When	Where	...
Insert1	User1	T1	...	...
Insert2	User2	T2	...	...
Insert3	User1	T3	...	...
Insert4	User3	T4	...	...
Update1	User2	T2	...	...
Update2	User4	T4	...	...
Delete1	User5	T5	...	...

- Log table for auditing purposes
- Batch inserts, updates, and deletes

Modeling Streaming Data: Single Values

Incoming Stream
T1, S1, Temp=50
T2, S1, Temp=52
T3, S2, Temp=51
T4, S3, Temp=49
T5, S1, Temp=52;
T5, S2, Temp=53

Key	Start	End	Sensor	Temp	Avg Temp
1		T1	S1	50	50
2	T1	T2	S1	52	51
5	T2	T5	S1	52	51,3
3		T3	S2	51	51
6	T3	T5	S2	53	52
4		T4	S3	49	49

- Key is unique artificial value
- Measurement is considered as temperature since previous measurement
- Sensor data is arriving in the right order

Modeling Streaming Data: Multiple Values

Incoming Stream
T1, S1, Temp=50
T2, S1, Temp=52
T3, S2, Temp=51
T4, S3, Temp=49
T5, S1, Temp=52; S2, Temp=53

Key	Start	End	Sensor	Temp	Avg Temp
1		T1	S1	50	50
2	T1	T2	S1	52	51
5	T2	T5	S1	52	51,3
3		T3	S2	51	51
6	T3	T5	S2	53	52
4		T4	S3	49	49

- Key is unique artificial value
- Stream records are flattened
- Measurement is considered as temperature since previous measurement
- Sensor data is arriving in the right order

Modeling Streaming Data: Delta Values

Incoming Stream
T1, S1, Temp=50
T2, S1, Temp=+2
T3, S2, Temp=51
T4, S3, Temp=49
T5, S1, Temp=+0
T5, S2, Temp=+2

Key	Start	End	Sensor	Temp	Avg Temp
1		T1	S1	50	50
2	T1	T2	S1	52	51
5	T2	T5	S1	52	51,3
3		T3	S2	51	51
6	T3	T5	S2	53	52
4		T4	S3	49	49

- Key is unique artificial value
- Measurement is considered as change in temperature
- Sensor data is arriving in the right order

Modeling Streaming Data: Log Data

Incoming Stream
T1, Insert, C1, London
T2, Update, C1, Leicester
T3, Insert, C2, Paris
T4, Insert, C3, Berlin
T5, Update, C1, Manchester
T5, Update, C3, Munich

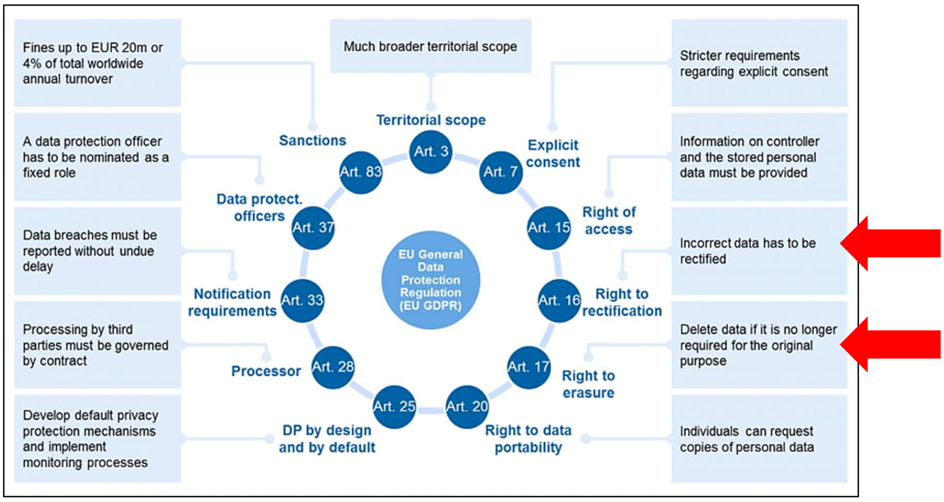
Cid	City	Start	End
C1	London	T1	T2
C1	Leicester	T2	T5
C1	Machester	T5	Now
C2	Paris	T3	Now
C3	Berlin	T4	T5
C3	Munich	T5	Now

- Business key used
- Stream is seen as data entry
- Careful with parallel inserts; order not unimportant
  - Loading with hashed keys?

GDPR – The Right to be Forgotten

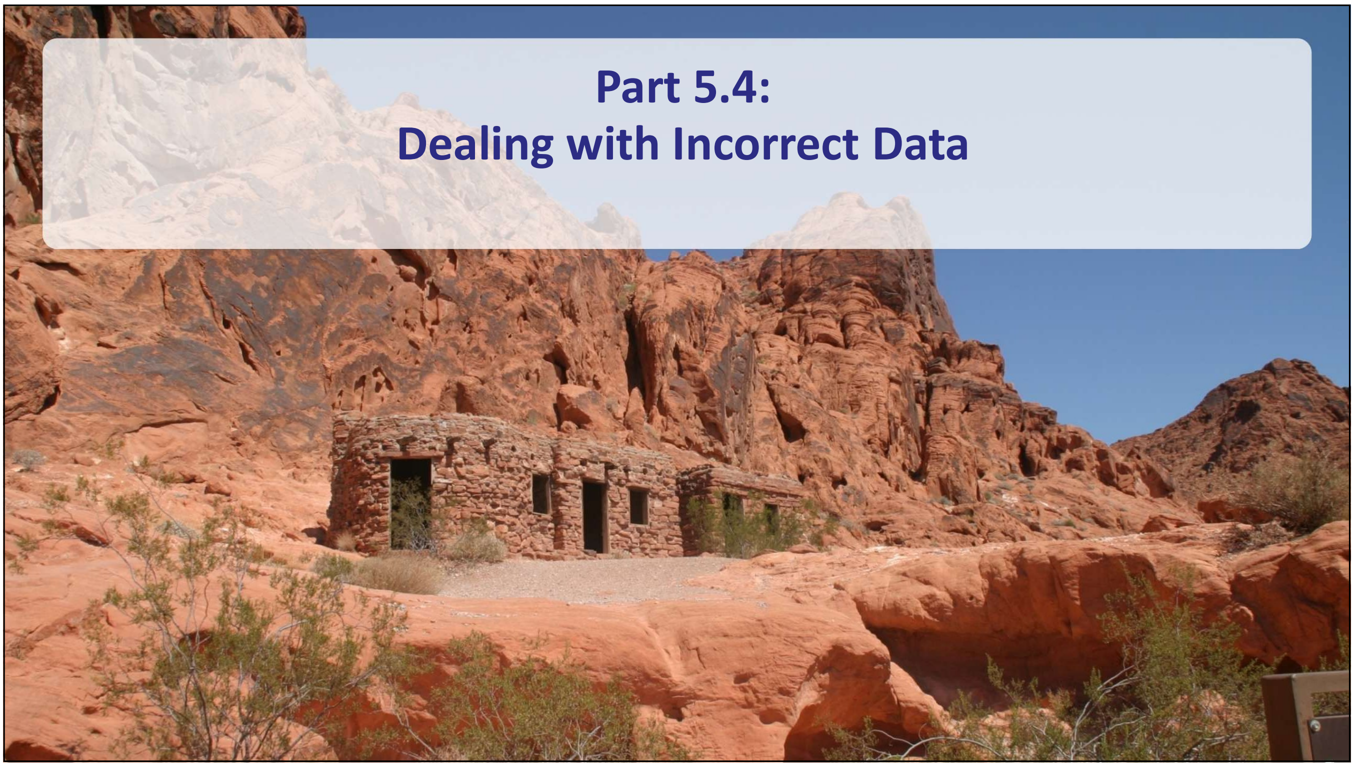
- Art. 17 GDPR Right to erasure (right to be forgotten)
  - The rule primarily regulates erasure obligations
- According to this, personal data must be erased immediately where the data are no longer needed for their original processing purpose, or the data subject has withdrawn his consent and there is no other legal ground for processing, the data subject has objected and there are no overriding legitimate grounds for the processing, or erasure is required to fulfill a statutory obligation under the EU law or the right of the Member States.
- In addition, data must naturally be erased if the processing itself was against the law in the first place
- A data subject should have the right to have personal data concerning him or her rectified

Requirements of GDPR

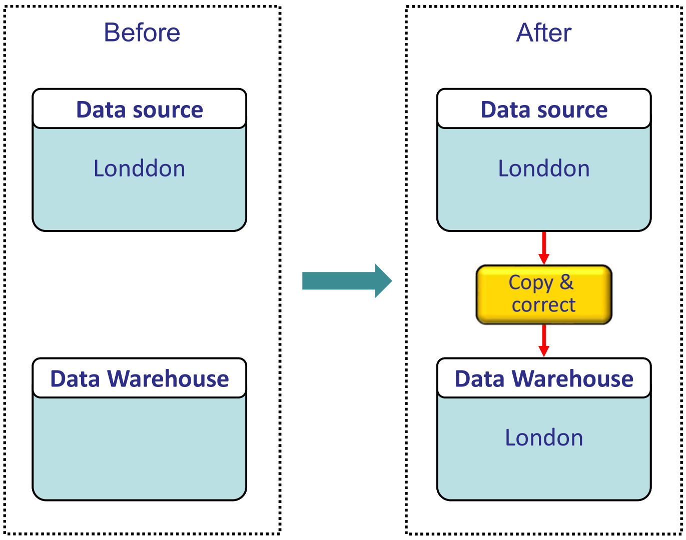


Source: Banking Hub, November 2017;
see https://www.bankinghub.eu/banking/finance-risk/gdpr-deep-dive-implement-right-forgotten

Part 5.4: Dealing with Incorrect Data

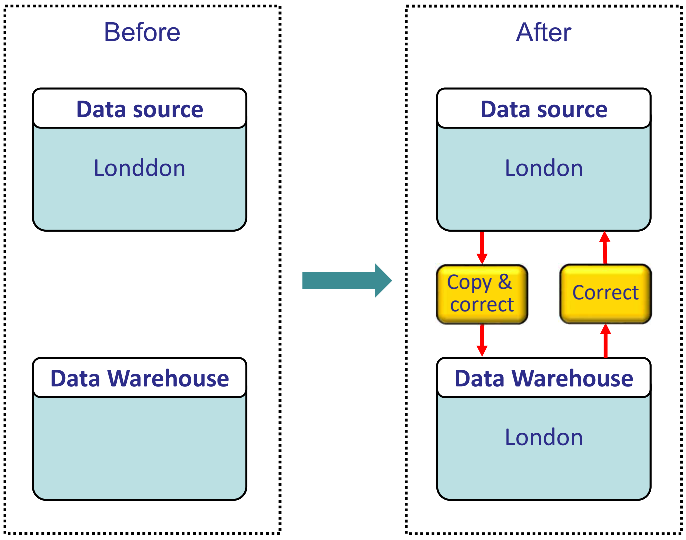


Data Correction Strategies: Simple



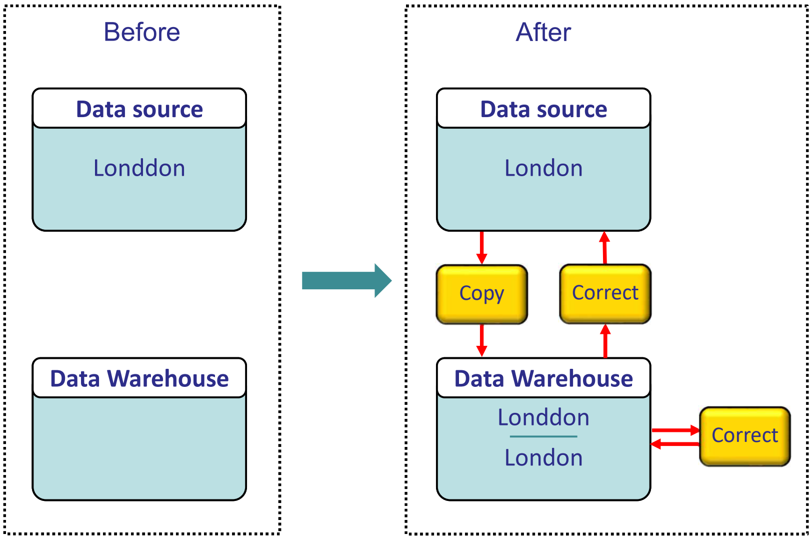
- Restricted to programmable cleansing operations
- Easy to implement
- Source doesn't benefit from cleansing
- Source and data warehouse inconsistent
- No impact on organization
- No time travel supported

Data Correction Strategies: Synchronize



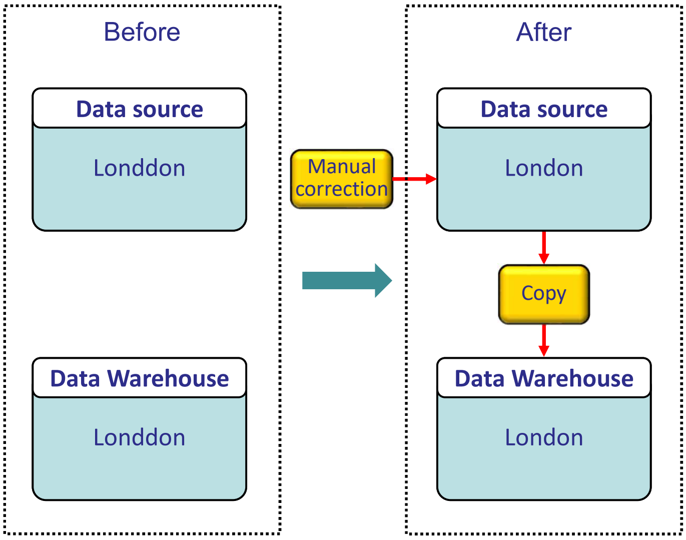
- Manual or automated process?
- Source benefits from cleansing
- Source and data warehouse consistent
- Impact on organization
- No time travel supported

Data Correction Strategies: Time Travel



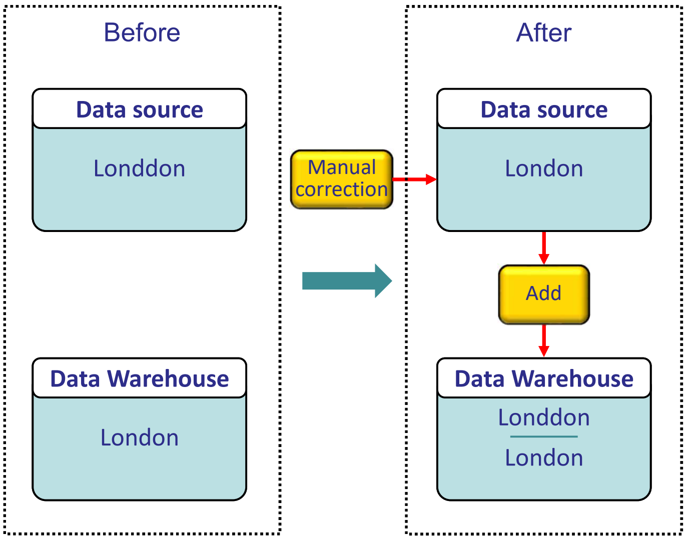
- Restricted to programmable cleansing operations
- Source benefits from cleansing
- Source and data warehouse consistent
- Time travel supported

Data Correction Strategies: Manual Corrections



- User corrections
- Source benefits from cleansing
- Source and data warehouse consistent
- Impact on organization
- No time travel supported

Data Correction Strategies: Manual Corrections + Time Travel



- User corrections
- Source benefits from cleansing
- Source and data warehouse consistent
- Impact on organization
- Time travel supported

Part 6:
Step 6: Select a Reference Data Architecture

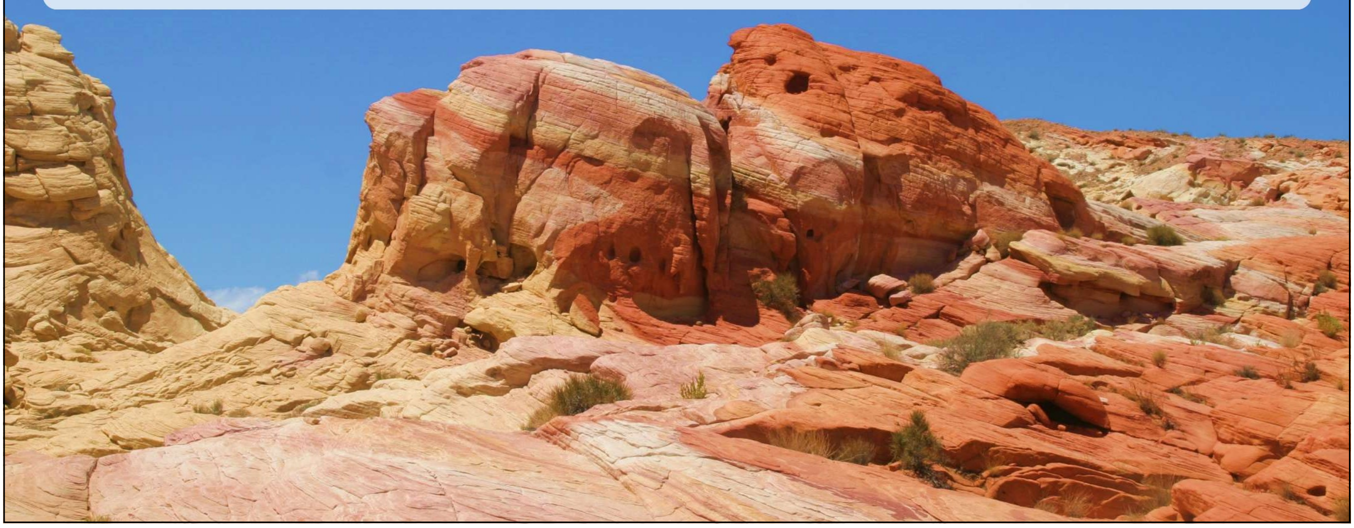
Roadmap for Designing Data Architectures

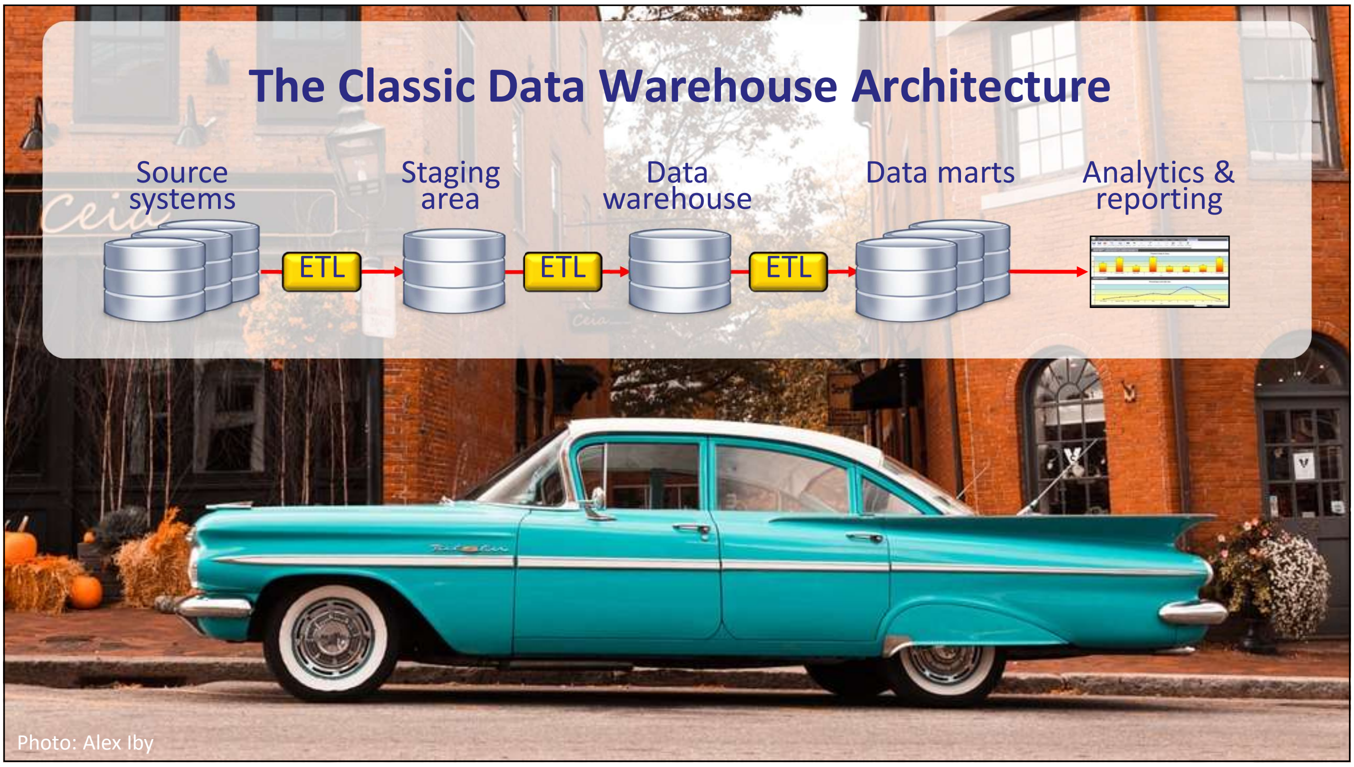
1. Determine business motivations
2. Determine new requirements
3. Analyze the existing environment
4. Study new products and technologies
5. Define architectural design principles
6. Select a reference data architecture
7. Design the new data architecture
8. Determine the Implementation approach
9. Select new products and technologies
10. Introduce the data architecture within the organization

Common Challenges

- Source data must be queryable
- Developers and data consumers can’t find data easily
- Every insert, update, delete and query should be logged for reconstruction purposes and transparency
- Horizontal and operational lineage
- CRUD interface for real-time synchronization
- Centralized, reusable, versioned business logic
- Proper authorization when integrating data

Part 6.1:
The Classic Data Warehouse Architecture





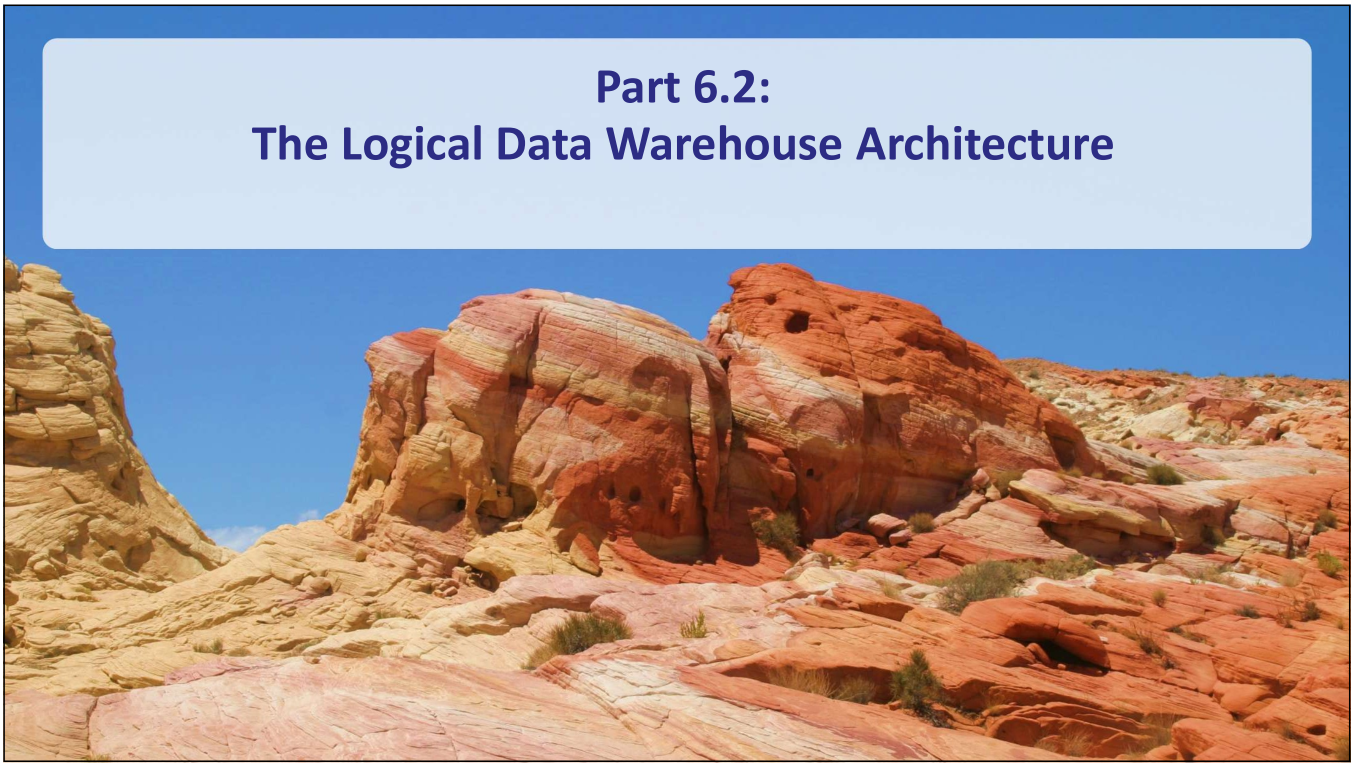
Yesterday: Data Warehouse and Data Consumption

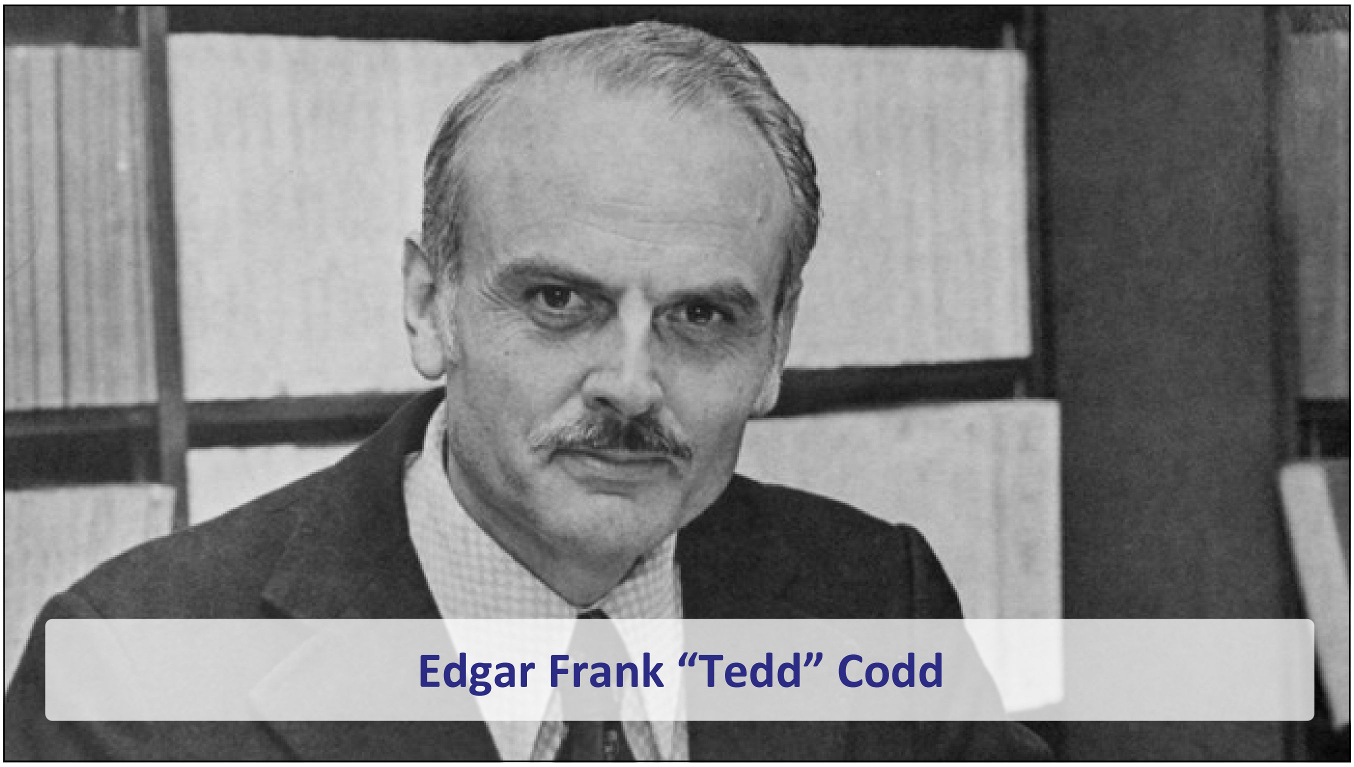
Developers	Development Styles	Report Types	Consumers
IT specialists	Pre-programmed, auditable, governable, formally tested	Batch and online business reports	Business users Legislators

Today & Tomorrow: Data Warehouse and Data Comsumption

Developers	Development Styles	Report Types	Consumers
IT specialists	Pre-programmed, auditable, governable, formally tested	Batch and online business reports	Business users
			Legislators
			External parties
		Customer-facing apps	Consumers
		Streaming analytics	Business users, machines
	Self-service, investigative	Ad-hoc reports	Business users
Business Users	Pre-programmed	Simple data retrieval	Business users
	Self-service, investigative	Ad-hoc reports	Business users
		Data mining, statistics	Data scientists
		Dark data analysis	Business users and IT

The Classic Data Warehouse Architecture is Like a Rigid Assembly Line


Part 6.2:
The Logical Data Warehouse Architecture


Edgar Frank “Tedd” Codd

Ted Codd – June 1970

> Future users of large data banks must be protected from having to know how the data is organized (…) application programs should remain unaffected when the internal representation of data is changed ...

Source: https://cs.uwaterloo.ca/~david/cs848s14/codd-relational.pdf

Ted Codd on Data Independence

The 1981 ACM Turing Award Lecture

Delivered at ACM '81, Los Angeles, California, November 9, 1981

The 1981 ACM Turing Award was presented to Edgar F. Codd, an IBM Fellow of the San Jose Research Laboratory, by President Peter Denning on November 9, 1981 at the ACM Annual Conference in Los Angeles, California. It is the Association's foremost award for technical contributions to the computing community.

Codd was selected by the ACM General Technical Achievement Award Committee for his "fundamental and continuing contributions to the theory and practice of database management systems." The originator of the relational model for databases, Codd has made further important contributions in the development of relational algebra, relational calculus, and normalization of relations.

Edgar F. Codd joined IBM in 1949 to prepare programs for the Selective Sequence Electronic Calculator. Since then, his work in computing has encompassed logical design of computers (IBM 701 and Stretch), managing a computer center in Canada, heading the development of one of the first operating systems with a general multiprogramming capability, contributing to the logic of self-reproducing automata, developing high level techniques for software specification, creating and extending the relational approach to database management, and developing an English analyzing and synthesizing subsystem for casual users of relational databases. He is also the author of *Cellular Automata*, an early volume in the ACM Monograph Series.

Codd received his B.A. and M.A. in Mathematics from Oxford University in England, and his M.Sc. and Ph.D. in Computer and Communication Sciences from the University of Michigan. He is a Member of the National Academy of Engineering (USA) and a Fellow of the British Computer Society.

The ACM Turing Award is presented each year in commemoration of A. M. Turing, the English mathematician

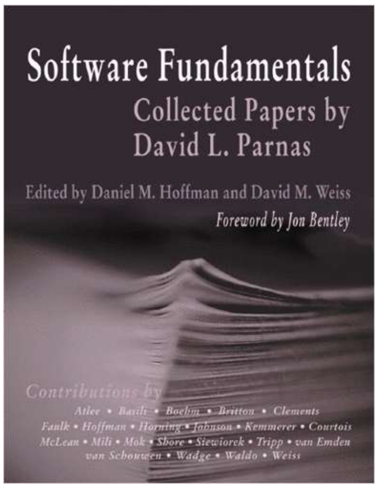
2. Motivation

The most important motivation for the research work that resulted in the relational model was the objective of providing a sharp and clear boundary between the logical and physical aspects of database management (including database design, data retrieval, and data manipulation). We call this the *data independence objective.*

A second objective was to make the model structurally simple, so that all kinds of users and programmers could have a common understanding of the data, and could therefore communicate with one another about the database. We call this the *communicability objective.*

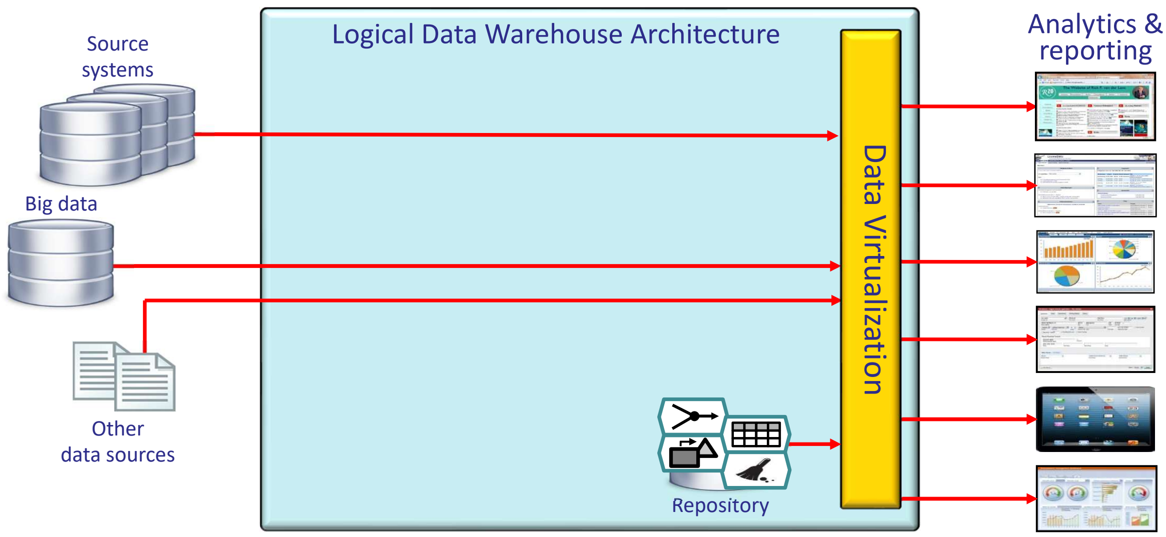
A third objective was to introduce high level language concepts (but not specific syntax) to enable users to express operations upon large chunks of information at a time. This entailed providing a foundation for set-oriented processing (i.e., the ability to express in a single statement the processing of multiple sets of records at a time). We call this the *set-processing objective.*

David Parnas - Information Hiding - 1972

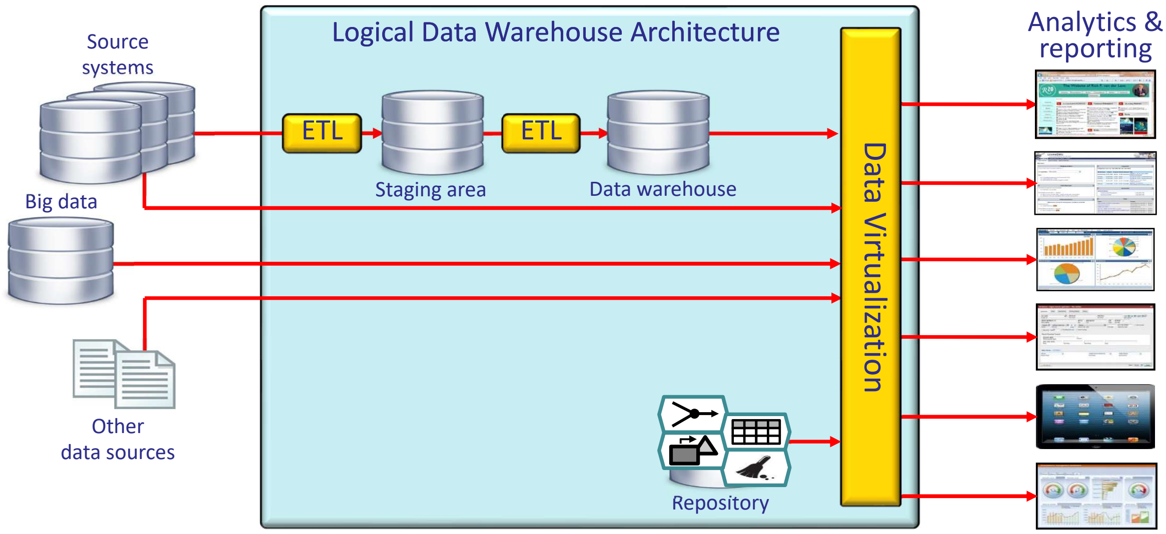


http://www.cs.umd.edu/class/spring2003/cmsc838p/Design/criteria.pdf

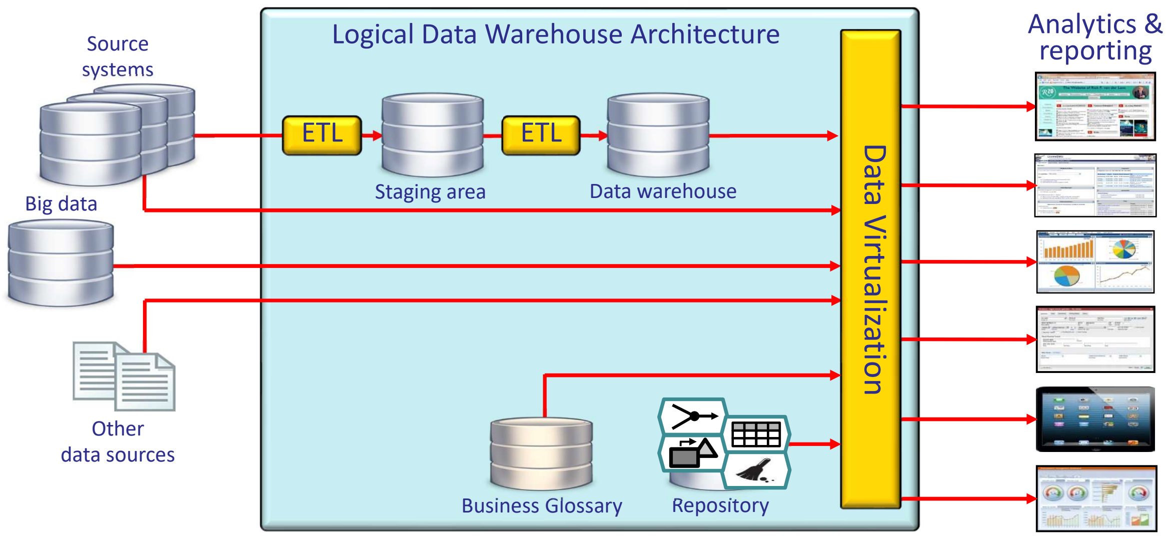
The Logical Data Warehouse Architecture (1)



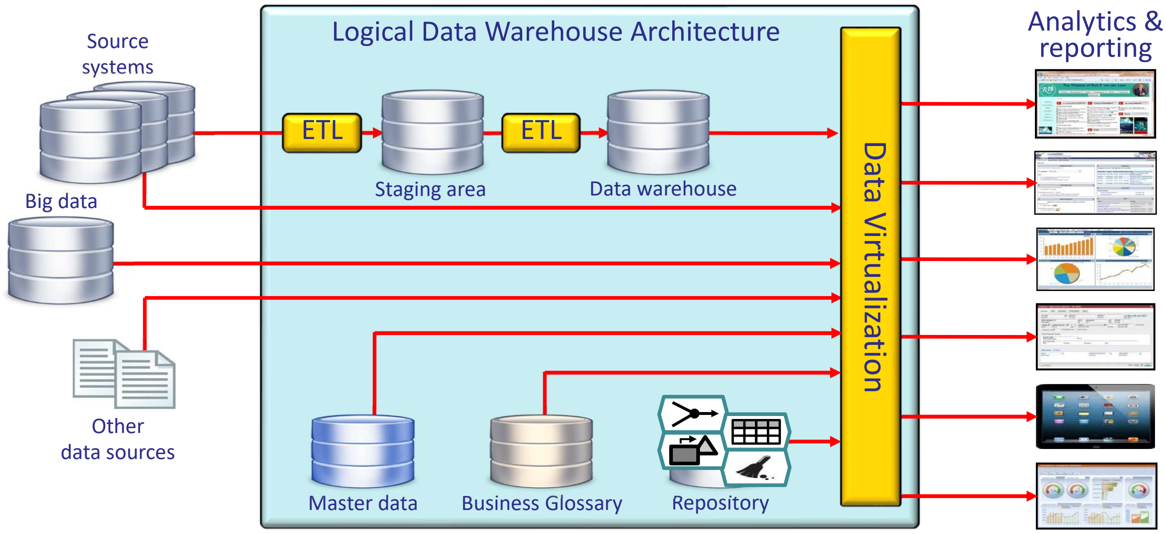
The Logical Data Warehouse Architecture (2)



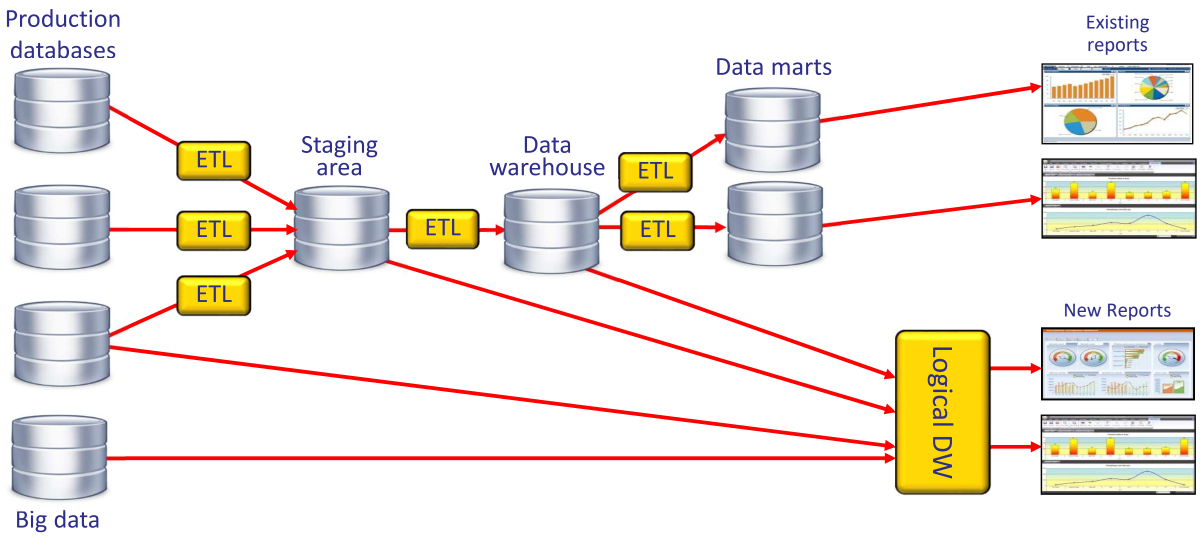
The Logical Data Warehouse Architecture (3)



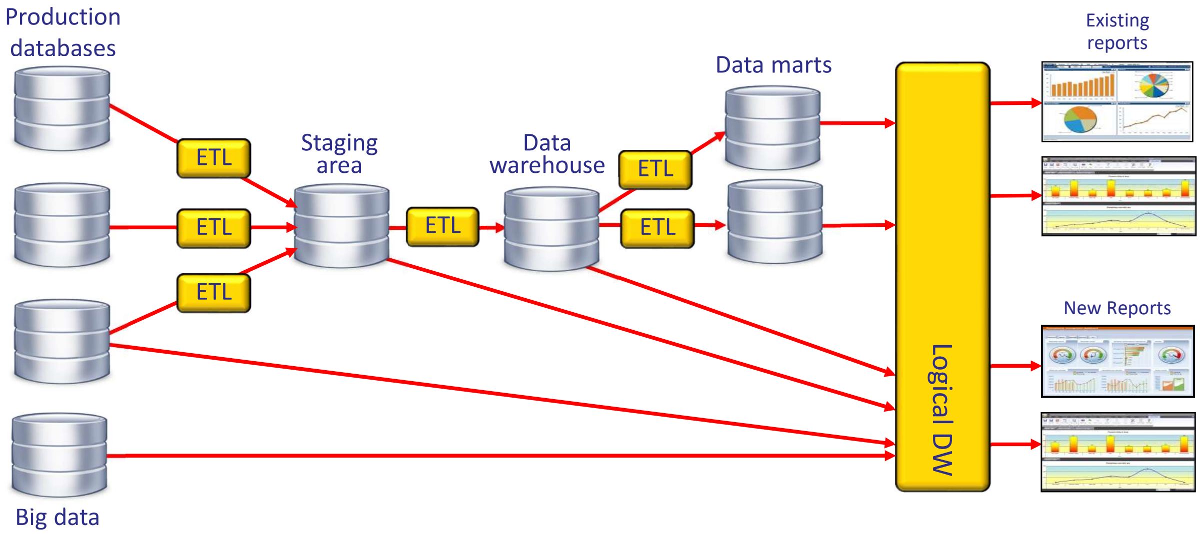
The Logical Data Warehouse Architecture (4)



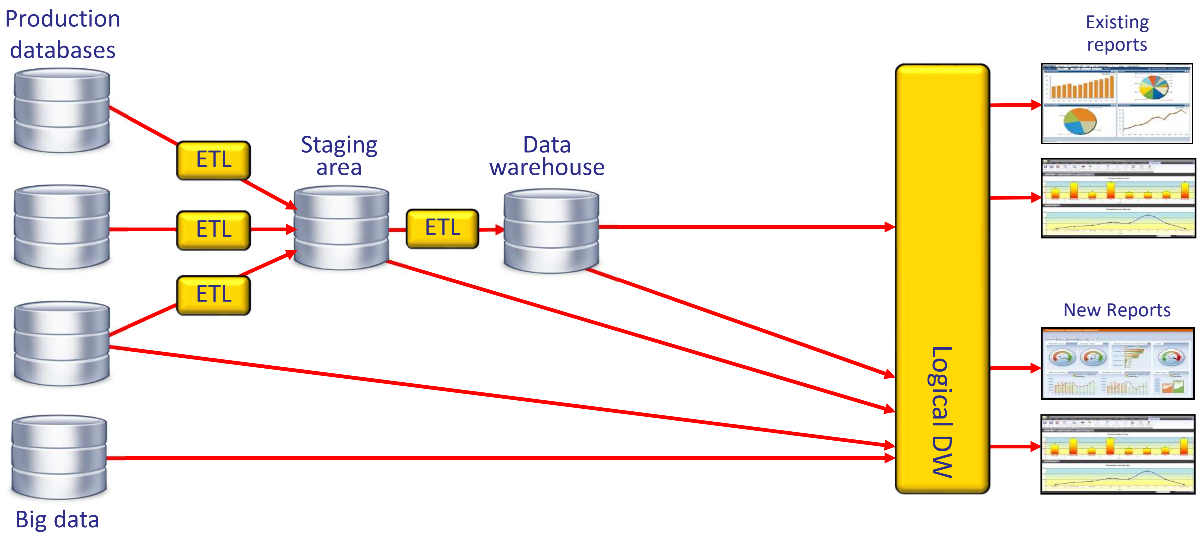
Wrap the Old Data Warehouse (1)

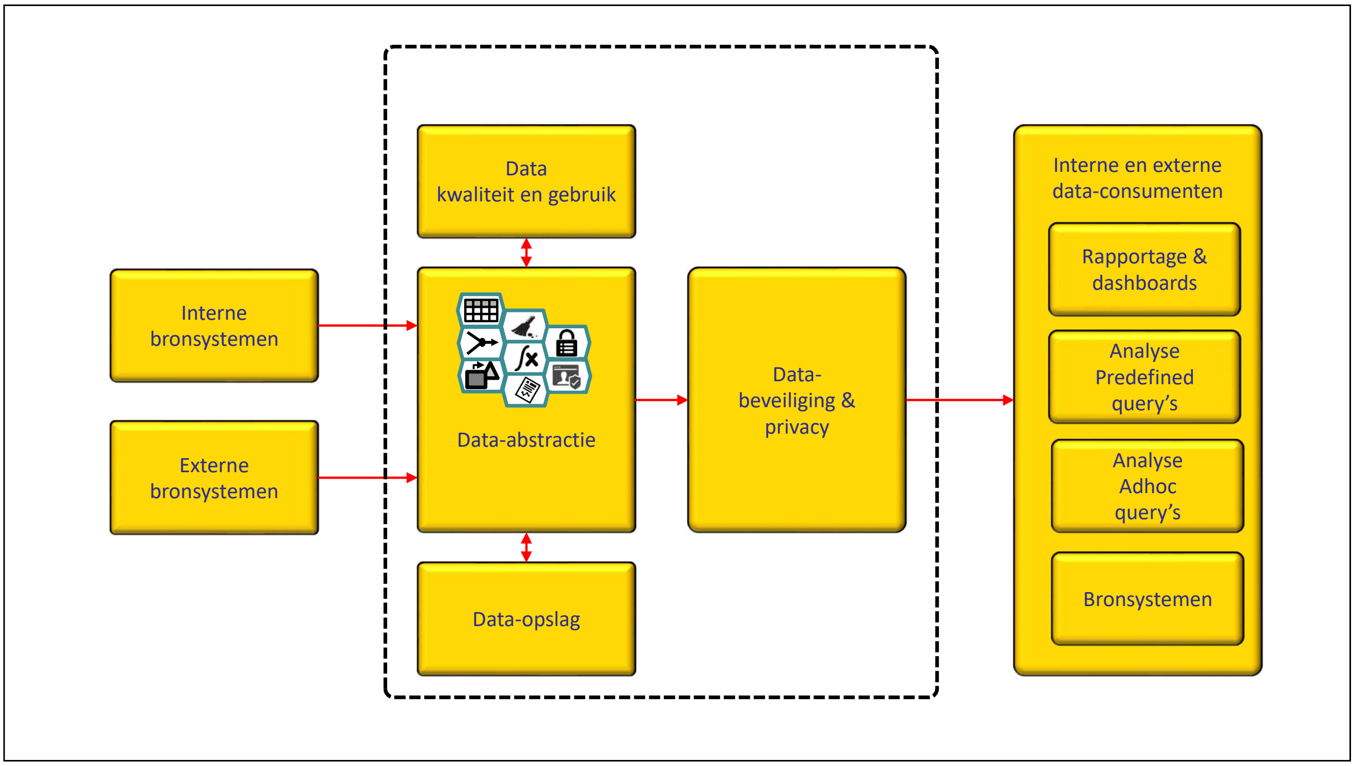


Wrap the Old Data Warehouse (2)

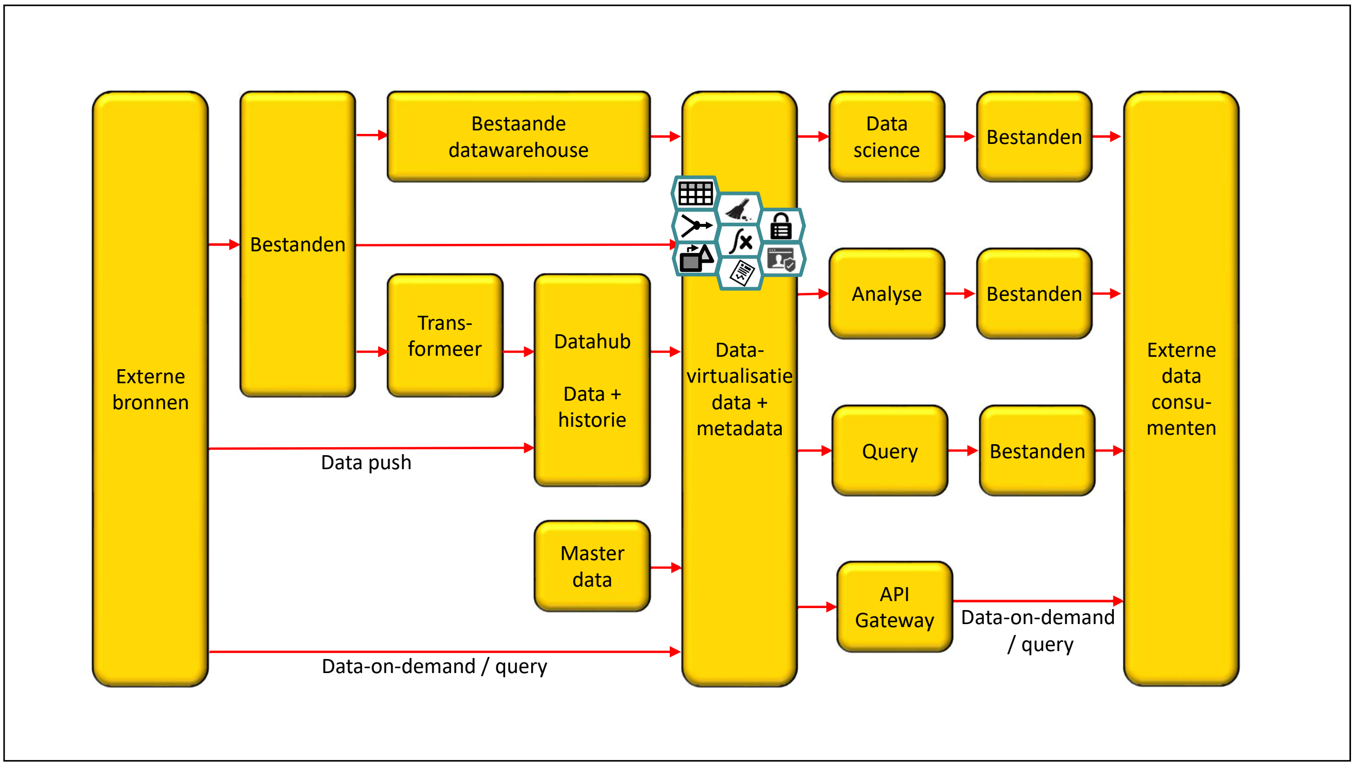


Wrap the Old Data Warehouse (3)

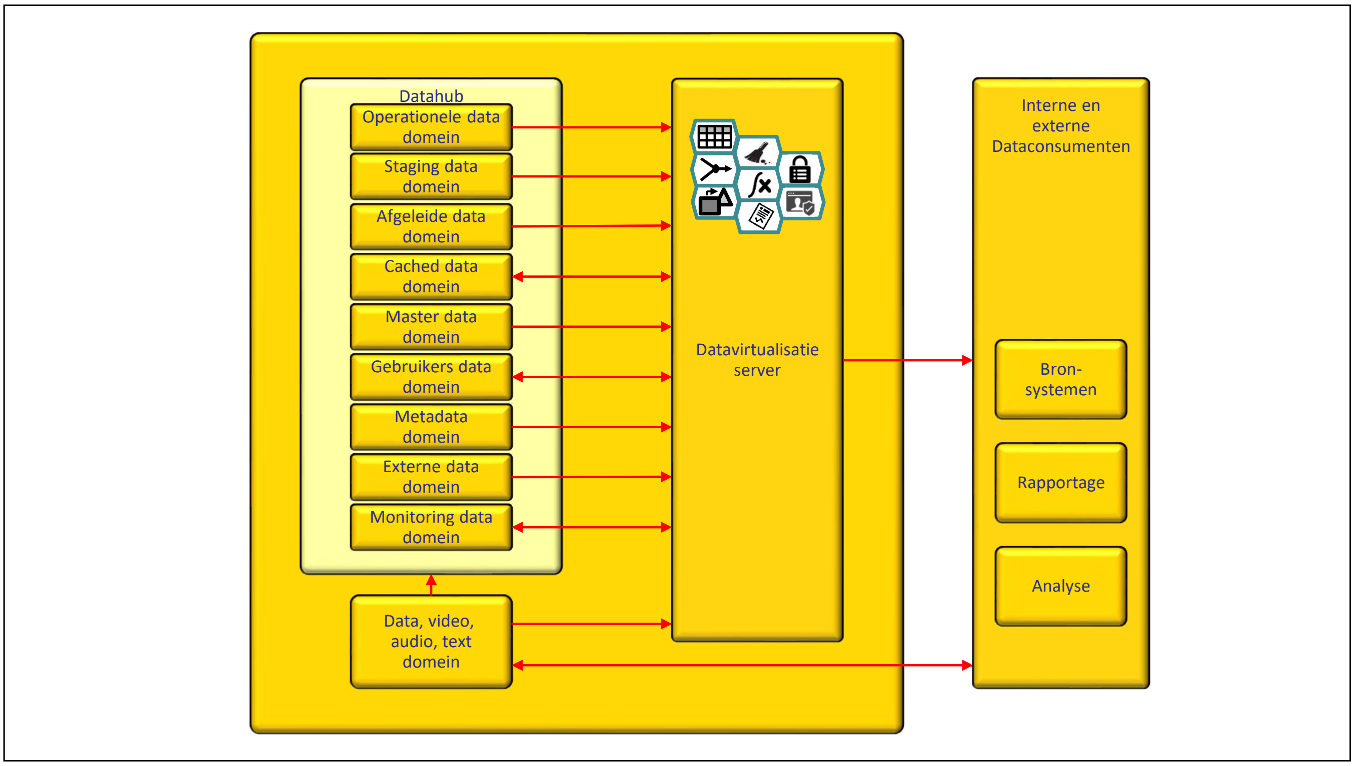


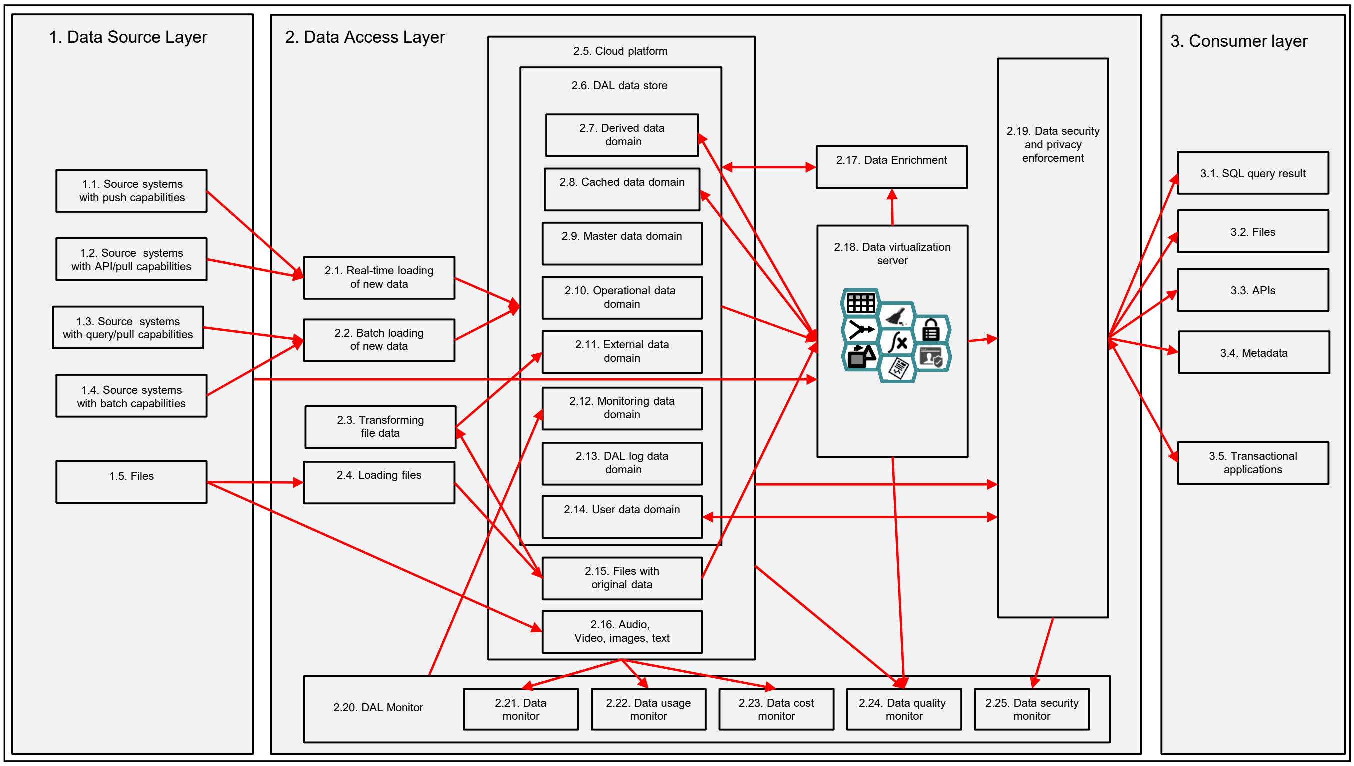


Data
kwaliteit en gebruik
Interne
bronsystemen
Externe
bronsystemen
Data-abstractie
Data-
beveiliging &
privacy
Data-opslag
Interne en externe
data-consumenten
Rapportage &
dashboards
Analyse
Predefined
query's
Analyse
Adhoc
query's
Bronsystemen



Externe
bronnen
Bestanden
Bestaande
datawarehouse
Trans-
formeer
Datahub
Data +
historie
Data push
Master
data
Data-on-demand / query
Data-
virtualisatie
data +
metadata
Data
science
Analyse
Query
API
Gateway
Bestanden
Bestanden
Bestanden
Data-on-demand
/ query
Externe
data
consu-
menten


Datahub
Operationele data domein
Staging data domein
Afgeleide data domein
Cached data domein
Master data domein
Gebruikers data domein
Metadata domein
Externe data domein
Monitoring data domein
Data, video, audio, text domein
Datavirtualisatie server
Interne en externe Dataconsumenten
Bron-systemen
Rapportage
Analyse

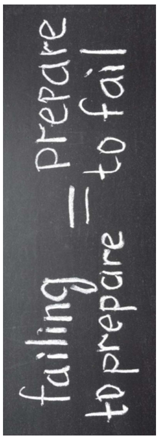

1. Data Source Layer
1.1. Source systems with push capabilities
1.2. Source systems with API/pull capabilities
1.3. Source systems with query/pull capabilities
1.4. Source systems with batch capabilities
1.5. Files
2. Data Access Layer
2.1. Real-time loading of new data
2.2. Batch loading of new data
2.3. Transforming file data
2.4. Loading files
2.5. Cloud platform
2.6. DAL data store
2.7. Derived data domain
2.8. Cached data domain
2.9. Master data domain
2.10. Operational data domain
2.11. External data domain
2.12. Monitoring data domain
2.13. DAL log data domain
2.14. User data domain
2.15. Files with original data
2.16. Audio, Video, images, text
2.17. Data Enrichment
2.18. Data virtualization server
2.19. Data security and privacy enforcement
2.20. DAL Monitor
2.21. Data monitor
2.22. Data usage monitor
2.23. Data cost monitor
2.24. Data quality monitor
2.25. Data security monitor
3. Consumer layer
3.1. SQL query result
3.2. Files
3.3. APIs
3.4. Metadata
3.5. Transactional applications

Part 6.3:
The Data Lake

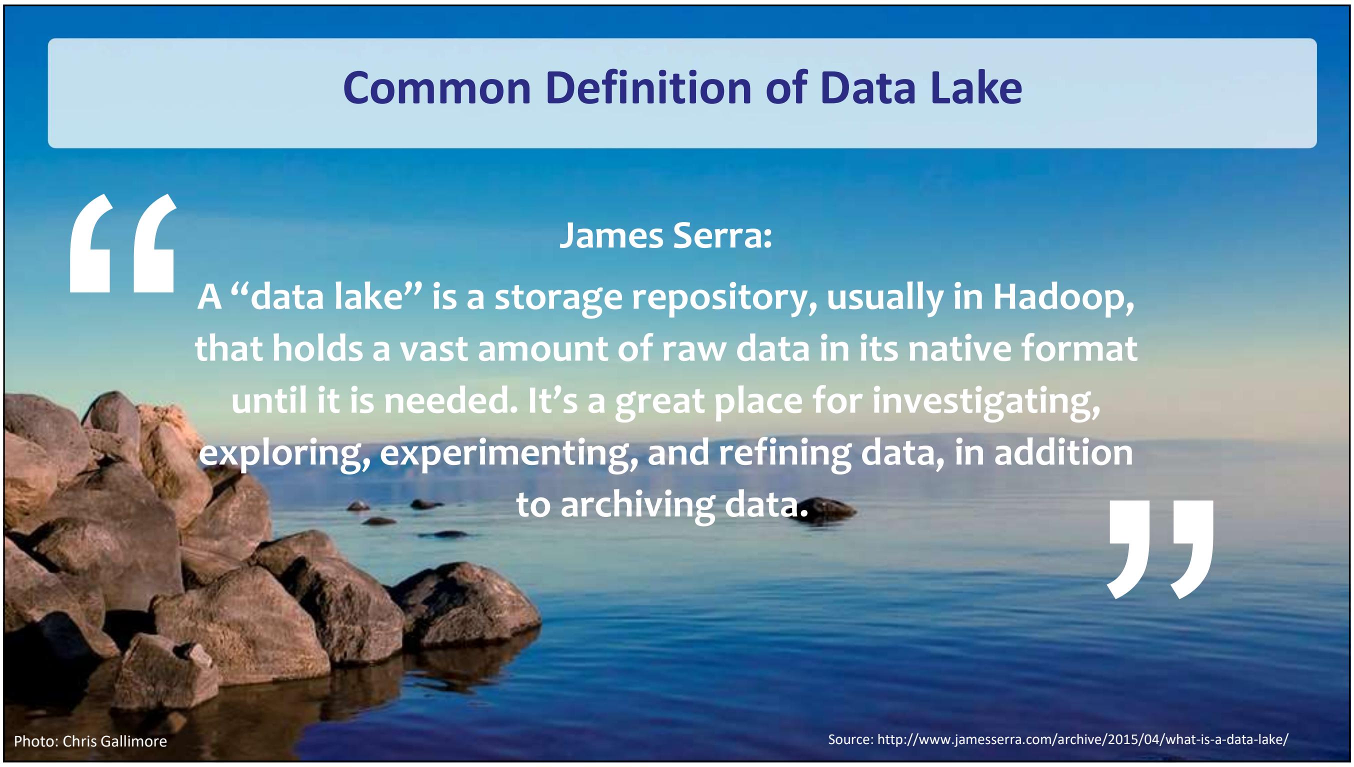
Data Science Steps

- Defining goals
- Data selection
- Data understanding
- Data enrichment
- Data cleansing
- Data coding
- Creating analytical model
- Analytics
- Understanding results

Data Coding



- Computation
  - examples: divide all monthly salaries by 1000; round all prices
- Grouping continuous values
  - example: all transaction between 08:00 and 10:30 will belong to group 1, all transactions between 10:31 and 12:00 will belong to group 2
  - do groups need equal sizes (with respect to ranges)?
  - do groups need equal numbers of values?
- Scaling
  - most neural networks accept numeric data only in the range 0.0 to 1.0 or -1.0 to 1.0; used for continuous values, such as salary and weight
- Normalizing
  - sum all elements, and divide each element by the sum
  - value represents the percentage of contribution
- Symbolic to numeric transformations
  - example: the string "yes" becomes 1, and "no" becomes 0
- Coding discrete values
  - transform a column with fixed set of values (F) into F columns with yes/no values



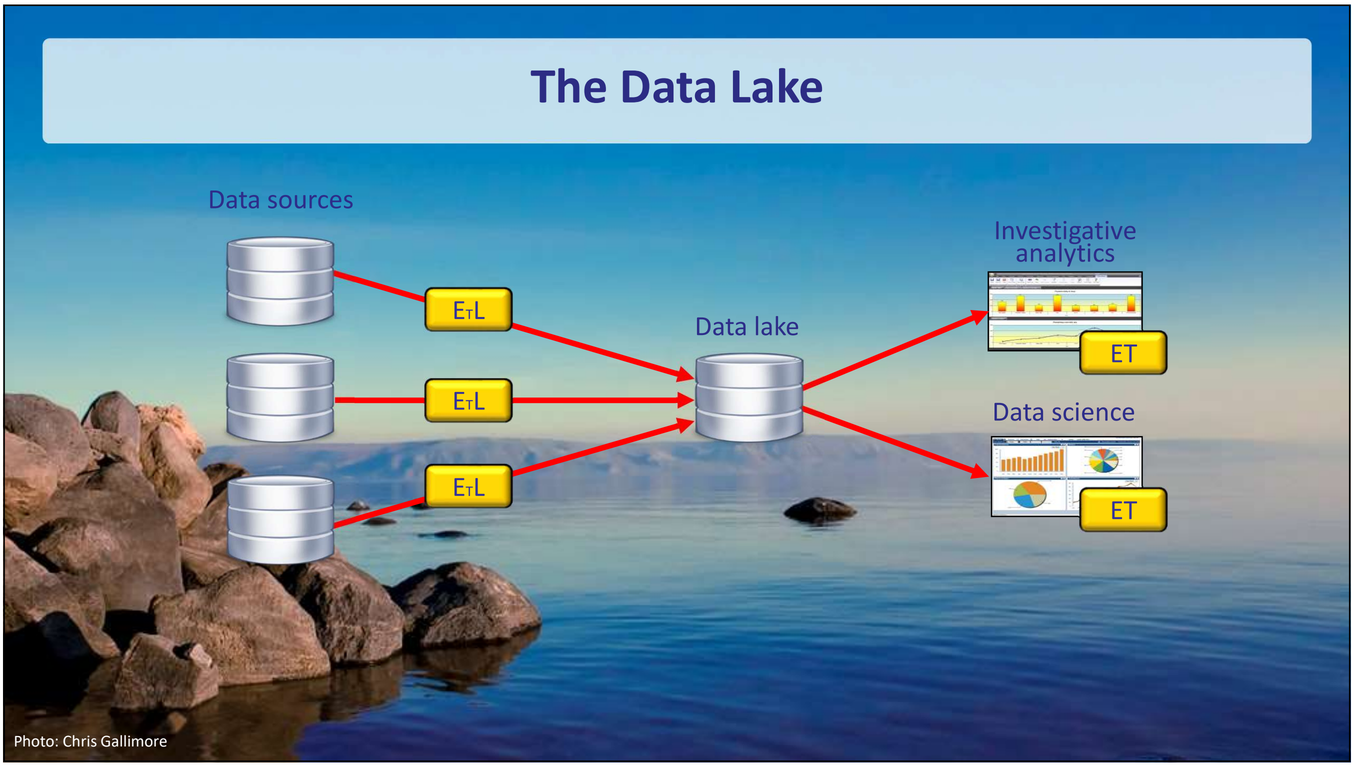
Common Definition of Data Lake

James Serra:

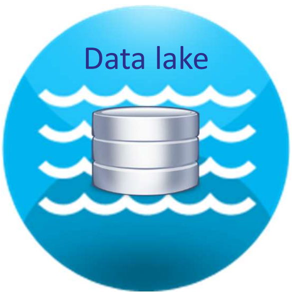
A “data lake” is a storage repository, usually in Hadoop, that holds a vast amount of raw data in its native format until it is needed. It’s a great place for investigating, exploring, experimenting, and refining data, in addition to archiving data.

Photo: Chris Gallimore

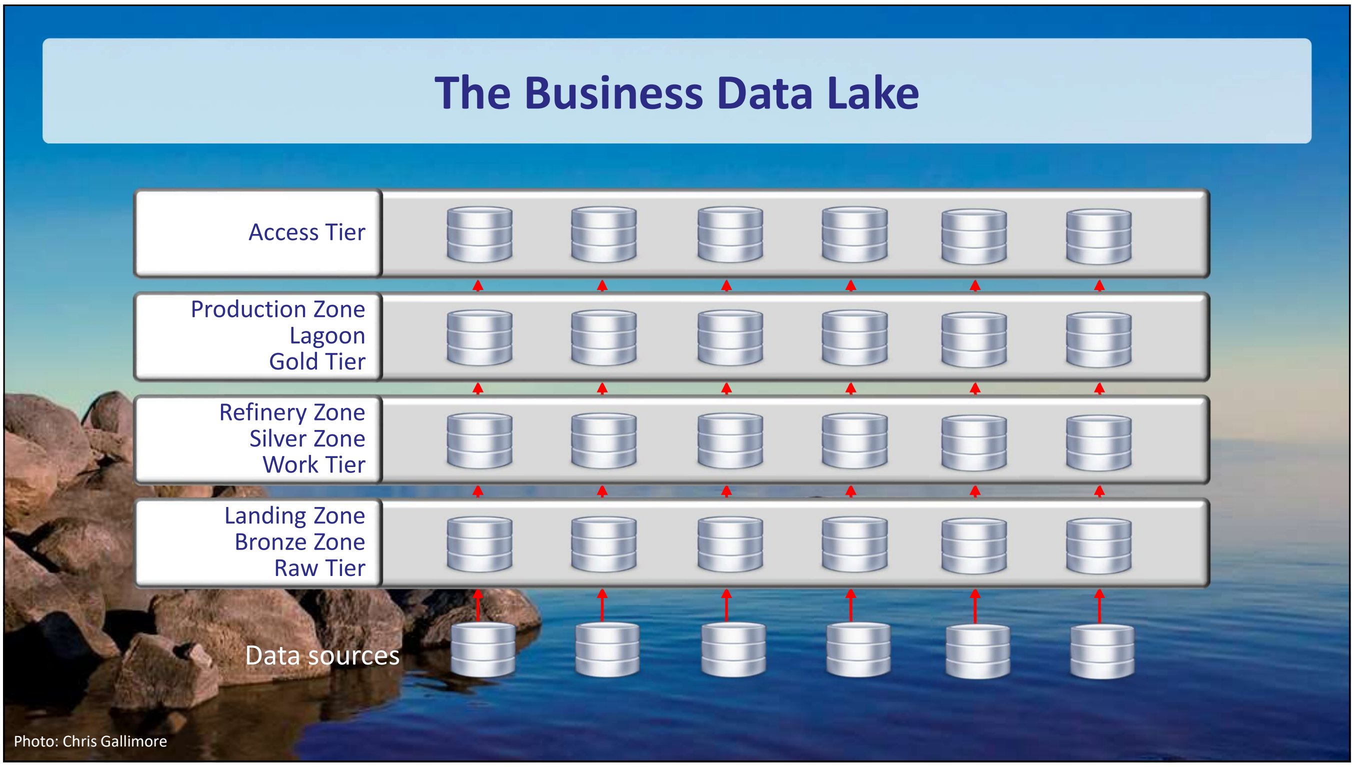
Source: http://www.jamesserra.com/archive/2015/04/what-is-a-data-lake/



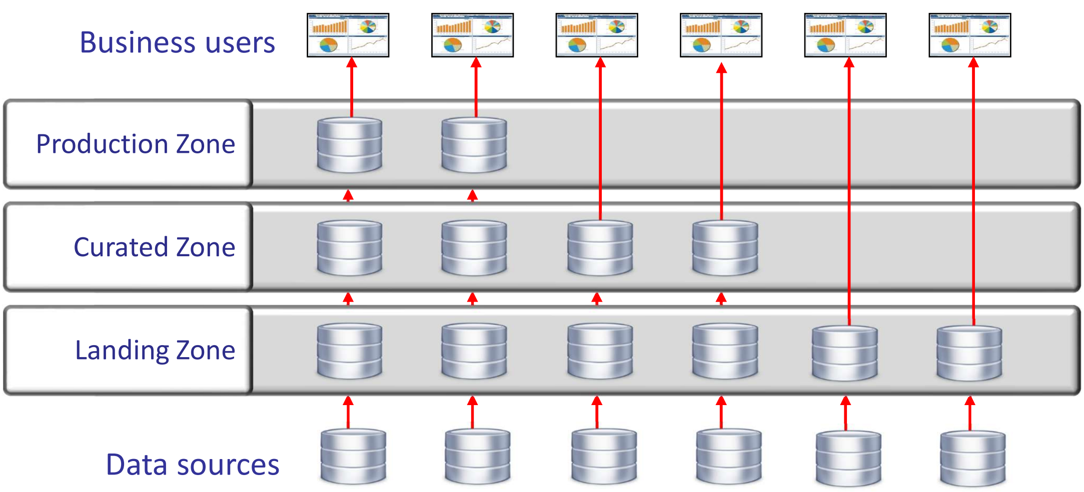
Challenges of a Physical Data Lake



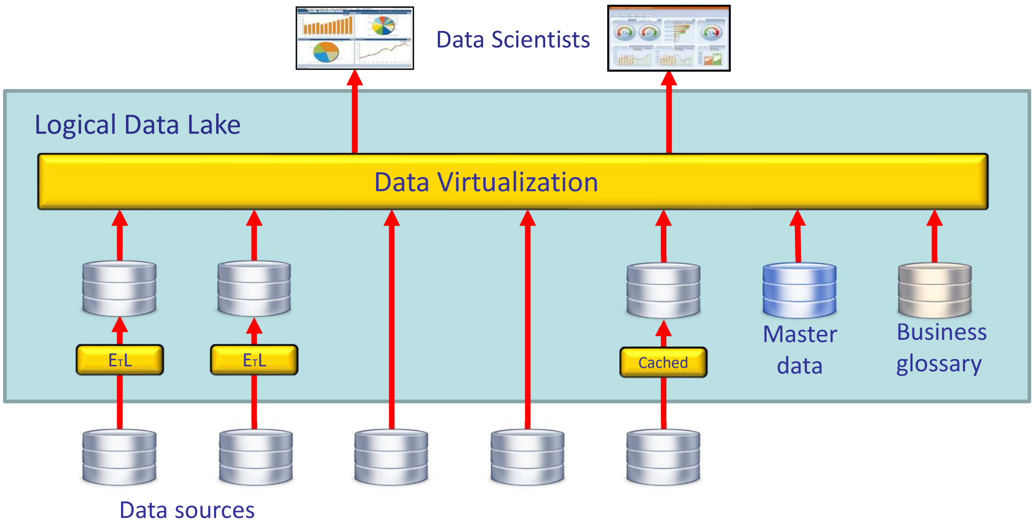
- Big data too big to move
  - Too slow to copy and bandwidth issues
- Complex “T” moved to data consumption
- Company politics
- Data privacy and protection regulations
- Data in data lake is stored outside original security realm
- Metadata to describe data
- Some sources are hard to copy
  - For example, mainframe data
- Refreshing of data lake
- Management of data lake required
- …



A Data Lake With Multiple Zones

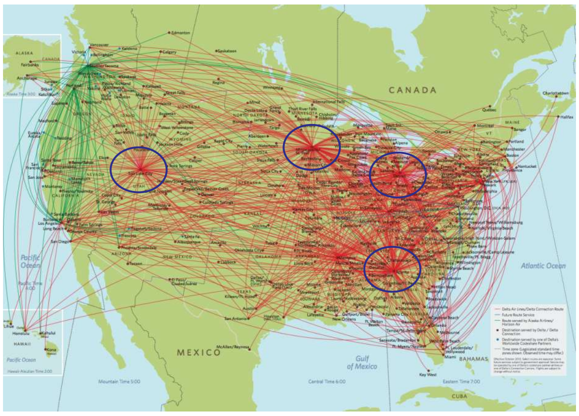


The Logical (Virtual) Data Lake



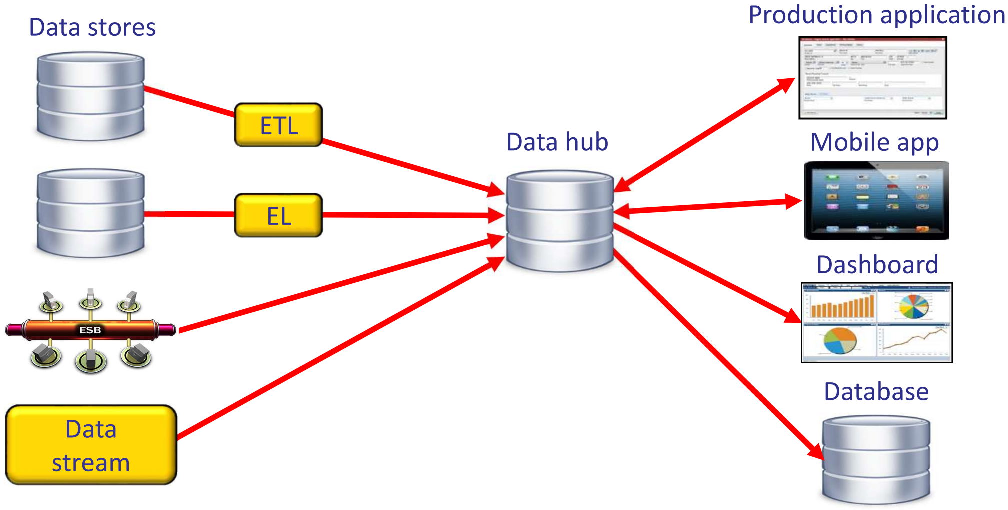
Part 6.4:
The Data Hub

What is a Data Hub?

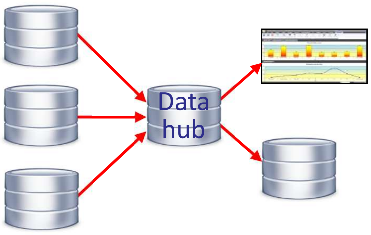


- Wikepedia: A data hub is a collection of data from multiple sources organized for distribution, sharing, and often subsetting and sharing. Generally this data distribution is in the form of a hub and spoke architecture

The Data Hub (Sharing of Data)

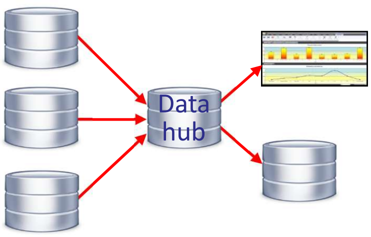


Characteristics of the Data Hub



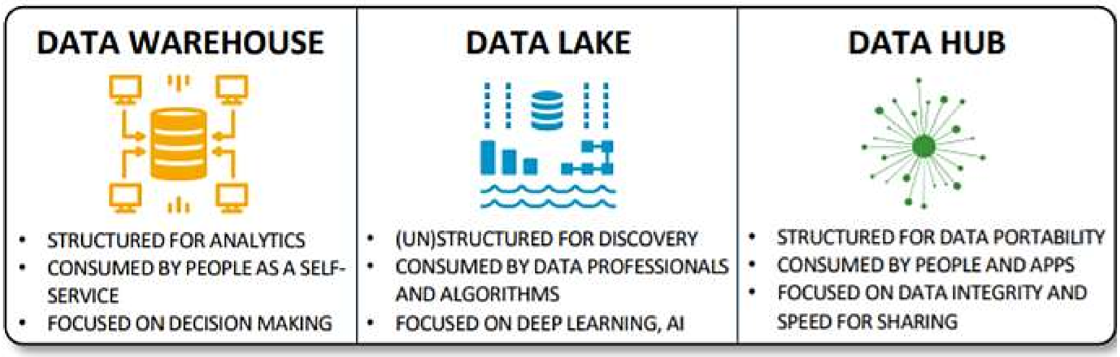
- The main goal of a data hub is to organize data efficiently, storing it in a cost-efficient manner and expose it towards key business functions
- It excels in easy integration, and enables de-duplication, security, quality and data standardization
- The data hub can be leveraged to enable data processing activities with the end use-case in mind, and typically has governance capabilities
- Although operationally focused, it can be trusted as an analytical data source

Data Hub Versus the Rest Of the World



- Data hub versus *data warehouse*: a data hub is generally non-integrated and often at different grains
- Data hub versus *operational data store*: a data hub does not need to be limited to operational data
- Data hub versus *data lake*: a data lake tends to store data in one place for availability, and allow/require the consumer to process or add value to the data
- Data warehouses and data lakes may be endpoints, data hubs are not endpoints, they serve as points of intermediation and data exchange

Comparison of Three Data Storage Environments



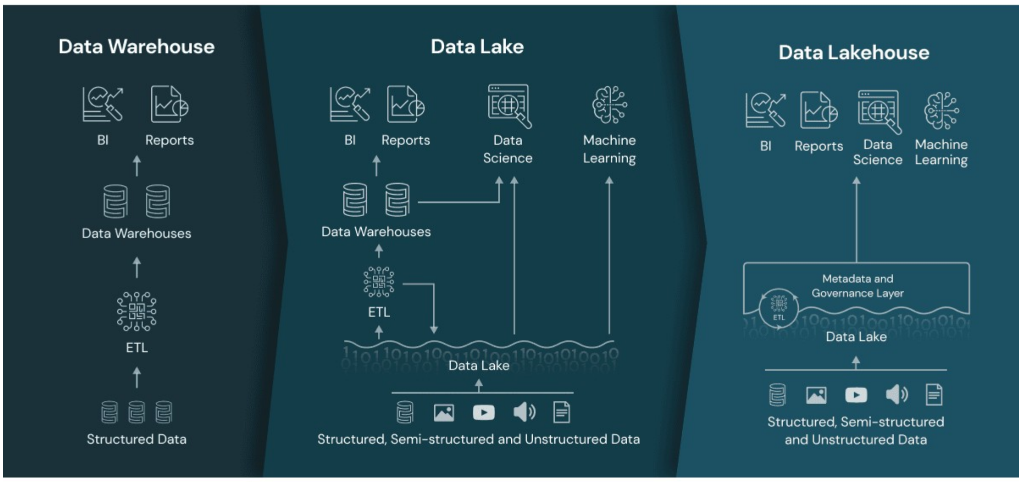
Source: Talend; see https://www.talend.com/resources/customer-360-data-hub/

Part 6.5: The Data Lakehouse

Definitions of Data Lakehouse

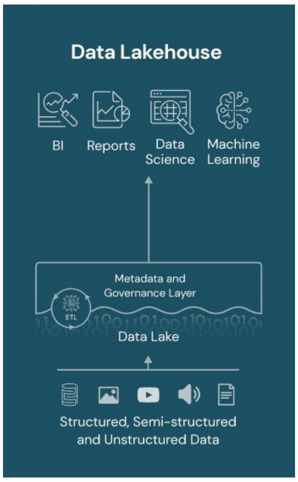
- DataBricks: "A data lakehouse is a […] open data management architecture that combines the flexibility, cost-efficiency, and scale of data lakes with the data management and ACID transactions of data warehouses, enabling business intelligence (BI) and machine learning (ML) on all data."
- Striim, John Kutay: "A data lakehouse is a new, big-data storage architecture that combines the best features of both data warehouses and data lakes. A data lakehouse enables a single repository for all your data (structured, semi-structured, and unstructured) while enabling best-in-class machine learning, business intelligence, and streaming capabilities."
- Dremio, Deepa Sankar: "A [data] lakehouse has the performance and optimization of a data warehouse combined with the flexibility of a data lake."
- Wikipedia: "Databricks develops and sells a cloud data platform using the marketing term lakehouse, a portmanteau based on the terms data warehouse and data lake."

A Comparison of Three Data Architectures



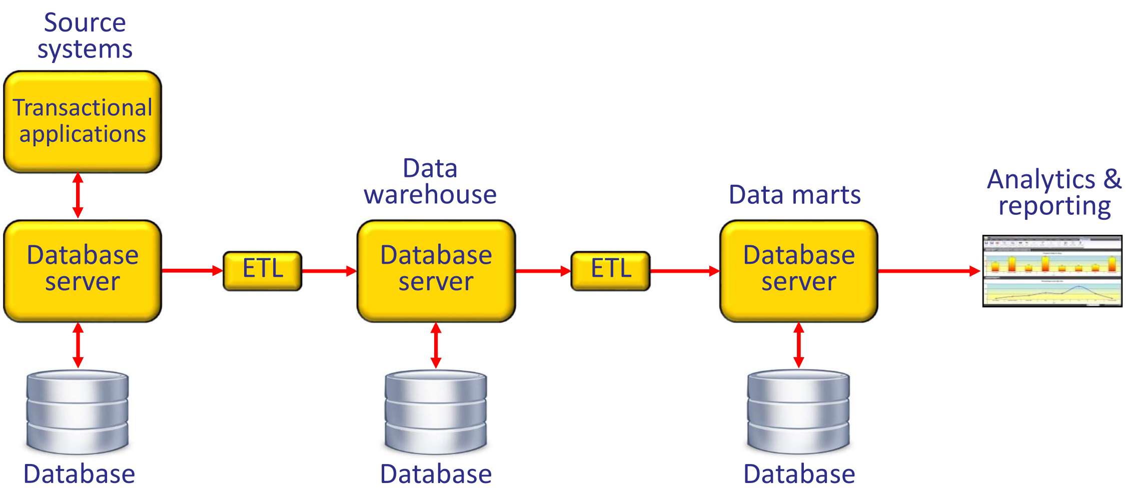
Source: Databricks.com

Key Characteristics of a Data Lakehouse

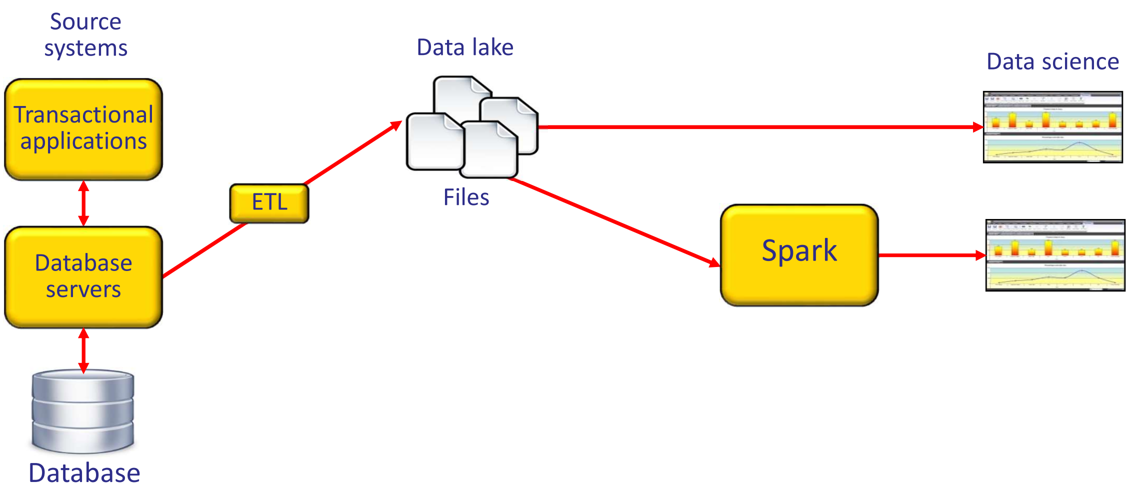


- Two use cases: BI and data science
- Data is stored once
- Supports structured and unstructured data
- Schema enforcement
- Open file formats
- Low-cost data storage
- ACID compliant

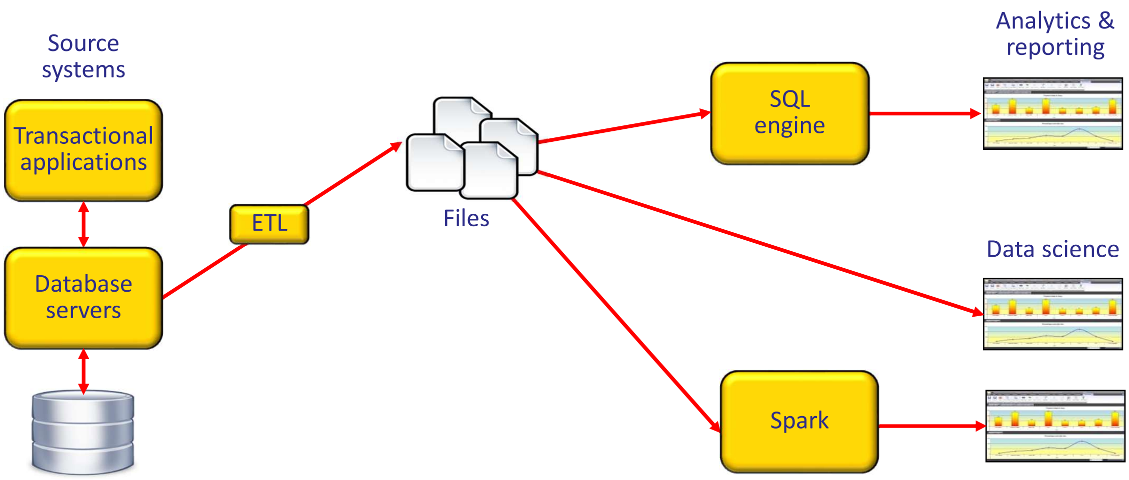
The Data Warehouse Architecture in More Detail



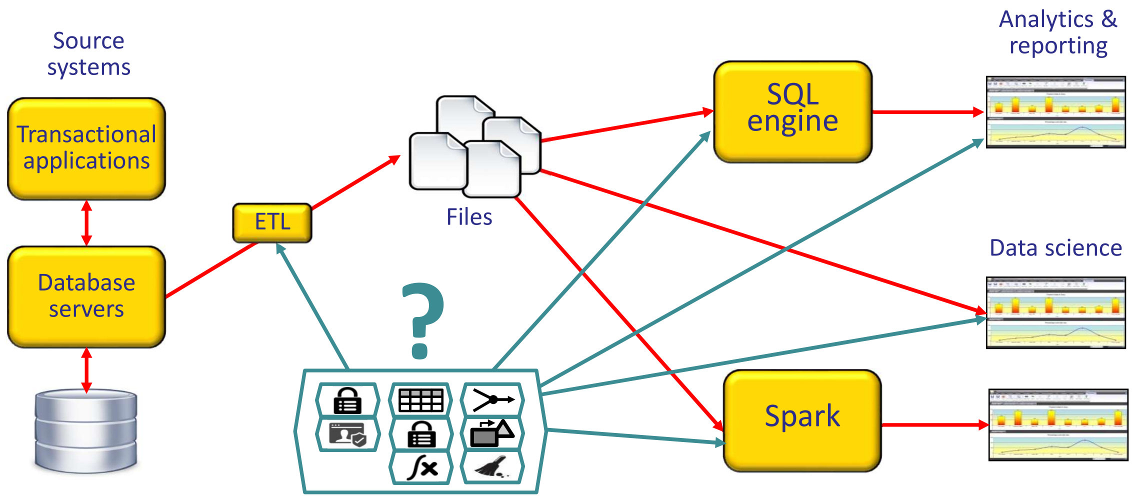
The Data Lake Architecture in More Detail



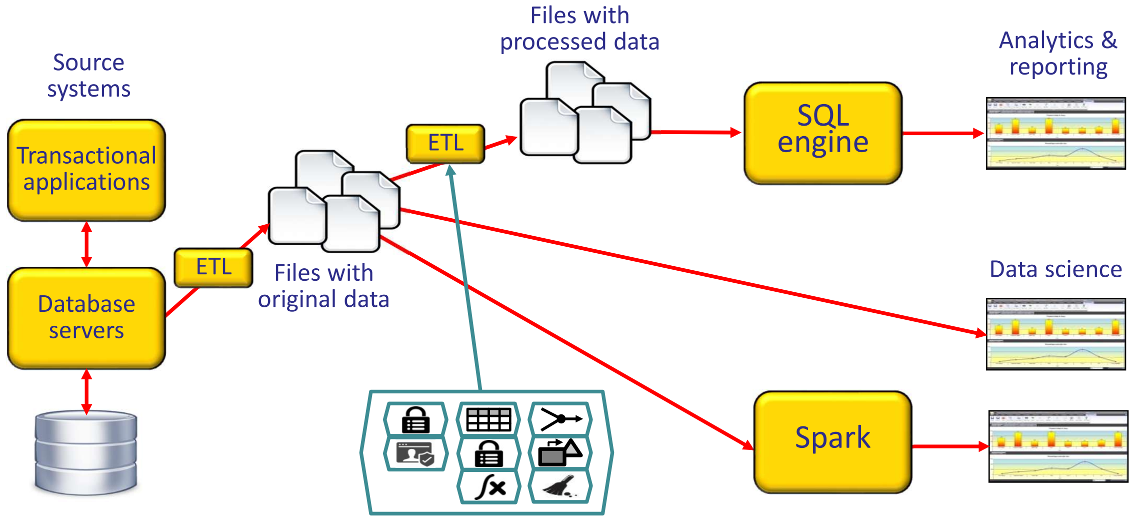
The Data Lakehouse Architecture in More Detail



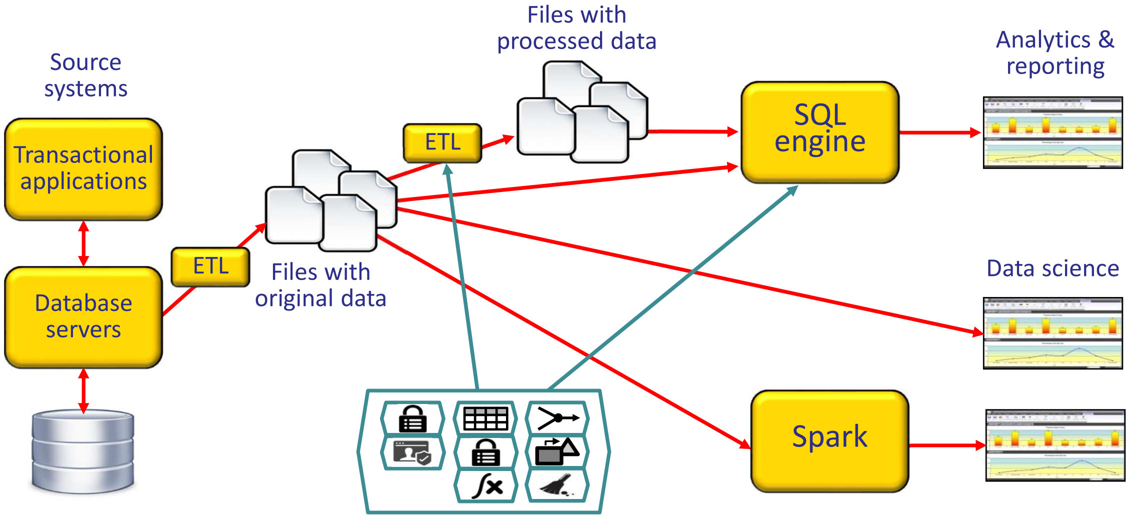
Where to Implement Data Processing Specifications?



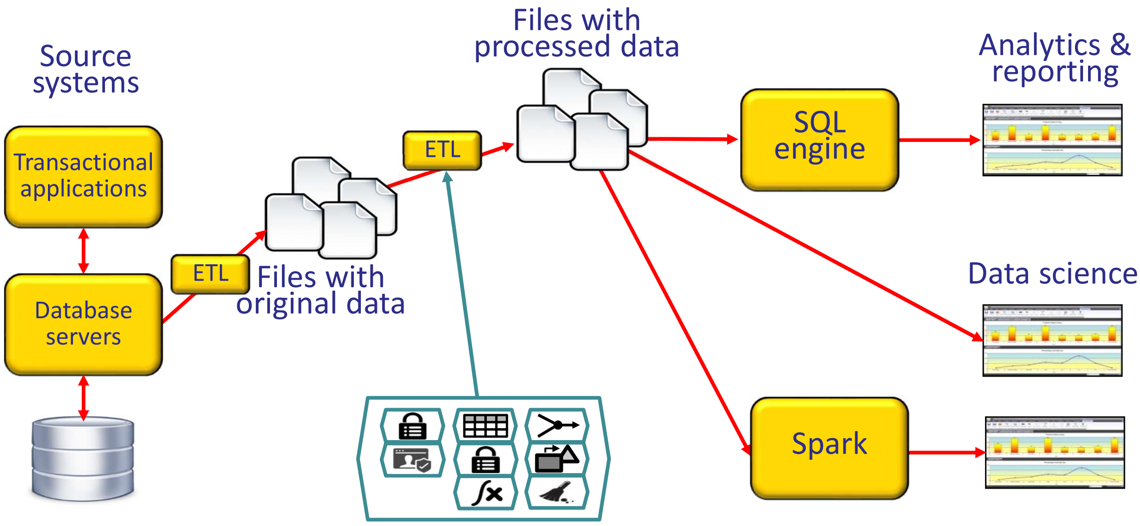
Solution 1: *All* BI via Processed Data



Solution 2: *Some* BI via Processed Data



Solution 3: Both Use Cases via Processed Data



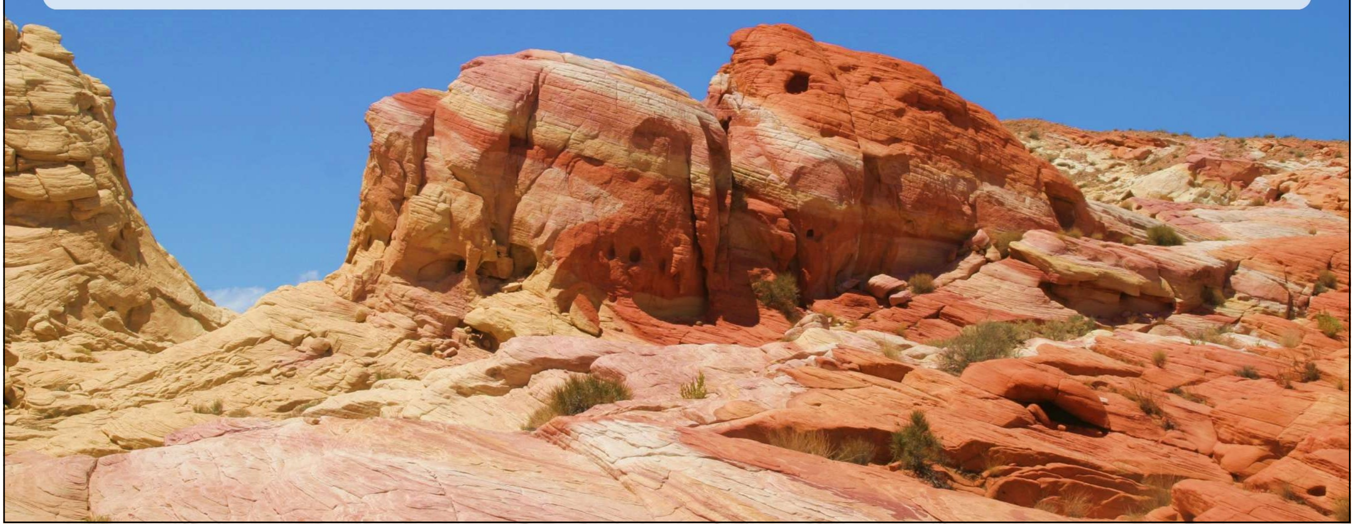
High-Level Comparison of Three Architectures

Characteristic	Data Warehouse	Data lake	Data Lakehouse
Type of data	Structured	Structured and unstructured	Structured and unstructured
Use cases	BI, reporting, dashboarding	Experimental, investigative	Both
Schema enforcement	Yes	Optional	Optional
Open file format	No	Yes	Yes
Low-cost data storage	No	Yes	Yes
ACID-compliant	Yes	No	Yes
Near real-time data	No	Yes	Yes
Non-siloed	No	No	Yes
Data copies minimal	No	No	Yes
Anonymization	Yes	Depends	Depends
Auditable	Yes	Depends	Depends
Performance optimized for BI	Yes	No	?
Performance optimized for data science	No	Yes	Yes

Based on assumptions

Source: Various articles in the Internet

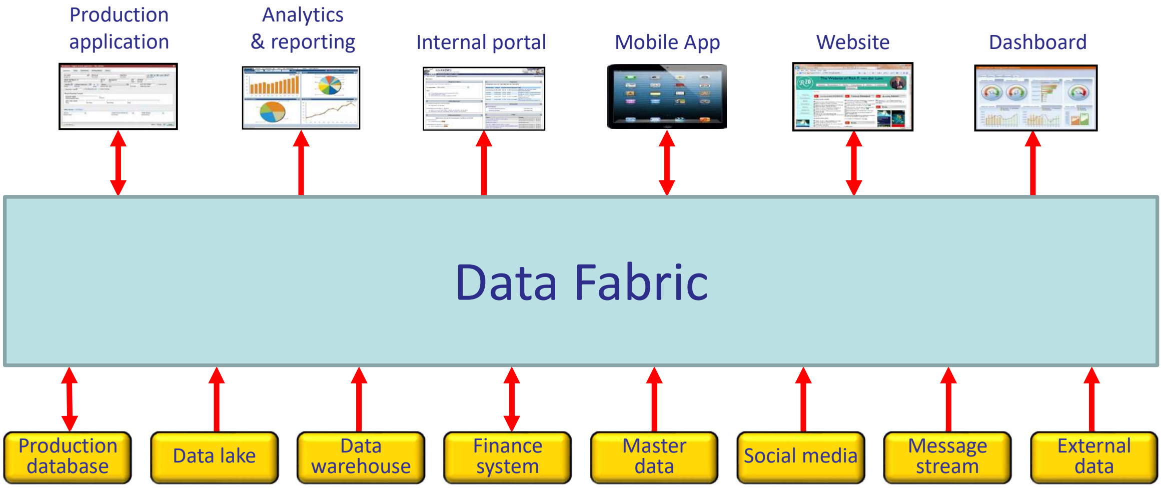
Part 6.6:
The Data Fabric



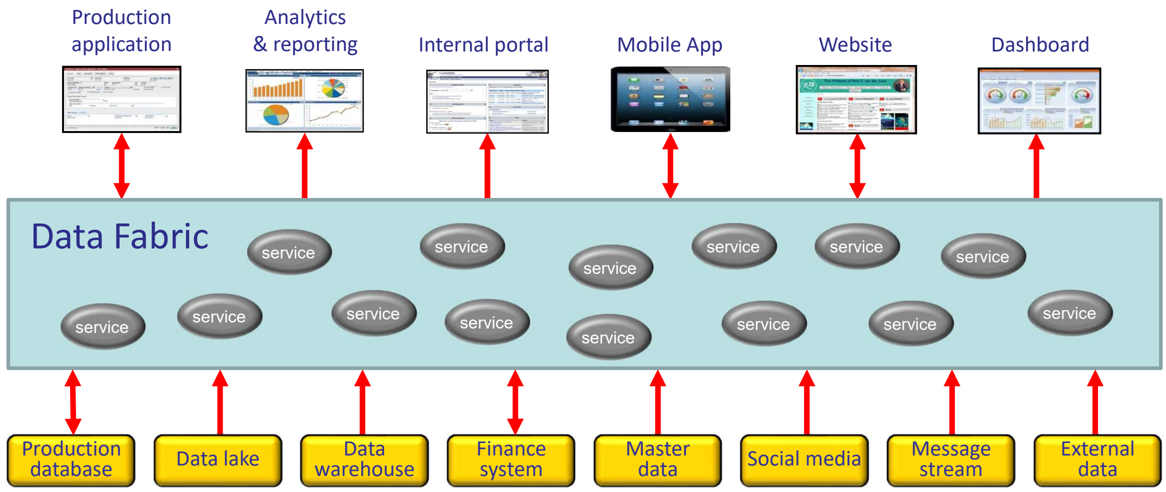
What is the Data Fabric?

- Gartner: A data fabric is generally a custom-made design that provides reusable data services, pipelines, semantic tiers or APIs via combination of data integration approaches in an orchestrated fashion.
- Gartner: Data fabric enables *frictionless access* and sharing of data in a distributed data environment. It enables a *single* and *consistent* data management framework, which allows seamless data access and processing by design across otherwise siloed storage.

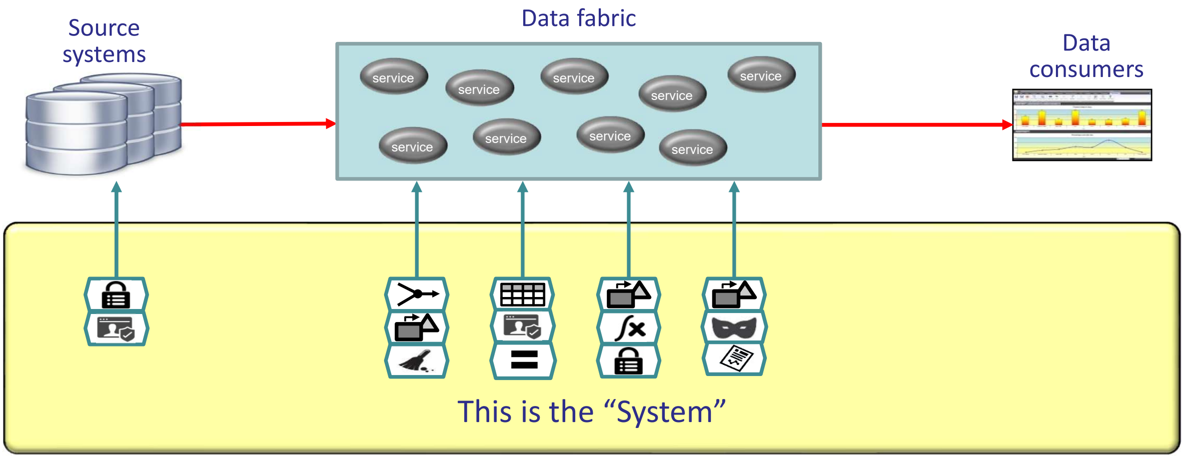
The Data Fabric



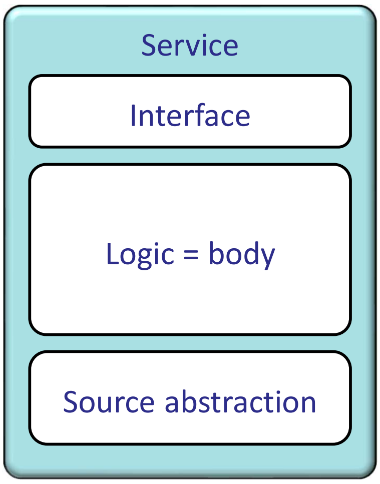
The Services of a Data Fabric



Data Fabrics and Data Processing Specifications

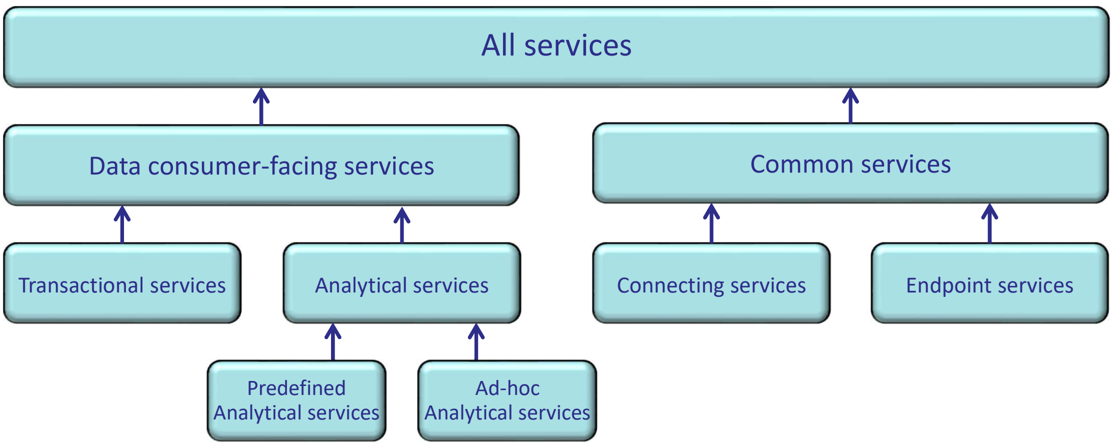


The Components of Services

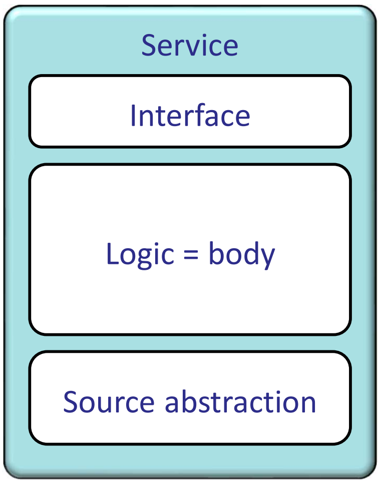


- The interface component is responsible for handling incoming parameters and outgoing results
- The logic of the service form the body
- The body deals with data processing specifications
- Abstraction layer to make it independent of changes to the IT systems

A Data Fabric Consists of Different Types of Services

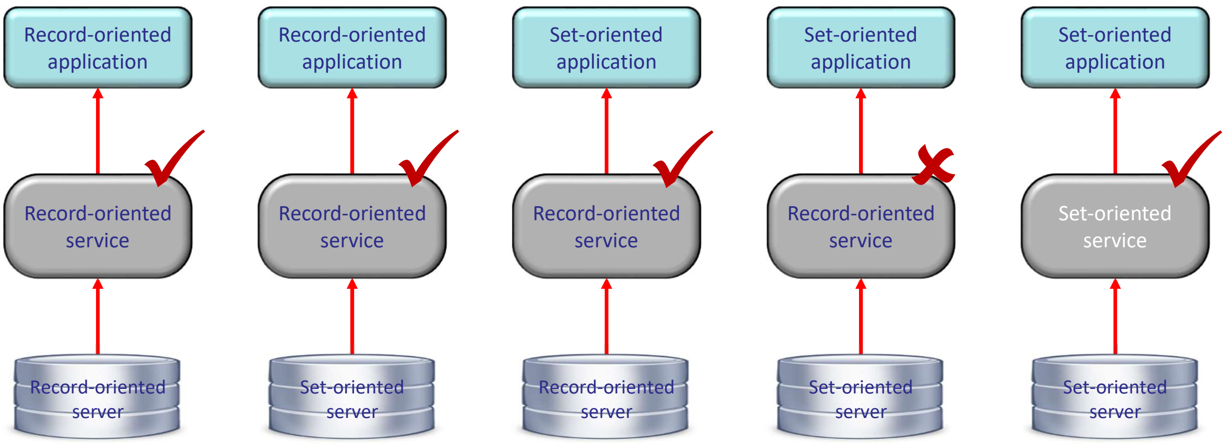


12 Capabilities for Frictionless Data Access

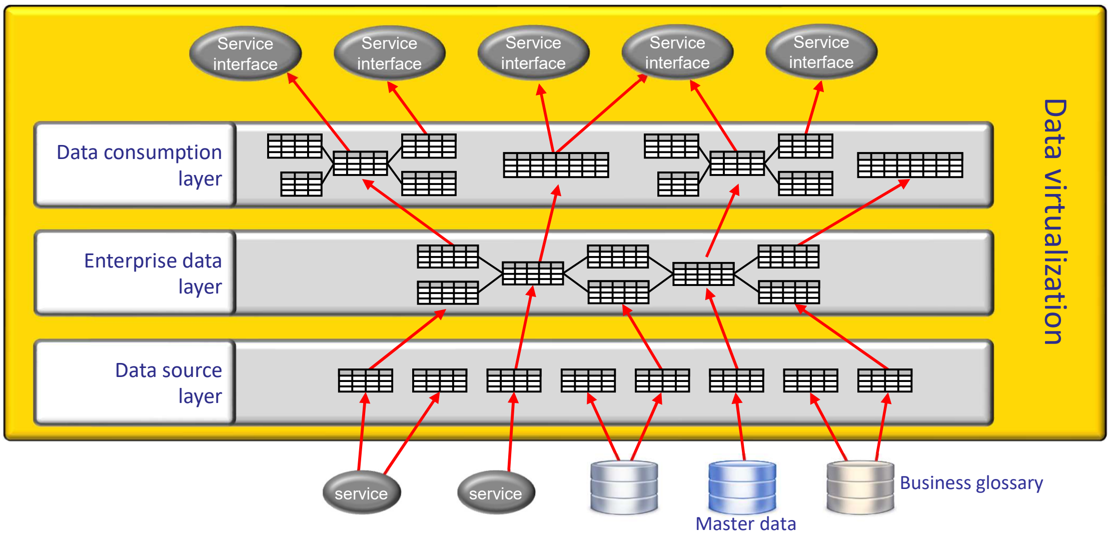


- Data preparation, such as transformations, calculations, aggregations, filters, joins, …
- Adaptable logic
- High performance
- Data access by many data consumption forms
- Access to all the data sources
- Processing of all types of data
- Data security and privacy
- Real-time data access
- Read and write data access
- Data minimization
- Event processing
- Technical and business metadata management
- Master and reference data management

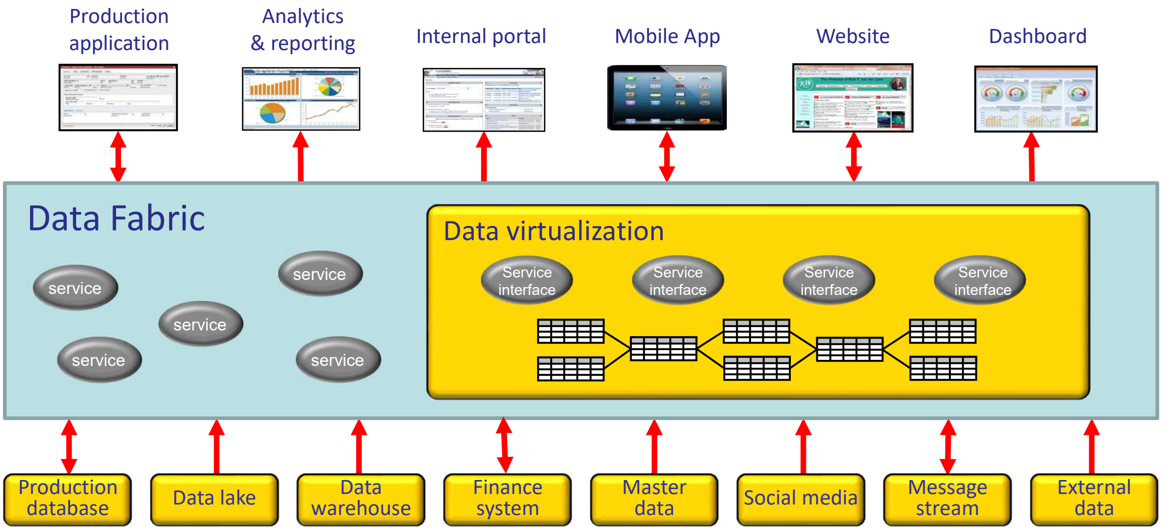
Record-Oriented or Set-Oriented Interface?



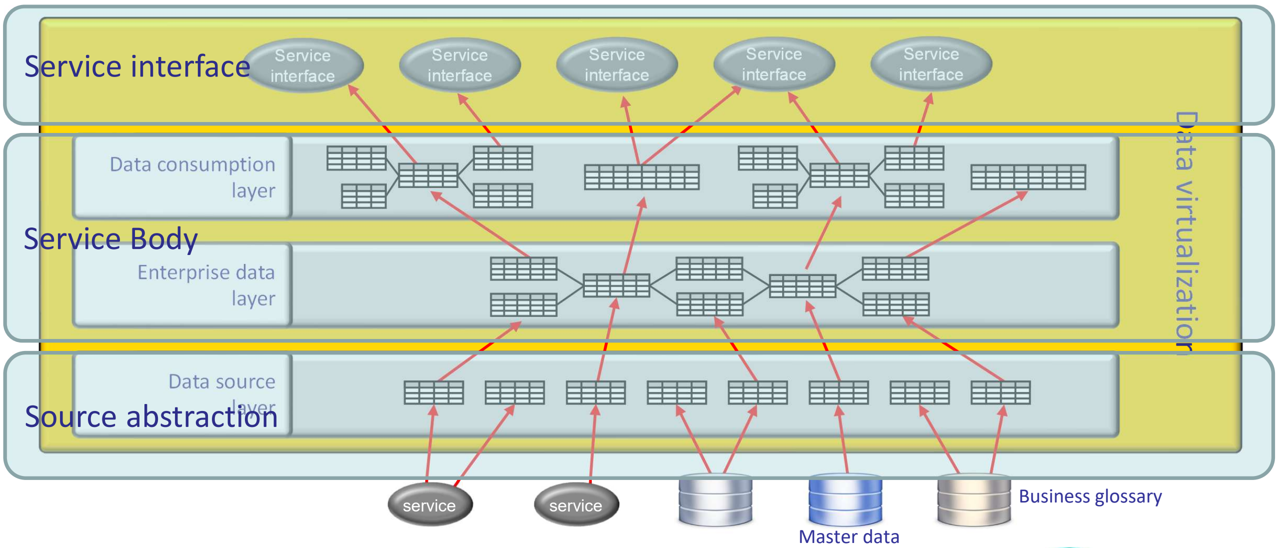
Developing Data Fabric Services with Data Virtualization



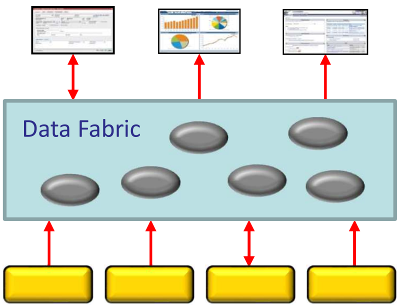
The Data Fabric



Developing Data Fabric Services with Data Virtualization

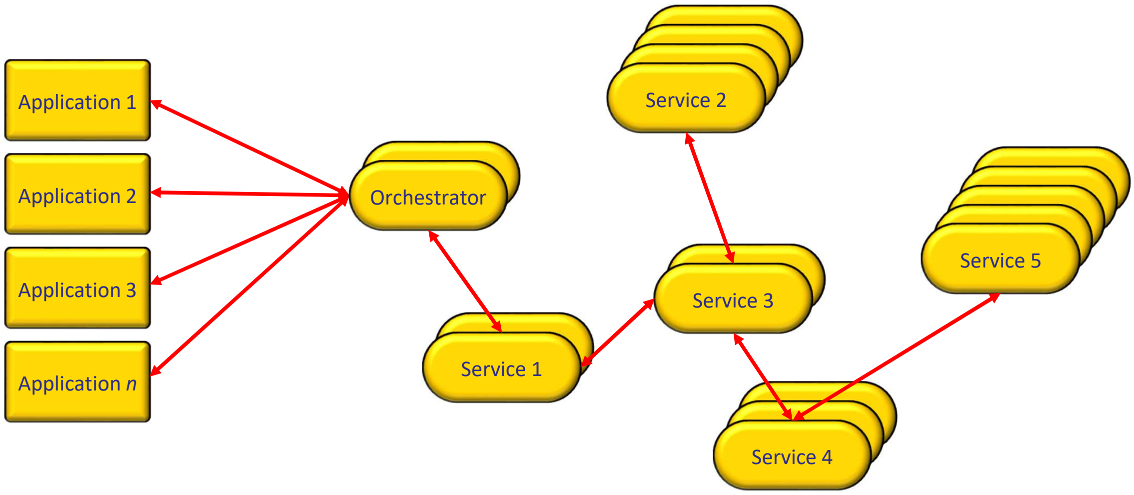


Remarks on Data Fabric



- Poorly defined concept
- Metadata is required
- Master data needed to integrate data correctly
- A data fabric may contain a data warehouse, data lake, or data hub
- Dedicated tool market is small, e.g. Cinchy
- Data fabric must support transactional and analytical workloads

Data Fabric ≠ Micro-Services Architecture

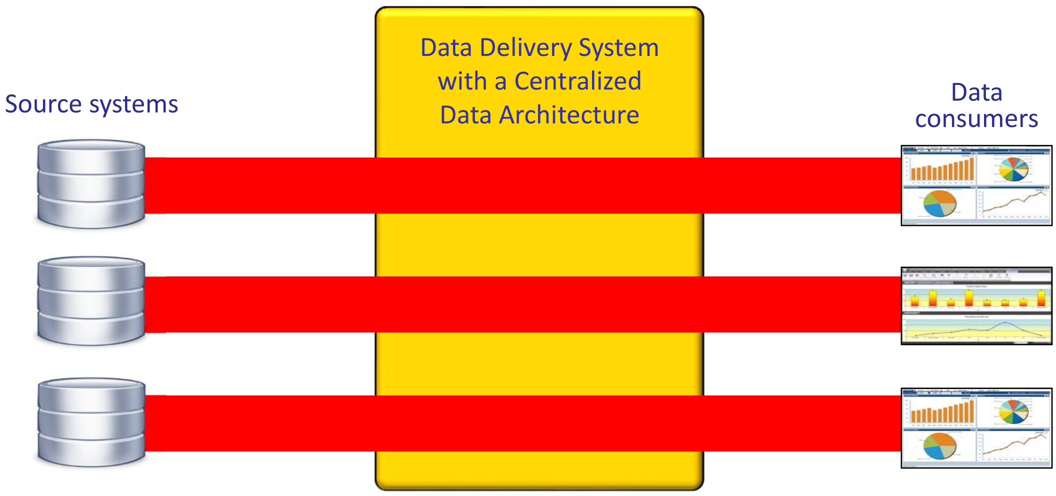


Part 6.7:
The Data Mesh

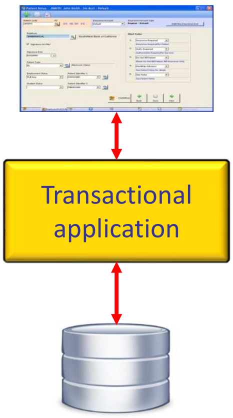
What is a Data Mesh?

- Introduced by Zhamak Dehghani:
- "Data platforms based on the data lake architecture have common failure modes that lead to *unfulfilled promises* at scale.
- To address these failure modes we need to shift from the *centralized paradigm* of a lake, or its predecessor data warehouse.
- We need to shift to a paradigm that draws from *modern distributed architecture*: considering *domains* as the first class concern, applying platform thinking to create self-serve data infrastructure, and *treating data as a product*."

Single-Domain Data Consumers

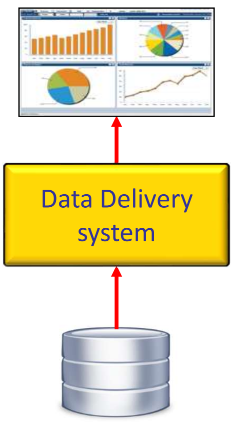


Data Engineers for Transactional Applications



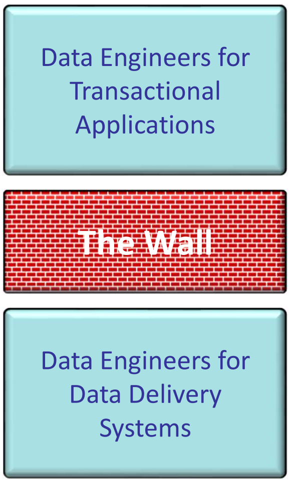
- They are single-domain experts
- They focus only on data requirements of transactional applications
  - Not on other forms of data consumption, such as BI
- They do not make the data easily consumable
- They implement all the business rules
- They know about all the exceptions
- They know when changes are implemented

Data Engineers for Data Delivery Systems



- They need to understand the data of all the domains
  - Hyper-domain experts
- They need to transform the data into consumable data
- They need to work with data not designed to be integrated
- They need to understand all the business rules that need to be applied
  - Complex ETL processes
- They need to understand the data requirements of the data consumers
- They need to deal with SLAs of the data consumers

The Wall between Two Groups of Data Engineers

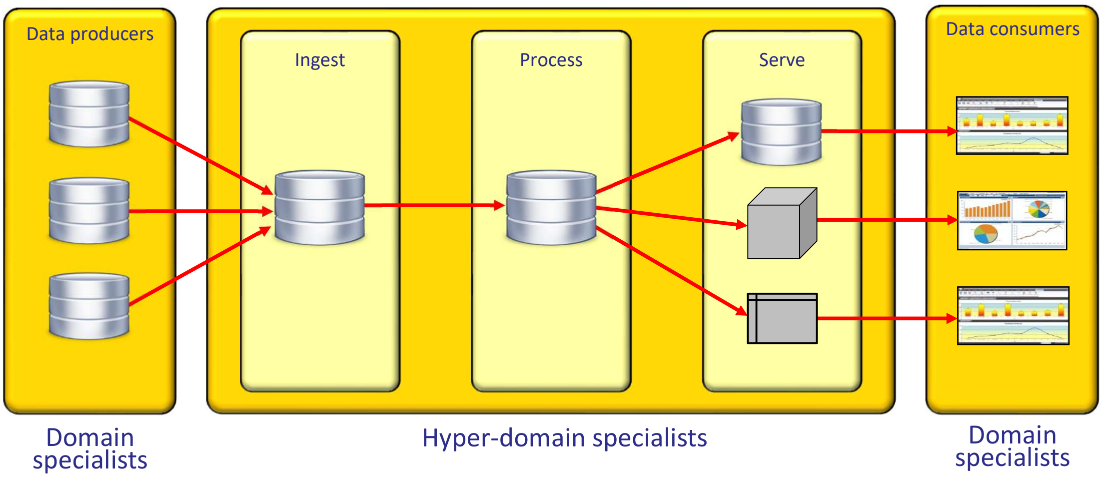


- Different groups of data engineers
- Different tool sets
- Different responsibilities
- Changes of data or data structures not always communicated
- Who owns the data and who is responsible for data quality?
- How to implement “the right to be forgotten/corrected”?

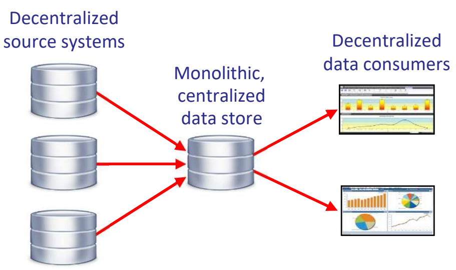
Interfaces of Transactional Applications

- Most are not developed to offer an interface
- Interfaces that do exist, are commonly developed for record-oriented access
- Direct database access complex or not always allowed
- Rarely support for historic data
- Risky to bypass multi-tenant systems

Current Centralized, Monolithic Data Architectures

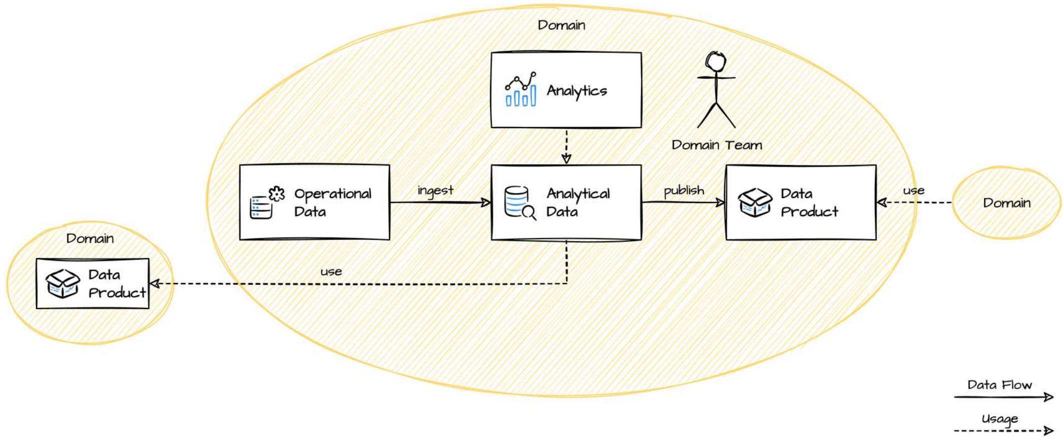


Traditional Centralized, Monolithic Architectures



- Large and complex monolithic solutions
- Single-domain versus multi-domain experts
  - Application developers = single-domain experts
  - Data engineers = multi-domain experts
  - Data consumers = single-domain experts
- Data engineers need to understand all the business logic
- Who owns the data in the central database?
- Storage of redundant data

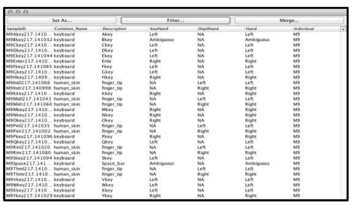
A Domain-Oriented Architecture



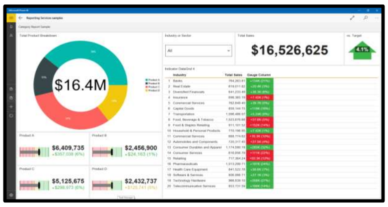
datamesh-architecture.com

Potential Data Products

Data as file



Report



Service

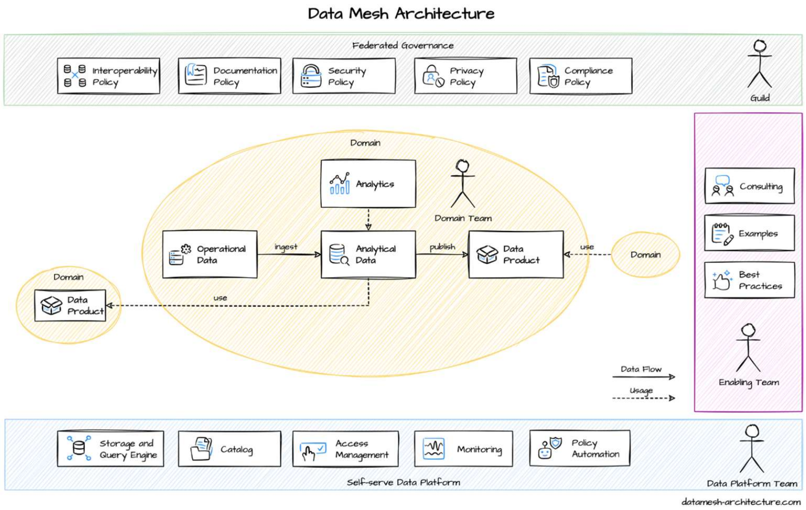
Data via SQL

Apps

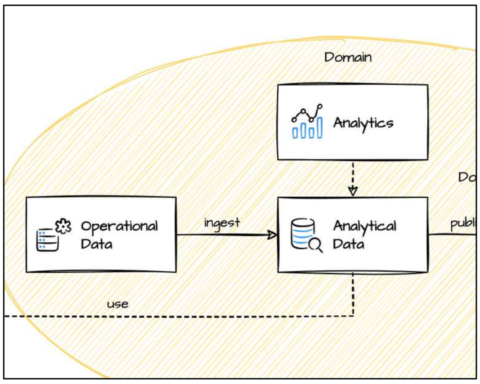
Embeddable KPI

Stream of Data

The Periphery of a Domain

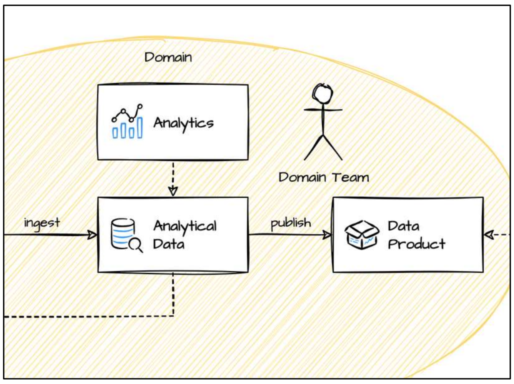


What is Ingest?



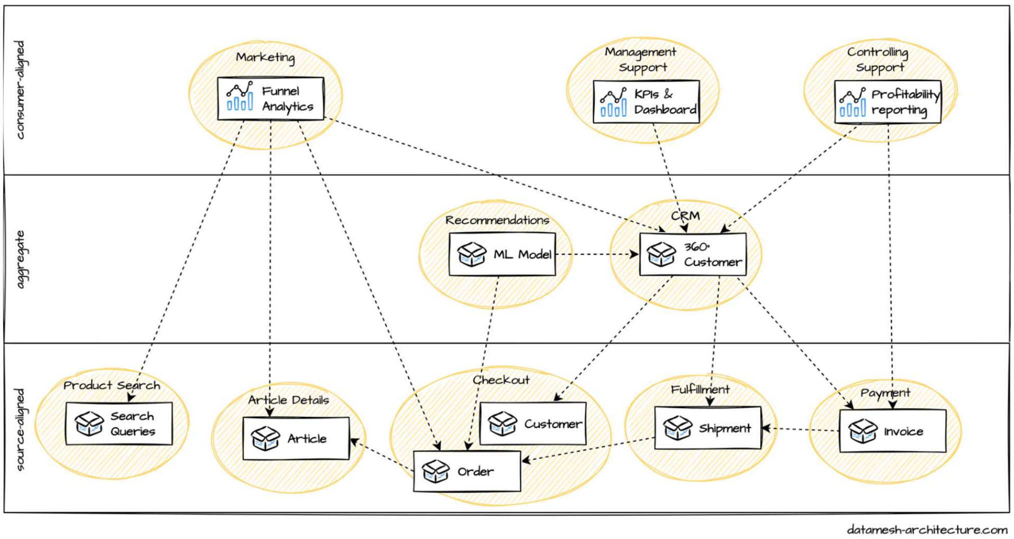
- Potential technologies:
  - ETL
  - Data replication (Change Data Capture)
  - ESB (Enterprise Service Bus)
  - Messaging
  - Database triggers
  - Data virtualization
- Involves data processing specifications
  - Transforming data values and structure
  - Masking
  - Cleansing
  - Calculations
  - …

Storing Analytical Data



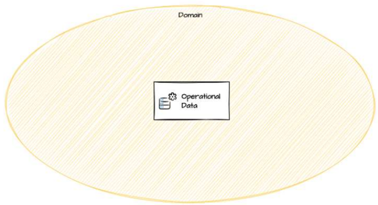
- One database-solution per domain
  - Multiple databases for different use cases
- Different architectural solutions
  - Data warehouse
  - Data warehouse + data marts
  - Data lake
  - Data lakehouse
  - Data hub
  - Translytical database

The Mesh Itself

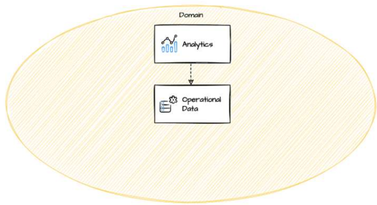


The Domain Team's Journey

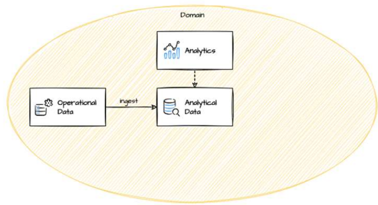
No data analytics



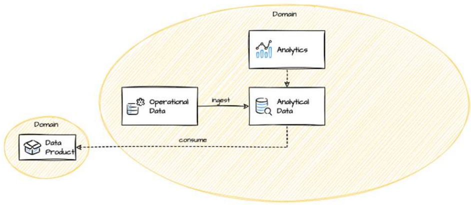
Operational database queries



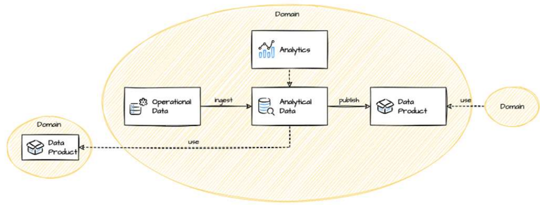
Analyze own data



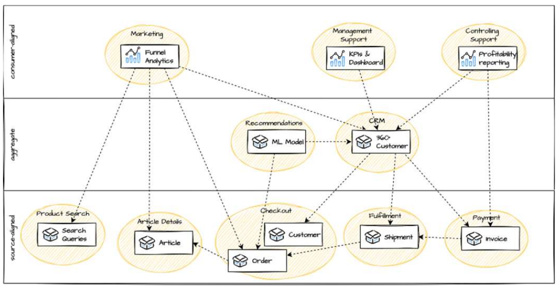
Analyze Cross-domain data

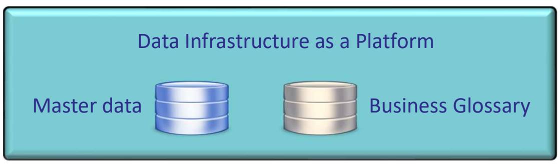


Publish data as a product



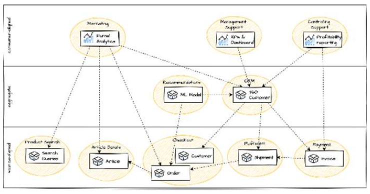
The Foundation: Data Infrastructure as a Platform





- Scalable polyglot big data storage
- Encryption for data at rest and in motion
- Data product versioning
- Data product schema
- Data product de-identification
- Unified data access control and logging
- Data pipeline implementation and orchestration
- Data product discovery, catalog registration and publishing
- Data governance and standardization
- Data product lineage
- Data product monitoring/alerting/log
- Data product quality metrics (collection and sharing)
- In memory data caching
- Federated identity management
- Compute and data locality
- ...

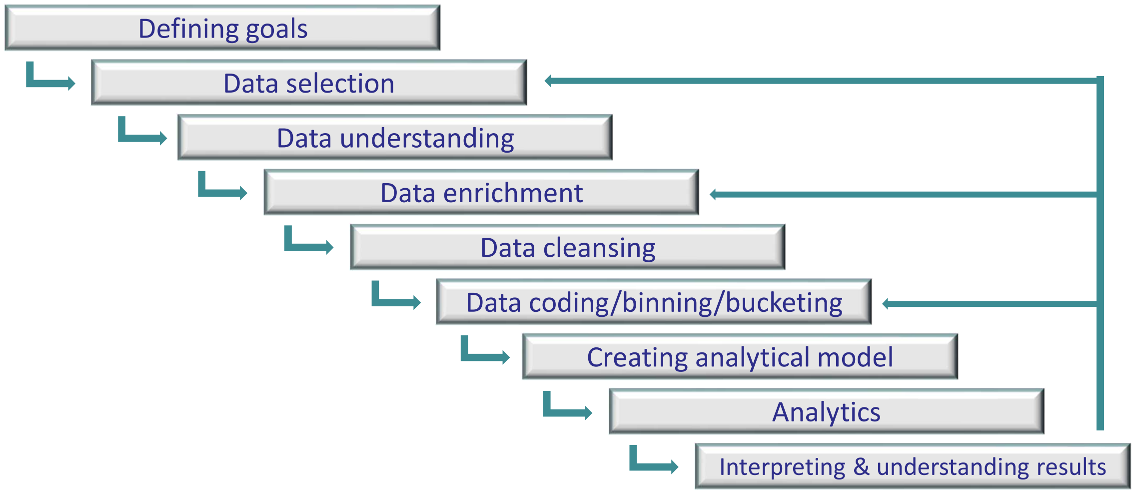
Data Mesh Challenges



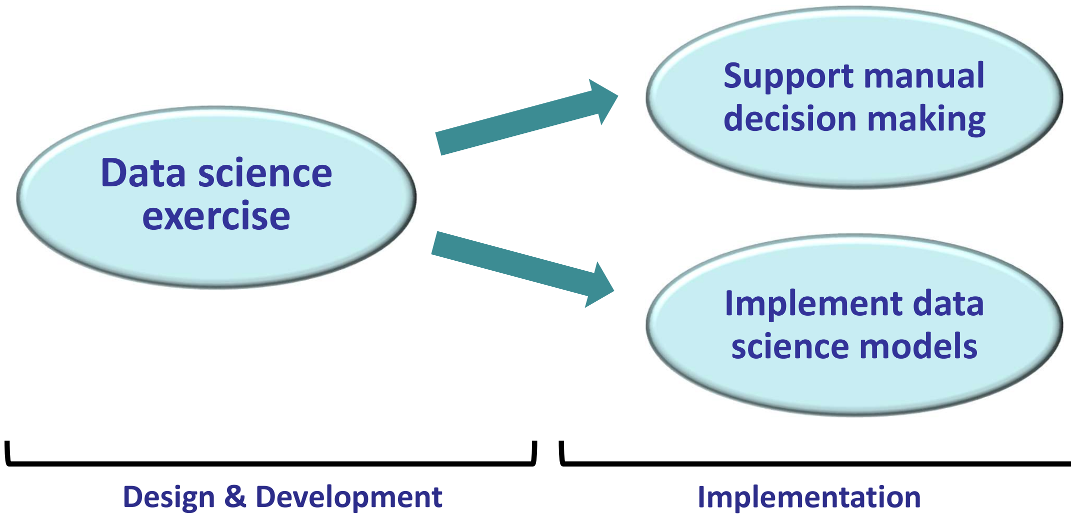
- Massive organizational change for IT
- Standardization of interfaces required
- How clear are BI requirements when new transactional applications are developed?
- Development of transactional applications more complex
- What about transactional applications that are bought?
  - Do they need to be wrapped?
- What about multi-domain transactional applications?
- Distribution of knowledge of data delivery technologies

Part 6.8:
Embedding Data Science Models in Data Architectures

Development Steps for Data Science



Operationalization of Data Science Models



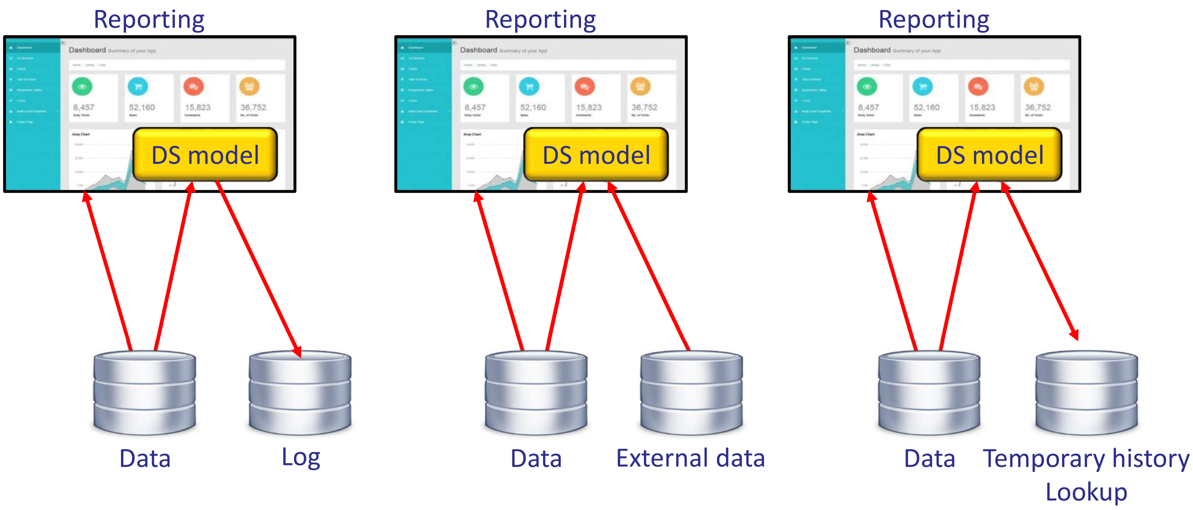
Operationalization of Data Science Models

Type of Decision	Example
Singular manual decision	What will be the impact on total sales of acquiring company X?
Repeatable manual decision	Does a specific location have the right characteristics for opening a new shop?
Partial automated decision	In a call center: What is the churn risk for a customer? What should we offer?
Full automated decision without automated reaction	When credit card payment is dubious, send message to operator
Full automated decision with automated reaction	When sensor indicates the component is heating up too fast, switch off machine
…	…

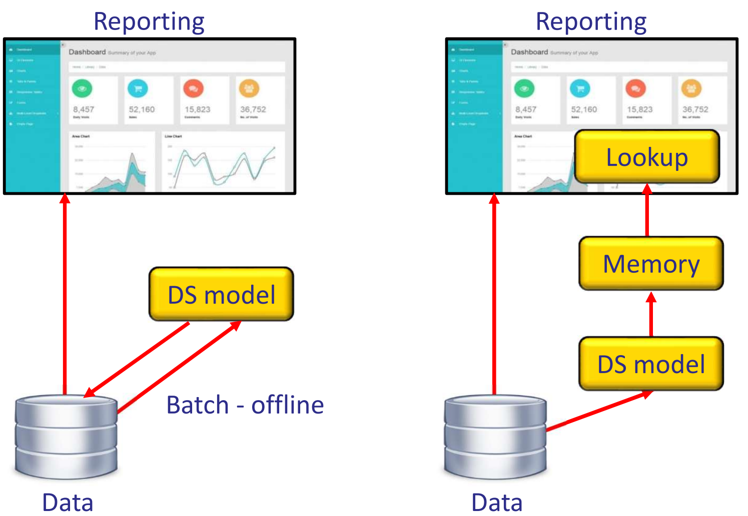
Requirements for a Supporting Data Architecture

- Versioning of data science models
  - Immutable models
- Auditable data science models
  - Reproducible data for reproducible models
  - Transparency of models
- Different codings must be easy and quick to apply
- Self-learning models or not?
- Delivering metadata
  - Descriptions, definitions, tags, relationships, searchable
- Fast evaluation of models
  - Max time to execute model, SLAs
- And many more …

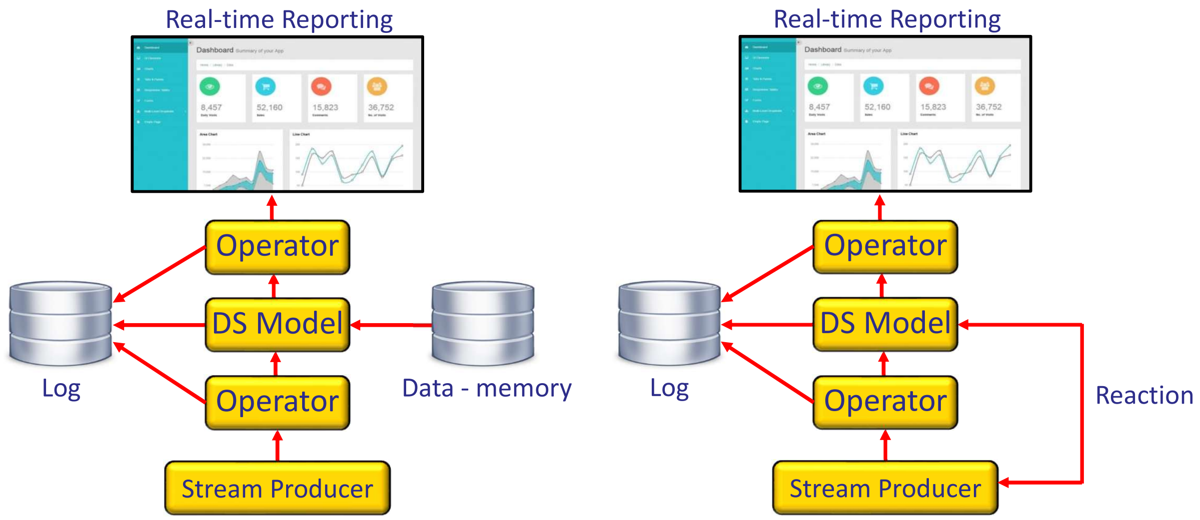
Architectural Aspects (1)



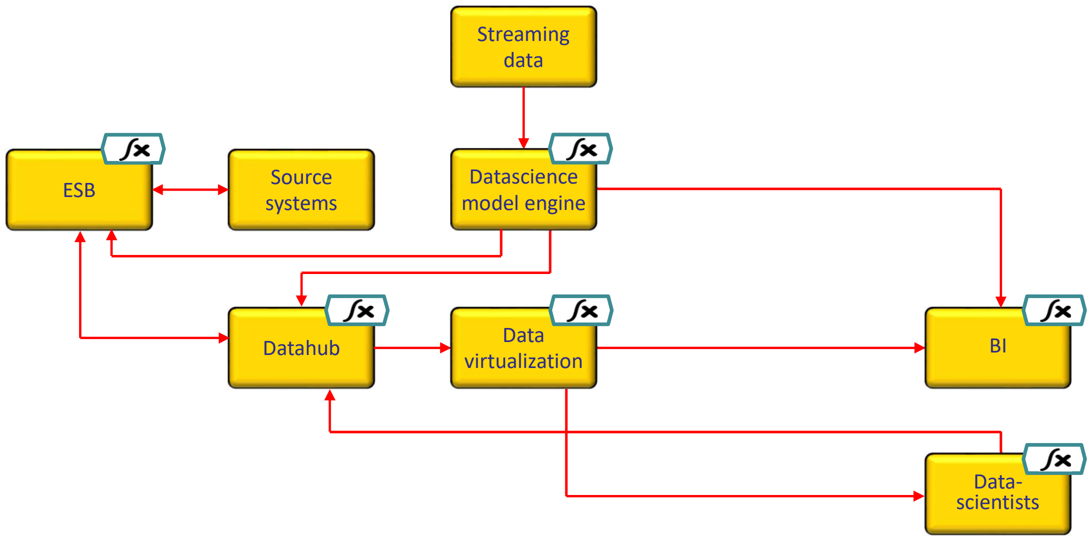
Architectural Aspects (2)



Architectural Aspects (3)



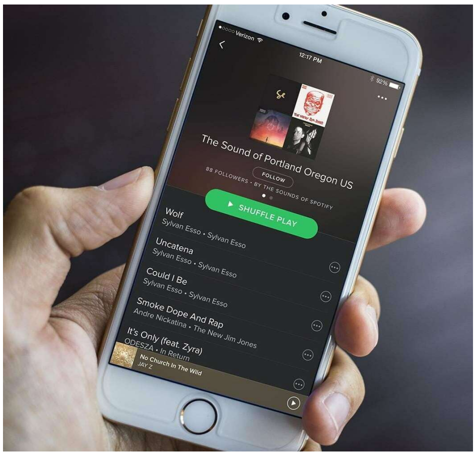
Example of a Data Architecture



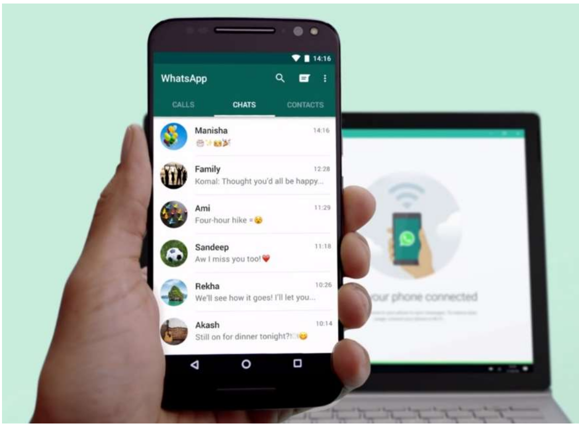
Part 6.9:
Netflixing Your Data

From Video-by-Sneakers to Video-on-Demand

From Music-by-Sneakers to Music-on-Demand



From Message-by-Pigeon to Message-on-Demand



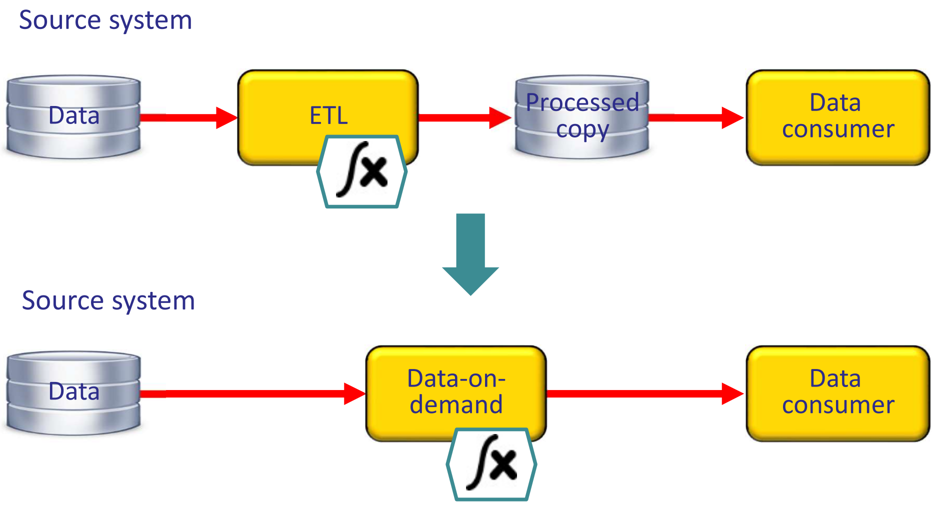
Video-on-Demand

Music-on-Demand

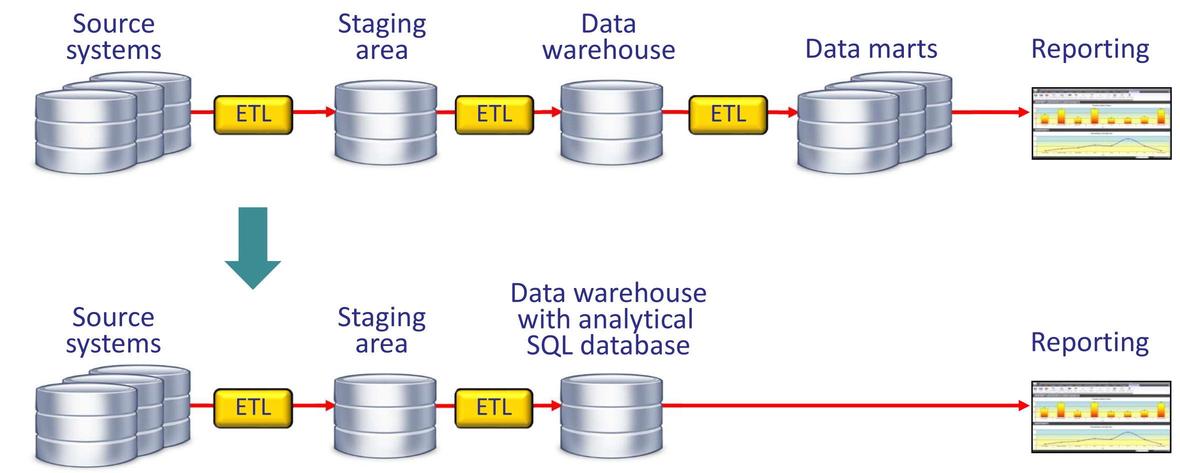
Message-on-Demand

Data-on-Demand ?

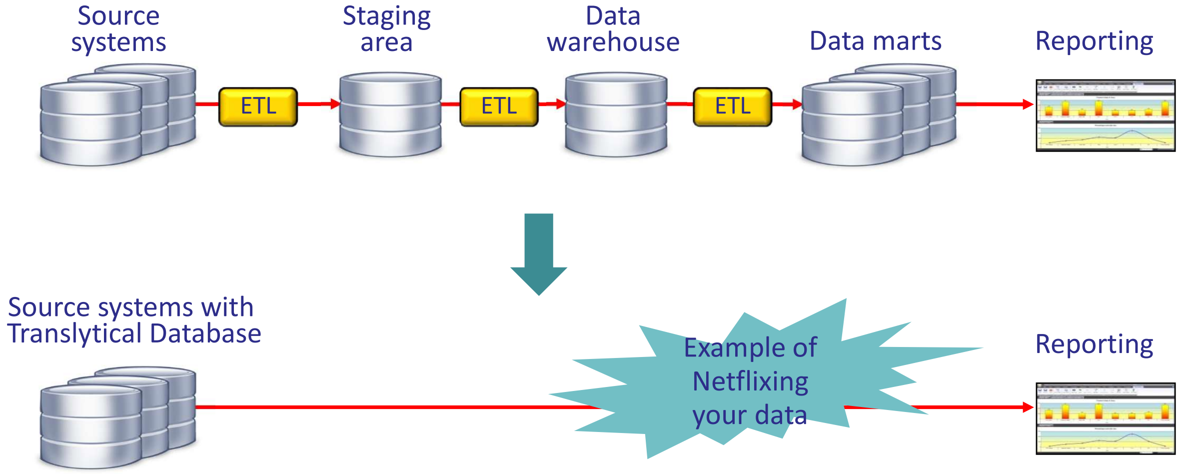
Replace Derived Data by Original Data (1)



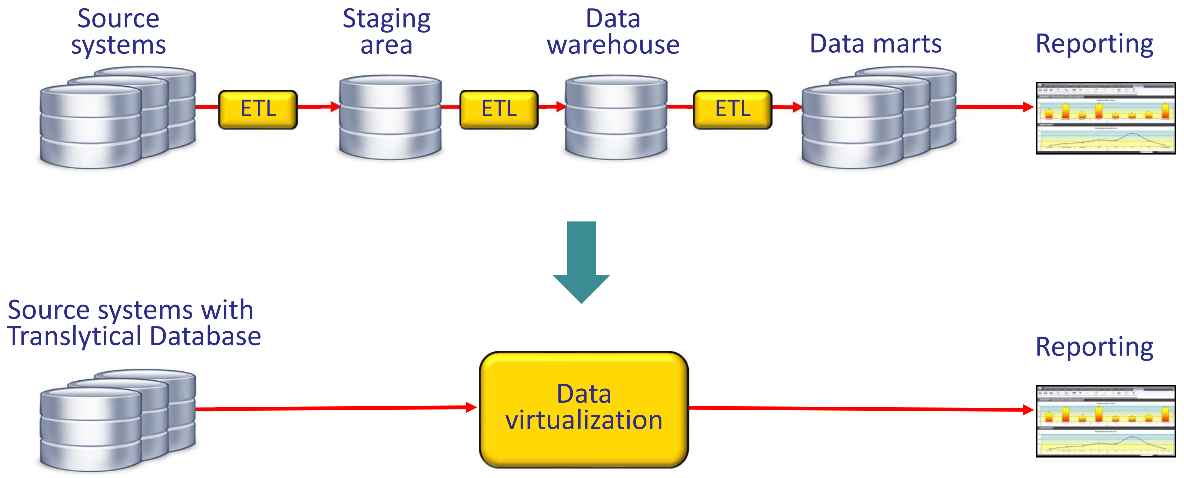
Replace Derived Data by Original Data (2)



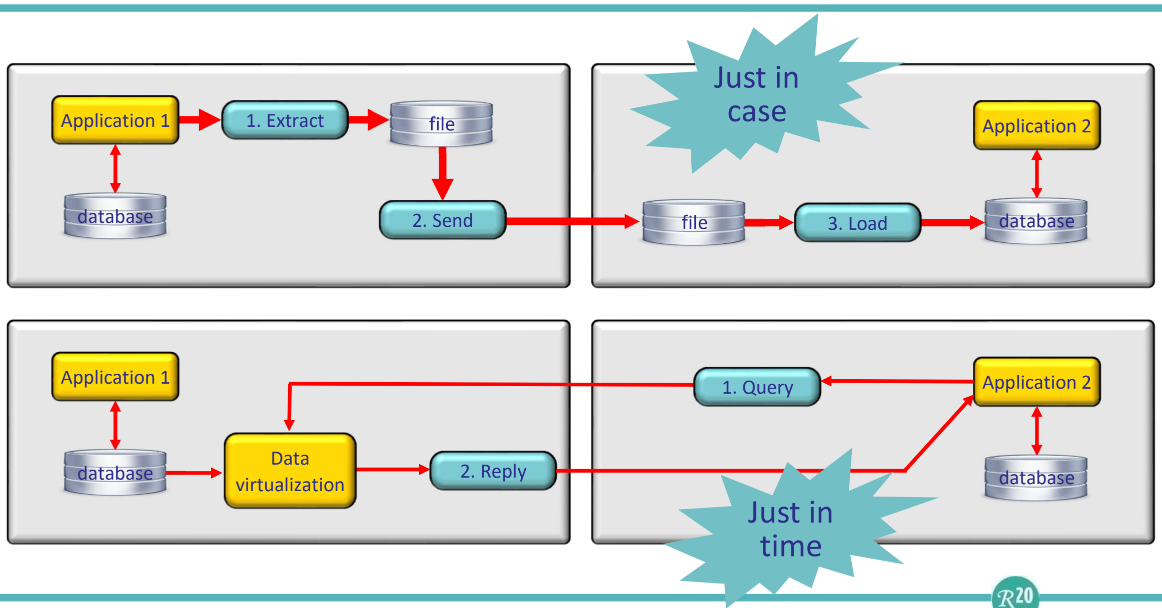
Replace Derived Data by Original Data (3)



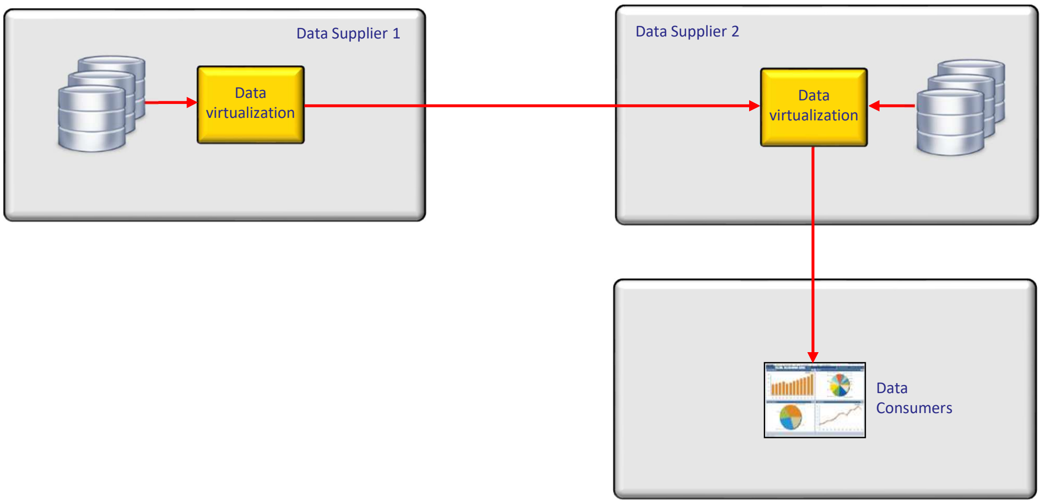
Replace Derived Data by Original Data (4)



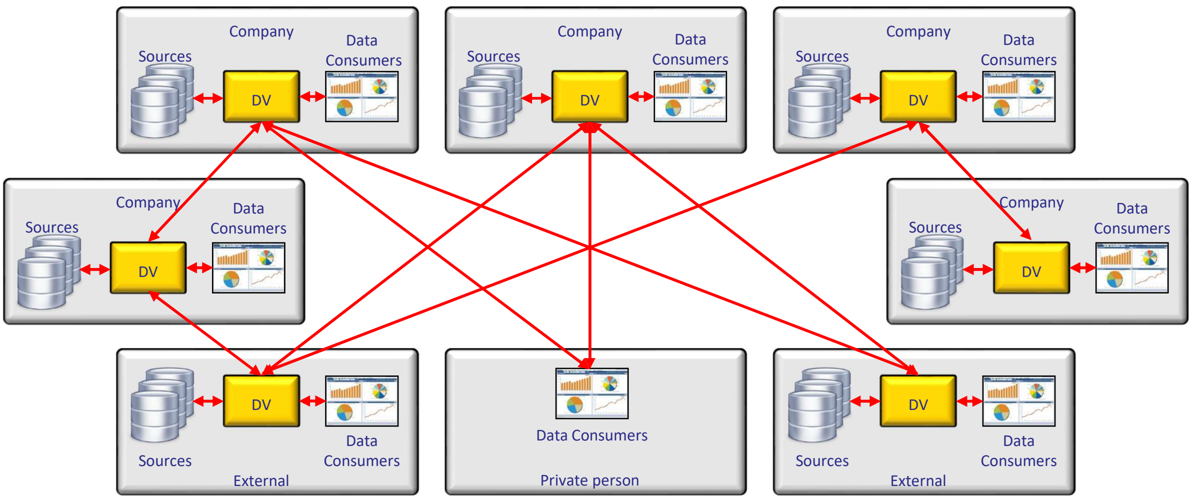
Replace File Transfer by Data-on-Demand

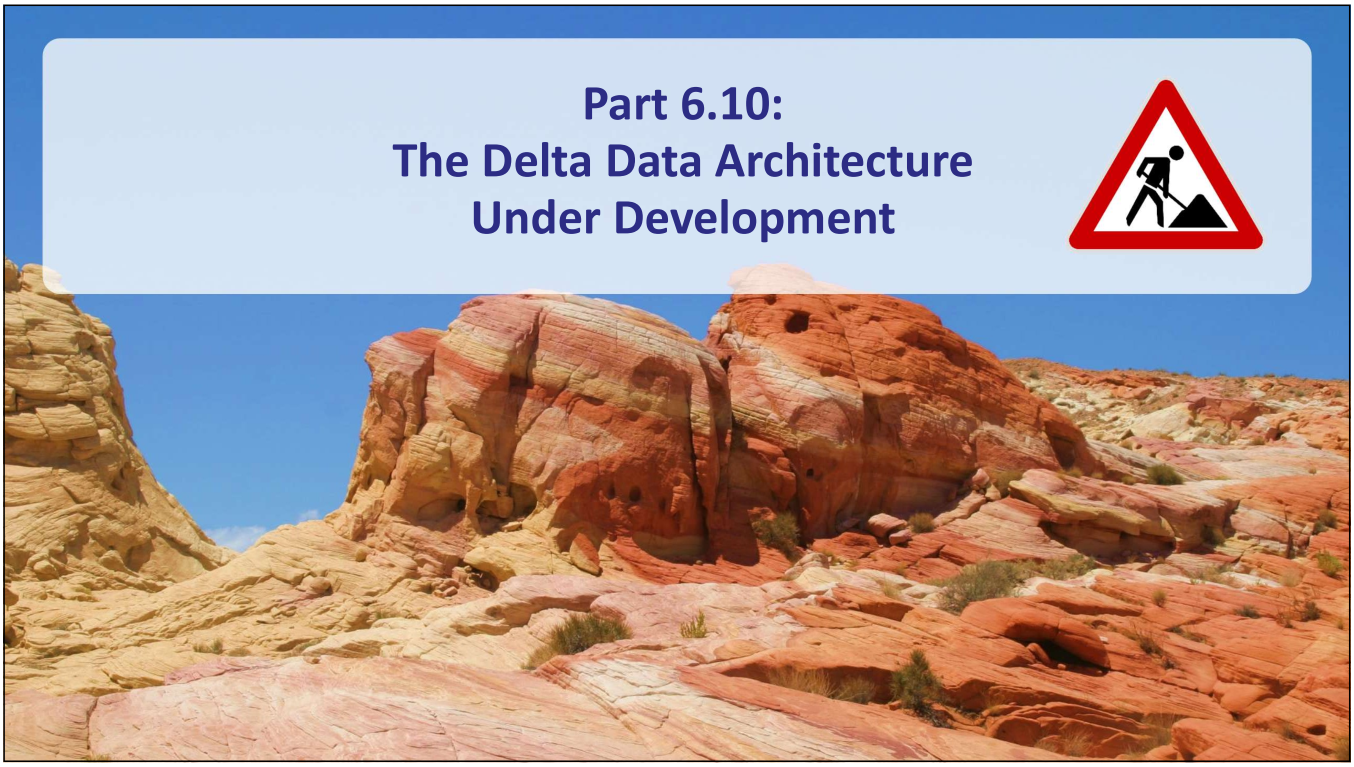


Global Data Architecture Based on Data Minimization



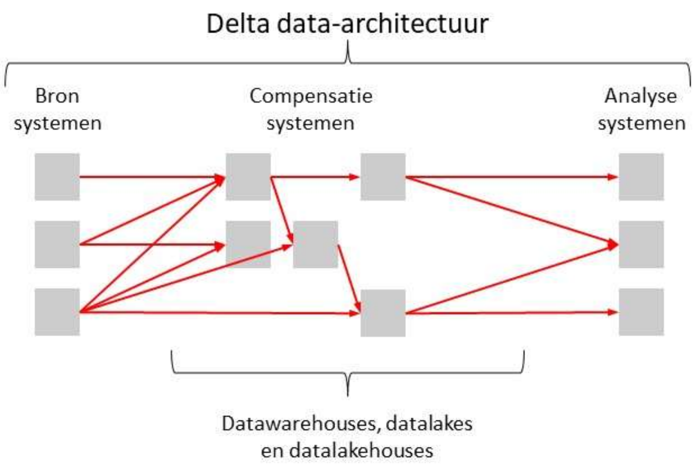
Global Data Architecture Based on Data Minimization



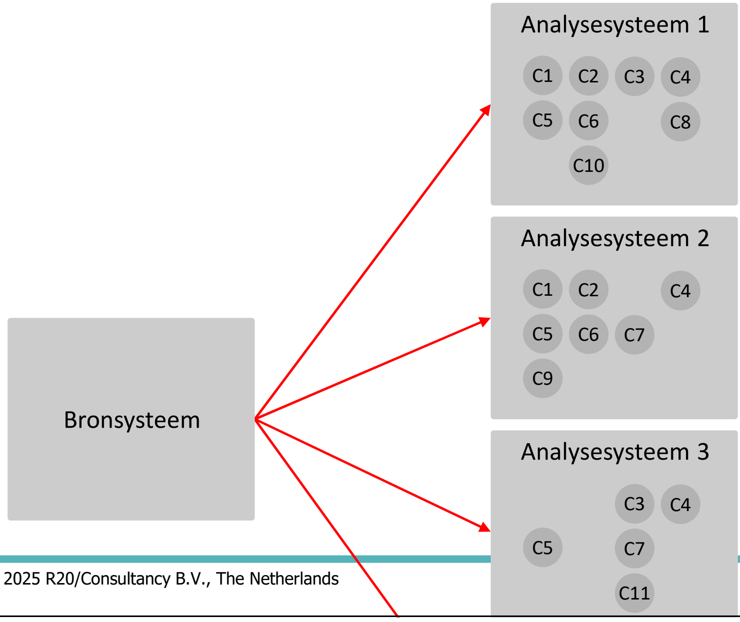


Part 6.10: The Delta Data Architecture Under Development

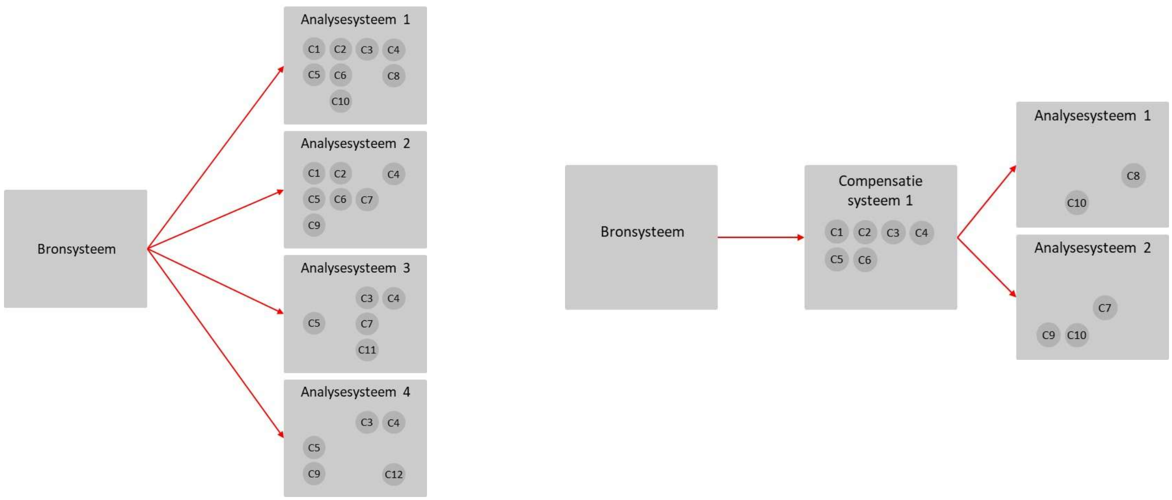
Delta data-architectuur: van bron tot inzicht



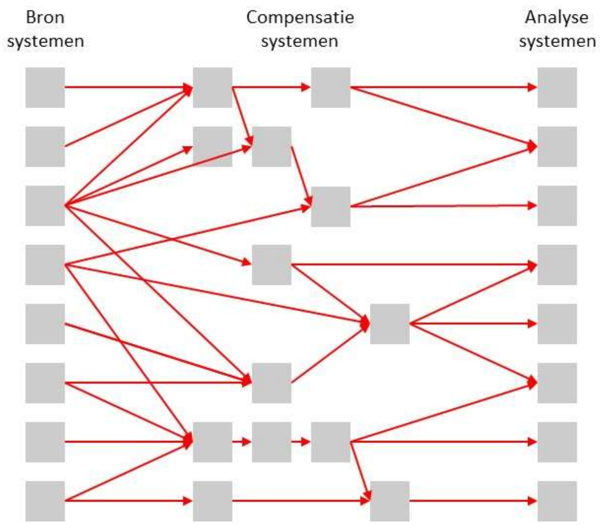
Ontbrekende functionaliteit huidige bronystemen



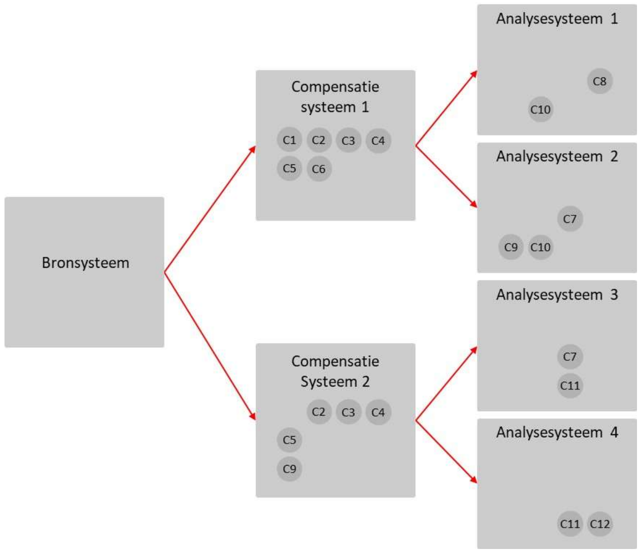
Ontbrekende functionaliteit huidige bronystemen



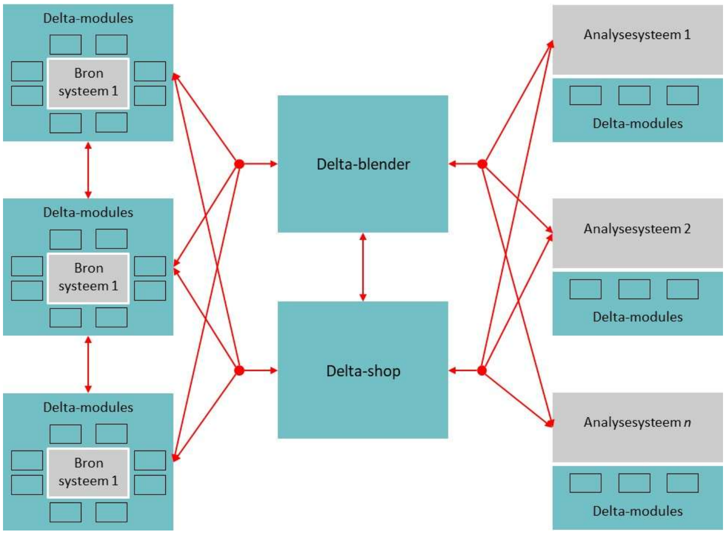
Niet één maar vele compensatiesystemen



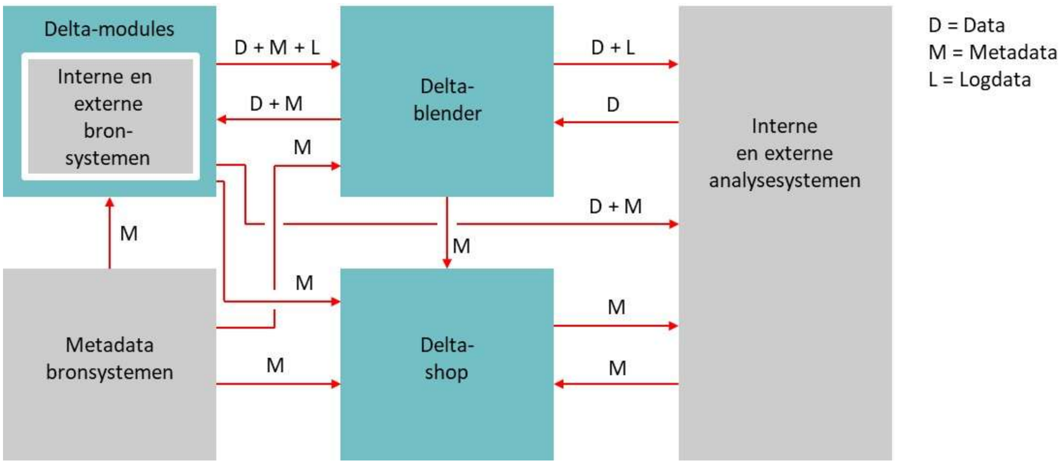
Duplicatie in bronsystemen



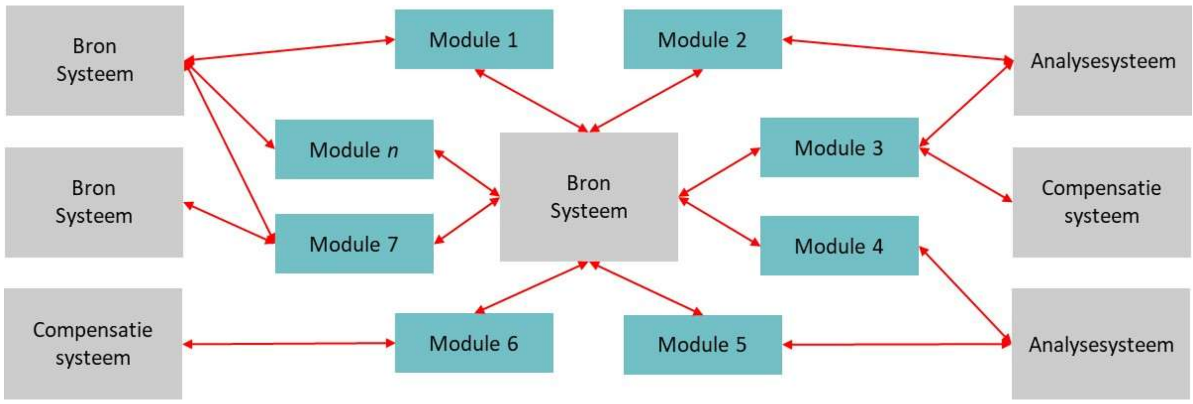
Globale Delta data-architectuur



Globale Delta data-architectuur



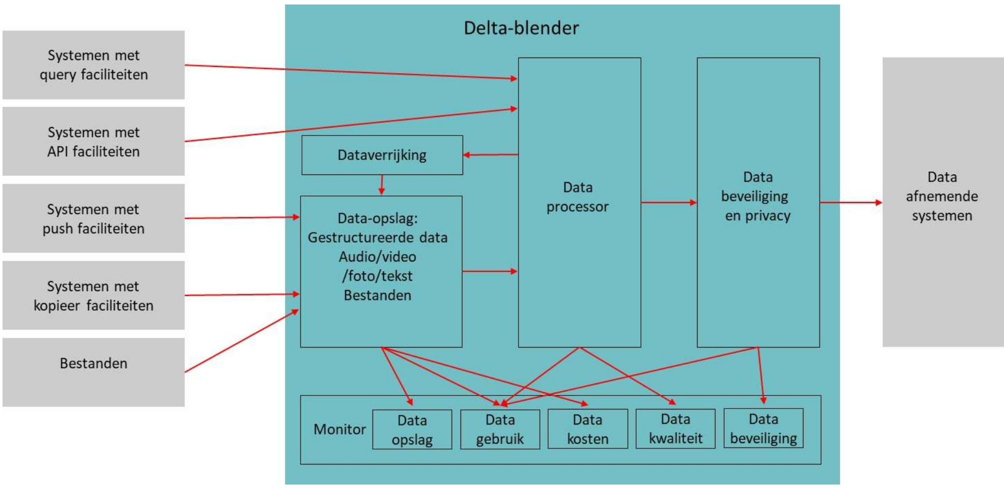
Delta-Modules ontkoppelen bronsystemen



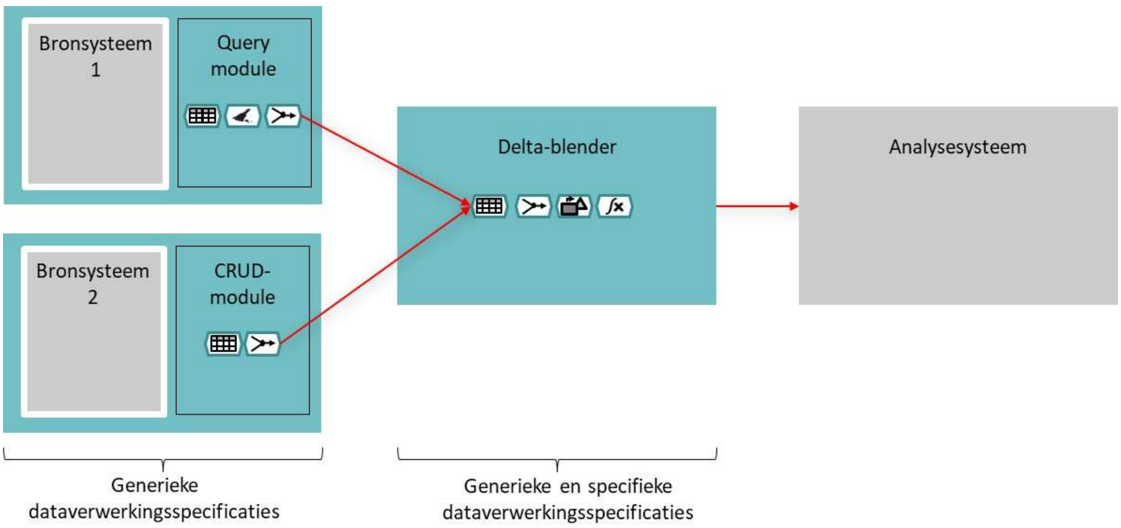
Voorbeelden van Delta-modules

- CRUD
  - Bewerken en opvragen van individuele business-objecten
  - Tijdreizen
  - Conform het *Enterprise datamodel*
- Query
  - Opvragen van verzamelingen van business-objecten
  - Tijdreizen
  - Conform het *Enterprise datamodel*
- Business logica
  - Bijvoorbeeld: complexe berekeningen, controles en beslissingen
- Databeveiliging
  - Focus op autorisatie
- Datakwaliteit
  - Actief en passief
- Log
  - Alle vormen van datagebruik
  - Benaderbaar voor analyses
- Metadata

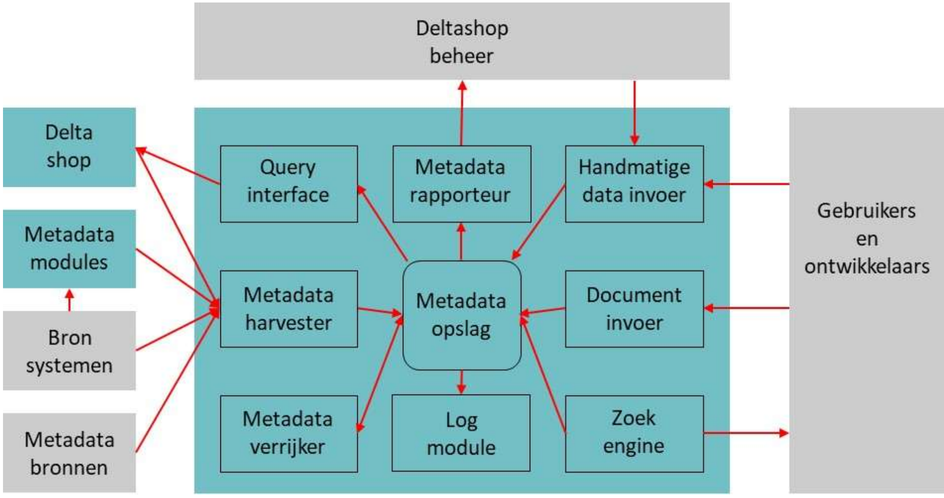
Delta-blender



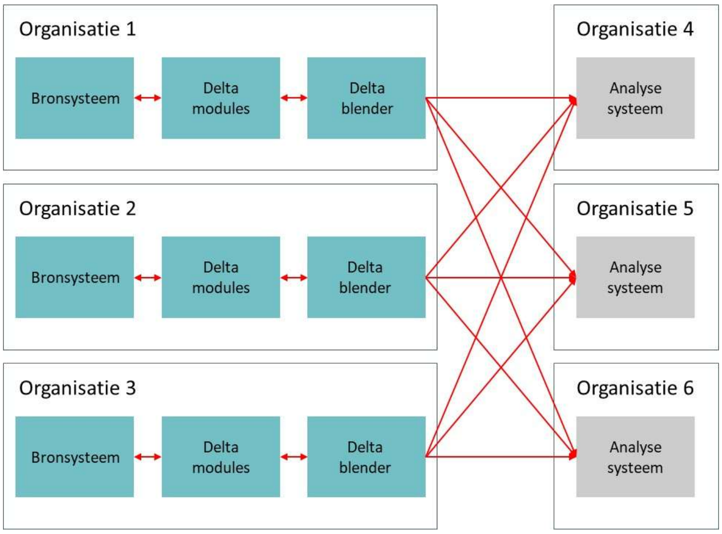
Welk onderdeel doet wat?



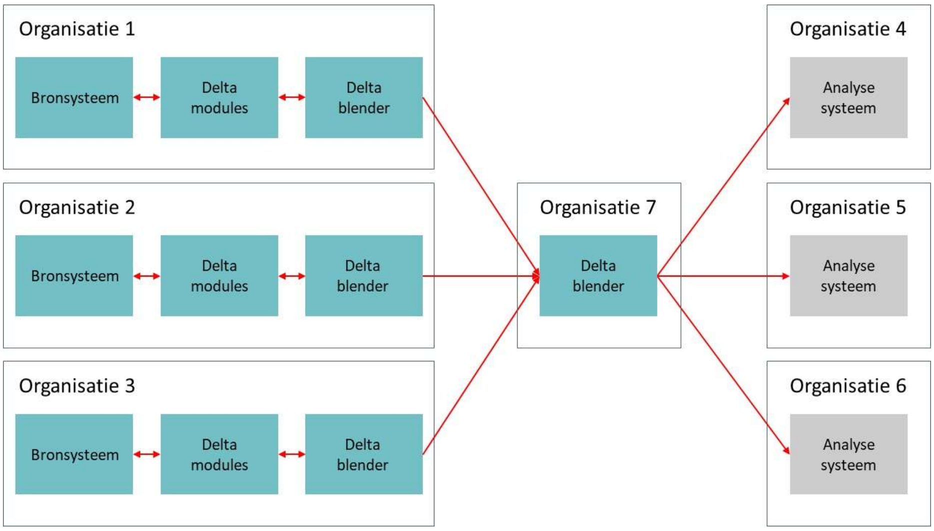
Delta-shop



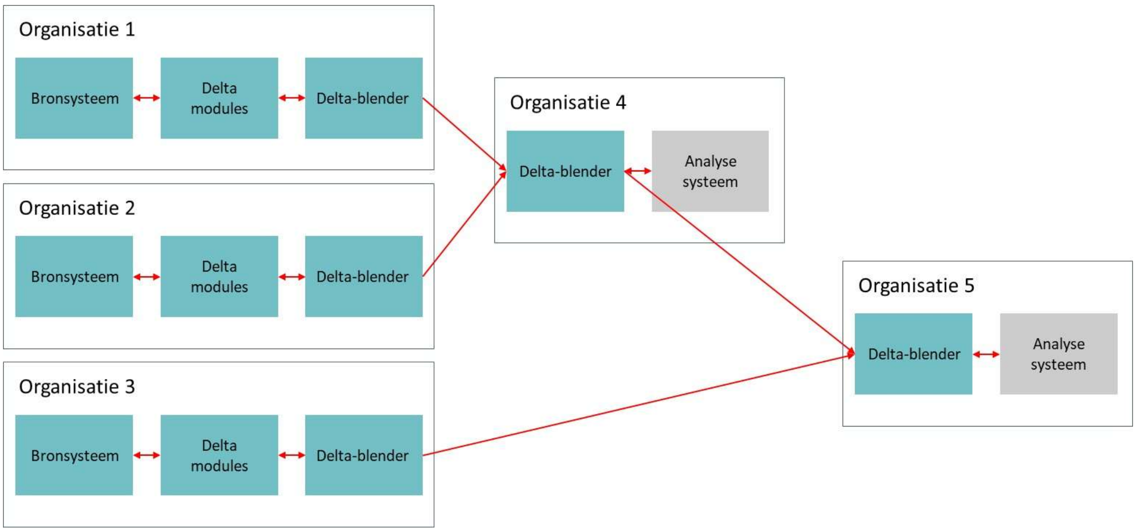
Organisatie-overstijgende vraagstukken voor overheid (1)



Organisatie-overstijgende vraagstukken voor overheid (2)



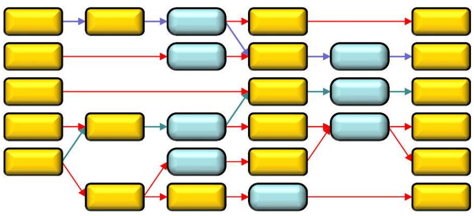
Organisatie-overstijgende vraagstukken voor overheid (3)



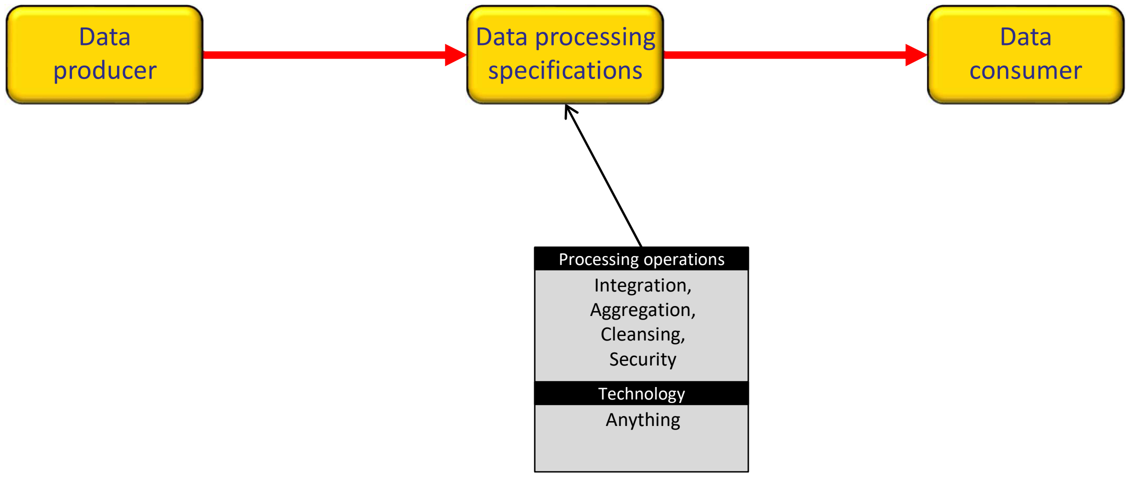
Part 7:
Steps 7-8: Design the New Data Architecture, Determine the Implementation Approach

What is a Data Track?

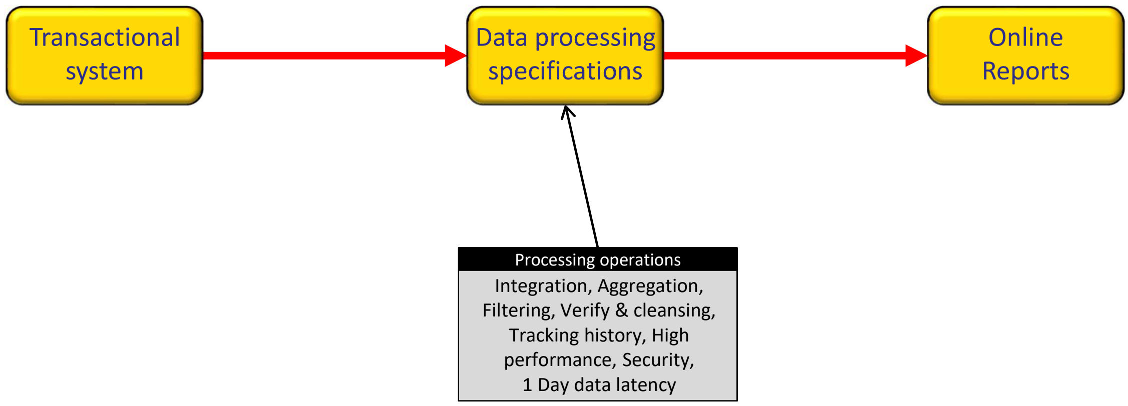
- A *data track* indicates how data “flows” from data producers to data consumers, and specifies the data processing specifications to be applied and by which module.
- Multiple data consumers can share one data track.
- Data tracks may merge and split.



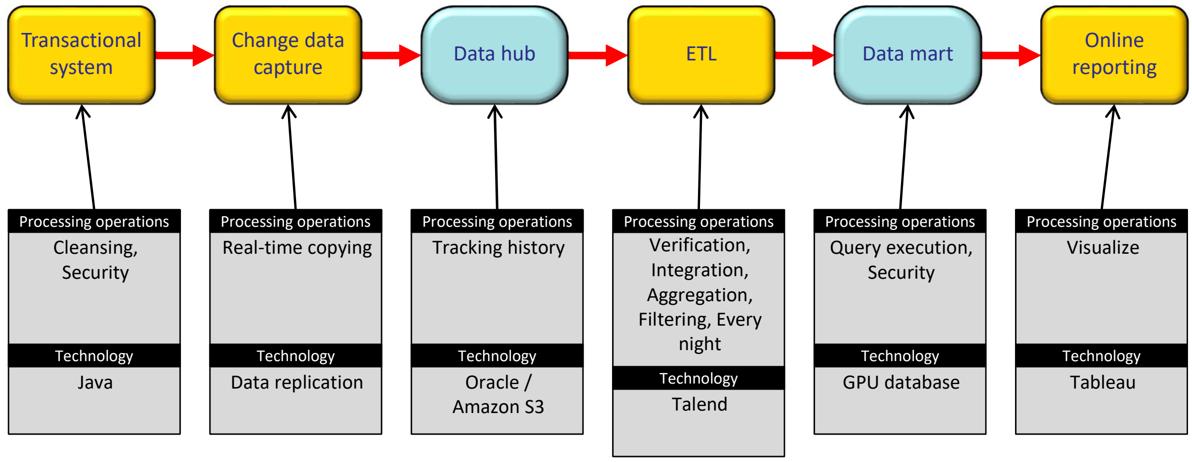
A Data Track Diagram



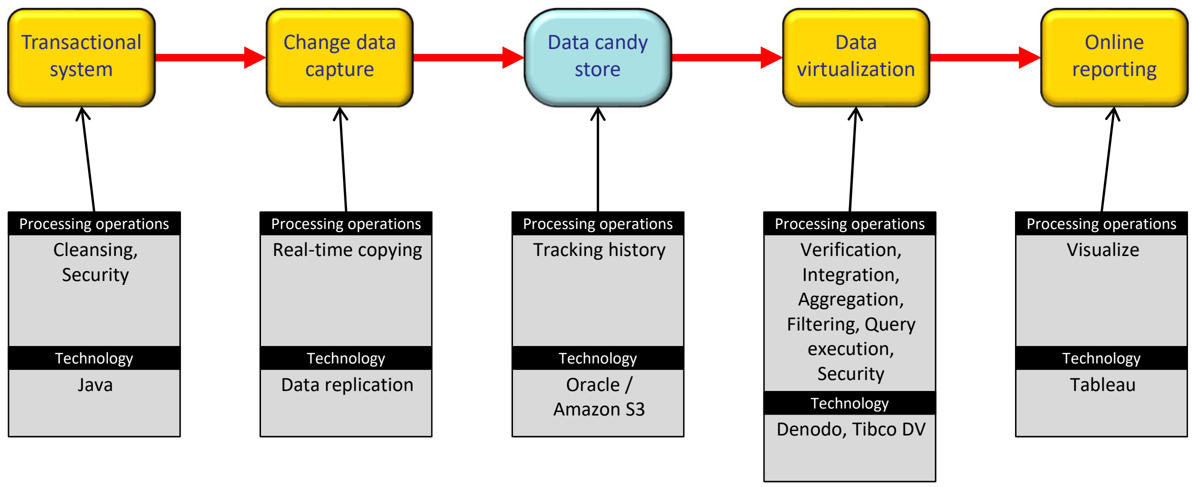
Data Track Example: Standard Online Reporting (1)



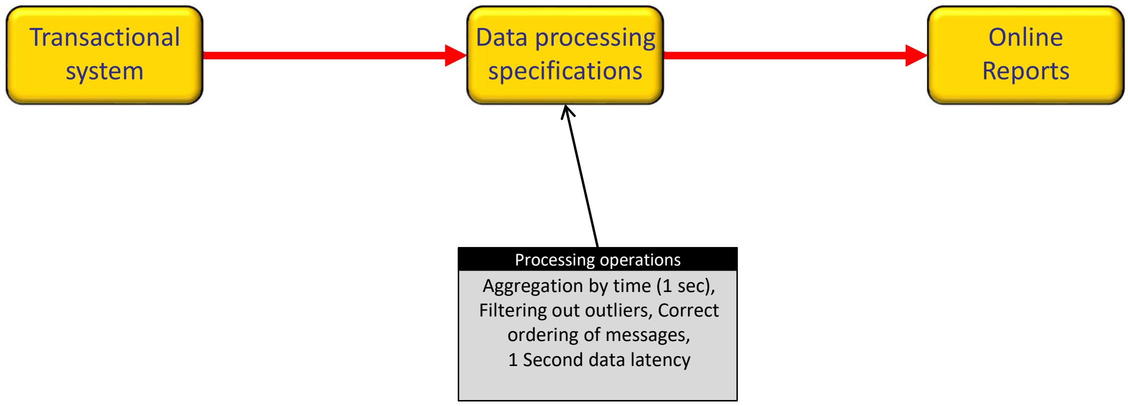
Data Track Example: Standard Online Reporting (2)



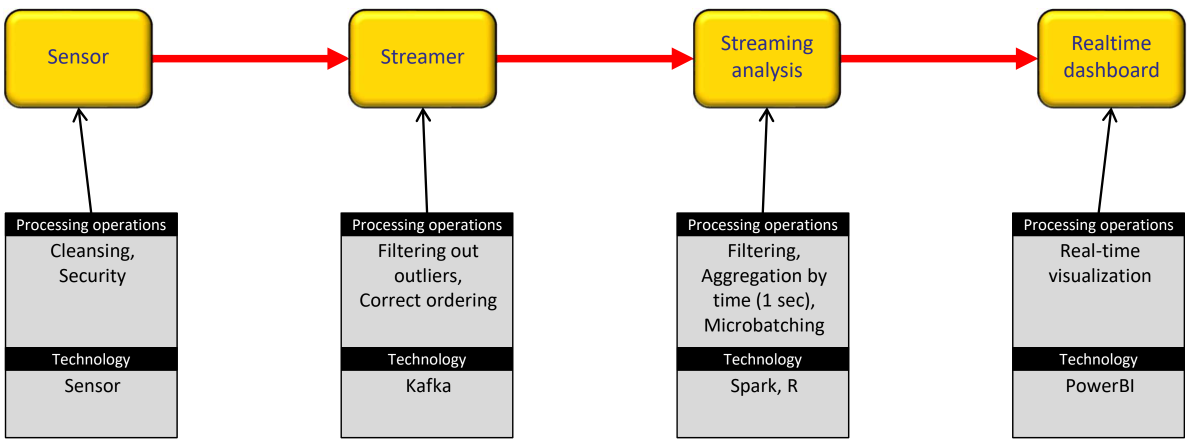
Data Track Example: Standard Online Reporting (3)



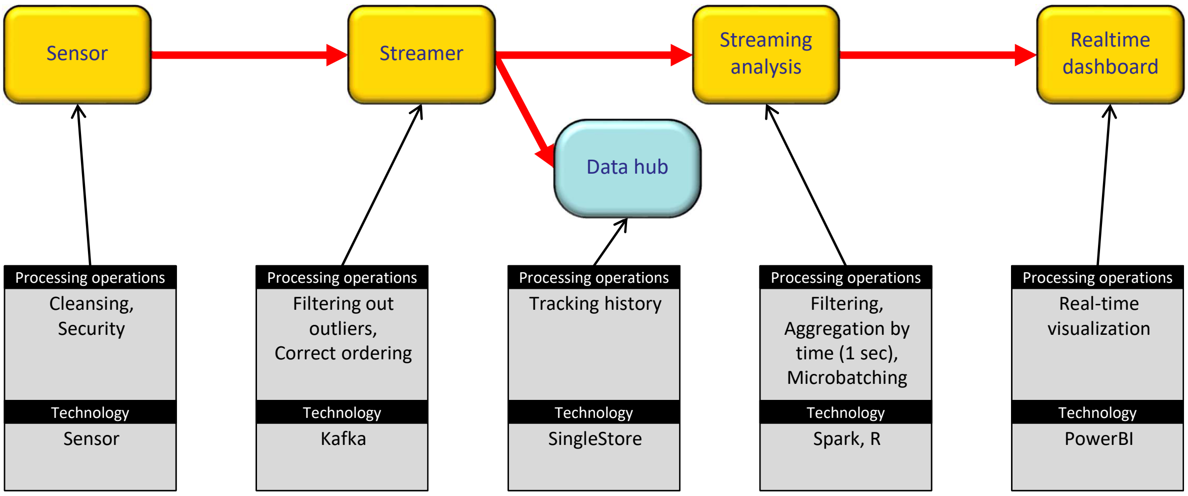
Data Track Example: Streaming Real-time Dashboard (1)



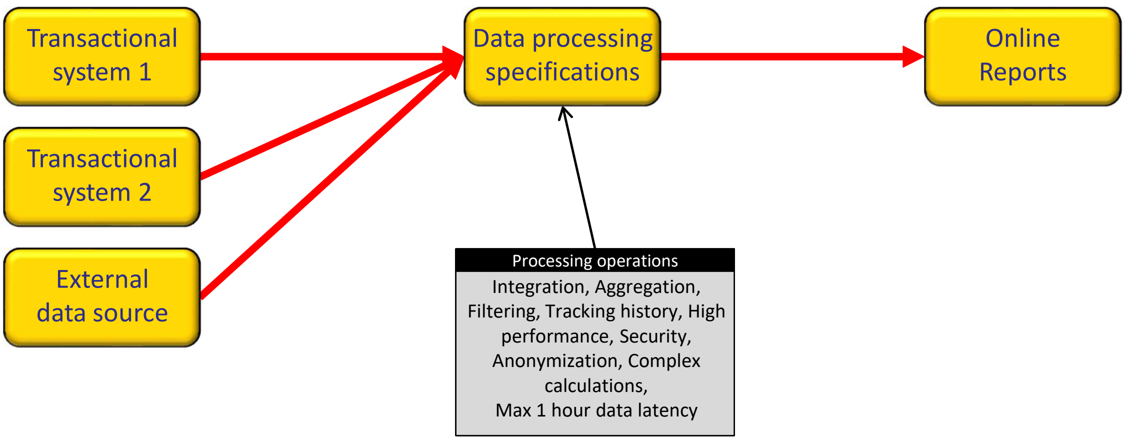
Data Track Example: Streaming Real-time Dashboard (2)



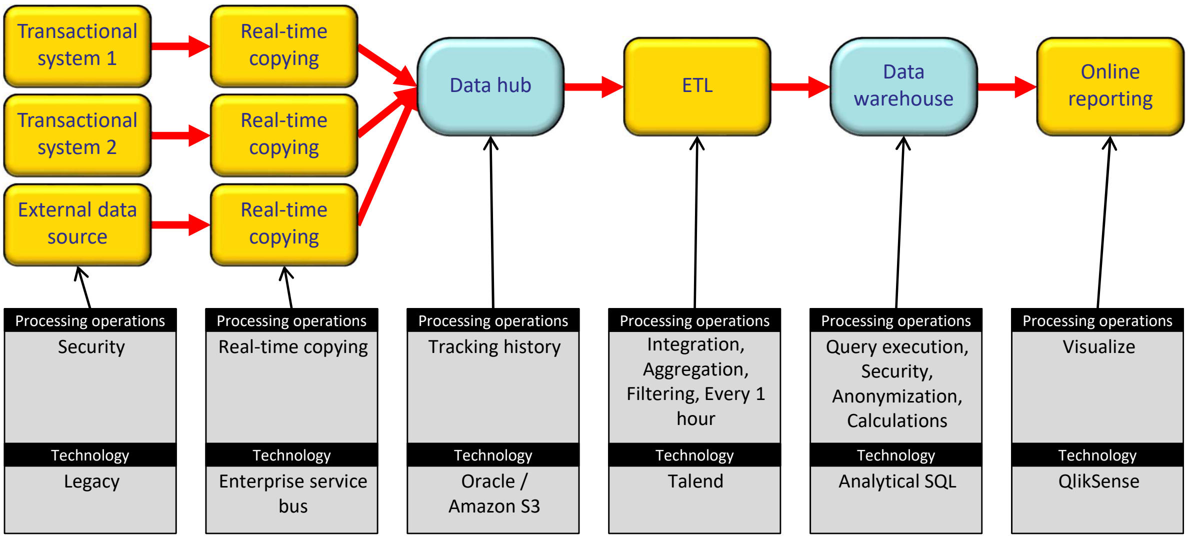
Data Track Example: Streaming Real-time Dashboard (3)



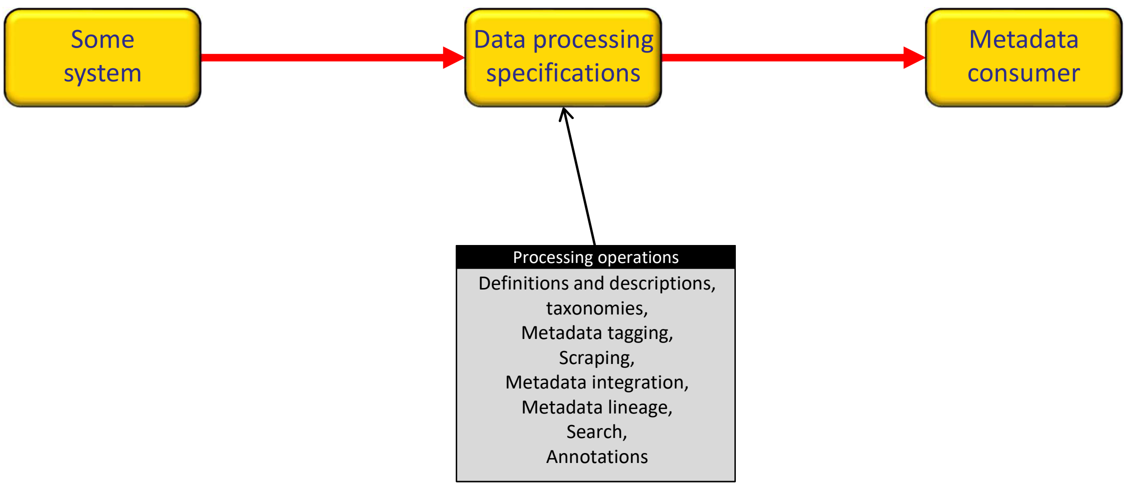
Data Track Example: Integrated Online Reporting (1)



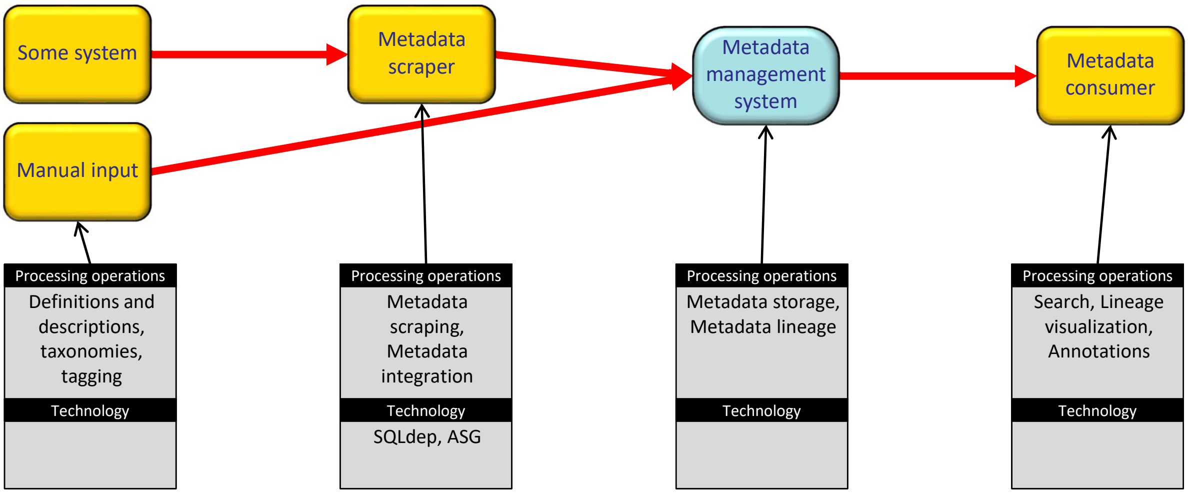
Data Track Example: Integrated Online Reporting (2)



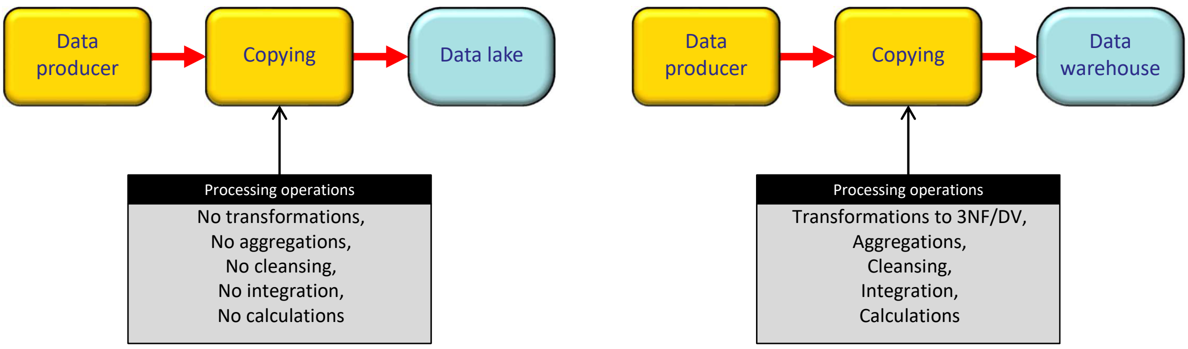
Data Track Example: Metadata Delivery (1)



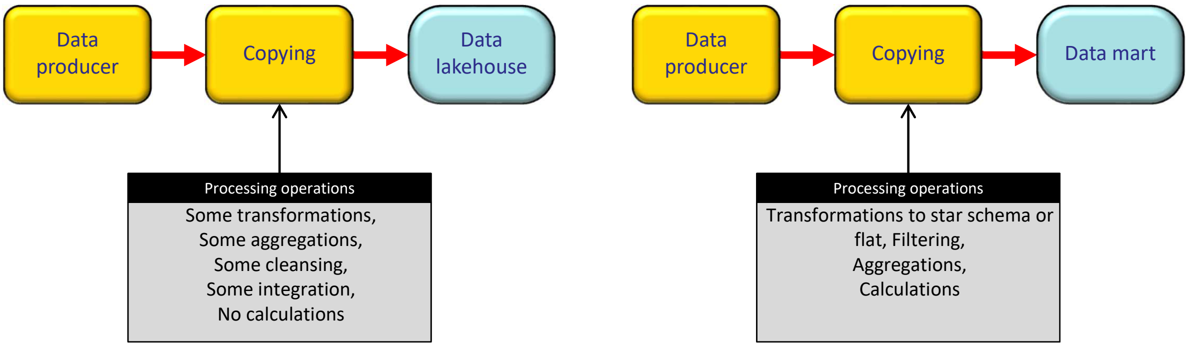
Data Track Example: Metadata Delivery (2)



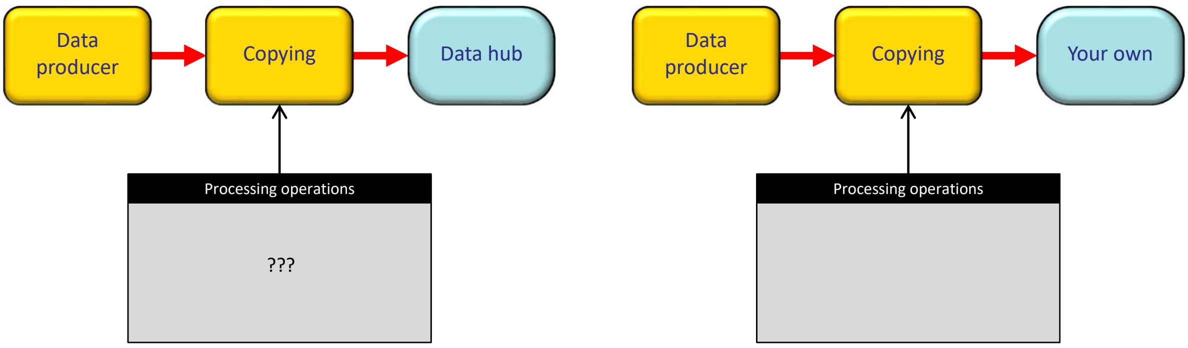
What's in a Name? (1)



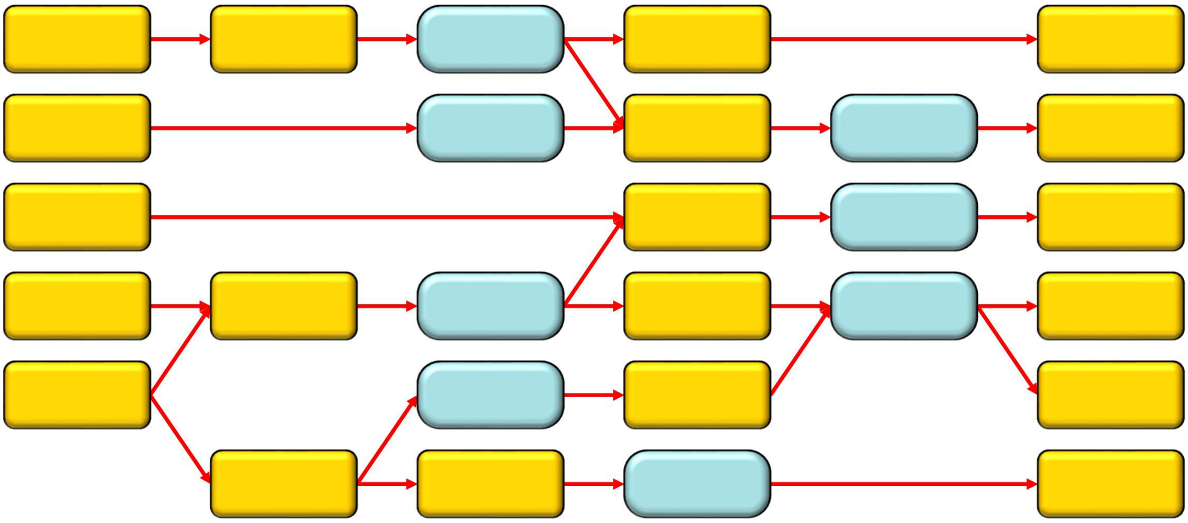
What's in a Name? (2)



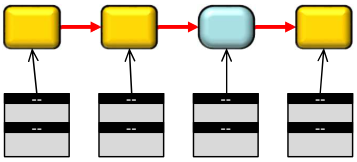
What's in a Name? (3)



High-Level View of the Tracks



Recommendations for Designing Data Tracks

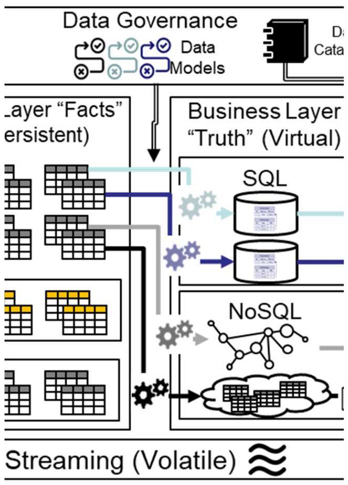


- Design them "backwards" (from consumer to producer)
- Identify data processing specifications first, before assigning of the specs to modules
- One data track can support many comparable data consumers
- Define data track patterns!
  - Architectural design principle?
- Don't solve specific problems

Some More Guidelines for Data Architectures

1. Treasure your data processing specifications
2. Centralize implementation of data processing specifications
3. Centralize technical and business metadata
4. Implement abstraction / decoupling
5. Make plug and play of technology possible
6. Store all data
7. Minimize stored data redundancy - compute over store
8. Choose productivity over performance
9. Minimize design exceptions
10. Implement cross checks
11. Don’t send data, let them get it
12. Source systems responsible for data quality
13. Deploy a holistic design approach

Determine the Intermediate Diagrams



- The current data architecture diagram
- The new data architecture diagram
  - The dream
  - Will never be reached
- The intermediate data architectures
  - The path from current to new
  - Make the steps as small as possible
  - Preferred: Each step leads to business value
- Think big, act small

Part 8:
Steps 9-10: Final Steps

Roadmap for Designing Data Architectures

1. Determine business motivations
2. Determine new requirements
3. Analyze the existing environment
4. Study new products and technologies
5. Define architectural design principles
6. Select a reference data architecture
7. Design the new data architecture
8. Determine the Implementation approach
9. Select new products and technologies
10. Introduce the data architecture within the organization

Part 8.1:
Step 9: Select New Products and Technologies

Developing the Request for Information

Photo: Brett Jordan

- Get access to in-depth technological know-how
  - To ask the right questions
- Use distinguishing questions
- Weighs of requirements – explain why
- Closed questions – easier for comparisons
- Deliver sufficient info to let vendor provide details for pricing and products
- Are extra modules/versions required?
- Remember the new requirements!

Product and Vendor Evaluation (1)

Photo: Clay Banks

- Evaluation of products
  - Features, performance, costs, market share
- Local support and partners
  - Experience?
- Extra software required
  - Master data management for complex integration
  - Data cleansing
  - Database server for reference tables and caches
  - Data security
  - Special connectors/drivers

Product and Vendor Evaluation (2)

Photo: Clay Banks

- Products need to “fit” the architecture
- The intended use cases of the products must match use cases of organization
- Do you need the best tool?
  - Remember Betamax and quadrophonic records
- One-stop shopping or best-of-breed?
  - Minimize number of vendors
  - Never independent of zero vendors

Product and Vendor Evaluation (3)


Photo: Clay Banks

- Standardize for back-end tools
  - If use cases allow
  - Not one BI tool for all forms of data consumption
- Open Source software
  - Open source ≠ Non-proprietary software
  - Standards = Non-proprietary software
  - Study how active the development group is
  - What if open source vendor goes commercial?
    - MySQL, Revolutionary Analytics (R), …

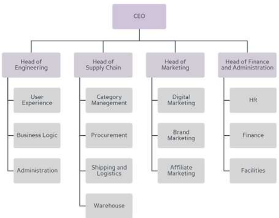
The Proof of Concept/Pilot


Photo: Haupes, Co

- The PoC must be representative of the new system
  - Data: size, characteristics (value distribution, uniqueness), anonymized data?
  - Applications and reports must have a representative complexity
- Performance
  - Multi-user tests
  - Experts required
- Tough SLAs must be tested!
  - Can be expensive
- Invite vendors to install and optimize software themselves
- Select ICT personnel for PoC
  - Developers who enjoy working with new technology and are willing to stay over the weekend

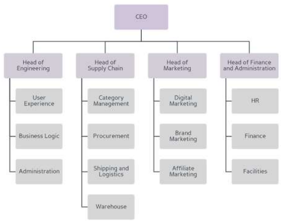
Part 8.2:
Step 10: Introduce the Data Architecture Within the Organization

Introduction Within the Organization (1)



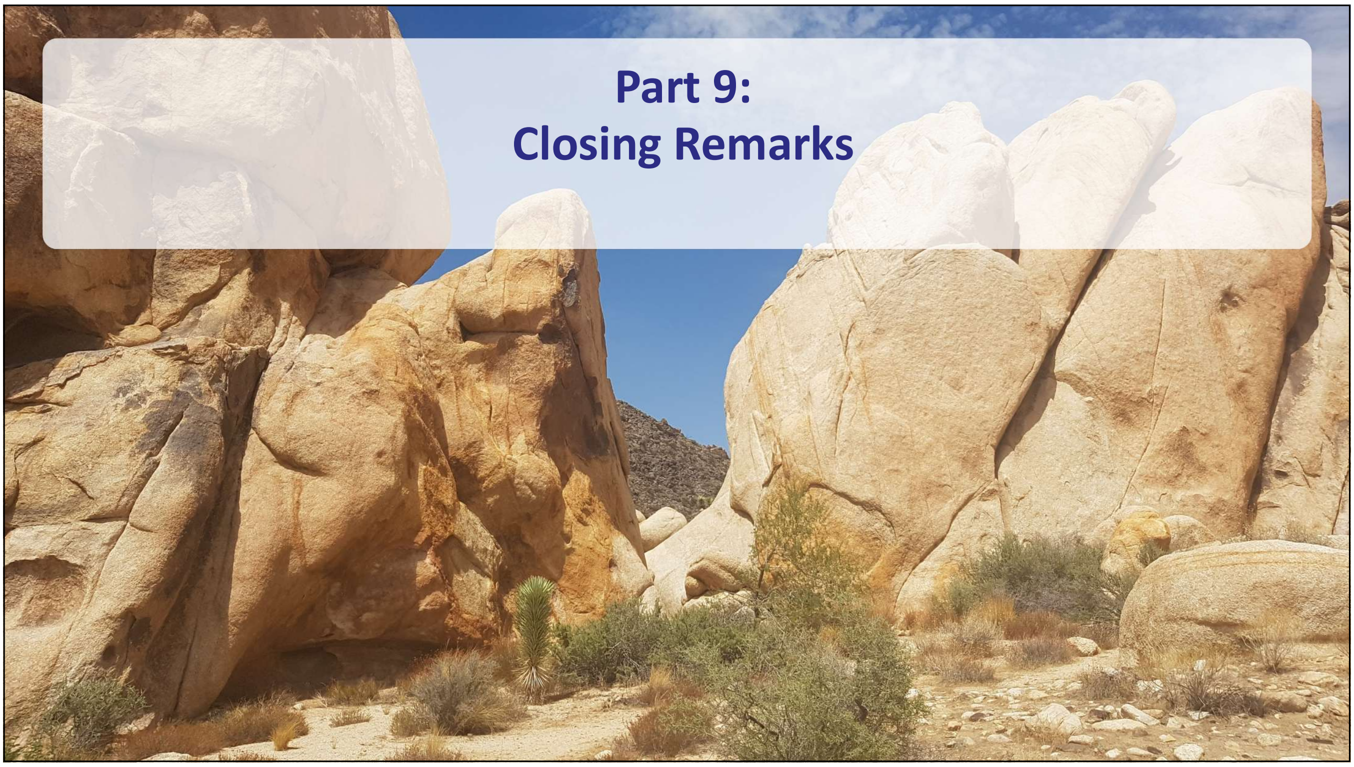
- Three approaches for introduction
  - Via business
  - Via IT bottom up (Trojan horse)
  - Via IT top down
- Identify resistance
  - DBA, source owners, …
- Educational/missionary program for everyone
  - From programmers to C-level management
  - De-mystify
  - Refute the mythical performance problem
  - Sell the new data architecture
  - Find a champion

Introduction Within the Organization (2)



- Impact of new data architecture on organization
  - New roles and new responsibilities, examples
    - New roles related to data stewardship
    - Ownership of data
    - Data science models to support business users cooperating
    - New BI tools
  - Training

Part 9: Closing Remarks

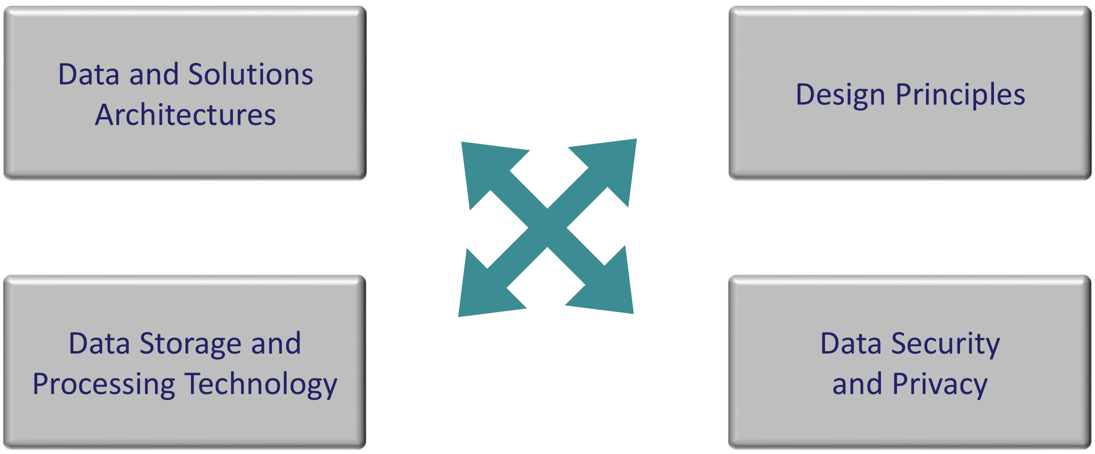


Time to Start!

Photo: Danielle MacInnes

- New data architectures are required
- Focus on data processing specifications, before drawing the storage “boxes”
- Architects must be familiar with the strengths, weaknesses, and use cases of data storage and processing technologies
  - Without this knowledge:
    - Unnecessarily complex architecture
    - Incorrect use of technology
    - Not able to use the full power of a technology
- Design guidelines impact architecture
- Design a data architecture from source to insight

From a Linear to a Holistic Approach




That's all Folks!