Welcome to the Online Architect Workshop Data Fabric/Mesh Architecture

Piethein Strengholt – ABN AMRO
Siddharth Rajagopal – Informatica

ABN·AMRO

Informatica™

Agenda

Introduction & Housekeeping

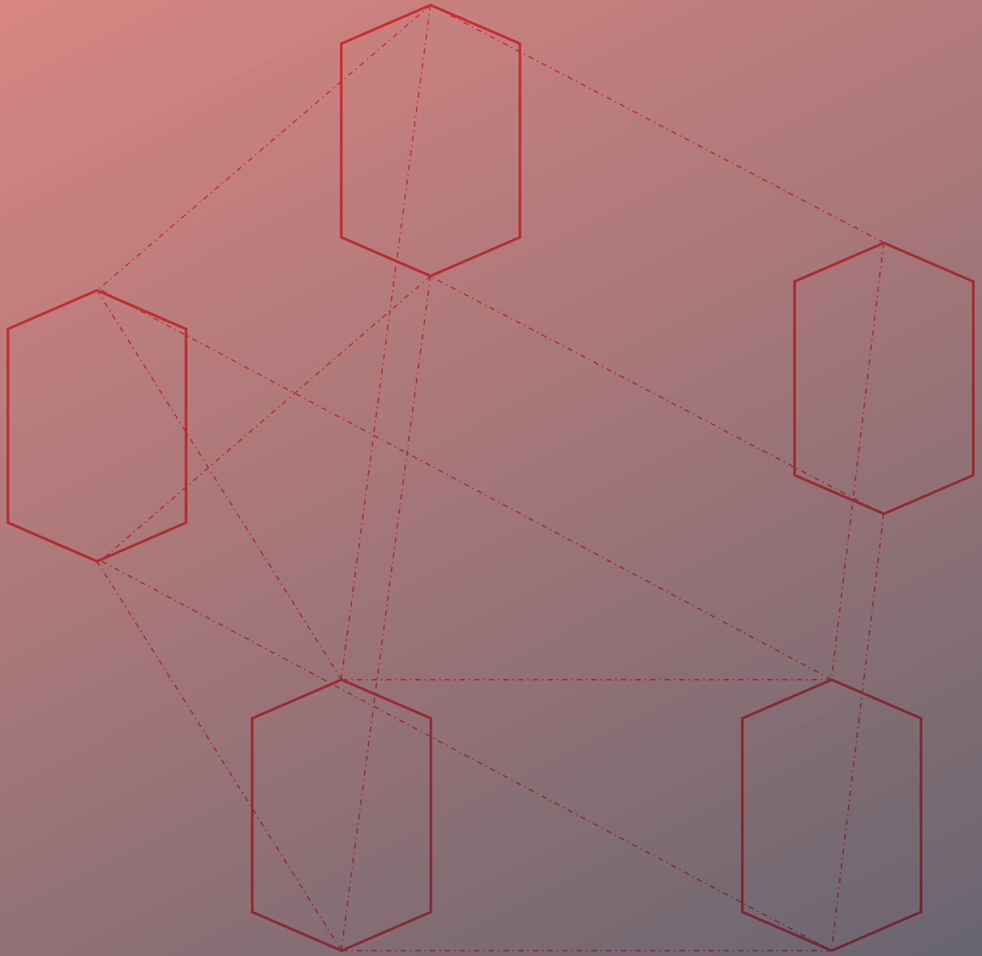
Presentation Data Mesh Architecture

Presentation ABN AMRO’s Integration Architecture

Q&A

Closing

DATA MESH /FABRIC



Sidd Rajagopal

Solution Architect - Informatica

Agenda

Why?

Why do Organizations need to consider a Data Mesh/Fabric?

What

What are some of the key components in a Data Mesh/Fabric?

How?

How do we go about designing & architecting?

Agenda

Why?

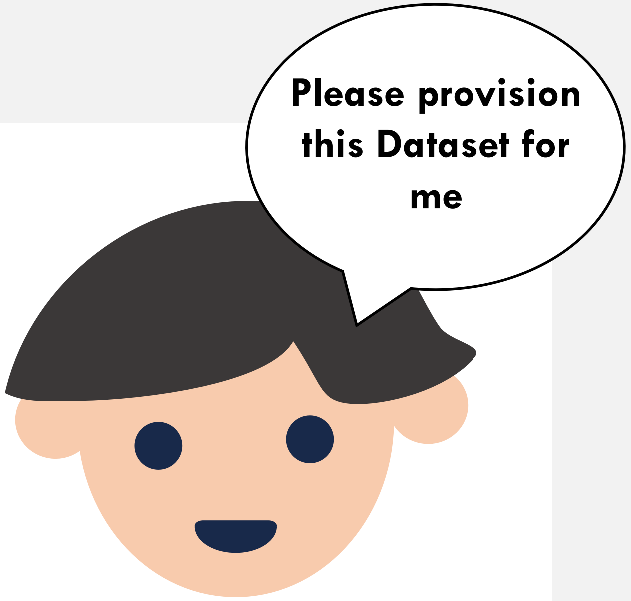
Why do Organizations need to consider a Data Mesh/Fabric?

Why?

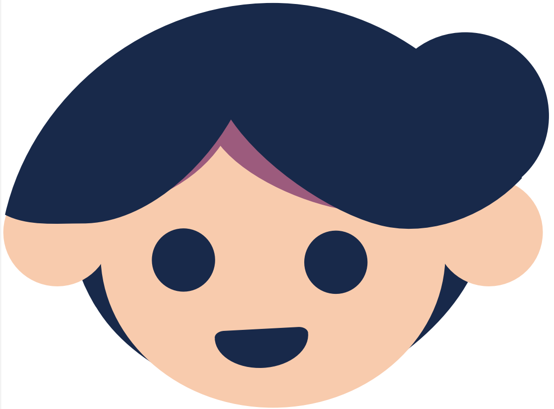
Chat Time – Type in one/two words your current Data challenges!

Why?

Desired Analysis Dataset



User



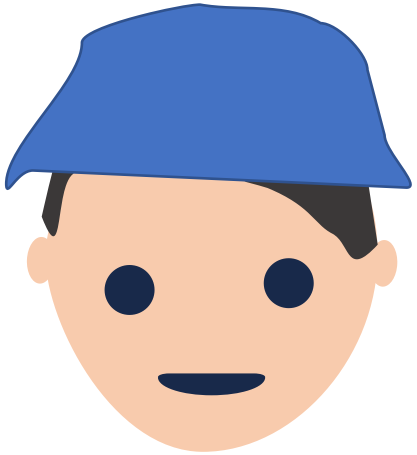
Data Team

Why?

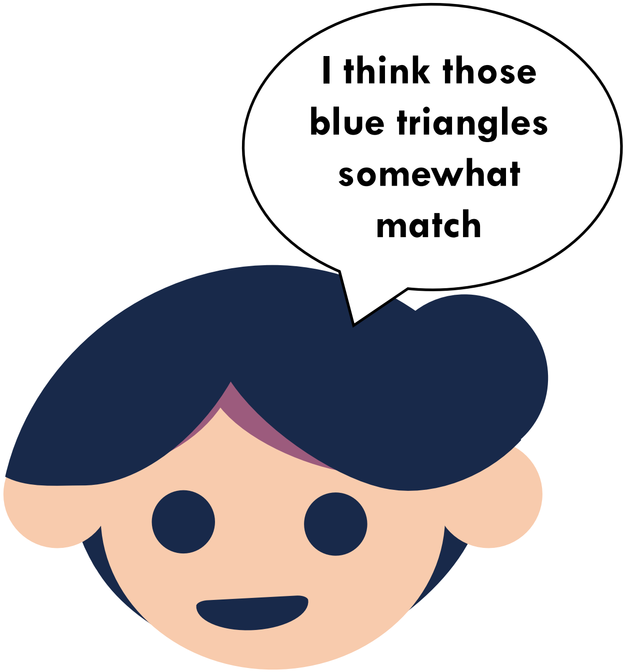
Desired Analysis Dataset

Why?

Source Systems

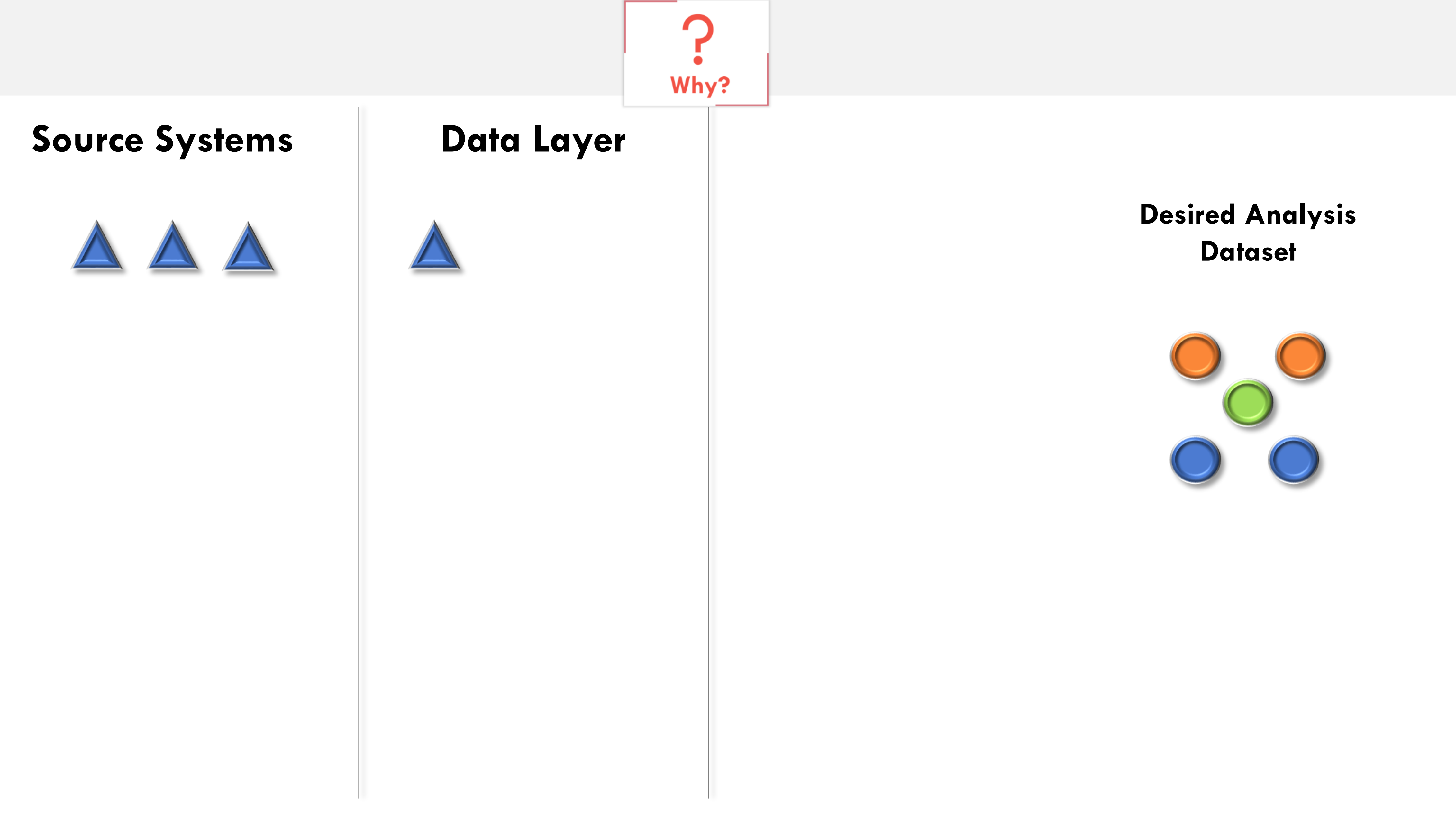


Source Systems Team



Data Team

Desired Analysis Dataset



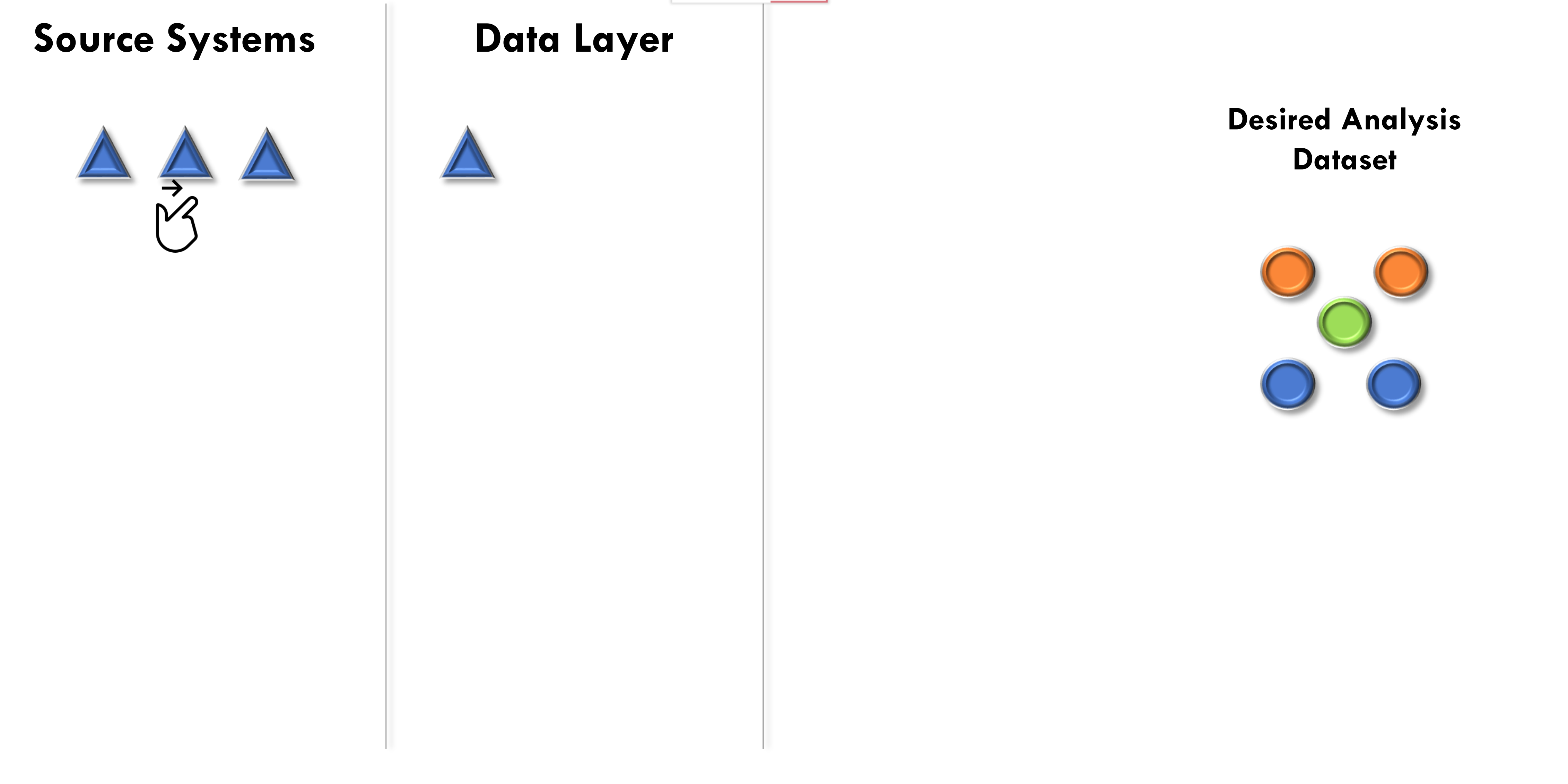
Why?

Source Systems

Data Layer

Desired Analysis Dataset

Why?

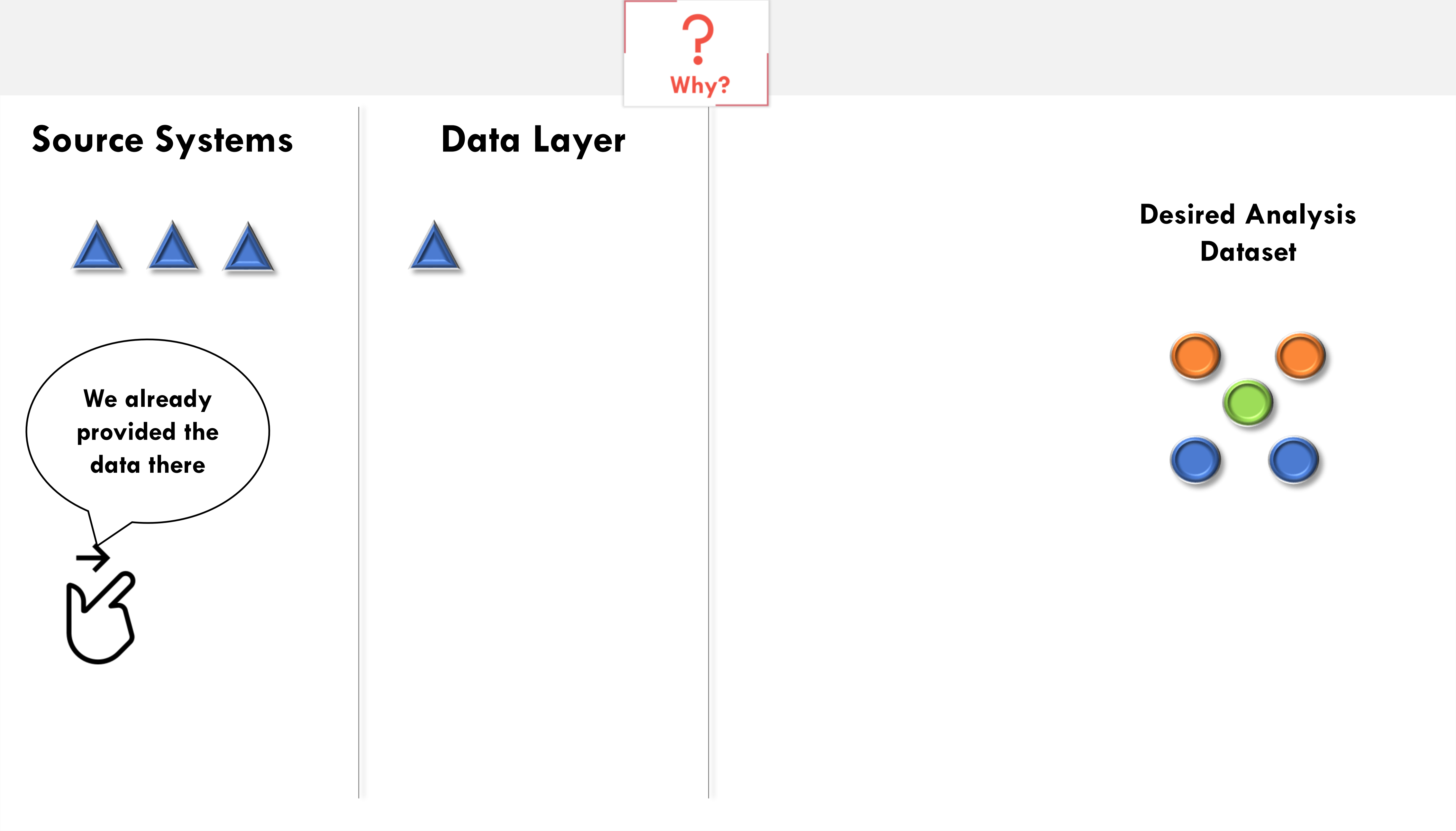


Source Systems

Data Layer

Desired Analysis Dataset

Why?



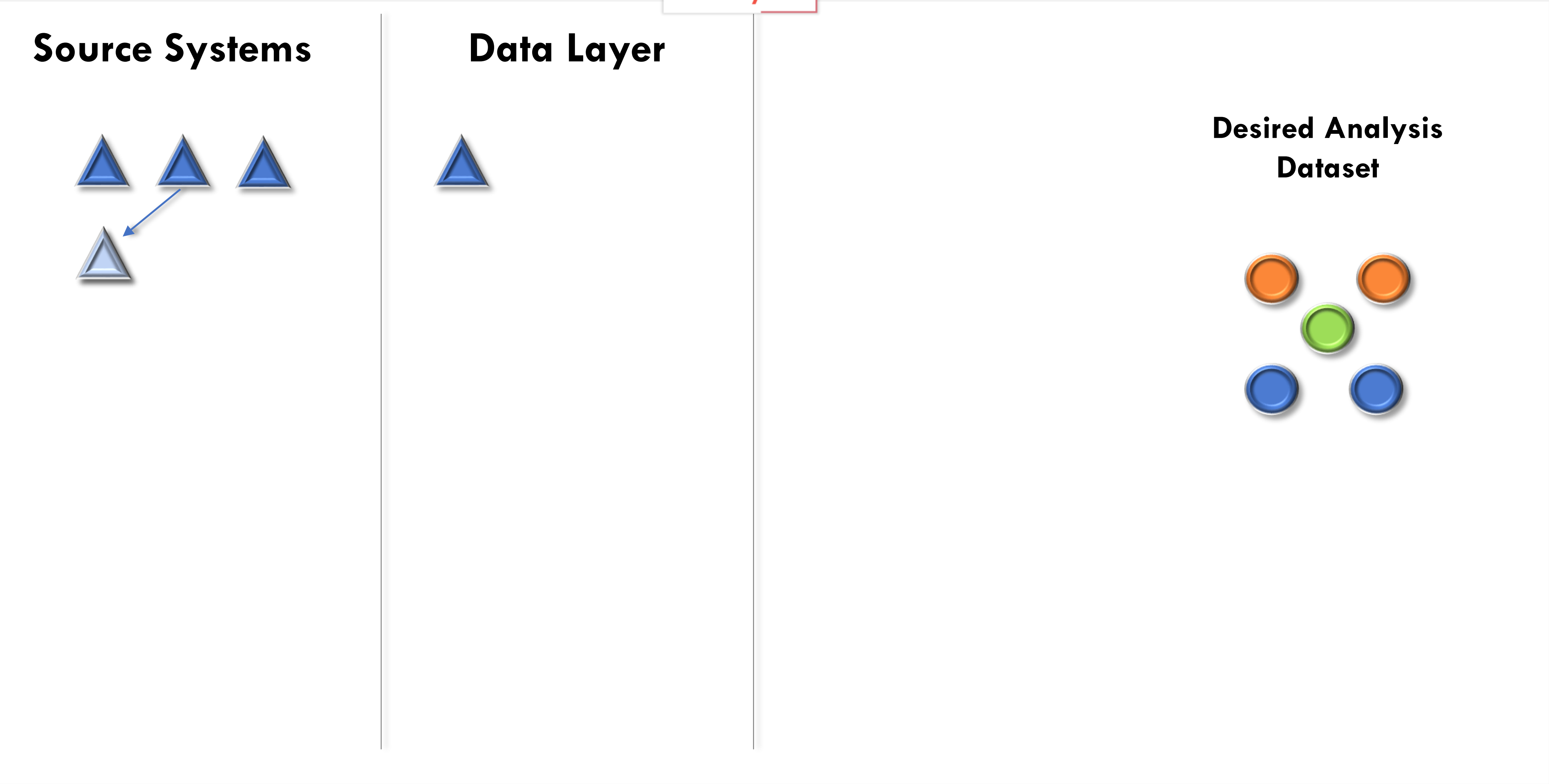
Source Systems

Data Layer

Desired Analysis Dataset

We already provided the data there

Why?

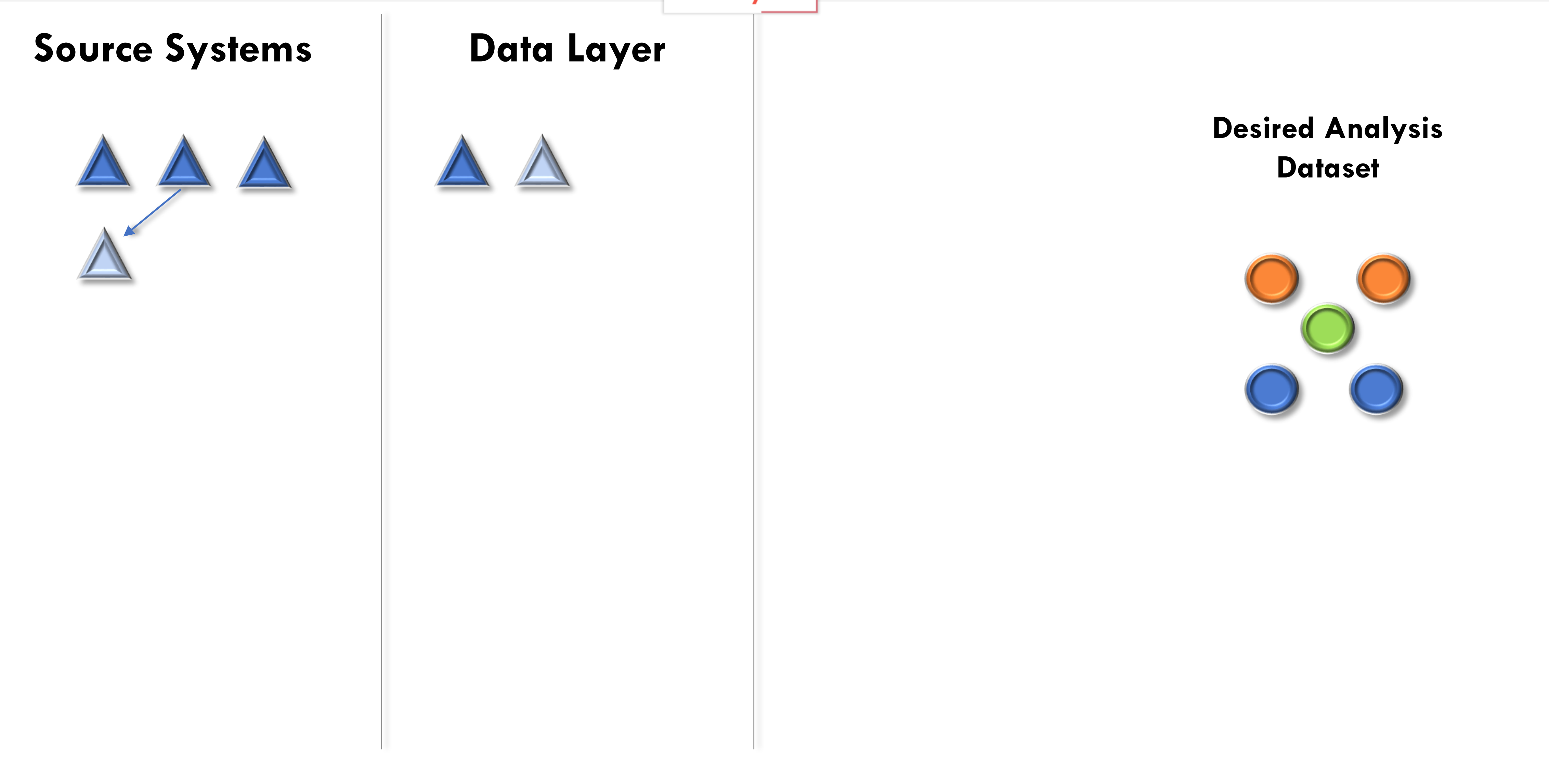


Source Systems

Data Layer

Desired Analysis Dataset

Why?

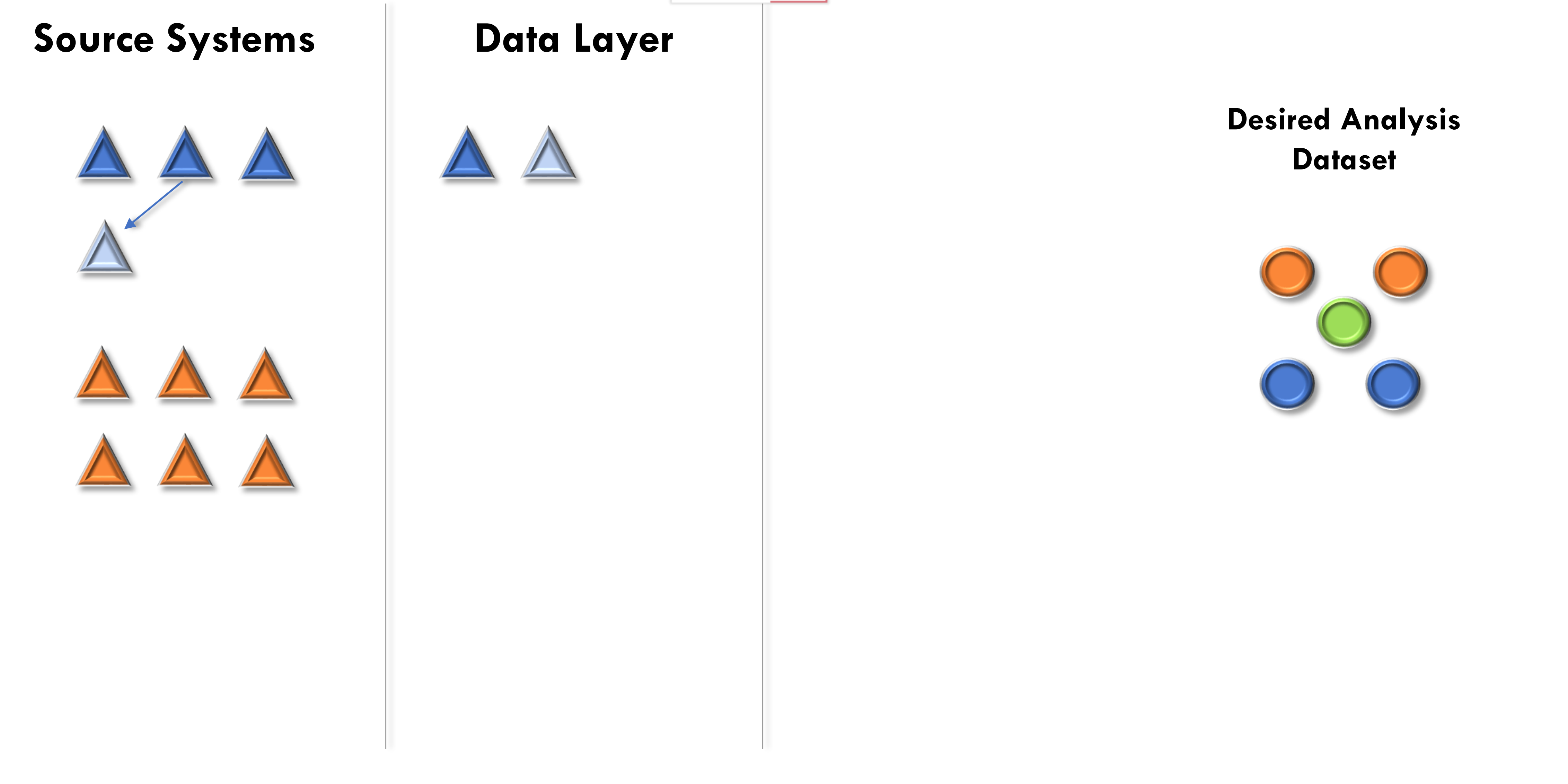


Source Systems

Data Layer

Desired Analysis Dataset

Why?



Source Systems

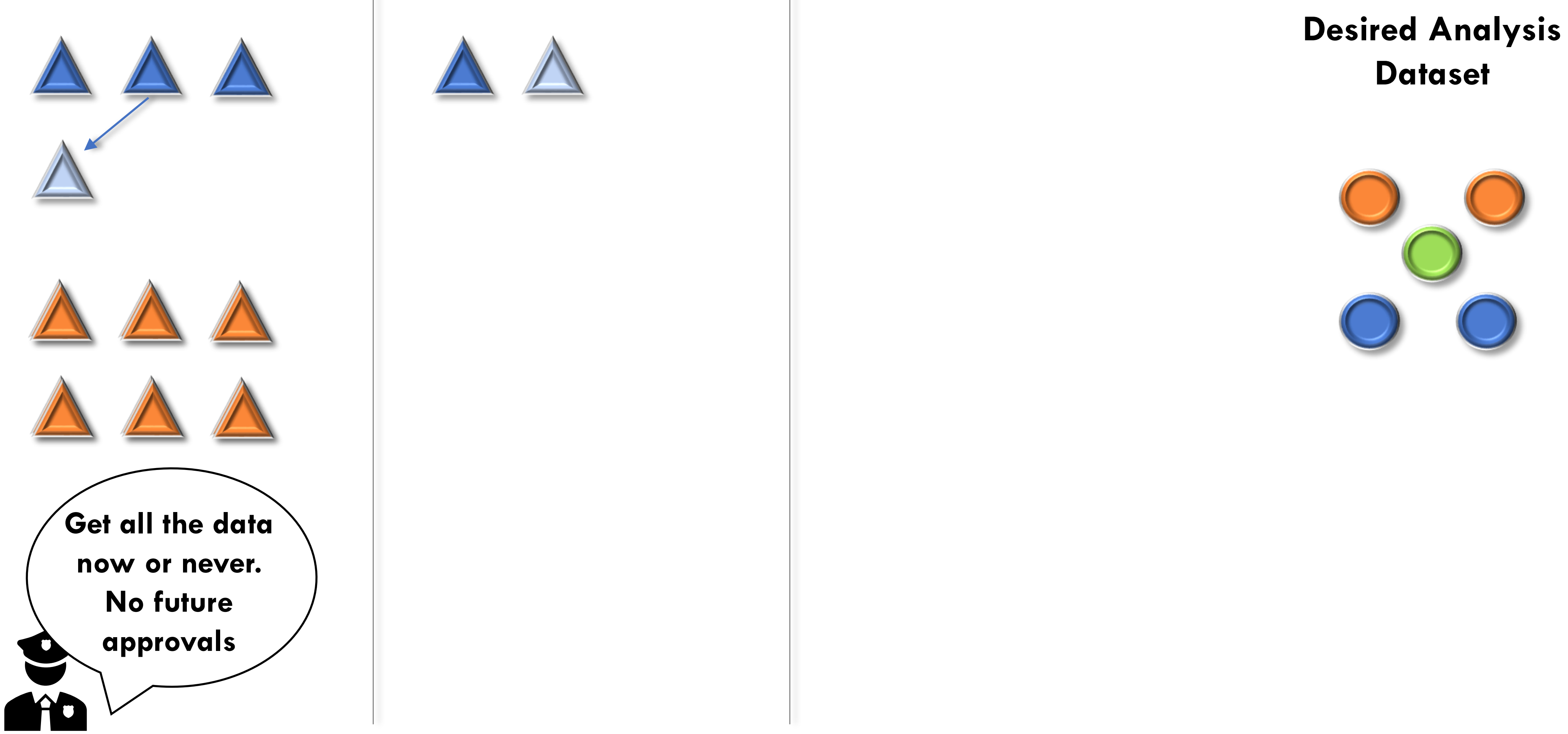
Data Layer

Desired Analysis Dataset

Why?

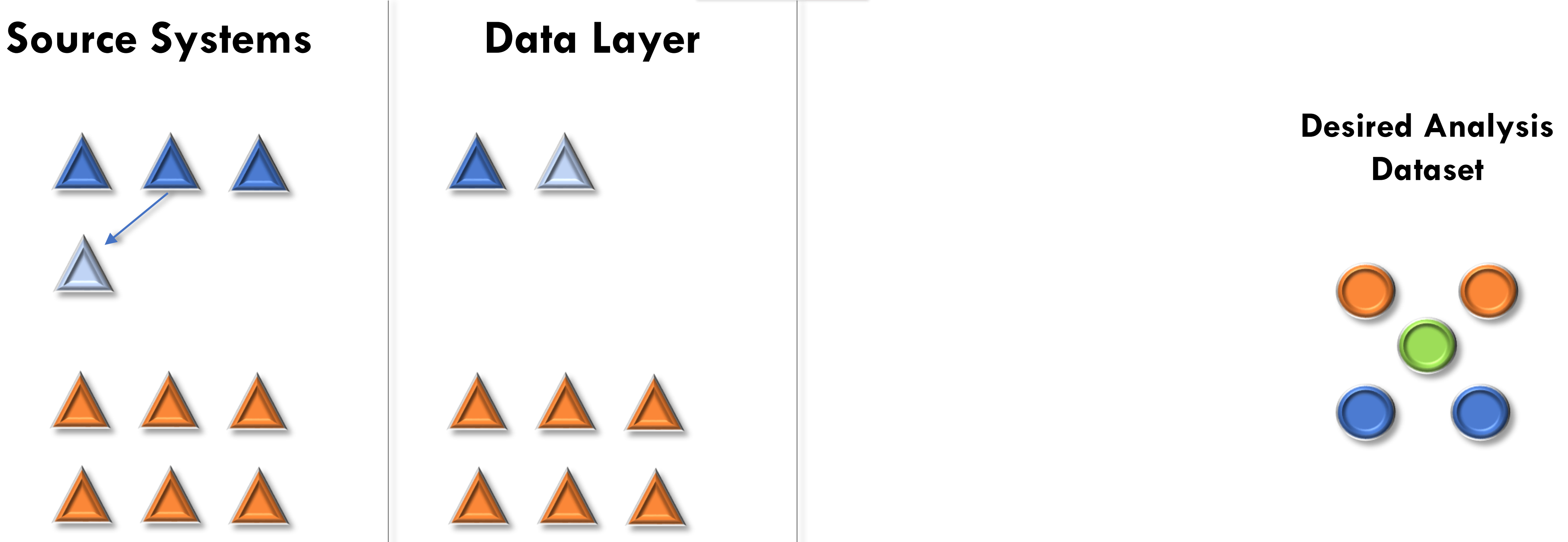
Source Systems

Data Layer



Desired Analysis Dataset

Why?

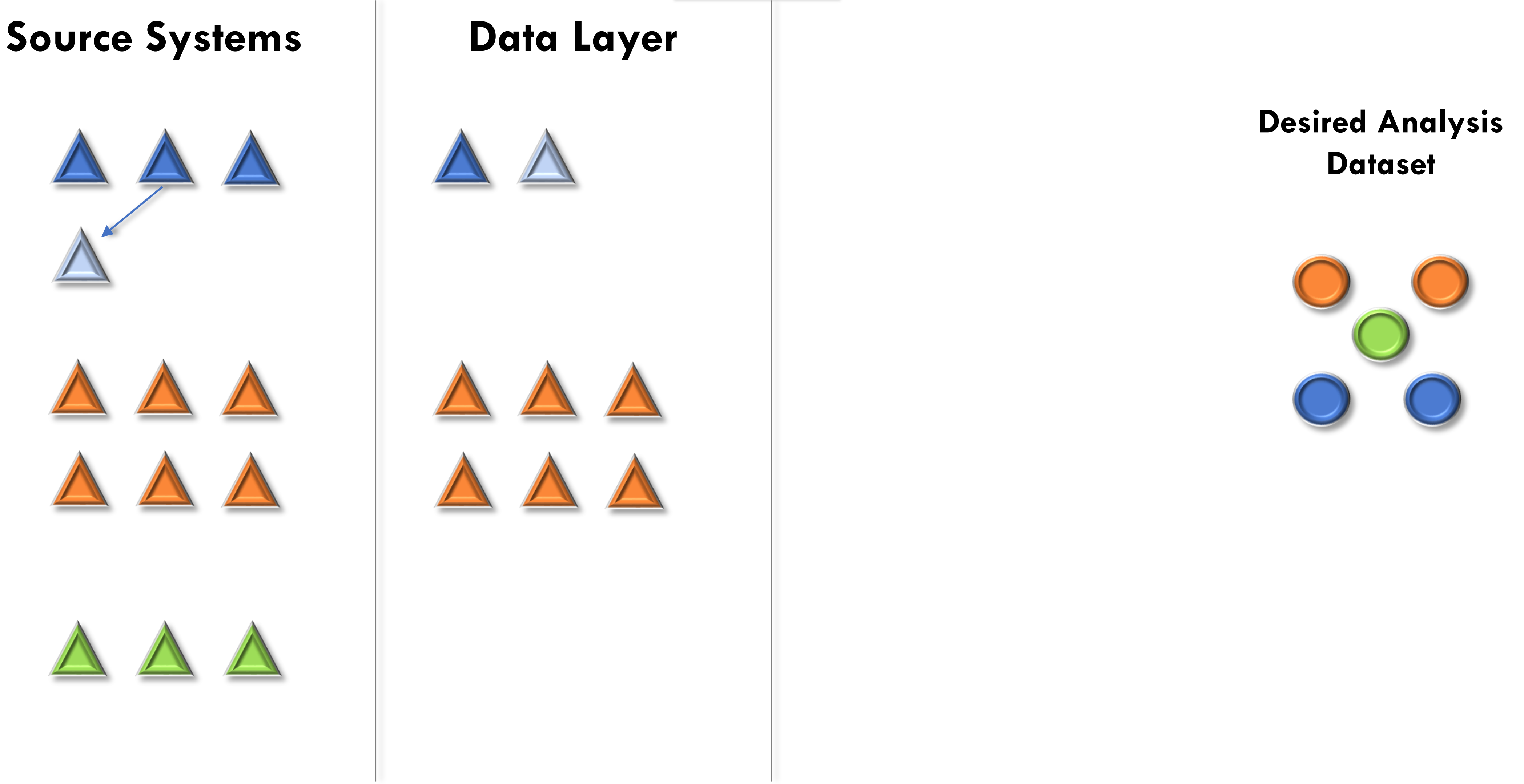


Source Systems

Data Layer

Desired Analysis Dataset

Why?



Source Systems

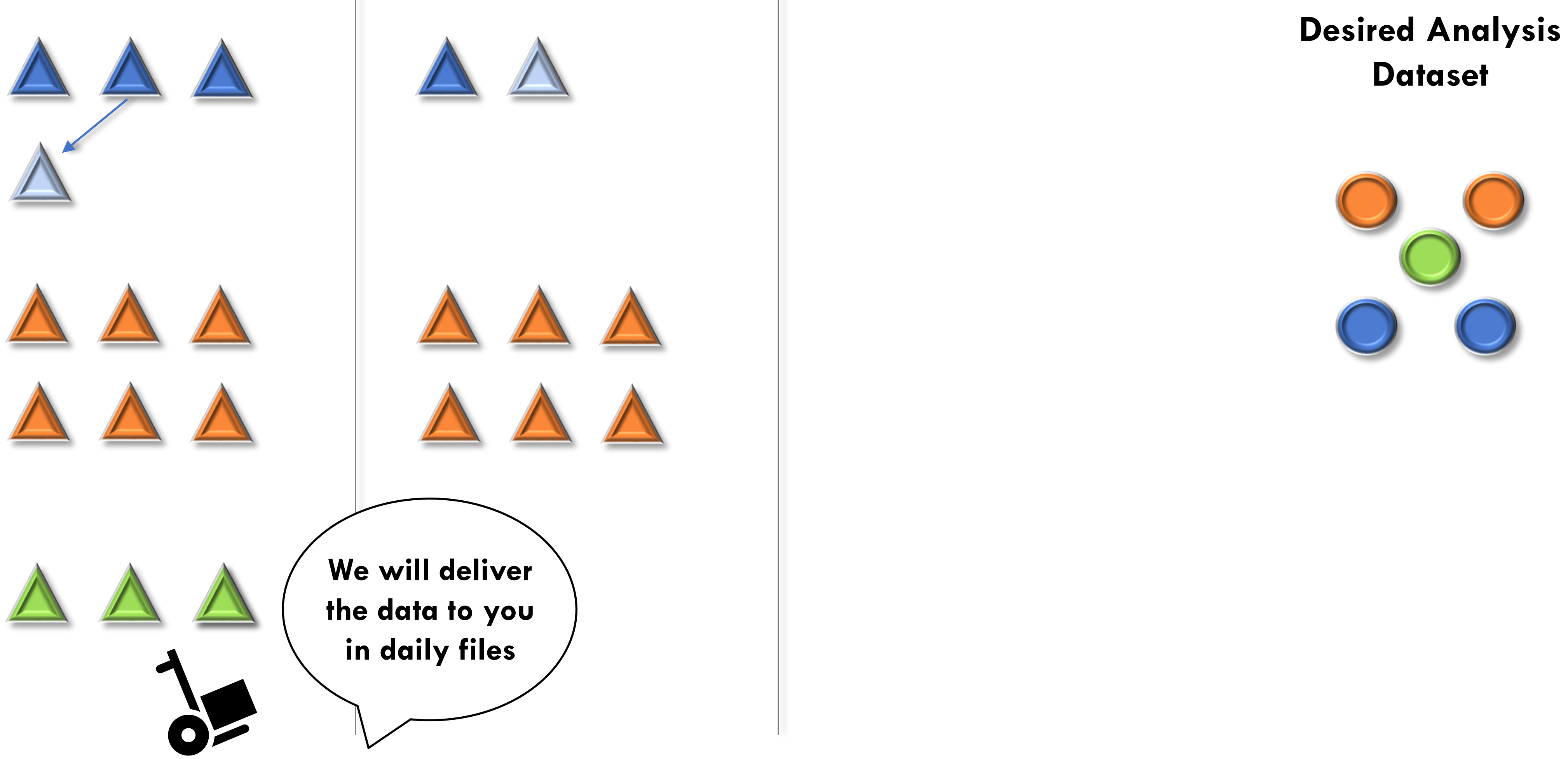
Data Layer

Desired Analysis Dataset

Why?

Source Systems

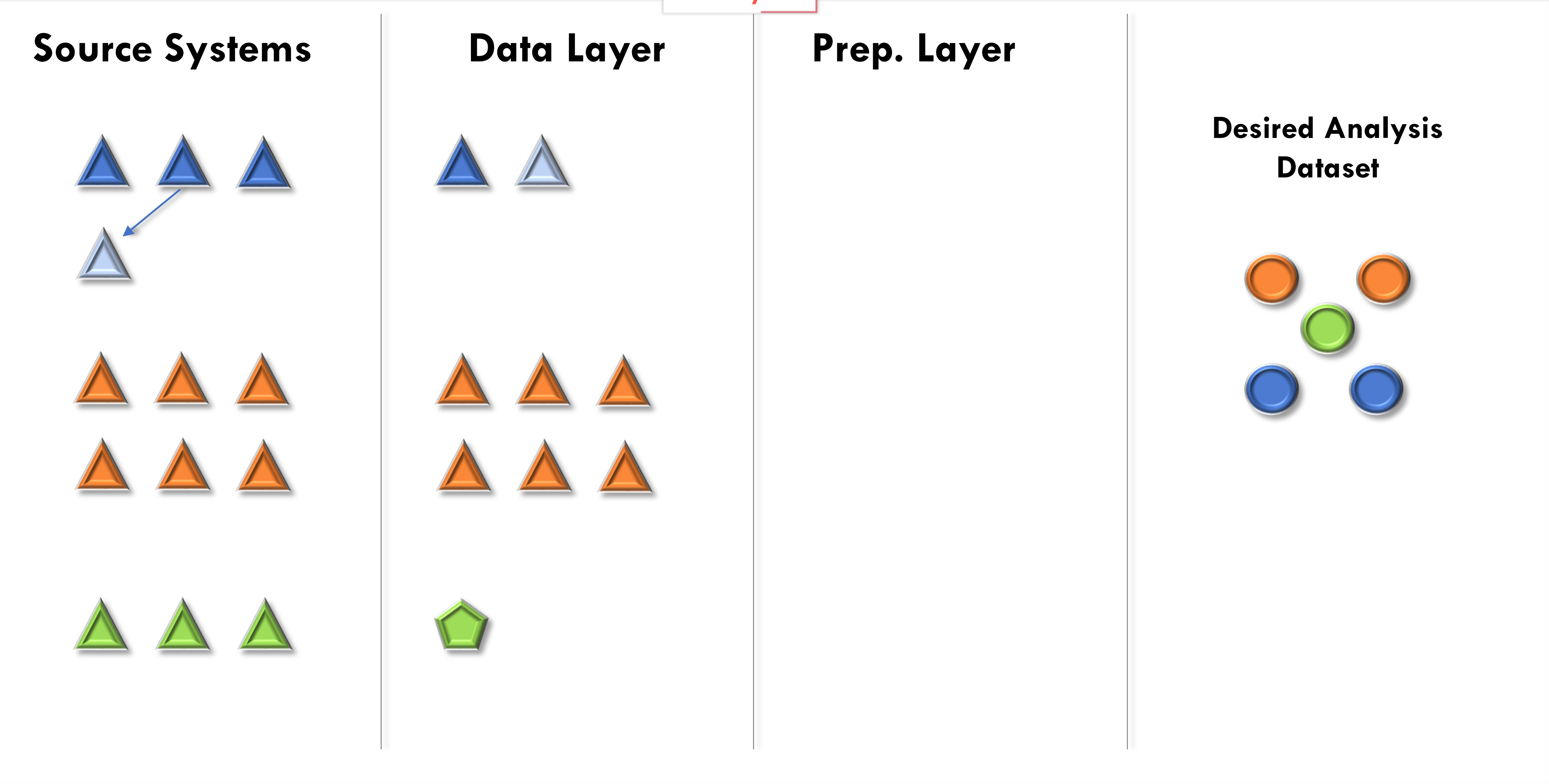
Data Layer



Desired Analysis Dataset

We will deliver the data to you in daily files

Why?



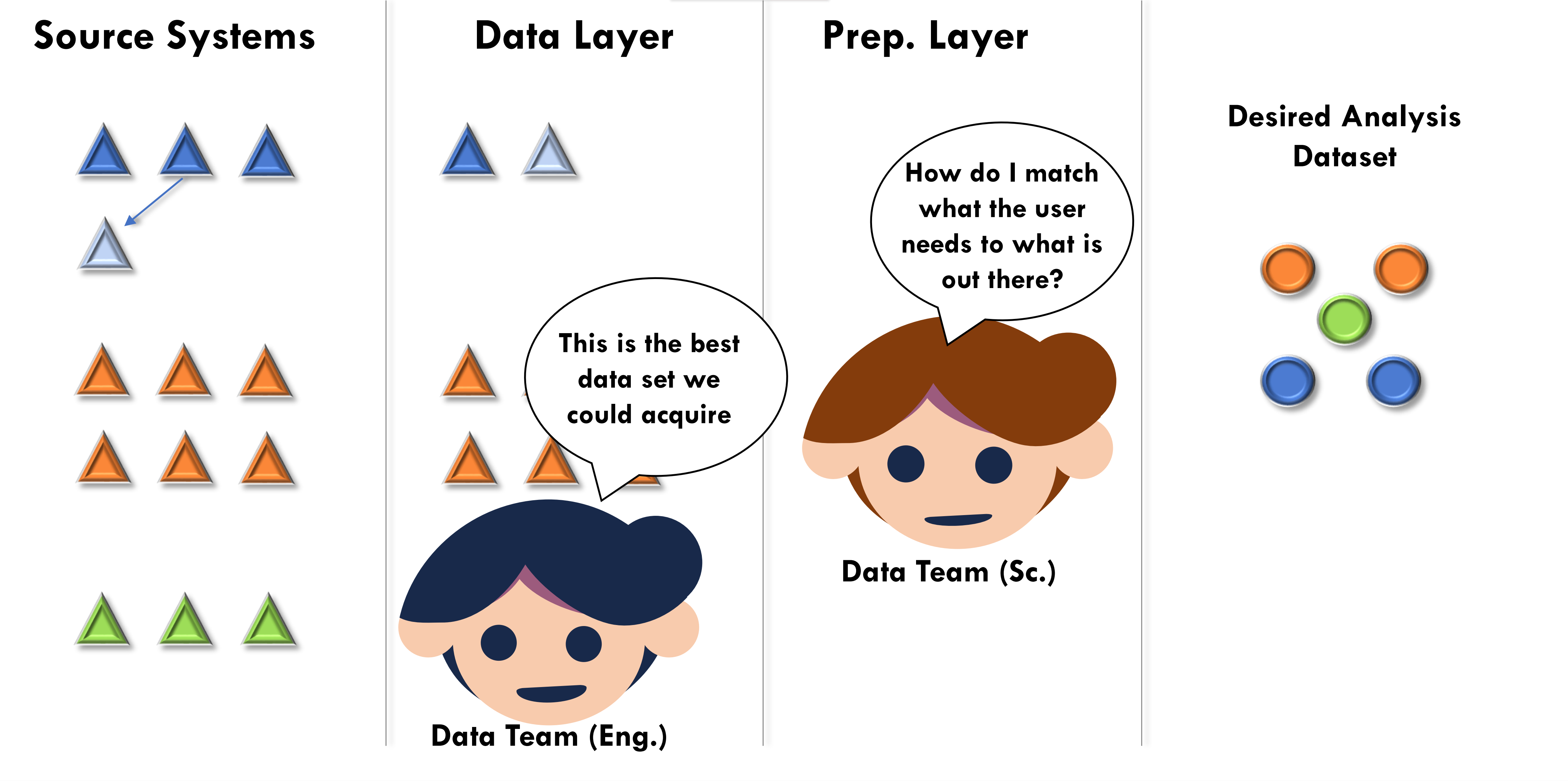
Source Systems

Data Layer

Prep. Layer

Desired Analysis Dataset

Why?



Source Systems

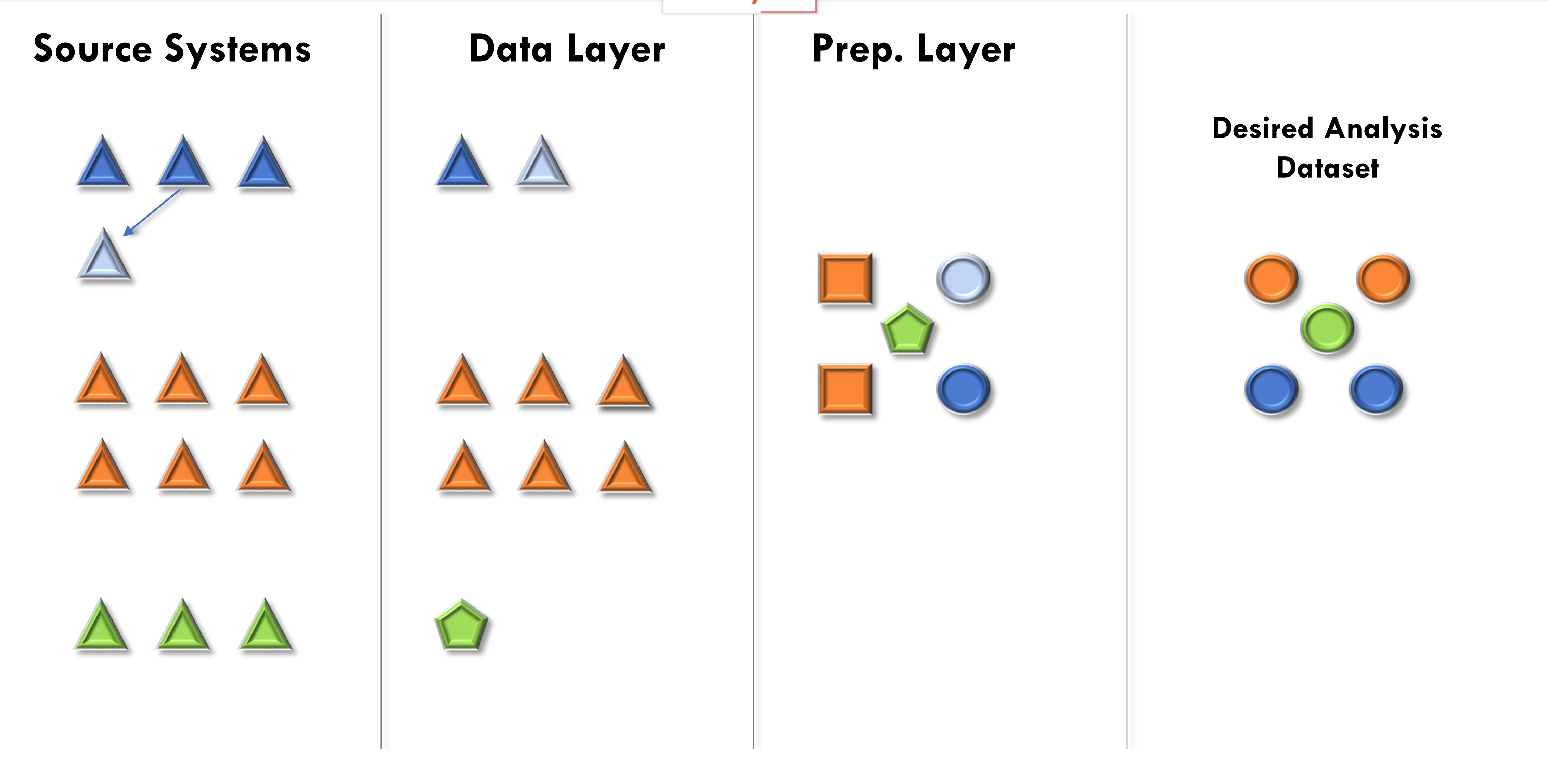
Data Layer

Prep. Layer

Desired Analysis Dataset

This is the best data set we could acquire
Data Team (Eng.)
How do I match what the user needs to what is out there?
Data Team (Sc.)

Why?

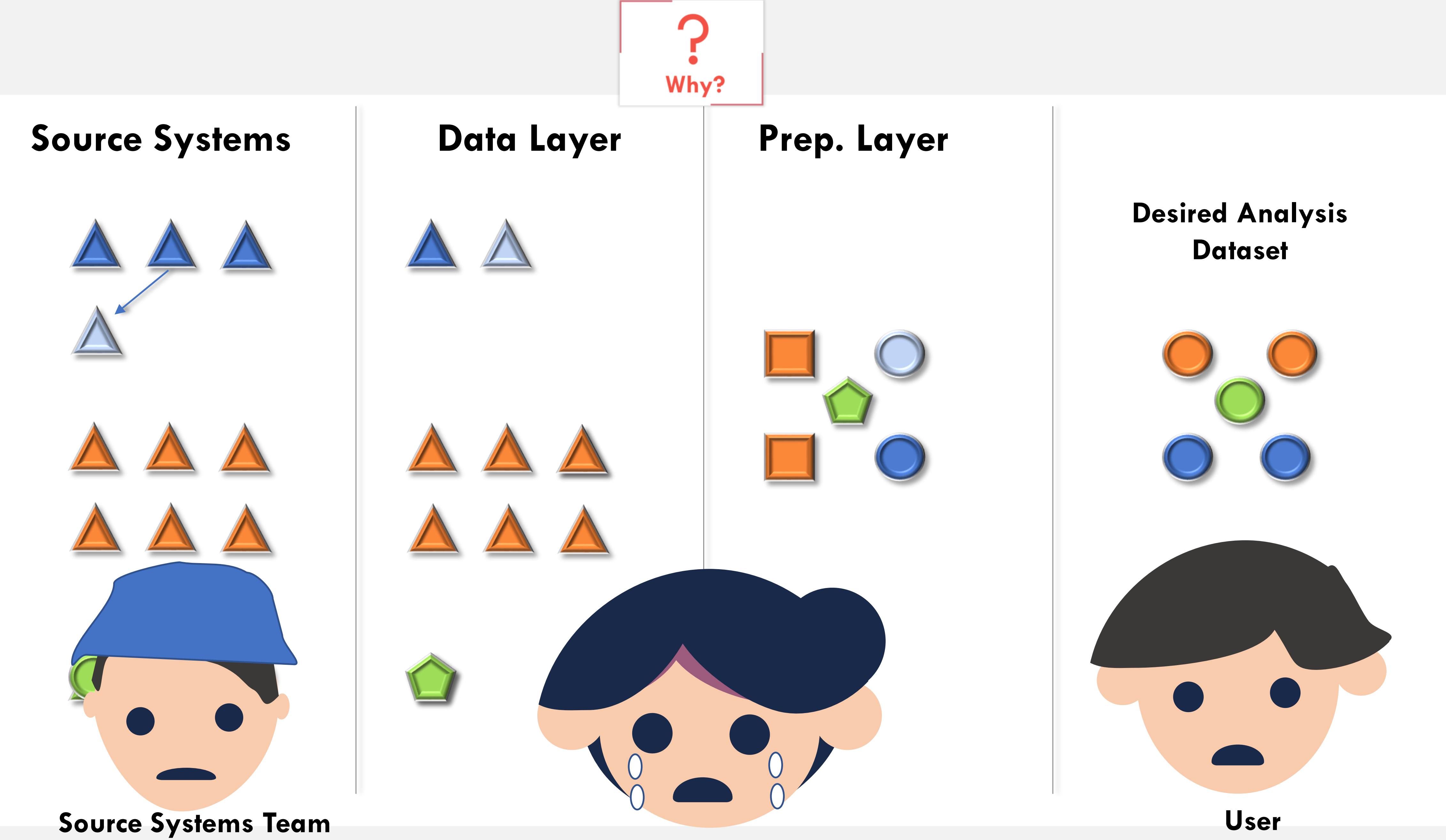


Source Systems

Data Layer

Prep. Layer

Desired Analysis Dataset



Why?

Source Systems

Data Layer

Prep. Layer

Desired Analysis Dataset

Source Systems Team

User

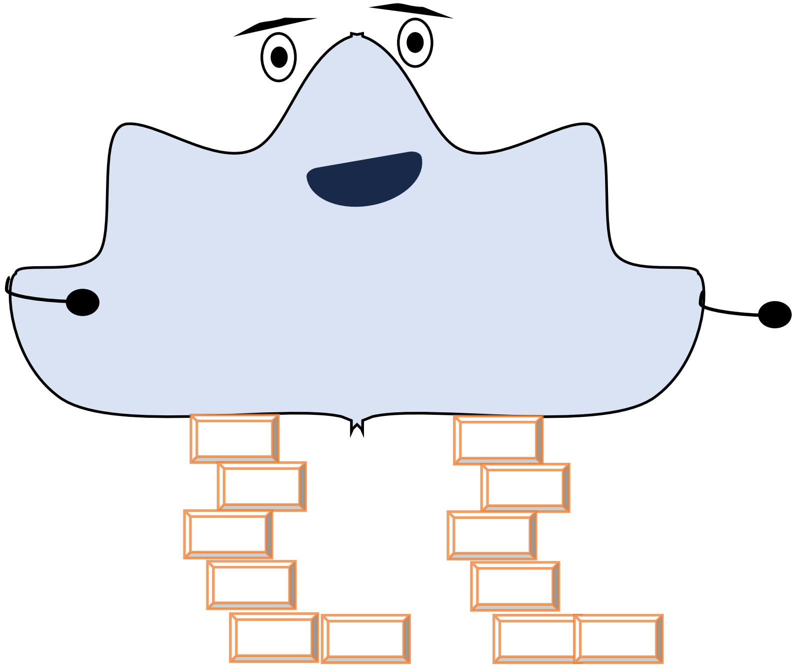
Data Team

Why?

But we think we have a Grand Solution!!!

Why?

Let's go to the Cloud



But we think we have a Grand Solution!!! Not really

Why?

Why?

Why?

Chat Time – Type in one/two words your current Data challenges!

Summary of Why?

Speed to Market

Time to respond to new Analytical Needs

1

Pseudo Teams

Business IT Teams and Solutions created as a side-effect

3

Centralized Bottleneck

Dependency on a Central Team of Data Engineers/Architects

2

Technology Focus

Focus on Technology/Data Layers than Business Domains and Usecases

4

Agenda

Why?

Why do Organizations need to consider a Data Mesh/Fabric?

What

What are some of the key components in a Data Mesh/Fabric?

How?

How do we go about designing & architecting one?

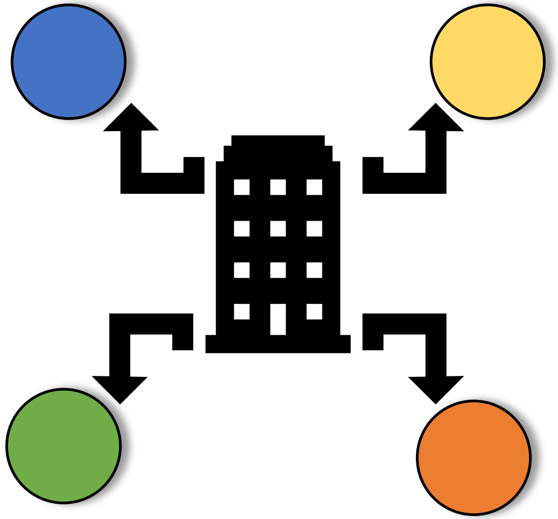
Agenda

What

What are some of the key components in a Data Mesh/Fabric?

What

Key Components



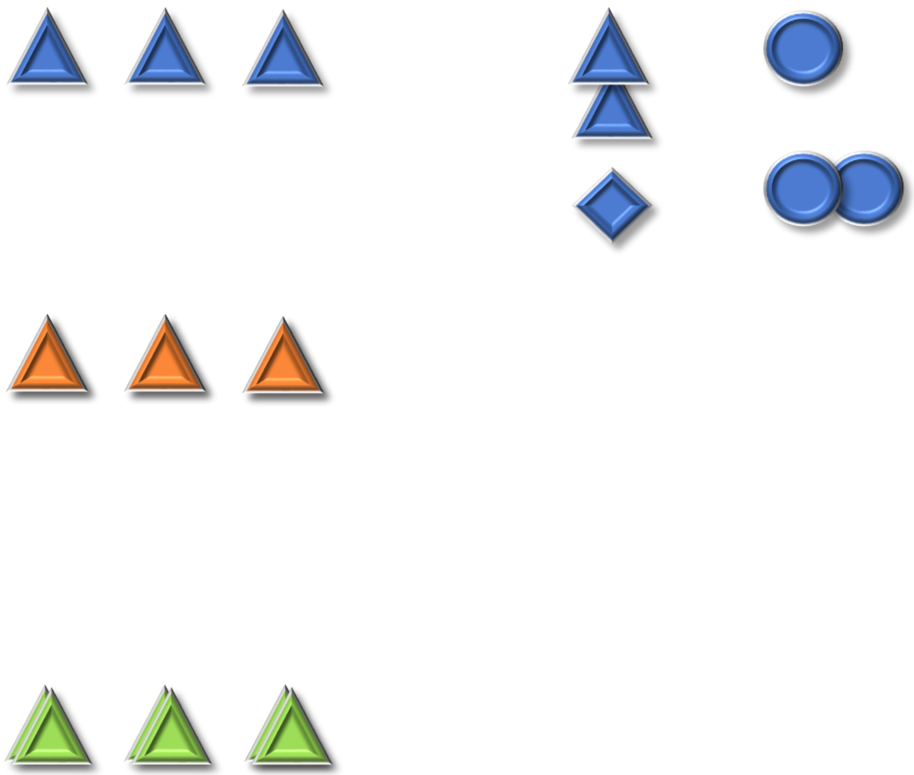
Domain Driven Design

What

Domain Driven

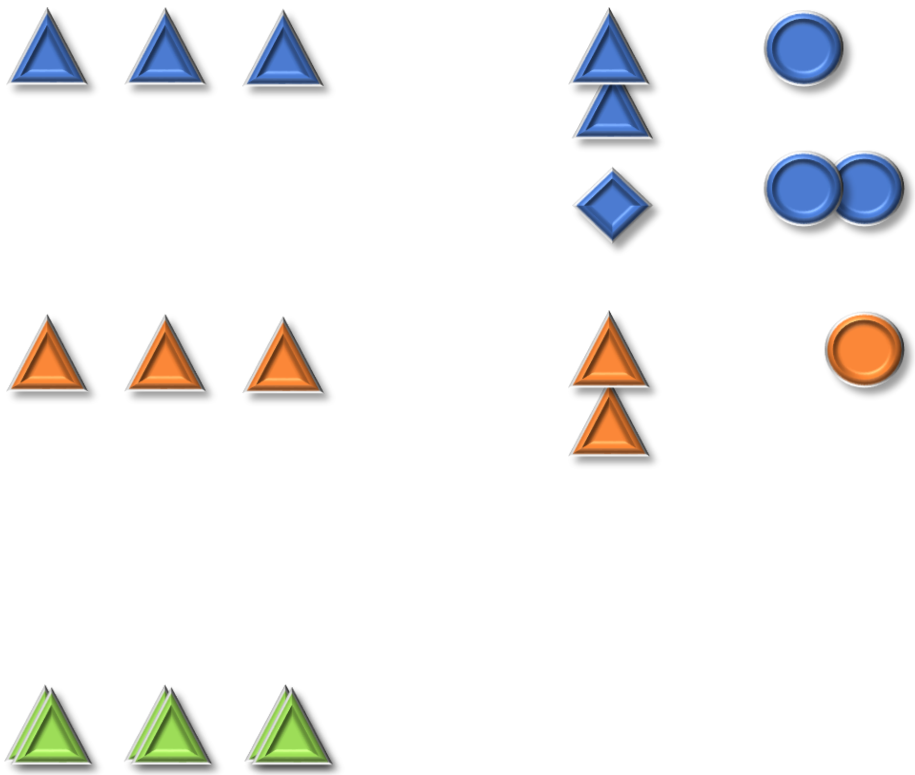
What

Domain Driven



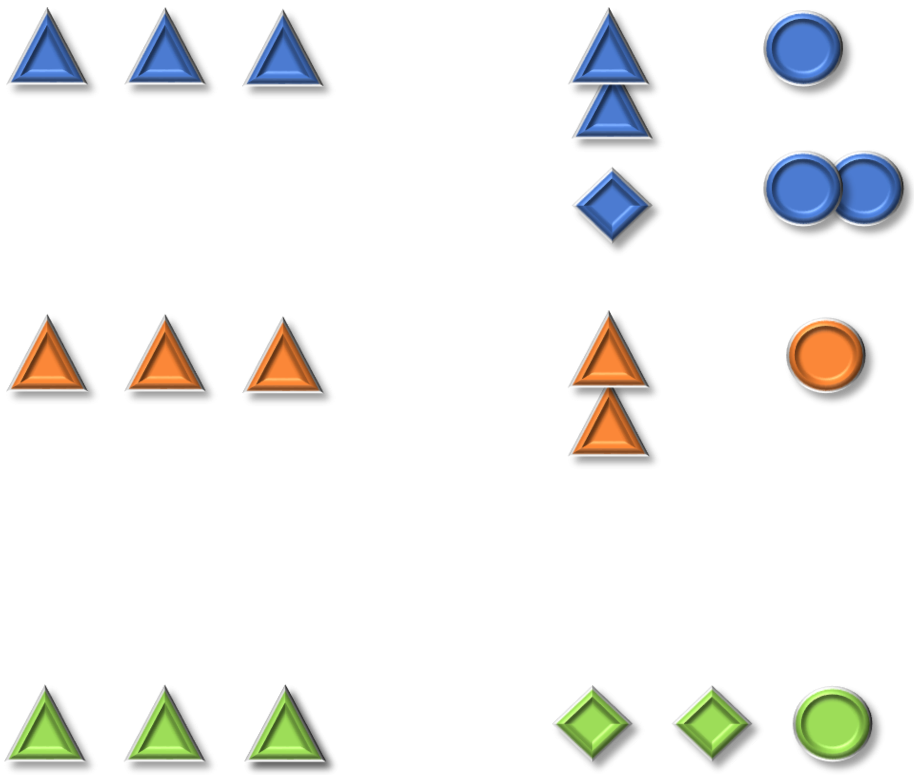
What

Domain Driven



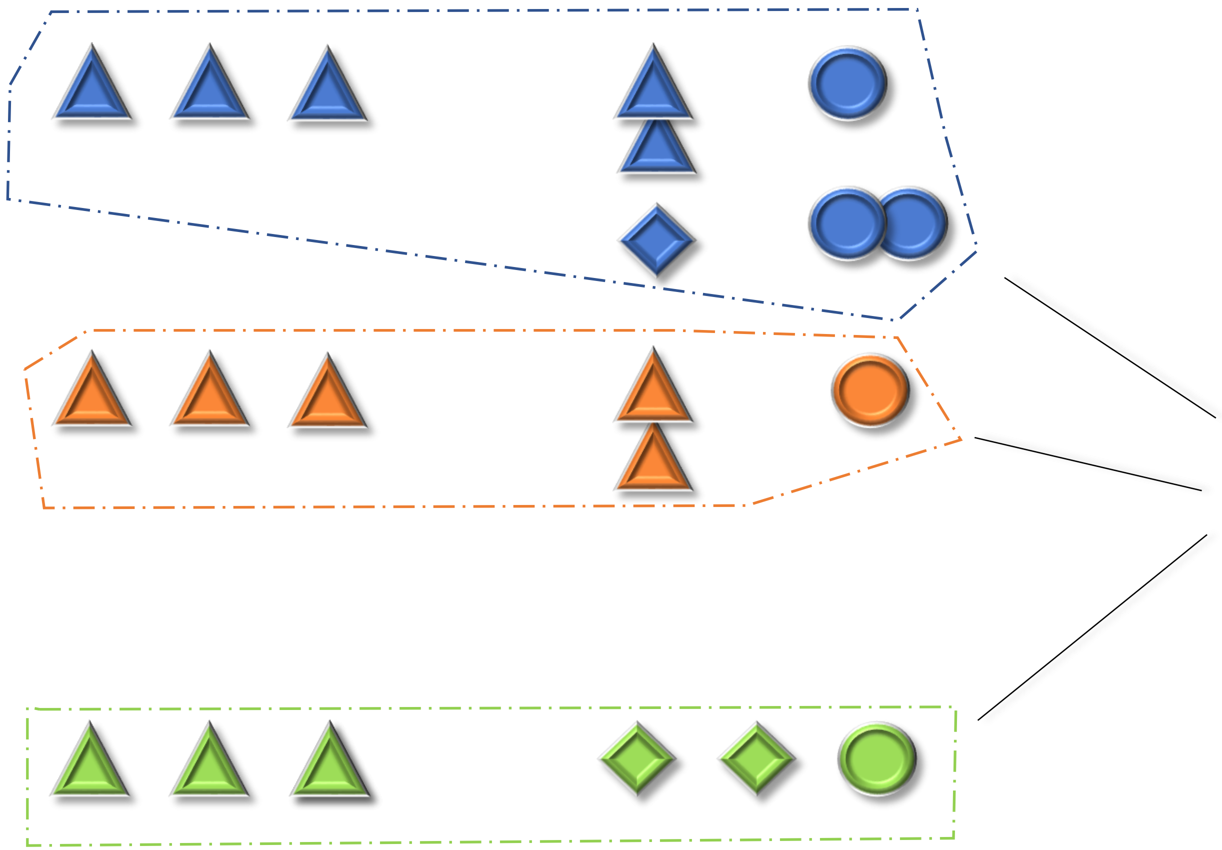
What

Domain Driven



What

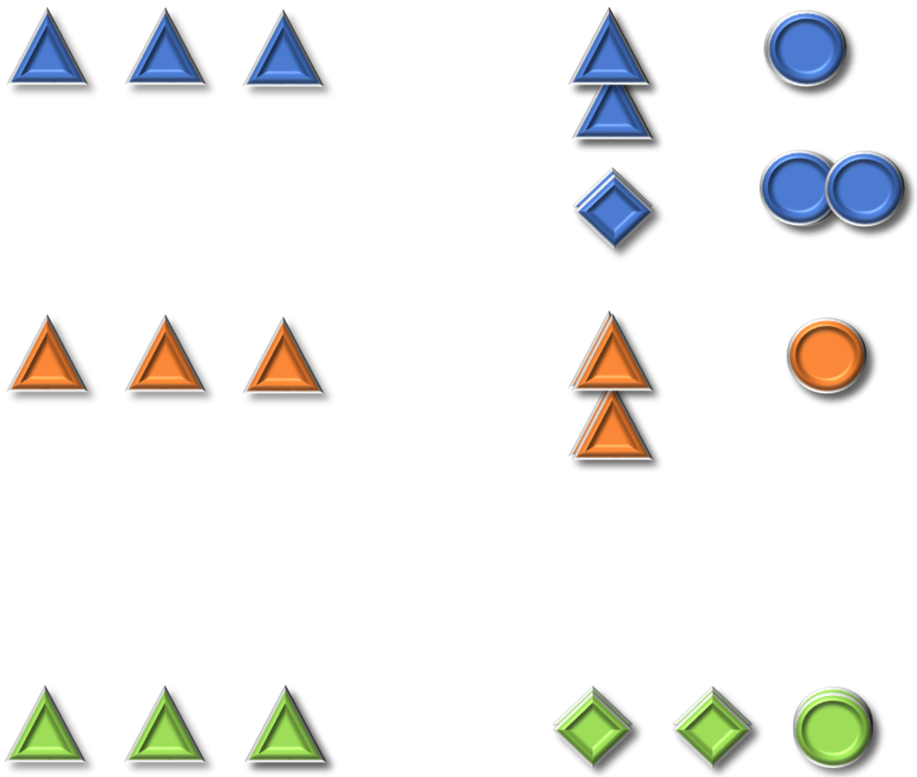
Domain Driven



Source/Operational Domains

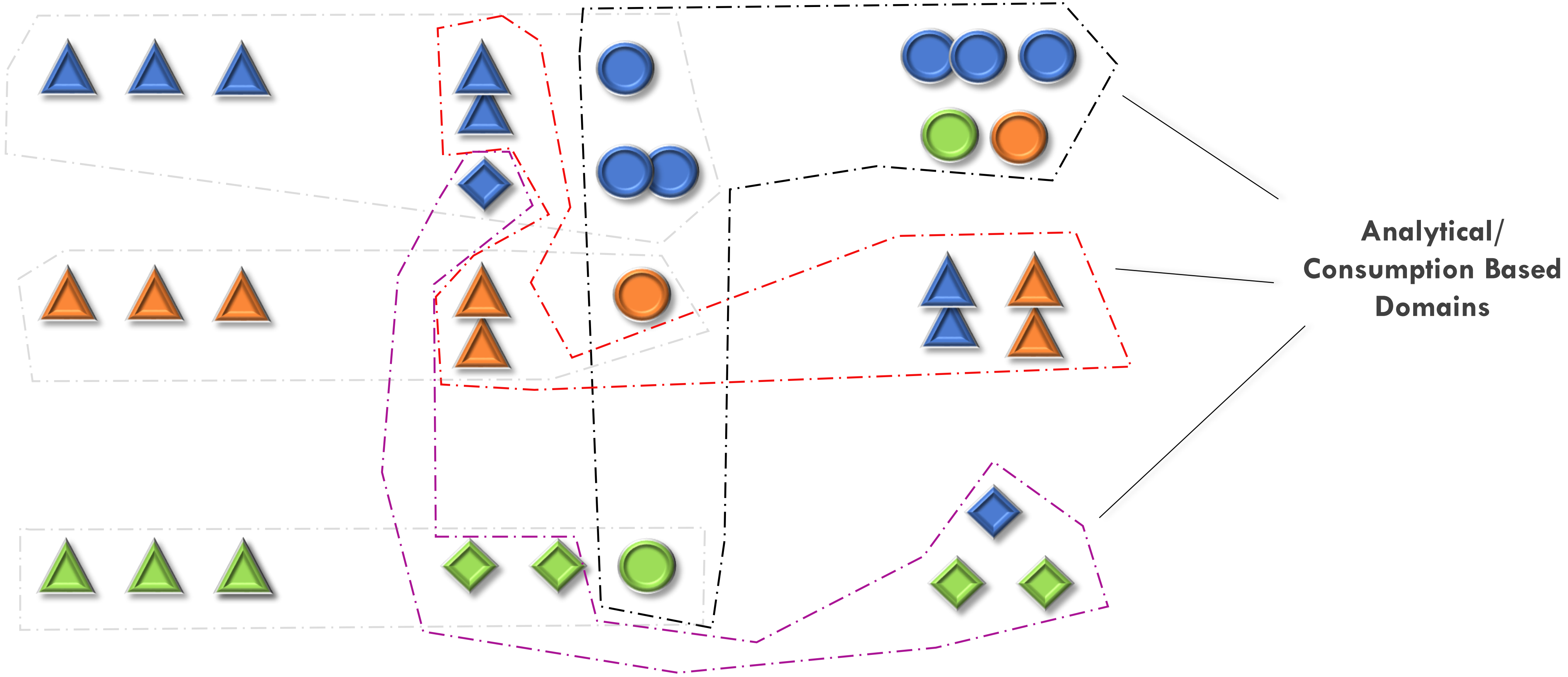
What

Domain Driven



What

Domain Driven



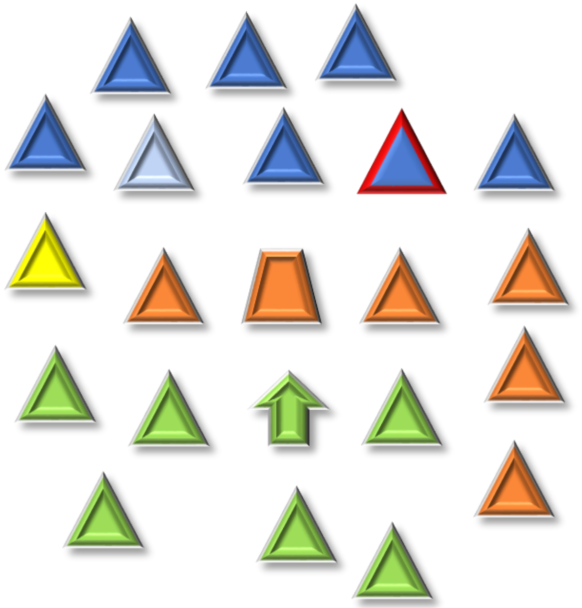
What

Key Components

Product Thinking on Data

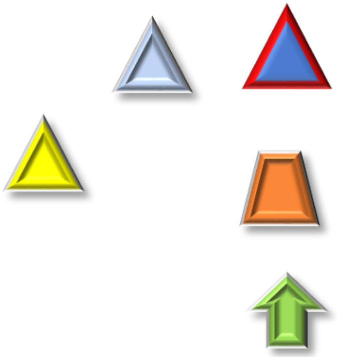
What

Product Thinking on Data

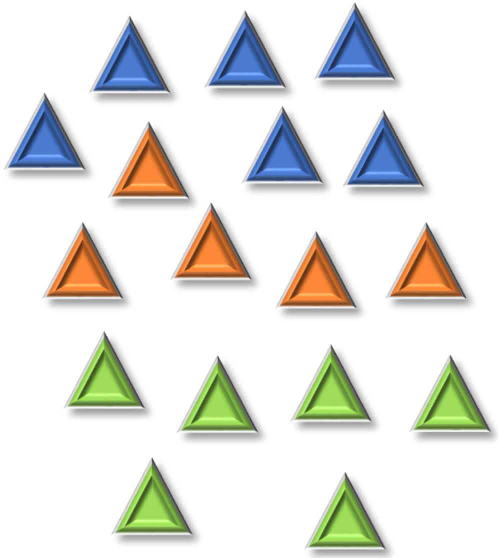


What

Product Thinking on Data



Data Quality as Design than a reaction

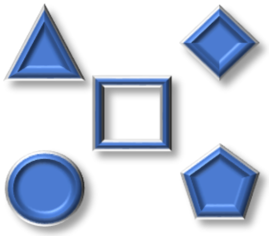


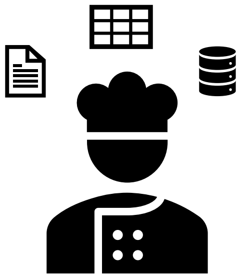
What

Product Thinking on Data

What

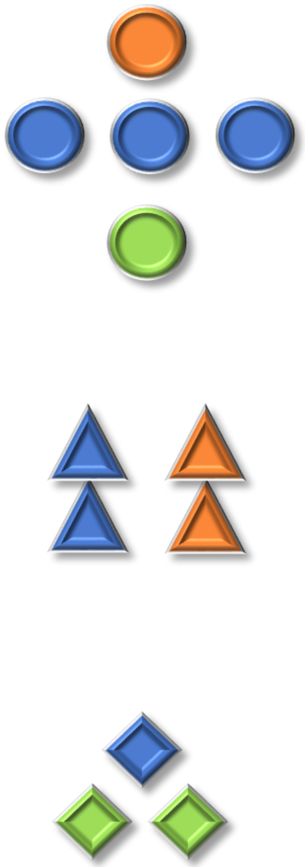
Product Thinking on Data



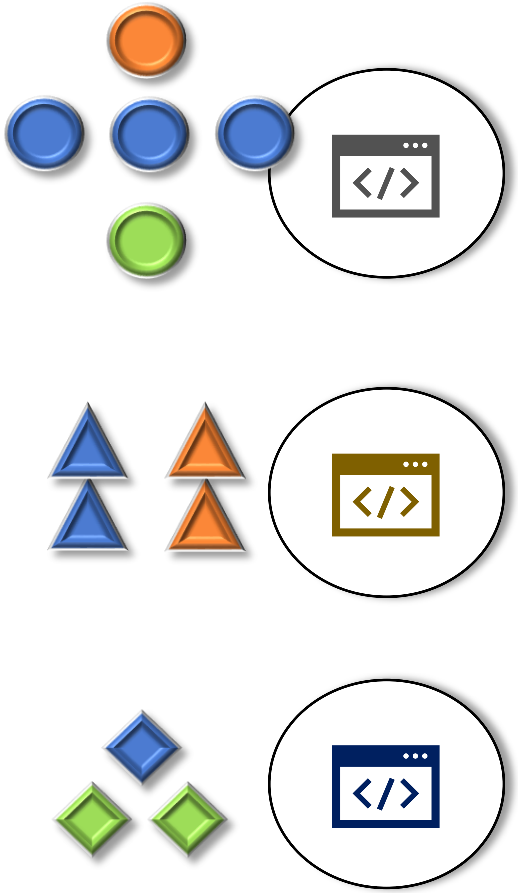


Data Served for different needs

What

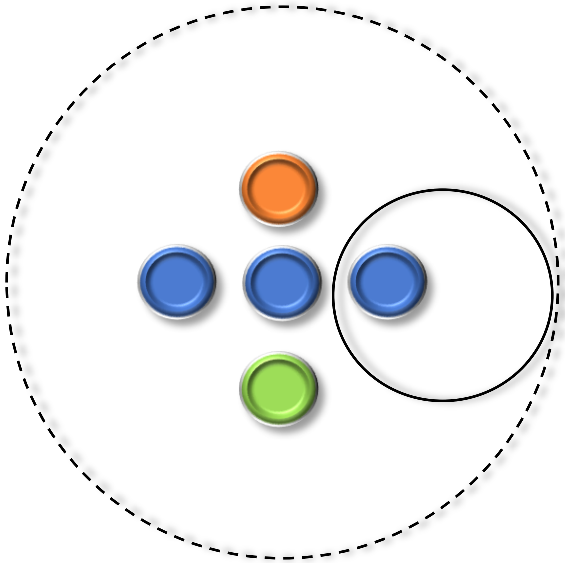


What



Metadata & Data Discovery

Product Thinking on Data



Data Security & Privacy

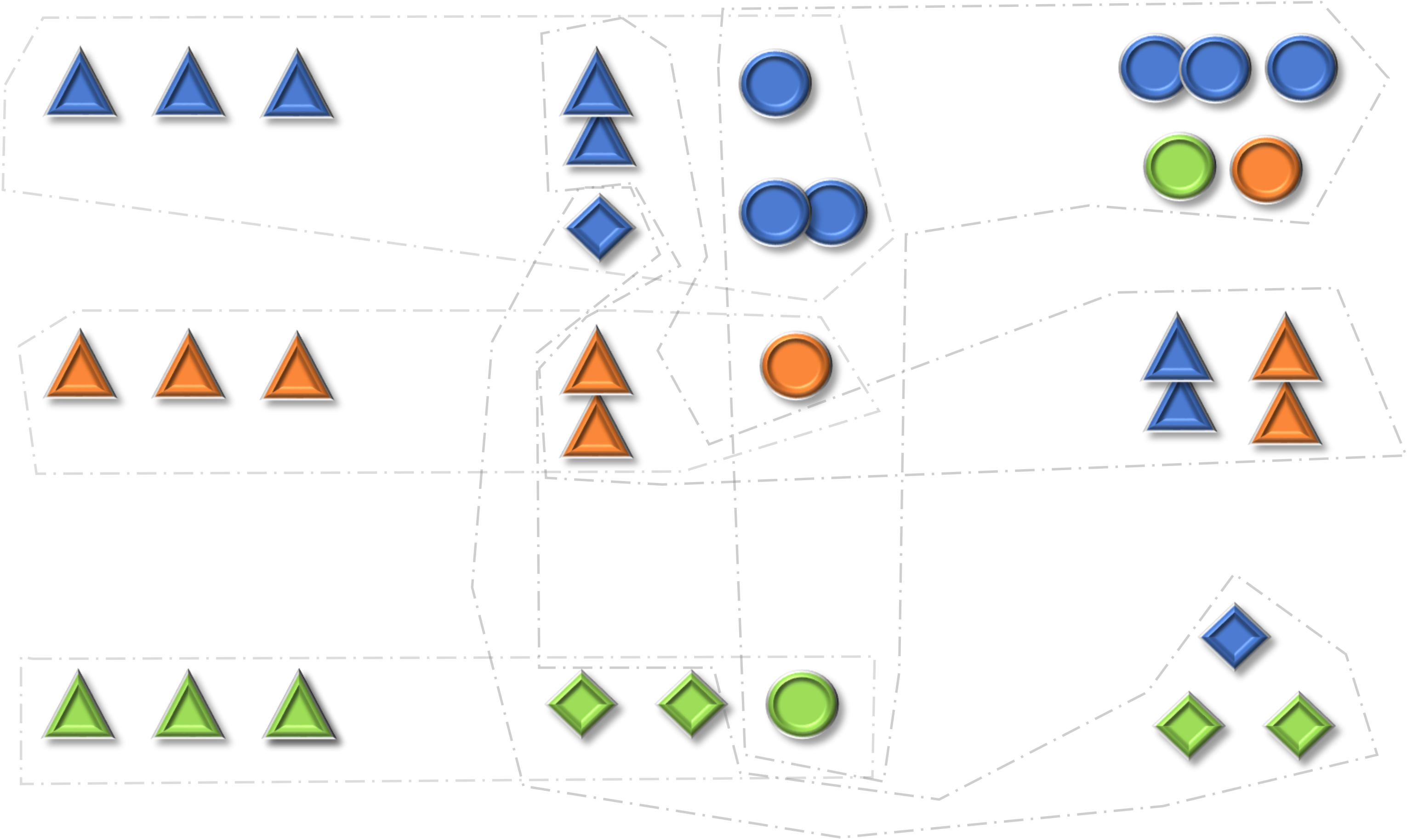
What

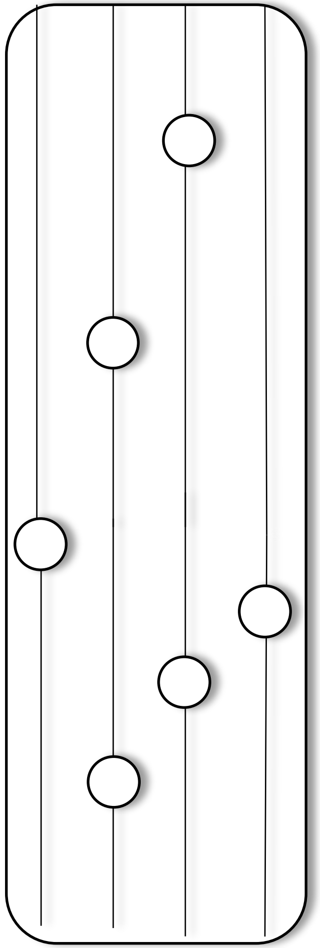
Key Components

Self Service
Platform

What

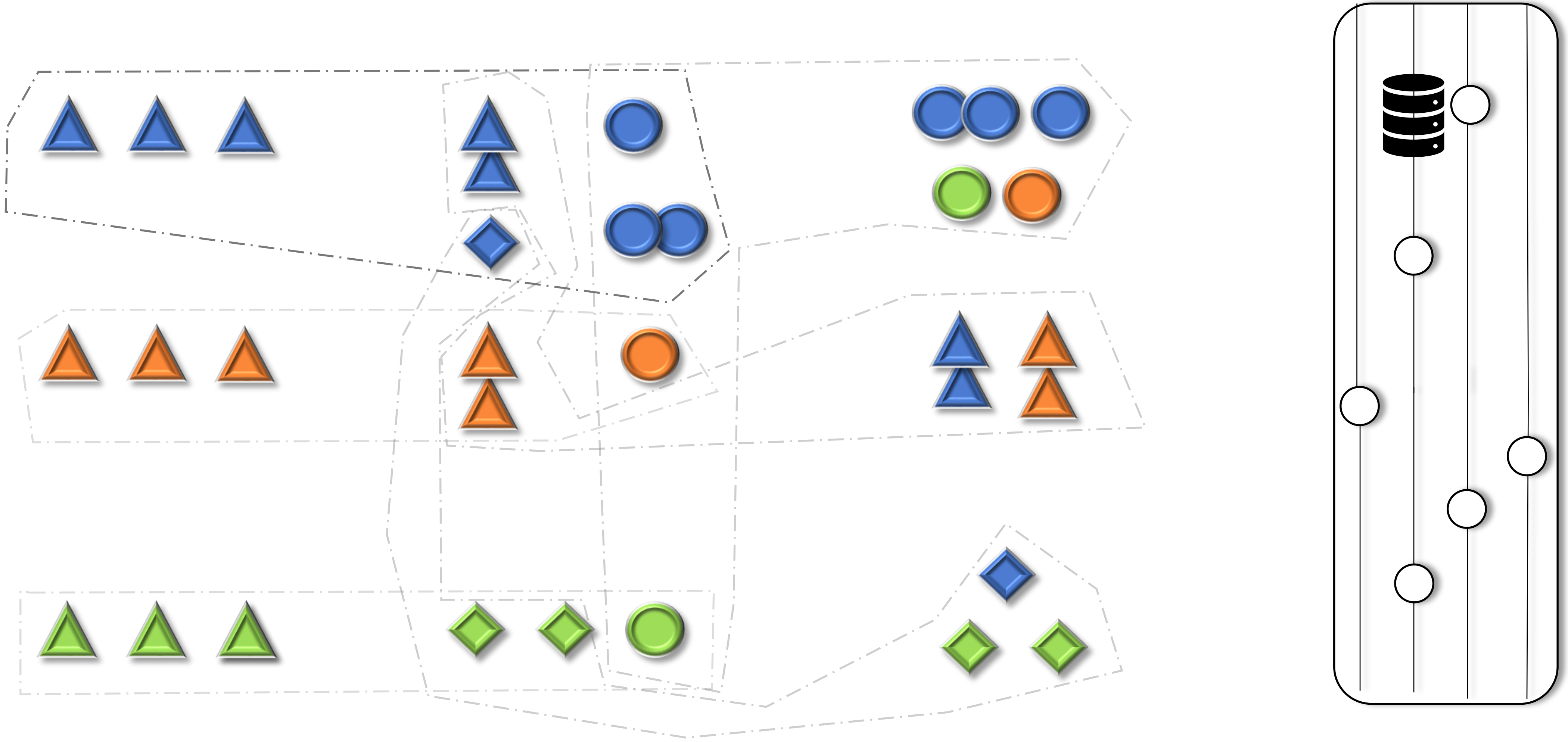
Domain Driven





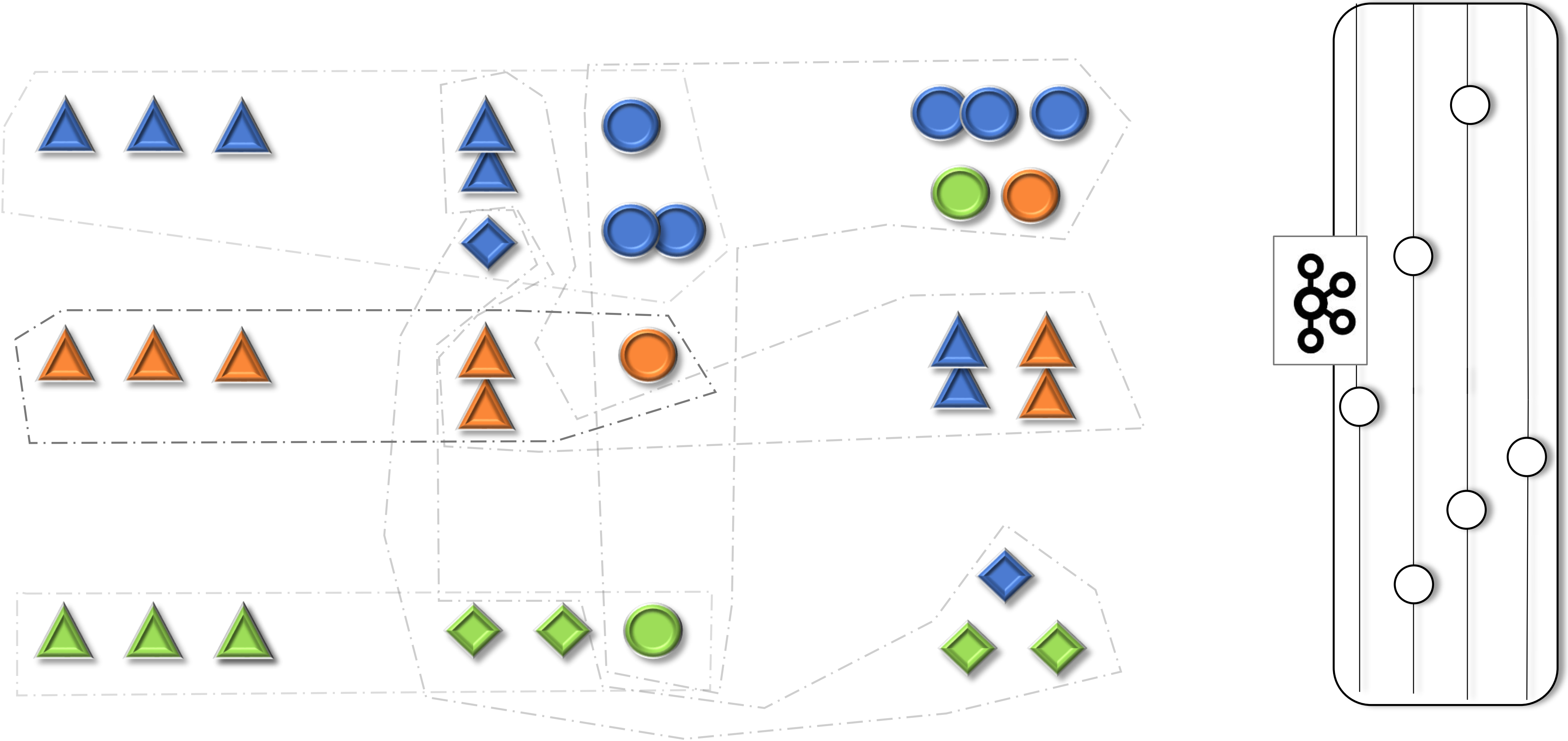
What

Domain Driven



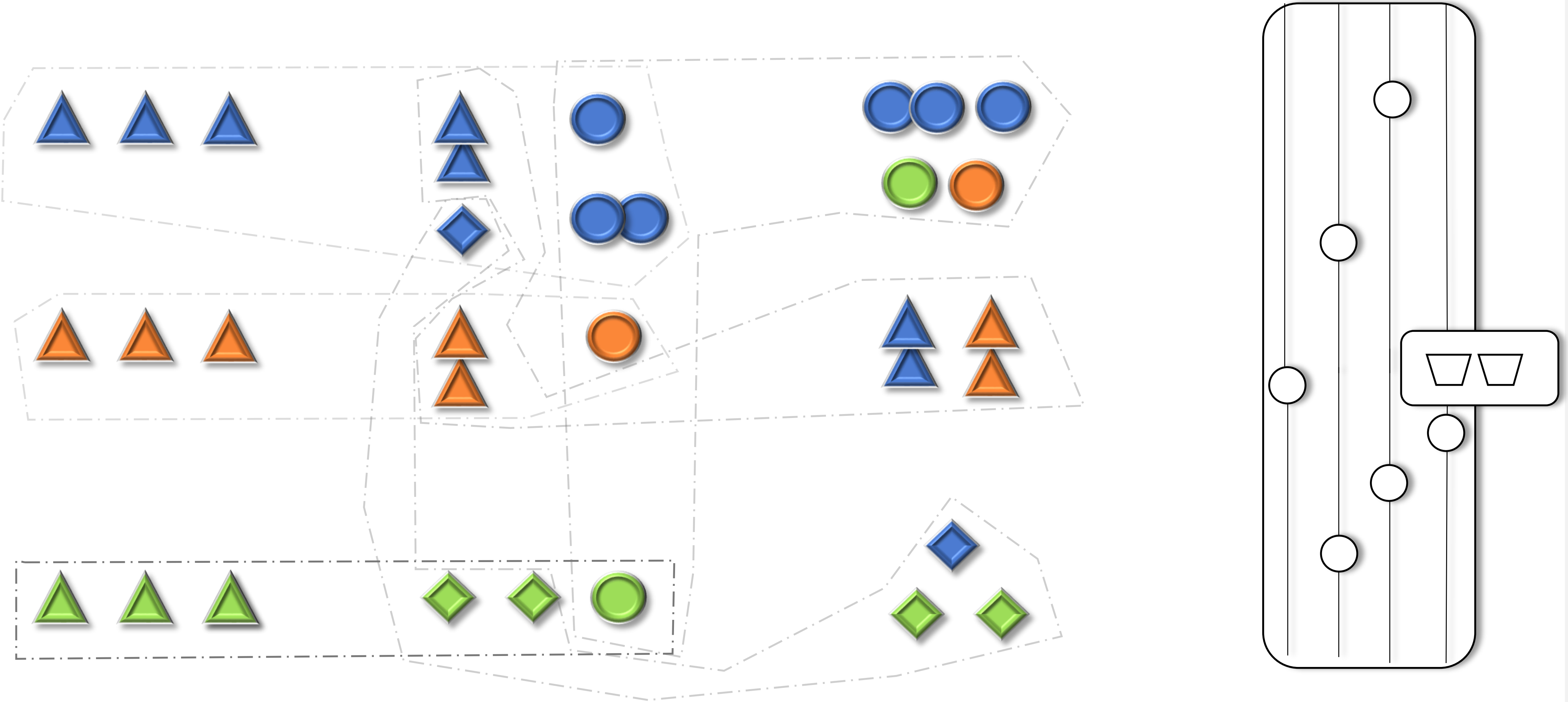
What

Domain Driven



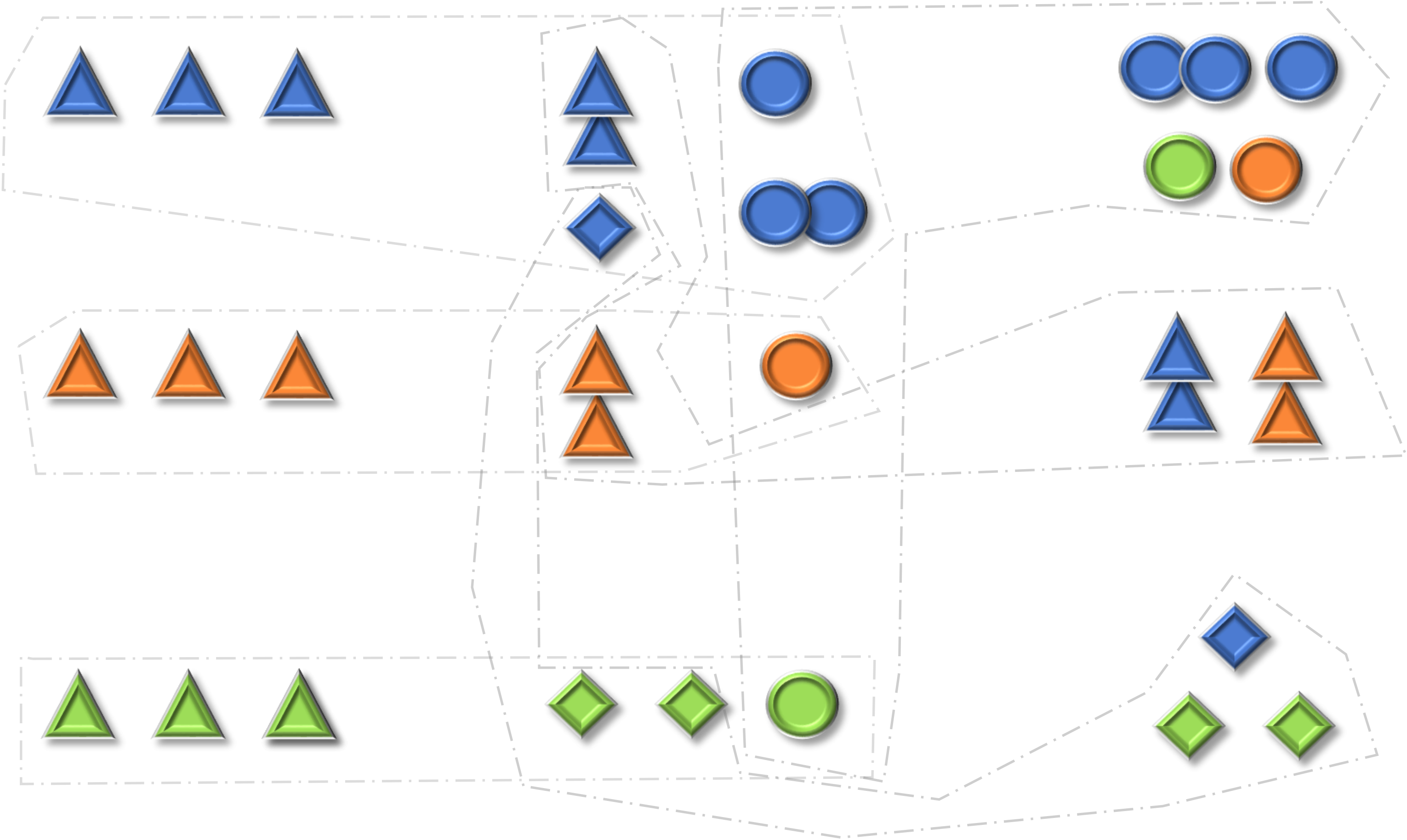
What

Domain Driven

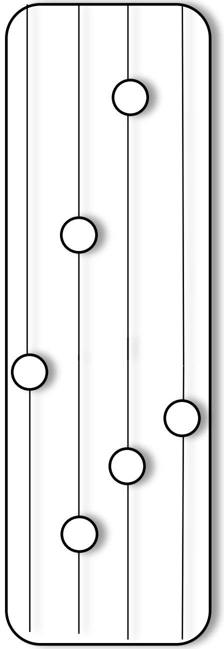


What

Domain Driven

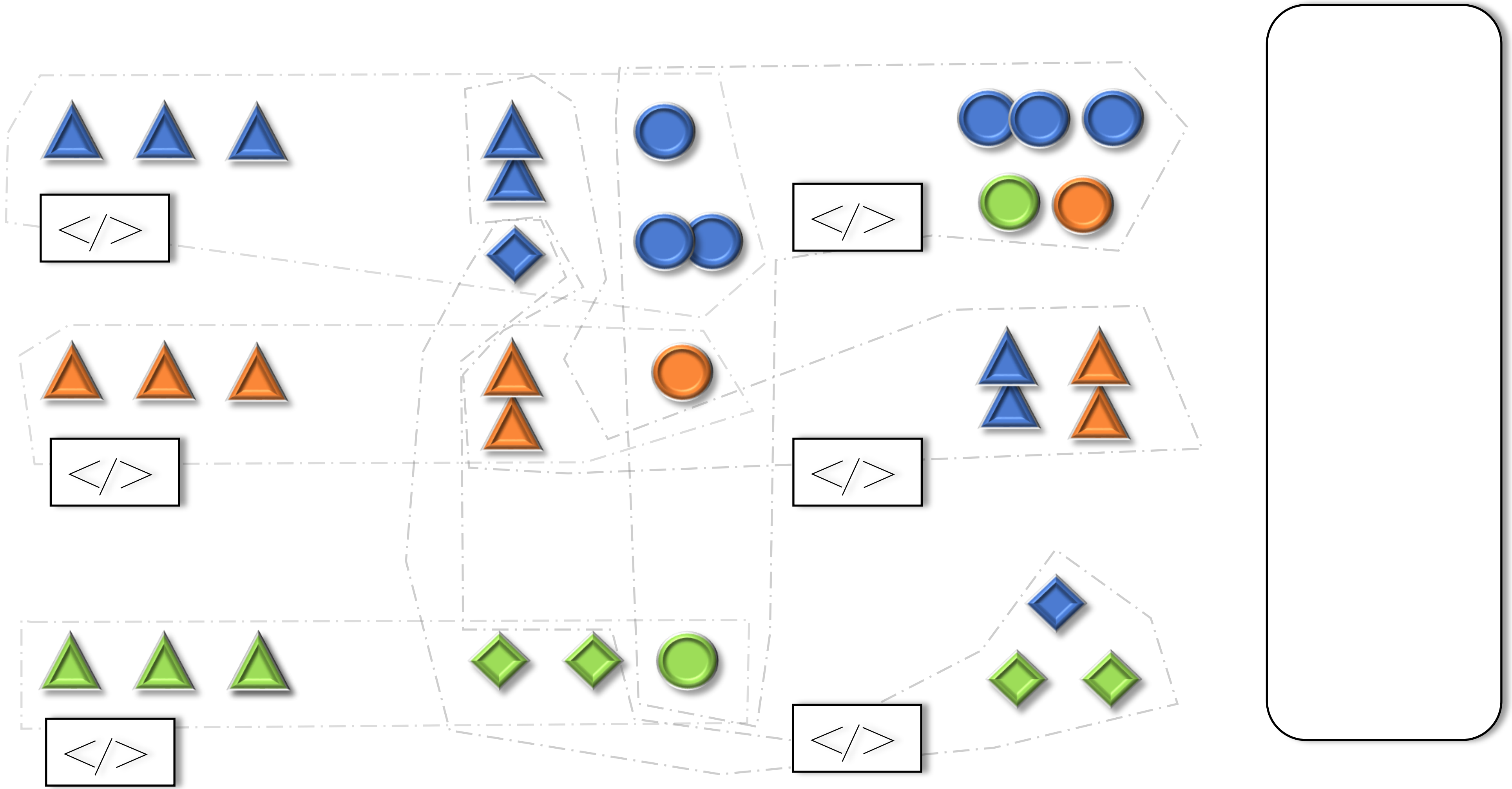


Infrastructure as a Platform



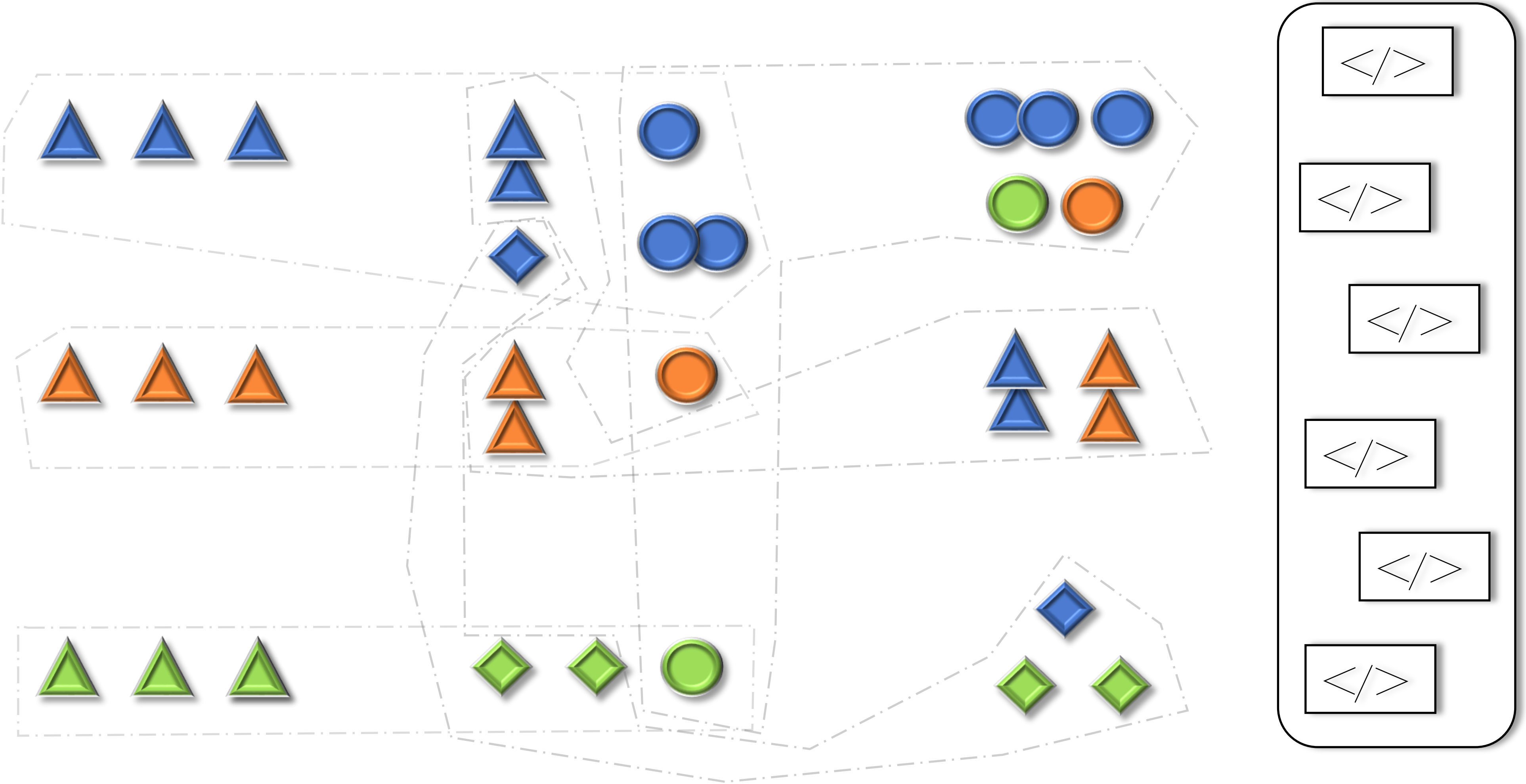
What

Domain Driven



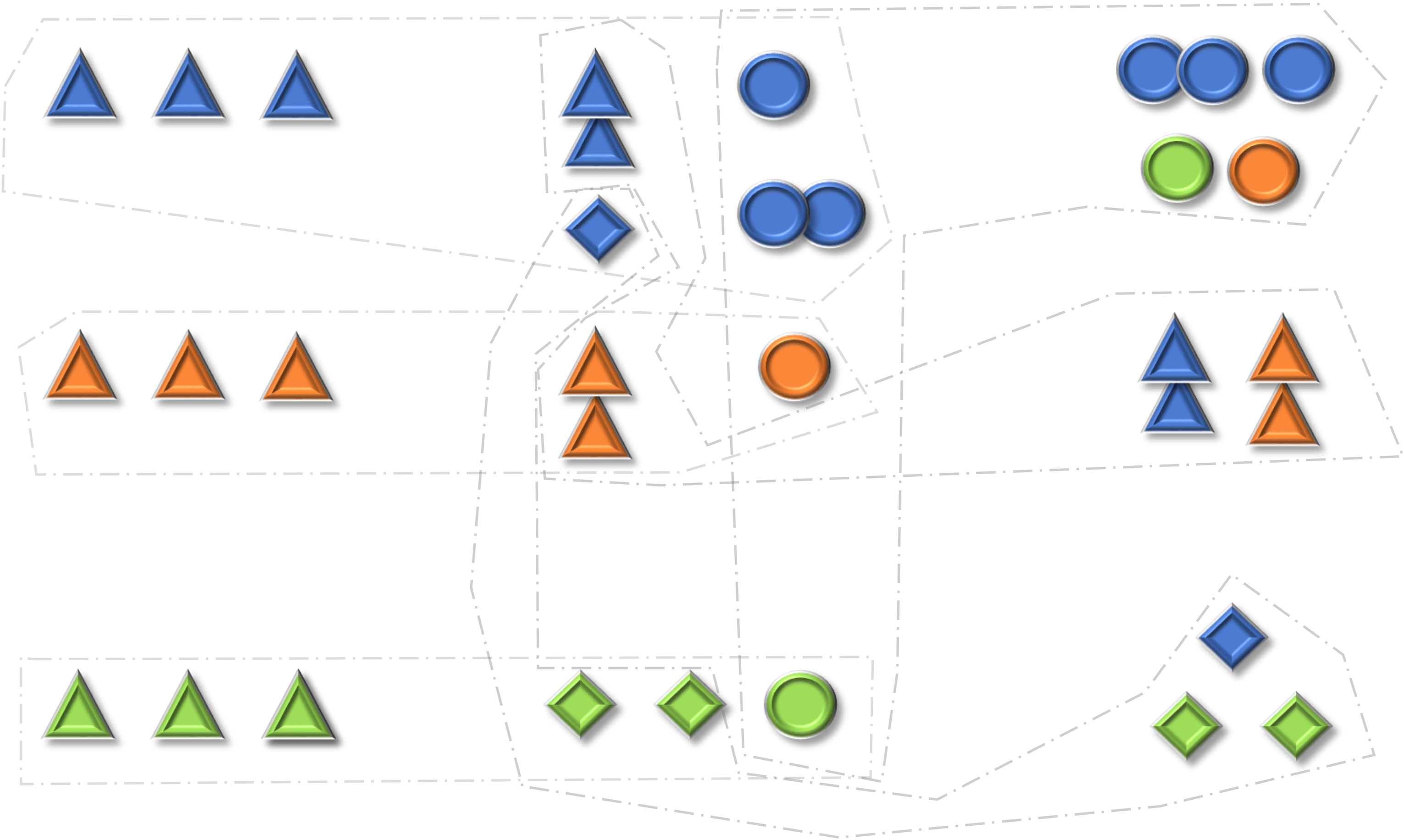
What

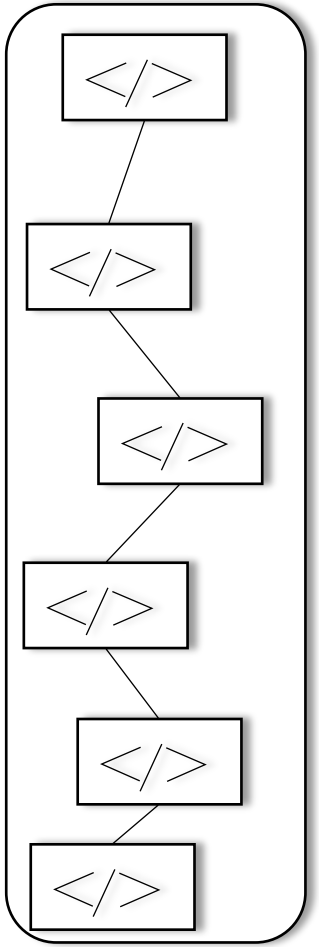
Domain Driven



What

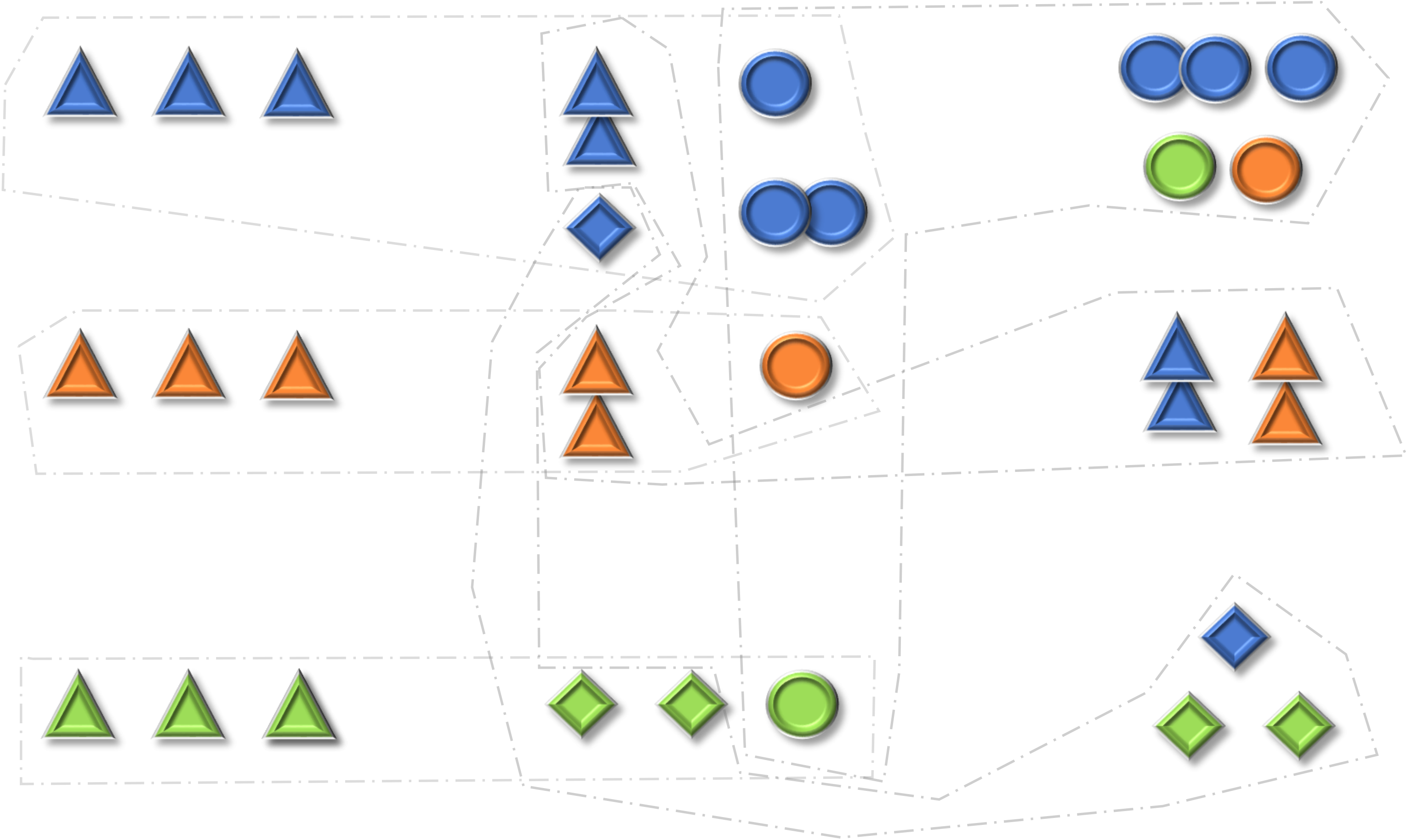
Domain Driven



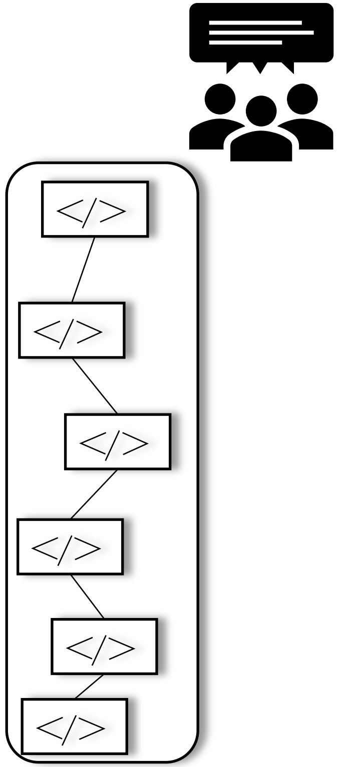


What

Domain Driven



Metadata & Lineage as a Platform



What

Mindset change

Agenda

Why?

Why do Organizations need to consider a Data Mesh/Fabric?

What

What are some of the key components in a Data Mesh/Fabric?

How?

How do we go about designing & architecting one?

Agenda

How?

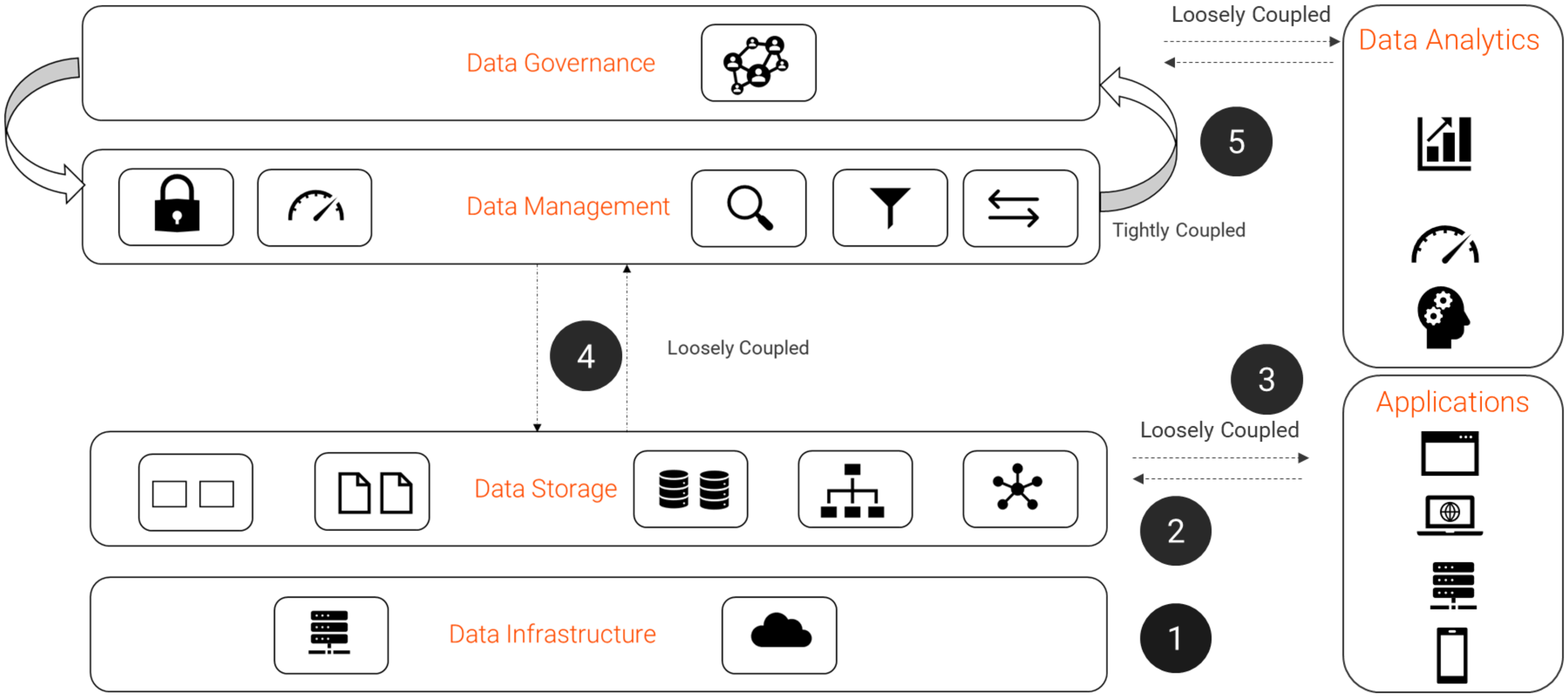
How do we go about designing & architecting?

How?

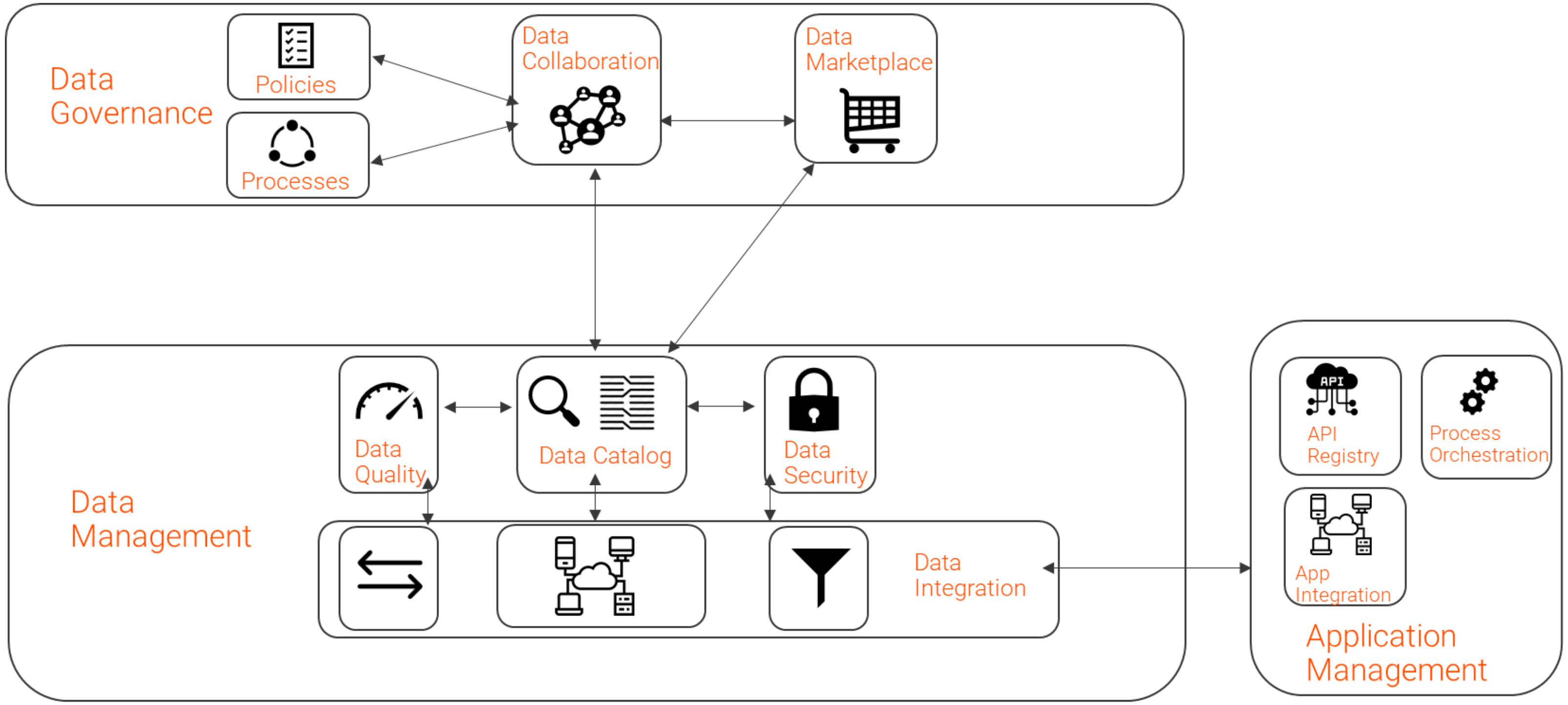
Chat Time – Type in one line - What do you think are key aspects in implementing such a Solution Architecture?

How?

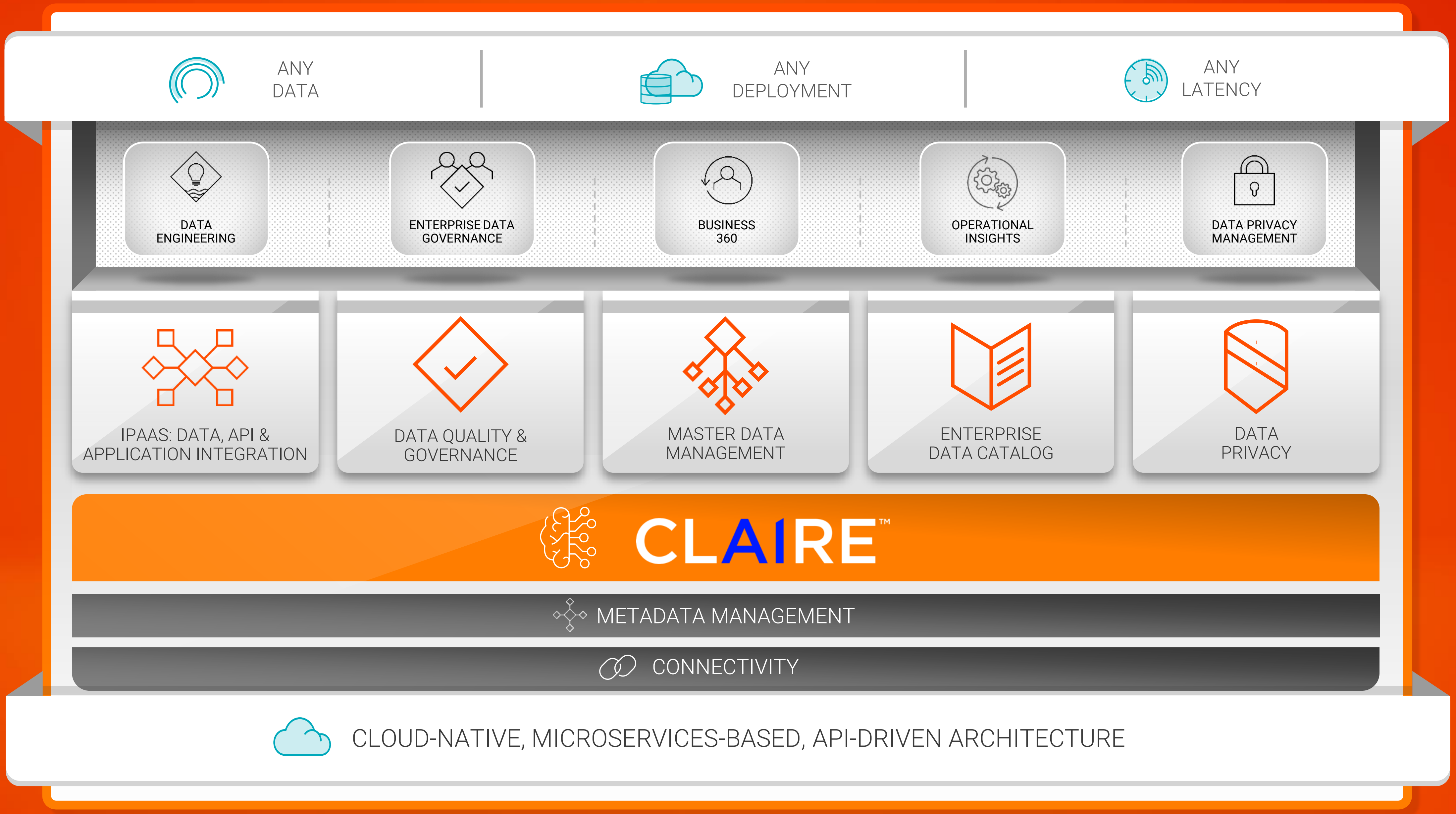
Lets take a step back



Data Management Platform



Informatica The Intelligent Data Platform



ANY DATA

ANY DEPLOYMENT

ANY LATENCY

DATA ENGINEERING

ENTERPRISE DATA GOVERNANCE

BUSINESS 360

OPERATIONAL INSIGHTS

DATA PRIVACY MANAGEMENT

IPAAS: DATA, API & APPLICATION INTEGRATION

DATA QUALITY & GOVERNANCE

MASTER DATA MANAGEMENT

ENTERPRISE DATA CATALOG

DATA PRIVACY

CLAIRE™

METADATA MANAGEMENT

CONNECTIVITY

CLOUD-NATIVE, MICROSERVICES-BASED, API-DRIVEN ARCHITECTURE

How?

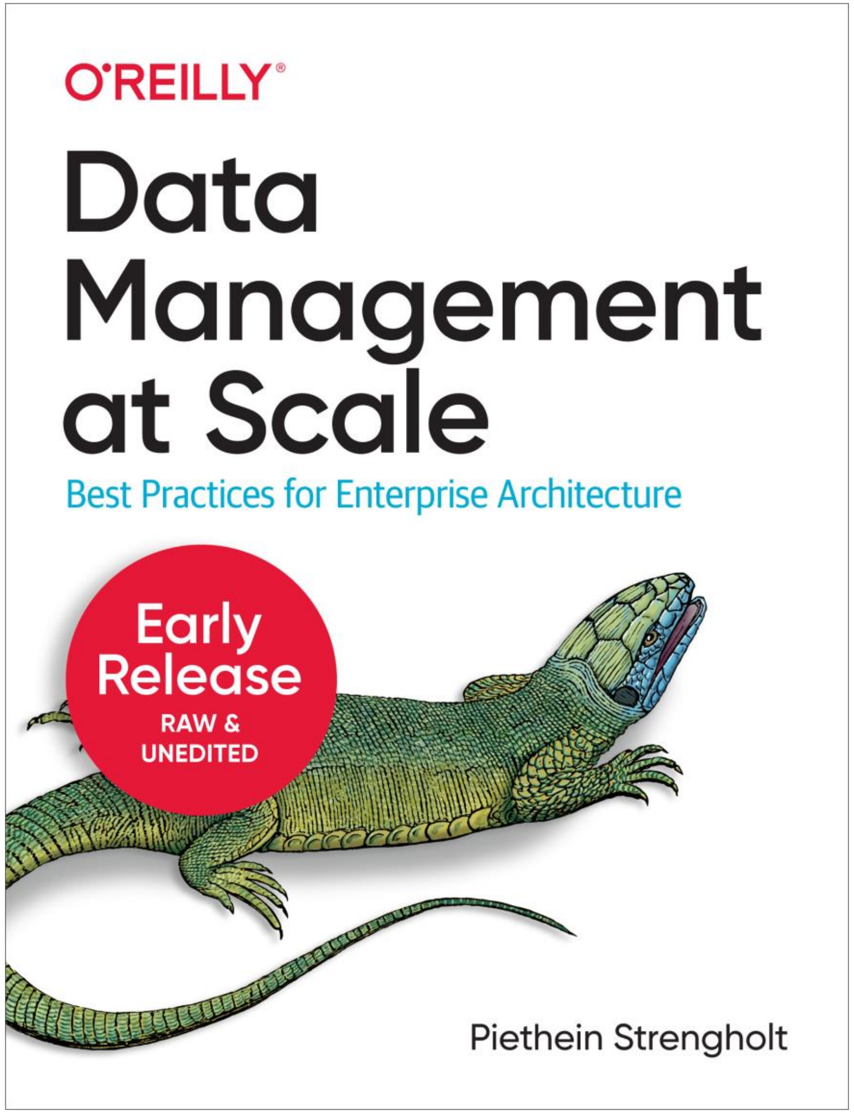
Chat Time – Type in one line - What do you think are key aspects in implementing such a Solution Architecture?

ABN·AMRO

Chief Architect & Data Management (CADM)

ABN AMRO’s Integration Architecture

External trends that will transform the current landscape and impact the business



O'REILLY

Data Management at Scale

Best Practices for Enterprise Architecture

Early Release

RAW & UNEDITED

Piethein Strengholt

Piethein Strengholt

- Principal Architect for Data & Integration @ ABN AMRO
- Author of the book Data Management at Scale
- 10+ Consultancy experience
- Cloud certified (both Azure & AWS)

External trends that will transform the current landscape and impact the business

New developments

New (open source) concepts are introduced, such as NoSQL database types, Block chain, new database designs, distributed models (Hadoop), new analytical, etc.

Cloud, Services & API connectivity

Cloud, API's make it easier to integrate. Software & Platform as a Service (SAAS, PAAS) offerings will push the connectivity and API usage even further.

Increased regulatory attention

Stronger regulatory requirements, such as BCBS 239. Data Quality and Data Lineage becomes more important.

Increase of computing power

Massive increase of computing power driven by hardware innovation (SSD storage, in-Memory, GPU) let us move the data to the compute.

Exponential growth of (outside) data

Exponential growth of data; especially external (open data, social), internal, structured, unstructured can all be used for delivering more insight.

The read/write ratio increases

The read/write ratio changes because of intensive data consumption. Data is read much more, increased real-time consumption, more search.

Internal drivers for change; putting an emphasis on more controls, governance and self-service

Outdated design

The current Data Warehouse & BI Architecture is based on the common practise of the late nineties. It lacks agility and fails delivering results quickly.

Design of Golden Sources

For current transactional applications the non-functional aspects are not sufficiently considered. Segregation of the transactional commands and read commands.

Self Service and finding the data

Self-Service is in demand. Users want to drive their own insights and initiatives. While on the other hand it is difficult to find where the data is and how get and use data.

More control & better governance

With the growing amount and usage of data it will be more difficult to control the situation without strong (meta) data management and governance controls.

New business models

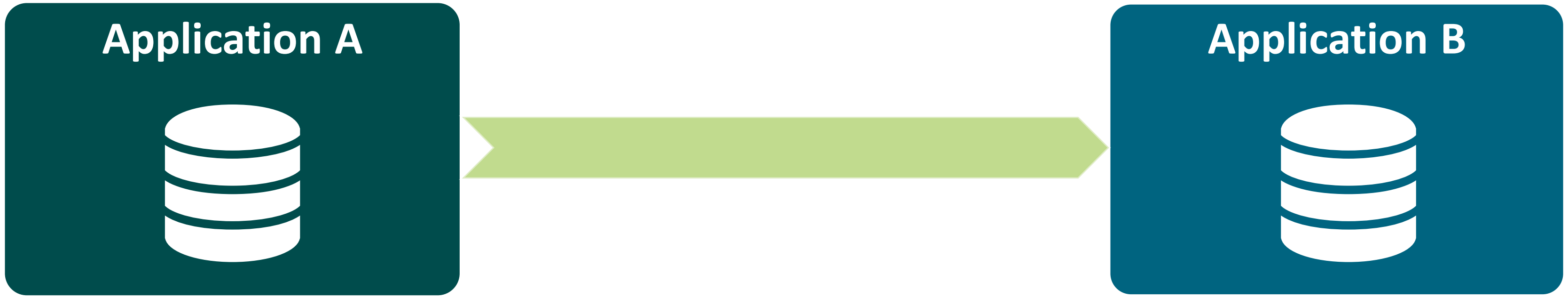
The business wants to develop new business models based on Data. Data becomes the core of future value propositions.

High pressure on costs

Costs for Change and Run are currently very high, due to long release cycles and high rebilling costs (both internal and external).

Our hypothesis is that every application that creates data, needs and will have a database

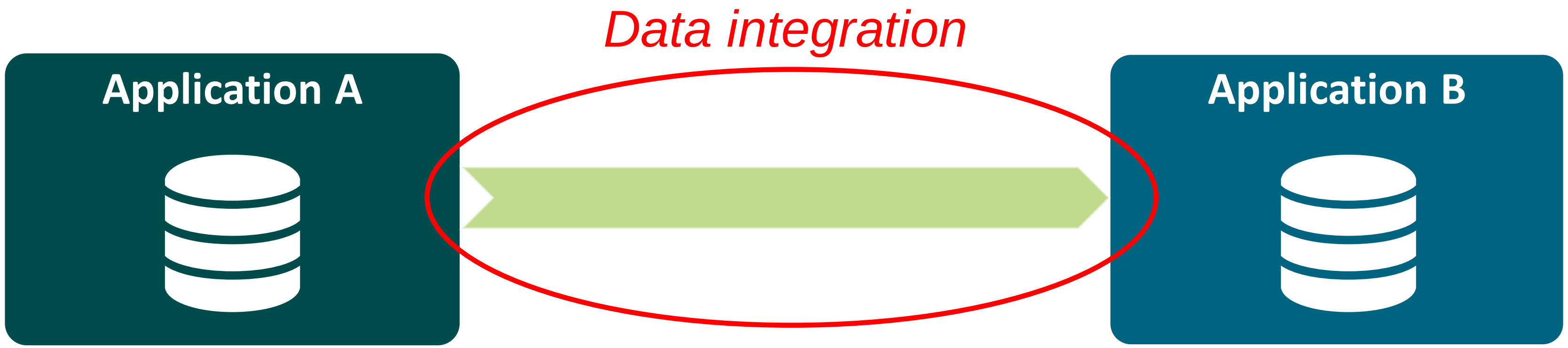
We made a hypothesis that every application (at least in the context of a banking application), that creates data, needs and will have a database. Even stateless applications that create data have 'databases'. In these scenarios the database typically sits in the RAM or in a temp file.



Consequently, when we have two applications, we hypothesize that each application has its own 'database'. When there is interoperability between these two applications, we expect data to be transferred from one application to the other.

Transferring data between applications is always complemented by data integration

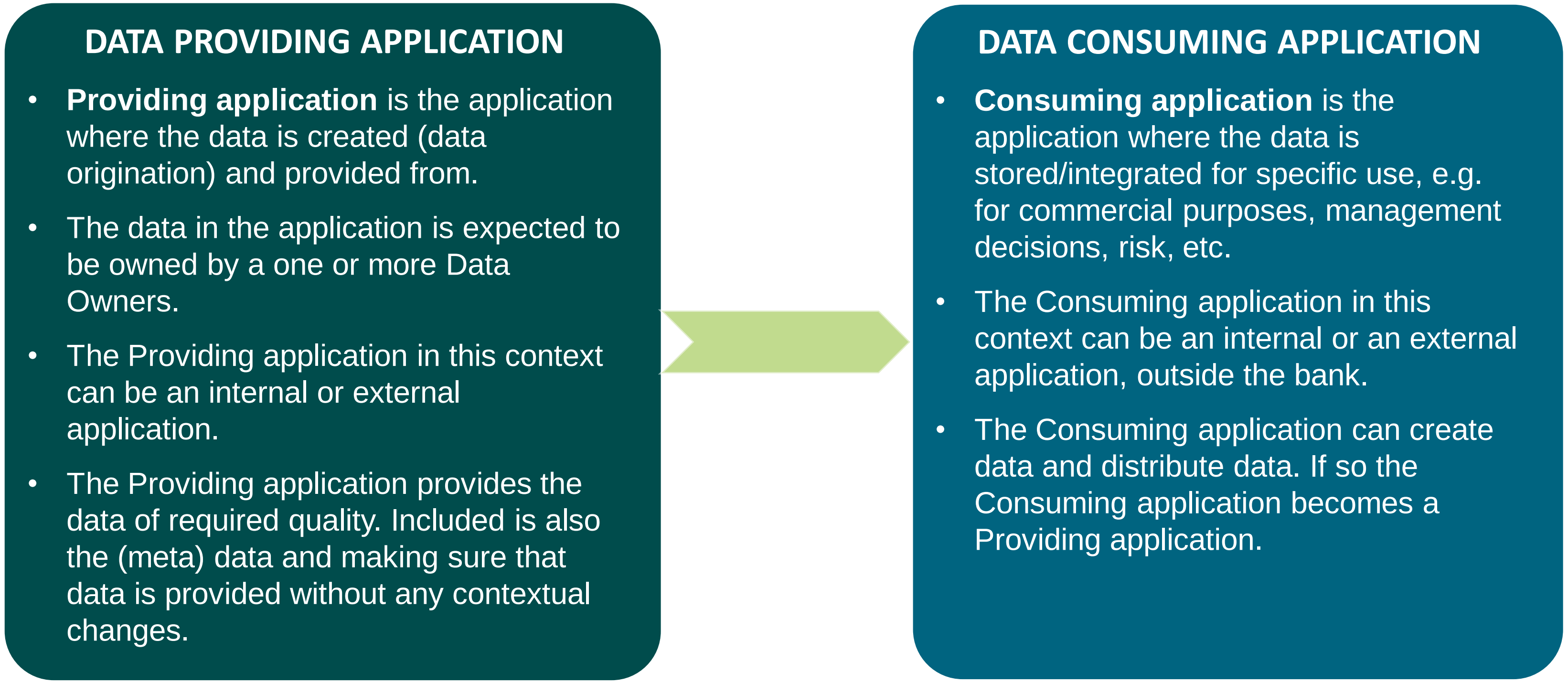
A crucial aspect when it comes to data transfer is that data integration is always right around the corner. Whether you do ETL or ELT, virtual or physical, batch or real-time, there's no escape from the data integration dilemma.*



The 'always' required data transformation lies in the fact that an application database schema is designed to meet the application's specific requirements. Since the requirements differ from application to application, the schema's are expected to be different and data integration is always required when moving data around.

Application definitions of Data Providing and Data Consuming used within the context of the FSA

Applications are either data providers or data consumers and, as we will see, sometimes both. These concepts will frame our future architecture



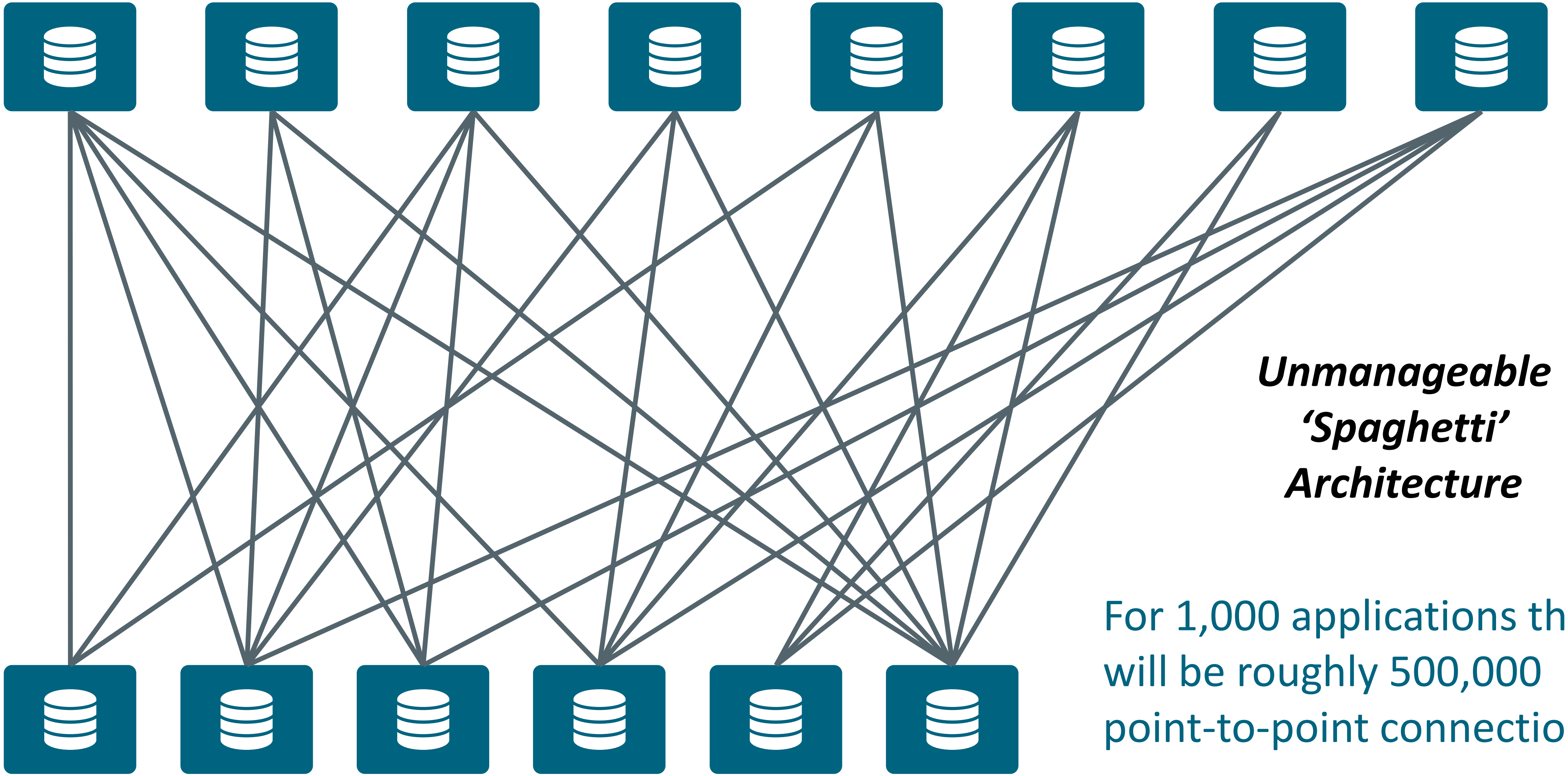
DATA PROVIDING APPLICATION

- **Providing application** is the application where the data is created (data origination) and provided from.
- The data in the application is expected to be owned by a one or more Data Owners.
- The Providing application in this context can be an internal or external application.
- The Providing application provides the data of required quality. Included is also the (meta) data and making sure that data is provided without any contextual changes.

DATA CONSUMING APPLICATION

- **Consuming application** is the application where the data is stored/integrated for specific use, e.g. for commercial purposes, management decisions, risk, etc.
- The Consuming application in this context can be an internal or an external application, outside the bank.
- The Consuming application can create data and distribute data. If so the Consuming application becomes a Providing application.

Point-to-point connections can't provide the control and agility enterprises need, because the sheer number of communication channels makes it nearly impossible to oversee all dependencies



Unmanageable 'Spaghetti' Architecture

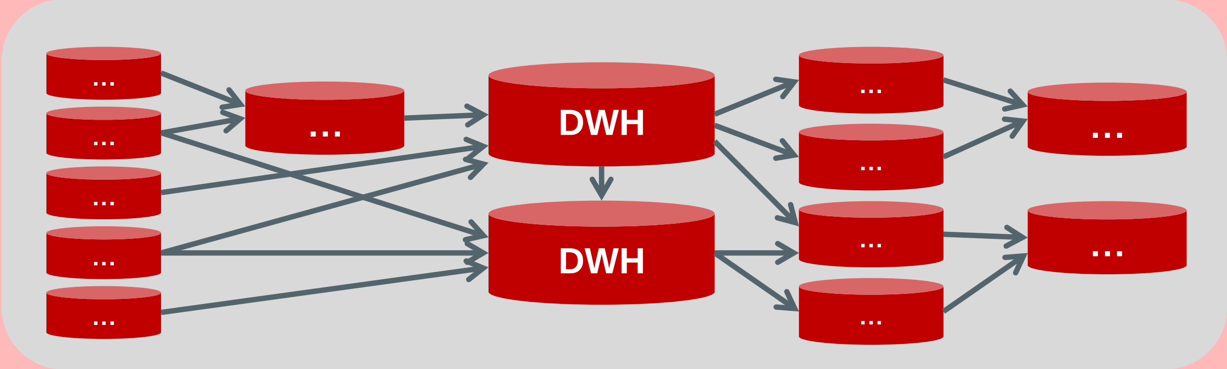
For 1,000 applications there will be roughly 500,000 point-to-point connections.

Silos (EDWs and DWHs) have the advantage of quickly getting all the data, but on the scale of a large enterprise, using silos for data distribution doesn't provide agility

DATA PROVIDER

- Data is delivered with no clear purpose.
- Limited definitions and schema information available.
- Data is off loaded and removed very quickly.
- Have no knowledge and control of data consumption and distribution.
- Data is provided with DQ issues
- Not designed for the future

Traditional Data Warehouse



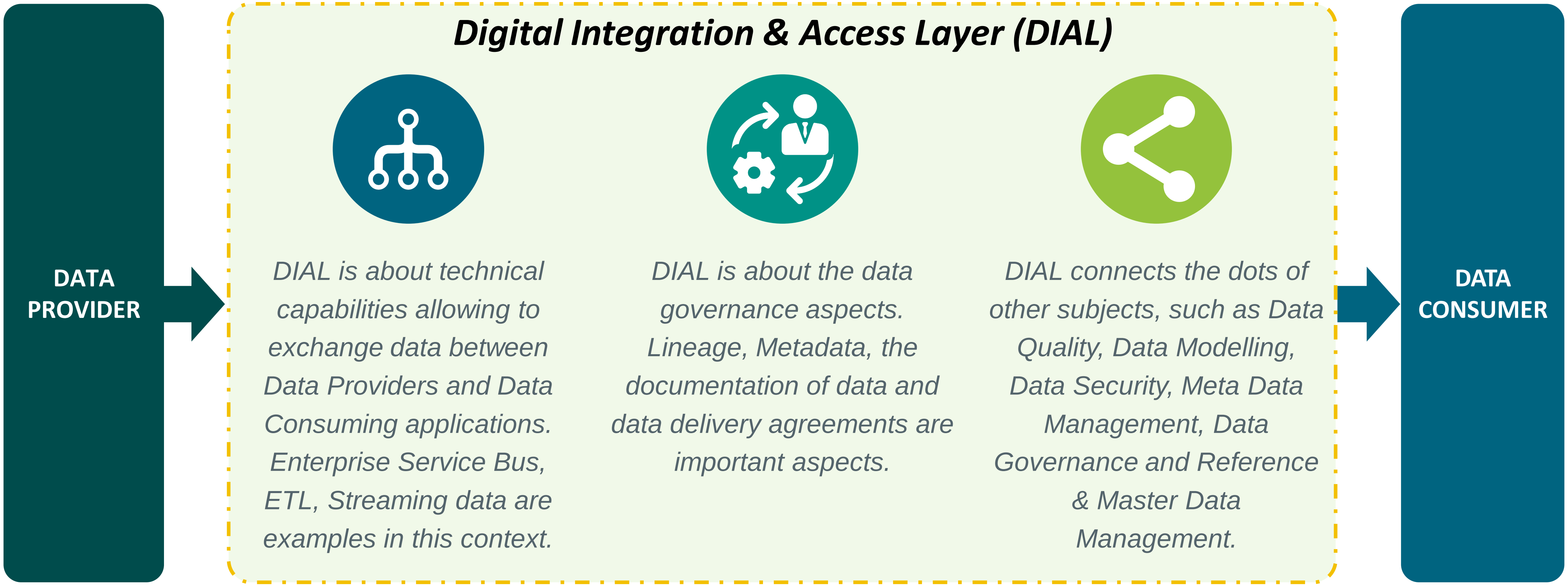
- Unification leads to loss of context and meaningless data
- Tremendous effort of integration and coordination leads to bypasses.
- Lot of centralized thinking; Single data model, single team, etc. People with data engineering skills, are separated from the people with domain and business knowledge
- Unable to optimize for specific read patterns. Consequently the underlying *expensive* hardware is only used for storage.
- Data Governance unclear. People with data engineering skills, are separated from the people with domain and business knowledge.
- All data is integrated upfront, without being consumed. Long release cycles. Many stakeholders involved.
- Data Quality is fixed in the ‘middle’. Difficult to judge what the origin is.

DATA CONSUMER

- The origin for next Data Consuming is unclear, since data is distributed further.
- Data changes not tracked and captured.
- Are impatient and act as Data Providing, resulting in point to point connections.
- Often don’t set any requirements. All data is first consumed.
- Don’t know where to find the right data.

Introduction of the “Digital Integration & Access Layer”

While the future will change a lot, the fundamental concept of Data Providing, Data Consuming and the need of data movement and data transformation won’t disappear.



DATA PROVIDER

Digital Integration & Access Layer (DIAL)

DIAL is about technical capabilities allowing to exchange data between Data Providers and Data Consuming applications. Enterprise Service Bus, ETL, Streaming data are examples in this context.

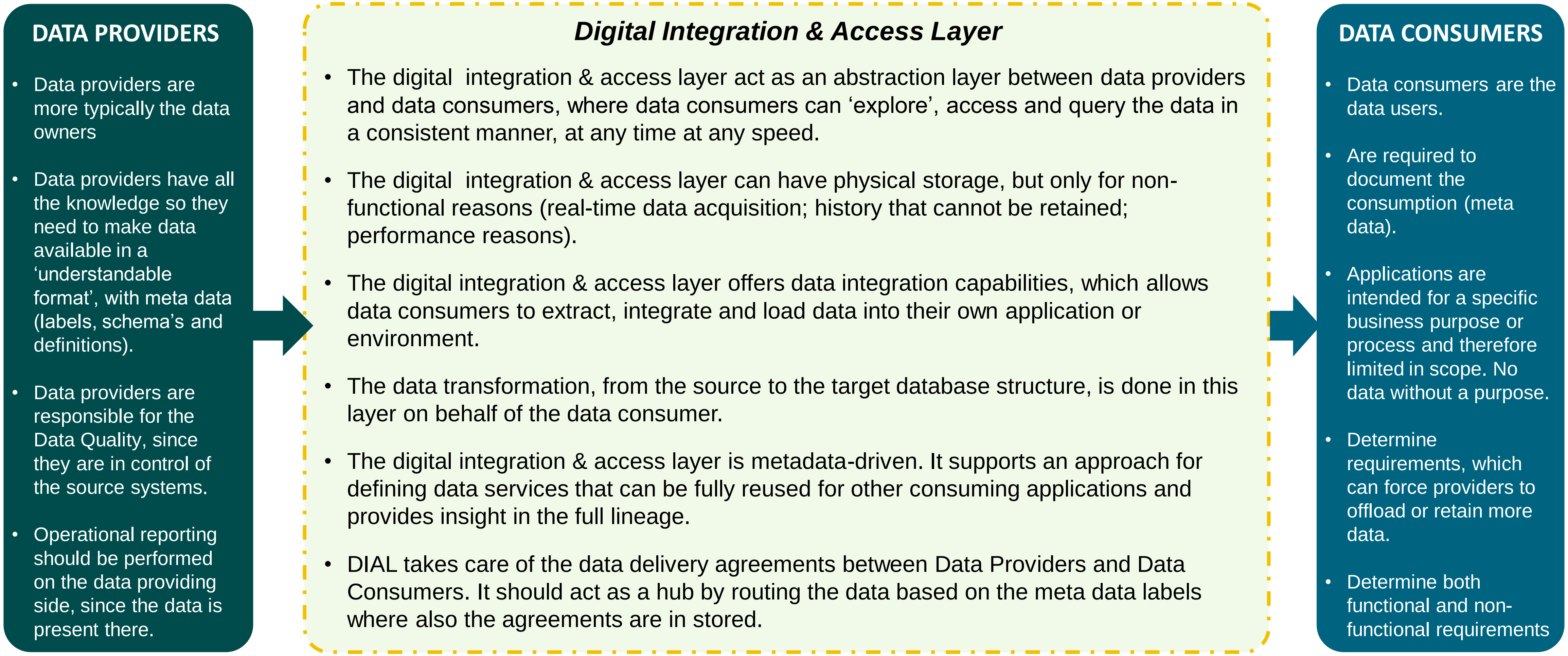
DIAL is about the data governance aspects. Lineage, Metadata, the documentation of data and data delivery agreements are important aspects.

DIAL connects the dots of other subjects, such as Data Quality, Data Modelling, Data Security, Meta Data Management, Data Governance and Reference & Master Data Management.

DATA CONSUMER

The Digital Integration & Access Layer is the answer for creating agility and having control when distributing and integration data

The “Digital Integration & Access Layer” is the modern just in time ‘warehouse’ for all data. It is used for connecting data providers and data consumers. It uses different techniques to meet the demands of the data consumers.



DATA PROVIDERS

- Data providers are more typically the data owners
- Data providers have all the knowledge so they need to make data available in a ‘understandable format’, with meta data (labels, schema’s and definitions).
- Data providers are responsible for the Data Quality, since they are in control of the source systems.
- Operational reporting should be performed on the data providing side, since the data is present there.

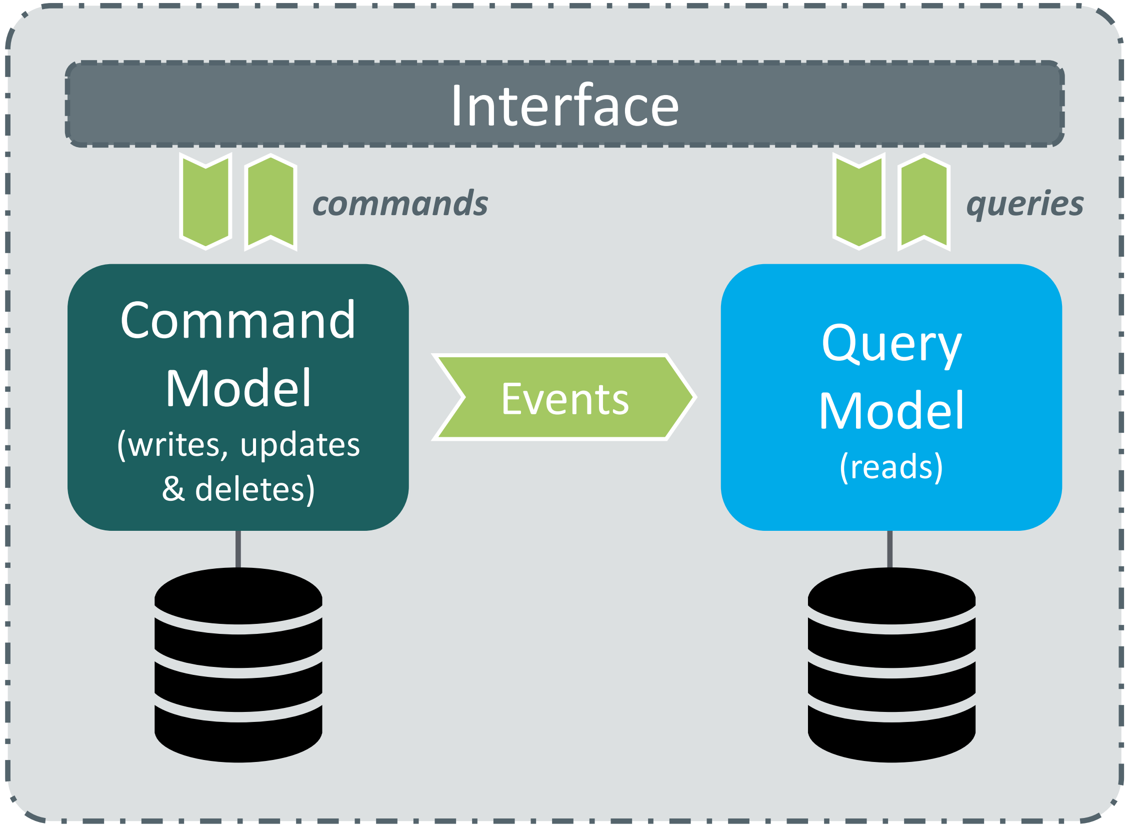
Digital Integration & Access Layer

- The digital integration & access layer act as an abstraction layer between data providers and data consumers, where data consumers can ‘explore’, access and query the data in a consistent manner, at any time at any speed.
- The digital integration & access layer can have physical storage, but only for non-functional reasons (real-time data acquisition; history that cannot be retained; performance reasons).
- The digital integration & access layer offers data integration capabilities, which allows data consumers to extract, integrate and load data into their own application or environment.
- The data transformation, from the source to the target database structure, is done in this layer on behalf of the data consumer.
- The digital integration & access layer is metadata-driven. It supports an approach for defining data services that can be fully reused for other consuming applications and provides insight in the full lineage.
- DIAL takes care of the data delivery agreements between Data Providers and Data Consumers. It should act as a hub by routing the data based on the meta data labels where also the agreements are in stored.

DATA CONSUMERS

- Data consumers are the data users.
- Are required to document the consumption (meta data).
- Applications are intended for a specific business purpose or process and therefore limited in scope. No data without a purpose.
- Determine requirements, which can force providers to offload or retain more data.
- Determine both functional and non-functional requirements

Command and Query Responsibility Segregation (CQRS) explained



CQRS is a software architecture design pattern that separates commands from queries by using two different models. Once separated, they must be kept in sync, which is typically done by publishing events with changes.*

Our operational systems are expected to suffer, as we continue to scale up, do more analytics, and intensive reads. A common application design pattern to overcome this problem is to separate the operational commands and analytical queries (often referred to as writes and reads). Separating the two brings a number of benefits:

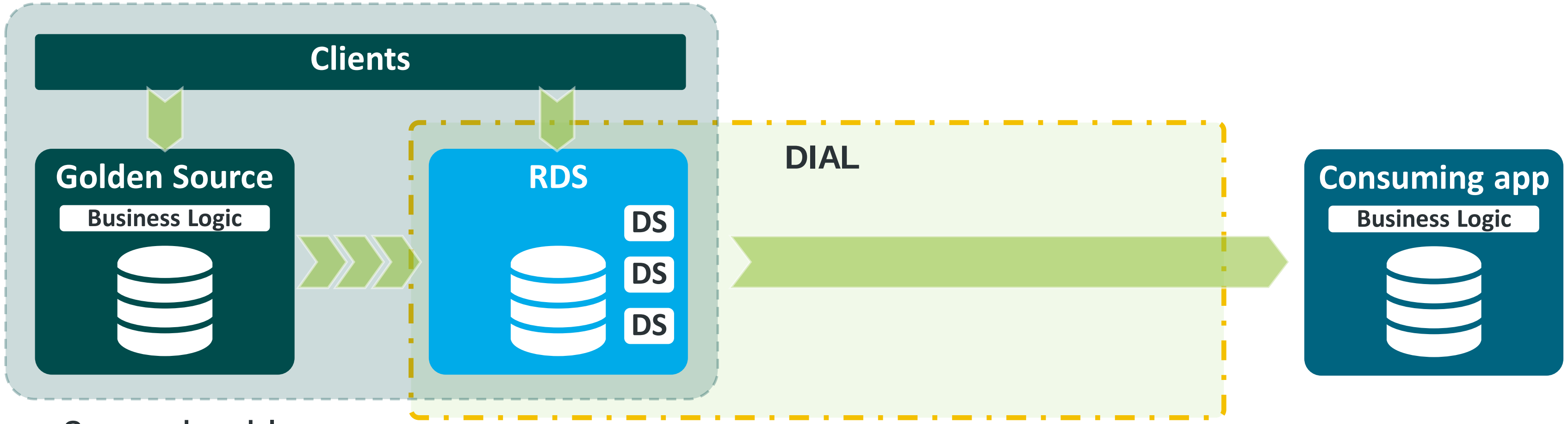
- *By separating you can optimize and choose the best technology for reading data intensively. Reading a database, compared to writing, takes a smaller amount of computing resources.*
- *By separating you are not tied to the same type of database model for both writes and reads. You can leave the write database objectively complex, but for the read database you can optimize for read performance. You can also use the same database, but configure it differently, for example by turning off 'locking'.*
- *If you have different requirements for different use cases, you can even create more than one read database, each with an optimized read model for the specific use case being implemented.*
- *There is no need to scale both the read and write operations simultaneously, so when you are running out of resources, you can scale one or the other.*

** The reads and synchronisation can be subject to different service levels, so variation and different patterns are expected.*

The new model of the architecture is that at least one RDS per application is created whenever other applications want to read the data intensively

The rationale behind DIAL is to get ready for a world of intensive data consumption. Instead of integrating all data into a silo or monolith, we have choosen for an unified approach of data distributing and integrating data between the application. By retaining the original context and capturing all data, we facilitate both operational and analytical-based consumption. Additionally, the architecture remains flexible, because applications are loosely coupled.

The change that DIAL introduces is to split the application model into models for update and read, which refers to as Command and Query. The rationale is facilitate the intensive reads we anticipate in a modern world of data consumption.



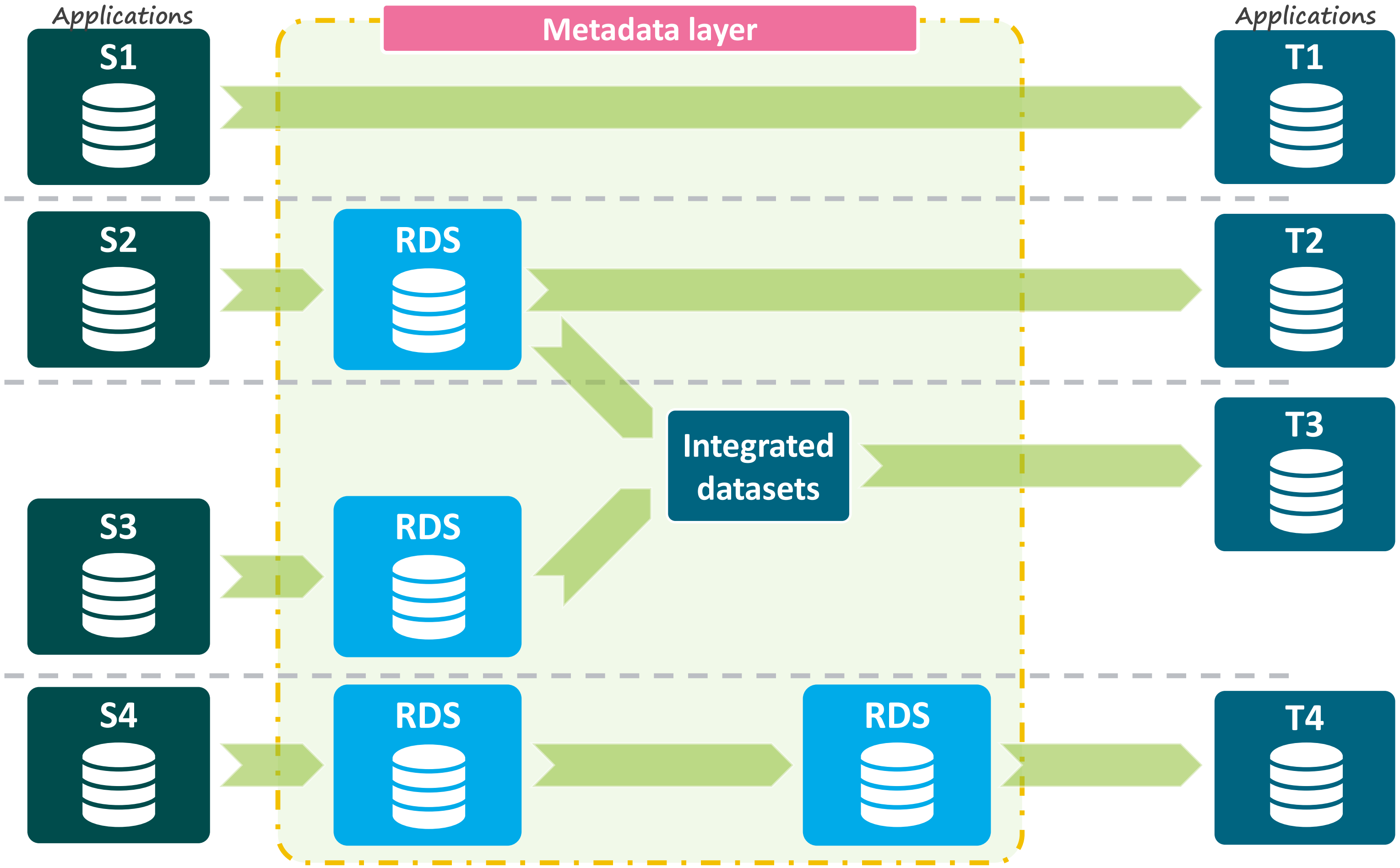
On the consumption side we expect a new application, and thus a new application data store. The data is stored in a different structure, because the context is different.

Command model executes business logic and receives input from the presentation layer

Query model that facilitates intensively reading data(sets)

Most important is to know the meaning of data, its quality, where it's coming from and where consumption takes place

The DIAL takes care for the distribution of data. How the data is transported using within the reliable environment and how integrity & service level through the chain is maintain is more a technical discussion. Most important to know for the business is the meaning of data, its quality, where it's coming from and where and what for consumption takes place.



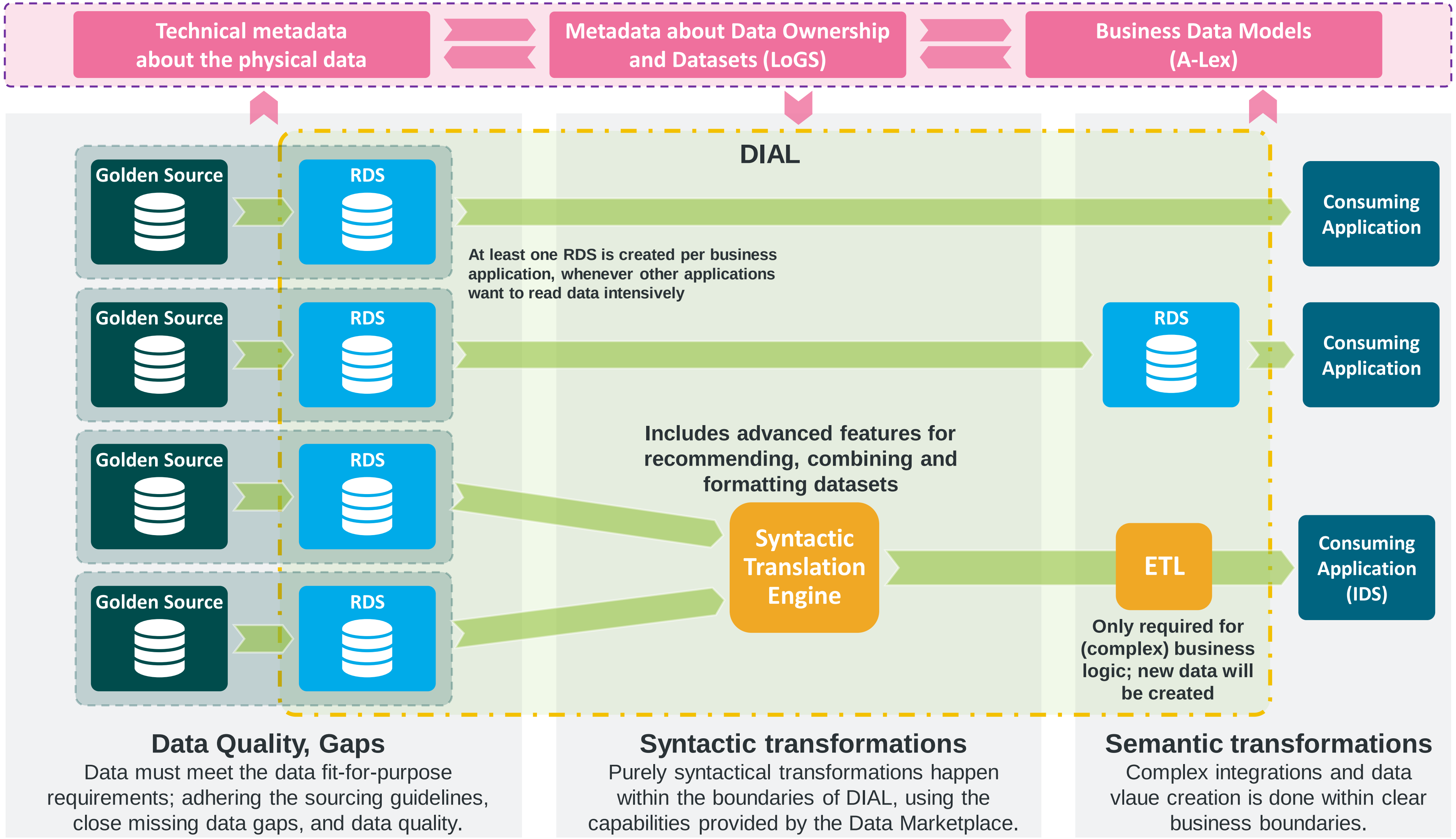
Applications directly exchanging can data be valid as an exception, but only when metadata is captured, knowing that applications are connected and having a dependency.

Applications directly exchanging data via a Read Data Store (RDS) is valid and very useful to separate the read and write performance.

Applications consuming and integrating from multiple RDS's is also valid when data is required to be combined from different sources.

RDS's, if required, can also exchange and 'cache' the data to solve latency or performance requirements.

Our target state architecture is to democratize data consumption through intelligent-assisted data generation


Technical metadata about the physical data
Metadata about Data Ownership and Datasets (LoGS)
Business Data Models (A-Lex)
DIAL
Golden Source
RDS
Golden Source
RDS
Golden Source
RDS
Golden Source
RDS
RDS
Syntactic Translation Engine
ETL
Consuming Application
Consuming Application
Consuming Application (IDS)

At least one RDS is created per business application, whenever other applications want to read data intensively

Includes advanced features for recommending, combining and formatting datasets

Only required for (complex) business logic; new data will be created

Data Quality, Gaps

Data must meet the data fit-for-purpose requirements; adhering the sourcing guidelines, close missing data gaps, and data quality.

Syntactic transformations

Purely syntactical transformations happen within the boundaries of DIAL, using the capabilities provided by the Data Marketplace.

Semantic transformations

Complex integrations and data vlaue creation is done within clear business boundaries.

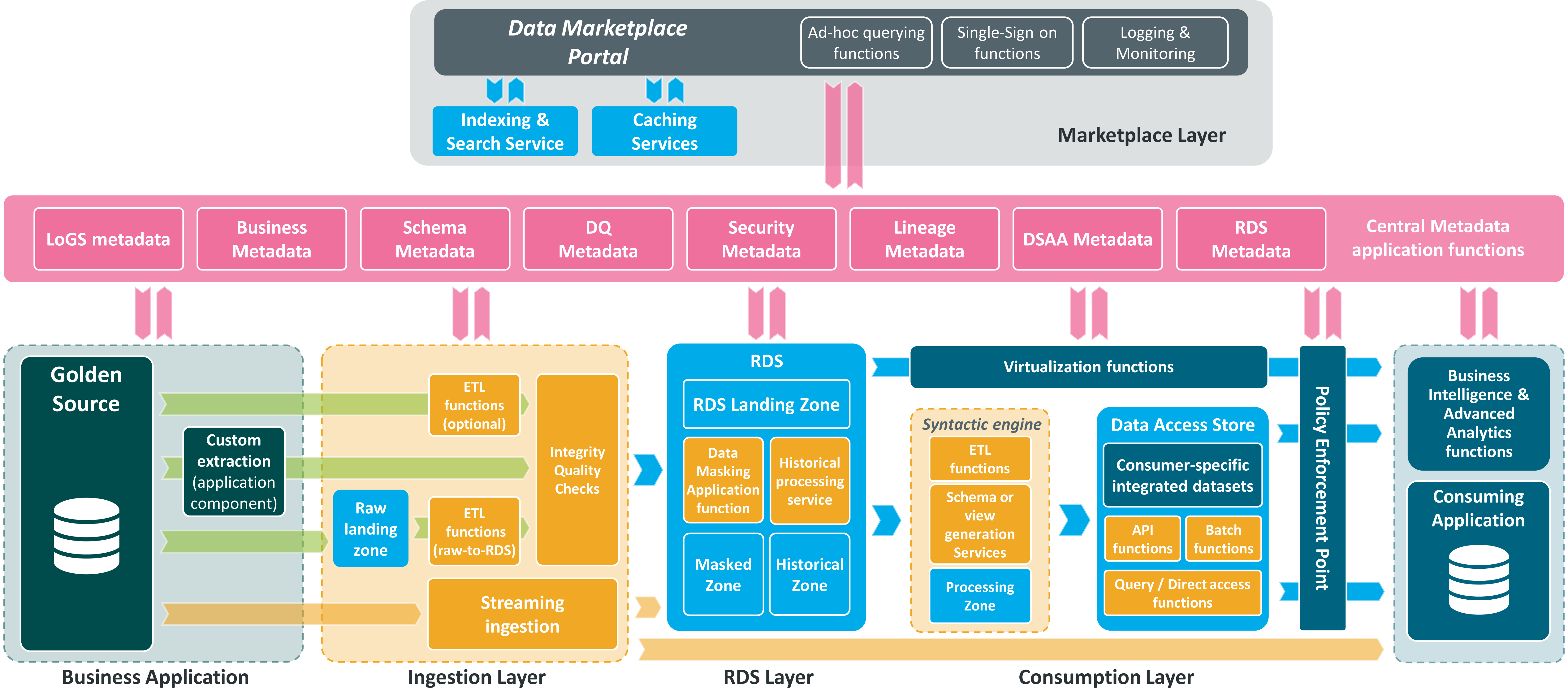
Metadata Layer; managed by the business and the critical glue to stitch DIAL and Marketplace together.

Applications directly exchanging data via an RDS is valid and useful for separating the read and write performance.

If required, RDSs can also exchange and 'cache' data for solving latency and performance issues.

Applications consuming, combining and integrating from multiple RDSs is valid. DIAL in this scenario takes care of the distribution of data.

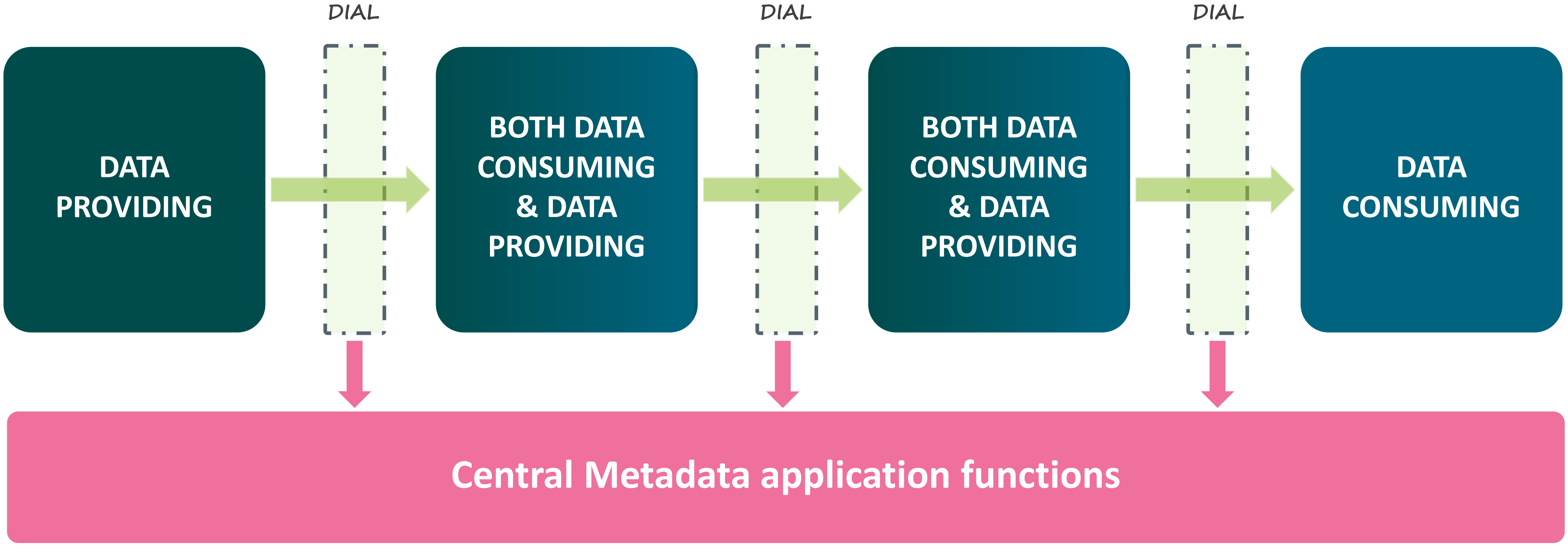
Decomposition of Data Distribution Capability



ABN·AMRO

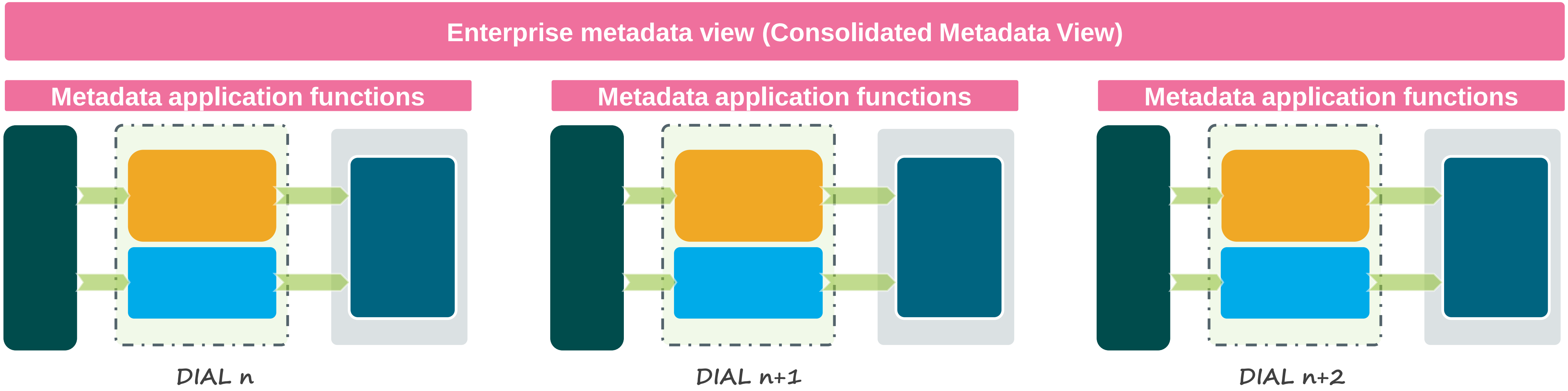
The Digital Integration & Access Layer is used repeatedly when applications play both the role of Data Providing and Data Consuming

The “Digital Integration & Access Layer” is used for connecting Data Providing and Data Consuming. It uses different techniques mainly based on the (non-)functional requirements of the Data Consuming. The Digital Integration & Access Layer is used repeatedly in case of applications transfer data across different domains.


DIAL
DIAL
DIAL
DATA PROVIDING
BOTH DATA CONSUMING & DATA PROVIDING
BOTH DATA CONSUMING & DATA PROVIDING
DATA CONSUMING
Central Metadata application functions

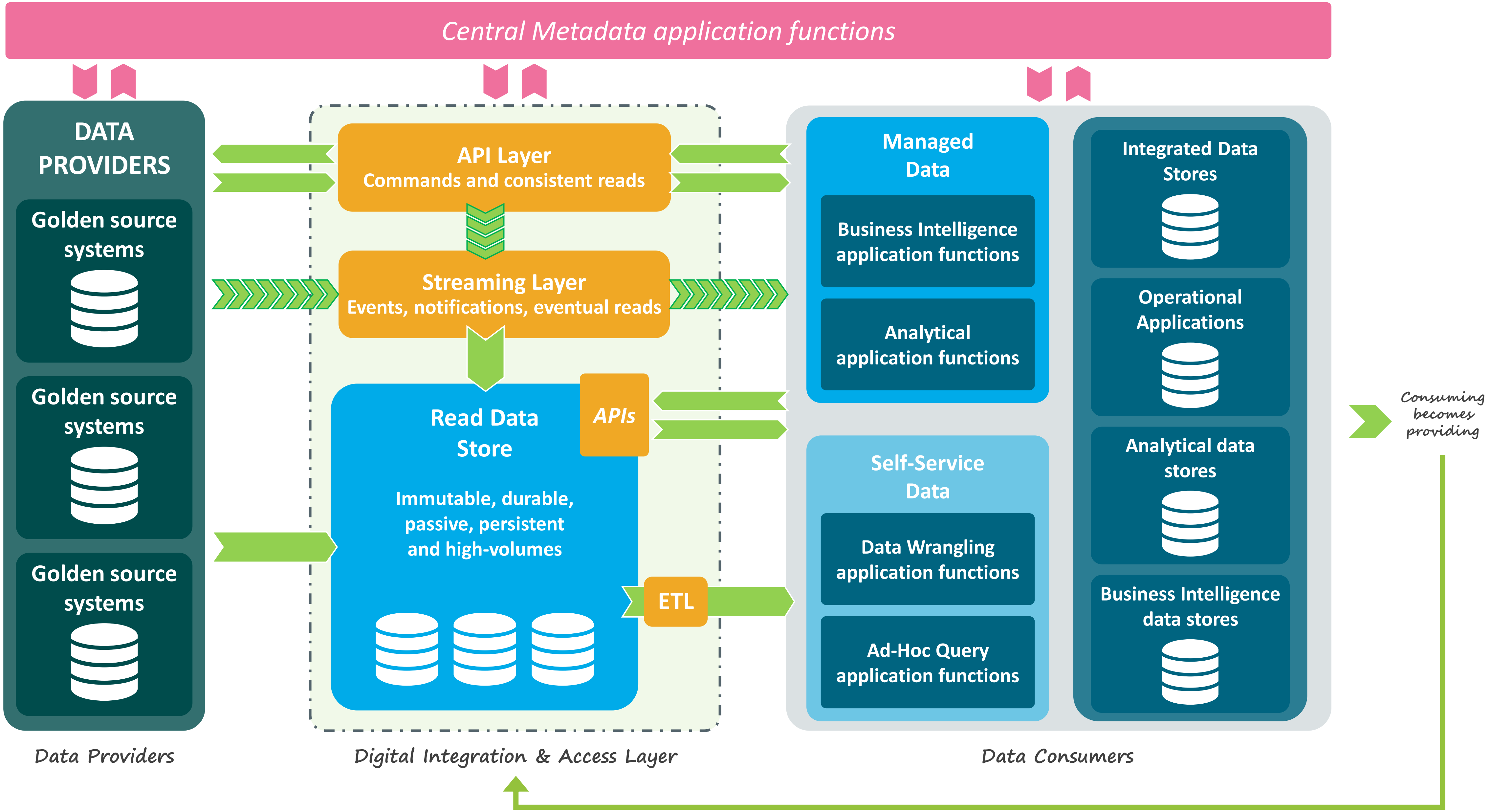
Stitching all Metadata together to create a single point of view of all data is the key effort

The expectation is that many tools from different vendors, each deployed at potentially different locations, will be used for the Metadata storage. Key effort is the ability to stitch all Metadata together to create a single point of view of all data. Automating and integrating the population of this repository can be achieved using the same integration techniques of DIAL.



Enterprise metadata view (Consolidated Metadata View)
Metadata application functions
Metadata application functions
Metadata application functions
DIAL n
DIAL n+1
DIAL n+2

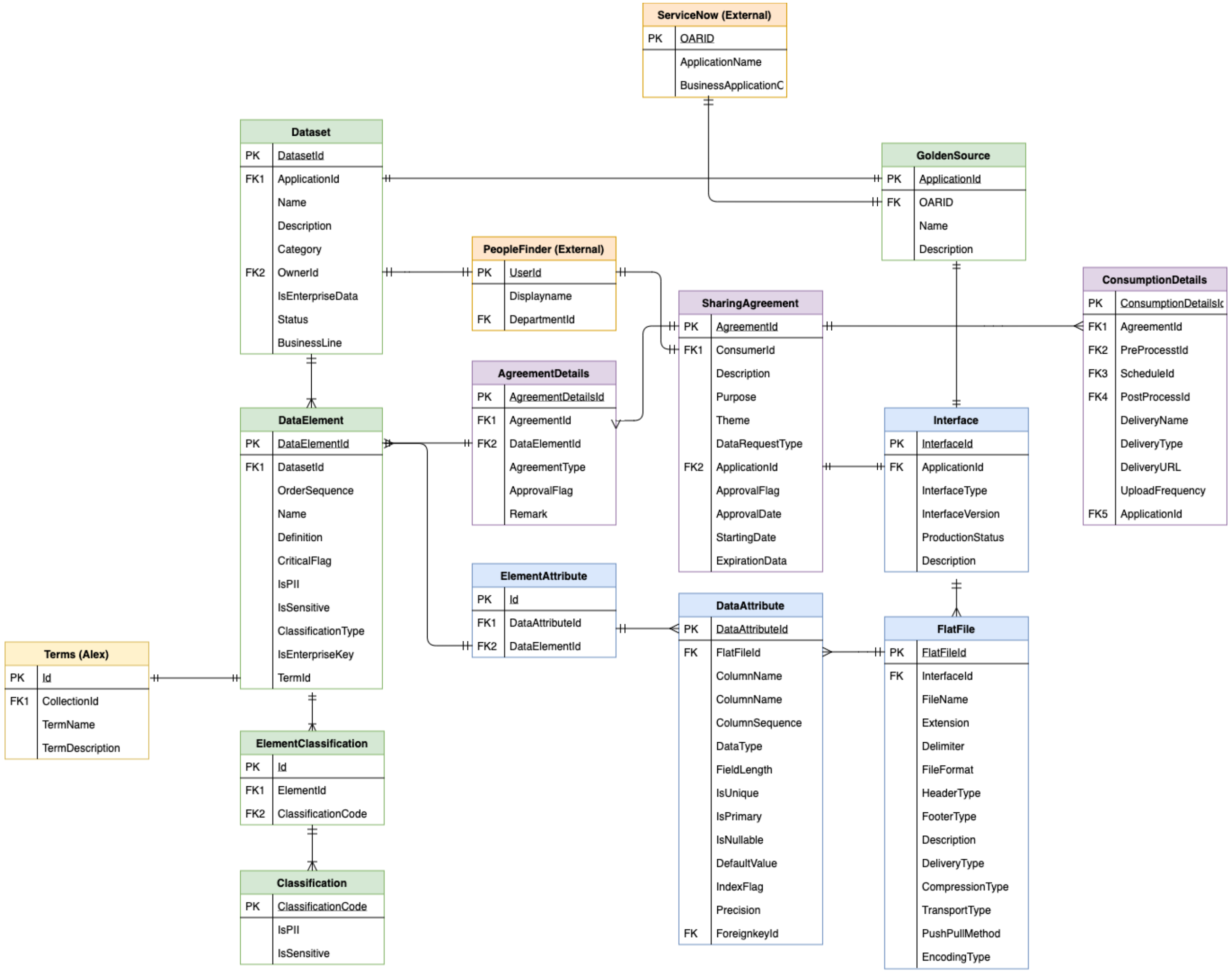
The Digital Integration & Access Layer reference architecture


Central Metadata application functions
DATA PROVIDERS
Golden source systems
Golden source systems
Golden source systems
API Layer
Commands and consistent reads
Streaming Layer
Events, notifications, eventual reads
Read Data Store
APIs
Immutable, durable, passive, persistent and high-volumes
ETL
Managed Data
Business Intelligence application functions
Analytical application functions
Self-Service Data
Data Wrangling application functions
Ad-Hoc Query application functions
Integrated Data Stores
Operational Applications
Analytical data stores
Business Intelligence data stores
Consuming becomes providing
Data Providers
Digital Integration & Access Layer
Data Consumers

Meta model

Developed with support from Microsoft:

- All data ownership, enterprise classifications and security captured by the green entities
- Technical interface metadata captured by the blue entities
- Fine-grained data sharing contracts via purple entities
- Relationships to our enterprise ontologies
- Lineage, Streaming, APIs are WIP



	ServiceNow (External)
PK	OARID
	ApplicationName
	BusinessApplicationC

	Dataset
PK	DatasetId
FK1	ApplicationId
	Name
	Description
	Category
FK2	OwnerId
	IsEnterpriseData
	Status
	BusinessLine

	GoldenSource
PK	ApplicationId
FK	OARID
	Name
	Description

	PeopleFinder (External)
PK	UserId
	Displayname
FK	DepartmentId

	ConsumptionDetails
PK	ConsumptionDetailsIc
FK1	AgreementId
FK2	PreProcesstId
FK3	ScheduleId
FK4	PostProcessId
	DeliveryName
	DeliveryType
	DeliveryURL
	UploadFrequency
FK5	ApplicationId

	SharingAgreement
PK	AgreementId
FK1	ConsumerId
	Description
	Purpose
	Theme
	DataRequestType
FK2	ApplicationId
	ApprovalFlag
	ApprovalDate
	StartingDate
	ExpirationData

	AgreementDetails
PK	AgreementDetailsId
FK1	AgreementId
FK2	DataElementId
	AgreementType
	ApprovalFlag
	Remark

	DataElement
PK	DataElementId
FK1	DatasetId
	OrderSequence
	Name
	Definition
	CriticalFlag
	IsPII
	IsSensitive
	ClassificationType
	IsEnterpriseKey
	TermId

	Interface
PK	InterfaceId
FK	ApplicationId
	InterfaceType
	InterfaceVersion
	ProductionStatus
	Description

	ElementAttribute
PK	Id
FK1	DataAttributeId
FK2	DataElementId

	DataAttribute
PK	DataAttributeId
FK	FlatFileId
	ColumnName
	ColumnName
	ColumnSequence
	DataType
	FieldLength
	IsUnique
	IsPrimary
	IsNullable
	DefaultValue
	IndexFlag
	Precision
FK	ForeignkeyId

	FlatFile
PK	FlatFileId
FK	InterfaceId
	FileName
	Extension
	Delimiter
	FileFormat
	HeaderType
	FooterType
	Description
	DeliveryType
	CompressionType
	TransportType
	PushPullMethod
	EncodingType

	Terms (Alex)
PK	Id
FK1	CollectionId
	TermName
	TermDescription

	ElementClassification
PK	Id
FK1	ElementId
FK2	ClassificationCode

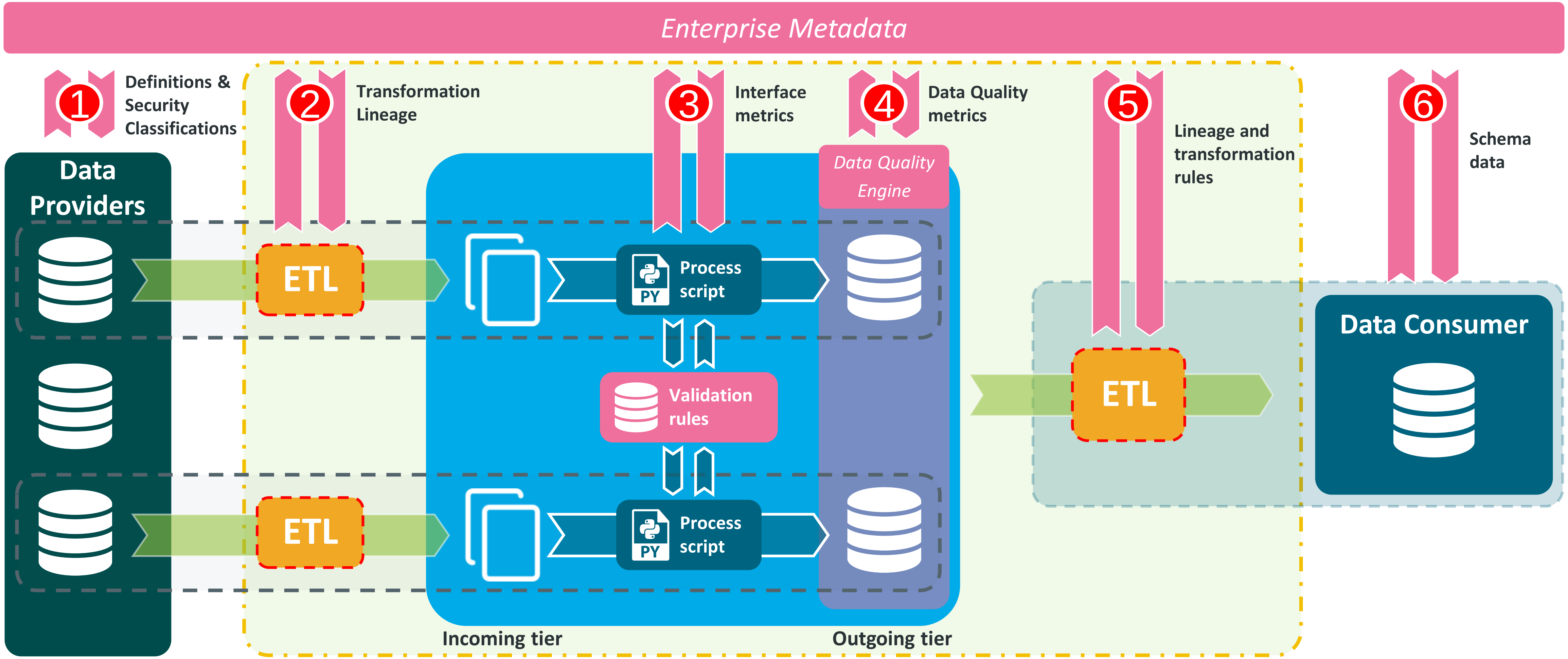
	Classification
PK	ClassificationCode
	IsPII
	IsSensitive

Other interesting links

- *ABN AMRO’s Integration LinkedIn post: https://www.linkedin.com/pulse/abn-amros-data-integration-architecture-piethein-strengholt%2F/*
- *Look to Data Management at Scale: https://learning.oreilly.com/library/view/data-management-at/9781492054771/*
- *Data mesh: https://martinfowler.com/articles/data-monolith-to-mesh.html*

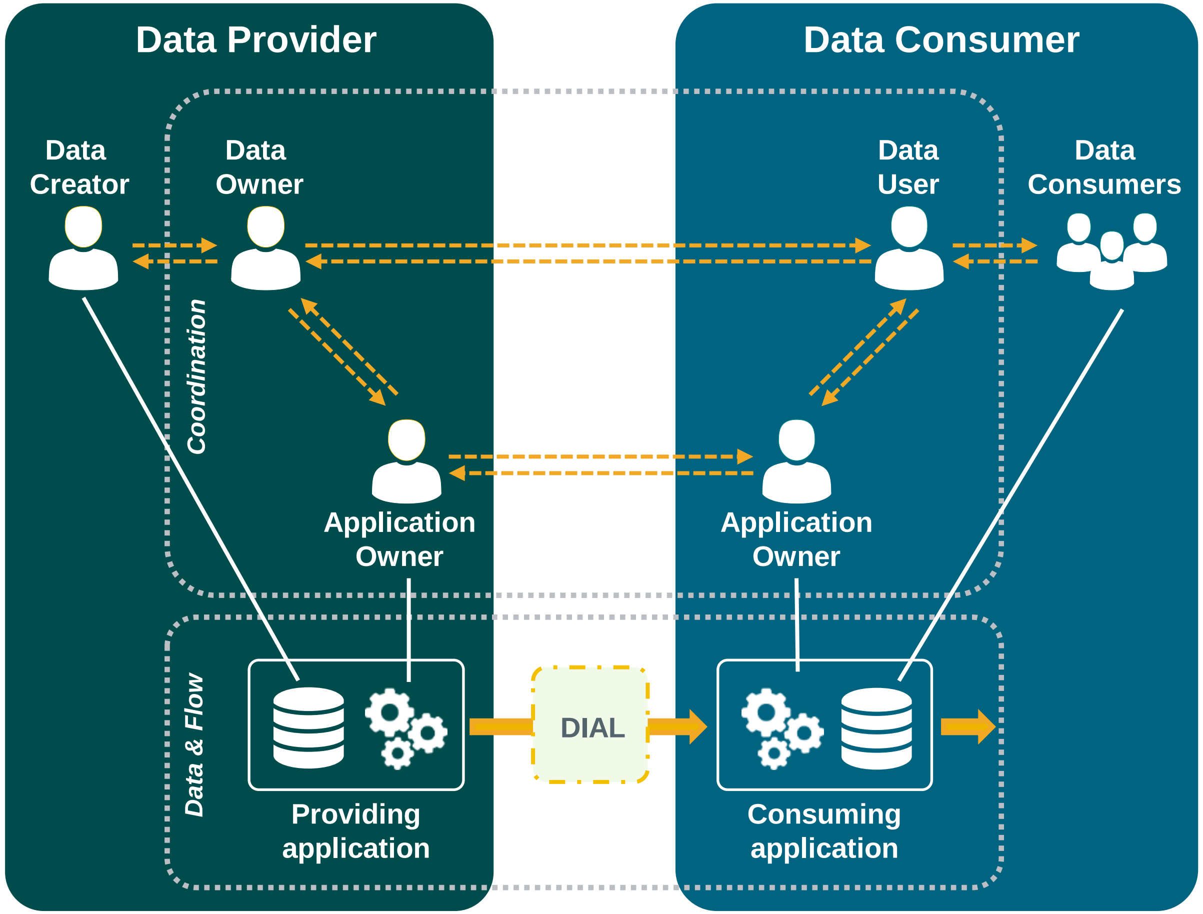
Metadata is generated from the RDS and many surrounding application functions

Once metadata integration is engineered correctly, you can automatically extract performance indicators of the interfaces, data quality, the lineage and transformation logic applied on the data, the schema data of the sources, etc.



Governance definitions used within the context of the FSA

To achieve an unambiguous defined Data Governance, the concepts Data Providing and Data Consuming are introduced. Different roles are defined, facilitating these two concepts.



**Also see ABN AMRO Data Quality Policy Version 3.0 – January, 2017*

Data Creator
A creator is a role of an internal or external party who creates the data as agreed with the Data Owner.

Data Owner*
A Data Owner is a role of an individual employee within the bank who has the ability to verify accuracy of Data and has the accountability to manage the data.

Data User*
A Data User is a role of an individual employee within the bank who intends to use data for a specific purpose and has the accountability to set requirements on the data.
Explanatory notes: In the consuming context, the Data User is accountable that the requirements are known by the Data Owner. These requirements will be reflected in the data transformation between the data providing application and the data consuming application by order of the Data Owner and Data User respectively.

Data Consumer
A data consumer is a role of an internal or external party who uses data as intended by the Data Owner and Data User.

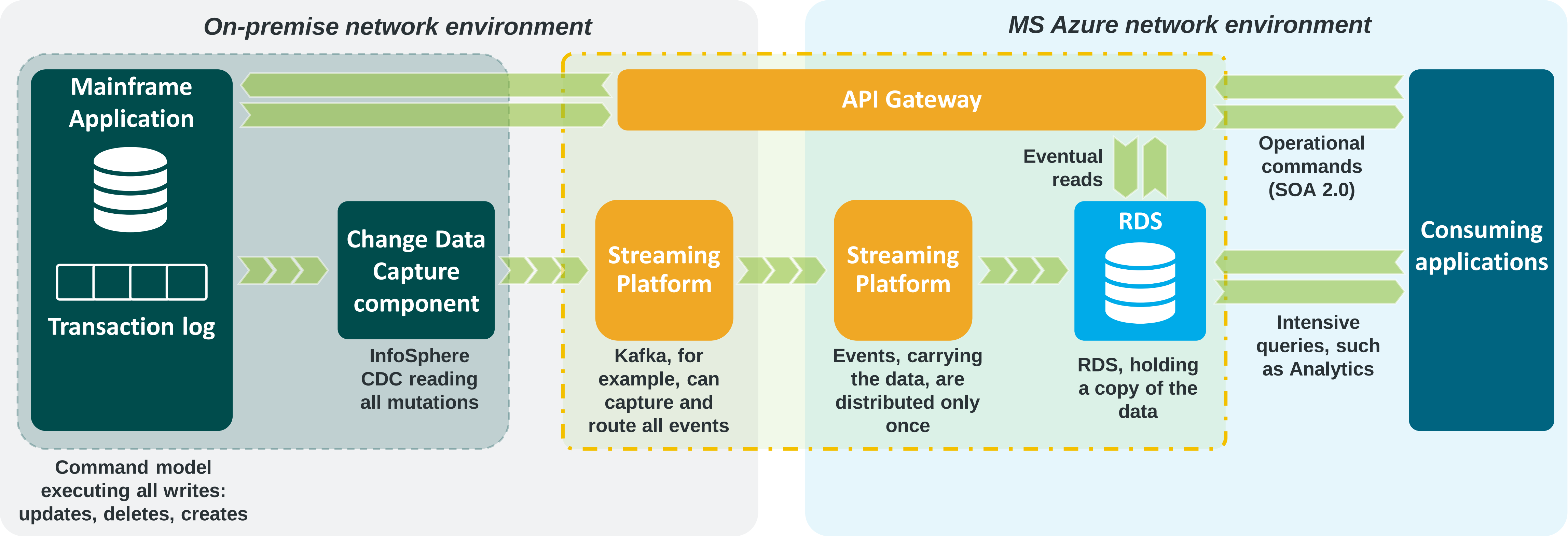
Application Owner
The application owner maintains the core of application & its interfaces. The application owner is responsible for the business delivery, functioning and services of the application, the maintenance of the application information and access control.
Explanatory notes: In the consuming context the Data User is accountable for the transformation (ETL) from the source (providing) to the target (consuming) database structure, and orders implementation by the (consuming) Application Owner. The requirements for this transformation are defined by the Data Users and agreed with the Data Owners.

Golden Source*
Golden Source is the application where the data is created (data origination), changed or deleted and provided (distributed) from.

Consuming Application
Consuming application is the application where the data is stored/integrated for specific use.

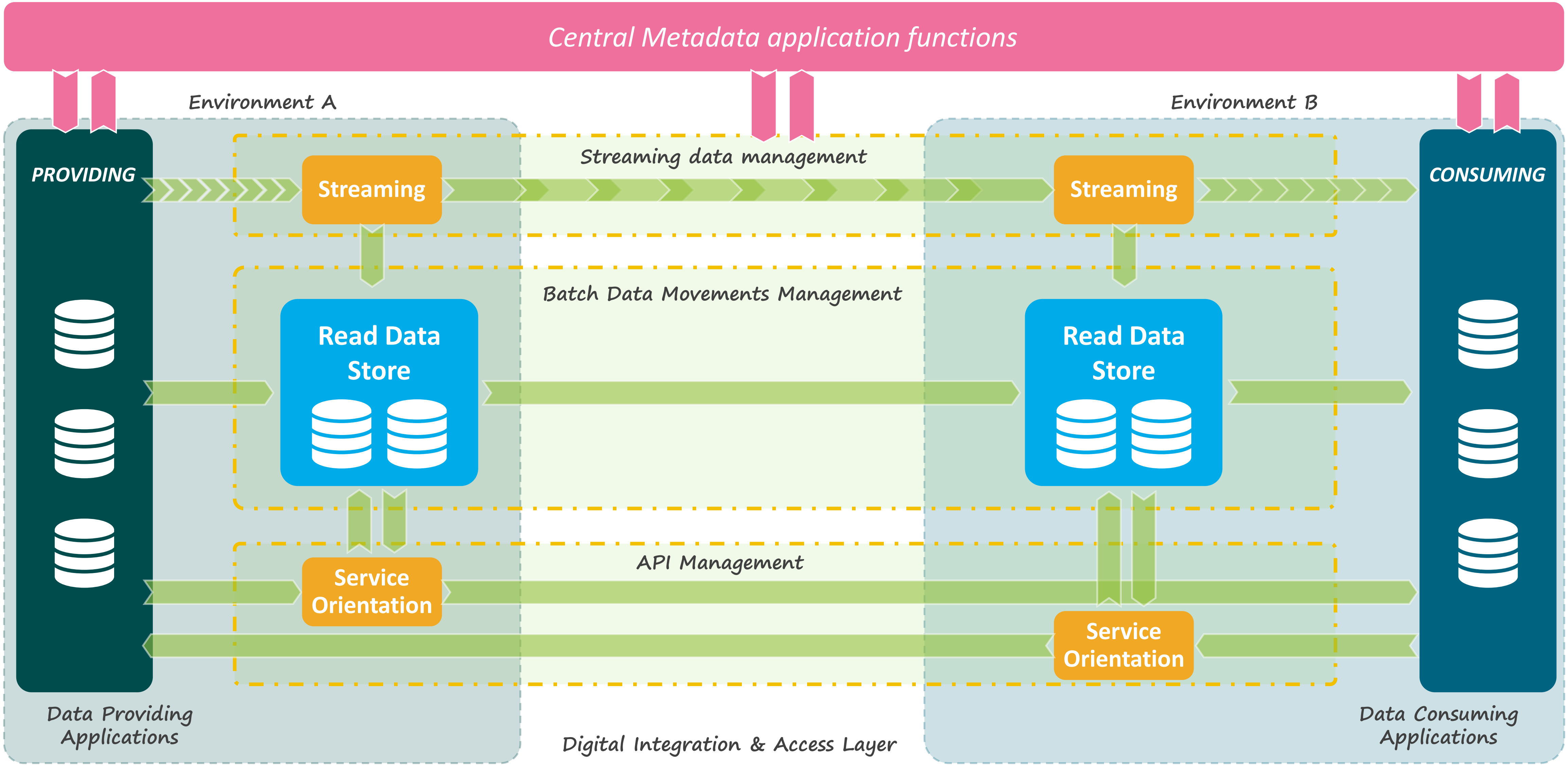
By off-loading through read caches we can reduce our costs significantly, accelerate our migration and optimize for specific workloads

A benefit of offloading and replicating data from the mainframes, is that data is only copied over the network ***once*** *and then distributed within Azure to the different applications. Reads are captured, for example with InfoSphere CDC, and then distributed, for example with Kafka, to an RDS on MS Azure. Any application requiring the same data can read it directly from there. This will be huge cost saving, because intensive reads are the majority of all workloads. The mainframe in this model will be only used for the operational commands (update, delete and create statements)*



On-premise network environment
MS Azure network environment
Mainframe Application
Transaction log
Change Data Capture component
InfoSphere CDC reading all mutations
API Gateway
Streaming Platform
Kafka, for example, can capture and route all events
Streaming Platform
Events, carrying the data, are distributed only once
Eventual reads
RDS
RDS, holding a copy of the data
Operational commands (SOA 2.0)
Intensive queries, such as Analytics
Consuming applications
Command model executing all writes: updates, deletes, creates

DIAL reference Architecture in a hybrid Cloud scenario


Central Metadata application functions
Environment A
Environment B
PROVIDING
CONSUMING
Streaming data management
Streaming
Streaming
Batch Data Movements Management
Read Data Store
Read Data Store
API Management
Service Orientation
Service Orientation
Data Providing Applications
Data Consuming Applications
Digital Integration & Access Layer

ABN·AMRO

Thank You

Informatica™