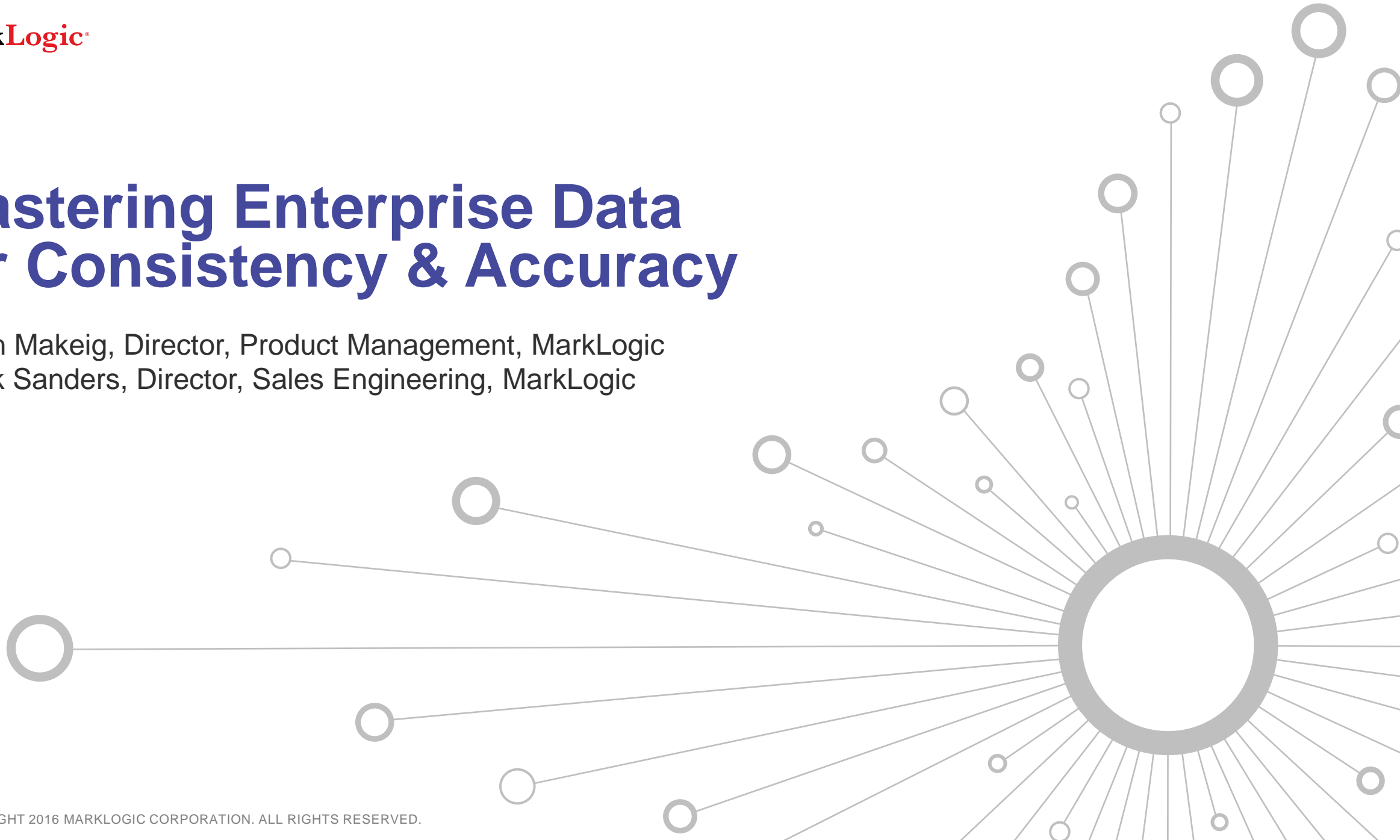


Mastering Enterprise Data for Consistency & Accuracy

Justin Makeig, Director, Product Management, MarkLogic
Frank Sanders, Director, Sales Engineering, MarkLogic



Hello, my name is Justin

- Product Manager for 8+ years at MarkLogic
- Background in consulting and web development
- Focus on software architecture, APIs, and data integration

jmakeig@marklogic.com

<https://github.com/jmakeig>

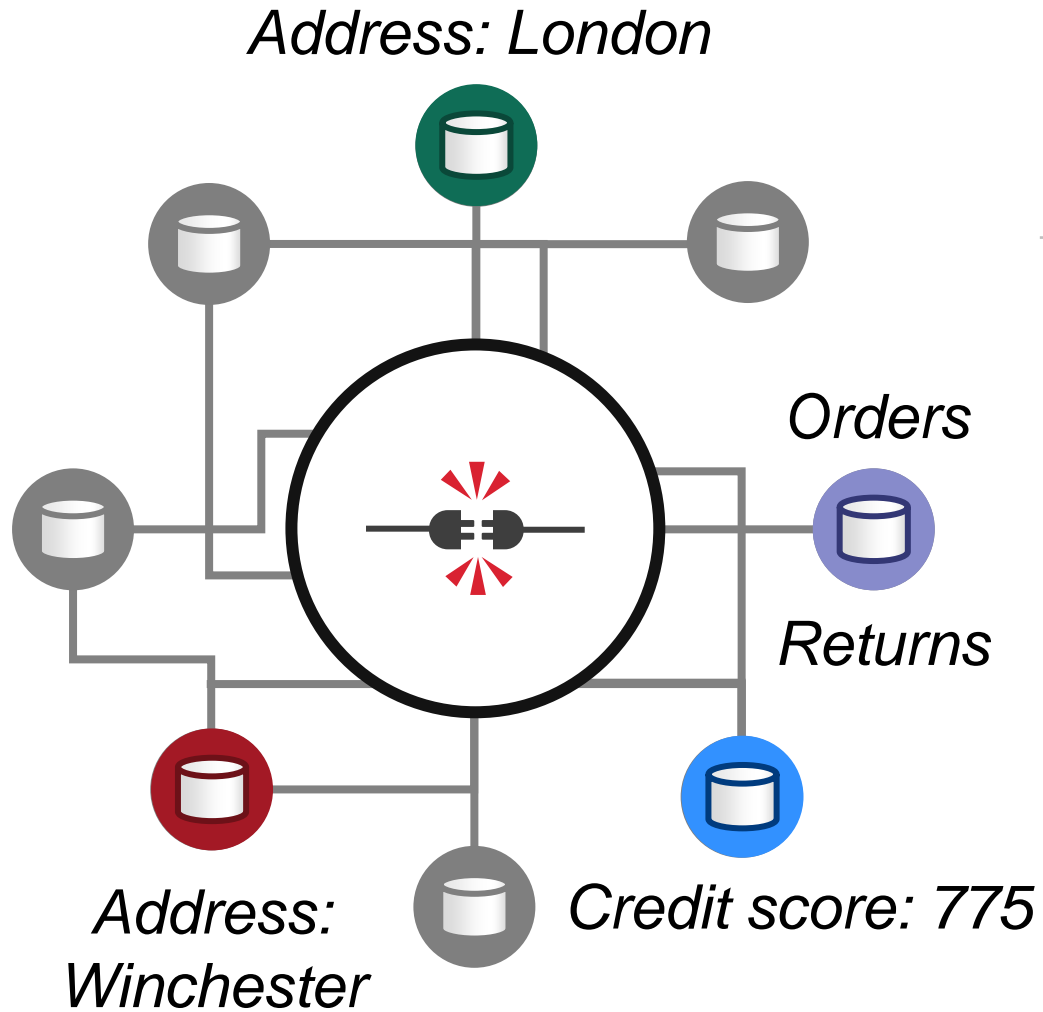
Hello, my name is Frank

- 9+ Years with MarkLogic
- Led architecture, implementation at many of MarkLogic's largest customers

fsanders@marklogic.com

Agenda

- What is Master Data and why should I care?
- Traditional approaches to Master Data Management
- How MarkLogic changes MDM
- Use cases: Techniques and tips
- Q&A

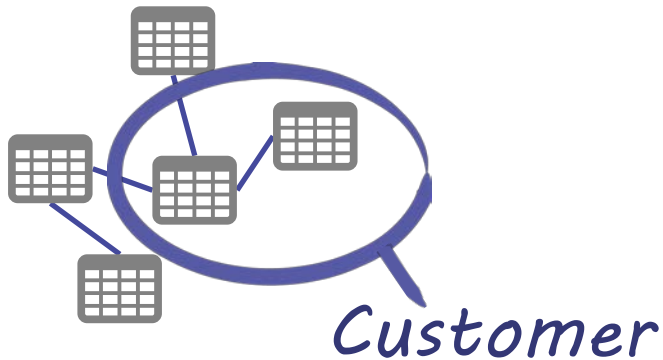


“Christy Haragan”

- What is “Christy Haragan”? Who decided that?
- What is our relationship with her?
- What has she purchased? Returned? Received support on?
- Where does she live? With whom? How do I contact her? When have we contacted her?
- Is she a “high-value” customer? What risk does she represent?
- Who is authorized to see Christy’s data? Who can change it? Under what conditions?

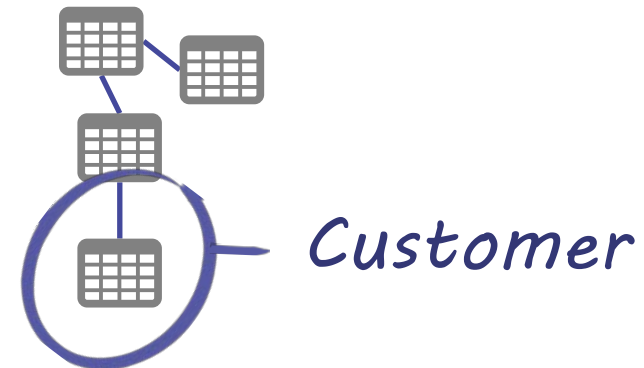
Relational strips important context from data

System A



```
CREATE TABLE Customers (  
  CustID          NUMBER(6) PRIMARY KEY,  
  StartDate      DATE NOT NULL  
  PartyID        NUMBER(6)  
  ...  
)
```

System B



```
CREATE TABLE CUST_MASTER (  
  CID            VARCHAR2(40) PRIMARY KEY,  
  Evt_Dt        TIMESTAMP  
  Cst_Type      VARCHAR2(120)  
  ...  
)
```

TRADITIONAL APPROACHES TO MDM



Master Data Registry



Customer: Christy Haragan

← ERP • PARTY_MASTER • HARA_C

← CRM • customers_1 • 100067

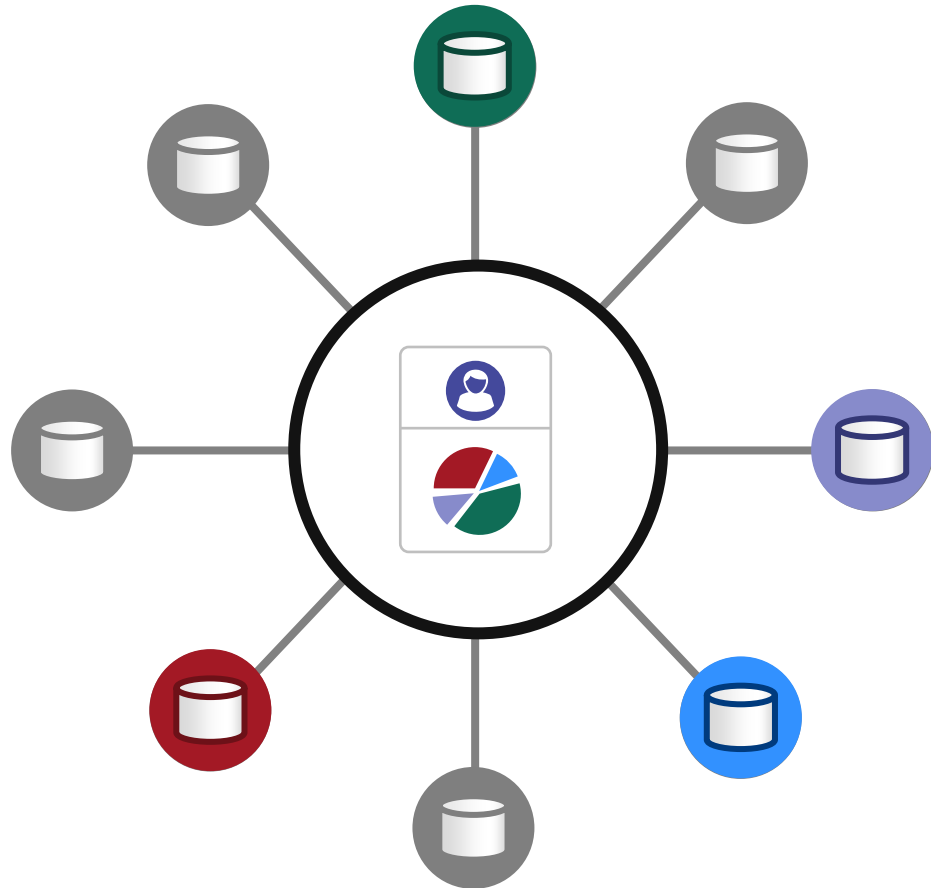
← Credit • http://... • 9930-221

← CRM2 • refcustview • H_3325

Repository of *references* to records in source systems

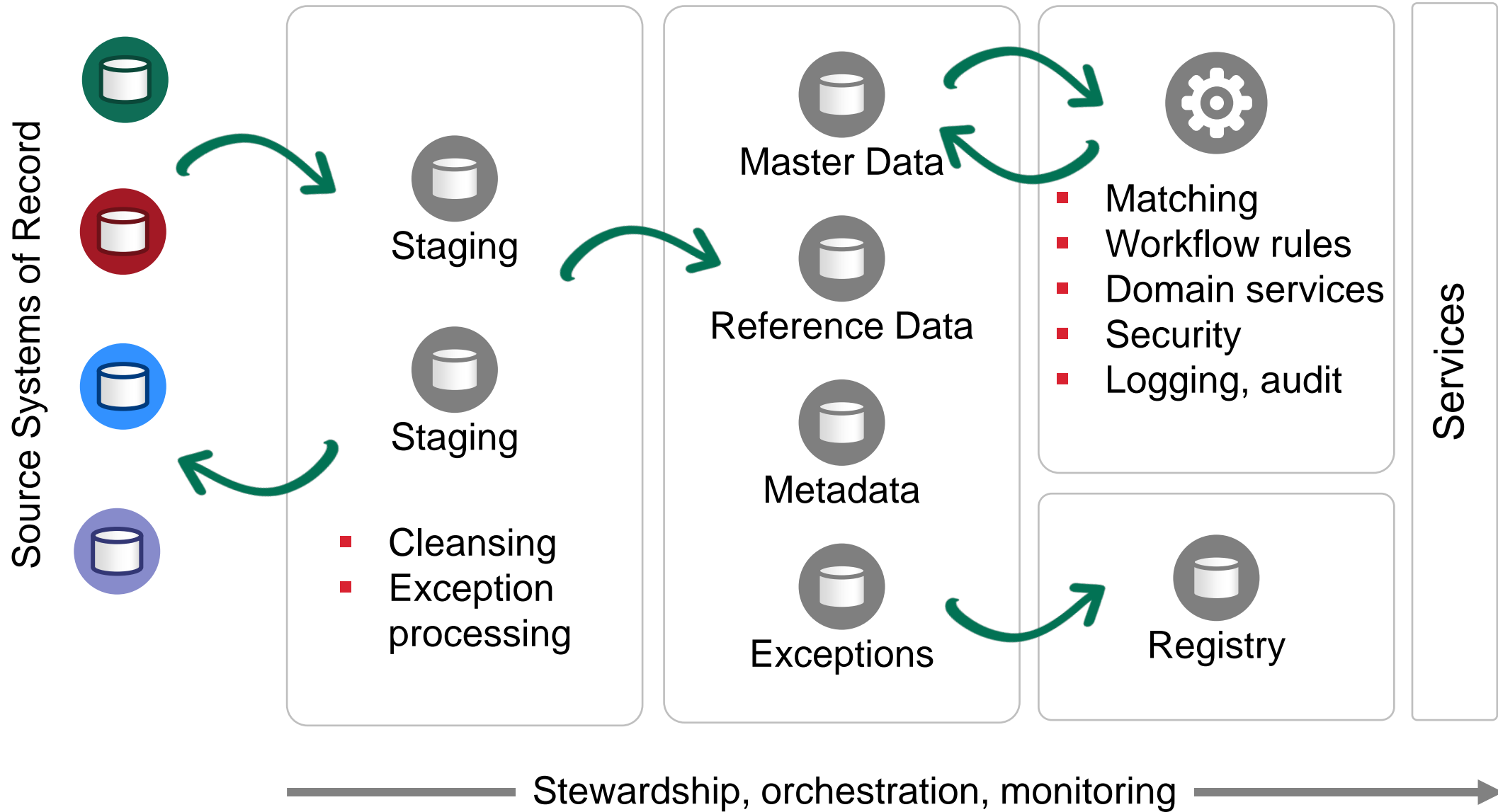
- ✓ Easy to start
- ✓ Clear separation from live systems
- ✗ Difficult to query, update
- ✗ Separation of context and data
- ✗ Dependence on source systems for quality, governance, security

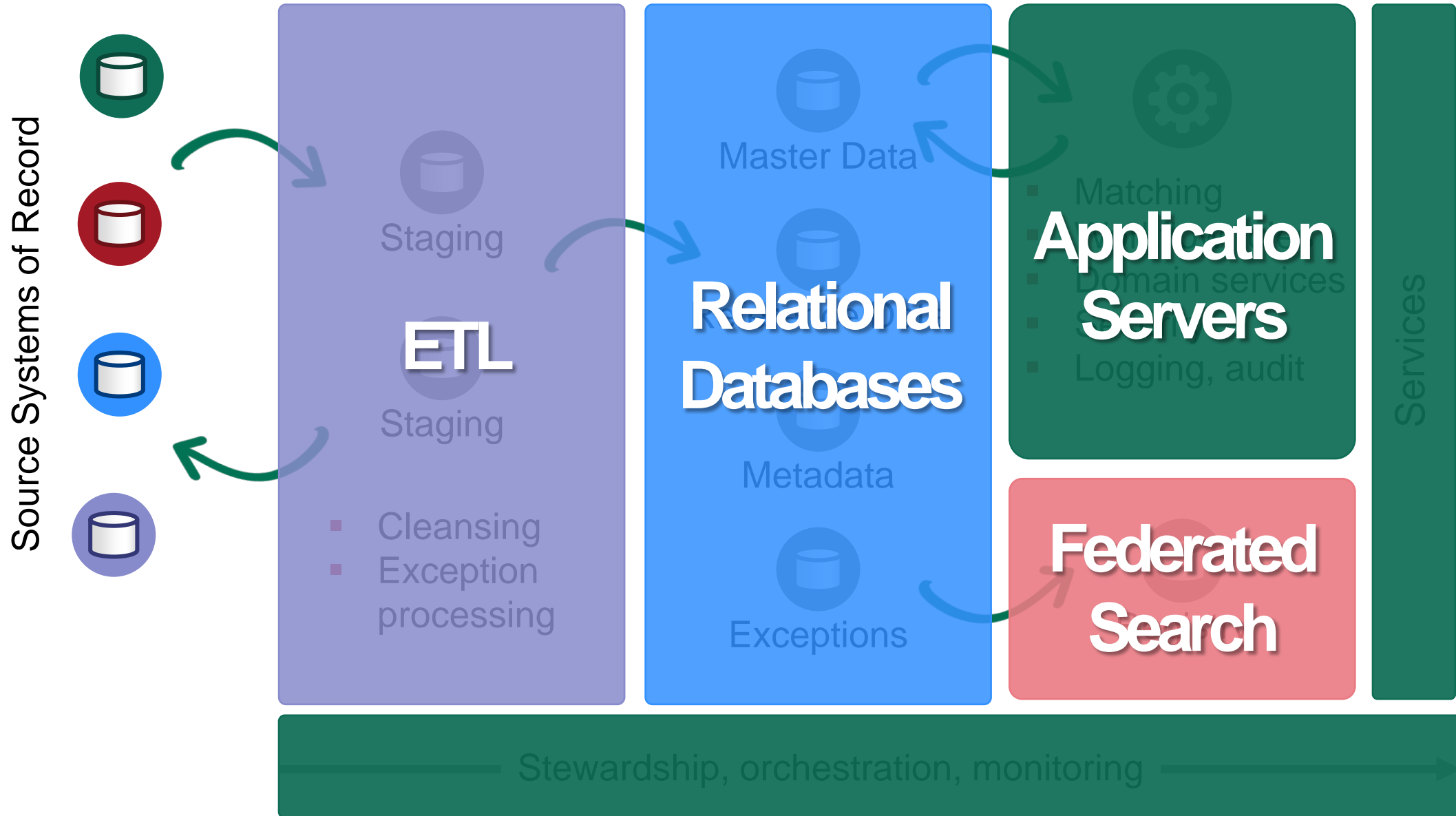
Master Data Hub



Aggregation of data from source systems

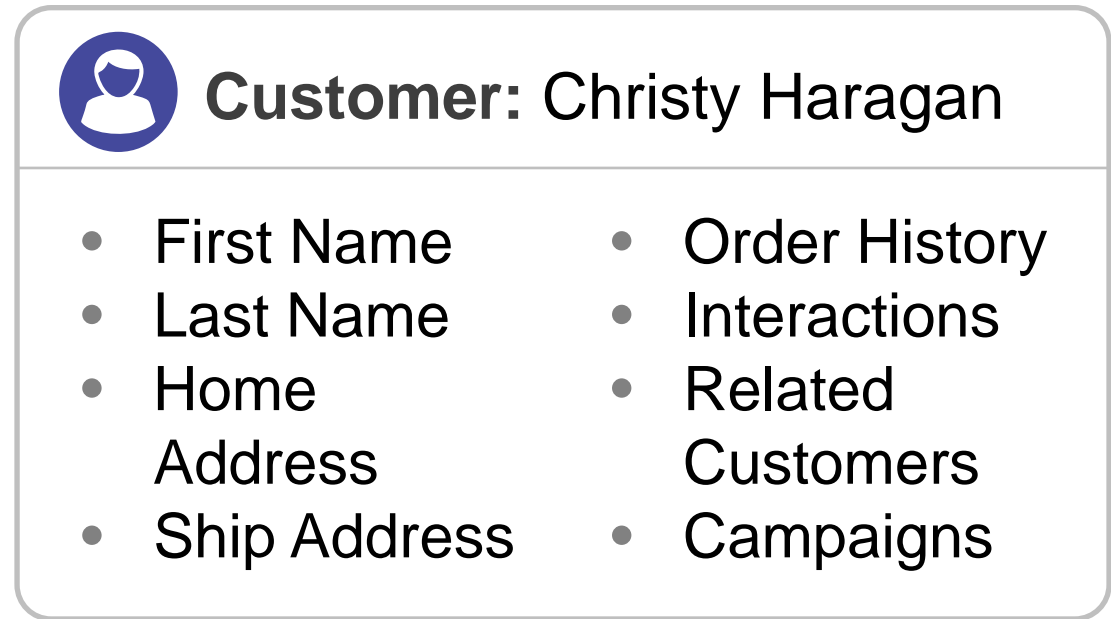
- ✓ Authoritative, bi-directional
- ✓ Centralized quality, governance, security
- ✓ Rich queries, aggregates, enrichment, and transactions
- ✗ Technical canonicalization
- ✗ Political collaboration





Shortcomings of Traditional MDM Approaches

- Canonical models require political and technical alignment
- Single version of the truth, but which version?
- Provenance: show your work
- Changing requirements, changing data
- “Unstructured”, unknown data



A customer profile card for Christy Haragan. The card has a header with a person icon and the name. Below the header is a list of attributes and related items, organized into two columns.

Customer: Christy Haragan

- First Name
- Last Name
- Home Address
- Ship Address
- Order History
- Interactions
- Related Customers
- Campaigns

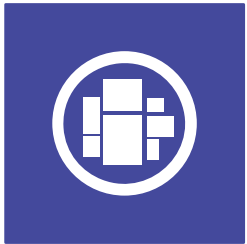


Implications of Traditional MDM Approaches

- **Time to Value:** Average ROI for an MDM implementation is *3 years*
- **Inflexibility:** Snapshot is unable to support multiple Business Units, domains, and changing requirements
- **Cost:** ETL is brittle, expensive and time-consuming to build and maintain
- **Accountability:** Difficult or impossible to explain past decisions on current data
- **Complexity:** Orchestration, governance, and security across many moving parts

A NEW APPROACH TO MASTER DATA MANAGEMENT





FLEXIBLE
DATA MODEL

Capture data and context
without having to rigorously
model upfront



UNIVERSAL
INDEXING

Discover data and project
views of business entities in
real-time

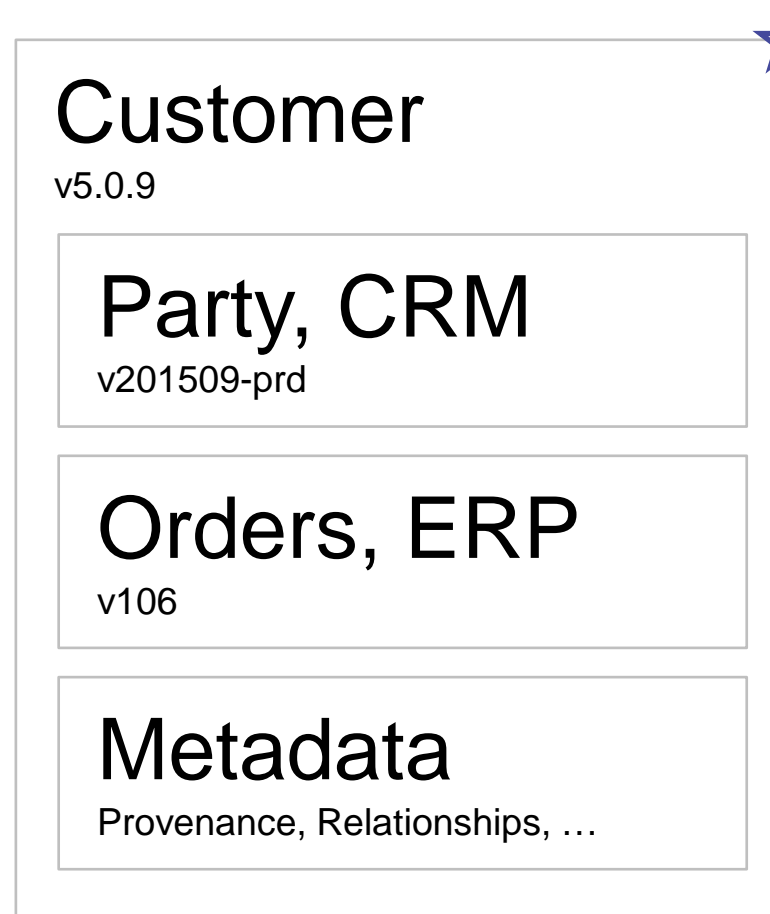


TRUSTED
MANAGEMENT

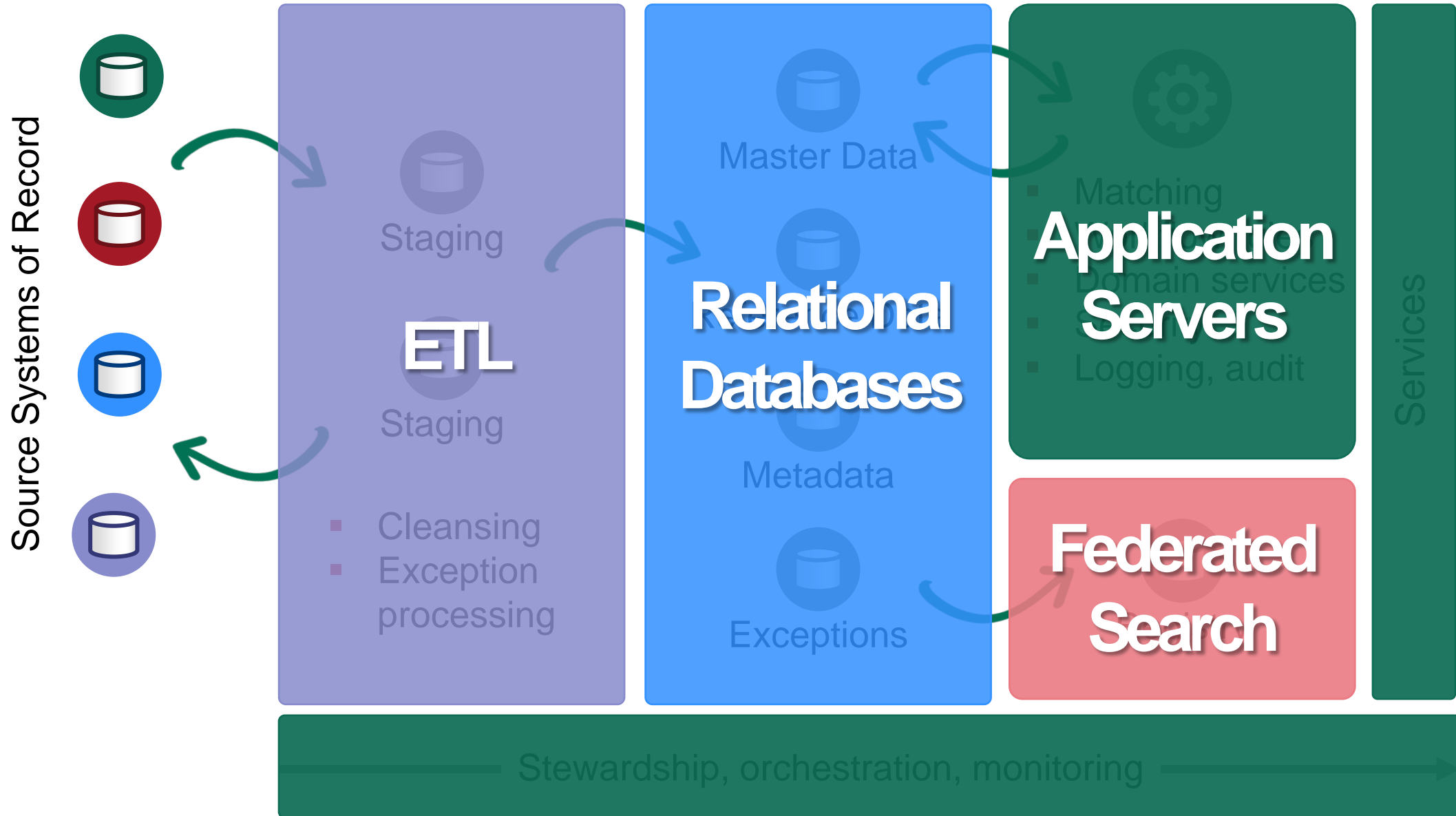
Enforce governance, security,
quality across the entire
lifecycle

Thinking in Entities

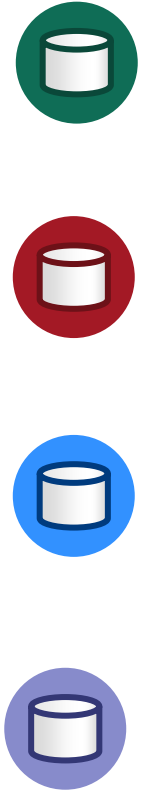
*Source
systems
"as-is"*



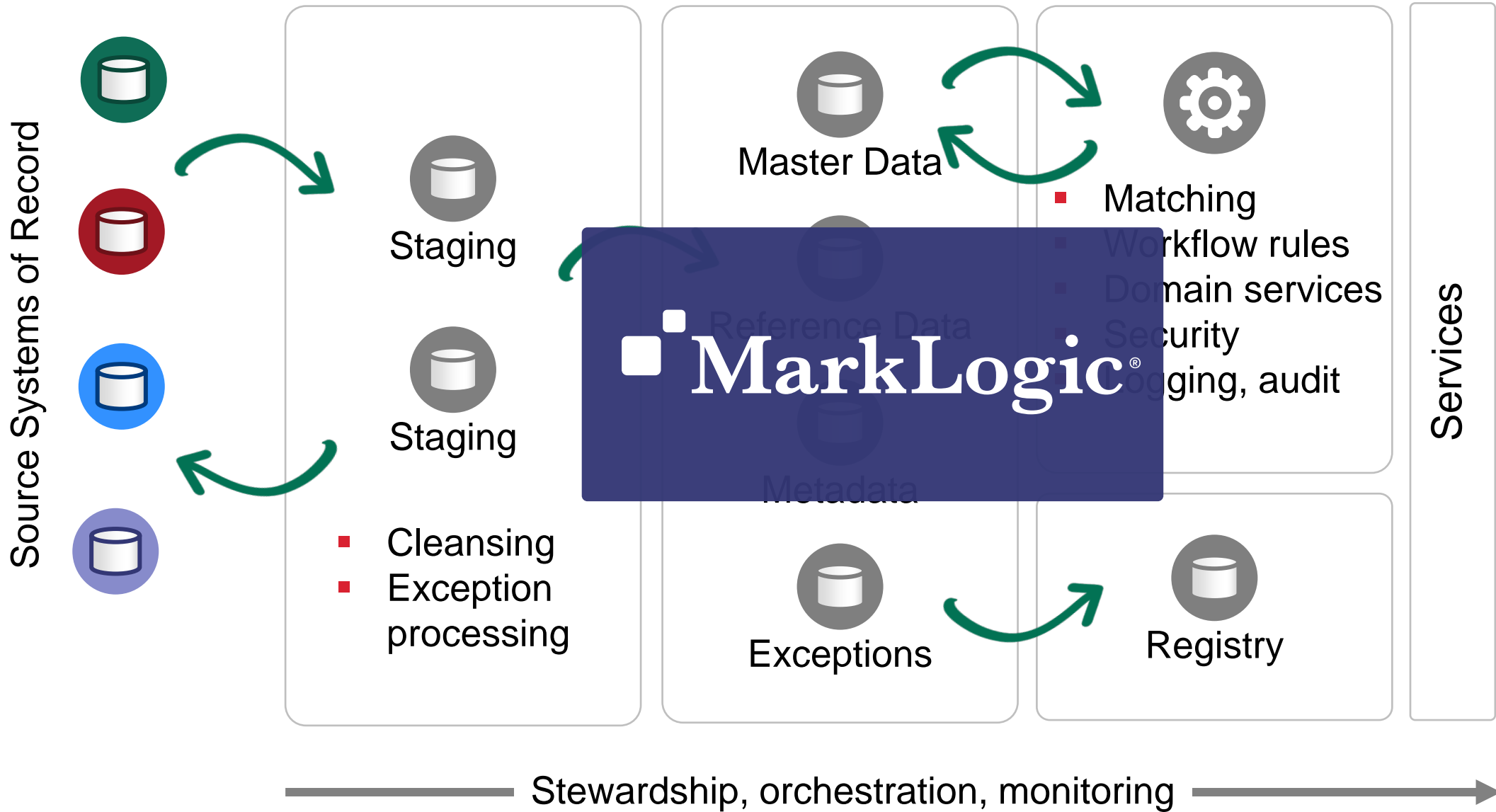
- Transactional updates
- Granular security
- Indexes
- Bitemporal history

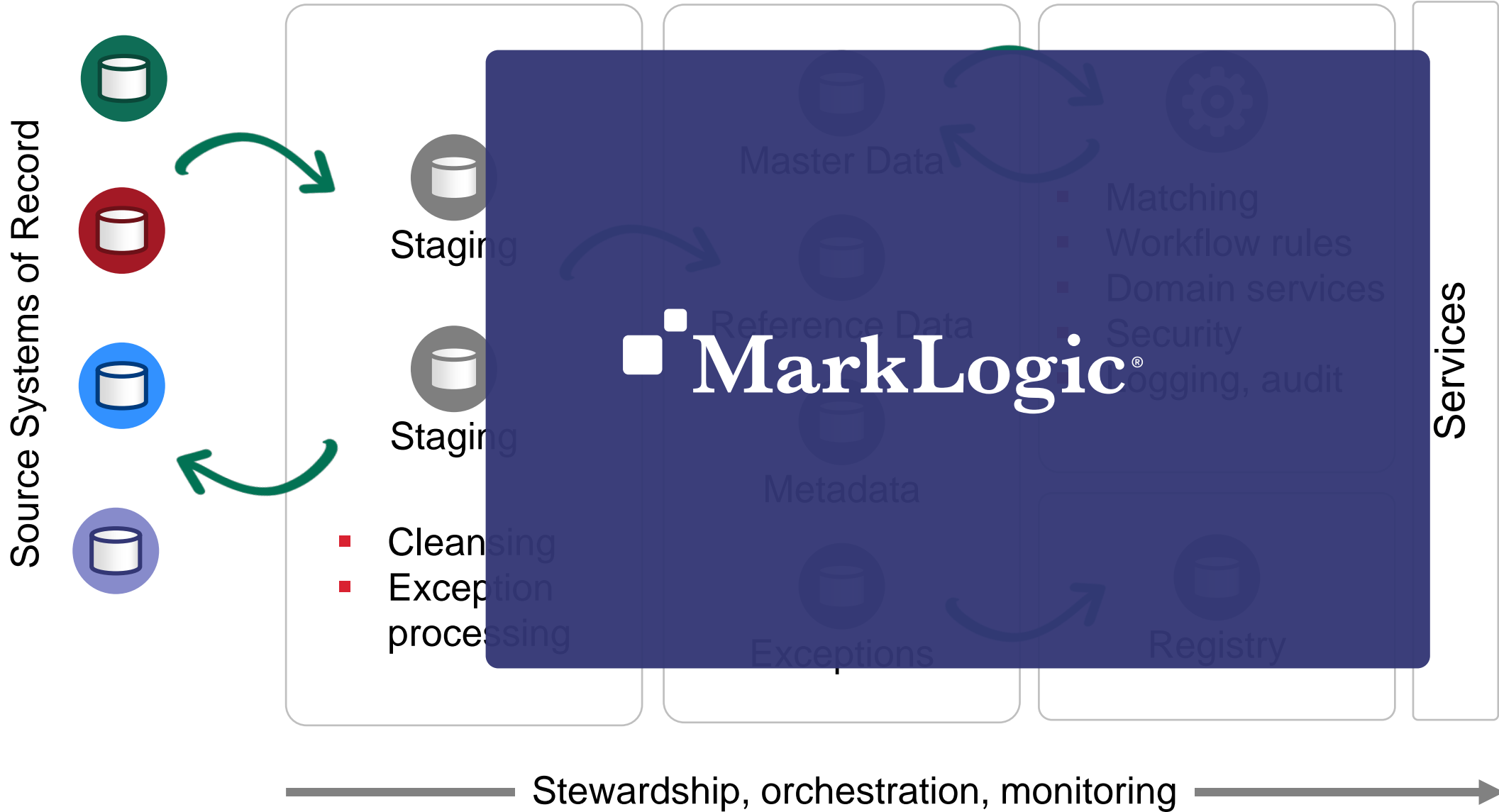


Source Systems of Record



Stewardship, orchestration, monitoring





MASTER DATA MANAGEMENT IN PRACTICE



Customers Leveraging MarkLogic for MDM

Capability

- Time to Value
- Evolving Business Requirements
- 5 year Replacement Cycle
- Legacy Systems Retirement
- Decision Accountability

Customers



HealthCare.gov

Mitchell1



KPMG

aetna®

Our customers are already doing this today...



Aetna

Hub for sharing master data among hundreds of source systems and hundreds of subscribers.



US Combatant Command

Secure sharing, exploitation, and analysis from dozens of sources at HQ, theater operations centers, and detached users.



Top Entertainment Company

Apply corporate standards for creation and use of digital assets from production through distribution across the global enterprise.



Healthcare.gov

Insurance marketplace and exchange hub for millions of consumers, thousands of providers, dozens of stakeholder agencies.

Techniques and Tips

- Envelope Pattern – Model what you need when you need it
- Common Services Layer – The distributed Hub
- Semantics – Relating to your Data
- Decision Accountability – Bi-temporal and Tiered Storage

Techniques and Tips – Envelope Pattern

- Data Modeling can easily take up 30% of a project's resources and is commonly cited as a key reason for project failure
- Need to be able to maintain the original data in context so systems can continue to use it
- Have to be able to expand the data model to meet existing and new requirements
- Solution:
 - The envelop pattern: Leave you data as is and wrap it with the information that you need

Techniques and Tips – Envelope Pattern

Example:

```
<master-envelope>
  <metadata>
    <uuid>...</uuid>
    <first-name>...</first-name>
    <last-name>...</last-name>
  </metadata>
  <raw-data>
    ...
  </raw-data>
</master-envelope>
```

Techniques and Tips – Envelope Pattern

- It's Envelope's all the way down – Let each organizational unit within the enterprise create their own envelopes and institute their own Data Quality controls and rules.
- Evolve your data model as your requirements change over time without impacting the original data

Techniques and Tips – Common Services Layer

- Business Units need to own and manage data that's important to them in the way that works best for their requirements
- The enterprise needs to be able to enforce consistent access, visibility and quality requirements on key pieces of data

Techniques and Tips – Common Services Layer

- Solution: A Common Services layer
 - MarkLogic's Application Services allows you to deploy the same code across the enterprise to different locally managed clusters. This allows each business unit to prioritize and adapt while still enabling anyone in the enterprise to leverage the data and services necessary to ensure full transparency. These can either be accessed independently or unified through the use of a super database.

Techniques and Tips - Semantics

- Not every OU within the enterprise will have the same naming conventions, cardinality rules, or perspective on master data
- The enterprise and each business unit may have one or more taxonomies or ontologies that apply to their data
- Solution:
 - Semantics can be used to model relationships between data across the enterprise and allow OUs to relate their data to others and vice-versa
 - Key relationships: sameAs, belongsTo, parentOf, childOf

Techniques and Tips – Decision Accountability

- Businesses Need to
 - Understand why a decision was made
 - Based on what data
 - For regulatory and compliance reasons
 - For back testing
 - Manage the costs of doing so

Techniques and Tips – Decision Accountability

- Solution: Bitemporality + Tiered Storage
 - Bitemporality lets you ask and explain what you knew and when you knew it so that you can test or execute a decision again at any point in time just as the system existed at that time
 - Tiered Storage lets you manage the cost associated with storing a rolling history of your database including transitioning that data to offline archival storage

Additional Resources

- MarkLogic University Training - <http://www.marklogic.com/training/>
- MarkLogic Developer Site – <http://developer.marklogic.com>
- MarkLogic Data Modeling - <https://developer.marklogic.com/learn/data-modeling>
- MarkLogic Data Hub Framework - <http://marklogic.github.io/marklogic-data-hub/>

QUESTIONS?

