

Establishing Real-World Data Integration Best Practices

CHRIS ATKINSON

Solutions Architect
MarkLogic

T.J. TANG

Software Architect
AbbVie

The AbbVie Journey to a COE

- 01 Introduction**
AbbVie and T.J. Tang – Who are they?
- 02 First Steps**
Taking those first big decisions
- 03 Evolution**
How and why the platform changed
- 04 Establishing a COE**
Learning from their own COE development
- 05 Help and Materials**
Useful resources when establishing your own COE





8,800+
R&D Workers



175+
Countries

abbvie



#1 Selling
Drug World
Wide



\$4.8B
R&D 2017

Key Facts

AbbVie is an American publicly traded biopharmaceutical company founded in 2013. It originated as a spin-off of Abbott Laboratories.



26M+
Patents



Software
Innovators

T.J. Tang – AbbVie MarkLogic COE

T.J. Tang

Software Engineer

20 years experience

Architect

Evaluates solutions and opportunities

Manager

Heads up the COE effort

The COE

Team

7 People

Purpose

Data integration best practices

Business Demand

Reduce costs

Accelerate R&D



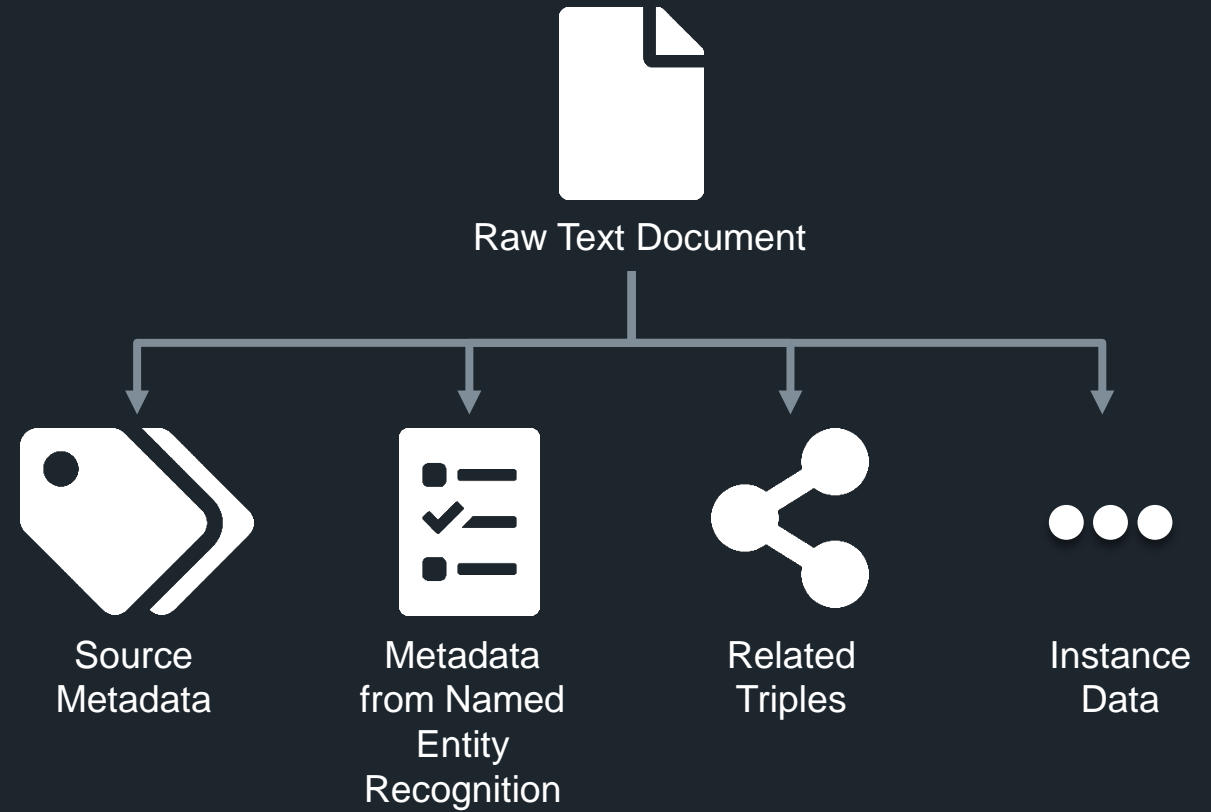
First Steps

Initial Implementation

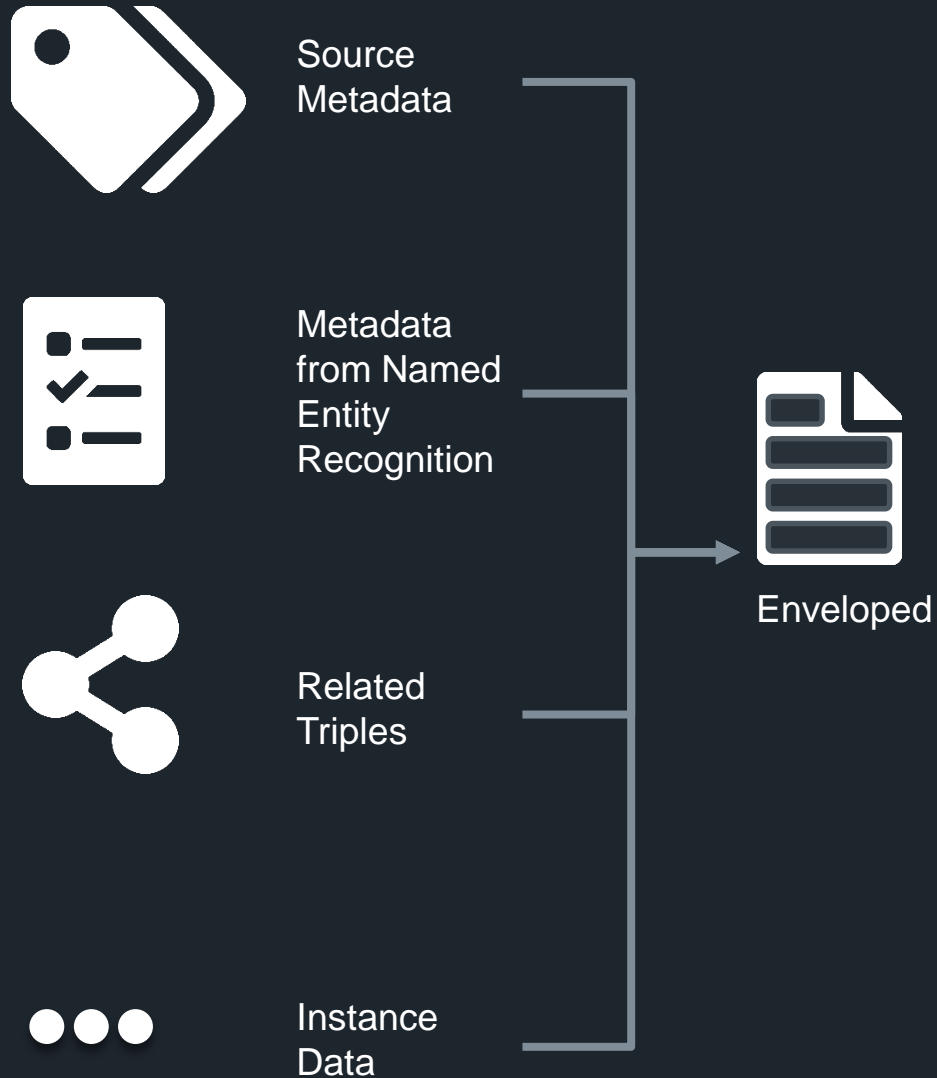
Modelled data to several smaller documents

CPF for processing

External content enrichment



Early Architecture Changes



Early Changes

Envelope pattern

Modelled data into larger documents

COE Takeaways

MarkLogic Experts often made available

Meetups for 1:1 discussions

Use SEs, SAs



Data Hub Implementation

COE Standards

V2 and later standards
Custom changes need migrating
From & to state needs mapping

COE Takeaways

Talk directly to PMs
SAs for roadmap
Request enhancements to MarkLogic



Evolution of the COE



The COE is Born

A natural evolution

Advice and expertise evolved

Servicing other LOBs

COE Takeaways

Experts are needed

Consider their role



COE Day-to-Day Activities



Educate



Qualify



Validate

COE Takeaways

Create standardized materials

Customize to need

Needs to be on-demand



Recent Innovations



NiFi Adoption

Less developer centric

Good performance

Some early challenges

COE Takeaway

One of a few options (e.g. MuleSoft)

Data Hub is innovating fast

MarkLogic relationship important

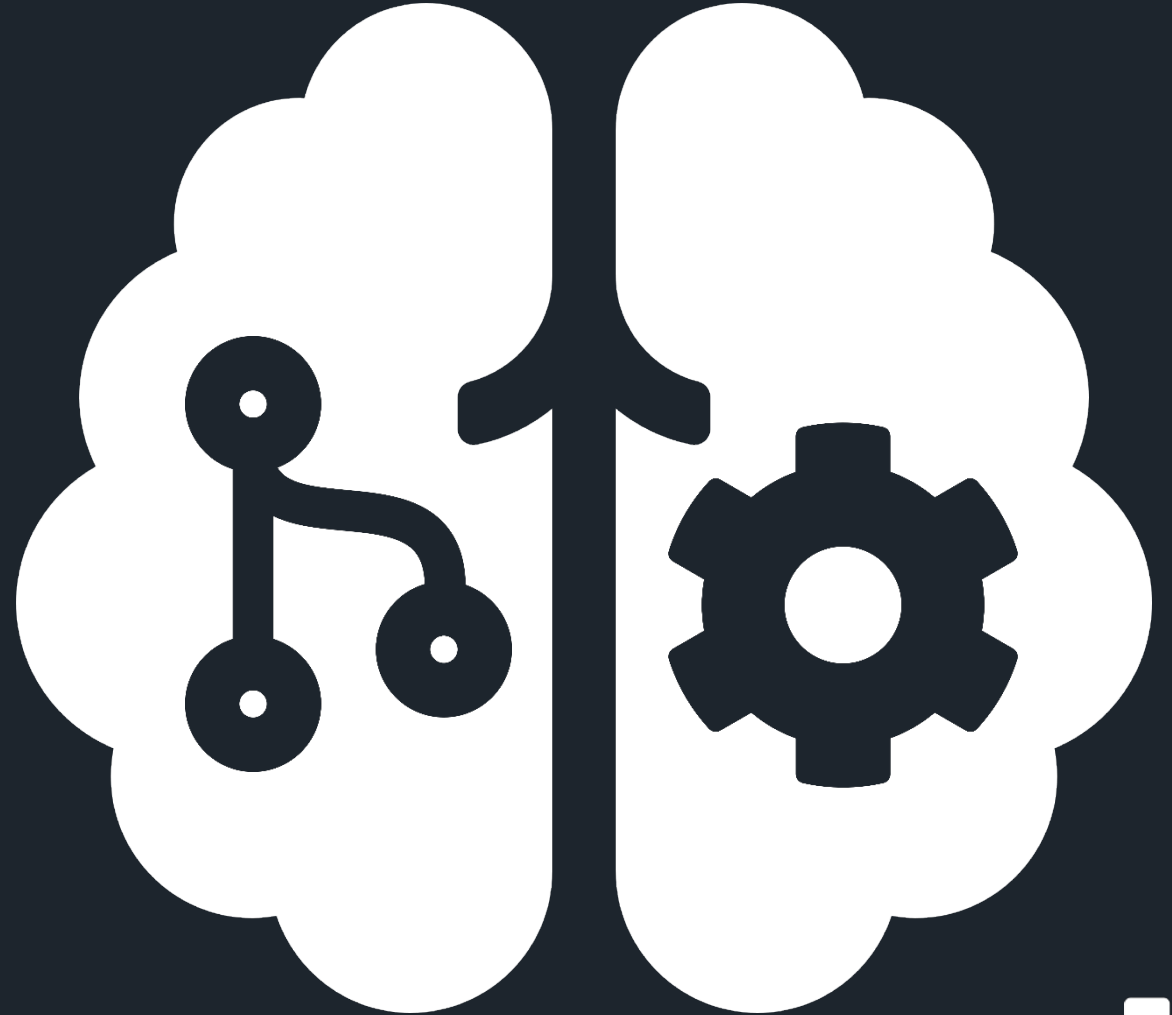
Request help/advice



Next Steps

Machine Learning

Avoids offloading data
Processing close to data



Establishing Your Own COE / Standards

Expert Help

MarkLogic Experts

Solution Architects

Solution Engineers



Internal Experts

An essential component



Data Integration Guides

Architecture

Value

Avoiding Pitfalls

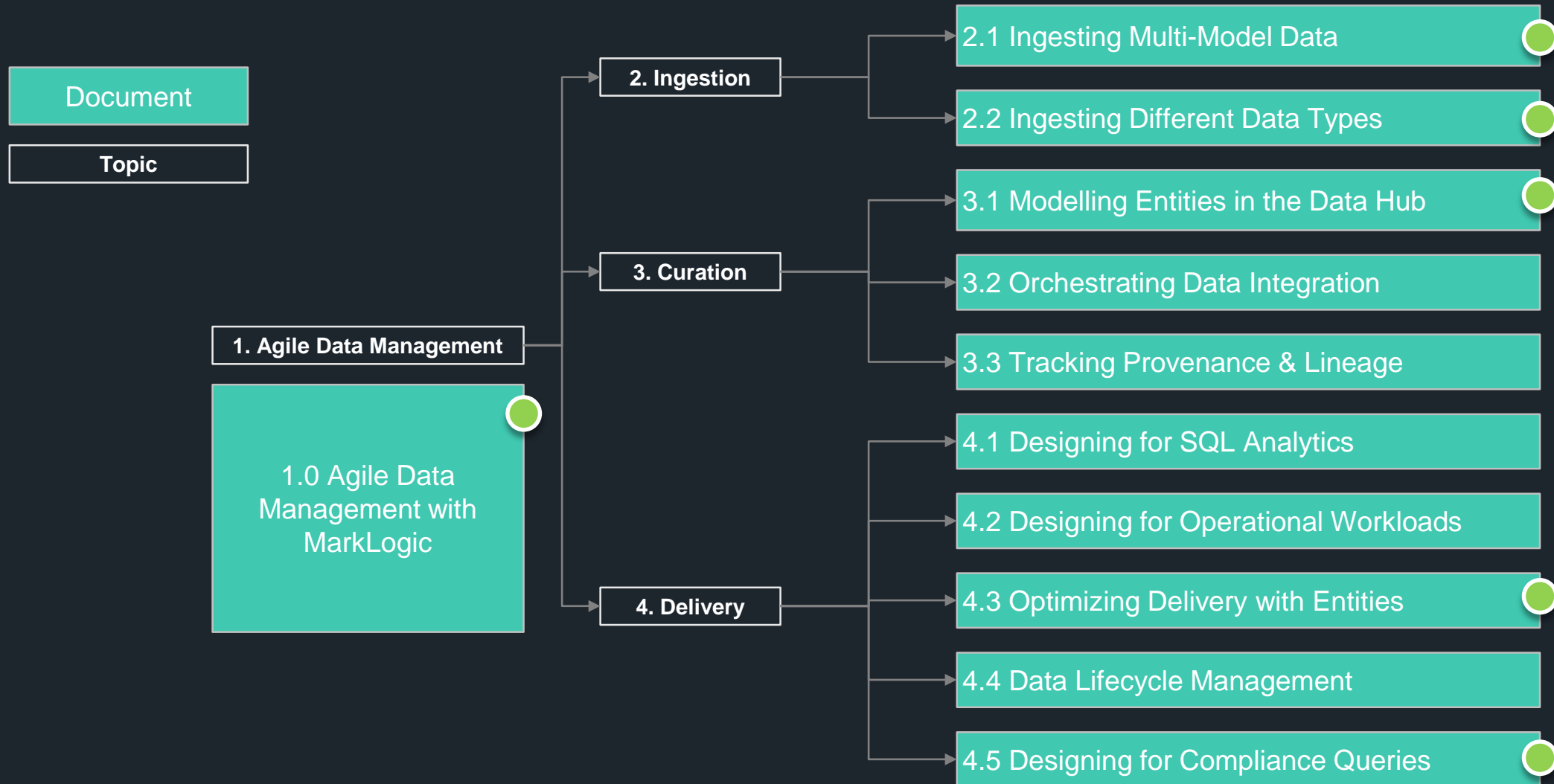


Technical Patterns

Code Examples



Data Integration Guides



Integration Guide Example

3.1 Modelling Entities in the Data Hub

Created by Chris Atkinson, last modified on Apr 10, 2019

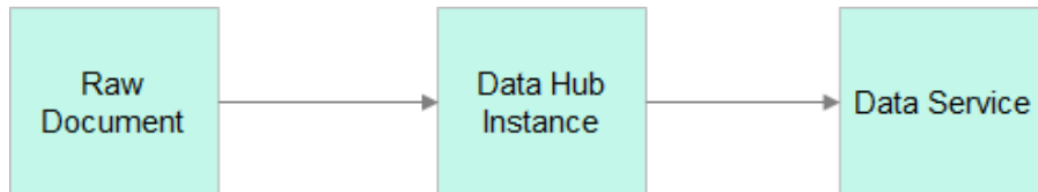
PATTERN SUMMARY.

Data Hub Entities are abstractions of logical business entities and are used within MarkLogic to define Data Hub Instances. Data Hub Instances are the main persistent XML or JSON document format that is stored in a Data Hub.

These instances contain data 'curated' from the raw source data to be used by data services. (see curation pattern).

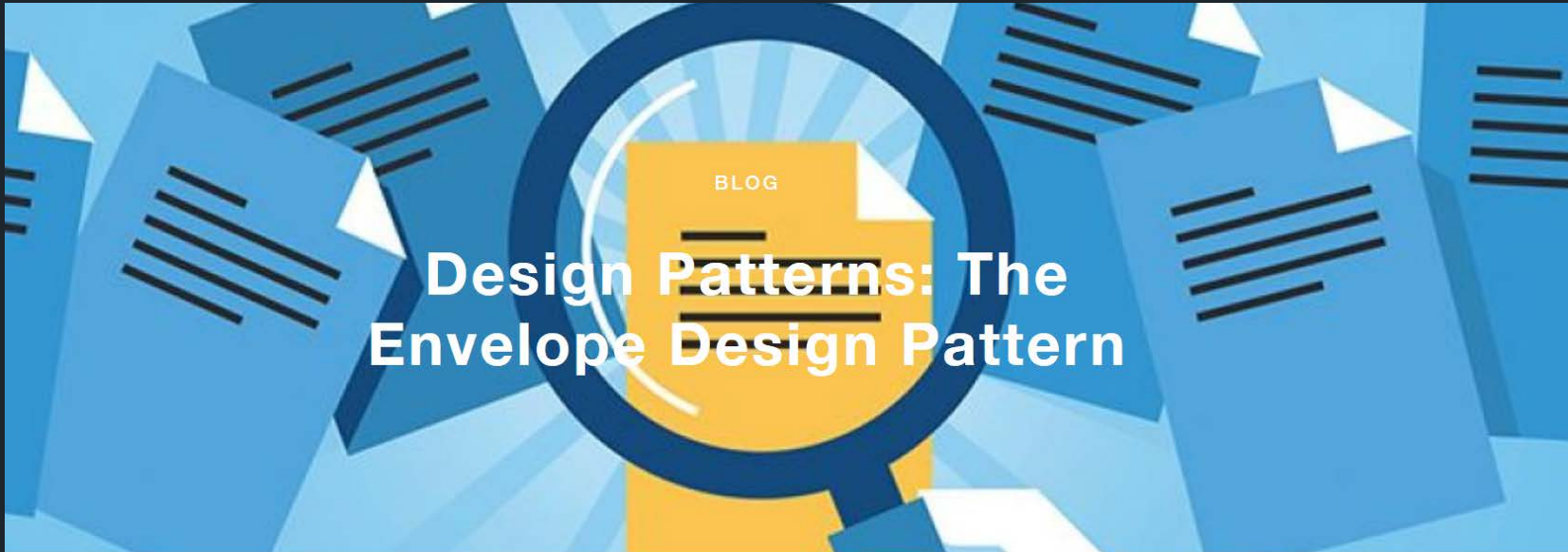
Unlike a "Data Hub Entity" the Data Hub Instances are concrete, persistent documents in MarkLogic that capture the primary division of data into useful groupings with a defined hierarchical structure.

Data Hub Instances are entirely owned by the Data Hub and are abstracted from the source models that provide data. Their purpose is to maximise reusability of data logic and abstract the source data from consumer demands. The Data Hub Instances should be natural for the domain and ideally, they are well aligned to the most important and typical Data Services that the Data Hub will be providing. This is part of the "Data Services First" (see Data Services First Blueprint Pattern) design approach for Data Hubs – the Data Services inspire and guide the physical model for the Data Hub Instances.



For example, typical good Data Hub Entities include things like: Insurance Policy, Financial Trade, Shipping Manifest, Person or Customer.

Technical Patterns



DAMON FELDMAN | AUGUST 10, 2018



MarkLogic design patterns are reusable solutions for many of the commonly occurring problems encountered when designing MarkLogic applications. These patterns may be unique to applications on MarkLogic or may be industry patterns that have MarkLogic specific considerations. Unlike recipes, MarkLogic design patterns are generally more abstract and applicable in multiple scenarios.

Envelope Design Pattern



Questions?





Thank you