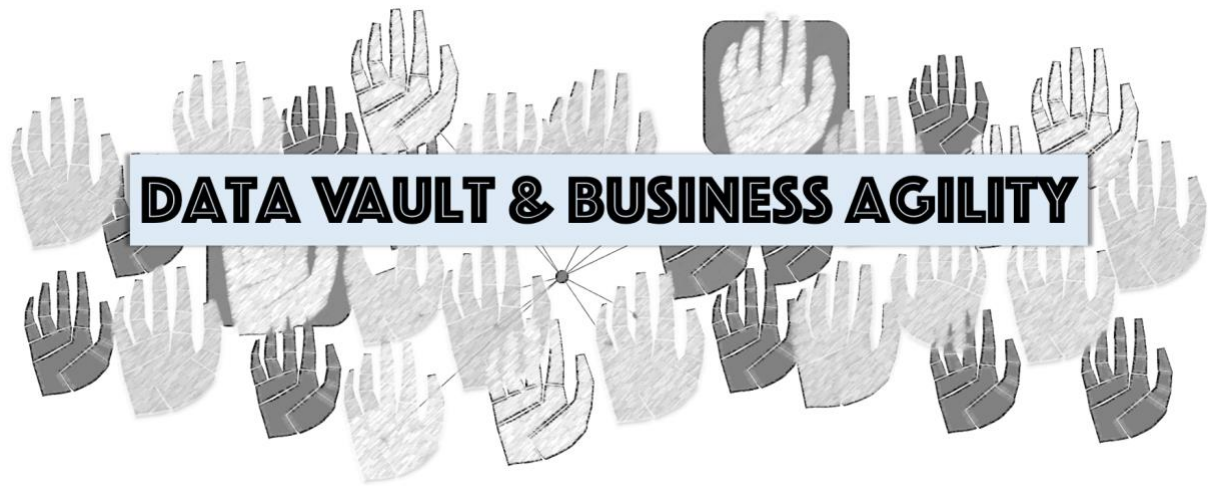


Time to upgrade your thinking on Data Vault



"If you're good at course correcting, being wrong may be less costly than you think"
Bezos, see: bit.ly/3DMen8o

Business Agility needs **Technical Agility**, we will hear this term later in this article. I'd like to extend that phrase; Business Agility needs **Data Agility**. Yes, you guessed it, Business Agility means you need to map business capabilities and value streams and how they map to the enterprise. Technical Agility (and Data Agility) aims to ensure that if a Business pivots its technology and data can easily pivot as well.

Here we will define

- Business Architecture
- Data Vault
- An Agile Data Platform
- Mapping Business to Technology (and Data)

Business Architecture

"If you cannot measure it, you cannot improve it." – Lord Kelvin

Enterprise Architecture frameworks seek to describe, organize, map and express common viewpoints over an organization that almost always starts with the **business architecture** domain. All other enterprise architecture domains enable, automate, and form the platform used to scale the business as it scales, so too then must the technology, data and applications be able to scale

Let's start with a few industry-backed definitions

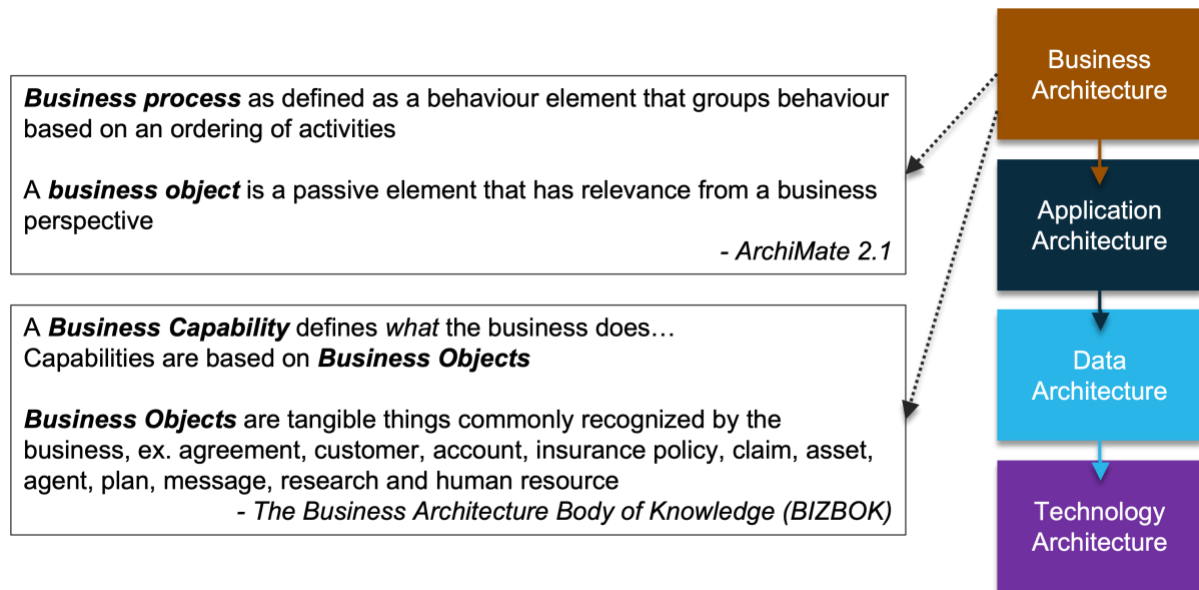


Figure 0-1 Business Architecture in an Enterprise

“Business Architecture represents holistic, multi-dimensional business views of: capabilities, end-to-end value delivery, information and organizational structure; and the relationships among the business views and strategies, products, policies, initiatives and stakeholders.” – Business Architecture Guide (BizBoK).

- **Business Object**, a representation of a thing active in the business domain, including at least its business name and definition, attributes and behavior, relationships and constraints that may represent for example a person, place, or concept --- **nouns**
It is a persistent thing that is of interest to the business - as represented as information concepts have definitions, types, states, and relationships to other information concepts. An **information concept** is comprehensive in terms of representing objects across a business ecosystem but may or may not be defined in a data model or a database
- Information concepts have two key attributes:
 - type – taxonomy of information
 - state –finite state, ex. open, closed ← used in **value stage** entrance and exit criteria, controlled by capability outcomes

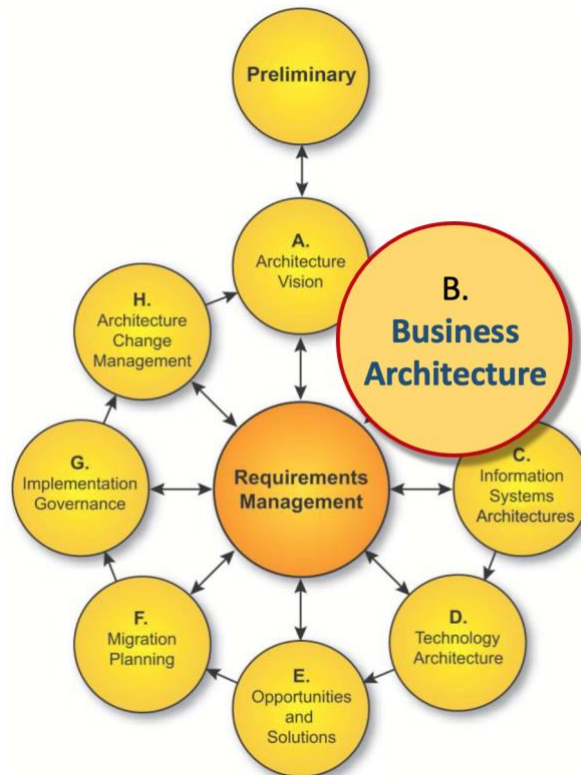


Figure 0-2 Business Architecture in TOGAF's Architecture Development Cycle (ADM)

The latest TOGAF standard (9.2 at the time of writing) has adopted many of the Business Architecture guild's descriptions and practices around Business Architecture. It too now defines Business Architecture in terms of tiers (strategic, core and supporting – see my data vault recipes discussion link below) and recognizes each capability has levels and operate to fulfill value streams to ultimately deliver **business value**.

Business Architecture also recognizes that “**Business Strategies** are only as good as the ability to measure progress towards them.” Under the business architecture blue prints you will find recommendations for templates to

- Setting goals and objectives
- Creating alignment
- **Measuring progress**
- Improving performance

Measuring models can be mapped using a combination of the following at different levels of the organization:

- **Balanced Scorecard** - a method that provides guidance on setting up a monitoring and measurement system for strategic initiatives

Strategic Objectives	KPIs	Targets	Initiatives
Financial	• XY • XX	• ↑% • ↓%	• ----- • -----
Customer	• XY • AB	• ↑% • ↓%	• ----- • -----
Internal Processes	• XY • ZZ	• ↑% • ↓%	• ----- • -----
Organizational Capacity	• XY • BB	• ↑% • ↓%	• ----- • -----

balanced scorecard

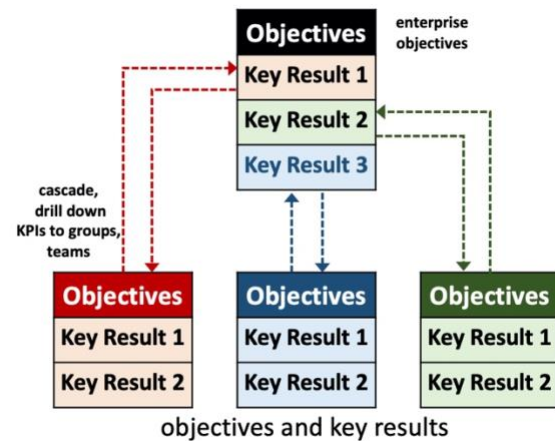


Figure 0-3 Measuring the business Example, see: bit.ly/3IMrWs2

- **Key Performance Indicators (KPIs)** – evaluates the success of an organization or activity by focusing on what’s important that are be used in a Balanced Scorecard *and* OKRs. KPIs should be defined as:
 - Specific objectives,
 - Measurable progress,
 - Attainable goals,
 - Relevant to your organization, and
 - Time-framed
- **Objectives and Key Results (OKRs)** – used by some of the largest software innovators, an OKR can be used to cascade a Balanced Scorecard with the intention of focusing on execution above else. Written with an overarching objective (what & why – qualitative, aspirational) with three to five supporting key results (how - quantitative) and initiatives, OKRs are set up at different levels of the organization hierarchy to even personal OKRs.

“Yes. No. Simple” – Andy Grove, cited by John Doer, see: bit.ly/3IMv40I

For a comparison: bit.ly/3ETbl81

Let’s now turn our attention to Data Vault.

Data Vault

The Data Vault methodology encompasses modelling techniques that are pattern based. And because they are pattern based, they can also be easily automated that ultimately leads to technical agility through repeatable processes that are templates themselves. To start with, what does the data vault *model* consist of?

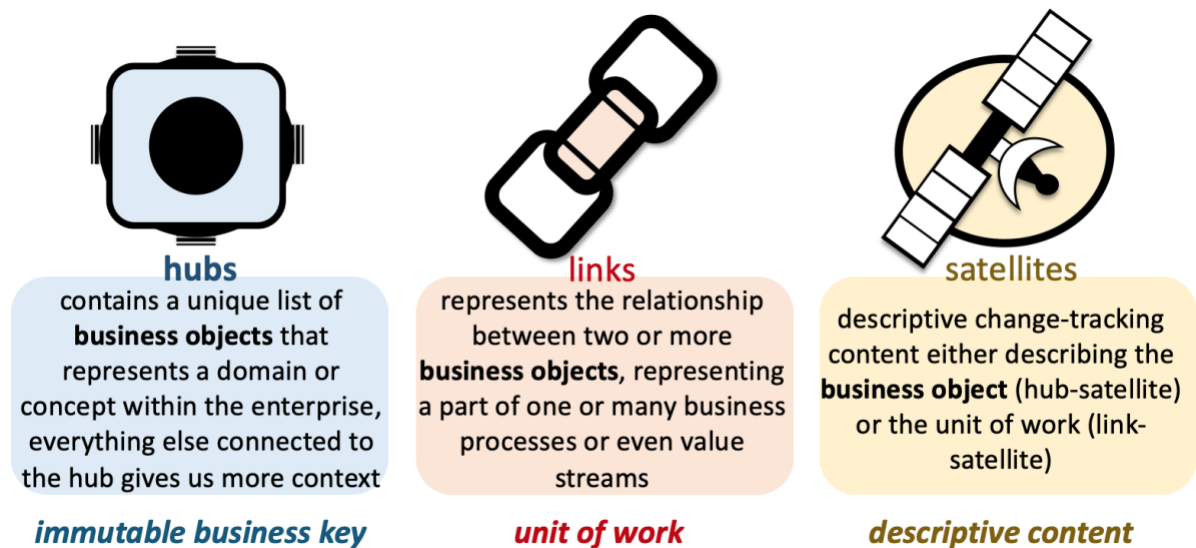


Figure 0-1 Hubs, Links and Satellites

Hub tables contain the list of unique business keys; a business key represents a [business object](#). Each hub table will essentially represent the business object as it pertains to the business capability. Some business objects can traverse business capabilities through business processes *and* value streams. Business objects interact with each other, and their interaction is captured as the **unit of work** within a business process. We capture those in the link table, the list of unique relationships between two or more business objects (keys).

These objects are tracked through automation using software and databases. Yes, in most enterprises there can be multiple databases and a common task amongst enterprises is the need to **master these business objects** through **integration rules**. All these databases and software also track the *state* of business objects and relationships and within the data vault model these are captured in the form of satellite tables. Hub tables form the integration points of all these source data models (business rule automation engines).

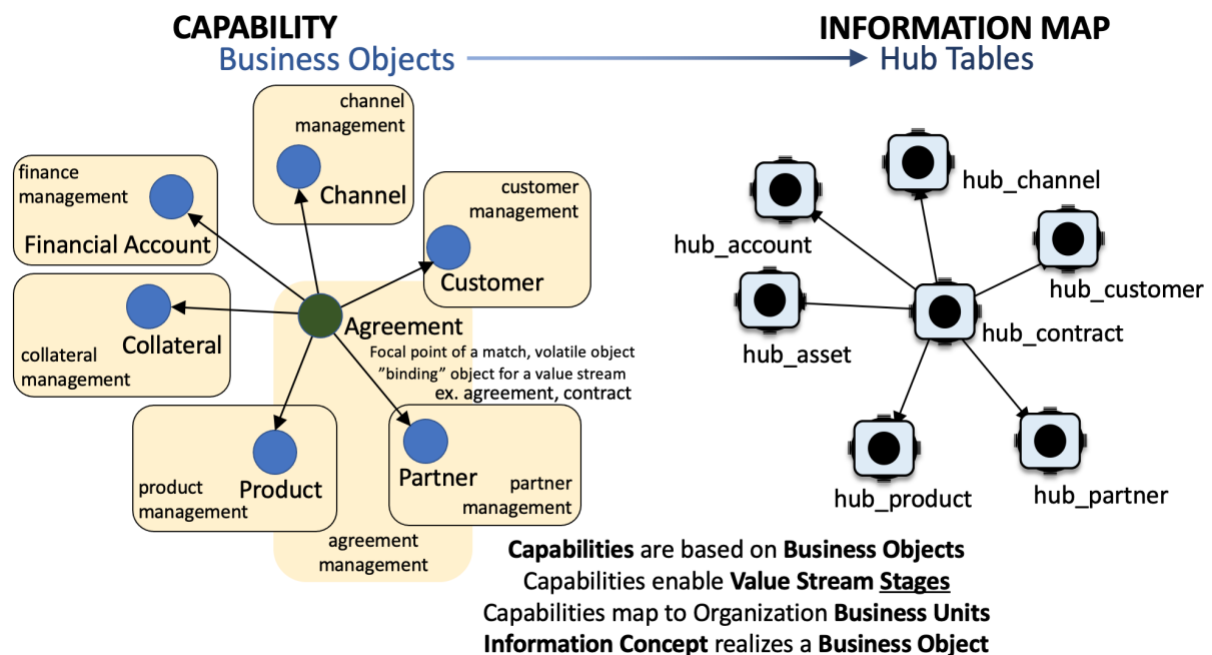


Figure 0-2 Possible Business Architecture to Data Vault map for attaining a financial contract with a bank.

A **binding** agreement (or contract) is what guarantees a unit of work. Each business object can exist on its own *except* for the *binding* business object that brings the other business objects together. An agreement will have a unique identifier (like the other business objects) and be based off pre-existing templates and business processes that collectively are known as the **Value Stream** because the outcome is **Business Value**. Multiple **Business Units** will participate in a Value Stream. Note that link tables were not included in the diagram above; hubs can be **industry standardized** but the capture of link tables reflect the business processes automated by source systems; there will likely be similar patterns within and across industries but no hard defined links that every business rule automation engine will match *exactly*.

For a deeper discussion on Data Vault please visit the articles listed below.

New to Data Vault:

- Learning Data Vault is Like Learning How to Make Beer! See: bit.ly/2ZYGpJP
- Data Vault Elevator Pitch, see: bit.ly/2RyoRjv
- Data Vault Recipes, see: bit.ly/3o4koB6
- Say NO to Refactoring Data Models! See: bit.ly/3tPI66B
- Advantage Data Vault 2.0, see: bit.ly/2ZZILcv

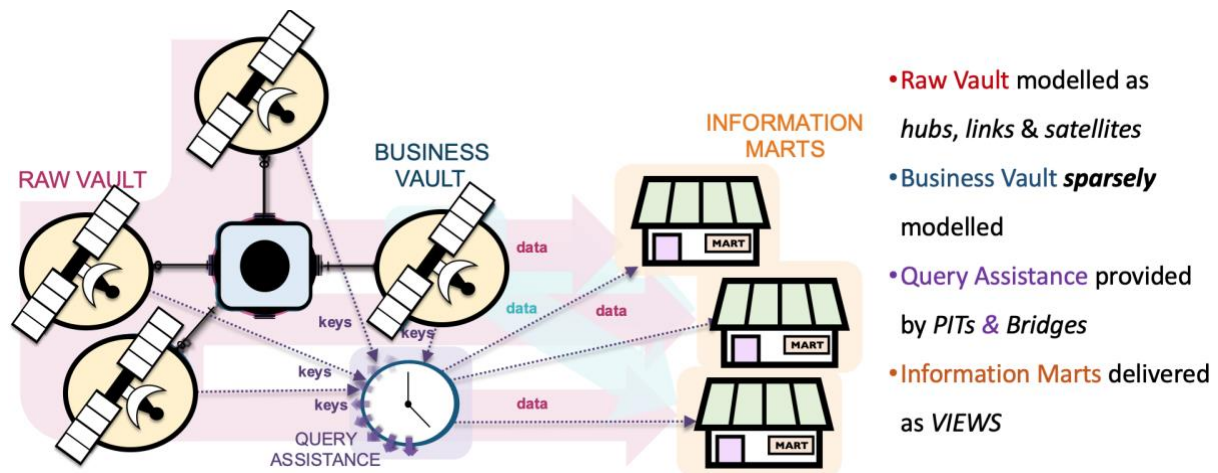


Figure 0-3 Raw and Business Vault as the base to Information Marts

Modelling techniques refer to these articles

- Decided to build your own Data Vault automation tool? See: bit.ly/3bRIV7U
- You might be doing #datavault Wrong! See: bit.ly/2V32eFu
- Data Vault PIT Flow Manifold, see: bit.ly/3iEkBJC
- Data Vault Mysteries... Business Vault, see: bit.ly/3BUt81s
- Data Vault Mysteries... Zero Keys & Ghost Records, see: bit.ly/3vjTXdg
- Data Vault Mysteries... Effectivity Satellite and Driver Key, see: bit.ly/3oS4k70
- A Rose by any other name... Wait.. is it still the same Rose? See: bit.ly/3xlFK0s
- Data Vault has a new Hero, see: bit.ly/3y4mUdV
- Building Data Vault modelling capability through the Mob, see: bit.ly/3zgP7OP

Measuring and monitoring

- Data Vault Test Automation, see: bit.ly/3dUHPIS
- Data Vault Dashboard Monitoring, see bit.ly/3BjSg1F

Finally, a deep dive into various modelling scenarios, sample code and automation patterns refer to the Guru.

- The Data Vault Guru: a pragmatic guide on building a data vault, see: bit.ly/3tXoyNK

What I want to highlight about the Data Vault as a methodology is this: *Agile, Adaptive, Audit, Automation, Autonomous.*

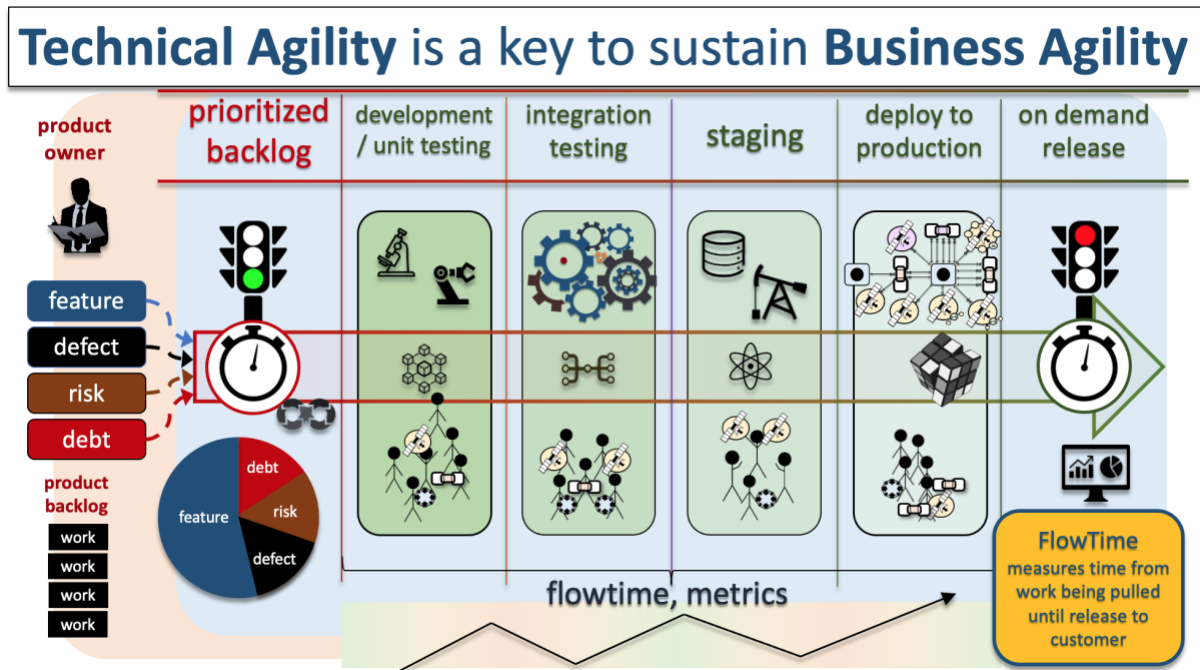


Figure 0-4 Note that cadence is measured

- It is **Agile** – “Technical Agility is a key to sustain Business Agility” – BizBok [presentation](#). An agile business is **adaptive** to change, thus the technology should be too. Every portion of this article is an overlap of repeatable patterns. From Agile best practices, to DevOps, [DataOps](#), the data platform itself (we will see later) and the data modelling methodology. Because they are template driven and governed through centralised standards and governance the delivery is repeatable the metrics to measure them are repeatable patterns too.

Only three table types, config-driven, automated output

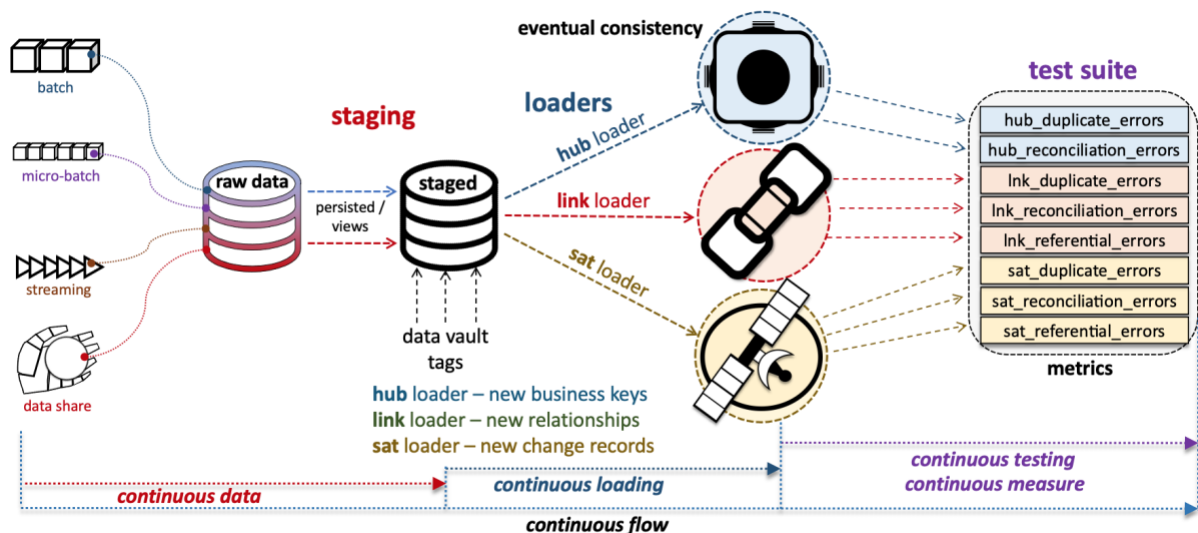


Figure 0-5 Note that delivery and quality is **continuously** measured all a part of **continuous delivery**, hub tables provide **continuous data integration by business object**

- It is **Automated** – recognize that data projects are **software builds**, like service-oriented architectures each component should have a **single purpose** and nothing more. The description or label of the data process describes what it does in its entirety and *nothing more*. If I were to call a Hub-Table-Loader, then that is all it does. A hub target table is a unique list of business objects and therefore the hub-loader must do just that, load new keys to the target hub and ensure it does not load duplicates. It does not:

- assign data-vault tags,
- perform any calculation,
- apply any hard rules

These are the job of staging (a separate process)... because this is a software build you might want to refer to the 12-factor app for further guidance (see: 12factor.net) on software build expectations. This means each software component is **Autonomous** and driven by configuration, the code remains the same, *a repeatable pattern*.

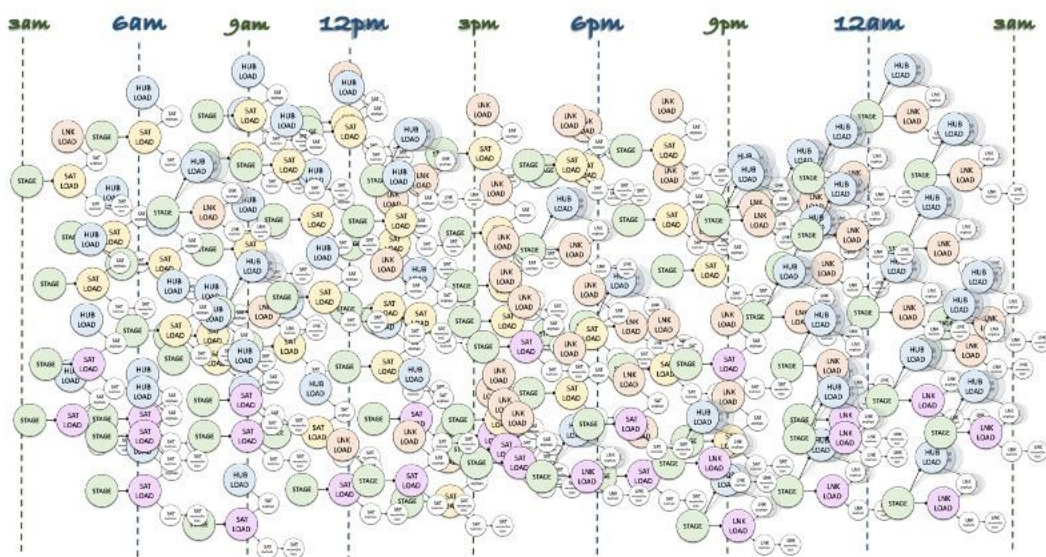


Figure 0-6 Automated Data Vault **continuous loading** and **eventual consistency**, any time of day, reduce insight latency

- it applies **Audit** – data **provenance** is increasingly becoming a must for every enterprise and especially global ones. From a data vault perspective this concept is enshrined through mandatory record-level metadata, and some optional ones that promote record level documentation (pun intended). *What are those?*
 - **Record Source** – defining where that record came from, it should list the filename and location.
 - **Applied Date** – the package of time date, all data is at it is on this applied date and the data in this “packet” may include business dates
 - **Load Date** – when the data enters the data platform
 - **Task / Job ID** – tying this record to what loaded it, we would be able to trace log history of even more metadata about how the record got there
 - **JIRA ID** – why not? At this level I would be able to tie the record to which initiative and its associated mandatory documentation got that record there.

Not to mention the other requirements like access history, object dependencies, business glossaries, taxonomies, ontologies, lineage... *oh my!*

Data Vault is ***not complicated***, one of the possible reasons for a failed data vault implementation is the failure to recognize that the data and analytics platform is ***not the repository for technical debt***. By not setting aside the budget to deal with debt the effort in managing and sustaining the platform snowballs; more technical debt may be introduced to work around an existing technical debt, that means the cost to deal with it has accumulated. Technical debt itself must be managed, controlled, and reduced. The metrics will show and reflect an organization’s health, as these smaller measures begin to show poor system health it will eventually impact the overall business measures as well. Data quality is discussed in the data vault recipes article above.

What’s next? Data needs somewhere to be available *at rest*...

Five Finger Data Platform

Before a model can be deployed you need a data architecture to house your data models. This refers to a platform layered approach that is designed to scale.

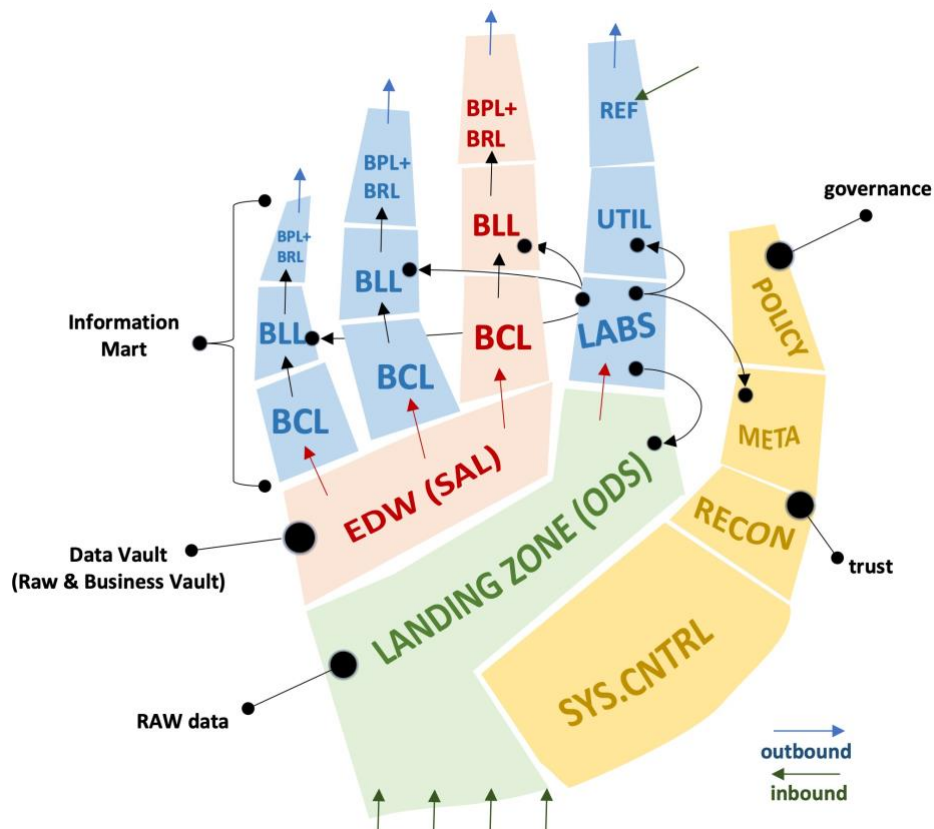


Figure 0-1 A Modern Data Platform Stack (see: bit.ly/3ERdpTm)

Automation is at the crux of data delivery on any **technology** platform, and it should be at the crux of any *growing business*; automation delivers **business capabilities** and thus automation is at the core of your **data** platform!

- **Green** zone – Landing zone, the area for all forms of data by source managed either externally or internally and available as **raw data** to the analytics platform
- **Red** zone – Source-aligned layer designed to integrate, historize and categorize data for consumers
- **Blue** zone – either managed **on behalf** of a business unit or the business unit manages their stack in the blue zone themselves.
- **Yellow** zone – content related to managing and **governing** the other zones

Let's now discuss the acronyms and principles around this delivery.

- **ODS** (Operational Data Store) is the lowest layer and represents the data source. Data can be landed there at any cadence and separated by schema names that reflect the source name.
- **SAL** (Source Aligned Layer) is the area where data integration and historization occurs. In this layer we find the integrated Data Vault, *Raw and Business*.
SAL forms the base of available integrated data for business units to participate in, each business unit will then have their own stack starting from BCL. Collectively this area is the Business Access Layer (**BAL**, or End-User-Layer (EUL)), BAL contains:
 - **BCL** – Business Conformed / Curated Layer. BCL can be provided pre-baked by the analytics team, or it can be completely managed by the business unit itself. This can also be described as the *private* Business Vault that contains historized business rule output only available to that business unit.

- **BLL** – Business Logic Layer. BLL combines data and applies complex business rules if not already supplied in BCL. These are the start of **Information Marts**.
- **BPL** – Business Presentation Layer, a separation of curated data in this layer can be shared with other business units or external business stakeholders. The layer below this is BLL. Keeping this separated from the logic and conformed layers promotes the isolation of access that is easily repeatable.
- **BRL** – Business Reporting Layer, Information Marts with Business Intelligence (BI) Tool specific requirements (*functional rules*) can be deployed here. This layer is based on BLL.
- Each business unit will also have a **LAB** that *can* have access to ODS to access raw data, the integrated data zone (for raw and shared business vault model), and *optionally* it could have access to another business unit's BAL.

In addition, a data platform should enshrine these principles

- Data always moves up, one layer at a time, *palm to tip*.
- Data can never move downwards. Intelligence created by each business unit must be ingested through a business rule promotion process, that is the output of such rules is ingested through ODS as a new source and follows the same rules for promoting to the integration layer (like raw vault) if it is to be shared with other business units. Should the business rule be isolated for private use then it is promoted directly into BLL.
 - **Private Business Rule:** based on BCL
 - **Shared Business Rule:** based on ODS and other content in SAL
- All data sources must come through ODS that will provide a **single version of the facts**.
- Reconciliation is a must. Automated testing occurs between sources and integrated layers.
- Reference data is managed by each business unit and stored under their own **REF** layer; a single shared reference data model can also be used that helps with data enrichment. It will, however, follow the same ingestion and integration process as live transactional data and be ingested through ODS. REF layer can be loaded directly from Excel or another Reference Data Management (RDM) solution and does not need to flow from ODS, which then promotes self-service and rapid ingestion of private reference data.

Data governance is *centrally* managed but can follow a hybrid approach for applying the governance rules, these are things like column classification and row-level access privileges. Additional policies that a business unit wishes to apply should be stored in their own **UTIL** area.

From a data architecture perspective, the layers are divided by horizontals and verticals that follow a flow of data as streams, always in one direction, *up*.

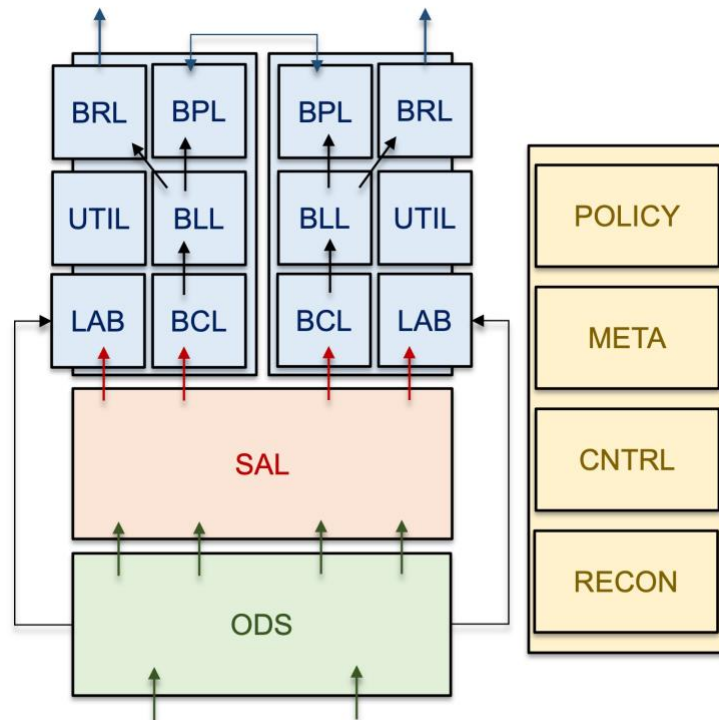


Figure 0-2 Modern Data Architecture for Data Vault

An enterprise will have many business units, some will want reporting delivered pre-canned, others will want the integrated data to build their own analytics, some will want access to the raw data as it is delivered, pre-modelled.

Data delivery models:

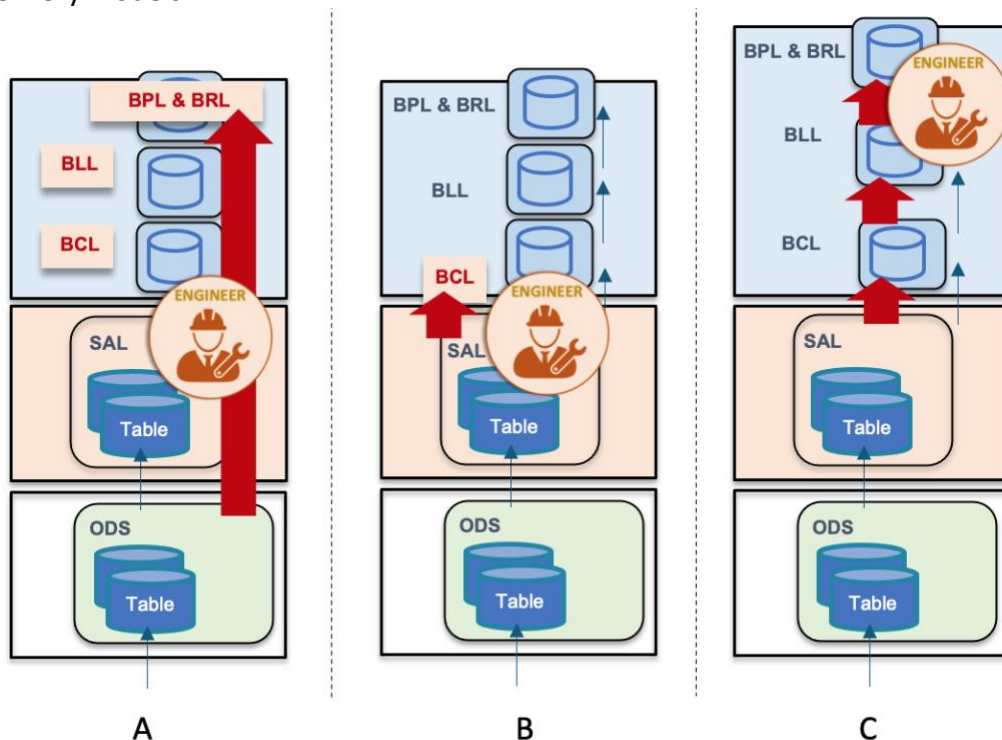


Figure 0-3 Delivering analytics

A – the delivery service within an agreed upon service level agreement (SLA) builds and delivers information marts and possibly all the way up to the dashboard reports for a

- **Raw Vault** is the modelled source, hard rules only
- **Business Vault** is the sparsely modelled derived content based on raw and business vault, soft rules
- **Private business vault** is the business unit specific modelled curated layer, BCL, soft rules (standardization, conformance)
- BLL contains more complex business rules, soft rules
- BRL + BPL contain functional rules
- Data Lab is the business rule development zone that leads to more business vault, private business vault or BLL content.

Through measures of data movement, access, counts, [time to value](#), governance; the platform (with the help of data vault's inherent scalability) will be in a state of eventual consistency delivering on that **business agility**. The pattern itself can be extended to multiple organizations that form the business, extending the business itself to new opportunities.

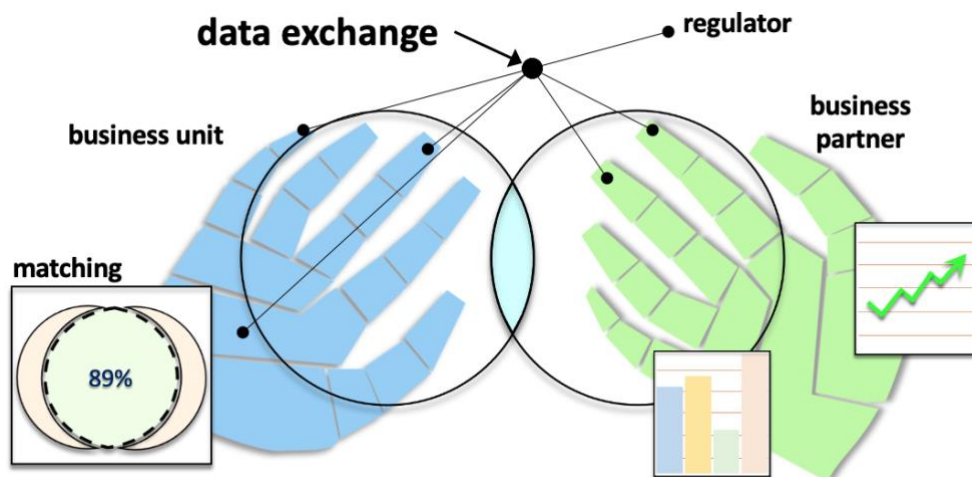


Figure 0-5 Sharing pattern based on BPL

If the entire business can use the same base (ODS) to deliver its entire business agility, then a separation of platform would *not be necessary*. **Implementation reality** however must take several external influencing factors, *such as* (but not limited to):

- A business is made up of business units *and* business partners to deliver on business capabilities. A capability may be enhanced with the use of a *data clean room* to provide secure [double-blind](#) analytics to improve marketing efficiency. Or another capability may rely on a partner for logistics, manufacturing, or raw materials but without this partner the business capability cannot operate or would be less economical to do themselves.
- The business itself may be influenced by global factors such as regulations in the jurisdiction it operates in, diplomatic relations of those jurisdiction and data sovereignty that prevent customer data to cross borders. In some cases, non-identifying data may still be allowed and in others the privacy of data may be subject to regulations like [GDPR's article 17](#).

At the centre is the data exchange, a central point enabling self-service that enriches each participant's own data and reducing the need to invest in their own capabilities to

build up their own. Each participant should in turn also provide *aggregated performance metrics* of their own data on the exchange!

Every participant is not only beholden to their own stakeholders and customers to their data quality and KPIs, but to their business partners as well. The ability in such a pattern-based architecture to pivot suggests that if a current capability isn't adequately performing that a course-correction can be easier to achieve.

Now over to the final section...

Hoshin Kanri for Data Vault

With a well-defined **business architecture**, the task of building a data vault is infinitely easier! If the data you're modelling for a business does not have a business architecture or *ontology*, then a suggestion here is that *you build one*. You will start to deliver more than just a data model but a way of representing the enterprise as well. Now this is not to say you need a complete enterprise ontology *before* you start to deliver a data vault model; no, *don't boil the ocean*, deliver on a [steel thread](#). But get the business involved (we discuss this in the [Data Vault Recipes](#) and [Mob Modelling](#) articles) because no one knows their business objects better than the business itself!

What is **Hoshin Kanri**? Popularised by Kaoru Ishikawa in the 1950s is a strategic and quality management technique and tool. Hoshin is Japanese for compass, a course, a policy or a plan indicating purpose and vision. Kanri represents management control or policy deployment, the intention of Hoshin Kanri is to ensure all employees in an organization understand the long-term goal and work together in their initiatives to the common goal, learn more about Hoshin Kanri here: bit.ly/3ET2BnF.

Why it is referenced here is because it is an opportunity to show how data vault contributes to the long-term goals of the enterprise. Earlier we explored business architecture, followed by data vault and how data vault's hub tables (business object representation) align to business objects and capabilities, with a Hoshin Kanri X-Matrix we can effectively describe an information map by reusing the same x-matrix concept to do the mapping of,

- a. **Business Objects** - tangible things commonly recognized by the business
- b. **Business Capabilities** - defines *what* the business does, is based on business objects
- c. **Business Processes** - behaviour element that groups behaviour based on an ordering of activities, *how* the business does what it does. We could take this up a level to a Value Stream instead, but the business process will be the grain expected to be automated by software and databases.
- d. **Business Units** - autonomous division of a large company that operates as an independent enterprise with responsibility for a particular *range of activities*, the *who*
- e. **Data Vault Hub Tables** – unique list of business keys (the business objects). This is separate to business object mapping because multiple stakeholders and business process actors could be involved in a business process that could in fact map to the same hub table! For instance, if the same business object has two different business keys because a department or business unit identifies a business object uniquely by a different key. This could also be based on the Business Definition of what that business object is, a party is a super-type to organization and person and will be

loaded to hub_party but both an organization and a person can participate in a business process because they are bound by an agreement as part of a unit-of-work.

Now that we have our components described above, let's build out our map

Step 1: Select the Business Process, there could be several Here we will map business processes around Credit Cards, that is:

- Credit Card Application – this can be in conjunction with a Home Loan Application a credit card is also issued. The card could be issued as a standalone card or in a primary + supplementary card configuration and does not necessarily need to have an origination to exist. This makes this business processes a reusable component of other business processes.
- Lost / Stolen Credit Card – what happens when lost or stolen card event takes place
- Card statements – issuance of a card statement is a regulated requirement,
- Credit score – should the applicant be issued a card based on past credit behaviour, and what should the allowable credit limit be issued?

Step 2: Identify the business objects

- Card id: super type of the type of card issued, each with their own configuration
 - Standalone
 - Bundle – Primary + optional Supplementary cards under a management card id
- Account number: Financial Account instrument used to track monies
- Customer id: identifying the customer uniquely
- Contract id: forming the bond between customer and financial institution, the contract can include more than just a card account, but the card account could be supplementary to a loan account
- Application id: before the prospect can become a customer with an account an application is identified and put through the credit scoring process
- Offer id: was the card application a part of a campaign, if the offer a “Next Best Activity”, an upgrade to an existing service as a form of retention campaign?
- Address: mandatory, this would be typically tracked against a party (customer) unless the address is tracked by a unique field identifier like that in an MDM system

Step 3: Map to / Identify hub tables

An enterprise data model will map the enterprise, thus at this point we map the integration point from business process, business object and business unit. Note that we do not map data vault links and satellites here as this activity should be in the exclusive realm of data modelling, *not information mapping*. Business Data Architects will not care for column and attribute mapping from the source, for more information on this activity turn to the mob! (see: bit.ly/3zgP7OP)

		X				X	Credit Card Application	M	M	O	M	O	M	M	S	S	O	M		O															
		X		X			Lost / Stolen Card	M		M				M	S	S	S																		
X		X			X	X	Mortgage Application	O	M		M		M							O															
							Settle Card Account	M		M				M																					
					X	X	Campaign								S	S	O			M															
				X			Perform Credit Check		M																										
		X					Credit Card Statement	M		M			M	M																					
							Authorize Card Payment							M	S	S	O		O																
Loan Servicing	Direct Banking	Credit Operations	Credit Card	Fraud Prevention & Management	Mortgage Banking	Database Marketing	Product	<div><div><div>BUSINESS PROCESS / VALUE STREAM STAGE</div><div><div>BUSINESS UNIT</div><div>BUSINESS OBJECT</div></div><div>HUB</div></div></div>												Financial Account	Applicant	Customer	Contract	Partner	Address	Card	Standalone Card	Primary Card	Supplementary Card	Management Card	Token id	Offer			
<div><div>M = Must</div><div>O = Optional</div><div>S = Select one of</div><div></div><div>X = Capability Matching</div><div></div><div><div>X</div> = Map to Hub</div><div><div>C</div> = Collision</div></div>							channel																												
							account	X																											
							asset																												
							customer									X																			
							contract											X																	
							product																												X
							offer																												
							partner													X															
							applicant									X																			
							card														X	X	X	X	X	X	C								

Figure 0-1 Possible Hoshin Kanri Matrix linking Business Process to Data Vault Hub tables, mapping DV conceptually

Note the “C” for Collision, this is stating that the business key from Token id could collide with another business key in the same hub table, card. For this look to Business Key Collision codes, bit.ly/3x1FK0s. Another thing to note, this depiction is not intended to be definitive list of hub tables in the data vault model, no. Hub definitions like information mapping (the business language) is determined before mapping the business processes to data vault. Although, it can highlight the need for a *new* hub or business object definition.

Step 4: Profile source; here we will identify (*using Steel Thread approach*):

- Business Keys and Grain (dependent-child characteristics)
- Unit of Work
- Descriptive content (*satellite splitting is the most underrated data vault activity*)
 - Confidential, Personally (+Quasi) identifier information
 - Critical Data Elements
- *more...*

Step 5: Does the source map the business process *completely*? Candidate Business Vault artefacts, bit.ly/3BUt81s. (*fyi, Business Vault never contains business vault hubs*)

With the enterprise layer modelled as your Data Vault, the layer above can be modelled into anything the Business Unit needs for their reporting/dashboarding tools. This is a technique in keeping with the approach of disciplined distributed parallel teams but centralised standards and governance.

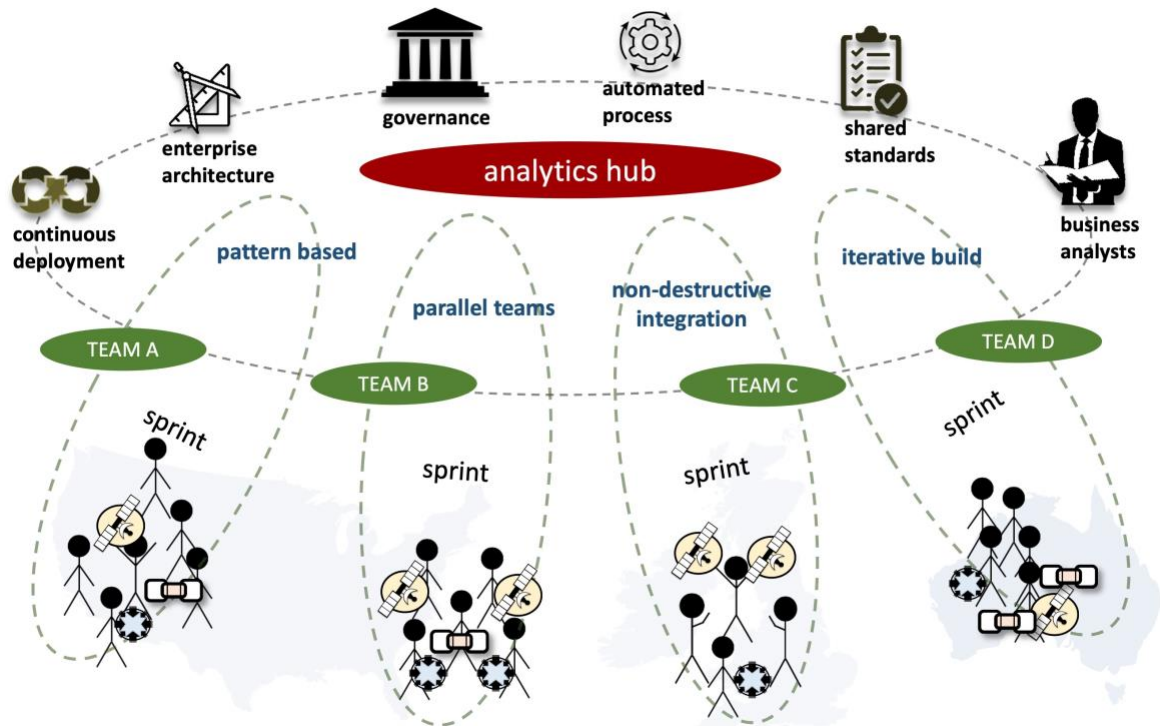
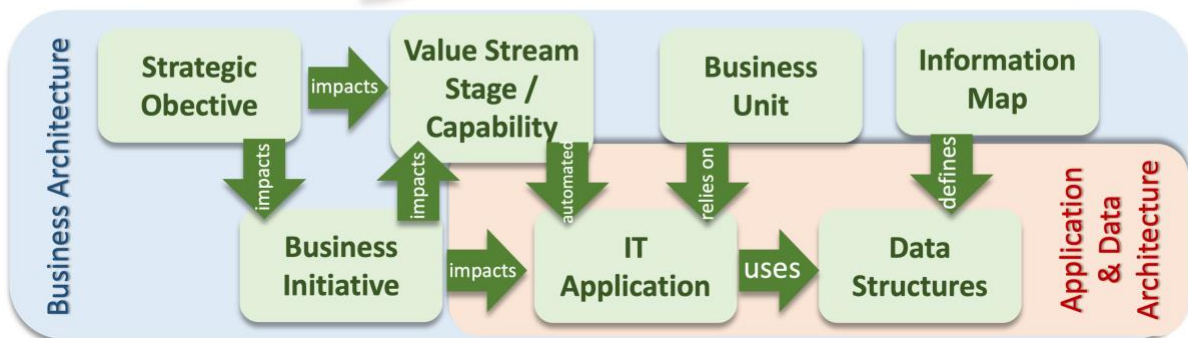
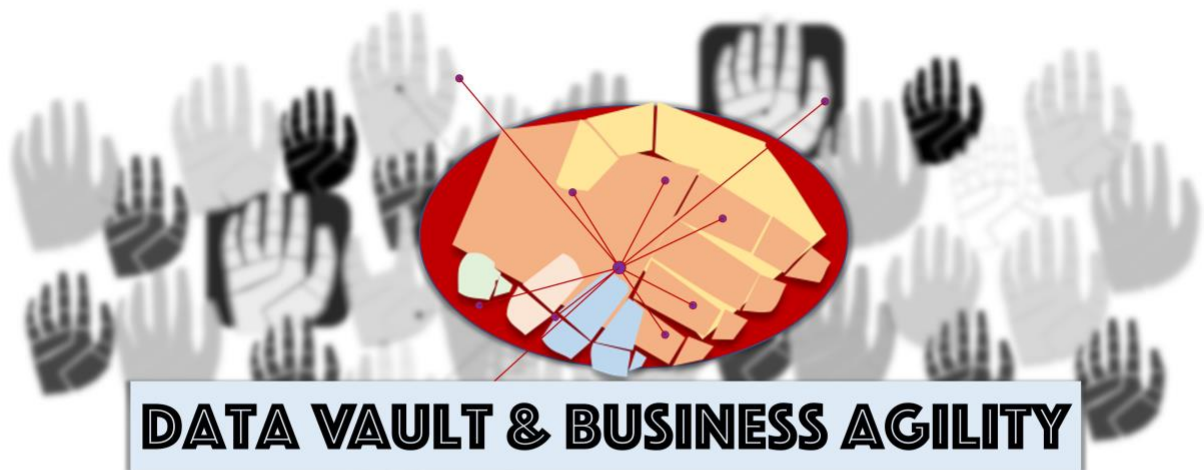


Figure 0-2 Centralised standards, distributed delivery

Conclusions

As data volumes grow and technology improves to provide the patterns to automate the business, a constant theme from the strategic down to the technology level relies on metrics to keep the business performing and efficiently tuned. Every metric contributes to the overall health of the enterprise. Everything is measured and thus everything can be governed, standardised, and improved. The ability to pivot needs **agile people, agile processes, agile technology & data** behind an enterprise. Every stakeholder in your business is contributing to the overall success no matter in which business unit, department, or division they operate in. *Inspired* individuals will inevitably believe in your enterprise goals if they are empowered and given the freedom to do so.



Yes, this article crosses many disciplines, *everything in the enterprise is inevitably connected.*

For more like this look to “Applying Top-Down, Big Picture Models to a Data Vault” – see: bit.ly/3vRXtf6

#businessagility #dataagility #agile #datavault #thedataflow #snowflake
#businessarchitecture

The views expressed in this article are that of my own, you should test implementation performance before committing to this implementation. The author provides no guarantees in this regard.