SmartData Fabric® (SDF)
Top Twelve Differentiators

November 2023

Top twelve SmartData Fabric® DIFFERENTIATORS – most, unique

1. **Leaves data where it is, but PRE-PROCESSES STRUCTURED AND UNSTRUCTURED DATA** for discovery, profiling, identification, classification, security, quality, standards and analytics, and master data management, through indexes and indexed views in data source-specific federated adapters
2. **Overcomes any data source limitations** by enabling advanced queries and ABSORBING UP TO 100% OF QUERY LOAD on indexes and indexed views in federated adapters, and external joins in federation servers
3. **Enables high performance queries** in open source PostgreSQL, Trino and other query engines
4. **Provides an enterprise-wide semantic layer** with metadata, standard data views, data and business objects, and master and reference data
5. **Provides advanced and uniform-across-all-data-sources SQL query capabilities**
6. **Fully integrates its own automatic near real-time Master Data Management (MDM)/Master Patient Index (MPI) or can use separate third-party system as source and/or target**
7. **Supports operations as well as analytics, meeting the requirements of an ideal data fabric/mesh** (ref. Rick van der Lans, R20/Consultancy) – also **supports digital transformation** and the automation of existing, and creation of new, processes/workflows
8. **Supports “live query” mode vs. the much more common “data extract” mode** for reporting, BI, analytics and other apps
9. **Enables federation at multiple levels** that allows three or more tiers for performance, responsibility and accountability, supporting data mesh and local data management
10. **Allows development of FHIR and other APIs** for data sources that do not have them or are less capable
11. **Imposes advanced enterprise-wide data governance, access control, and data security and privacy, on all data sources – Forrester Zero Trust Data Security Framework**
12. **Distributed analytics, including Distributed AI (DAI) and virtual graph database and link analysis** – to be migrated in future from legacy database to PostgreSQL

Data problems, and SDF differentiators and solution comparison

Data PROBLEMS facing almost all enterprises	SmartData Fabric® DIFFERENTIATORS	SmartData Fabric® solution
Data is everywhere in multiple locations/countries, sources, source types, formats and platforms	**Leaves data where it is – parallel, distributed/federated processing – supports both operations and analytics, meeting requirements of an ideal data fabric/mesh**	**Leaves data where it resides** in sources and/or local/regional data lakes/warehouses
Cannot/will not copy/move ALL data to a single location, e.g., data sovereignty and ownership		**Does not copy or move data from sources to a single location**
Some important data is not readily useable as is and needs discovery, identification, classification, security, cleansing, transformation, standardization and analytics	**Pre-processes and standardizes data** for discovery, profiling, identification, security, quality, standards and analytics, supporting an Enterprise Semantic Layer	**Leverages the power of independent indexes with federated data adapters** to - discover, identify, classify, secure, cleanse, transform, standardize, analyze and connect data used to build and update indexes, indexed views for pre-aggregations, pre-calculations and pre-joins, and data and business objects - present multiple standardized views of data as an enterprise semantic layer - generate/use, and seamlessly and automatically integrate master data to enable data integration - execute uniform PostgreSQL SQL queries from standard apps across all federated data adapters - either read data from indexes, or read pointers from indexes and use to read, transform, standardize and secure raw results data from sources, and - provide integrated clean, standardized, secure, accurate and complete results data to standard apps
Data needs to be connected within and across data sources	**Standardized data and pre-join views**, and, in the future, Link Indexes™ that support virtual graph database	
Data needs to be integrated through entity normalization/deduplication	**Pre-processed and standardized data enables high quality MDM/MPI in healthcare – automatic, distributed and integrated**	
Some raw data needs to be aggregated to be useful to reporting, BI and analytics	**Derived value indexes, and pre-aggregated, pre-calculated and pre-joined indexed views**	
Some key performance indicators (KPIs) need to be calculated, monitored and acted on in near real-time	**Business views based on KPIs can be maintained and monitored for event processing**, exposed as REST APIs, microservices, business process management and other apps	
Many data sources do not have the query capabilities or load capacity for external queries by conventional data virtualization/federated data access/query engines	**External processing for data, indexes, indexed views and queries** can be 100% independent of data sources - supports "live query" mode vs. much more common "data extract" mode, and multi-level federation	
Large investment in existing systems and solutions	**Data fabric/mesh adapters and federation servers are independently and individually configurable and accessible – highly flexible and fit where needed**	**Leverages, complements and agnostic to existing systems**, data sources, data virtualization software and query engines – avoids vendor lock-in
Need for rigorous enterprise-wide access control, and data governance, security and privacy	**Each adapter is a security gatekeeper to a data source, regardless of whether indexed or not, and has built-in data governance**	**Enforces advanced enterprise-wide access control**, and data governance, security and privacy – Zero Trust Data Security

Before first query made

Unique to SmartData Fabric®

The top twelve differentiators in detail

Differentiator #1

Leaves data where it is, but PRE-PROCESSES STRUCTURED AND UNSTRUCTURED DATA for discovery, profiling, identification, classification, security, quality, standards and analytics, and master data management, through indexes and indexed views in data source-specific federated adapters

- BEFORE the first query is made on data sources or data lakes - avoids incomplete and incorrect results, maximizes query efficiency, and eliminates multiple federated queries to data sources to accommodate poor data quality and standards
- Indexes and indexed views typically maintained in near real-time using Changed Data Capture (CDC), polling or other methods, e.g., open source Airbyte
- Conventional data virtualization and query engines address data issues only AFTER it is extracted from sources in query result-sets/data extracts/cache or data lakes, or through ETL to a data warehouse
- Query engines are focused on scalability and high-performance, typically, only for analytics, but do not address data issues
- Query engines assume that data issues are addressed either before landing data in a data lake or in a post-query process, such as ETL to a data warehouse or ELT in a data lake, which causes considerable work for data engineers and analysts to sift through and prepare data – AI can help but not eliminate this
- Query engines, in particular, and conventional data virtualization, to a lesser extent, do not address all the multiple use cases associated with an ideal data fabric/mesh, e.g., operations, process automation, digital transformation, interoperability, reporting, BI and analytics
- SmartData Fabric® adapters and federation servers can run anywhere and be 100% consistent, including Hybrid Cloud 1.0 (on-premises, in data centers and cloud) and Hybrid Cloud 2.0 where data stays in sources and compute is in the cloud

Differentiator #2

Overcomes any data source limitations by enabling advanced queries and ABSORBING UP TO 100% OF QUERY LOAD on indexes and indexed views in federated adapters, and external joins in federation servers

- Including pre-aggregations, pre-calculations and pre-joins, and data and business objects, that can be updated in near real-time, enabling high performance queries and resources saved
- All conventional data virtualization and query engines are 100% dependent on data source systems and data in these systems for query execution
- Most operations/transaction systems are not designed or optimized for external queries, in particular, reporting, BI and analytics
- Many operations/transaction system owners will not accept external queries because of the load these impose and/or security issues, particularly, if source data quality and standards could lead to incorrect data in results
- Some data sources are not available for external queries, in particular, SQL
- Some data sources have up to 100% unstructured data, which is externally unqueryable "as is" and requires pre-processing, e.g., entity extraction, and maybe indexing to be queryable and connected to other data

Differentiator #3

Enables high performance queries in open source PostgreSQL, Trino and other query engines

- Can also operate as federated adapters and federation servers for third-party data virtualization and query engines
- Open source allows for the latest technologies to be rapidly deployed without vendor lock-in and cost
- Open source allows multiple other components to integrate well, many of which are also open-source
- Open source allows companies, such as WhamTech, to focus on value-add capabilities instead of query capabilities, performance and standards

Differentiator #4

Provides an enterprise-wide semantic layer with metadata, standard data views, data and business objects, and master and reference data

- Recent study[1] cites improvements of over 4 times in speed, scale and cost savings by using an enterprise-wide semantic layer
- Semantic layer allows access to more data by more business users in the enterprise – not just data engineers and analysts
- Data fabric/mesh and semantic layer provide a perfect combination, and are similar to a data catalog, except that the data fabric/mesh is actionable, and the semantic layer significantly simplifies and improves understanding and use of data
- Semantic layer also interacts well with API lifecycle management and API catalogs/gateways
- Semantic layer also needs to integrate with data governance, access control and data security, which is less complicated as enterprise-wide semantic dictionaries can be used instead of multiple mappings to data specific to sources

[1] “Business Impact of Using a Semantic Layer”, by DBP, July 1, 2022

Differentiator #5

Provides advanced and uniform-across-all-data-sources SQL query capabilities

- No need to transform queries for different data sources, dumb down queries for less capable data sources, or completely copy non-queryable data sources
- Using the same federated adapter database technology for indexing and query processing on every data source, regardless of the data source - avoids the need with conventional federated queries to transform and optimize high-level queries for multiple individual data sources, or compromise to accommodate different source system query capabilities, data quality and standards
- With conventional federated queries, because of data source limitations, some sources can only support simple data extract queries and result-sets are post-processed to filter out data not meeting more complex query conditions – this loads data source systems, represents a data security risk and a decline in query performance, and creates a need for result-set cache and management
- Complete control over pre-processing data in indexes and query processing in adapters, allows scalable and high performance queries, especially running these in the cloud using containers and parallel distributed data pre-processing and indexing, and query processing, e.g., Hybrid Cloud 2.0 architecture
- Python, R, GraphQL, AI and other languages are also supported by PostgreSQL, but not tested by WhamTech

Differentiator #6

Fully integrates its own automatic near real-time Master Data Management (MDM)/Master Patient Index (MPI) or can use separate third-party system as source and/or target

- Not an afterthought or separate system from data fabric/mesh
- MDM is the link between a physical operations/transaction data store of some form (sources, ODS or lake) and enabling virtual data warehouses, data marts and graph views of data, except that these virtual views are not subject to physical and latent data storage or complex data schemas and schema transformation – allows schema-on-read vs. schema-on-write
- MDM systems are, typically, very expensive to buy and populate, and NOT integrated with operations or even with reporting, BI and analytics – more of an afterthought or separate system
- WhamTech usually distributes master data to each data source adapter, improving performance, security and management, when adding or removing data sources in a data fabric/mesh – data source owners tend to accept this more
- MDM benefits considerably from addressing upfront data quality, standards and mapping to a semantic layer
- Full integration of MDM means that applications using, and users of, SmartData Fabric® data fabric/mesh do not need to be aware of it – it is an integrated and automatic part of the data fabric/mesh
- MDM enables the types of views of data needed for APIs and business objects such as customer/patient 360 views of data, regardless of where that data resides
- MDM is essential to almost any and all reporting, BI and analytics – it has been proven many times that the results from these apps significantly improves when data quality, standards and MDM have all been addressed upfront

Differentiator #7

Supports operations as well as analytics, meeting the requirements of an ideal data fabric/mesh (ref. Rick van der Lans, R20/Consultancy) – also supports digital transformation and the automation of existing, and creation of new, processes/workflows

Ideal data fabric/mesh:

1. Data preparation, such as transformations, aggregations, filters and joins
2. Adaptable logic
3. High performance
4. Data access by many data consumption forms
5. Access to all the data sources
6. Processing of all types of data
7. Data security and privacy
8. Real-time data access
9. Read and write data access
10. Data minimization
11. Event processing
12. Technical and business metadata management
13. Master and reference data management

Differentiator #8

Supports "live query" mode vs. the much more common "data extract" mode for operations, reporting, BI, analytics and other apps

- "Live query" mode is a major benefit for leaving data in multiple data sources in a data fabric/mesh, and data residency, multi-jurisdictional data and privacy regulations
- "Data extract" mode entails submitting queries to data sources to isolate and extract/copy data that may or may not be needed to execute more complex queries for reporting, BI and analytics – extracted data is used to populate a data mart that these apps query against
- "Data extract" mode is the only option for conventional data virtualization and query engines against multiple data sources, and after that, some form of ETL, or ELT followed by ETL, is required to populate a data mart for reporting, BI and analytics apps to query against
- "Live query" mode is generally not an option for conventional federated data systems, but IS an option for SmartData Fabric®, as multiple data sources can appear as if a single database through a federation server with a semantic layer and uniform-across-all-data-sources SQL query capabilities
- "Data extract" mode causes many issues for data residency, multi-jurisdictional data and privacy regulations, whereas "live extract" mode minimizes these issues, and post-processes results data that may need anonymization and aggregation to allow combination at a higher, e.g., global-level

Differentiator #9

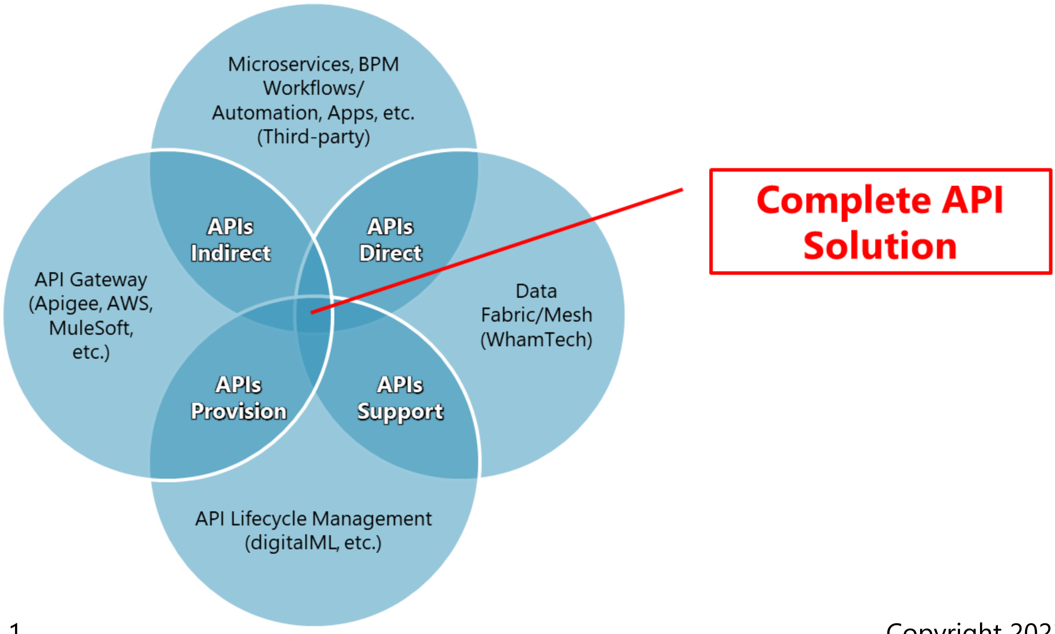
Enables federation at multiple levels that allows three or more tiers for performance, responsibility and accountability, supporting data mesh and local data management

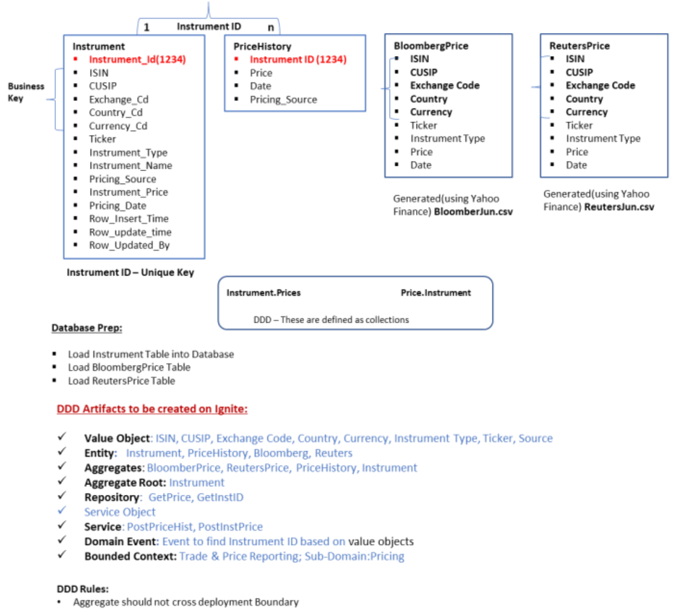
- Vs. the usual two tiers for conventional data virtualization and query engines, which are, generally, a hub and spoke architecture and require direct access to all data sources from a central location
- Two-tier architectures introduce two bottlenecks – one at the data source-level where external queries are executed and data is extracted – performance 100% dependent on source systems, and the other is in the cloud (usually) on a data lake or, eventually, a data warehouse, where there is 100% control over performance at a cost
- The three or more tiers of SmartData Fabric® debottleneck the data source-level by abstracting (i) data pre-processing, indexing and query execution to associated federated data adapters, and (ii) query distribution, federated query execution and results integration to federation servers, which can be on many levels
- Federation enables data fabric/mesh to work at multiple levels, and allow department, local, regional, country, etc. levels for data fabric/mesh operations, digital transformation, interoperability, reporting, BI and analytics
- Allows federation at multiple levels, e.g., country, regulations to be applied before results data is used at a higher-level, fulfilling any data residency, multi-jurisdictional data and privacy regulations
- Similar to SmartData Fabric® federated data adapters, federation servers can reside anywhere, including all in the cloud in a Hybrid Cloud 2.0 architecture
- SmartData Fabric® federated data adapters and federations servers support high performance, distributed edge processing of data and queries

Differentiator #10

Allows development of APIs for data sources that do not have them or are less capable

- Many APIs require developing views into data or business objects that data sources may not support, but SmartData Fabric® federated data adapters (and federation servers) can externally and independently enable these views
- SmartData Fabric® was used to develop FHIR HL7 REST APIs for various healthcare systems that did not support these at the time, using open-source Swagger, and WhamTech developed a process and a tool to build REST APIs
- Recently, SmartData Fabric® was used to develop APIs, microservices and orchestration for an Open Banking APIs POC, using Swagger, RabbitMQ and Amazon API Gateway


Microservices, BPM Workflows/ Automation, Apps, etc. (Third-party)
API Gateway (Apigee, AWS, MuleSoft, etc.)
Data Fabric/Mesh (WhamTech)
API Lifecycle Management (digitalML, etc.)
APIs Indirect
APIs Direct
APIs Provision
APIs Support
Complete API Solution



1 Instrument ID n

Business Key

Instrument
- Instrument_Id(1234)
- ISIN
- CUSIP
- Exchange_Cd
- Country_Cd
- Currency_Cd
- Ticker
- Instrument_Type
- Instrument_Name
- Pricing_Source
- Instrument_Price
- Pricing_Date
- Row_Insert_Time
- Row_update_time
- Row_Updated_By

Instrument ID – Unique Key

PriceHistory
- Instrument ID (1234)
- Price
- Date
- Pricing_Source

BloombergPrice
- **ISIN**
- **CUSIP**
- **Exchange Code**
- **Country**
- **Currency**
- Ticker
- Instrument Type
- Price
- Date

Generated(using Yahoo Finance) **BloomberJun.csv**

ReutersPrice
- **ISIN**
- **CUSIP**
- **Exchange Code**
- **Country**
- **Currency**
- Ticker
- Instrument Type
- Price
- Date

Generated(using Yahoo Finance) **ReutersJun.csv**

Instrument.Prices **Price.Instrument**

DDD – These are defined as collections

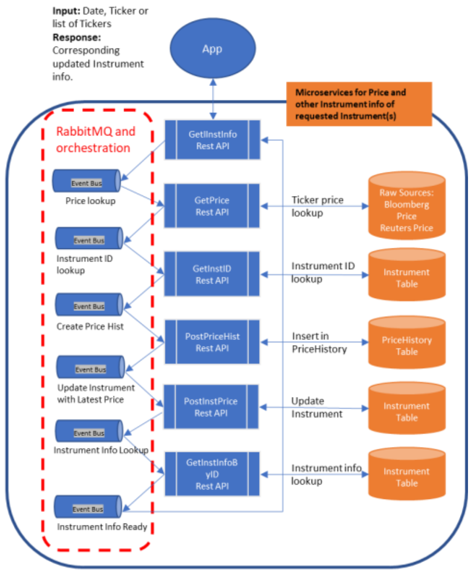
Database Prep:

- Load Instrument Table into Database
- Load BloombergPrice Table
- Load ReutersPrice Table

DDD Artifacts to be created on Ignite:

- ✓ **Value Object**: ISIN, CUSIP, Exchange Code, Country, Currency, Instrument Type, Ticker, Source
- ✓ **Entity**: Instrument, PriceHistory, Bloomberg, Reuters
- ✓ **Aggregates**: BloomberPrice, ReutersPrice, PriceHistory, Instrument
- ✓ **Aggregate Root:** Instrument
- ✓ **Repository**: GetPrice, GetInstID
- ✓ Service Object
- ✓ **Service**: PostPriceHist, PostInstPrice
- ✓ **Domain Event**: Event to find Instrument ID based on value objects
- ✓ **Bounded Context:** Trade & Price Reporting; Sub-Domain:Pricing

DDD Rules:
- Aggregate should not cross deployment Boundary



Differentiator #11

Imposes advanced enterprise-wide data governance, access control, and data security and privacy, on all data sources – Forrester Zero Trust Data Security Framework

- Including those that have no built-in support, and for non-indexed (aka conventional) federated data source adapters
- Developed over almost two years with General Dynamics and seen as a federated data governance, access control and data security solution, first, and data virtualization and integration, second
- One of a few, if any other, data virtualization/fabric/mesh vendors that conforms to Forrester Zero Trust Data Security Framework, as INDEXES at-risk data
- Addressing data quality, standards and mapping to a standard enterprise-wide semantic layer, automatic and integrated MDM/MPI, and uniform SQL greatly simplifies data governance, access control and data security
- Enables AD/LDAP-based IAM, SSO, RBAC, TLS, RLS, CLS and multi-level classification for ANY data source regardless of its support for any of these security protocols, as 100% control of data and query execution is at each level of federation and federated data adapters
- Queries are either not made or modified within adapters before execution, depending on the specific user, role and/or app permissions – results data is not subsequently modified to conform to queries, lessening the security risk
- Dynamic data masking, tokenization and/or encryption is applied to results data depending on user and/or app permissions – complete anonymization is also an option
- Data source owners still retain the option to veto or override access from SmartData Fabric® federated adapters

Differentiator #12

Distributed analytics, including Distributed AI (DAI) and virtual graph database and link analysis – to be migrated in future from legacy database to PostgreSQL

- See separate presentations

The End