DecisionCAMP 2017

Patterns for Business Logic in Modern Technical Architecture

Sapiens DECISION

www.sapiensdecision.com

Decision Management is ABSENT from Most Architectures

- Wikipedia describes DM as *"Decision management is an 'emerging important discipline, due to an increasing need to automate high-volume decisions across the enterprise and to impart precision, consistency, and agility in the decision-making process.' Decision management is implemented 'via the use of rule-based systems and analytic models for enabling high-volume, automated decision making'*
- Decision Management has won a battle and is now recognized as a principle component of the business architecture, but hasn't begun the critical battle which is its place in technical architecture
- <u>Our thesis is that technical architectures have to be re-imagined to promote Business Logic as a first-class component in order to enable Decision Management</u>

Decision Management Scope: Established Usage

- Traditional: Business logic prescribed by policy or operational considerations
- Contained and integrated within discrete applications and seen as an integral component of these applications
- Traditional Analytics: Transactional data provides fodder for machine learning or classical statistical techniques producing predictive models that are incorporated into traditional decision models

"Do you have any other collateral... besides this e-mail from a Nigerian prince?"

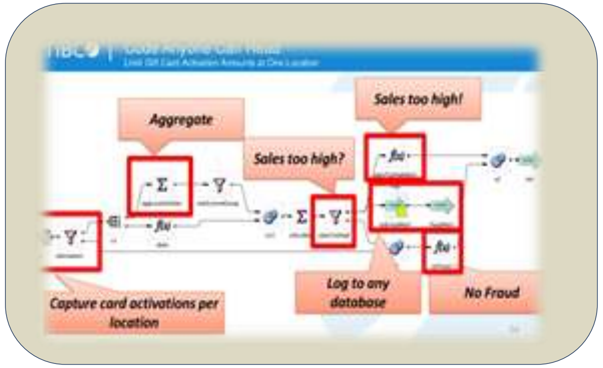
Emergent Technologies Impacting Decision Management

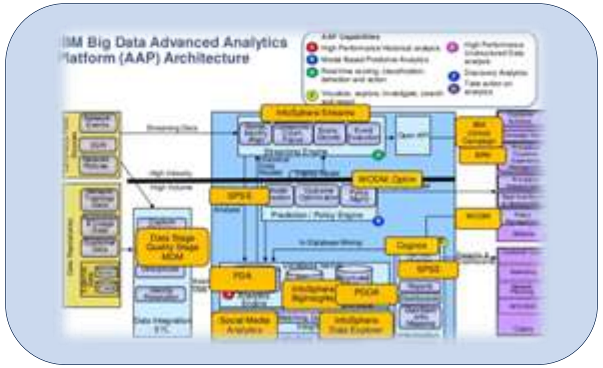
Failure of these emerging technologies to recognize in their stack the role of Decision Management impacts DM's ability to deliver on its promises to the Enterprise:

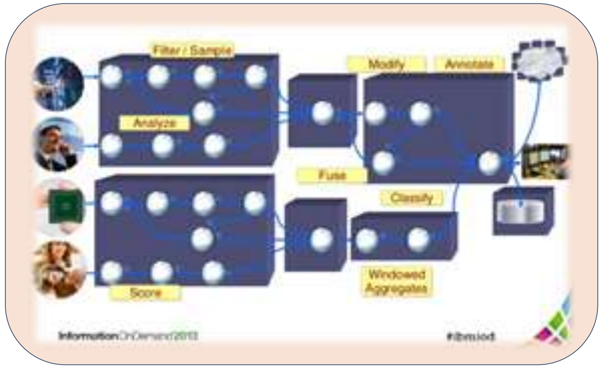
- Big Data – The scale of execution throughput requirements exceed the capacity of traditional decision architectures; Architectures increasingly bring business logic to the data in a distributed computing environment over Data Lakes
- Streaming Analytics – The size and nature of real-time data streams coupled with the perishable nature of the insights has created a myriad of new architectures
- API Economy – The discoverable nature of business logic where the boundaries of usage are not prescribed beforehand and are opened to 3rd-parties to creatively integrate has similarly required architectural innovation

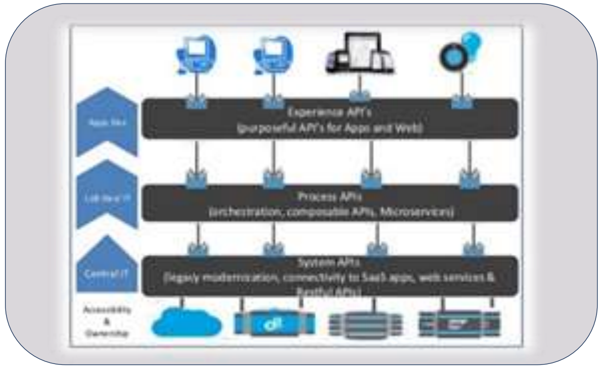
Where is the Business Logic Represented?

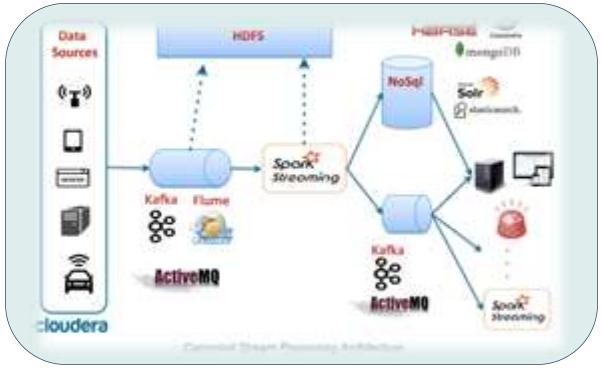
Graphics are hyperlinked

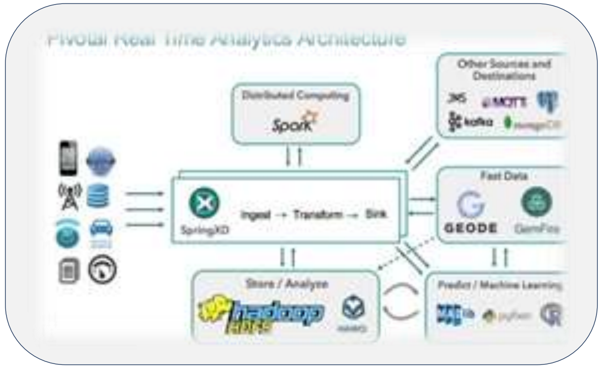


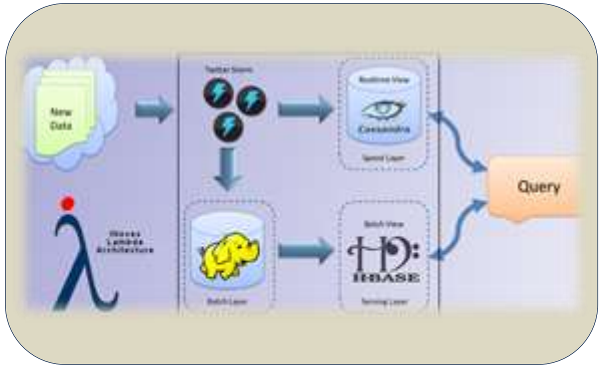


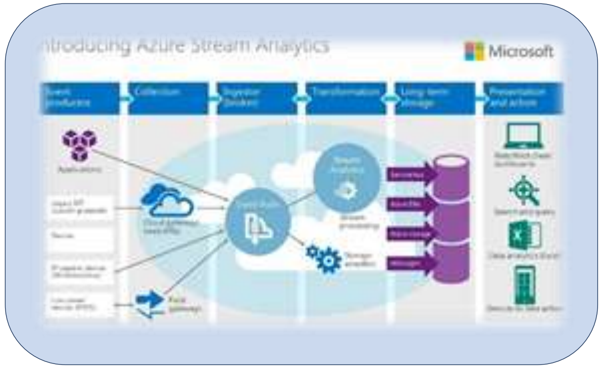



Experience API's
(purposeful API's for Apps and Web)
Process APIs
(orchestration, composable APIs, Microservices)
System APIs
(legacy modernization, connectivity to SaaS apps, web services & RESTful APIs)

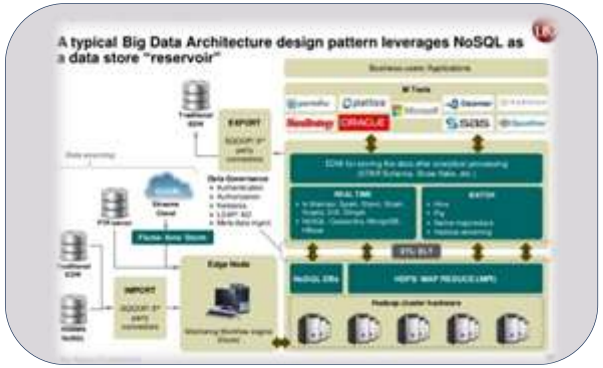



Pivotal Real Time Analytics Architecture
Distributed Computing
Spark
Other Sources and Destinations
SpringXD
Ingest → Transform → Sink
Fast Data
GEODE
GemFire
Store / Analyze
hadoop
Predict / Machine Learning





Spring Cloud Stream/Data Flow as Event Driven Architecture
Data Microservices
Filter, Enrich, Transform, Aggregation, Rules, CEP, Analytics
Data
Binding Abstraction Layer
Data
RabbitMQ
kafka
GEODE
Messaging Options
Data Pipeline


A typical Big Data Architecture design pattern leverages NoSQL as a data store "reservoir"
ORACLE
SAS

SAPIENS DECISION

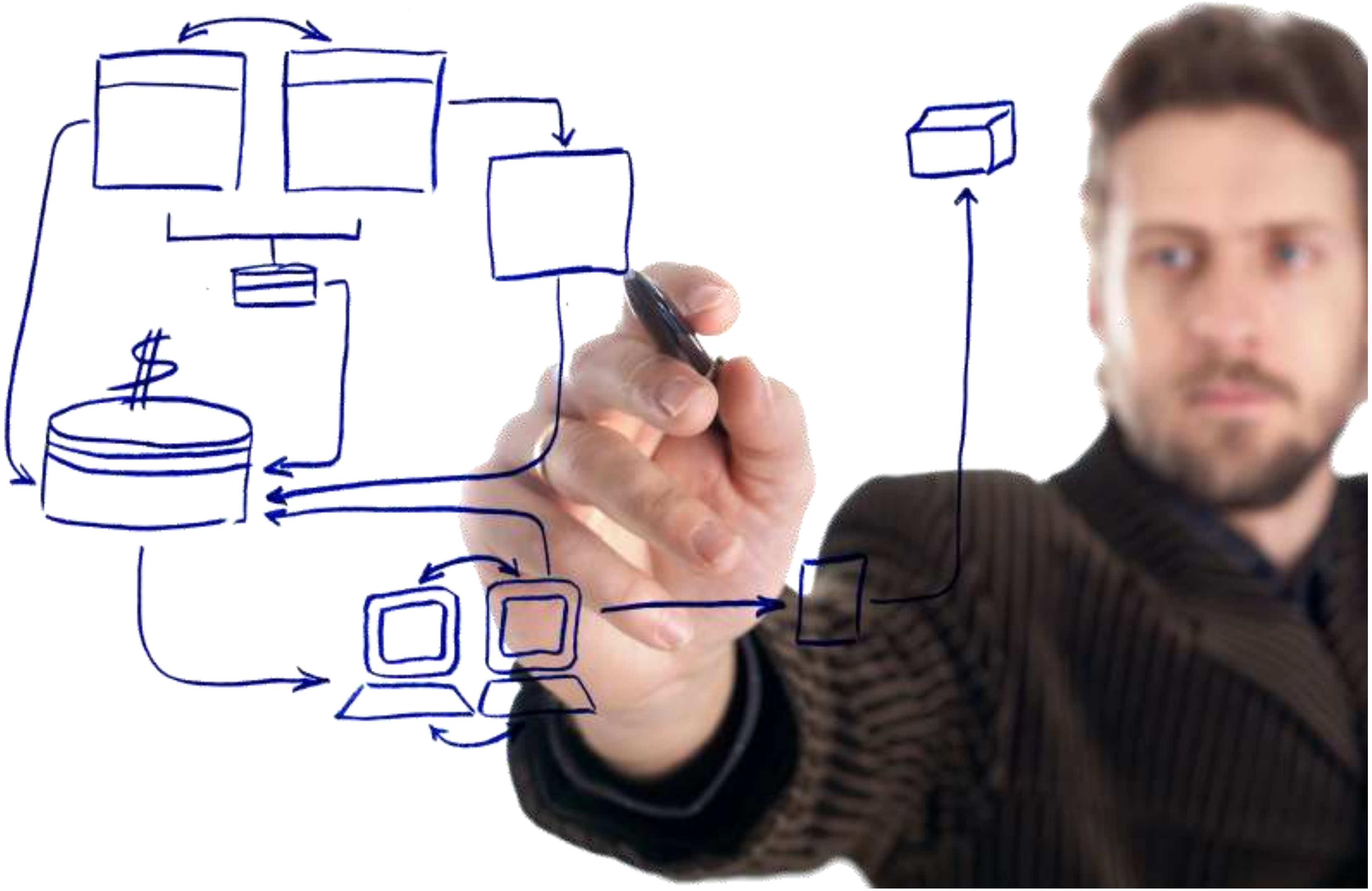
Why is this a problem?

- Allowing the business logic to remain buried in idiosyncratic architectures reduces its visibility and agility, and raises cost and risk
- DM requires basic characteristics on a consistent basis:
  - Visibility of logic to the business
  - Business control of logic lifecycle including validation and governance
  - Re-use requires common representation
- The representation of the business logic in the technical architecture in an explicit and consistent manner carries with it a particular set of capabilities needed to realize the vision of Decision Management

Solution: Common Ground Across Architectures

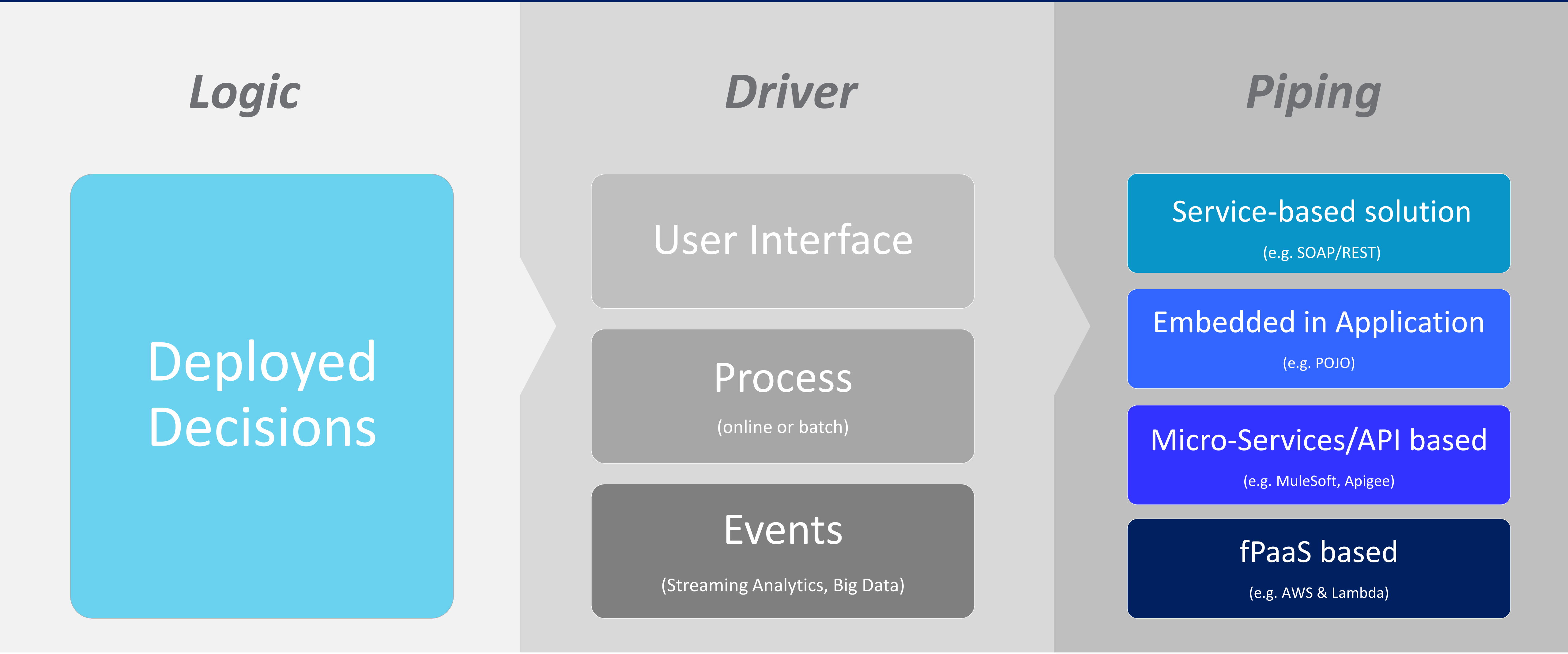
1. Establish common patterns to represent business logic across architectures

2. Establish core features that Decision Management provides across architectures through tooling and methodology



1. Establish Common Patterns to Represent Business Logic Across Architectures

***One place** to create, validate, test, manage and govern the business logic, many different architectures and methods to deploy the executable logic*



Logic

Deployed Decisions

Driver

User Interface

Process

(online or batch)

Events

(Streaming Analytics, Big Data)

Piping

Service-based solution

(e.g. SOAP/REST)

Embedded in Application

(e.g. POJO)

Micro-Services/API based

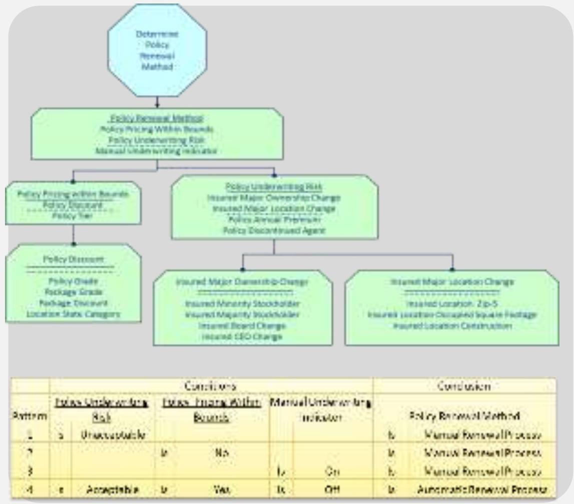
(e.g. MuleSoft, Apigee)

fPaaS based

(e.g. AWS & Lambda)

1.1 Decision Logic Embedded into JVM based Apps

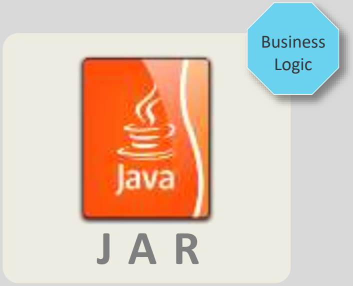
Model-to-Code


Determine Policy Renewal Method
Policy Renewal Method
Policy Pricing Within Bounds
Policy Underwriting Risk
Manual Underwriting Indicator
Policy Pricing within Bounds
Policy Discount
Policy Tier
Policy Underwriting Risk
Insured Major Ownership Change
Insured Major Location Change
Policy Annual Premium
Policy Discontinued Agent
Policy Discount
Policy Grade
Package Grade
Package Discount
Location State Category
Insured Major Ownership Change
Insured Minority Stockholder
Insured Majority Stockholder
Insured Board Change
Insured CEO Change
Insured Major Location Change
Insured Location Zip-5
Insured Location Occupied Square Footage
Insured Location Construction

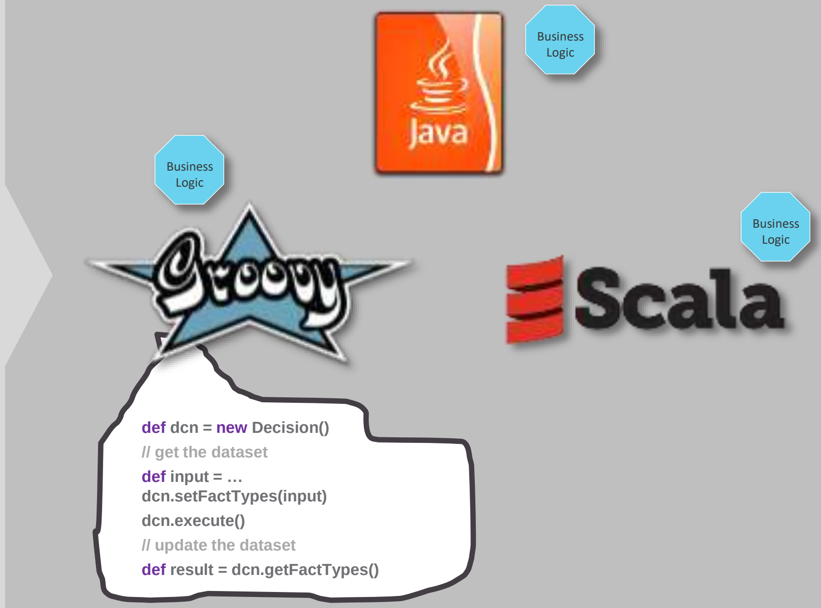
	Conditions						Conclusion	
Pattern	Policy Underwriting Risk		Policy Pricing Within Bounds		Manual Underwriting Indicator		Policy Renewal Method	
1	is	Unacceptable					is	Manual Renewal Process
2			is	No			is	Manual Renewal Process
3					is	On	is	Manual Renewal Process
4	is	Acceptable	is	Yes	is	Off	is	Automatic Renewal Process

BINARY REPOSITORY

Deployable Assets



Drop existing business logic managed in Java, for logic managed by the business and embedded into Java



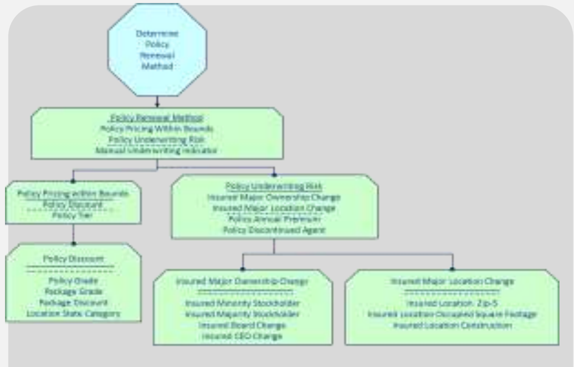
```
def dcn = new Decision()
// get the dataset
def input = …
dcn.setFactTypes(input)
dcn.execute()
// update the dataset
def result = dcn.getFactTypes()
```

Replacing hand-crafted business logic code with reliably-generated code

Reference Architecture Example 1

1.2 Decision Logic in Traditional Service Oriented Architecture

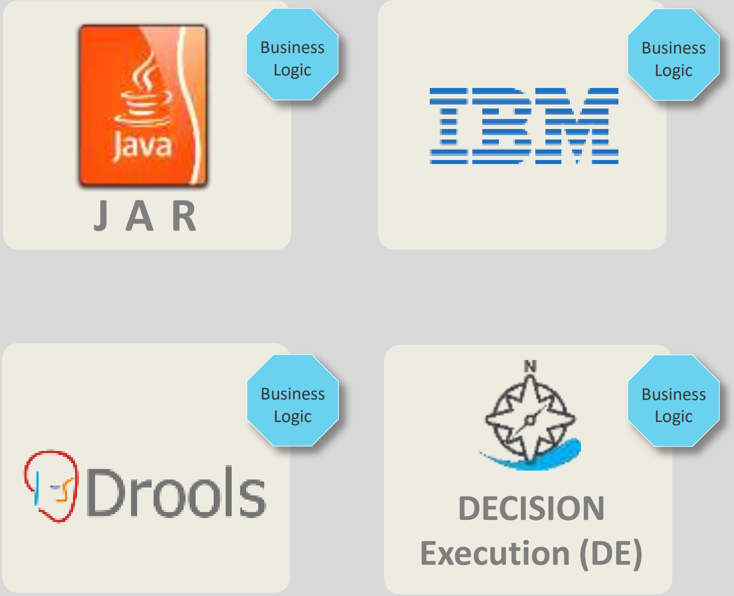
Model-to-Code


Determine Policy Renewal Method
Policy Renewal Method
Policy Pricing Within Bounds
Policy Underwriting Risk
Manual Underwriting Indicator
Policy Pricing within Bounds
Policy Discount
Policy Tier
Policy Underwriting Risk
Insured Major Ownership Change
Insured Major Location Change
Policy Annual Premium
Policy Discontinued Agent
Policy Discount
Policy Grade
Package Grade
Package Discount
Location State Category
Insured Major Ownership Change
Insured Minority Stockholder
Insured Majority Stockholder
Insured Board Change
Insured CEO Change
Insured Major Location Change
Insured Location: Zip-5
Insured Location Occupied Square Footage
Insured Location Construction

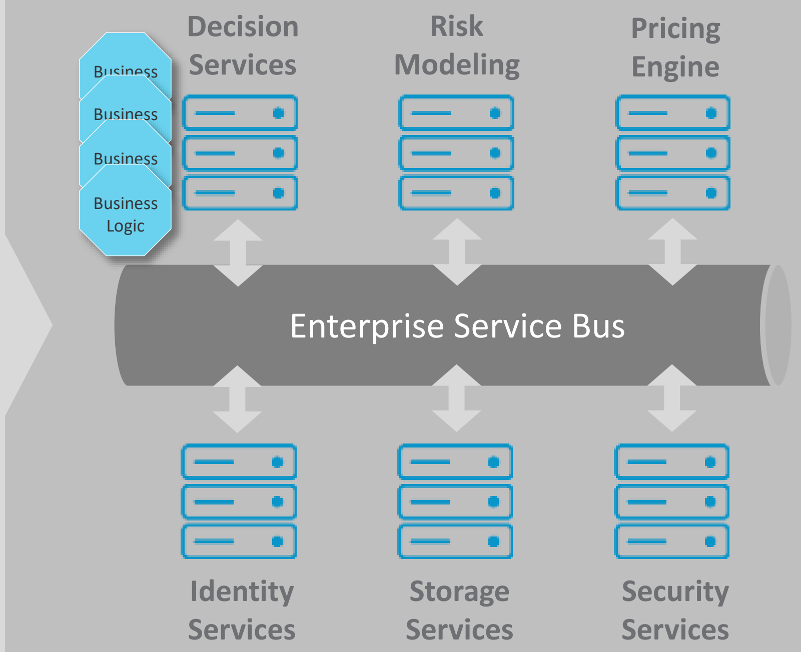
	Conditions						Conclusion	
Pattern	Policy Underwriting Risk		Policy Pricing Within Bounds		Manual Underwriting Indicator		Policy Renewal Method	
1	is	Unacceptable					is	Manual Renewal Process
2			is	No			is	Manual Renewal Process
3					is	On	is	Manual Renewal Process
4	is	Acceptable	is	Yes	is	Off	is	Automatic Renewal Process

BINARY REPOSITORY

Deployable Assets



Services

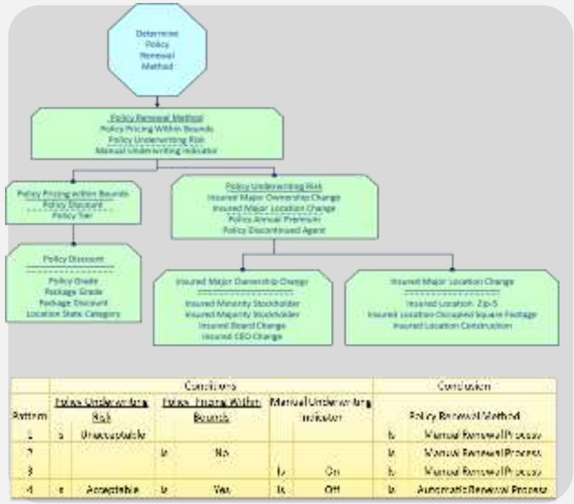


Invoke decision logic from any application in the organization as a service through a scalable centralized decision execution server

Reference Architecture Example 2

1.3 Bringing Decision Logic to Big Data

Model-to-Code

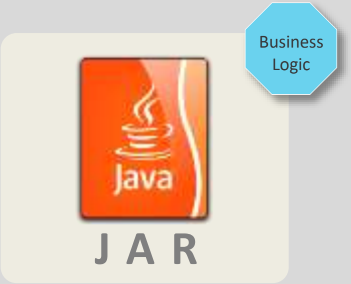


Determine Policy Renewal Method
Policy Renewal Method
Policy Pricing Within Bounds
Policy Underwriting Risk
Manual Underwriting Indicator
Policy Pricing within Bounds
Policy Discount
Policy Tier
Policy Underwriting Risk
Insured Major Ownership Change
Insured Major Location Change
Policy Annual Premium
Policy Discontinued Agent
Policy Discount
Policy Grade
Package Grade
Package Discount
Location State Category
Insured Major Ownership Change
Insured Minority Stockholder
Insured Majority Stockholder
Insured Board Change
Insured CEO Change
Insured Major Location Change
Insured Location: Zip-5
Insured Location Occupied Square Footage
Insured Location Construction

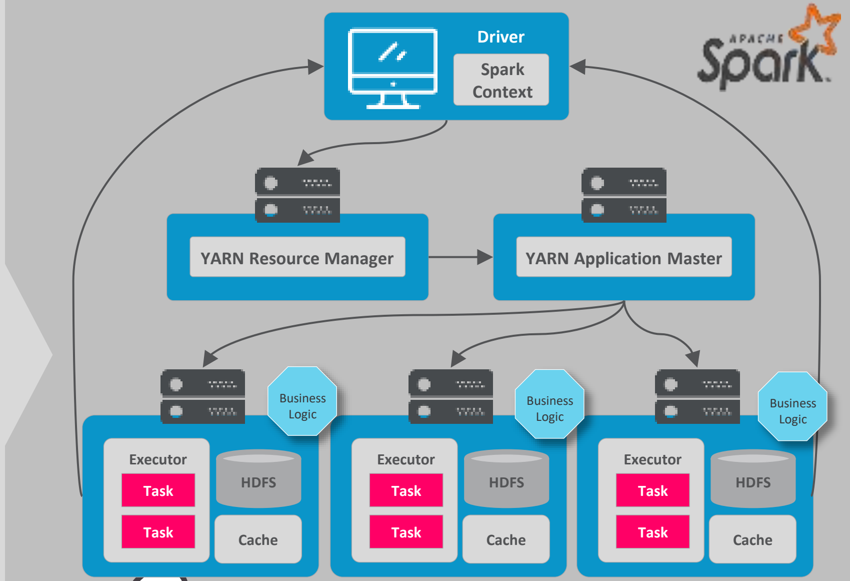
	Conditions						Conclusion	
Pattern	Policy Underwriting Risk		Policy Pricing Within Bounds		Manual Underwriting Indicator		Policy Renewal Method	
1	is	Unacceptable					is	Manual Renewal Process
2			is	No			is	Manual Renewal Process
3					is	On	is	Manual Renewal Process
4	is	Acceptable	is	Yes	is	Off	is	Automatic Renewal Process

BINARY REPOSITORY

Deployable Assets



Divide and conquer: Bring the decision computations to the data



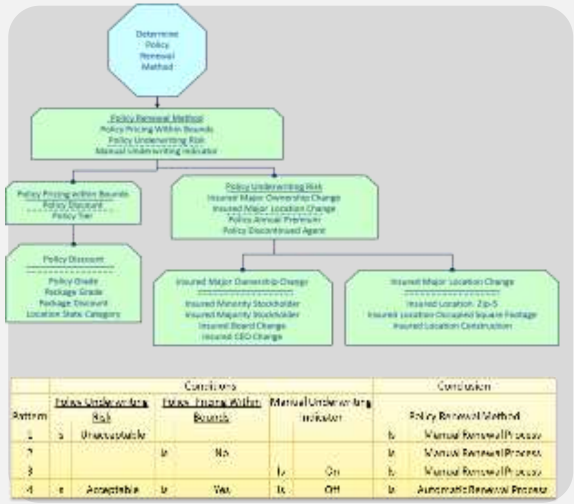
```
def dcn = new Decision()
// get the dataset
def input = … dcn.setFactTypes(input)
dcn.execute()
// update the dataset
def result = dcn.getFactTypes()
```

Elastic scaling for huge data sets by using distributed computing

Reference Architecture Example 3

1.4 Decision Logic in Event Processing/Streaming Analytics

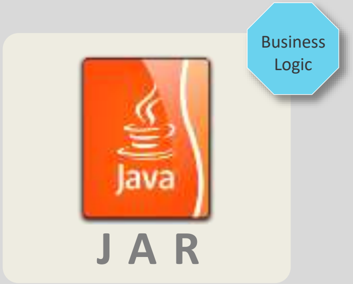
Model-to-Code


Determine Policy Renewal Method
Policy Renewal Method
Policy Pricing Within Bounds
Policy Underwriting Risk
Manual Underwriting Indicator
Policy Pricing within Bounds
Policy Discount
Policy Tier
Policy Underwriting Risk
Insured Major Ownership Change
Insured Major Location Change
Policy Annual Premium
Policy Discontinued Agent
Policy Discount
Policy Grade
Package Grade
Package Discount
Location State Category
Insured Major Ownership Change
Insured Minority Stockholder
Insured Majority Stockholder
Insured Board Change
Insured CEO Change
Insured Major Location Change
Insured Location Zip-5
Insured Location Occupied Square Footage
Insured Location Construction

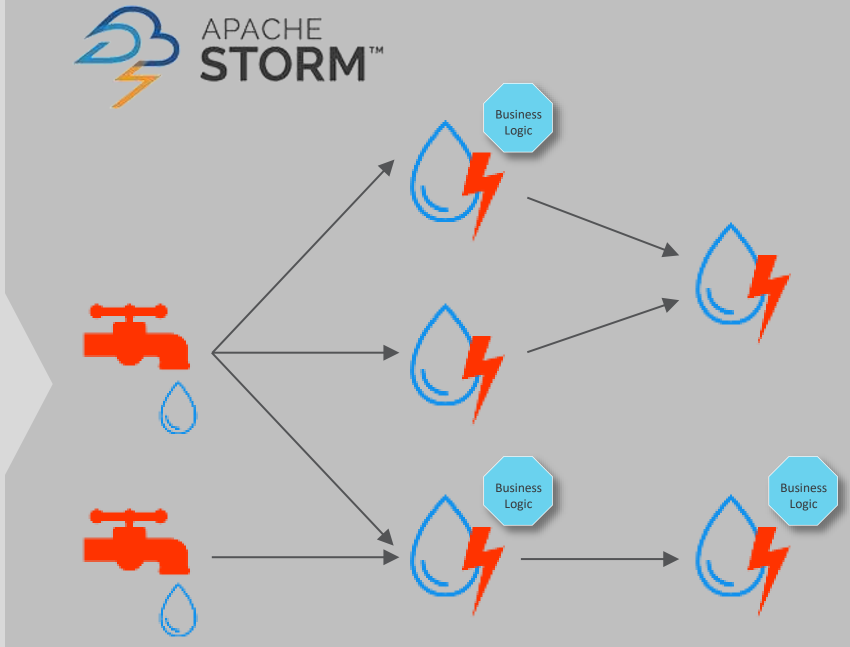
	Conditions						Conclusion	
Pattern	Policy Underwriting Risk		Policy Pricing Within Bounds		Manual Underwriting Indicator		Policy Renewal Method	
1	is	Unacceptable					is	Manual Renewal Process
2			is	No			is	Manual Renewal Process
3					is	On	is	Manual Renewal Process
4	is	Acceptable	is	Yes	is	Off	is	Automatic Renewal Process

BINARY REPOSITORY

Deployable Assets



Use as Bolt in Storm Topology

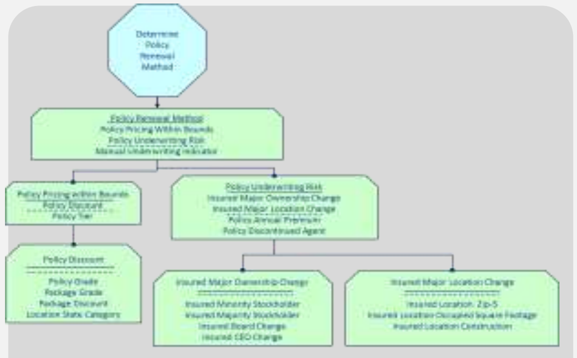


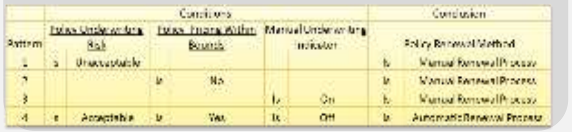
Decisions can be embedded as Bolts in a Storm topology, allowing distributed processing of data streams and making Decisions based on a continuous stream of data

Reference Architecture Example 4

1.5 Decision Logic a Critical Component of the API Economy

Model-to-Code

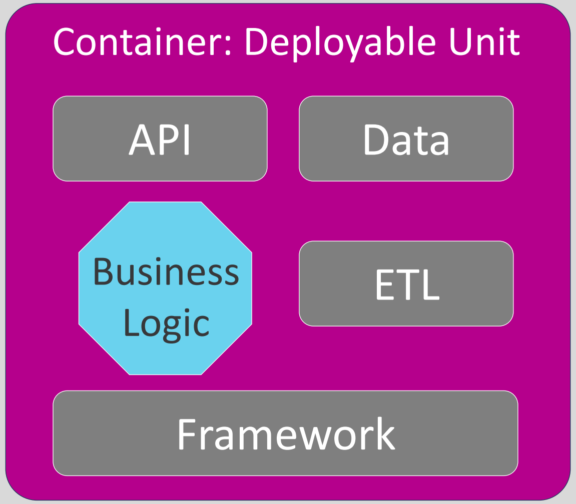

Determine Policy Renewal Method
Policy Renewal Method
Policy Pricing Within Bounds
Policy Underwriting Risk
Manual Underwriting Indicator
Policy Pricing within Bounds
Policy Discount
Policy Tier
Policy Underwriting Risk
Insured Major Ownership Change
Insured Major Location Change
Policy Annual Premium
Policy Discontinued Agent
Policy Discount
Policy Grade
Package Grade
Package Discount
Location State Category
Insured Major Ownership Change
Insured Minority Stockholder
Insured Majority Stockholder
Insured Board Change
Insured CEO Change
Insured Major Location Change
Insured Location Zip-5
Insured Location Occupied Square Footage
Insured Location Construction



	Conditions						Conclusion	
Pattern	Policy Underwriting Risk		Policy Pricing Within Bounds		Manual Underwriting Indicator		Policy Renewal Method	
1	is	Unacceptable					is	Manual Renewal Process
2			is	No			is	Manual Renewal Process
3					is	On	is	Manual Renewal Process
4	is	Acceptable	is	Yes	is	Off	is	Automatic Renewal Process

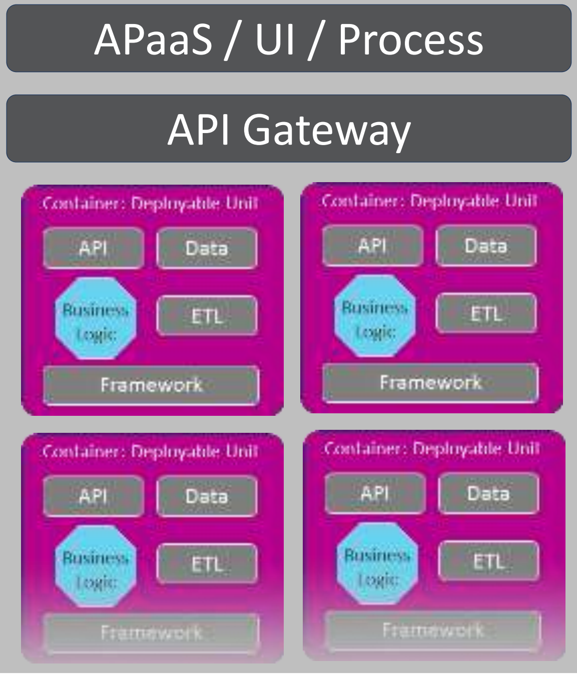
BINARY REPOSITORY

Domain-driven Design Containerization



Framework: Technology stack to support capabilities such as an API Gateway wrapper for discovery, data access (e.g. Hibernate), data persistence (e.g. DB, file, XML), security, ETL

Discoverable Service

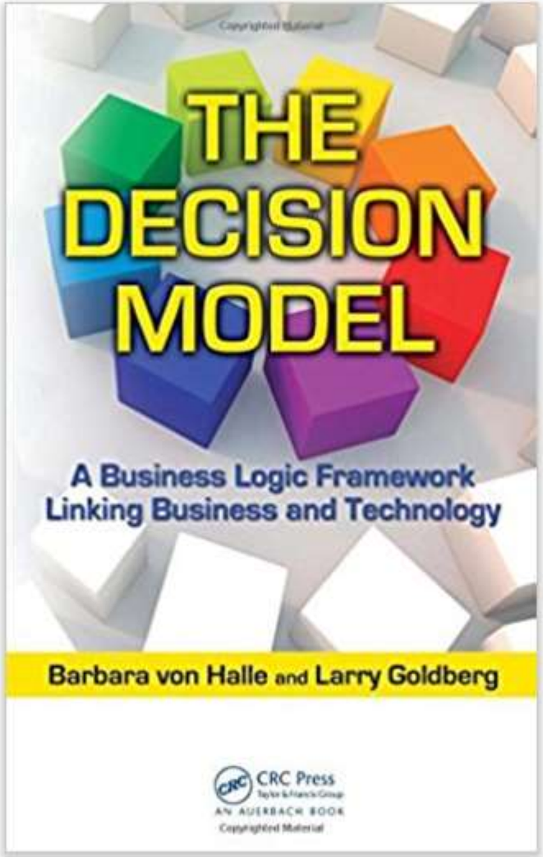


Small, deployable components are assembled to create run time applications accessible through an API Gateway

Reference Architecture Example 5

2. Establish Core Features that Decision Management Provides Across Architectures

- Provide Rigor in Models: *The Model is the Code* and thus creates a link in the technical tool-chain
- The Decision Model (TDM) is a way of representing business logic that is platform and technology independent. It models logic based on the inherent structure of that logic, eliminating style and other subjective preferences, ensuring a consistent and stable representation
- TDM, as implemented in Sapiens DECISION, requires that models follow rigorous requirements that ensure they are executable when valid
- DECISION is a tool that provides the Business with visibility into the Enterprise business logic and treats it as an Asset
- DECISION supports a full lifecycle wherein the business can model, validate, test, deploy, and manage these assets within a governance framework

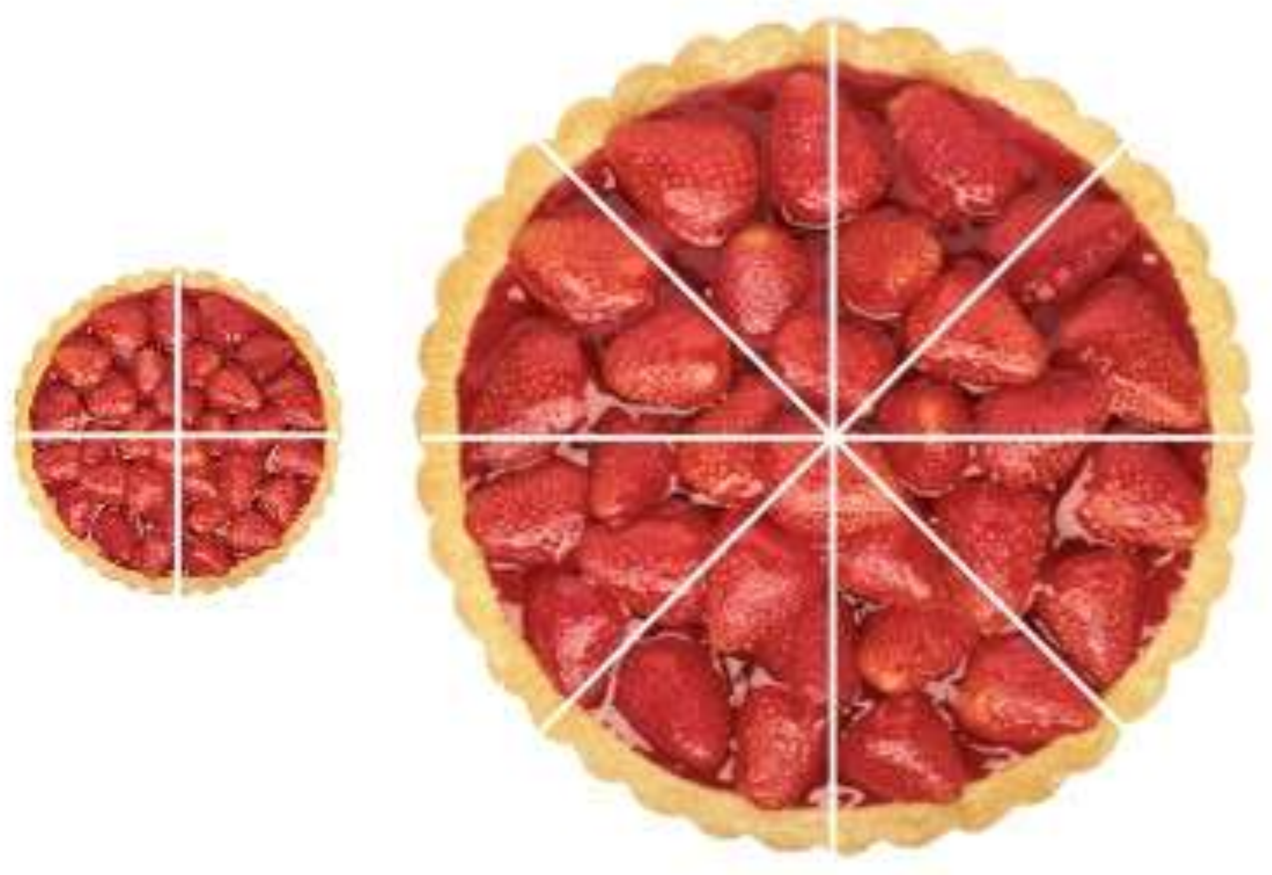

THE DECISION MODEL
A Business Logic Framework Linking Business and Technology
Barbara von Halle and Larry Goldberg
CRC Press
AN AUERBACH BOOK

"The Decision Model: A Business Logic Framework Linking Business and Technology" (von Halle & Goldberg) © 2009

SAPIENS DECISION

Call to Action: DM Architectural Visibility will Create a Bigger Pie

1. Establish a common way to represent business logic across architectures
2. Establish core features that Decision Management provides across architectures through tooling and methodology

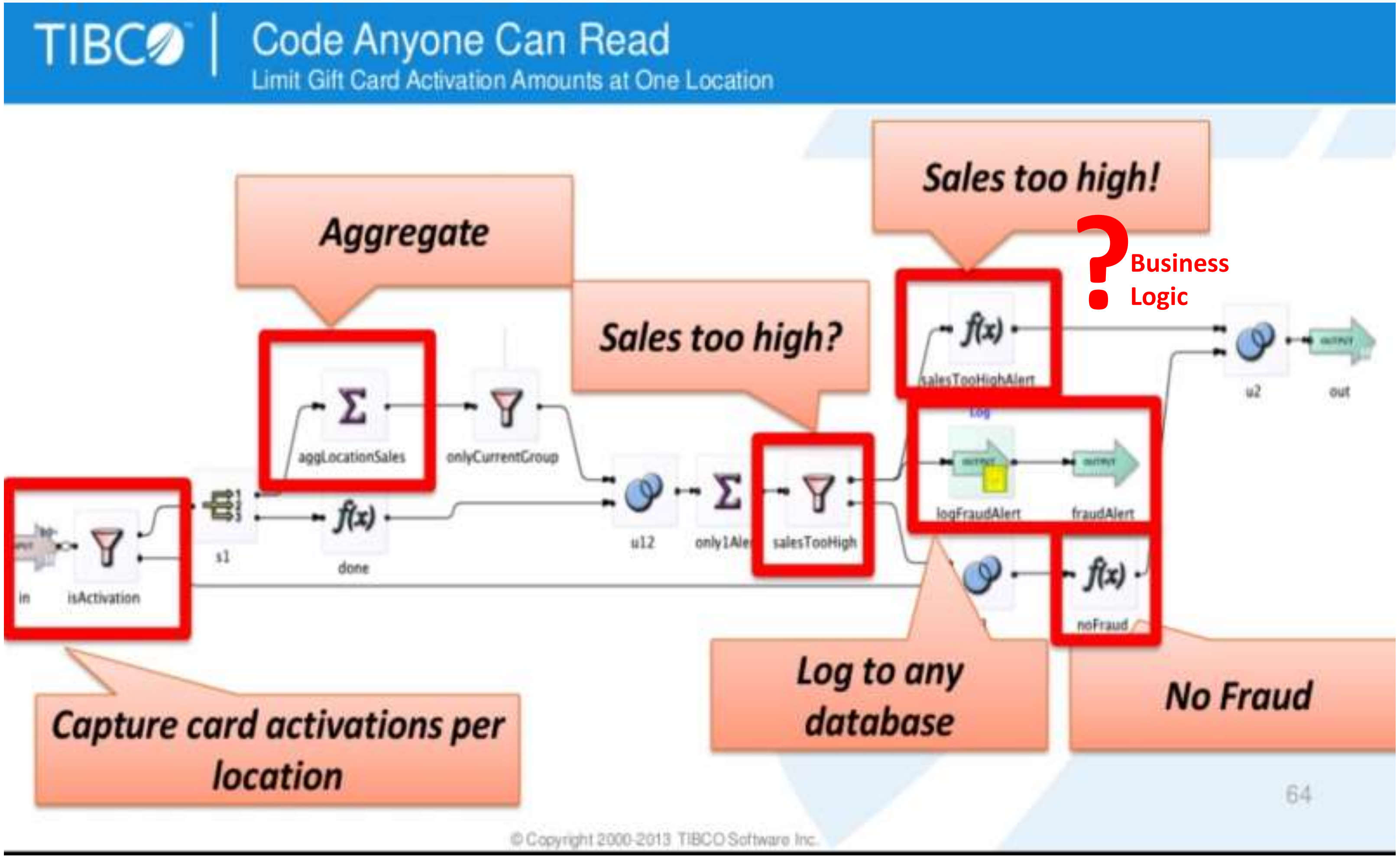


Appendix A: Technical Architecture Sampling

www.sapiensdecision.com

Streaming Analytics: Where is the Business Logic?

- Business logic (and Decisions) are not represented in most architectural diagrams as an identifiable element
- There is no designated layer or component that provides visibility into the business logic that leads to system actions
- Business logic is not treated as an asset that is created, validated, tested, managed, and governed



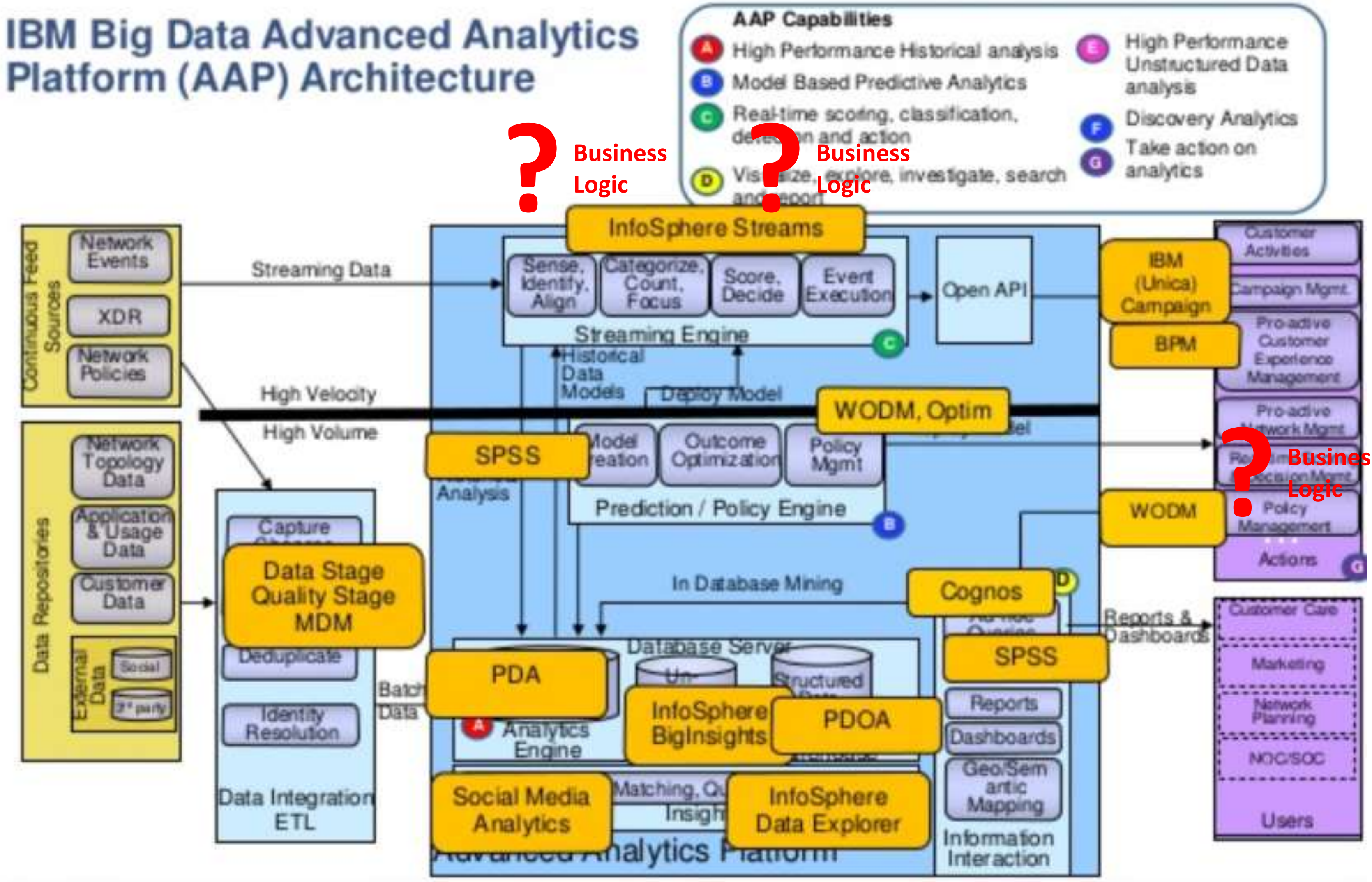
TIBCO | Code Anyone Can Read

Limit Gift Card Activation Amounts at One Location

https://www.slideshare.net/KaiWaehner/streaming-analytics-comparison-of-open-source-frameworks-and-products?qid=c53daefd-23f0-4f3b-b768-59188f776b8e&v=&b=&from_search=4

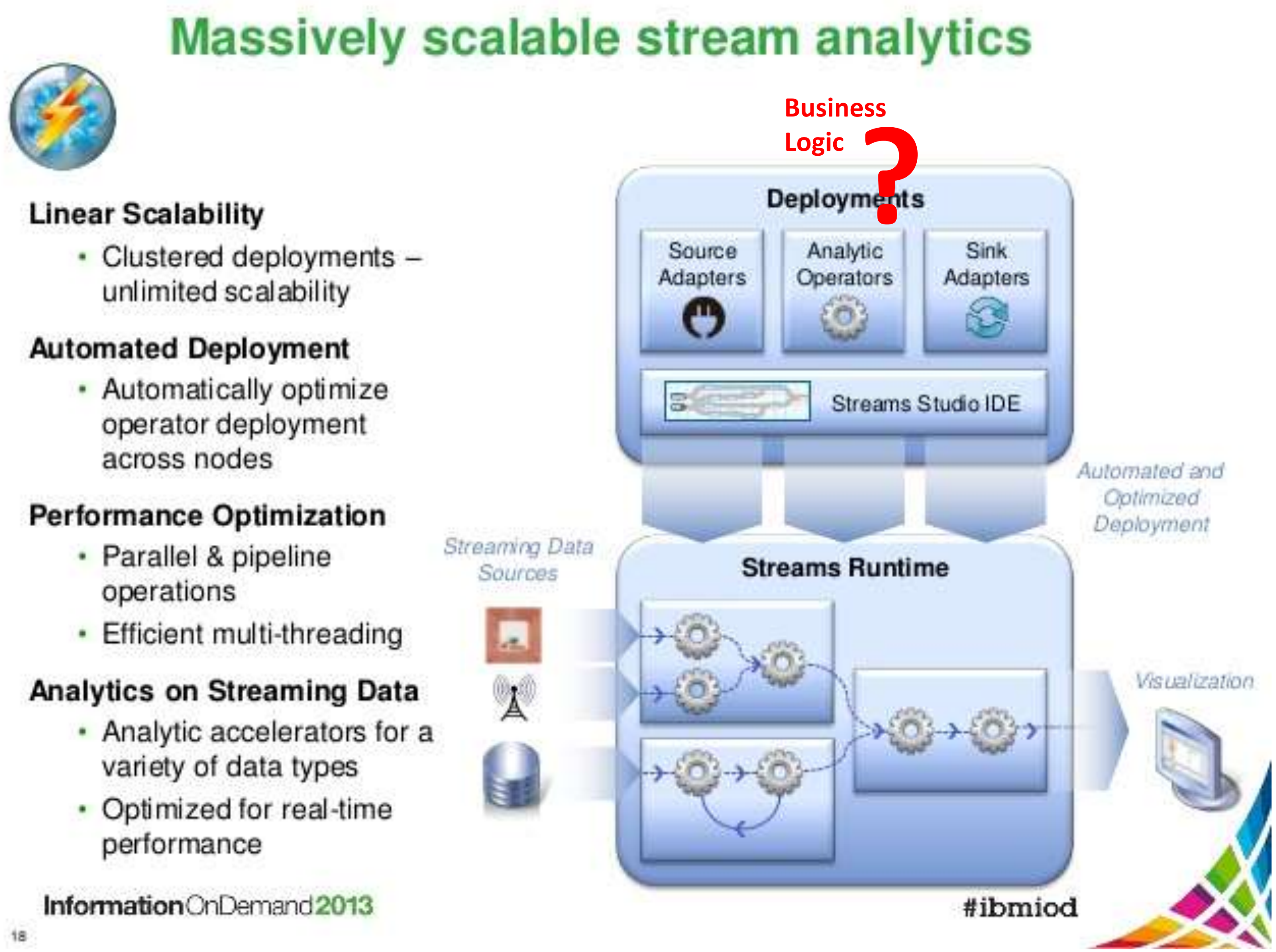
Streaming Analytics: Where is the Business Logic?

- Clue: the term 'Decide'



Streaming Analytics: Where is the Business Logic?

- Clue: the term 'Analytic Operators'



Massively scalable stream analytics

Linear Scalability

- Clustered deployments – unlimited scalability

Automated Deployment

- Automatically optimize operator deployment across nodes

Performance Optimization

- Parallel & pipeline operations
- Efficient multi-threading

Analytics on Streaming Data

- Analytic accelerators for a variety of data types
- Optimized for real-time performance

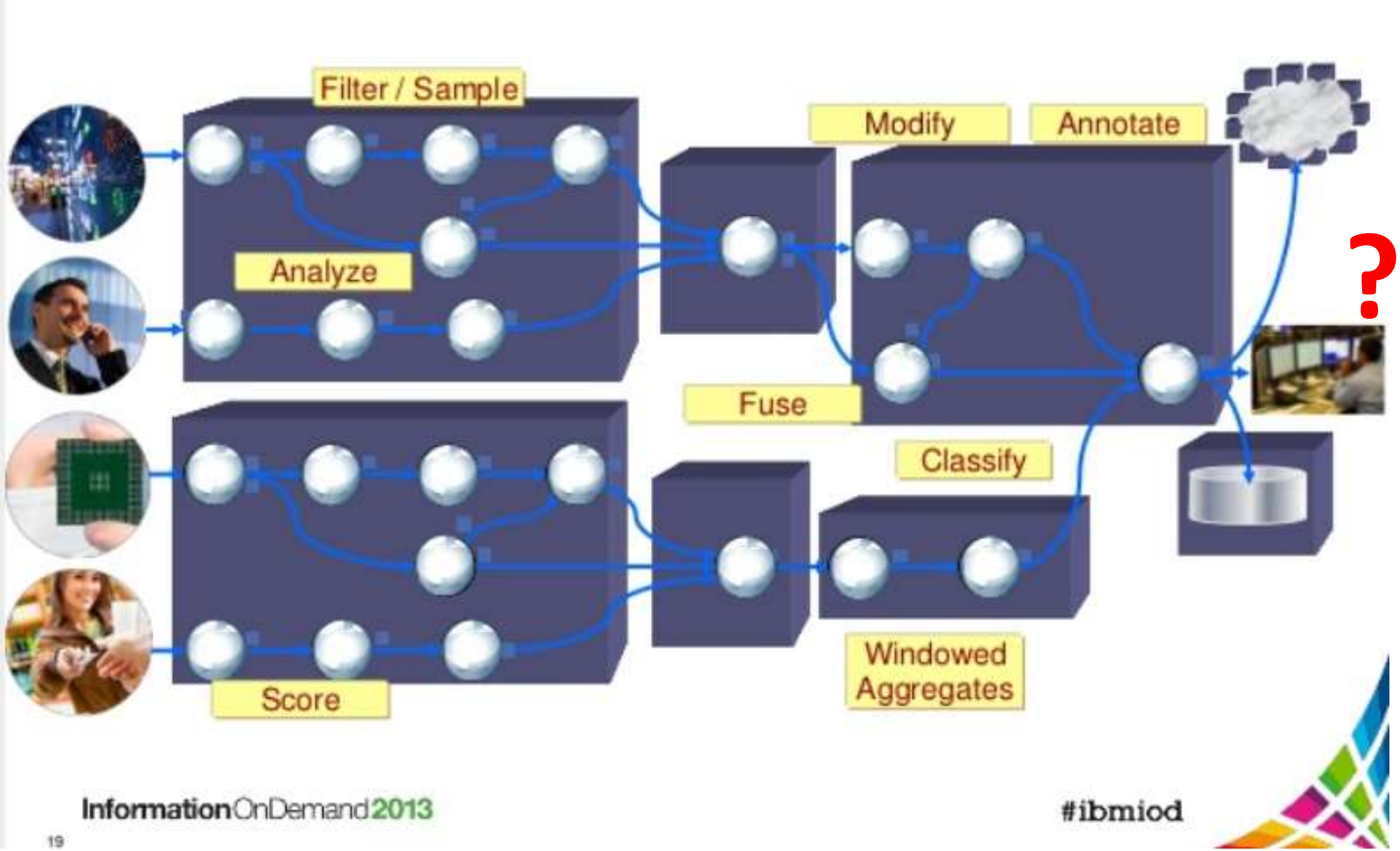
InformationOnDemand2013

18

#ibmiod

Streaming Analytics: Where is the Business Logic?

- Clue: A person looking pensive

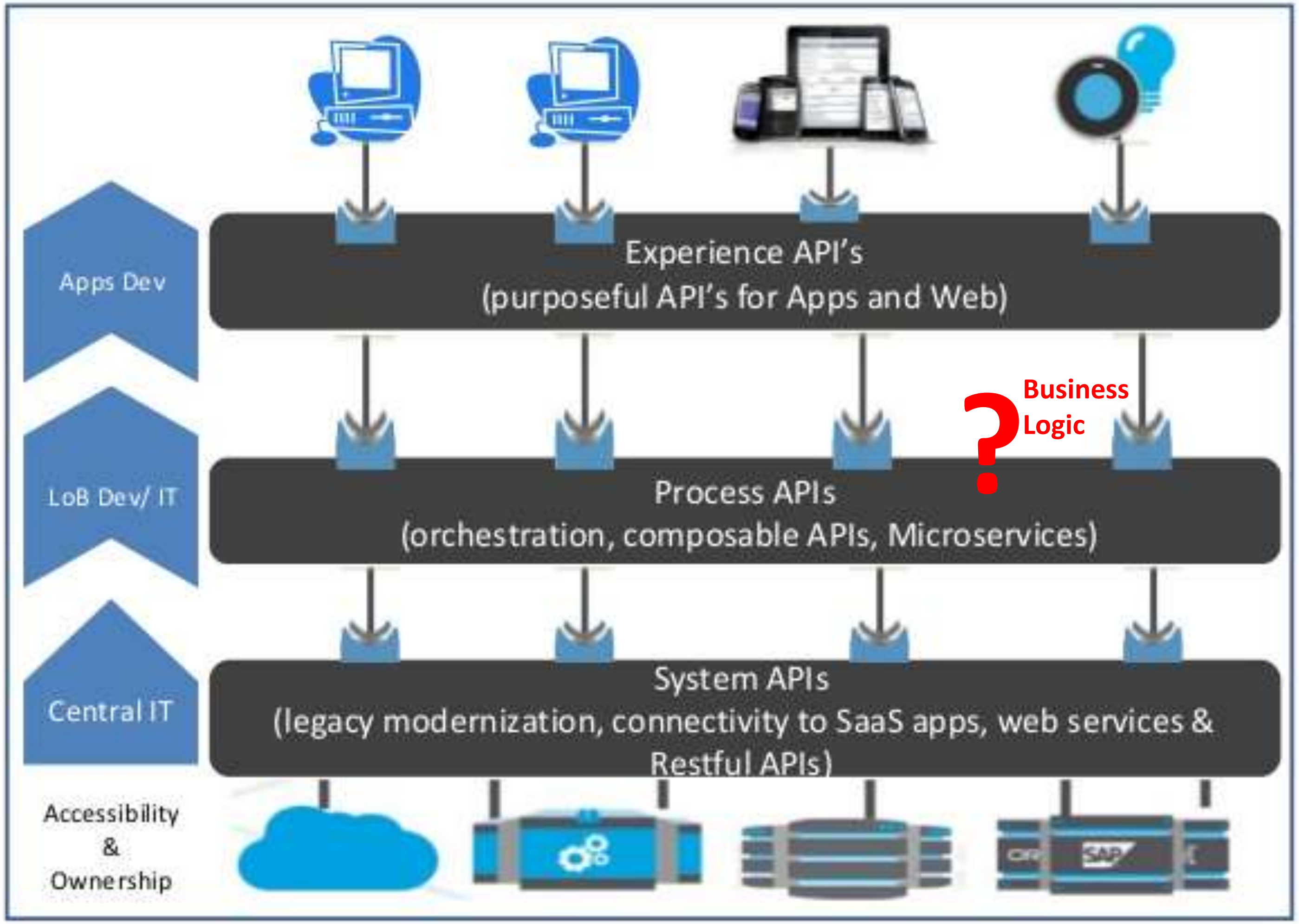


? Business Logic

API Economy (MuleSoft): Where is the Business Logic?

- Clue: ‘Service’ in layer referencing the actor ‘LoB Dev’

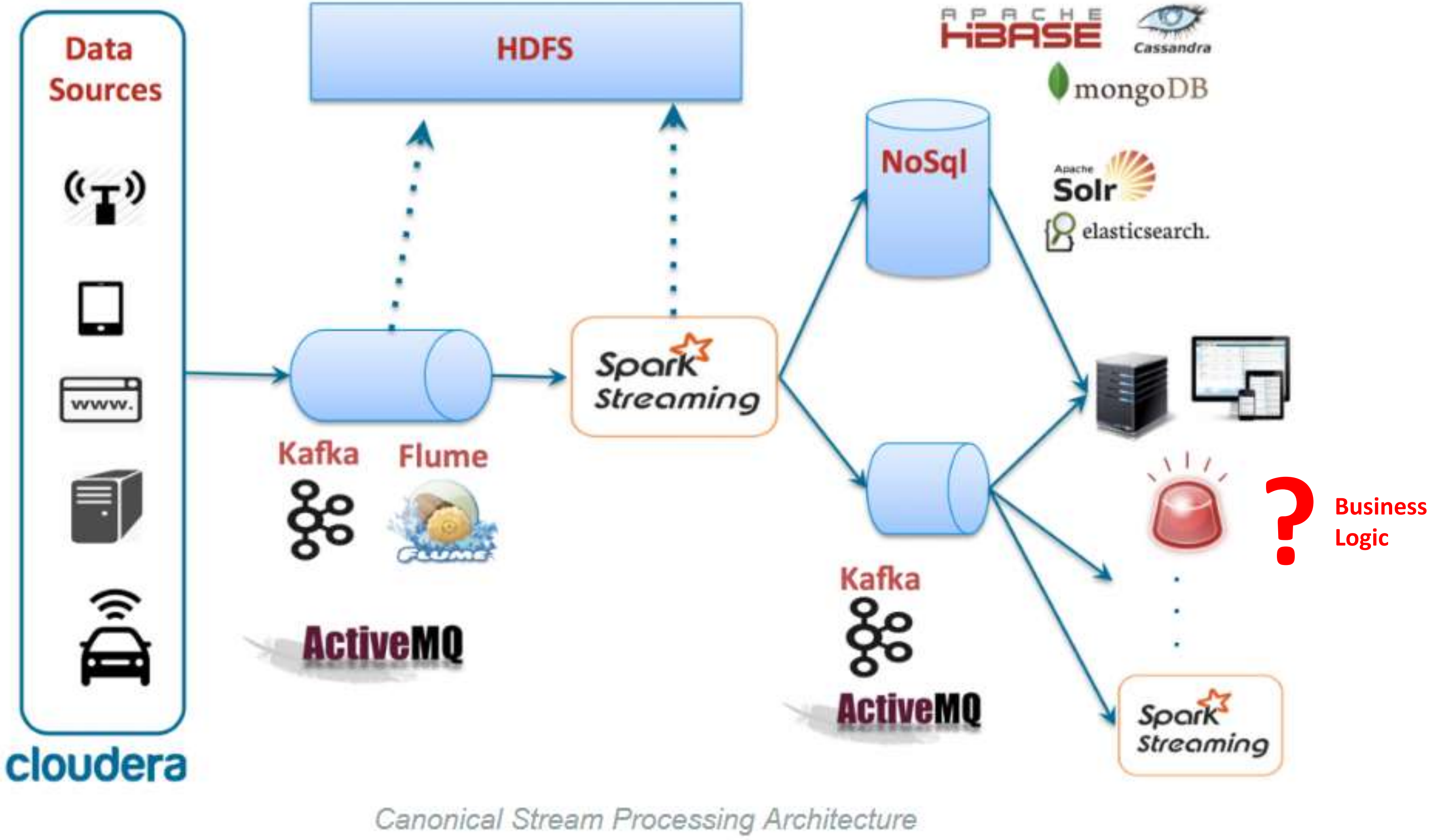
Three Tier API Layer Architecture



Apps Dev
Experience API’s
(purposeful API’s for Apps and Web)
Business
Logic
?
LoB Dev/ IT
Process APIs
(orchestration, composable APIs, Microservices)
Central IT
System APIs
(legacy modernization, connectivity to SaaS apps, web services & Restful APIs)
Accessibility
&
Ownership
SAP

Streaming Analytics: Where is the Business Logic?

- Clue: Red alarm

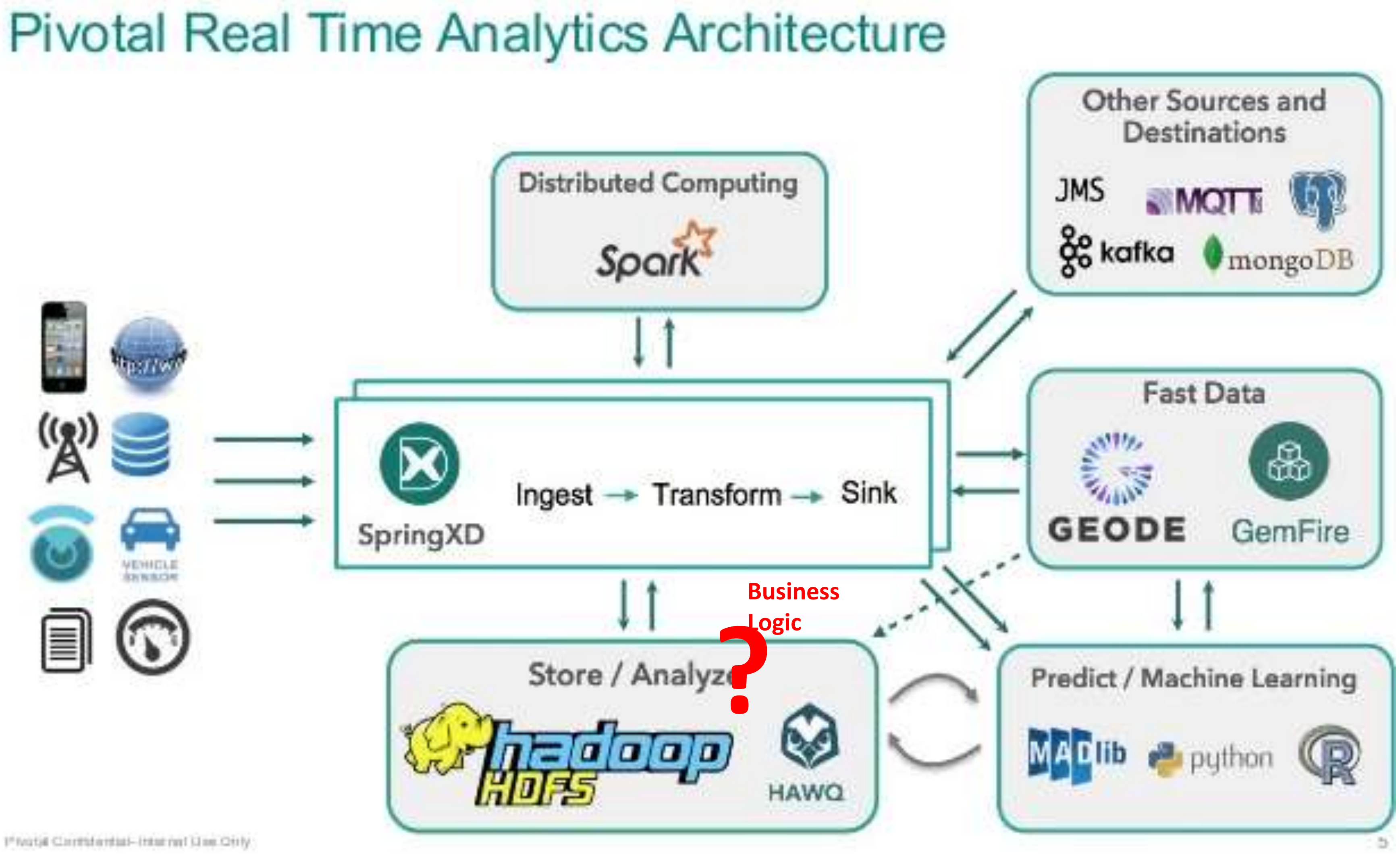


Canonical Stream Processing Architecture

http://ingest.tips/2015/06/24/real-time-analytics-with-kafka-and-spark-streaming/

Streaming Analytics: Where is the Business Logic?

- Clue: The term 'Analyze'

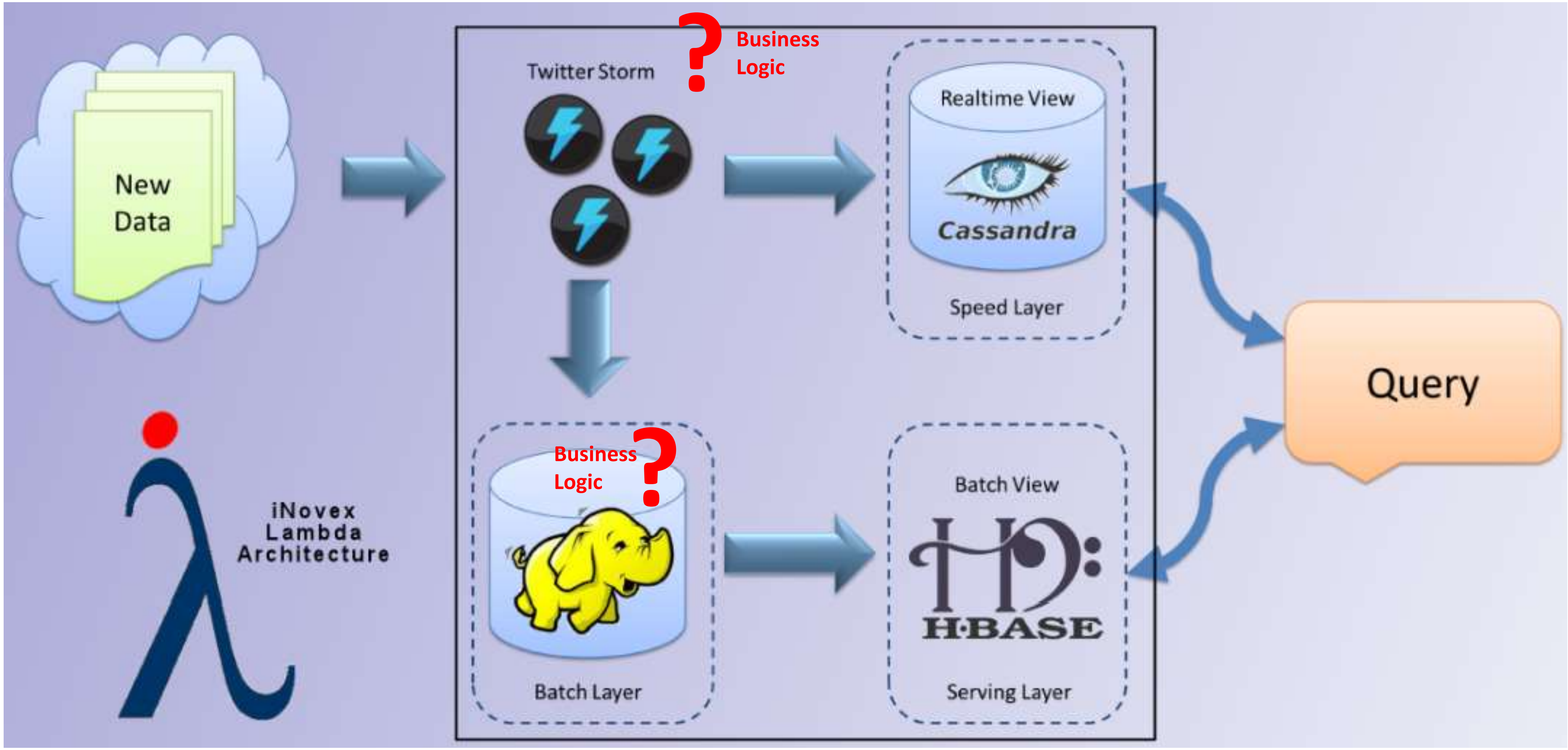


Pivotal Real Time Analytics Architecture
Other Sources and Destinations
JMS
MQTT
kafka
mongoDB
Distributed Computing
Spark
SpringXD
Ingest → Transform → Sink
Fast Data
GEODE
GemFire
Business Logic
?
Store / Analyze
hadoop HDFS
HAWQ
Predict / Machine Learning
MADlib
python
R
VEHICLE SENSOR
Pivotal Confidential–Internal Use Only

https://www.slideshare.net/kgshukla/pivotal-48330128

Streaming Analytics: Where is the Business Logic?

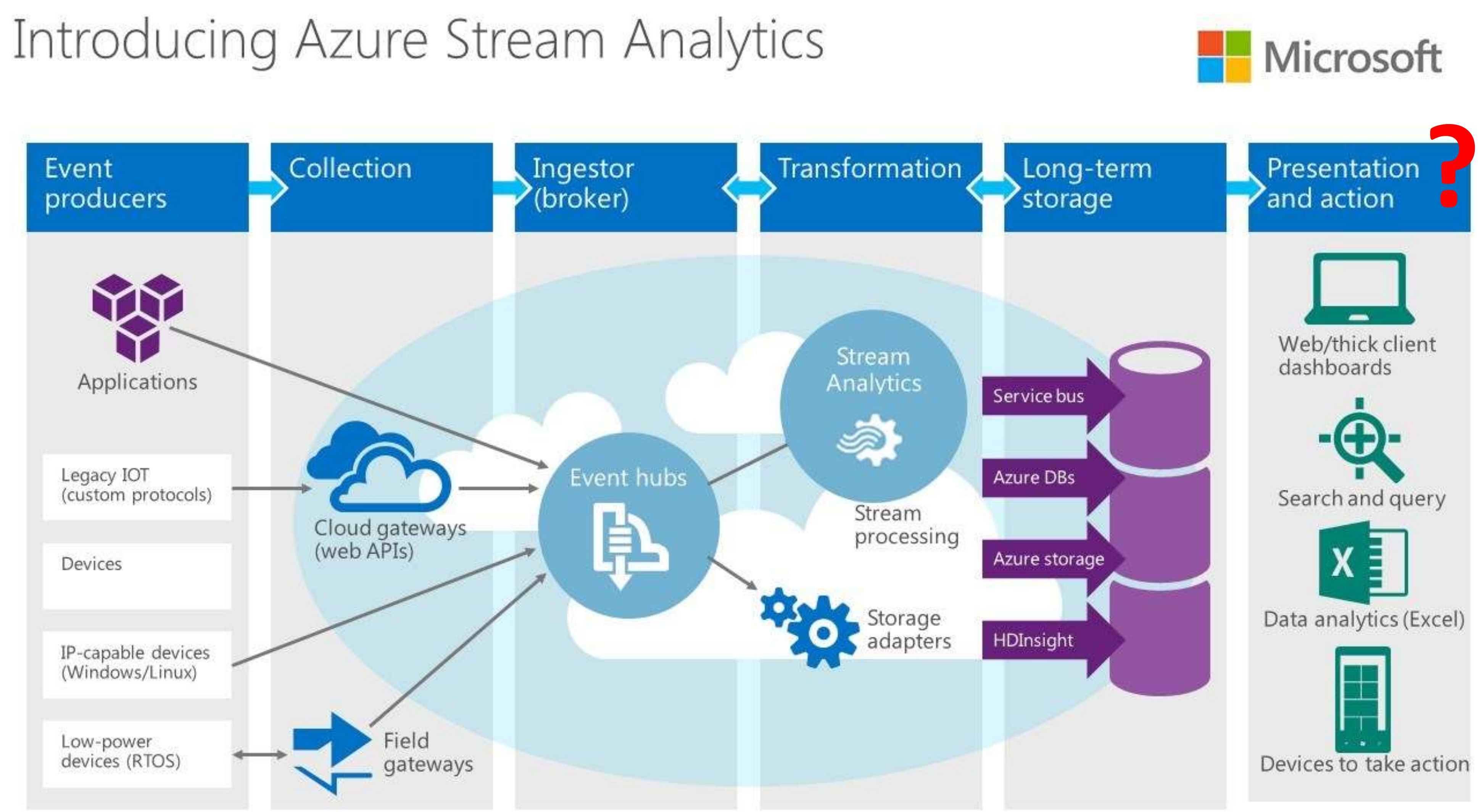
- Clues: Bolts, batch



https://www.inovexcorp.com/jim/index.php/ir-d-case-studies/streaming-cloud-analytics

Streaming Analytics: Where is the Business Logic?

- Clue: The term ‘Action’

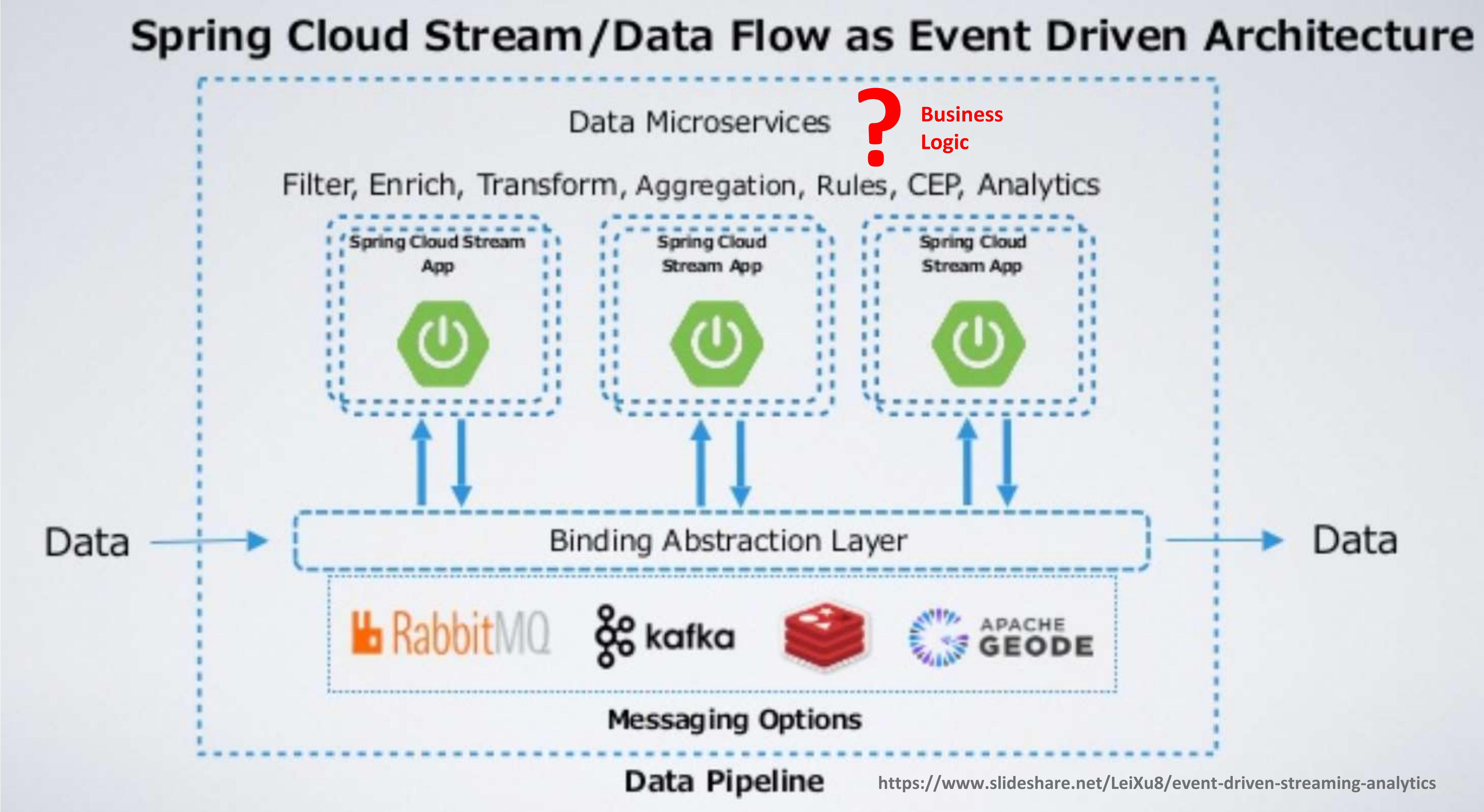


? Business Logic

http://www.jamesserra.com/archive/2015/06/what-is-microsoft-azure-stream-analytics/

Streaming Analytics: Where is the Business Logic?

- Clue: The term ‘Rules’

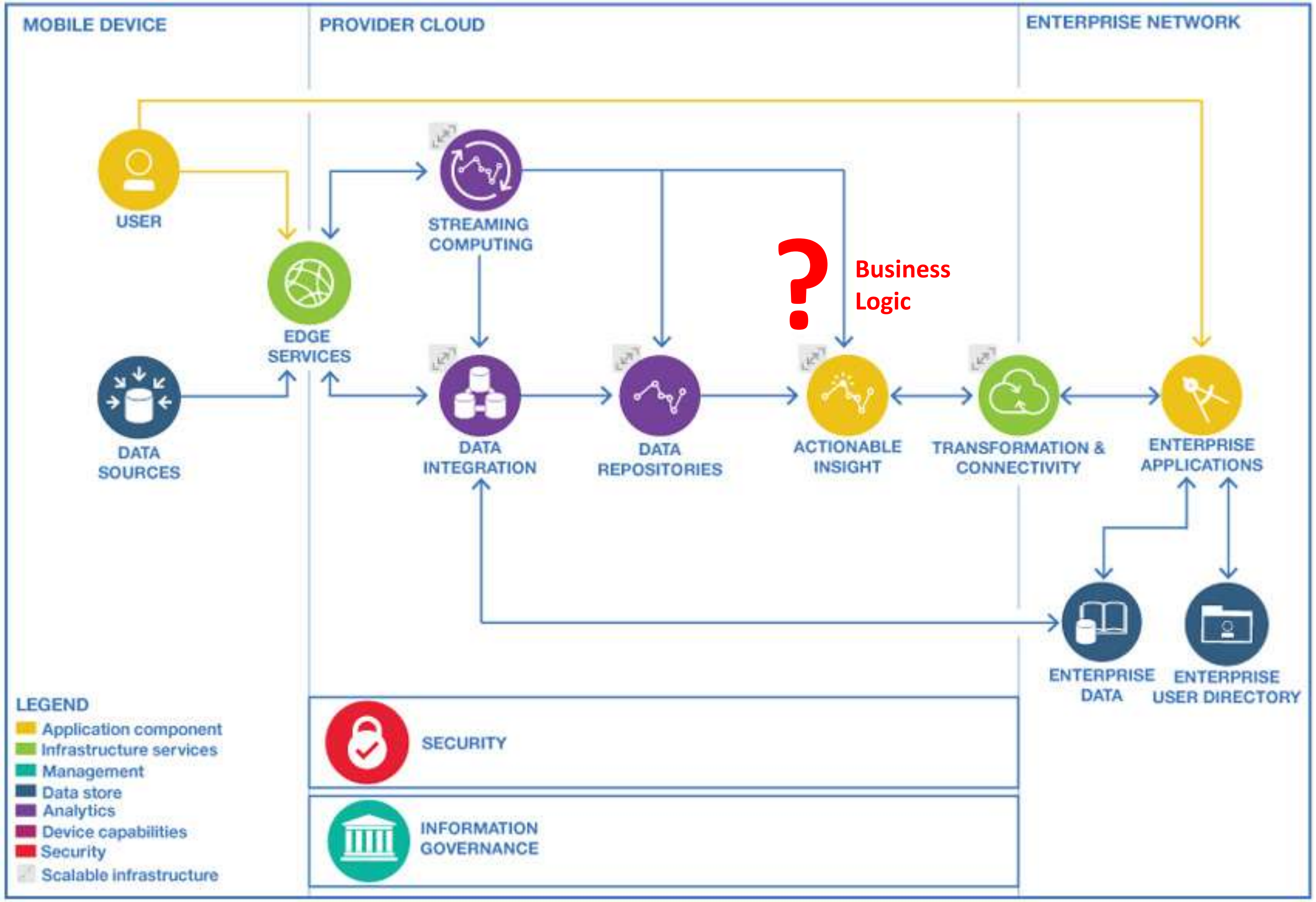


Spring Cloud Stream/Data Flow as Event Driven Architecture

https://www.slideshare.net/LeiXu8/event-driven-streaming-analytics

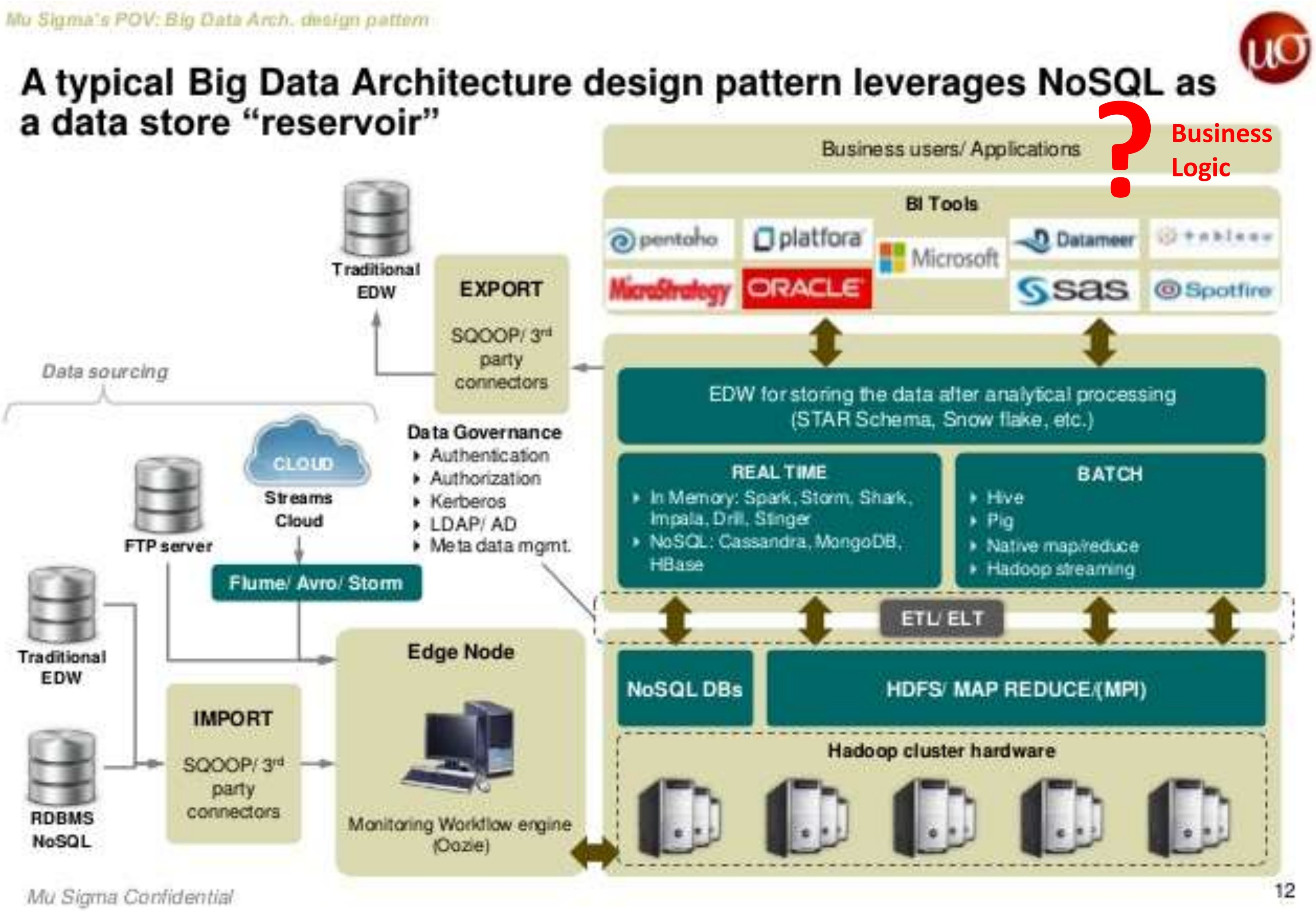
Streaming Analytics: Where is the Business Logic?

- Clue: The term 'Actionable Insight'



Big Data: Where is the Business Logic?

- Clue: The term ‘Business Users/Applications’



Mu Sigma's POV: Big Data Arch. design pattern

A typical Big Data Architecture design pattern leverages NoSQL as a data store “reservoir”

? Business Logic

Business users/ Applications

BI Tools

pentaho

platfora

Microsoft

Datameer

tableau

MicroStrategy

ORACLE

sas

Spotfire

Traditional EDW

EXPORT

SQOOP/ 3rd party connectors

EDW for storing the data after analytical processing (STAR Schema, Snow flake, etc.)

REAL TIME

- In Memory: Spark, Storm, Shark, Impala, Drill, Stinger
- NoSQL: Cassandra, MongoDB, HBase

BATCH

- Hive
- Pig
- Native map/reduce
- Hadoop streaming

Data sourcing

CLOUD

Streams Cloud

FTP server

Data Governance

- Authentication
- Authorization
- Kerberos
- LDAP/ AD
- Meta data mgmt.

Flume/ Avro/ Storm

ETL/ ELT

Traditional EDW

Edge Node

NoSQL DBs

HDFS/ MAP REDUCE (MPI)

IMPORT

SQOOP/ 3rd party connectors

RDBMS NoSQL

Monitoring Workflow engine (Oozie)

Hadoop cluster hardware

Mu Sigma Confidential

12

Appendix B: Decision Management Common Features

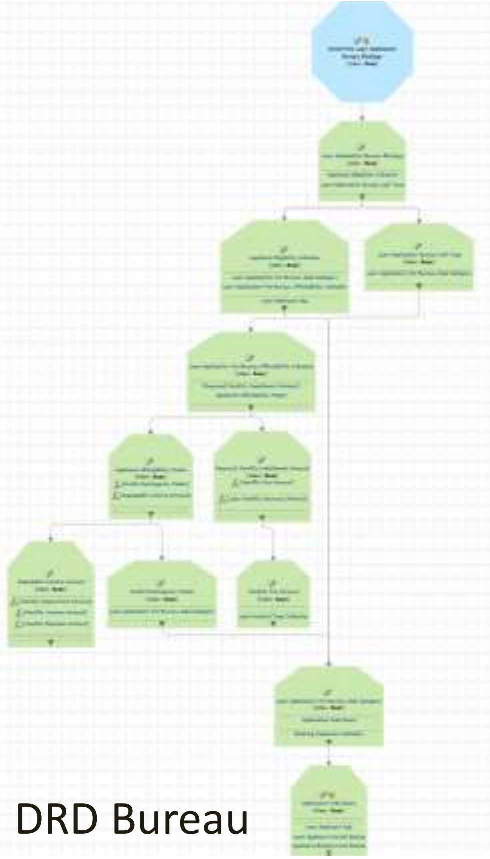
SAPIENS DECISION

www.sapiensdecision.com

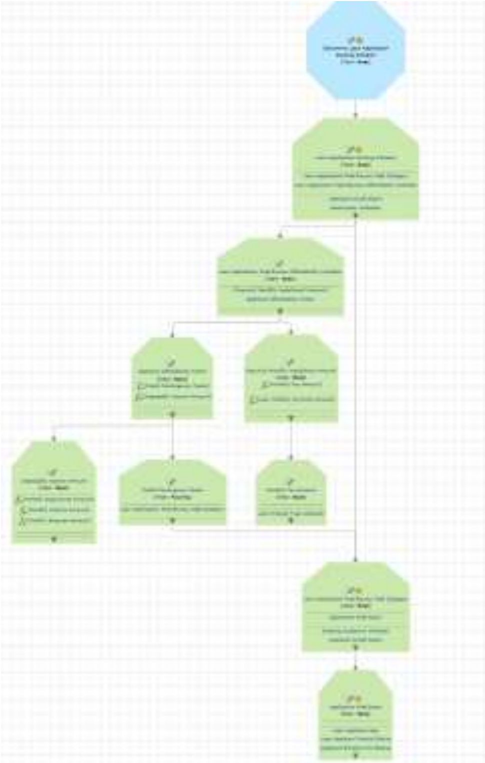
2.1 Decision Services the Business wants to Maintain

Can a loan be approved without accessing a credit bureau for a credit score? A Decision Service is called to find out

Can a loan be auto-approved without human intervention? A Decision Service is called to find out



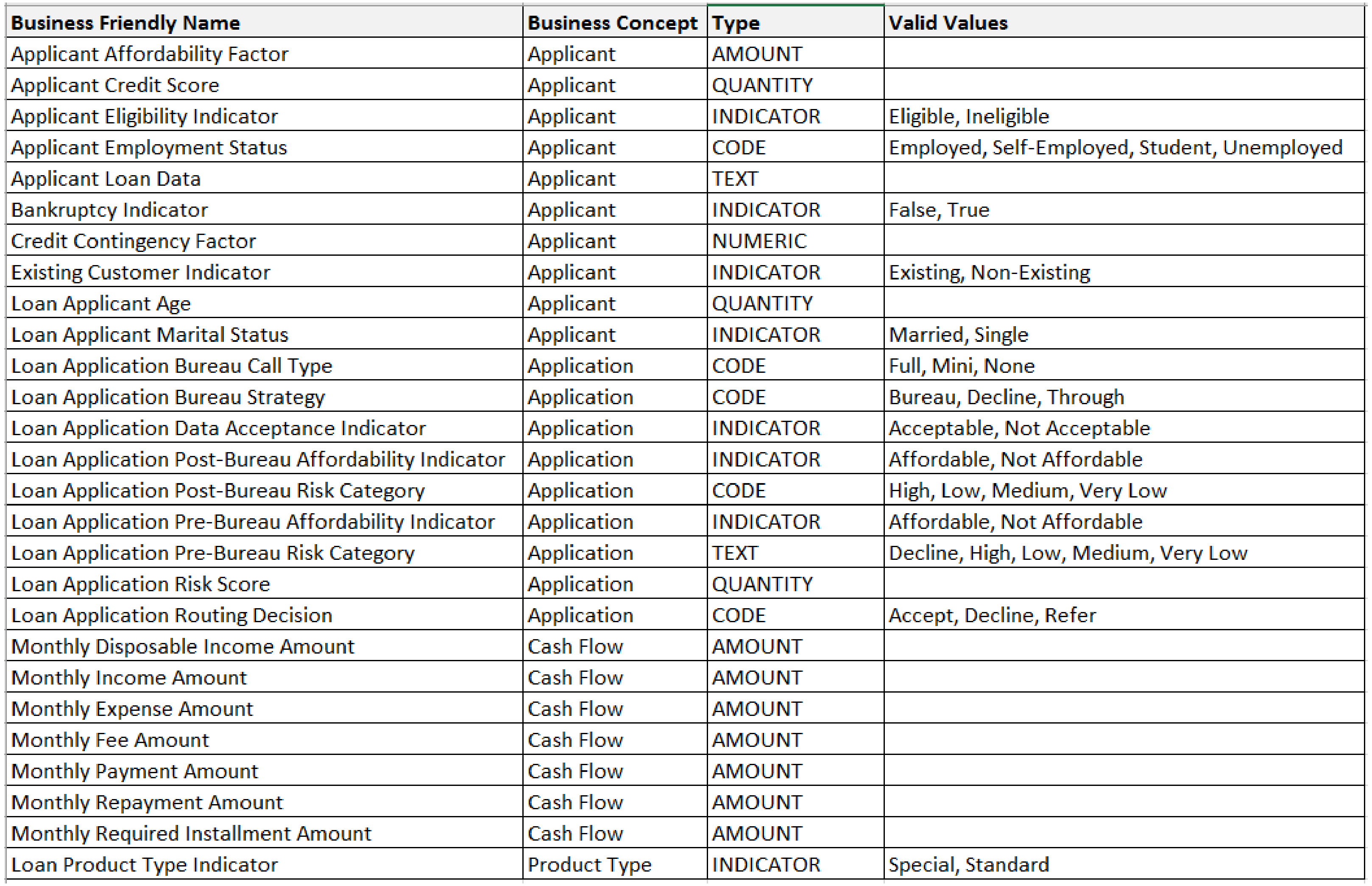
DRD Bureau



DRD Routing

2.2 Define Business Glossary Capabilities

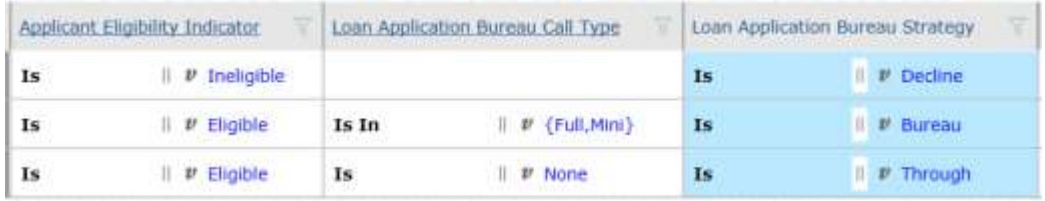
- The glossary contains an entry and meta data about every attribute in a community of decision models. This ensures **consistency** of terminology across decision models and **maximum re-use** of business assets
- There may be multiple glossaries in an enterprise and they are arranged hierarchically so as to afford re-use but also distinctions between the terms used by different organizations within the enterprise
- Glossaries capture the **business-friendly** attribute names, business-friendly attribute types, and business-friendly, not technical, allowed values for these attributes. Technical resources may map the business-friendly names and values to one or more technical models
- The attributes in a particular glossary are organized by Business Concepts such as Applicant, or Application



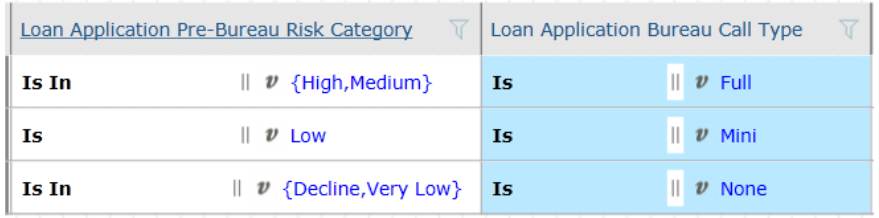
Business Friendly Name	Business Concept	Type	Valid Values
Applicant Affordability Factor	Applicant	AMOUNT	
Applicant Credit Score	Applicant	QUANTITY	
Applicant Eligibility Indicator	Applicant	INDICATOR	Eligible, Ineligible
Applicant Employment Status	Applicant	CODE	Employed, Self-Employed, Student, Unemployed
Applicant Loan Data	Applicant	TEXT	
Bankruptcy Indicator	Applicant	INDICATOR	False, True
Credit Contingency Factor	Applicant	NUMERIC	
Existing Customer Indicator	Applicant	INDICATOR	Existing, Non-Existing
Loan Applicant Age	Applicant	QUANTITY	
Loan Applicant Marital Status	Applicant	INDICATOR	Married, Single
Loan Application Bureau Call Type	Application	CODE	Full, Mini, None
Loan Application Bureau Strategy	Application	CODE	Bureau, Decline, Through
Loan Application Data Acceptance Indicator	Application	INDICATOR	Acceptable, Not Acceptable
Loan Application Post-Bureau Affordability Indicator	Application	INDICATOR	Affordable, Not Affordable
Loan Application Post-Bureau Risk Category	Application	CODE	High, Low, Medium, Very Low
Loan Application Pre-Bureau Affordability Indicator	Application	INDICATOR	Affordable, Not Affordable
Loan Application Pre-Bureau Risk Category	Application	TEXT	Decline, High, Low, Medium, Very Low
Loan Application Risk Score	Application	QUANTITY	
Loan Application Routing Decision	Application	CODE	Accept, Decline, Refer
Monthly Disposable Income Amount	Cash Flow	AMOUNT	
Monthly Income Amount	Cash Flow	AMOUNT	
Monthly Expense Amount	Cash Flow	AMOUNT	
Monthly Fee Amount	Cash Flow	AMOUNT	
Monthly Payment Amount	Cash Flow	AMOUNT	
Monthly Repayment Amount	Cash Flow	AMOUNT	
Monthly Required Installment Amount	Cash Flow	AMOUNT	
Loan Product Type Indicator	Product Type	INDICATOR	Special, Standard

2.3 Decision Logic Validations Support Executability

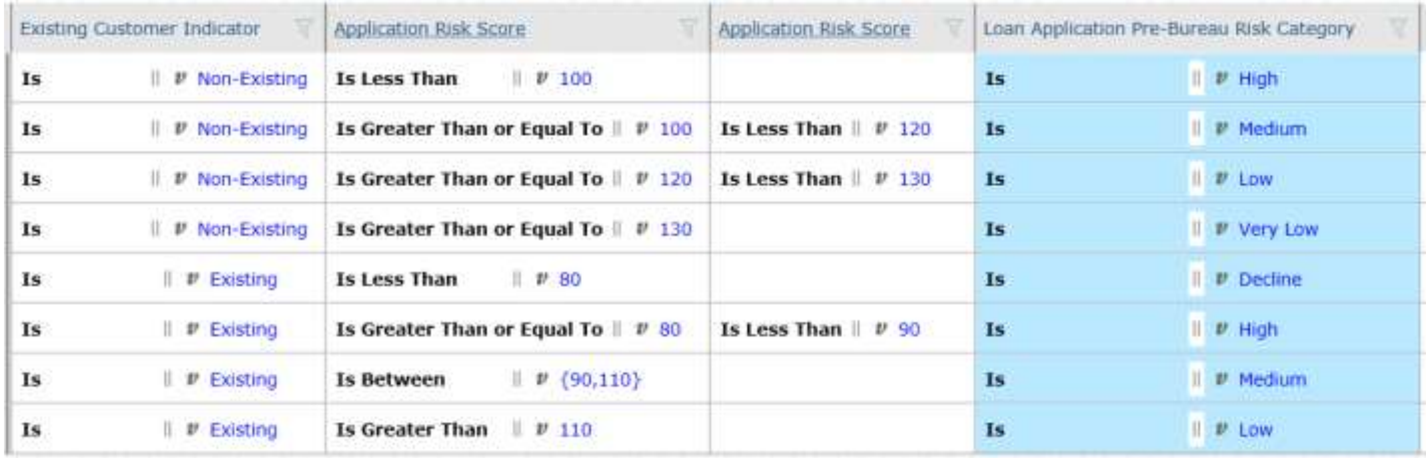
- Each Decision is modeled as a Decision Table where the logic adheres to the **15 principles of TDM** (see ***The Decision Model: A Business Logic Framework Linking Business and Technology***)
- The DECISION Manager module used for modeling as part of the Sapiens DECISION product suite validates that the logic complies with the principles to ensure that it is executable and can be used to generate code



Applicant Eligibility Indicator	Loan Application Bureau Call Type	Loan Application Bureau Strategy
Is Ineligible		Is Decline
Is Eligible	Is In {Full,Mini}	Is Bureau
Is Eligible	Is None	Is Through



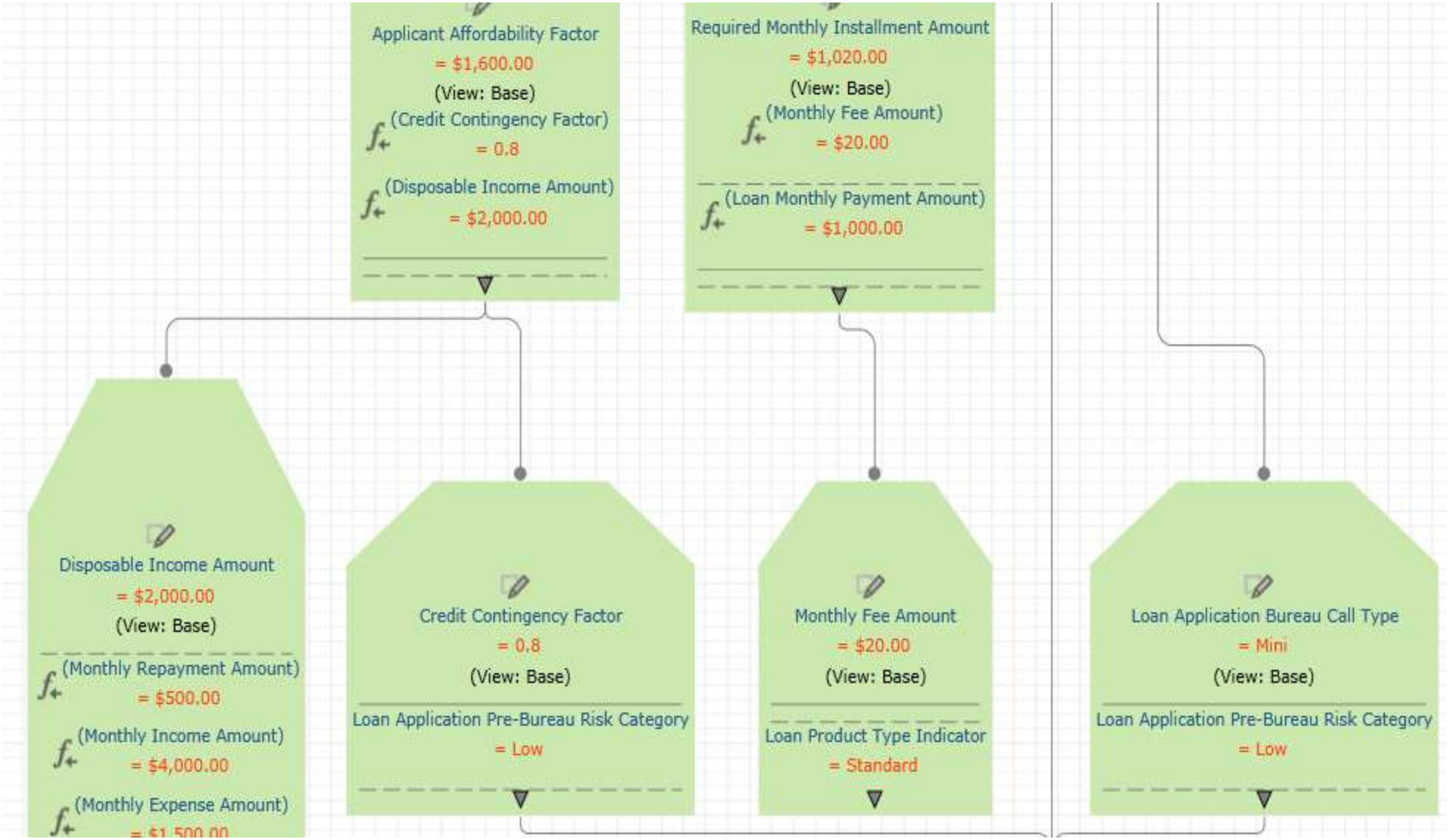
Loan Application Pre-Bureau Risk Category	Loan Application Bureau Call Type
Is In {High,Medium}	Is Full
Is Low	Is Mini
Is In {Decline,Very Low}	Is None



Existing Customer Indicator	Application Risk Score	Application Risk Score	Loan Application Pre-Bureau Risk Category
Is Non-Existing	Is Less Than 100		Is High
Is Non-Existing	Is Greater Than or Equal To 100	Is Less Than 120	Is Medium
Is Non-Existing	Is Greater Than or Equal To 120	Is Less Than 130	Is Low
Is Non-Existing	Is Greater Than or Equal To 130		Is Very Low
Is Existing	Is Less Than 80		Is Decline
Is Existing	Is Greater Than or Equal To 80	Is Less Than 90	Is High
Is Existing	Is Between (90,110}		Is Medium
Is Existing	Is Greater Than 110		Is Low

2.4 Testing Enhances Business Control

- **Testing by the business modeler is a critical element** of the model development process
- Testing is performed at the decision table level, for each decision table. Typically, decision table testing is comprehensive for all possible attribute values
- Next, testing is performed at the decision service level. For complex services, branch testing may also occur
- Finally, process-level testing is performed
- Test suites are **typically exported and provided to IT** for acceptance and regression needs



Applicant Affordability Factor
= $1,600.00
(View: Base)
(Credit Contingency Factor)
= 0.8
(Disposable Income Amount)
= $2,000.00

Required Monthly Installment Amount
= $1,020.00
(View: Base)
(Monthly Fee Amount)
= $20.00
(Loan Monthly Payment Amount)
= $1,000.00

Disposable Income Amount
= $2,000.00
(View: Base)
(Monthly Repayment Amount)
= $500.00
(Monthly Income Amount)
= $4,000.00
(Monthly Expense Amount)
= $1,500.00

Credit Contingency Factor
= 0.8
(View: Base)
Loan Application Pre-Bureau Risk Category
= Low

Monthly Fee Amount
= $20.00
(View: Base)
Loan Product Type Indicator
= Standard

Loan Application Bureau Call Type
= Mini
(View: Base)
Loan Application Pre-Bureau Risk Category
= Low

Appendix C: Emergent Architecture Detail

www.sapiensdecision.com

Decision Management Emerging Usage: Big Data

- “extremely large data sets that may be analyzed computationally to reveal patterns, trends, and associations, especially relating to human behavior and interactions” (Forester)
- Structured or unstructured data
- Early on, typically processed in batch
- More and more, scale of execution throughput requirements exceeds capacity of traditional decision architectures – business logic must be brought to the data and not the data to the decisions
- Examples:
  - Securities fraud
  - Financial system stress testing/Value at risk (VaR)
  - Margin call collateral and position evaluation
  - Medical assessment – a single full human genome 600GB

Decision Management Emerging Usage: Streaming Analytics

- Real-time action on perishable insights via processing of incoming data streams
- “Streaming analytics filter, aggregate, enrich, and analyze a high throughput of data from disparate live data sources to identify patterns, detect urgent situations, and automate immediate actions in real-time” (Forrester)
- Examples:
  - Security breach detection
  - Network outages detection
  - Sports betting – 80% of sports betting is done after game begins[1]
  - Transportation driver monitoring
  - Retail stock outs
  - Retail pricing opportunities
  - Telecom bandwidth monitoring
  - Medical assessment – sensor data 10GB raw data per second

Decision Management Emerging Usage: API Economy

- “API Economy (Application Programming Interface Economy) is a general term that describes the way APIs can positively affect an organization's profitability”
- Discoverable decision APIs enable users to learn about available decisions and how they are used
- While execution can be anywhere, integration processes are centralized
- Boundaries of usage are not prescribed beforehand and are opened to 3rd-parties to creatively integrate
- Predictions: 50% of business-to-business collaboration will take place through Web APIs by 2017, and by next year 75% of Fortune 1000 firms will offer public Web APIs (Gartner)
- Examples:
  - 3rd-party insurance calculators used by multiple insurers
  - Bank app APIs are used by partners to sign up for services and access information
  - CD rates or other product configurations can be made visible to comparison sites

Thank You

Please visit us at www.sapiensdecision.com

SAPIENS DECISION

www.sapiensdecision.com