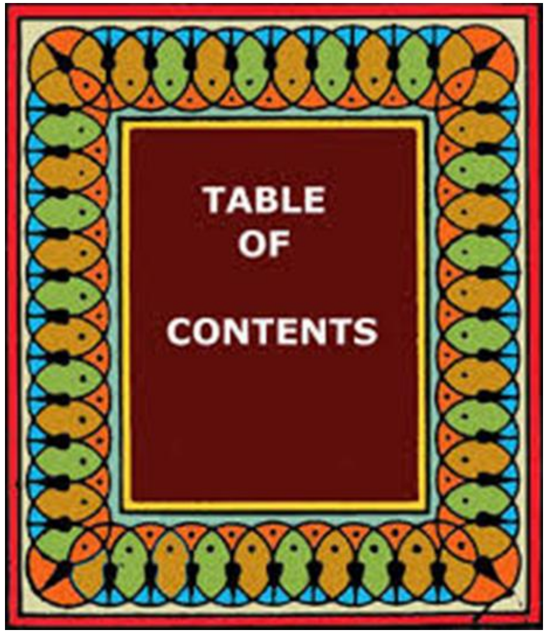
Unlocking Insights for Health and Wellness: Exploring the Data Curation Framework for Big Data

AIRA: Seminar Presentation

15/06/2023

Usman Akhtar

Agenda



About Me

❑ Myself

Now: Post. Doctoral Researcher (Adjunct Prof.), Jagiellonian University, Faculty of Math. & Computer Science, Poland 2022 ~ -

Past: Assistant professor, Riphah International University, Pakistan 2021~2022

Past: Research Scientist, Halmstad University, Sweden 2018

Past: Ph.D. Student, KyungHee University South Korea, 2015 ~ 2021

❑ Outside Academia (past)

Software Developer, Plumgrid Inc. cloud infrastructure solution (Acquired by VMware), Islamabad, Pakistan

❑ My area

Specializing in the areas of Big data & Distributed Systems, Cloud Computing, Semantic Web and Computational Complexity of Ride-Pooling

Brief History of the Projects (UCLab, Korea)

uLCRC
동서신의학 u-라이프케어 연구센터
ubiquitous Life Care Research Center

CDSS (Clinical Decision Support System)

nipa 정보통신산업진흥원 National IT Industry Promotion Agency

CDSS Platform

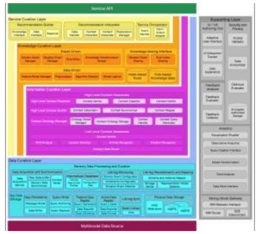
Smart CDSS Clinical Decision Support System

(2006.11~2014.12)
USD 800 thousand/year

MINING MINDS

Mining Minds Healthcare Platform

산업통상자원부 MINISTRY OF TRADE, INDUSTRY & ENERGY MOTIE



(2014.06~2018.05)
USD 1million/year

Intelligent Medical Platform

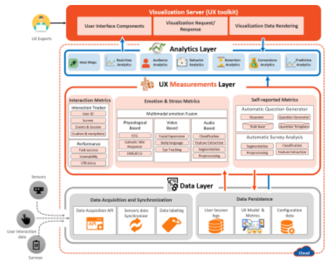
IITP 정보통신기술진흥센터

(2017.05~2022.12)
USD 800 thousand/year

Lean UX Platform

Lean User Experience Platform

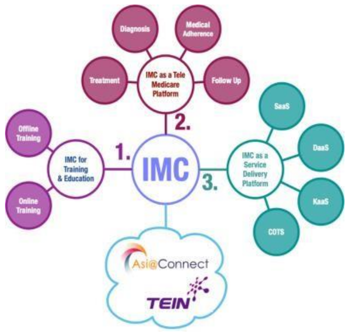
IITP 정보통신기술진흥센터



(2017.06~2024.12)
USD300 thousand/year

iMS TEIN

Intelligent Medical Services over TEIN



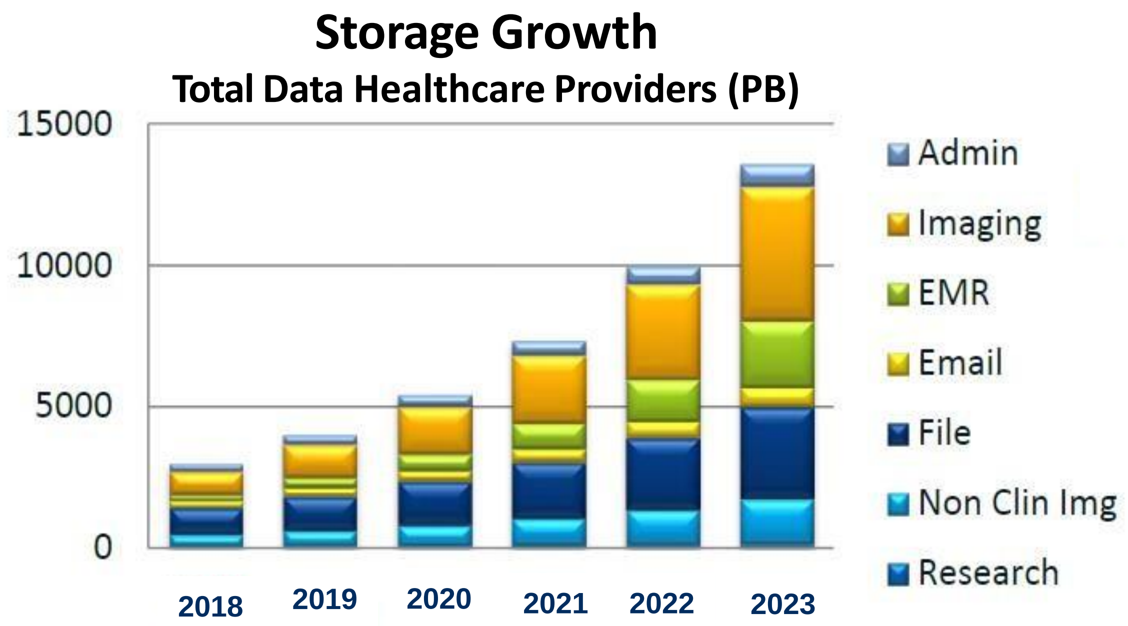
(2019.05~2020.12)
225 thousand EURO

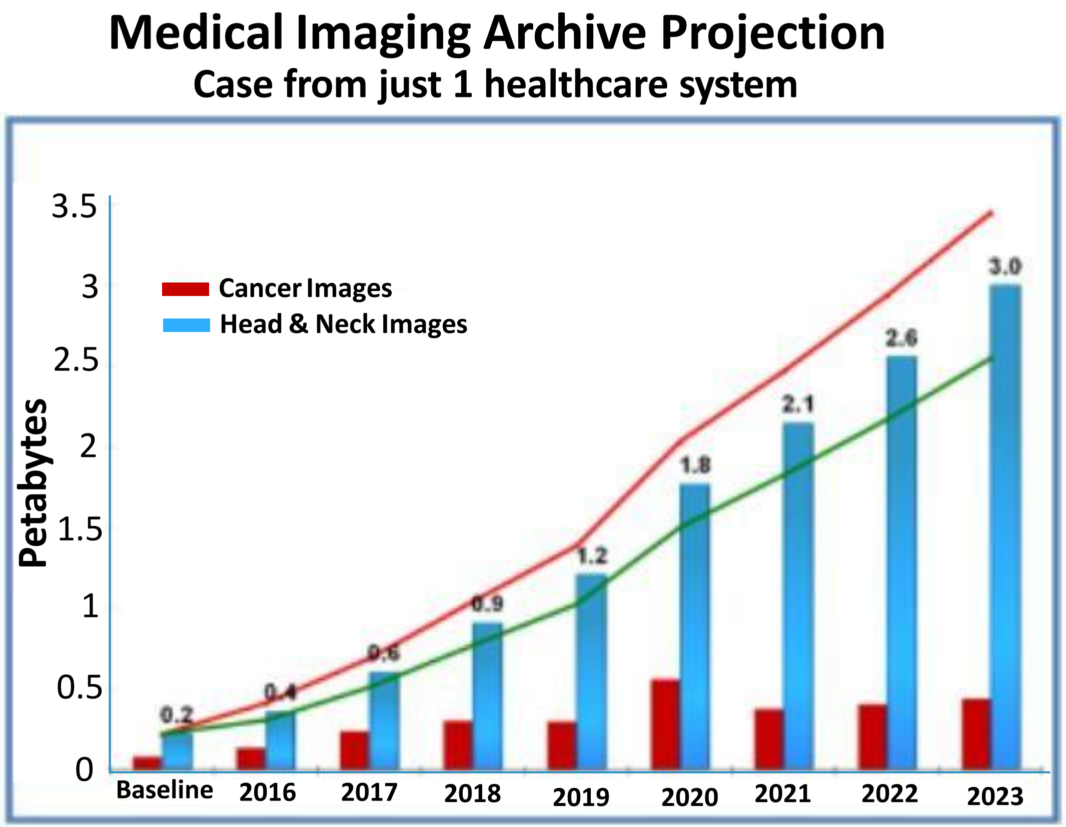
2006 2014 2017 2017 2019

Data Curation Framework (DCF)

5

We are at an Inflection Point in Healthcare - TRENDS

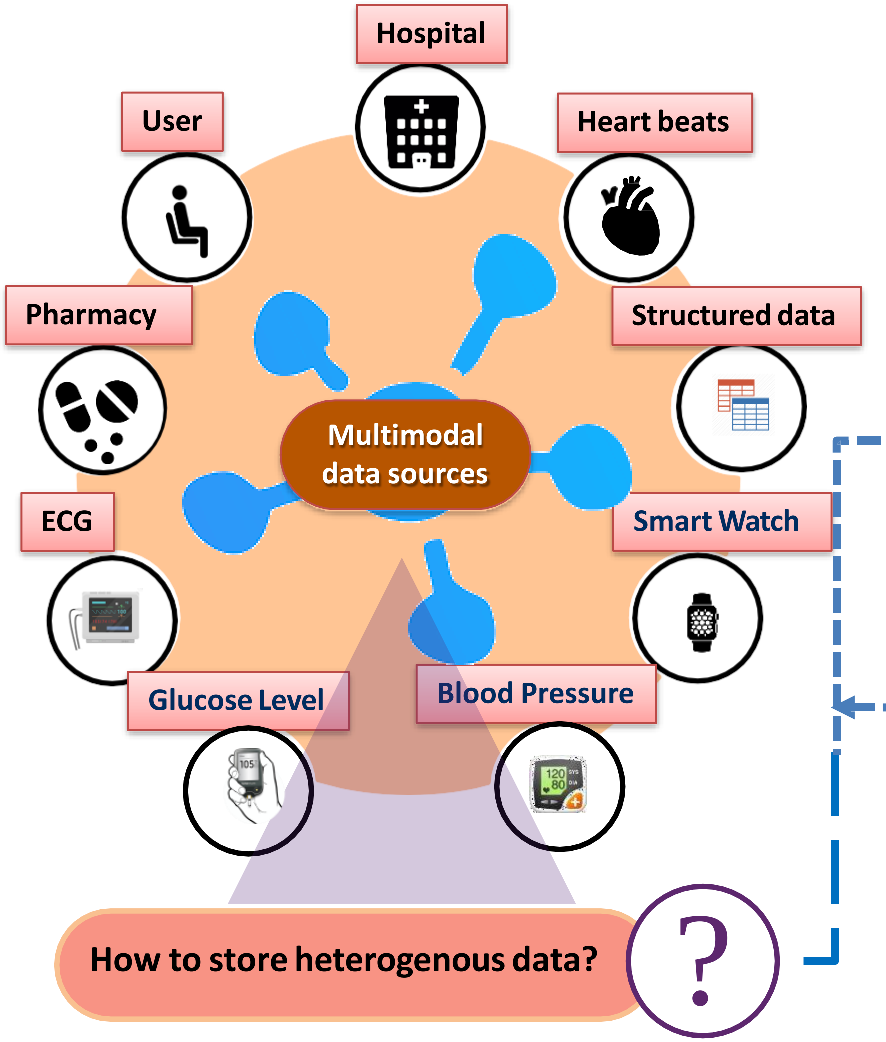




Source: McKinsey Global Institute Analysis
ESG Research Report 2023
– North American Health Care Provider Market Size and Forecast

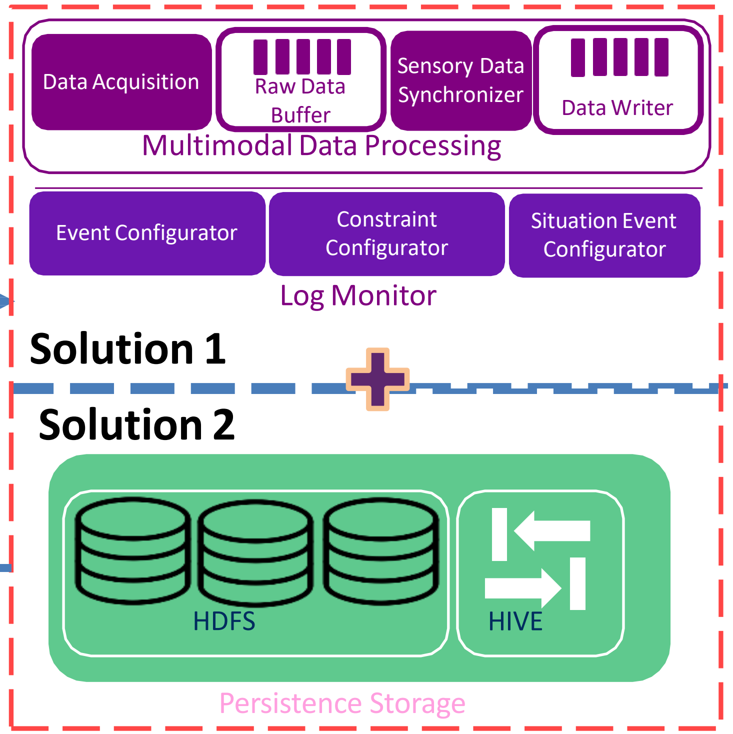
Data Explosion projected to reach 50 Zetabytes by 2025, with a 44-fold increase from 2023

Challenges & Solution



Limitations & Challenges

- *Lack of support to acquire device independent data acquisition*
- *Lack of support for evolution and reusability of sensory data*



Limitations & Challenges

- Lack of support to non-volatile persistence of healthlog in Big data environment
- Lack of support for data driven knowledge generation

Benefits

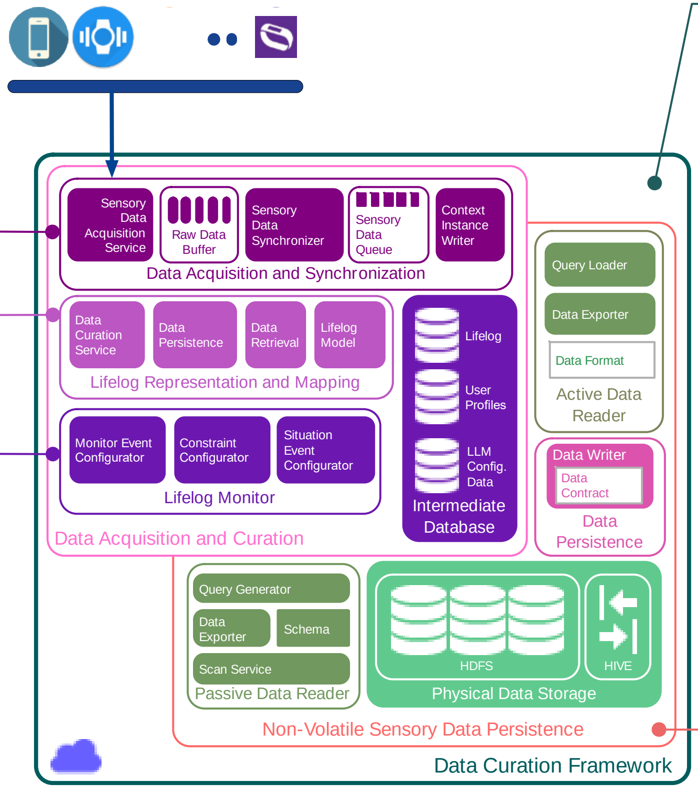
- Data variety and volume
- Understanding variety of unstructured data formats
- Design and development of consistent data models for different sources of data
- Identification of current activities form life-log to generate the alarm notification.

Introduction to Data Curation Framework (DCF)

- Acquisition of multimodal raw sensory data in real-time
- Synchronization of multimodal raw sensory data in a distributed
- Preparation of data instances for context determination

- Curation and persistence of user context in the form of user lifelog
- CRUD operations for user lifelog evolution

- Monitoring of user lifelong for situations to respond
- Hosting and execution of static situations
- Incorporation and execution of dynamic situations

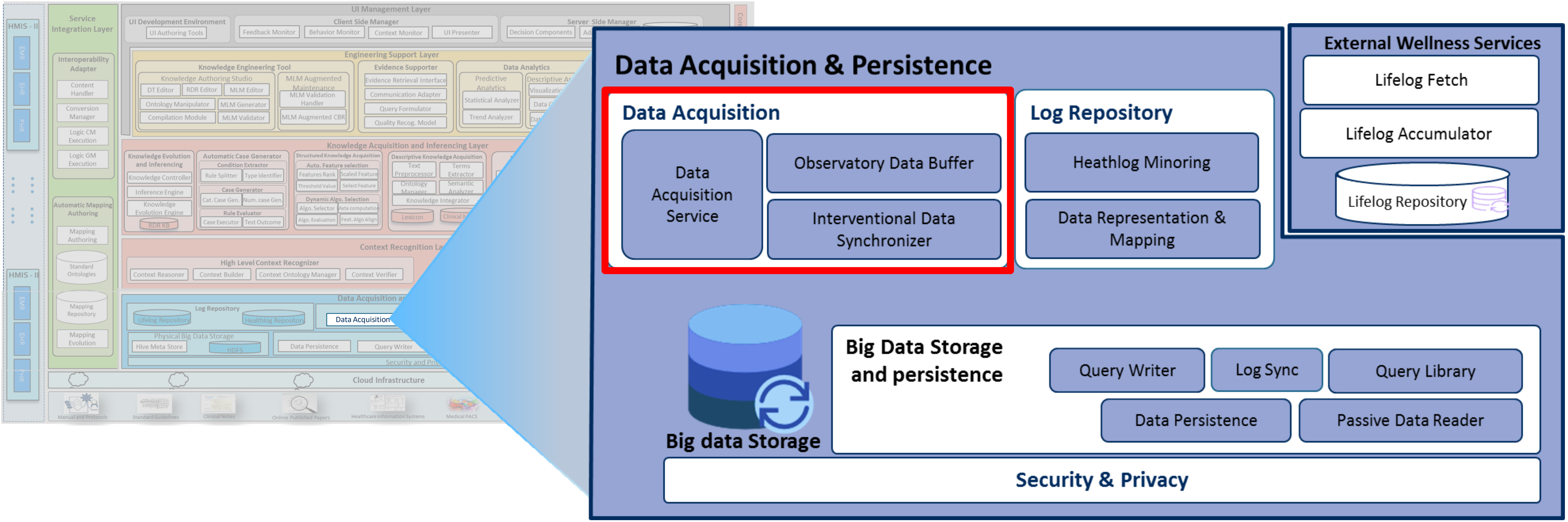


<u>Responsibilities</u>

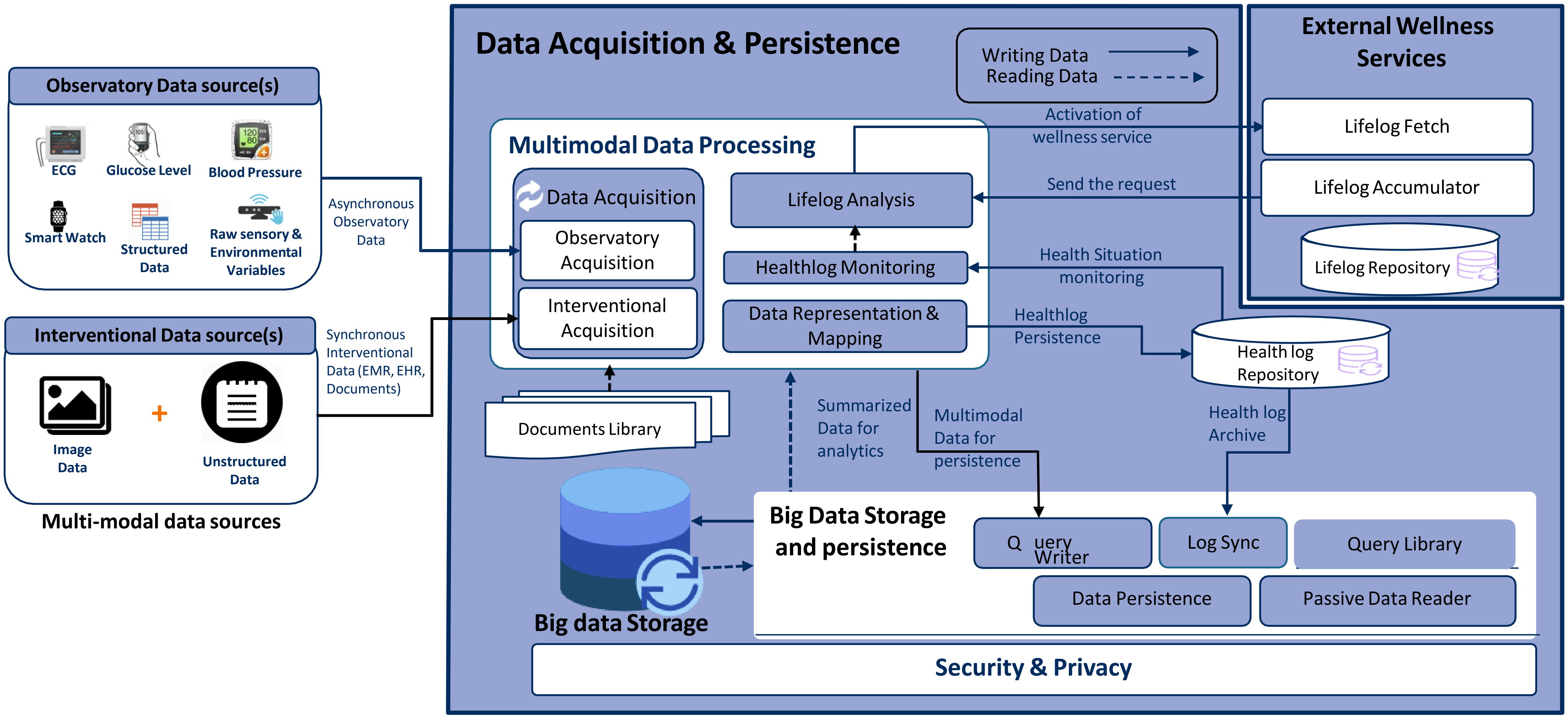
1. Device Independent sensory data acquisition
2. Curation of context-rich user lifelog
3. Monitoring of user lifelog for push-based interventions
4. Support for the evolution and re-usability of sensory data
5. Integrated as a core foundation to health and wellness platforms

- Non-volatile persistence of raw sensory data and user lifelog in a Big Data environment
- Active interface to Big Data for visualization and analytics
- Passive interface to Big Data for data driven knowledge generation

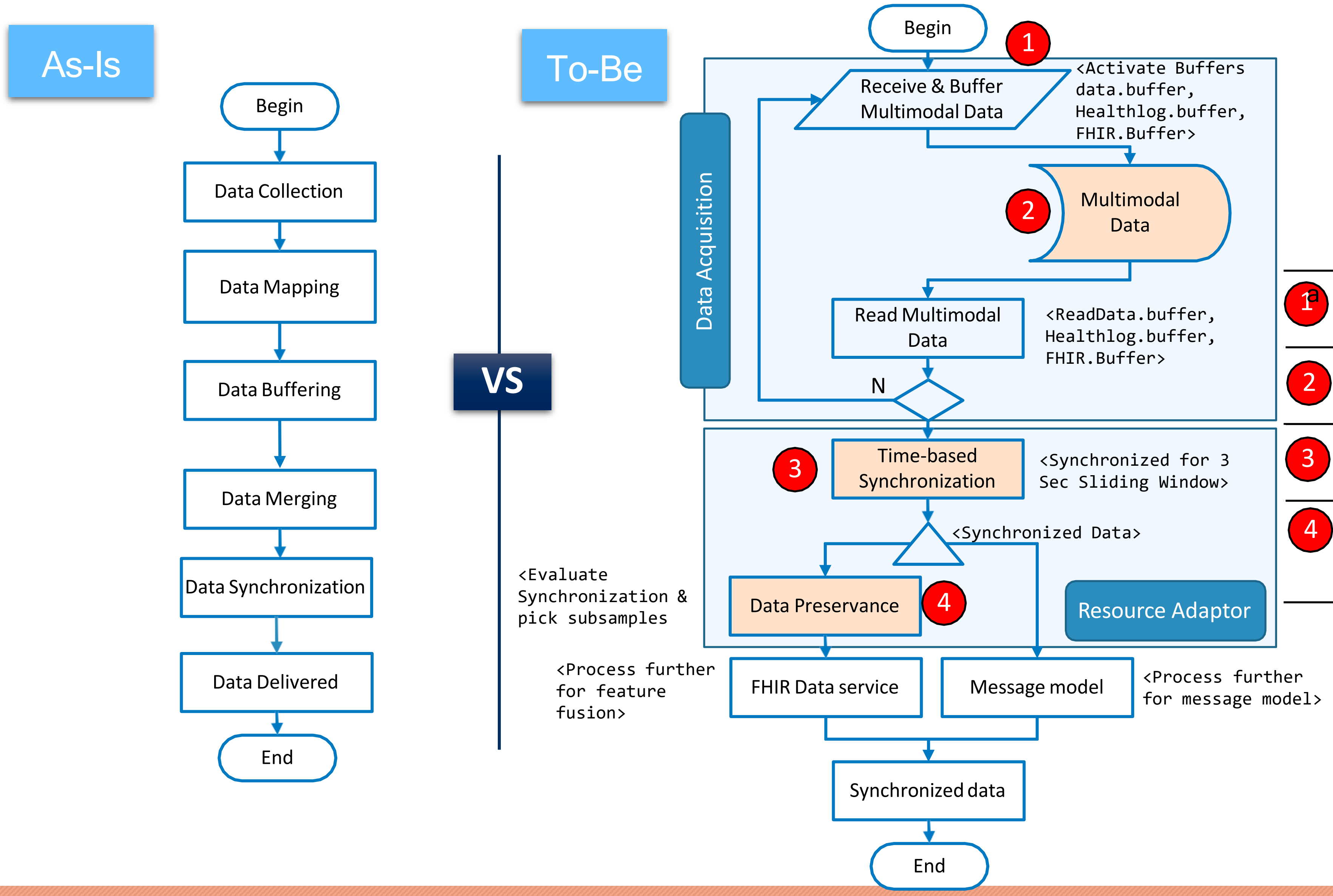
Data Acquisition and Persistence – Functional Architecture



Data Acquisition & Synchronization – Detailed Architecture



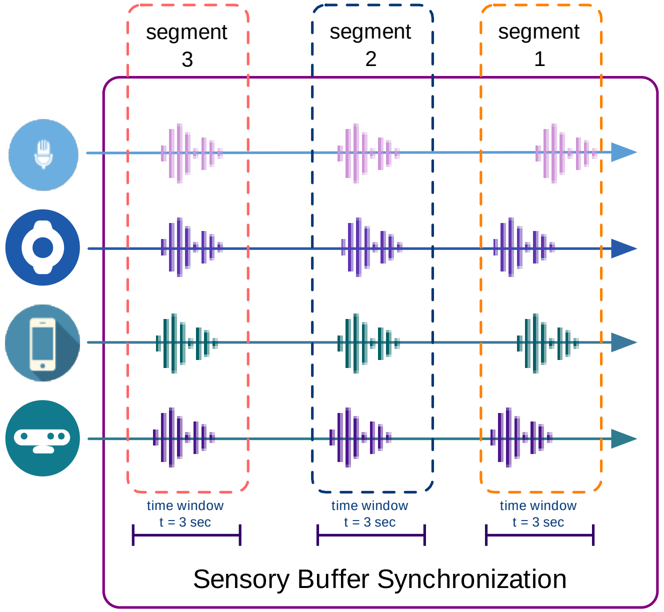
Data Acquisition – Flowchart



1. Receive the multimodal data from the heterogenous sources
2. Proposed the time-based sliding window of size 3sec
3. Perform time-based synchronization for healthcare data.
4. With change in standard scheme, Data preservance can re-iterate over preserved data for transformation

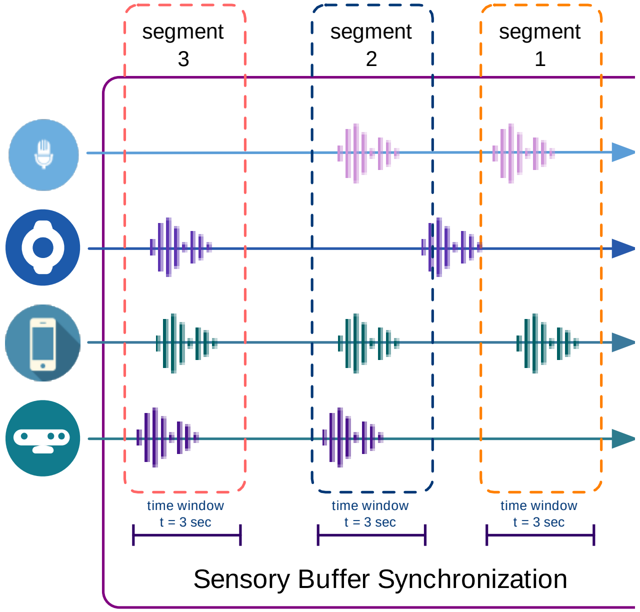
Sensory Data Synchronization Strategies

1. Complete Sync



- Executes when all the required sensory data is received in time window
- Support for highly accurate context determination
- Only possible when all the data sources are almost time- and communication-synced

2. Eager Sync



- Executes in regular interval time without the dependence on data sources
- Support for real-time execution no delays
- Ignores out-windowed packets; resulting in lower accuracy for c determination

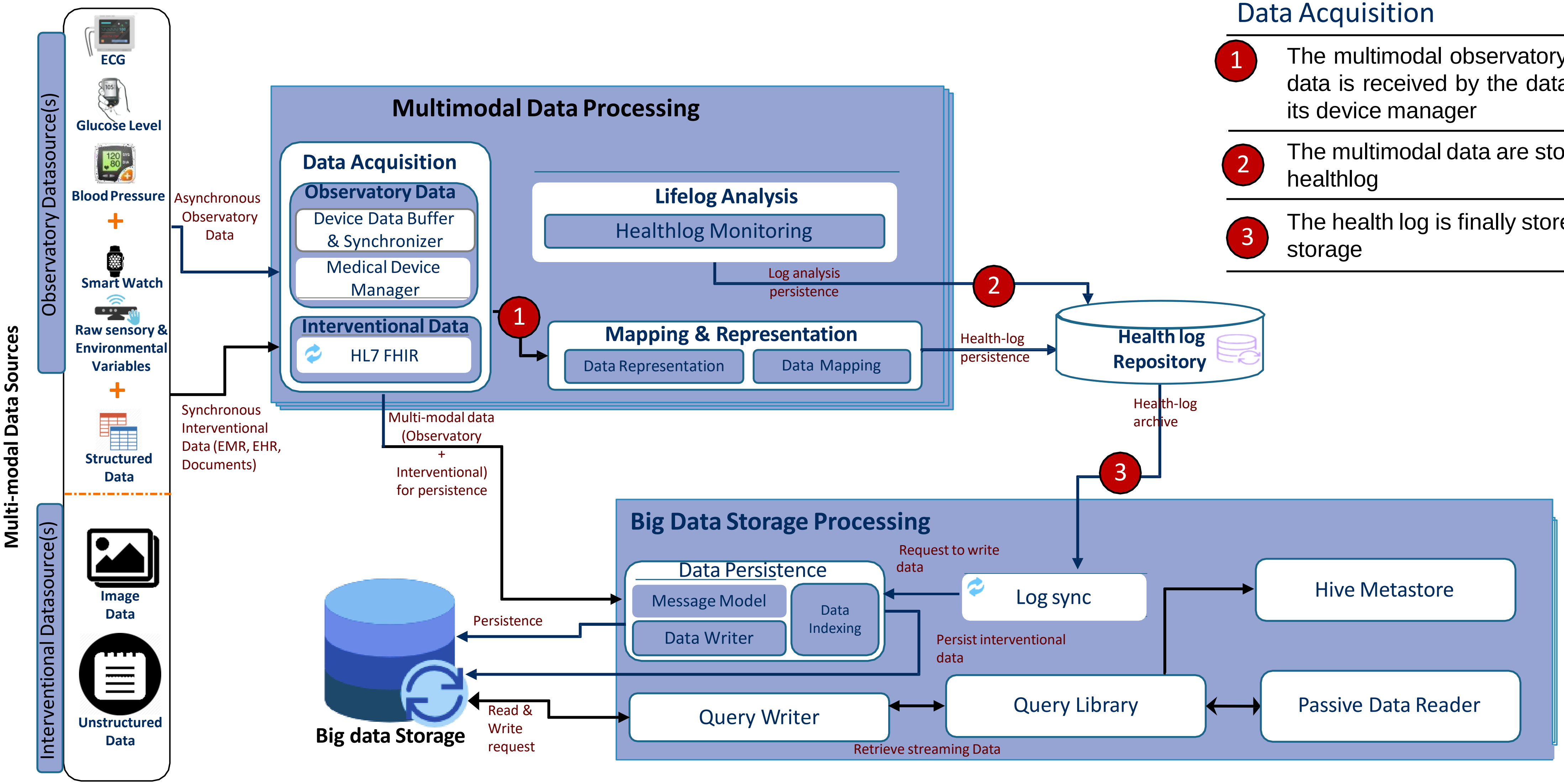
Algorithm 1 Time-based synchronization for raw-sensory data acquisition.

Require: $buffer_{src}[1, \ldots, n]$: n is the total number of data sources

Ensure: $buffer_{dst}$: queue for time-synchronized data packets

procedure SYNC($buffer_{dst}$)
    $msg \leftarrow create\_msg(NULL)$
    while $i \le No\_of\_datasources$ **do**
        $buffer_{src}[i] \leftarrow Recv(data)$ ▹ Complete-sync execution
        $msg.add(create\_msg(buffer_{src}[i]))$
        if $time_{sec} > time\_window$ **then**
            if $send\_only = TRUE$ **then** ▹ Incomplete-sync: Eager execution
                $break$
            end if
            while $j \le No\_of\_datasources$ **do** ▹ Incomplete-sync: Rendezvous execution
                $j \leftarrow i + 1$
                if $buffer_{src}[j].has\_contents$ **then**
                    $msg.add(create\_msg(buffer_{src}[j]))$
                end if
            end while
            $break$
        end if
    end while
    $msg.timestamp \leftarrow buffer_{src}[i].timestamp$
    $buffer_{dst}.enqueue(msg)$
end procedure

Data Acquisition – Execution Flow



Data Acquisition

1. The multimodal observatory & interventional data is received by the data acquisition with its device manager
2. The multimodal data are stored inside the healthlog
3. The health log is finally stored in the big data storage

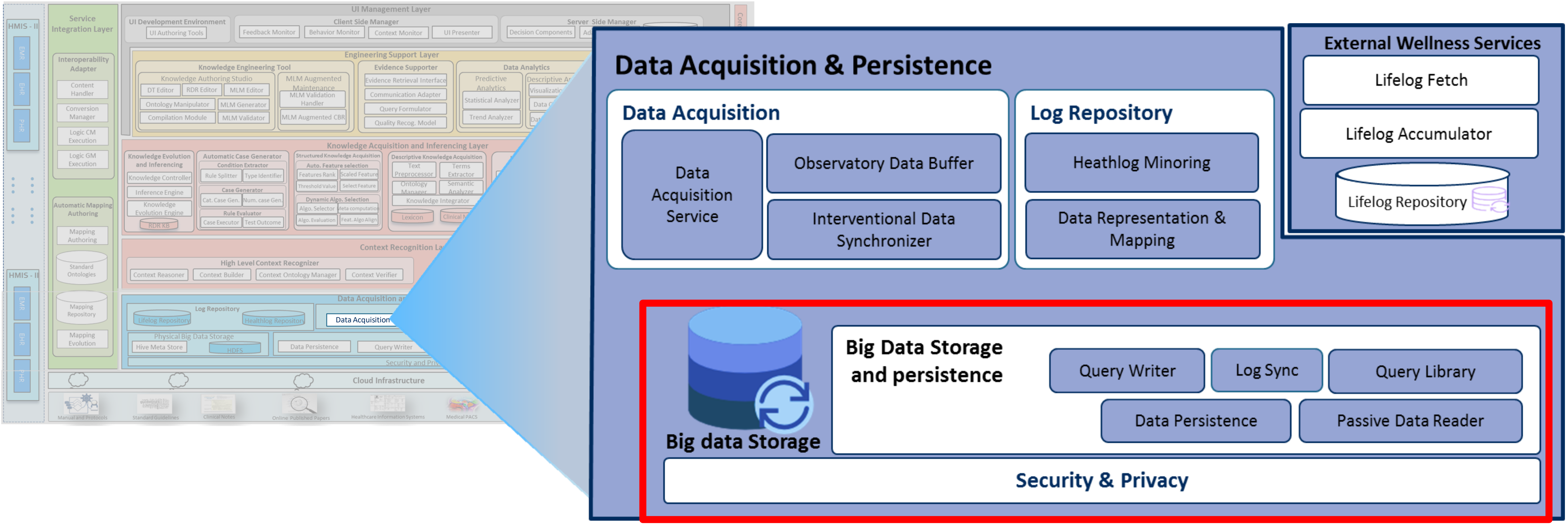
Contribution

- Acquisition of **multimodal data** at real-time
- Synchronization of Heterogeneous data per **medical devices** and timestamp

Benefits

- Buffered pipe-lining of data to **mapping and representation** and for Big data storage.
- **Non-blocking IO** to avoid Communication bottlenecks

Big Data Storage and Persistence



Big Data Challenges...

And Require New Technologies

1 Volume

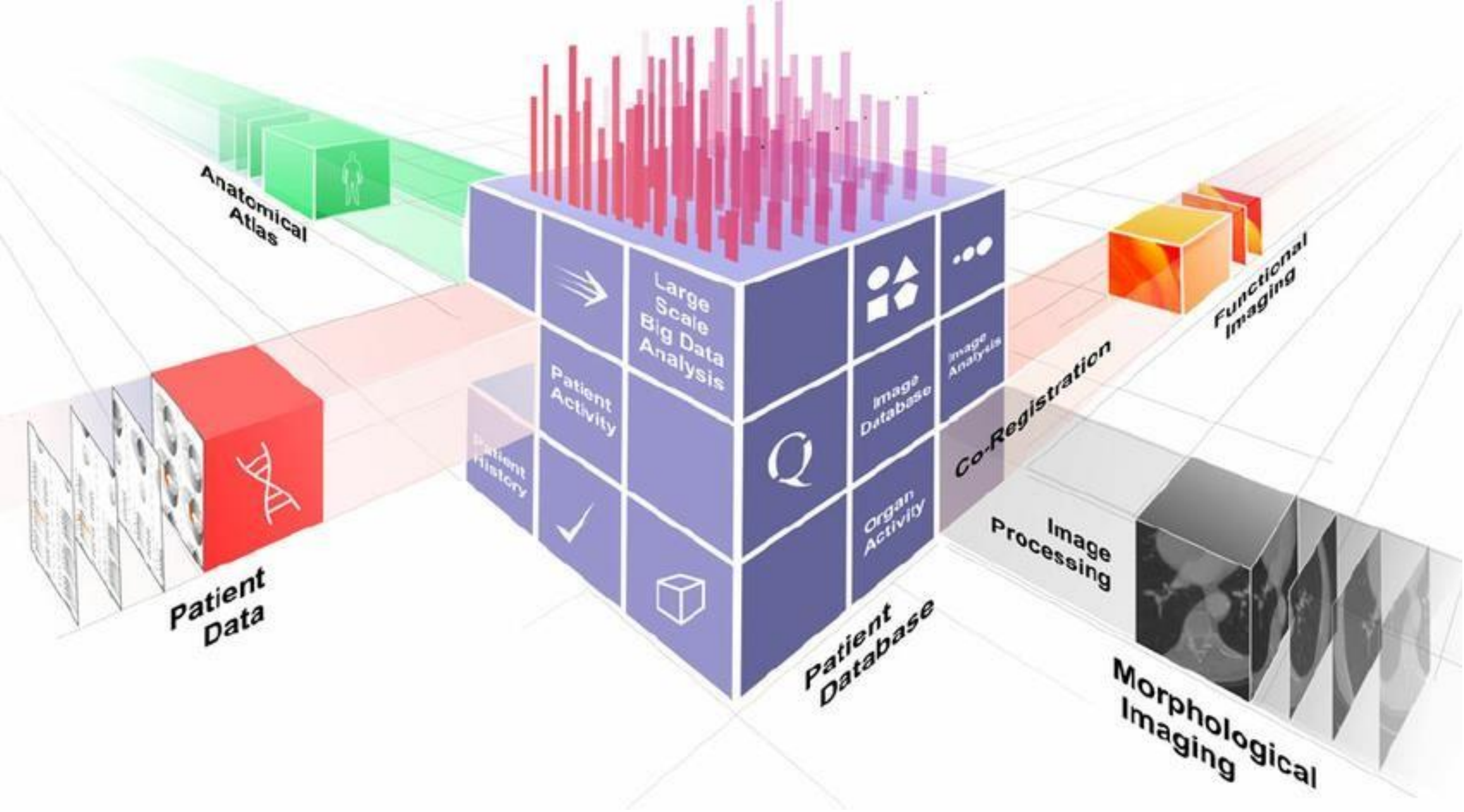


Fig. Showing Processing schema of imaging toward big data

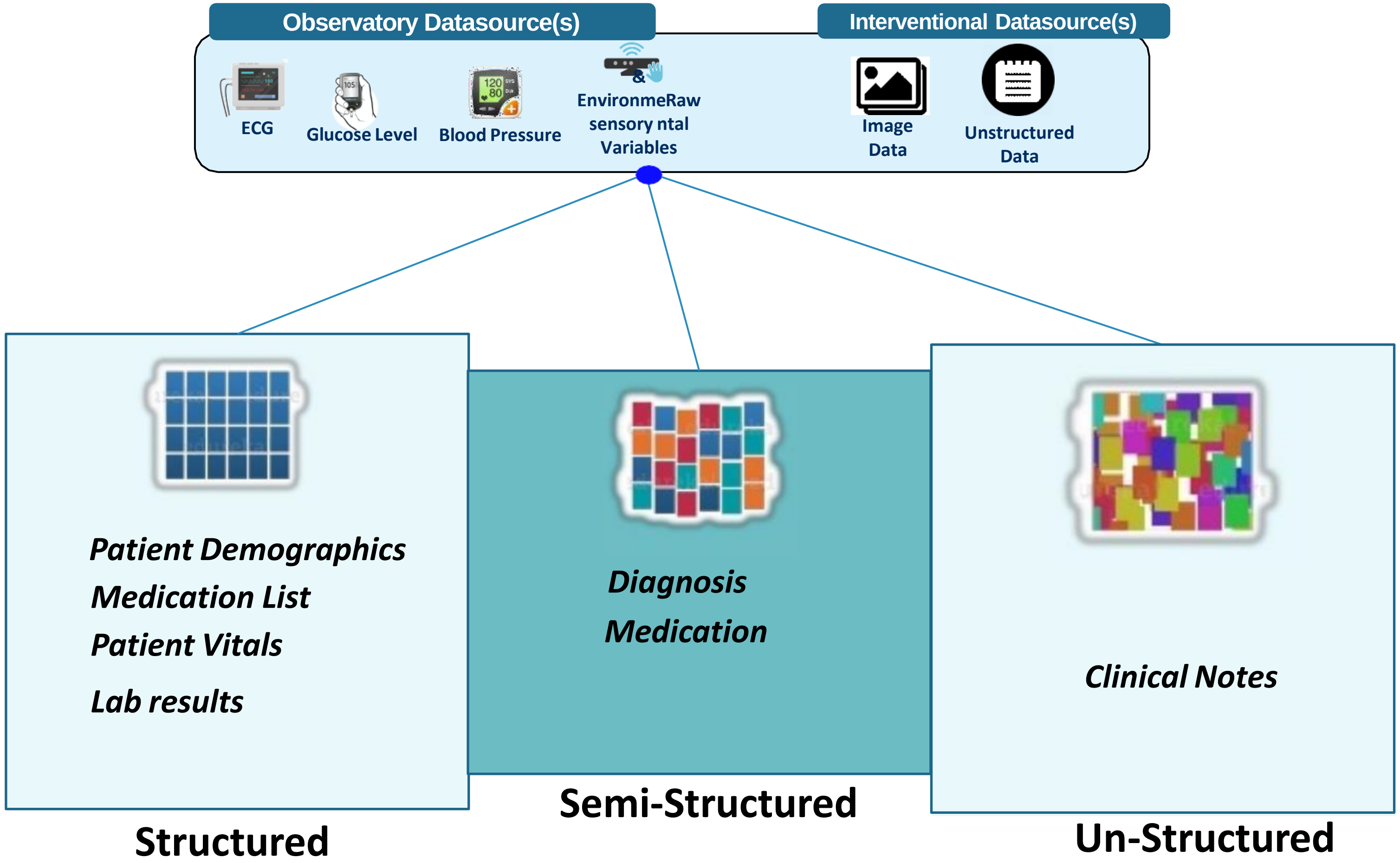
Source: Andreu-Perez, Javier, et al. "Big data for health." *IEEE journal of biomedical and health informatics* 19.4 (2015): 1193-1208.

Big Data Challenges...

And Require New Technologies

1. Volume
2. Variety

Different kinds of data being generated from various sources



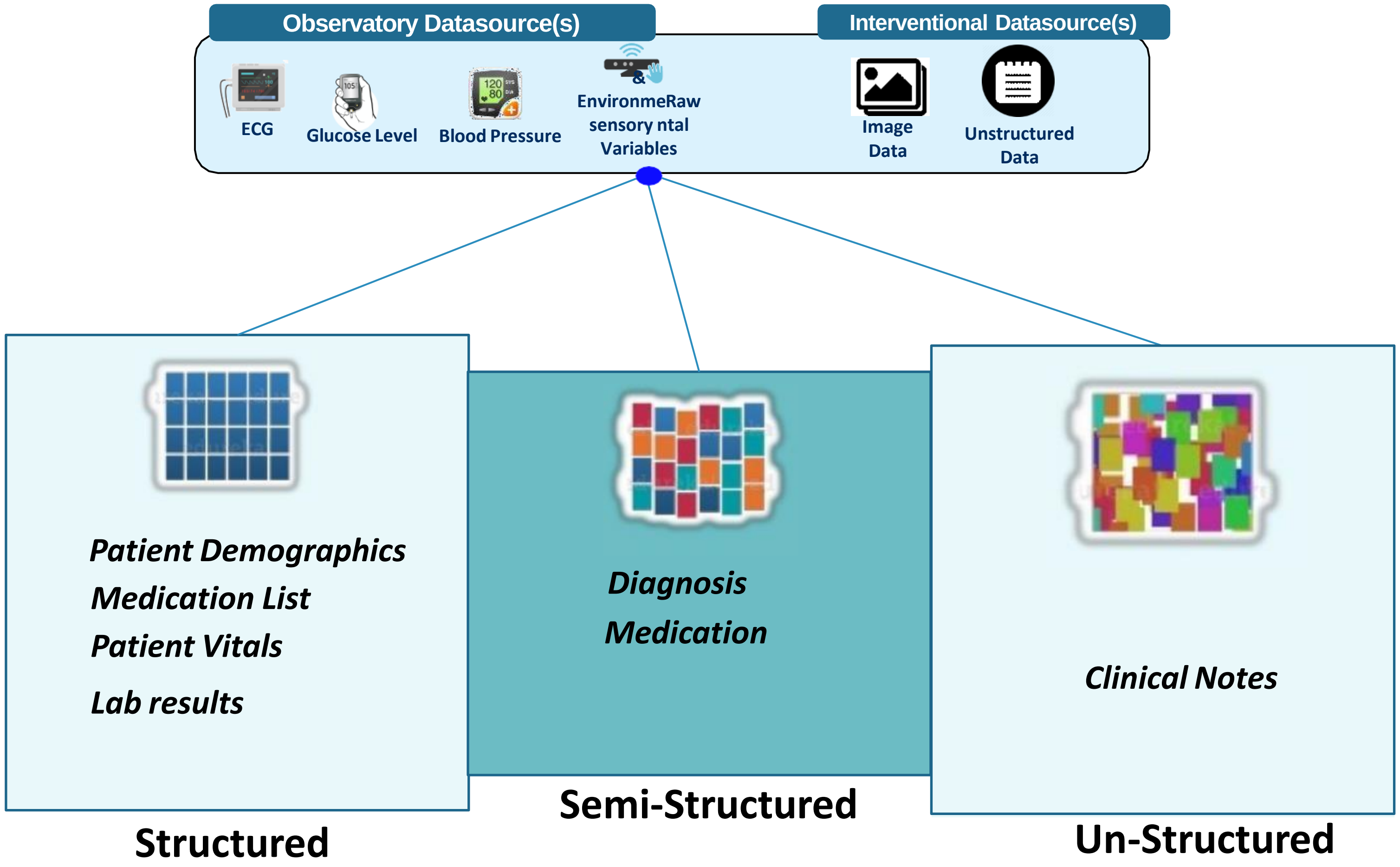
Big Data Challenges...

And Require New Technologies

1 Volume

2 Variety

Different kinds of data being generated from various sources



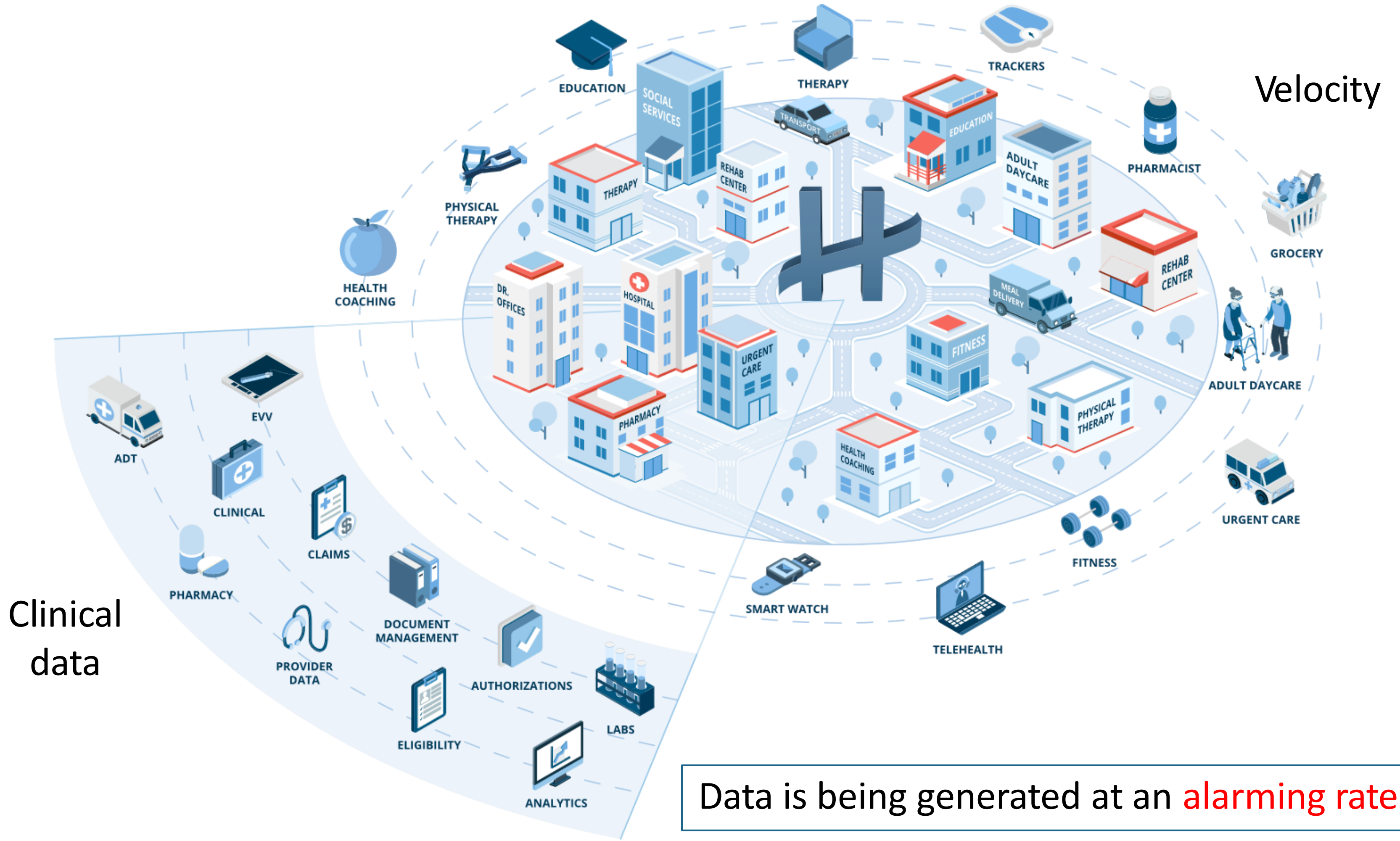
Big Data Challenges...

And Require New Technologies

1 Volume

2 Variety

3 Velocity

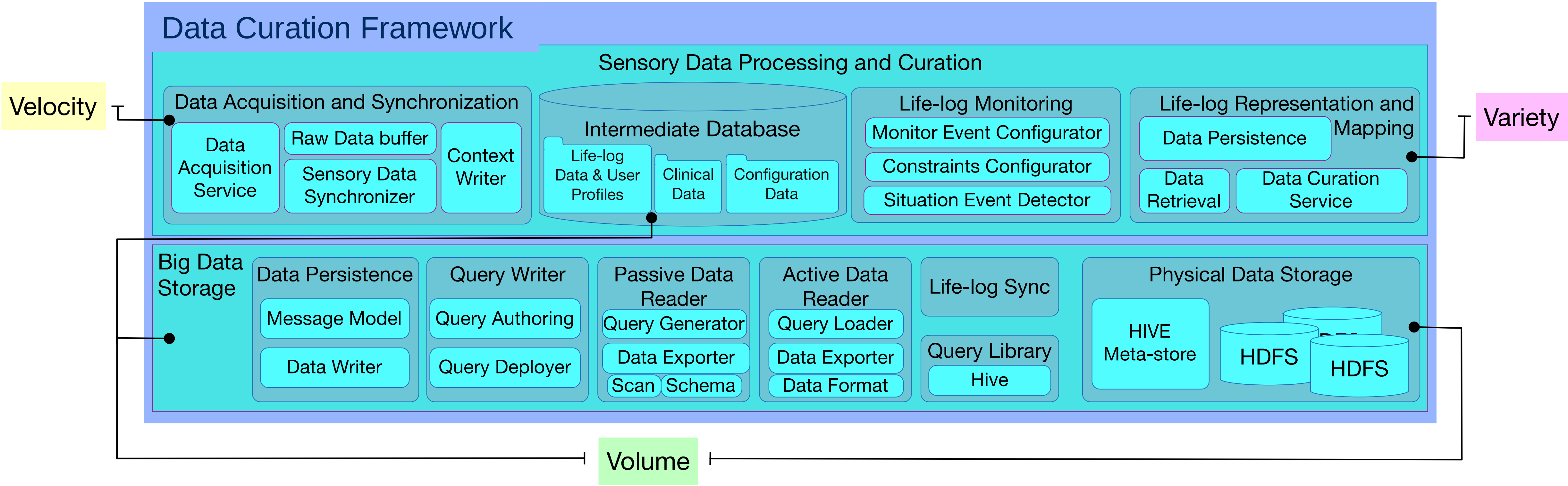


Data is being generated at an alarming rate

Source: Lee, Choong Ho, and Hyung-Jin Yoon. "Medical big data: promise and challenges." *Kidney research and clinical practice* 36.1 (2017): 3.

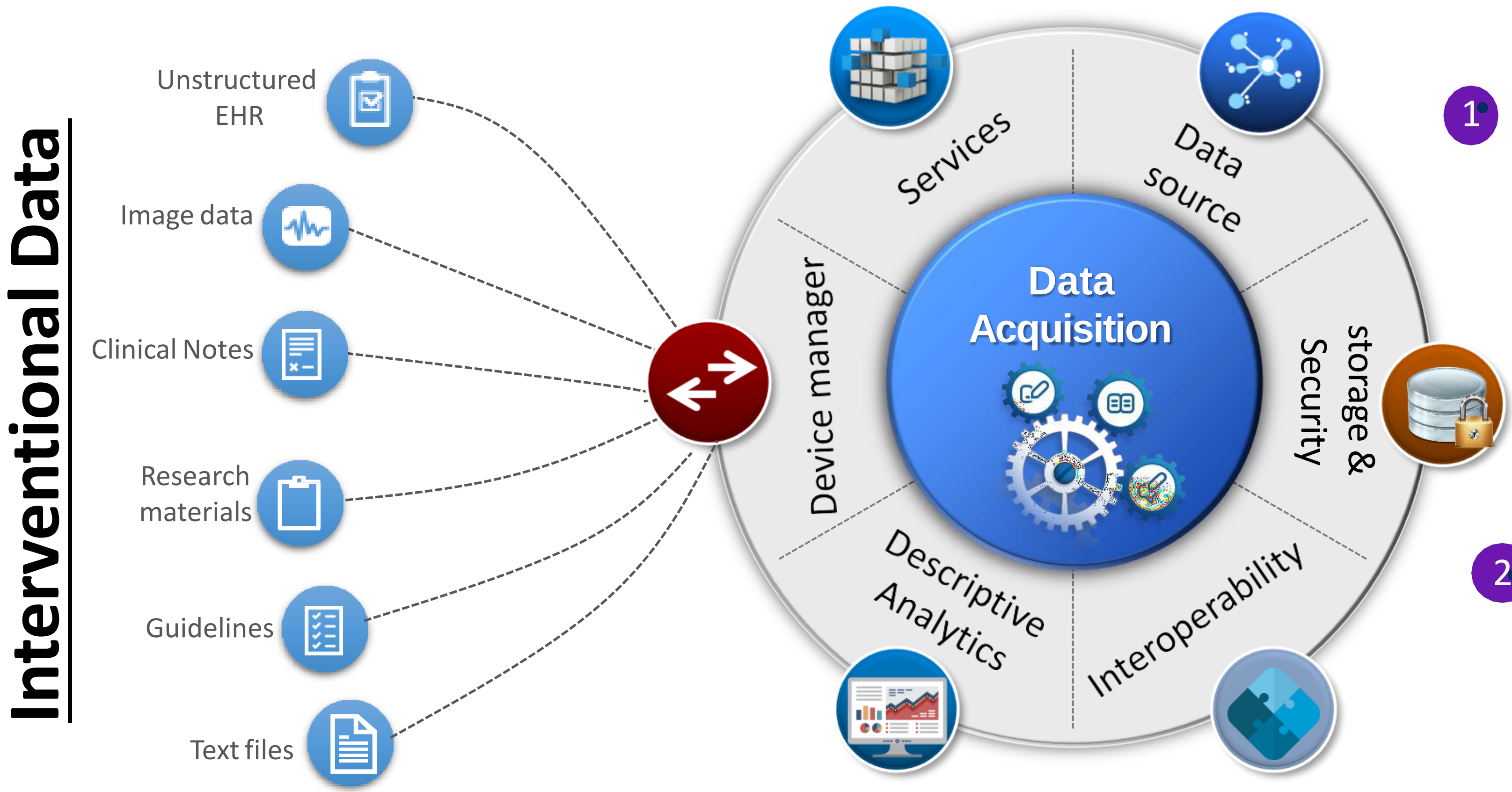
Data Curation Framework (DCF) & Big Data

- Data Curation Framework has been adopted as the foundation for Mining Minds platform (Health and wellness platform) as independent layer called Data Curation Layer or DCL



Source Code: https://github.com/ubiquitous-computing-lab/Mining-Minds

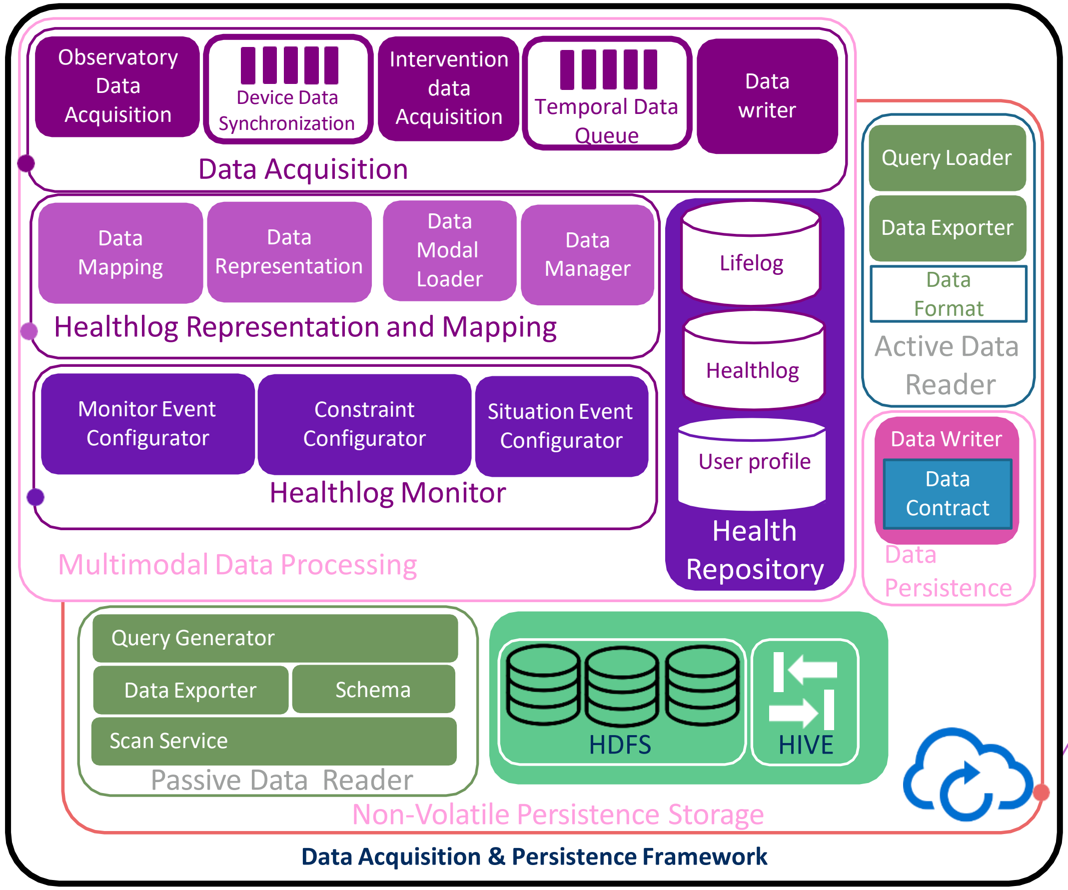
Motivation



1. To store raw sensory, environmental variables in a large-scale non-volatile persistence (Big Data) with CRUD operations.
    - *Real-time data storage*

2. For model training and rule generation, knowledge extraction requires interface to selected historic medical data
    - *Passive data read operations*

Goal and Objectives



Goal:

- To store interventional data and FHIR based data in a large-scale non-volatile persistence (Big Data) with CRUD operations.

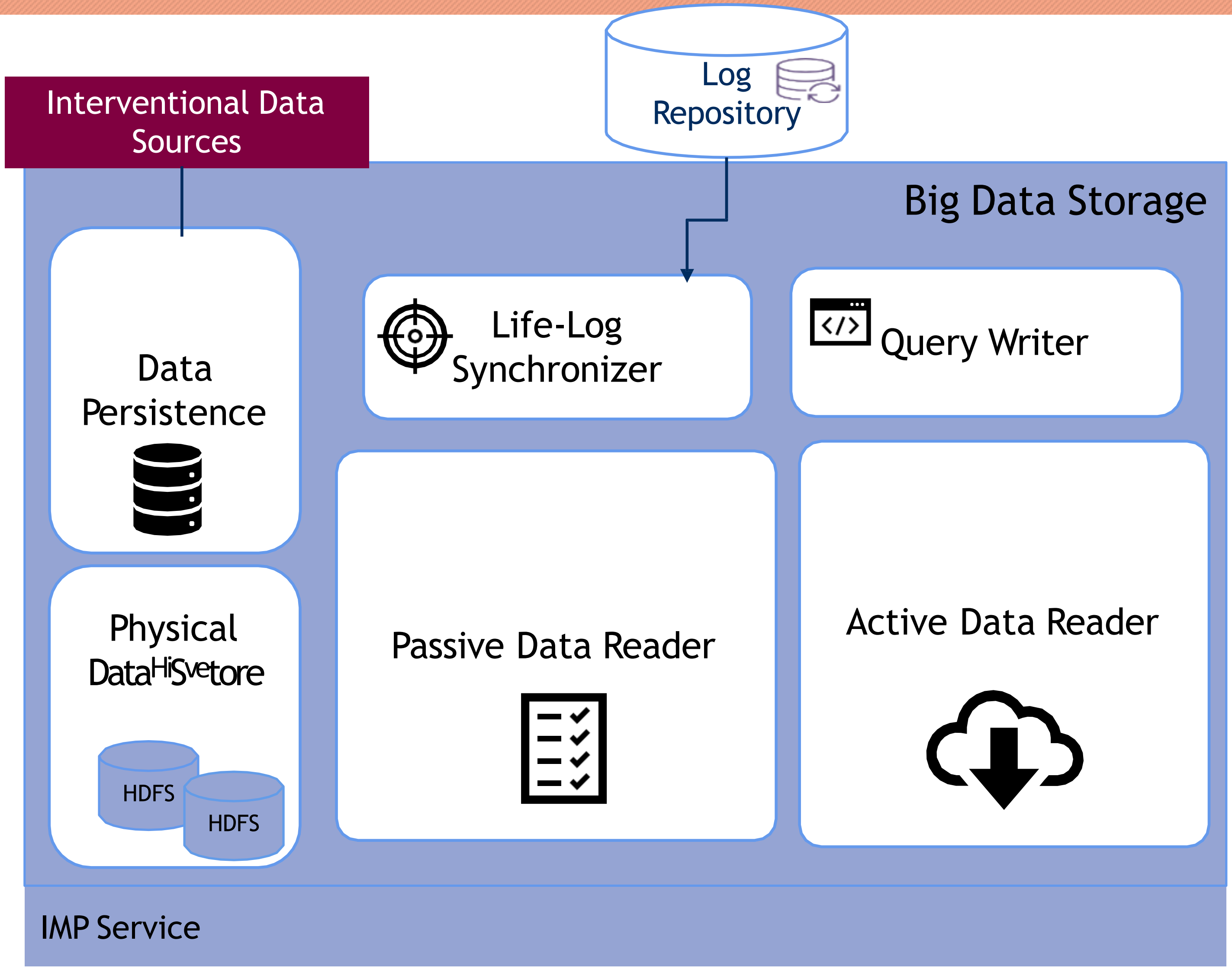
Objectives:

- Non-volatile storage of data from heterogeneous sources with CRUDS operations.
  - Supports of CRUD operations and REST Service end-point
- Build to handle the REST request and provide data for Visualization and Analytics.
  - High Scalability and Interactive.

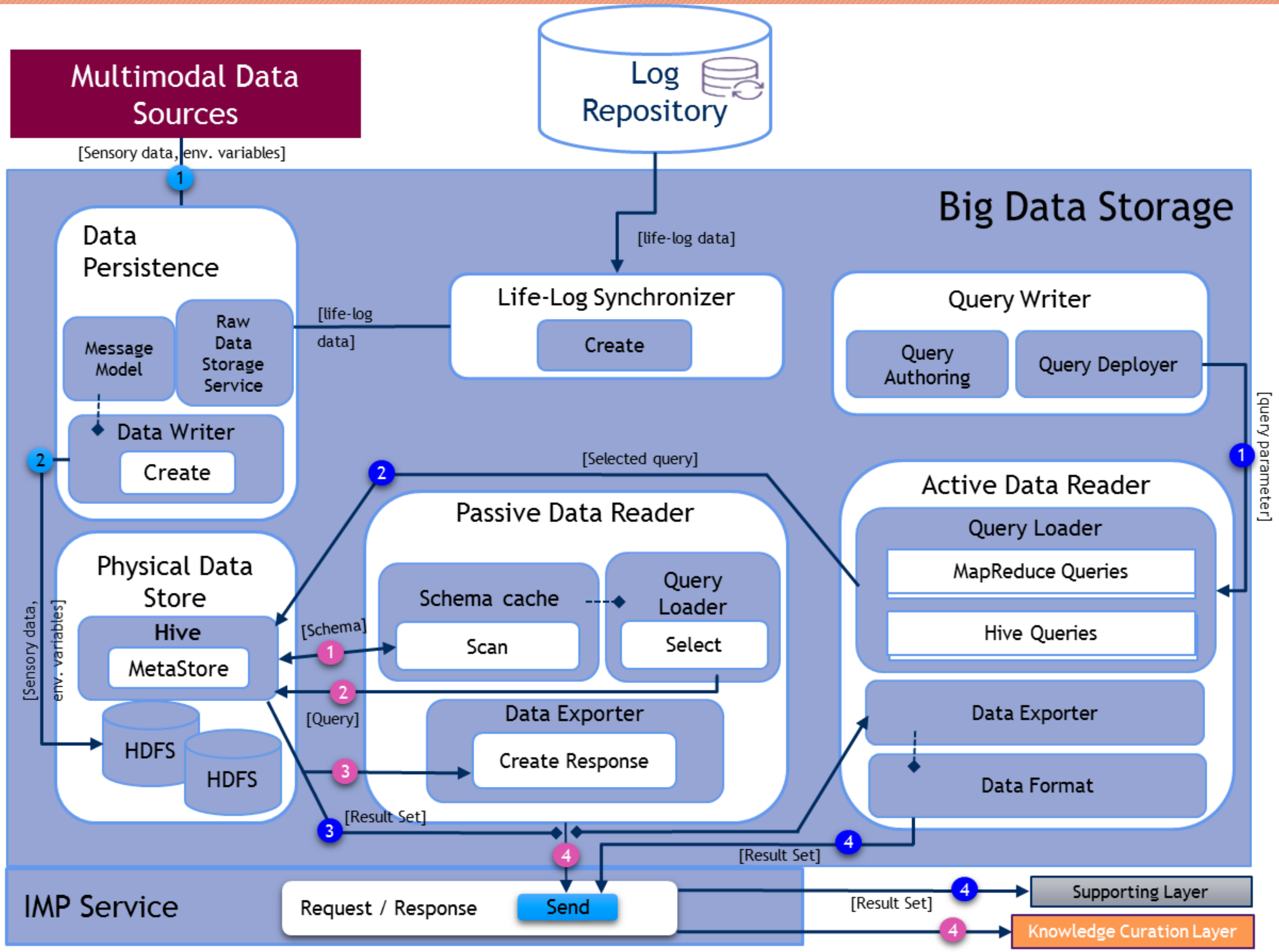
Challenges and Solutions

Motivation	Challenges	Solution	Methods
Support for CRUD operations	Data is distributed over the cloud	Data Reader/Writer	Data reader use the metastore to perform the CRUD operations on the cluster.
Non – volatile persistence	To store the sensory, environmental variables in a large-scale non-volatile persistence	Real time data storage	Data persistence method allows the real-time storage
Scalability with respect to the data	Handling of the large amount of the data	Big data storage	Big data storage should handle large amount of the data and keep scaling with growth.

Big Data Storage Processing - Abstract Architecture

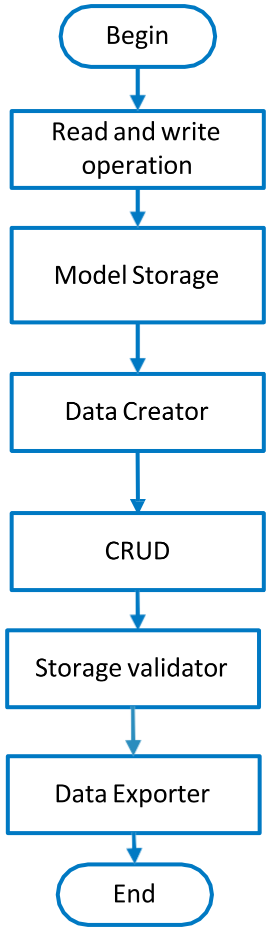


Big Data Storage Processing – Detailed Architecture



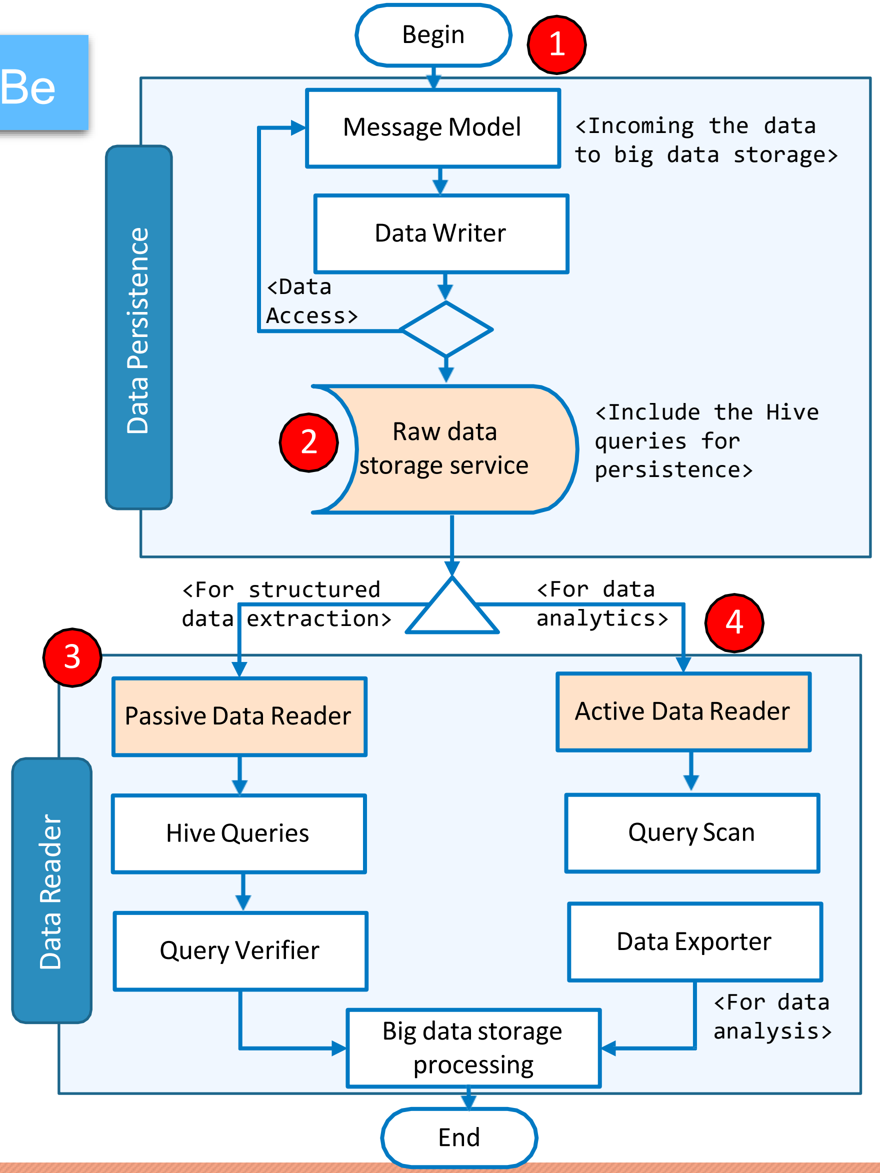
Big Data Storage and Processing – Flowchart

As-Is



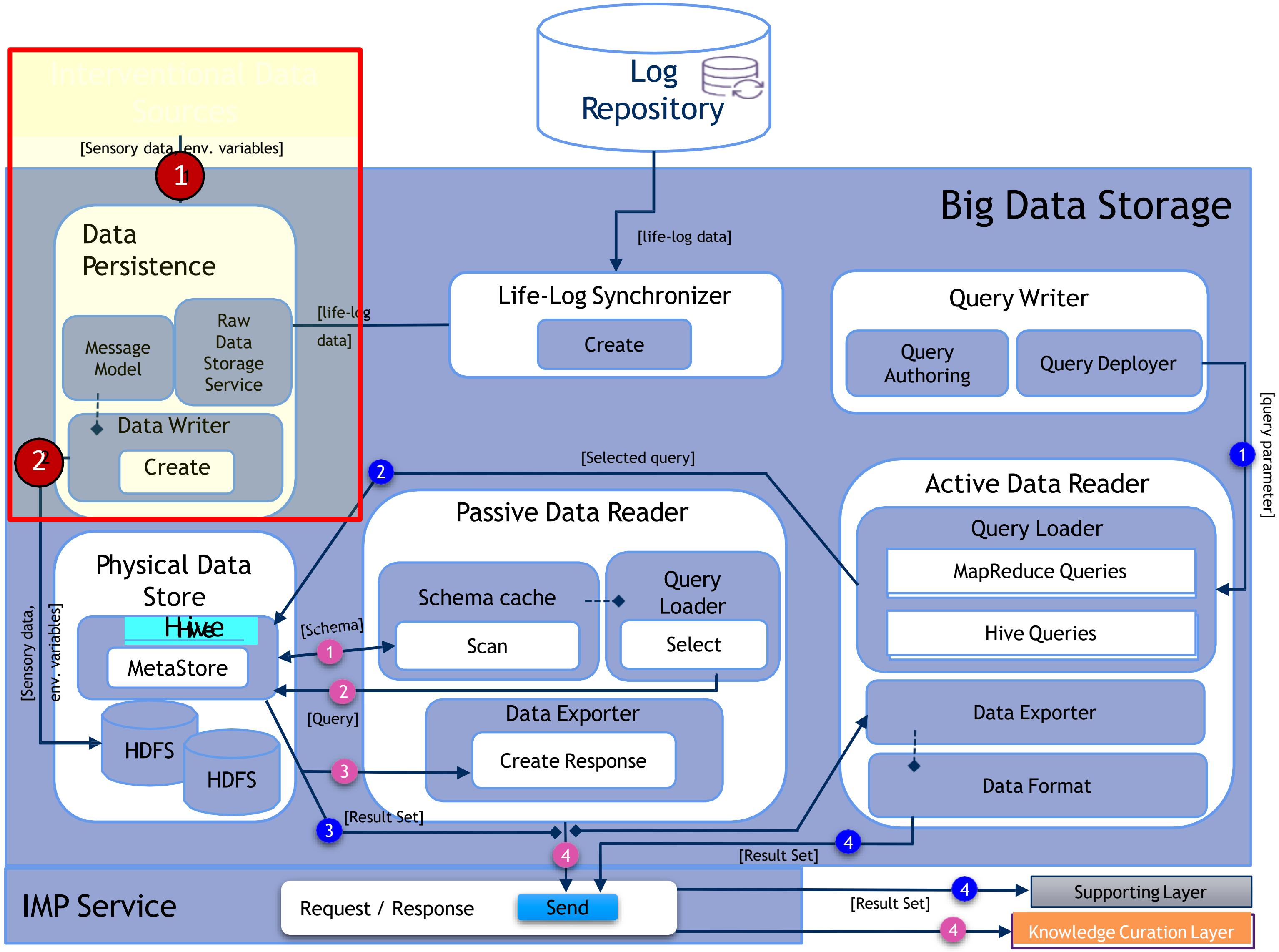
VS

To-Be



1. Interventional data is received by Data Writer of Data Persistence Component.
2. De-serialized message is sent to HDFS for Persistence.
3. Data reader is responsible for handling online data request for visualization and analytics. Data reader selects the query and sent to Physical Data Store for execution
4. Finally the data is stored inside the HDFS

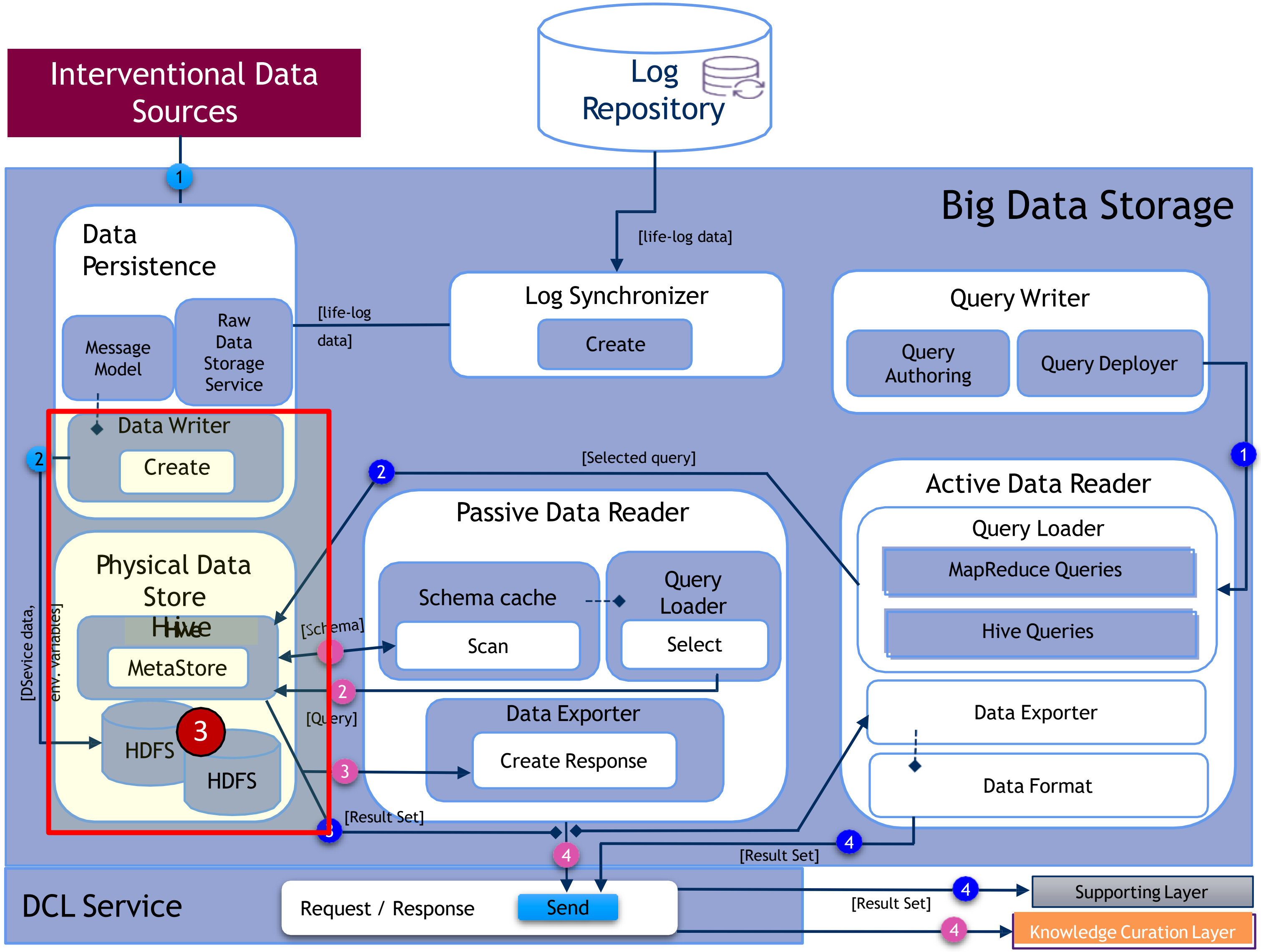
Big Data Storage Processing - Execution Flow



Physical Data Storage

1	Interventional data is received by Data Writer of Data Persistence Component.
2	Data is de-serialized according to the message model

Big Data Storage Processing - Execution Flow



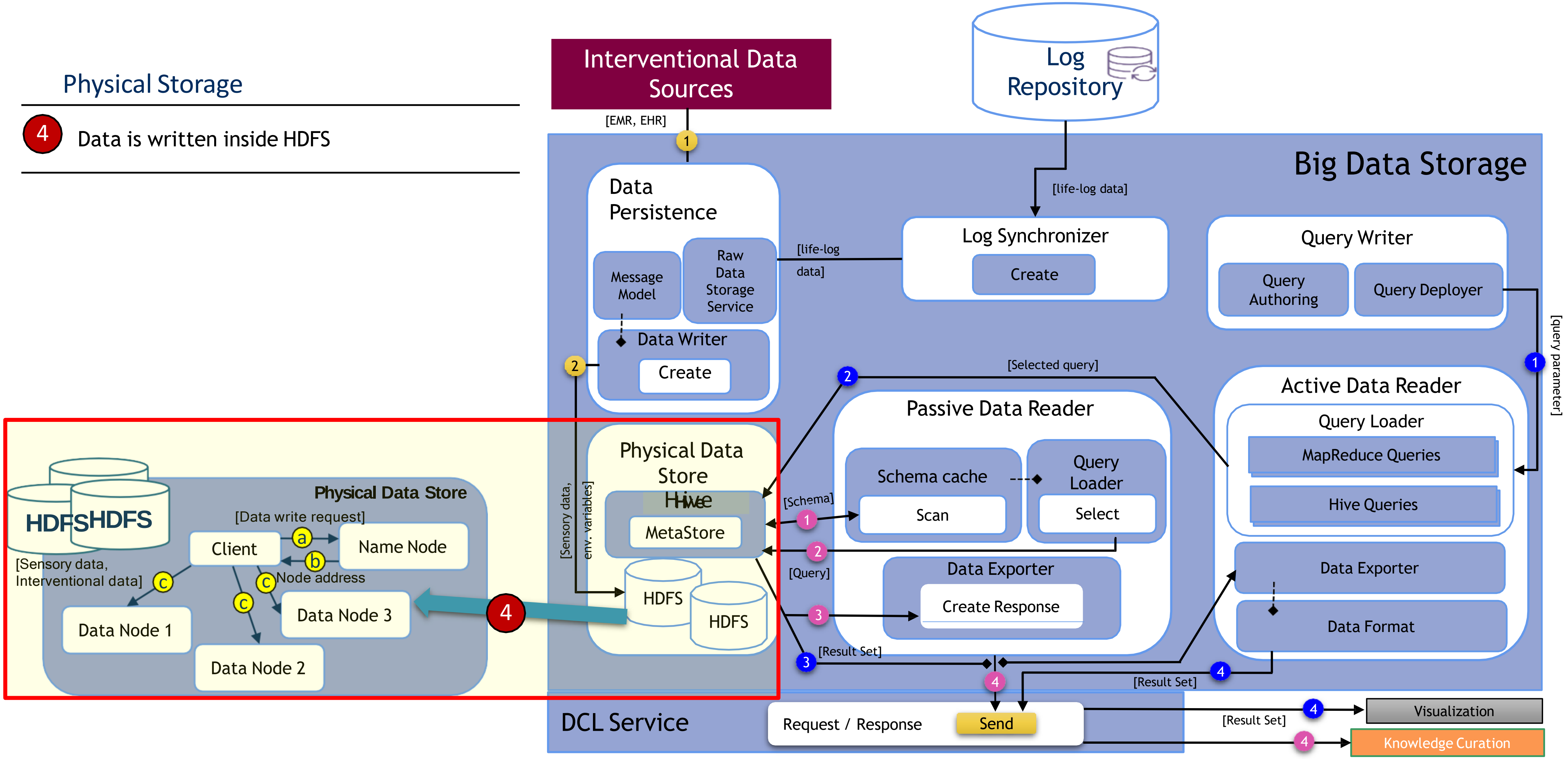
Data is stored

3 De-serialized message is sent to HDFS for Persistence.

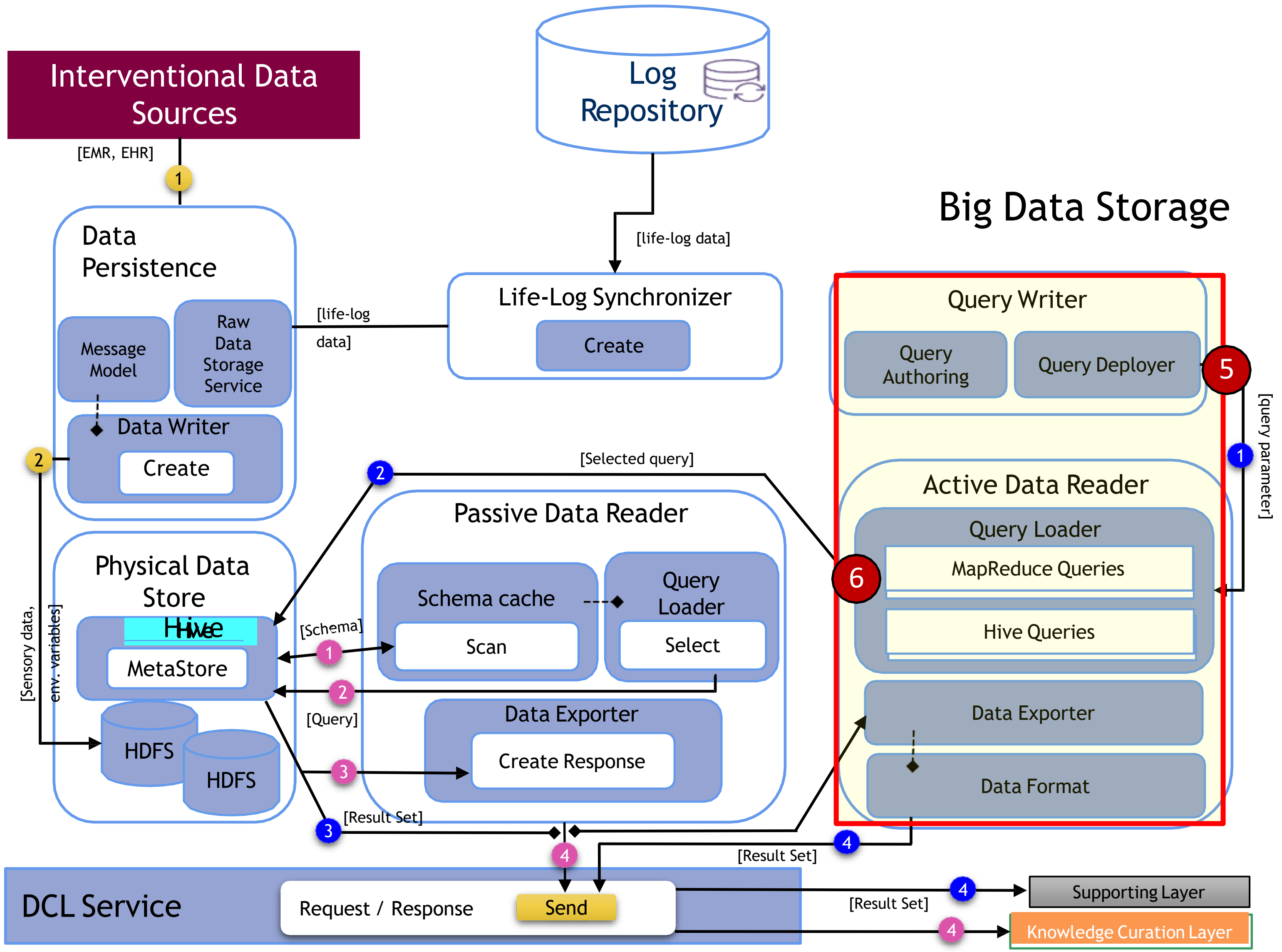
Big Data Storage Processing - Execution Flow

Physical Storage

4 Data is written inside HDFS



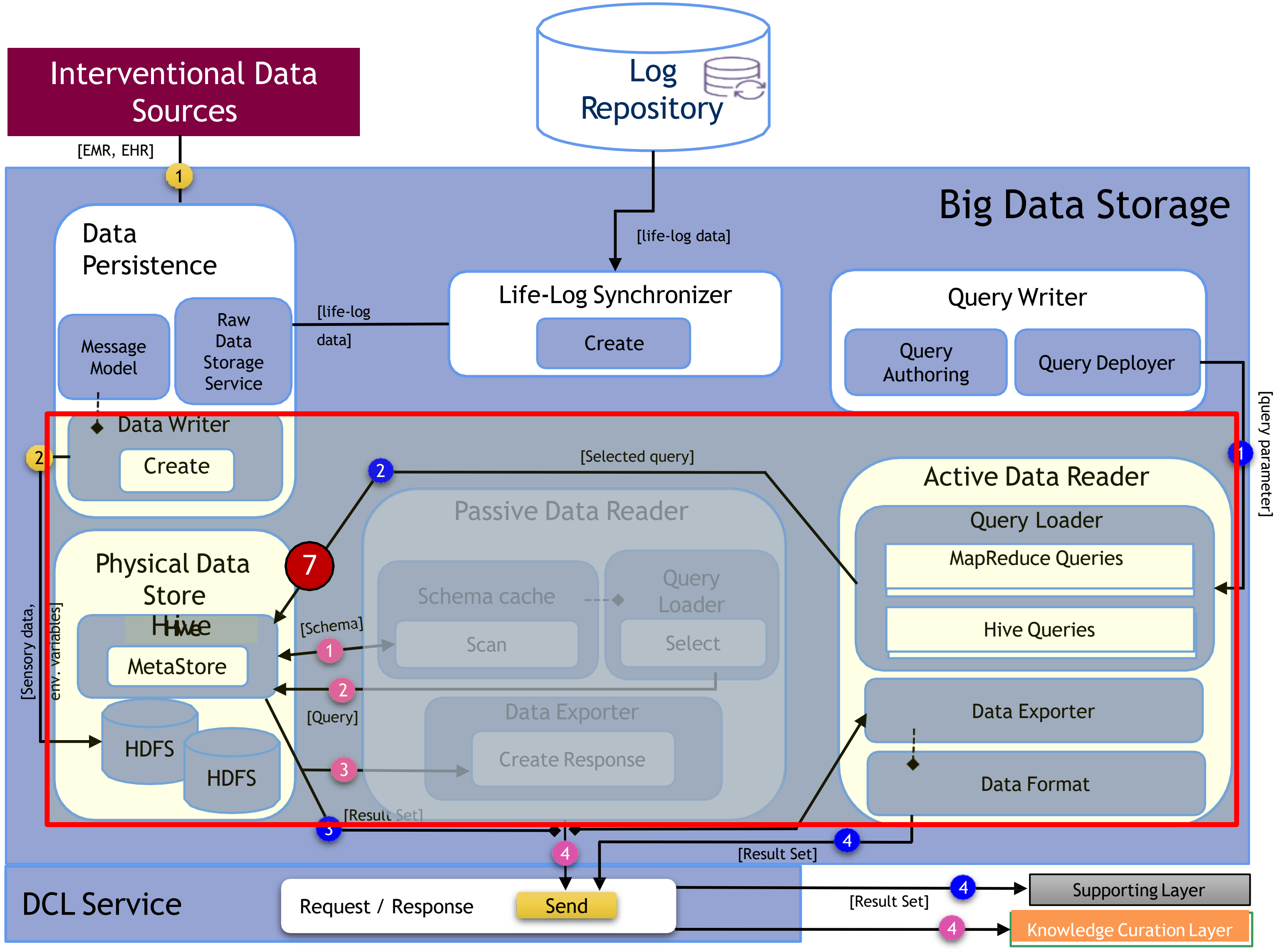
Big Data Storage Processing - Execution Flow



Active Data Reader

5	Active Data reader is responsible for handling online data request for data visualization and analytics
6	Active data reader selects the query depending upon the query parameters

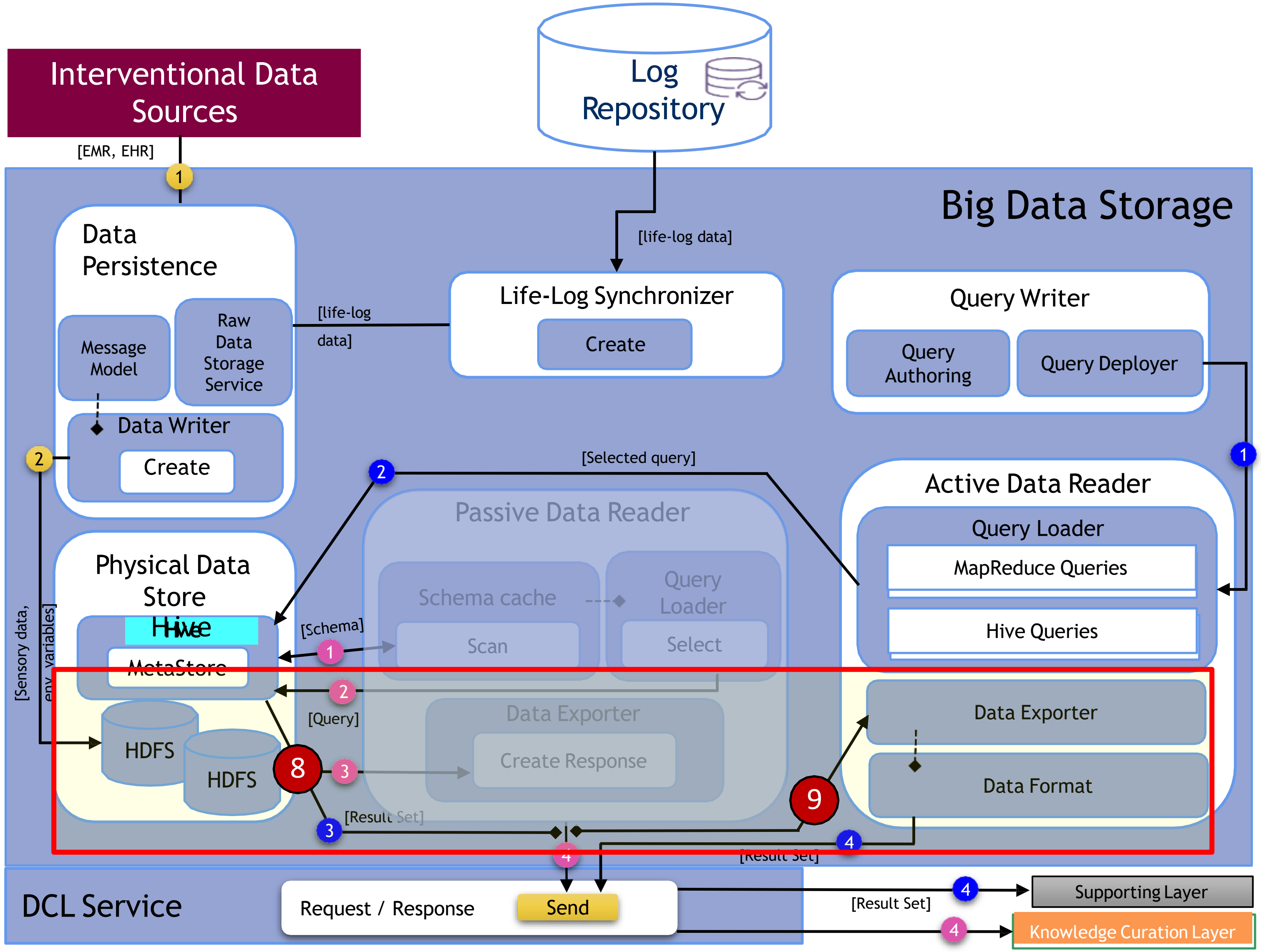
Big Data Storage Processing - Execution Flow



Active Data Reader

7	Selected query is sent to Physical Data Store for Execution

Big Data Storage Processing - Execution Flow

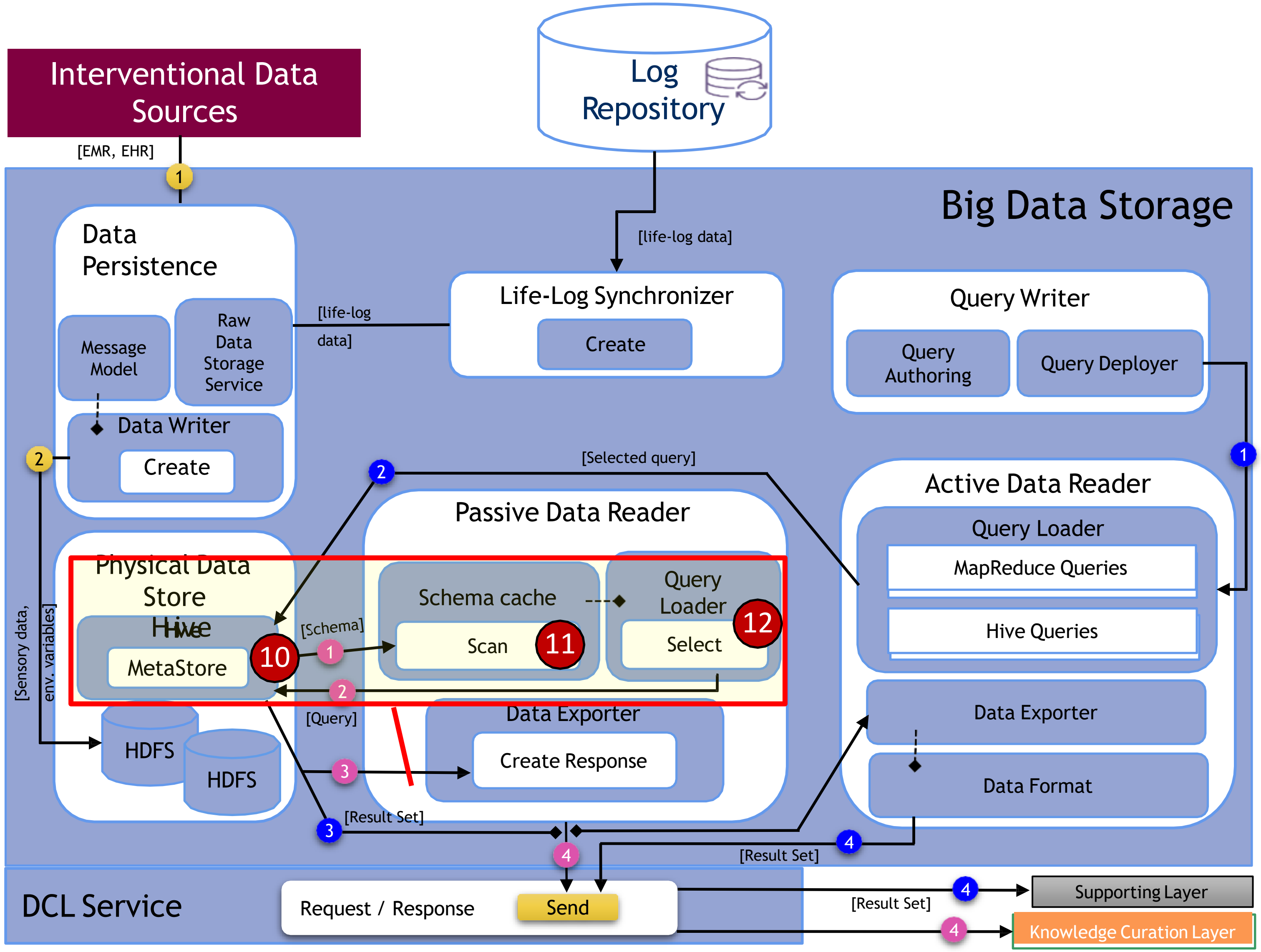


Active Data Reader

8 Requested data is returned as a result set to data exporter

9 Result set is converted into data message per defined data format and send to analytics

Big Data Storage Processing - Execution Flow



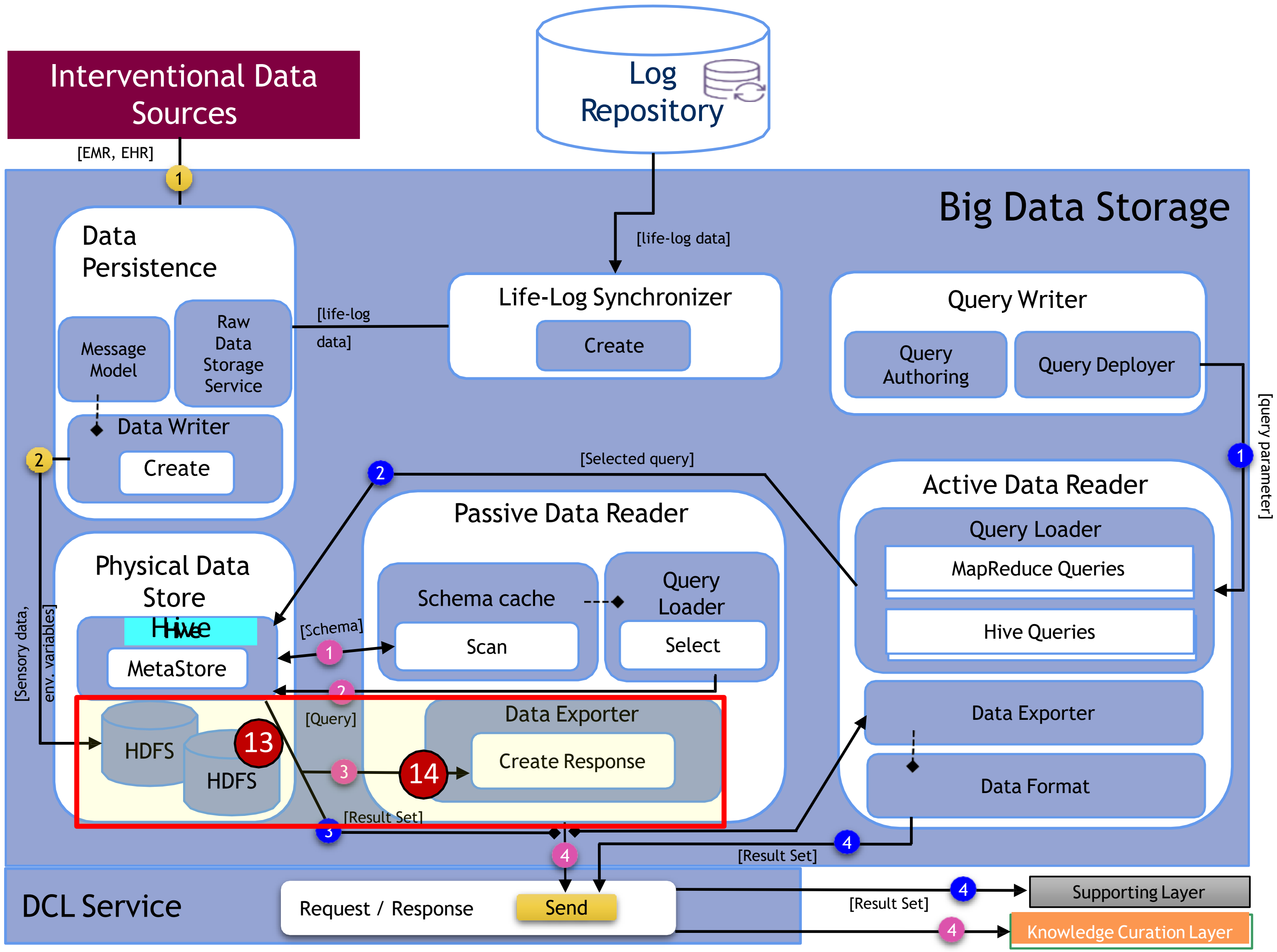
Passive Data Reader

10 Scanned and most updated schema from Non-volatile storage is returned to SKA (Core 1).

11 SKA (Core 1) selects the parameters from the schema to generate a query and submit to Passive Data Reader

12 Passive data reader selects the query and sent to Physical Data Store for execution

Big Data Storage Processing - Execution Flow



Passive Data Reader

13	Required data is returned as a result set to data exporter
14	The result set is returned to Structure Knowledge Acquisition

Tools and Technologies

Platform Libraries

Apache Hive 2.1.1

Apache Hadoop 2.5.2

Service Libraries

Spring MVC

Testing Libraries

JUnit

REST-assured

Tools and Technologies

Library	API Details	License
Hive Thrift Server API	Provide a JDBC based Connections to connect the Hive and Hadoop	Apache 2.0 http://www.apache.org/licenses/LICENSE-2.0.txt
Gson API	Used to convert Java Objects into their JSON representation and JSON to Java Objects	Apache 2.0 http://www.apache.org/licenses/LICENSE-2.0.txt
Spring MVC	Used as a Library to Provider a REST based services on the top of the Hadoop.	Apache 2.0 http://www.apache.org/licenses/LICENSE-2.0.txt
Junit API	JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks	Eclipse Public License 1.0 http://www.eclipse.org/legal/epl-v10.html
REST ASSURED	Provide the REST based end points testing.	Eclipse Public License - v 1.0

Contribution

- Storage of **heterogenous data** at Realtime.
- Stream-based **soft real-time** data read for Analytics and Visualization
- Schema-based query selection and execution over **Big Data Storage**

Benefits

- **Temporal backups** of healthlog data for non-volatile storage
- Able to build the **big data ecosystem** that facilitate request from the other layers.

I- MiningMinds platform and core technology

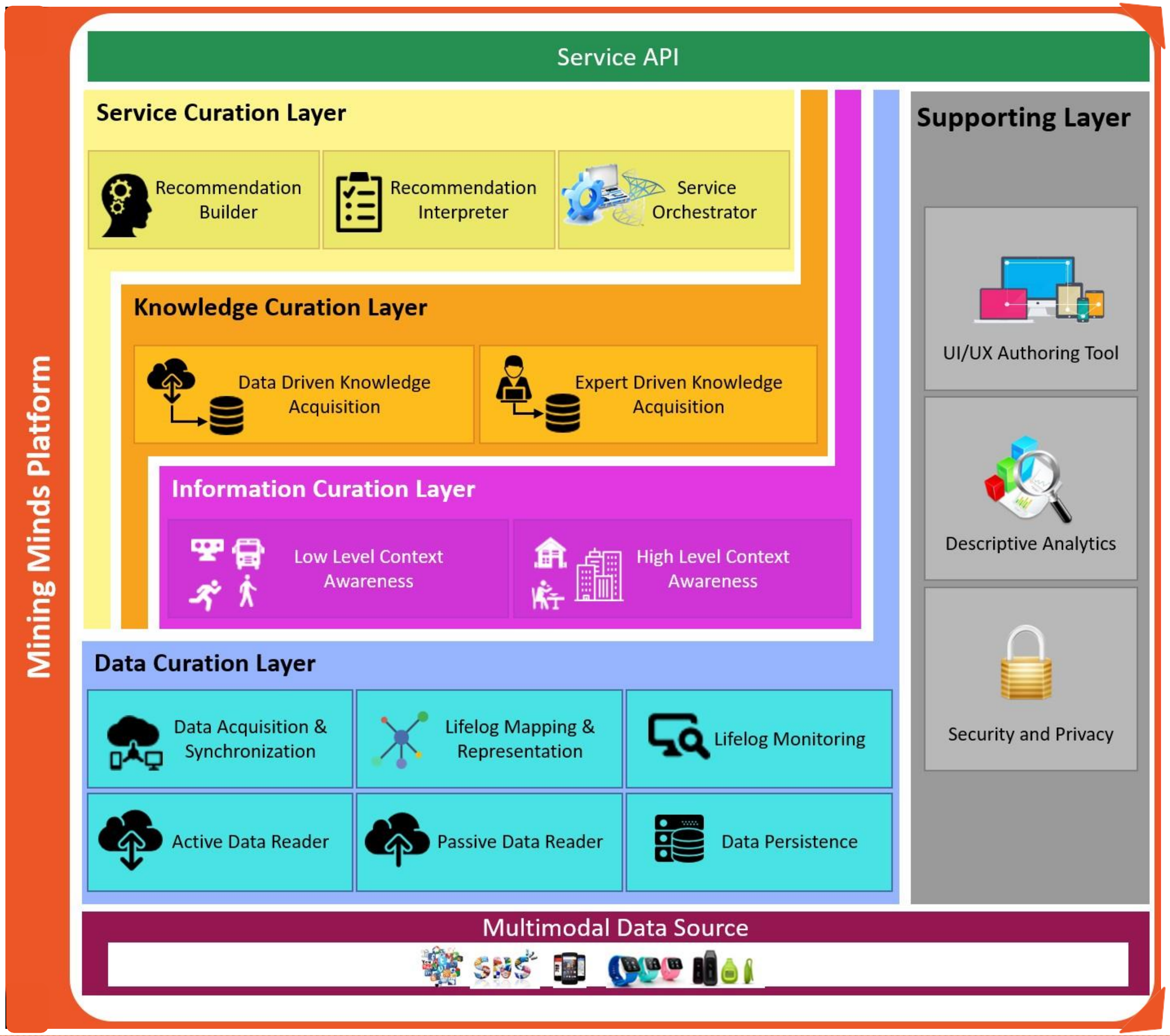
Key Limitations of Existing Digital Health Frameworks

- Most mobile health frameworks are bound to the computational **capabilities of the smartphone**, require continuous **maintenance** and **updates of end-user applications** and normally **trap data** into their devices
- Moreover, multiple systems and applications can be generate similar health data and outcomes leading to **unnecessary redundancy and overcomputation**
- These systems mostly operate on-demand, thus **determinants of health and wellness** states can be also lost if **not registered in a continuous manner**
- Platforms devised to share and integrate health and wellness data **underuse cloud resources**, by only utilizing them for storage

Wellness Platform-Mining Minds

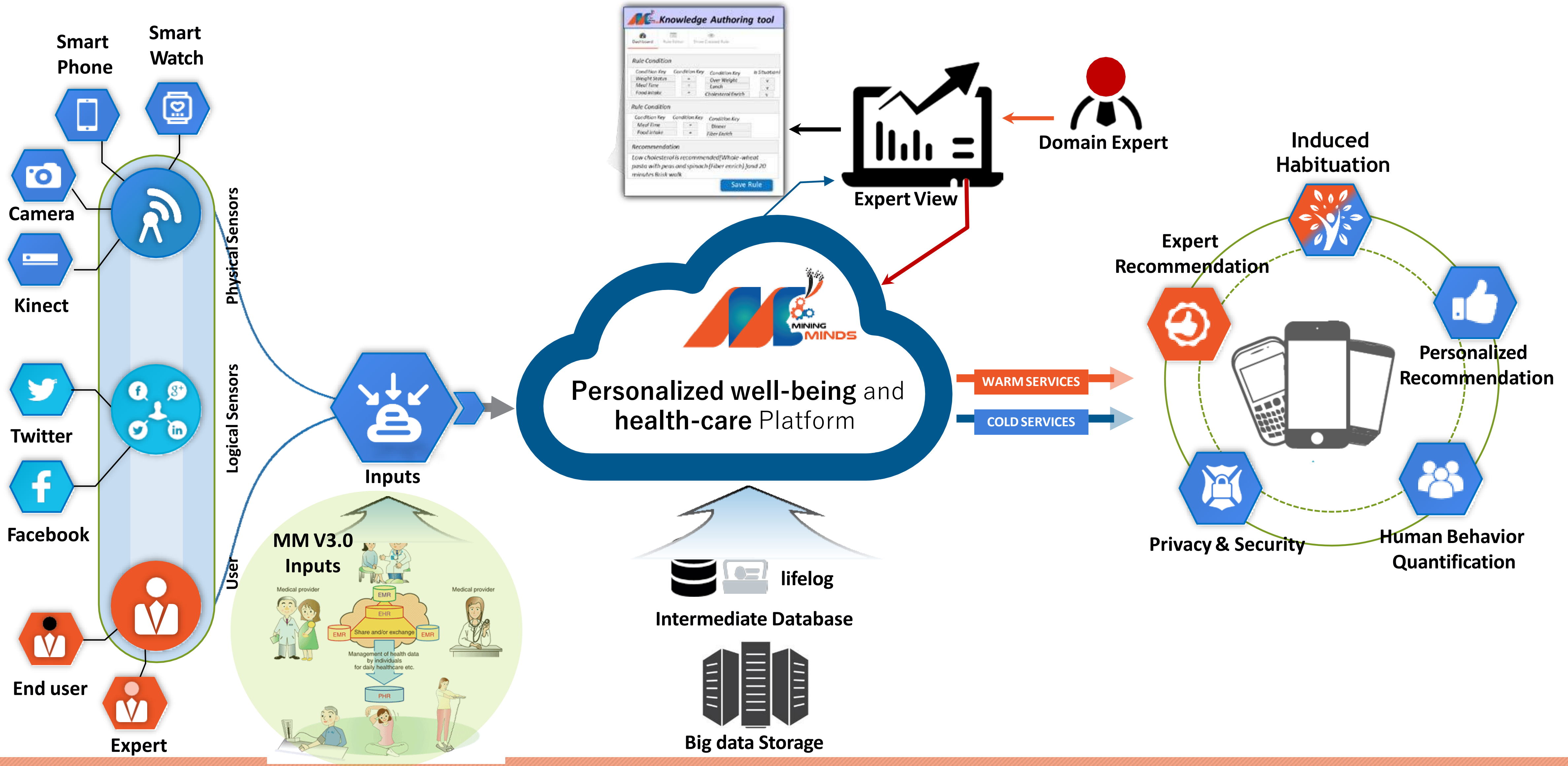
Mining Minds is a novel platform aimed at comprehensively mining human's daily life data generated from heterogeneous resources for producing personalized health and wellness support.

Mining Minds philosophy revolves around the concepts of **data**, **information**, **knowledge** and **service curation**, which refer to the discovery, processing, adaptation and evolution of both contents and mechanisms for the provision of high quality **support** services.

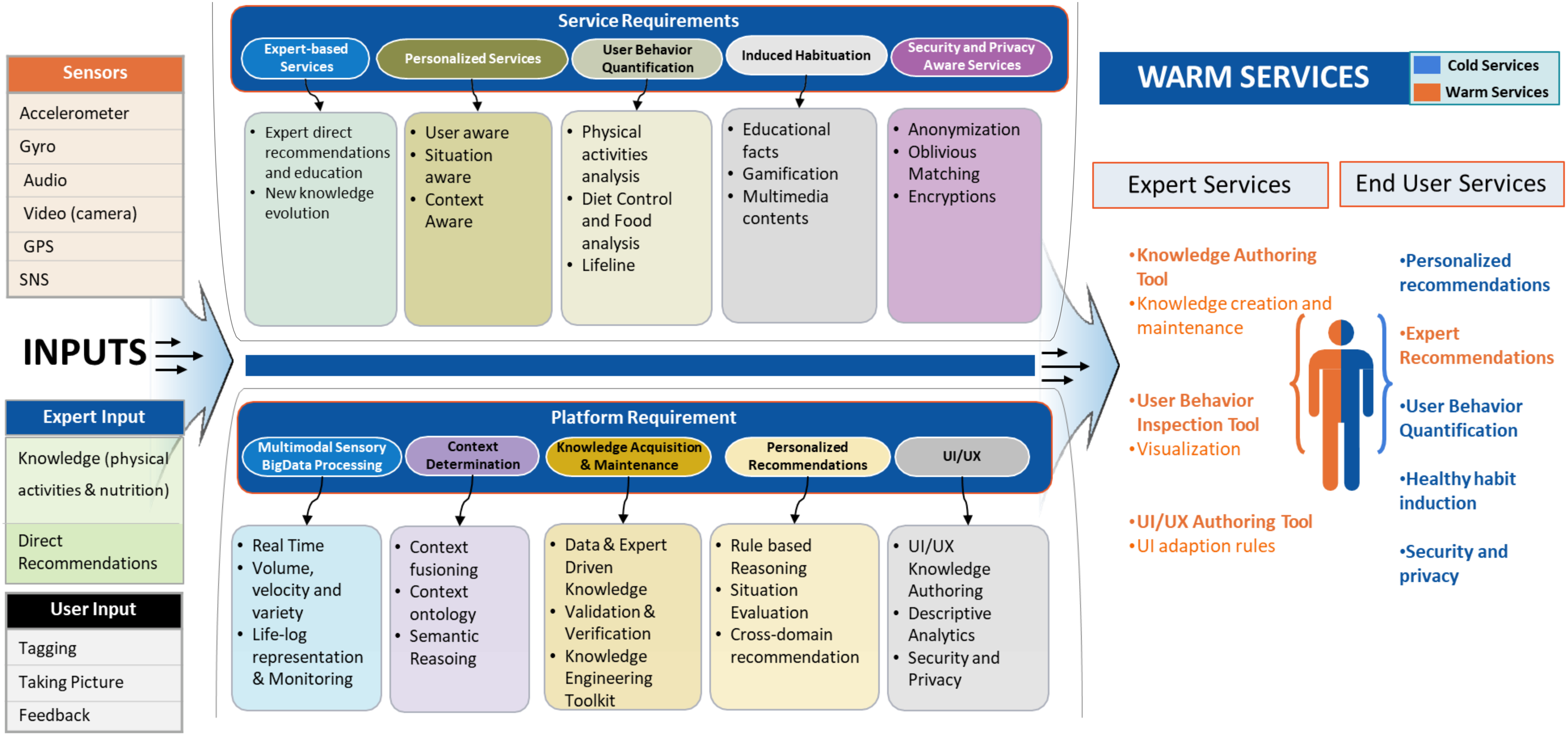


http://www.miningminds.re.kr/

Mining Minds Services – The Big Picture



Mining Minds Platform and Services



Overall Technical Uniqueness

Google Fit

SAMSUNG Digital Health

Microsoft Health

HealthKit

noom coach

Push Service Model

SNS & Lifelog based Big Data Analytics

Multimodal Heterogeneous Data Acquisition

User Experience Quantification

Low Level Context Fusion

Knowledge based Recommendations

Automatic High Level Context Awareness

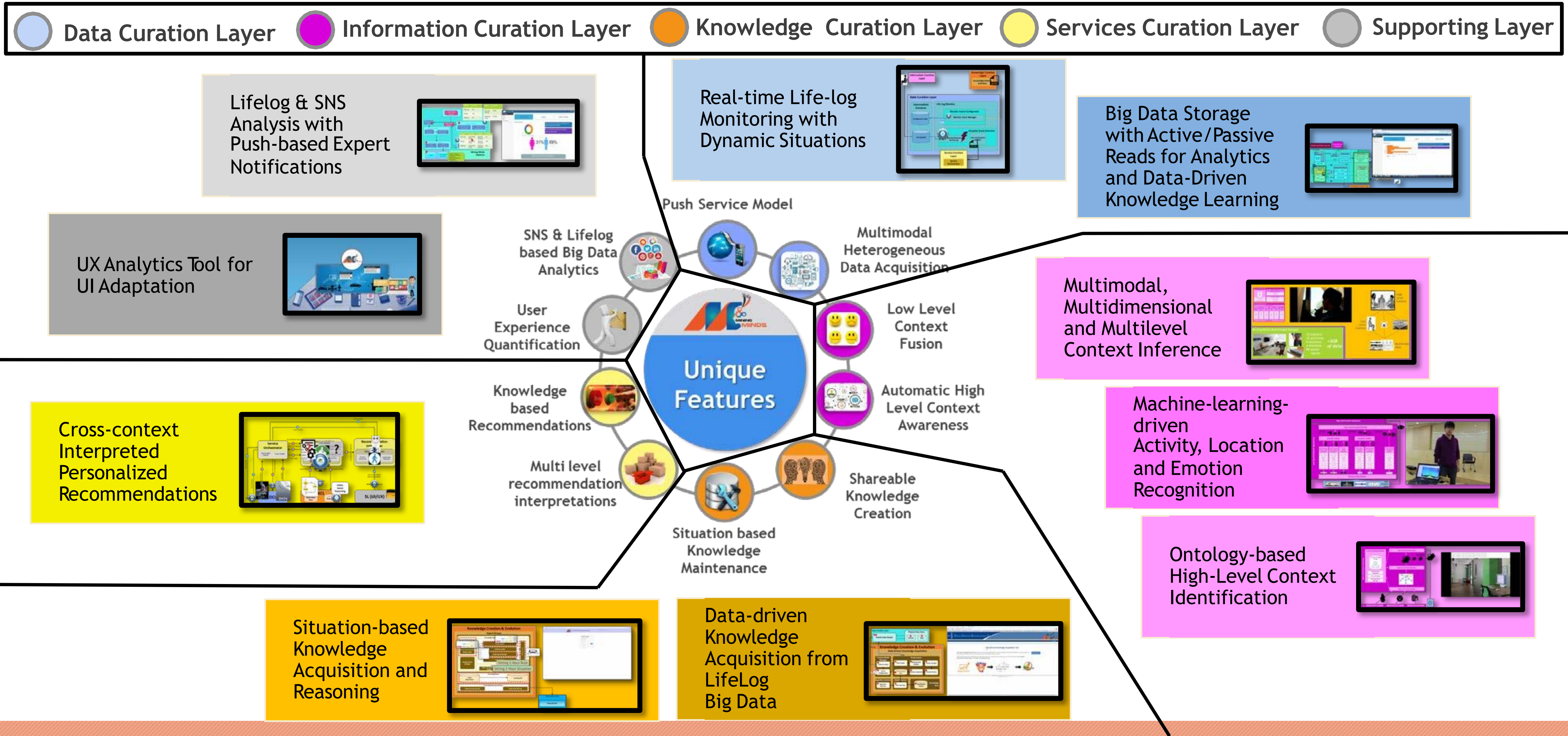
Multi level recommendation interpretations

Shareable Knowledge Creation

Situation based Knowledge Maintenance

- Supporting Layer
- Services Curation Layer
- Knowledge Curation Layer
- Information Curation Layer
- Data Curation Layer

Technical Contribution



Comparison with Existing Systems

Platforms

- ❑ Google Fit
- ❑ Samsung S Health
- ❑ Microsoft Health
- ❑ Apple Health Kit
- ❑ Open mHealth

Applications

- ❑ NoomCoach
- ❑ Fitbit
- ❑ Argus
- ❑ Runtastic
- ❑ RunKeeper
- ❑ ZombieRun

Services

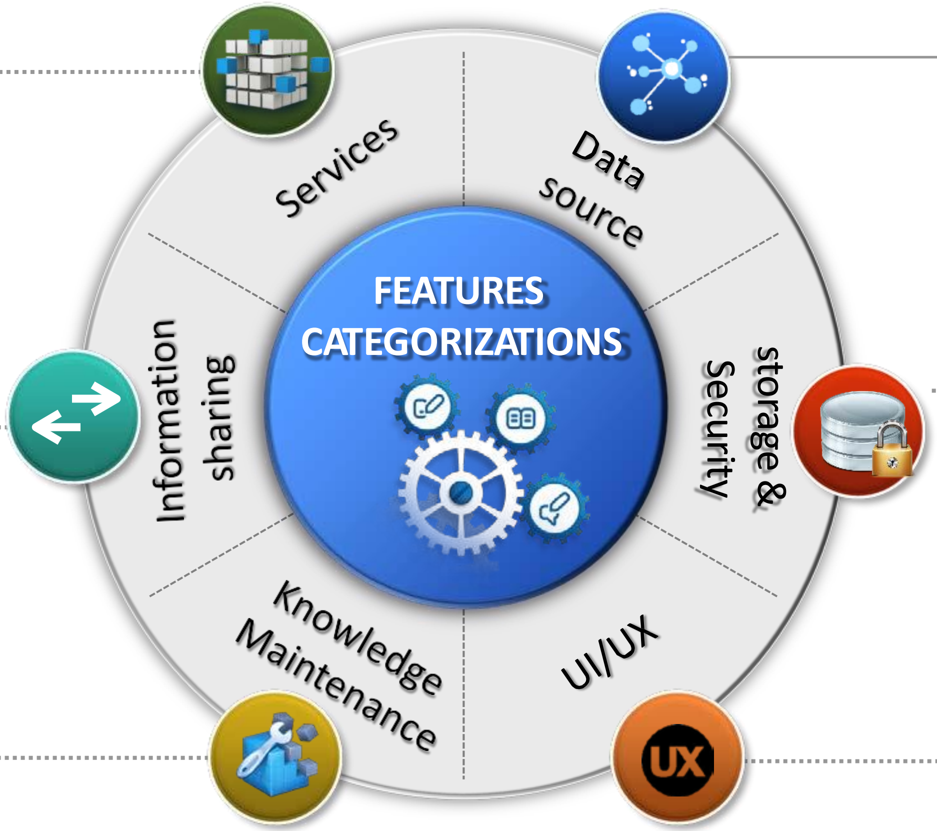
1. Activity Recognition
2. Expert Services
3. Wellness services
4. Personalized recommendations
5. Clinical services
6. SDK/API

Information Sharing

1. With other apps
2. Social media sharing
3. Other users (authorized circle)

Knowledge Maintenace

1. Open knowledge
2. Knowledge acquisition
3. Knowledge evolution



Data Sources

1. Sensory Data
2. User profile
3. IOT
4. Other apps
5. Clinical data
6. Social media

Storage & Security

1. User device storage
2. Cloud storage
3. Bigdata storage
4. Encrypted data storage
5. Anonymized access

UI/UX

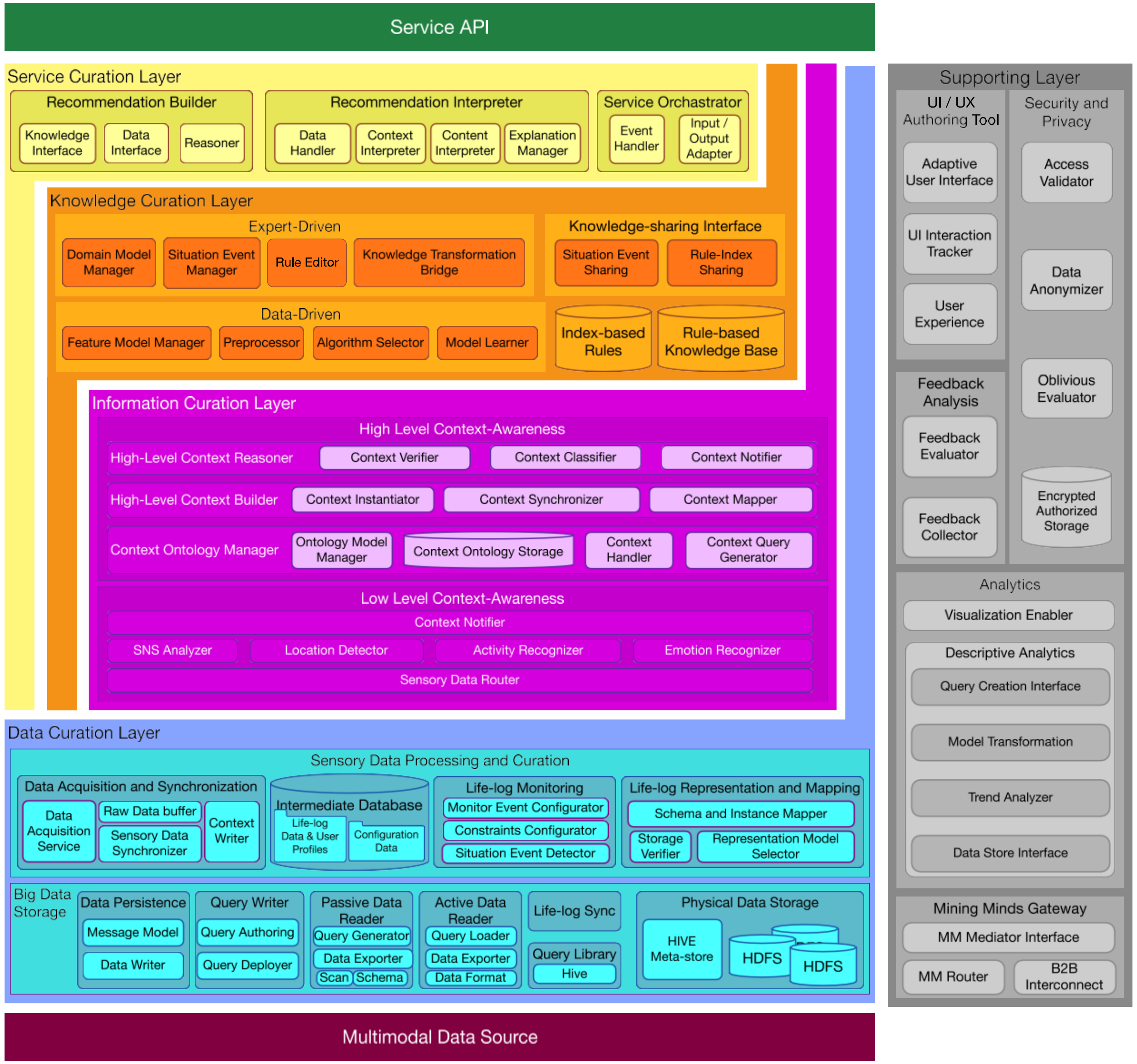
1. User Experience
2. User Modelling
3. Adaptation of UI

Re-usability of Mining Minds Platform

Domain Dependent

Domain Transition

Domain Independent



Domain-Centric

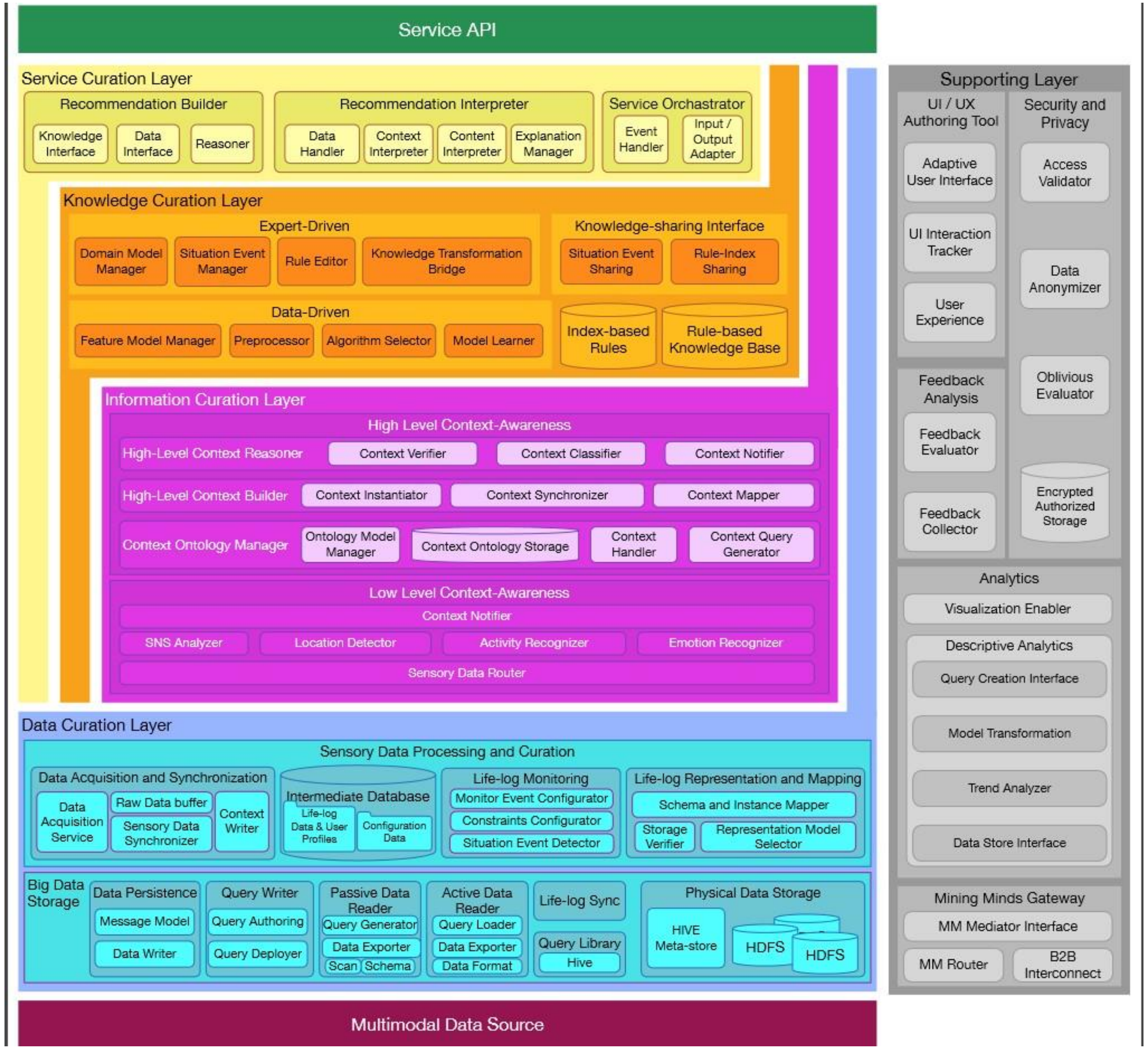
Layered Architecture

Delivers timely and accurate personalized cross-domain recommendation based on domain knowledge and users preferences/context

Creates and maintains health and wellness knowledge using expert-driven and data-driven approaches

Converts the data obtained from the user interaction with the real and cyberworld, into abstract concepts or categories, such as physical activities, emotional states, locations and social patterns, which are intelligently combined to determine and track context and behavior

Provides real-time data acquisition from multimodal data sources and its persistence using big data technologies. Context data are mapped for life-logging and personalized predictions from life-log



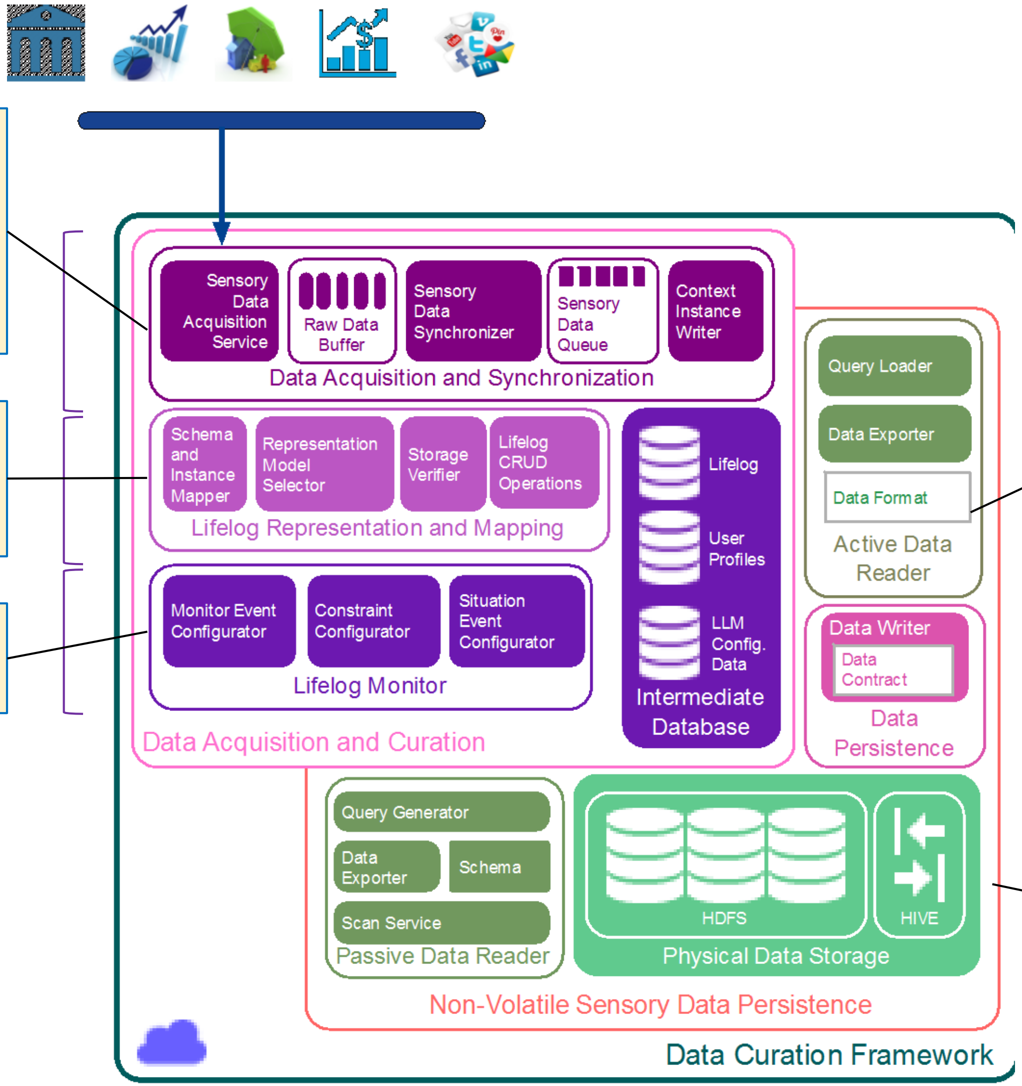
Facilitates information to the users in the most intuitive manner, in a secure environment reflecting their personal needs and preferences

Reusability of Mining Minds Platform : DCL

- Data source independent data acquisition
- Synchronization will be trivial as most of the transactional financial data is timestamped

- Lifelog will be context defined, as context can be user spending, user transactions etc.

- Lifelog monitoring will be goal and rule driven as per definition in KCL

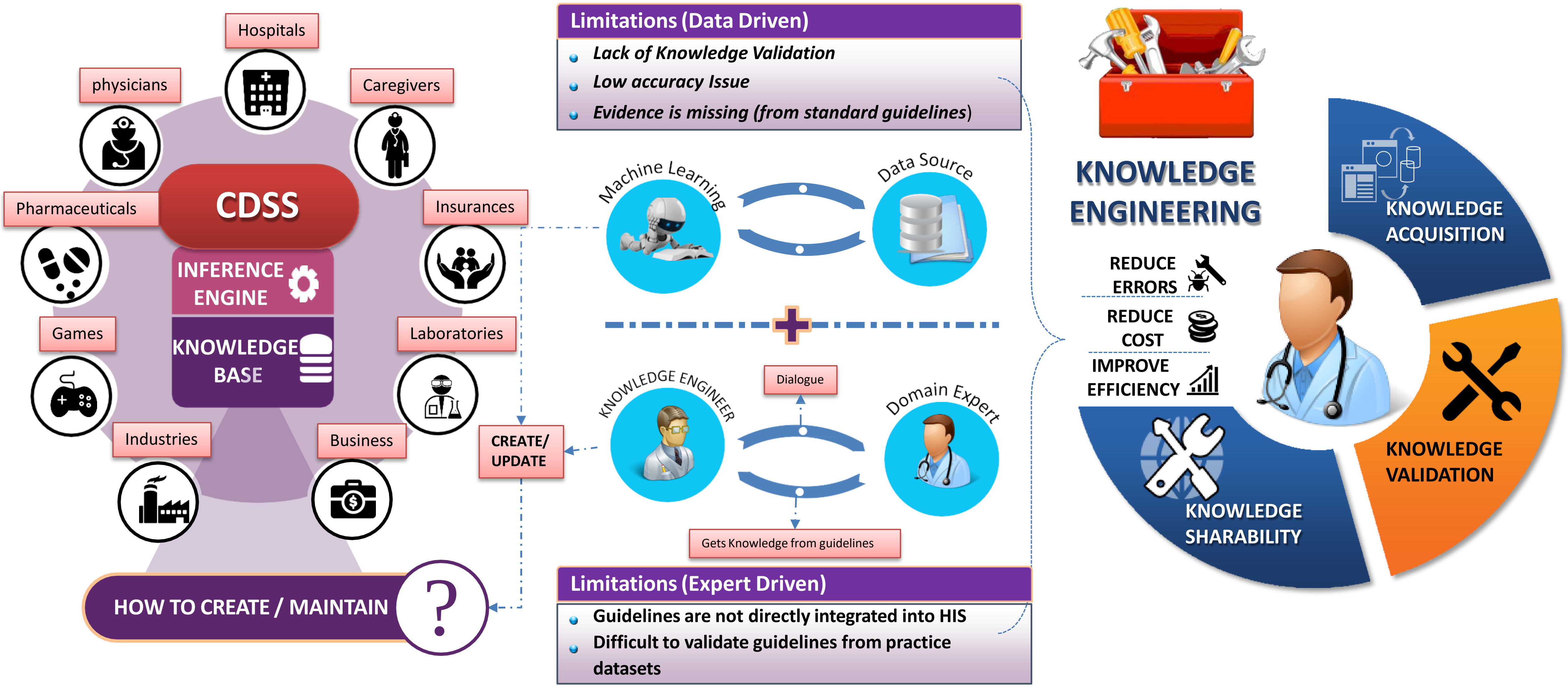


- Accumulated financial data for real-time visualization

- Accumulated financial data for offline data driven rule derivation for KCL

II - Intelligent Medical platform (IMP)

Motivation


Hospitals
physicians
Caregivers
Pharmaceuticals
CDSS
Insurances
INFERENCE ENGINE
KNOWLEDGE BASE
Games
Laboratories
Industries
Business
CREATE/ UPDATE
HOW TO CREATE / MAINTAIN ?
Limitations (Data Driven)
Lack of Knowledge Validation
Low accuracy Issue
Evidence is missing (from standard guidelines)
Machine Learning
Data Source
KNOWLEDGE ENGINEER
Dialogue
Domain Expert
Gets Knowledge from guidelines
Limitations (Expert Driven)
Guidelines are not directly integrated into HIS
Difficult to validate guidelines from practice datasets
KNOWLEDGE ENGINEERING
REDUCE ERRORS
REDUCE COST
IMPROVE EFFICIENCY
KNOWLEDGE ACQUISITION
KNOWLEDGE VALIDATION
KNOWLEDGE SHARABILITY

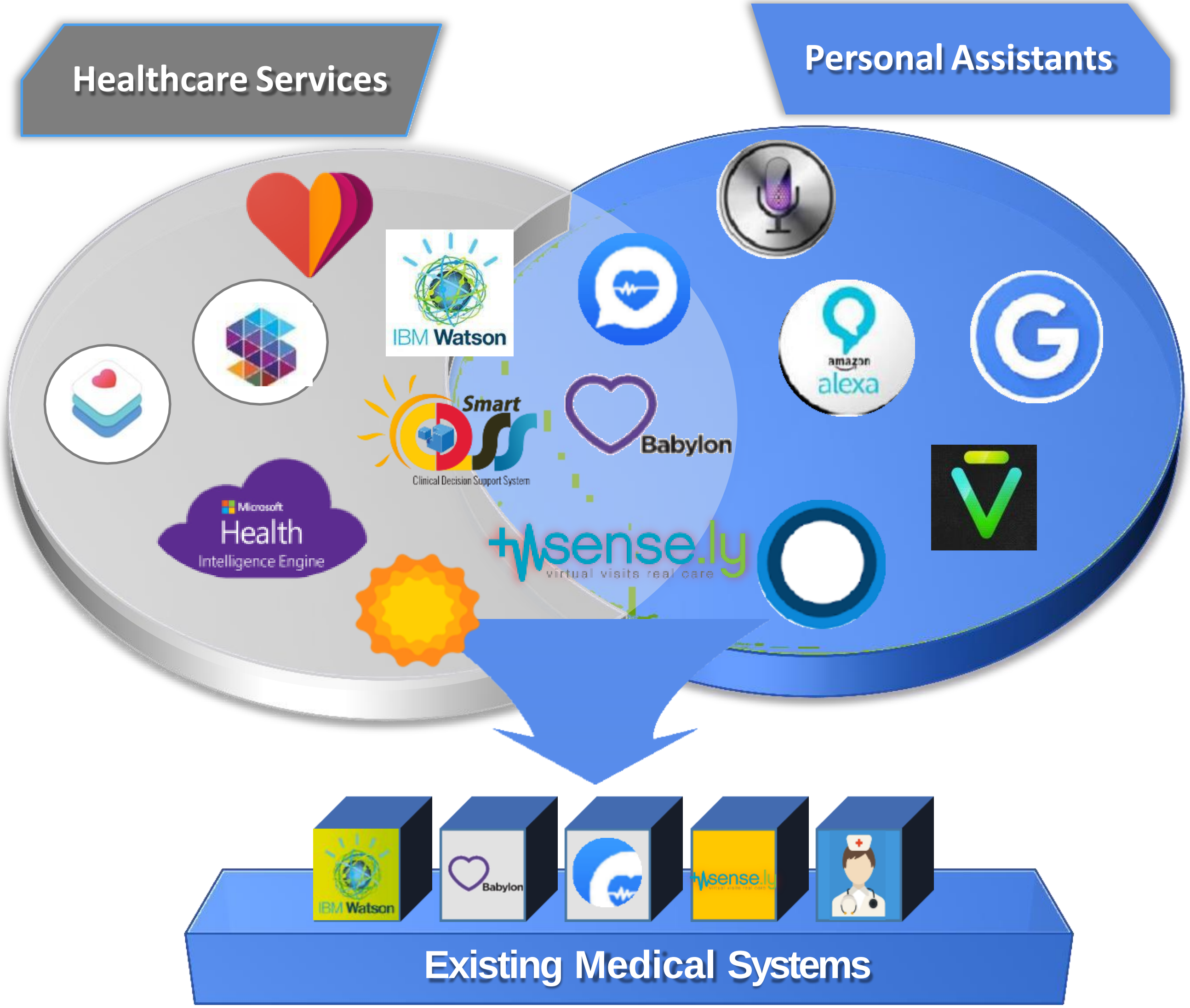
Healthcare Services vs Personal Assistants

Strength

- ❑ Actionable Knowledge
- ❑ Empowering the expert user
- ❑ Evidence support for domain experts

Weakness

- ❑ Non-Interactive
- ❑ Rigid Knowledge Structures
- ❑ Lack Context-awareness



Strength

- ❑ Highly Interactive
- ❑ Natural User Interfaces
- ❑ User Engagement

Weakness

- ❑ No Actionable Knowledge
- ❑ No Concept of appraised evidence support
- ❑ Limited Visual Understanding

Limitations and Solutions

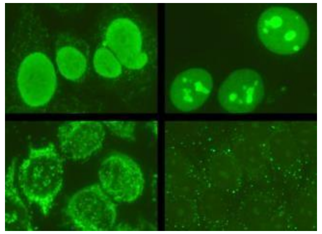
Existing medical support system

INTELLIGENT MEDICAL PLATFORM

Limitations	Solutions
Limitations of Knowledge Acquisition Methods	(Data Driven + Human Driven) Knowledge Acquisition
Difficult to construct knowledge by domain experts	Support Knowledge Authoring Tools
Minimal Evidence Support	Evidence Support from PubMed
Lack of User Interaction	Support UI/UX Tools
Lack of Interoperability for Heterogeneous HIS	Standard based Interoperability

Intelligent Medical Platform

Clinical Decision Support Systems



Clinical diagnosis

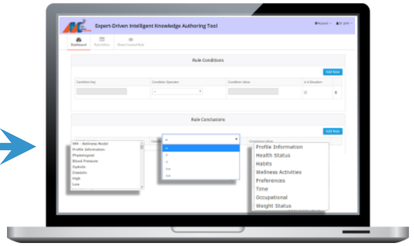
Patient education

Intelligent communication

Engineering Support

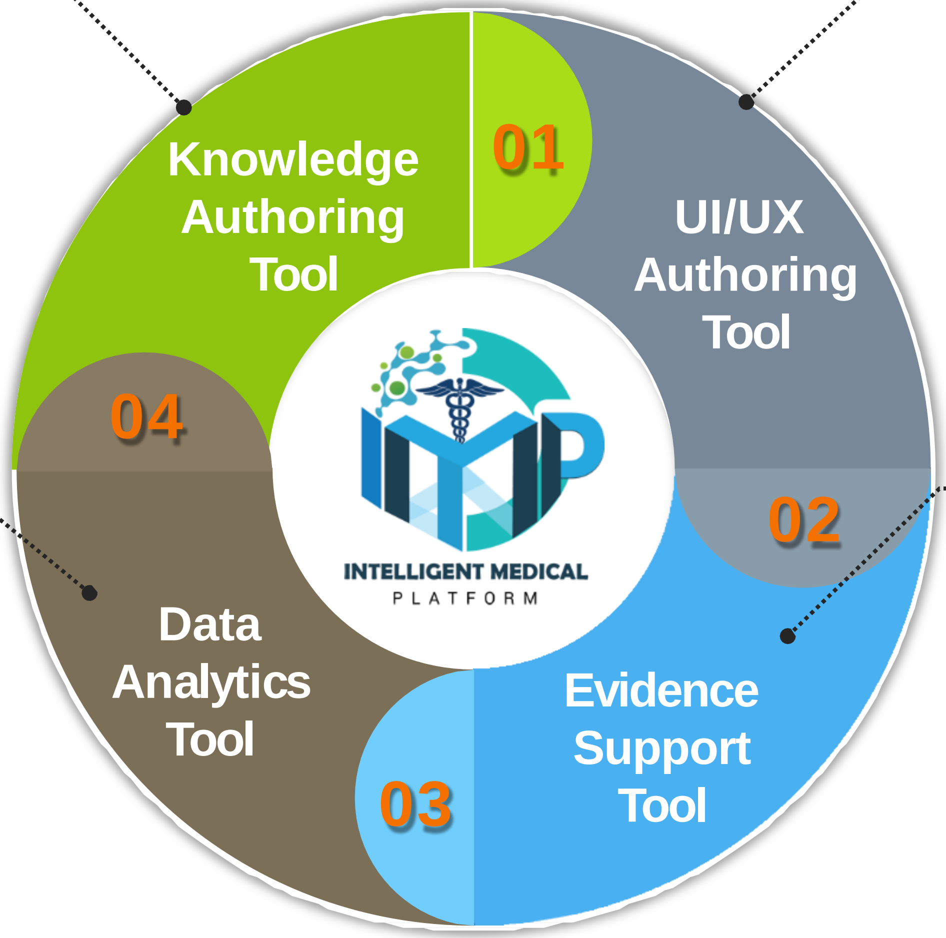
- Incremental Learning-based validation and verification
- Intelli-sense support

Physician



- Overall UX quantification over time
- Adaptive UI based on UX

UX Expert

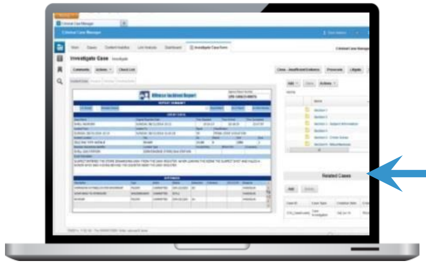


- Real time monitoring
- Health-log visualization

Patient

Physician

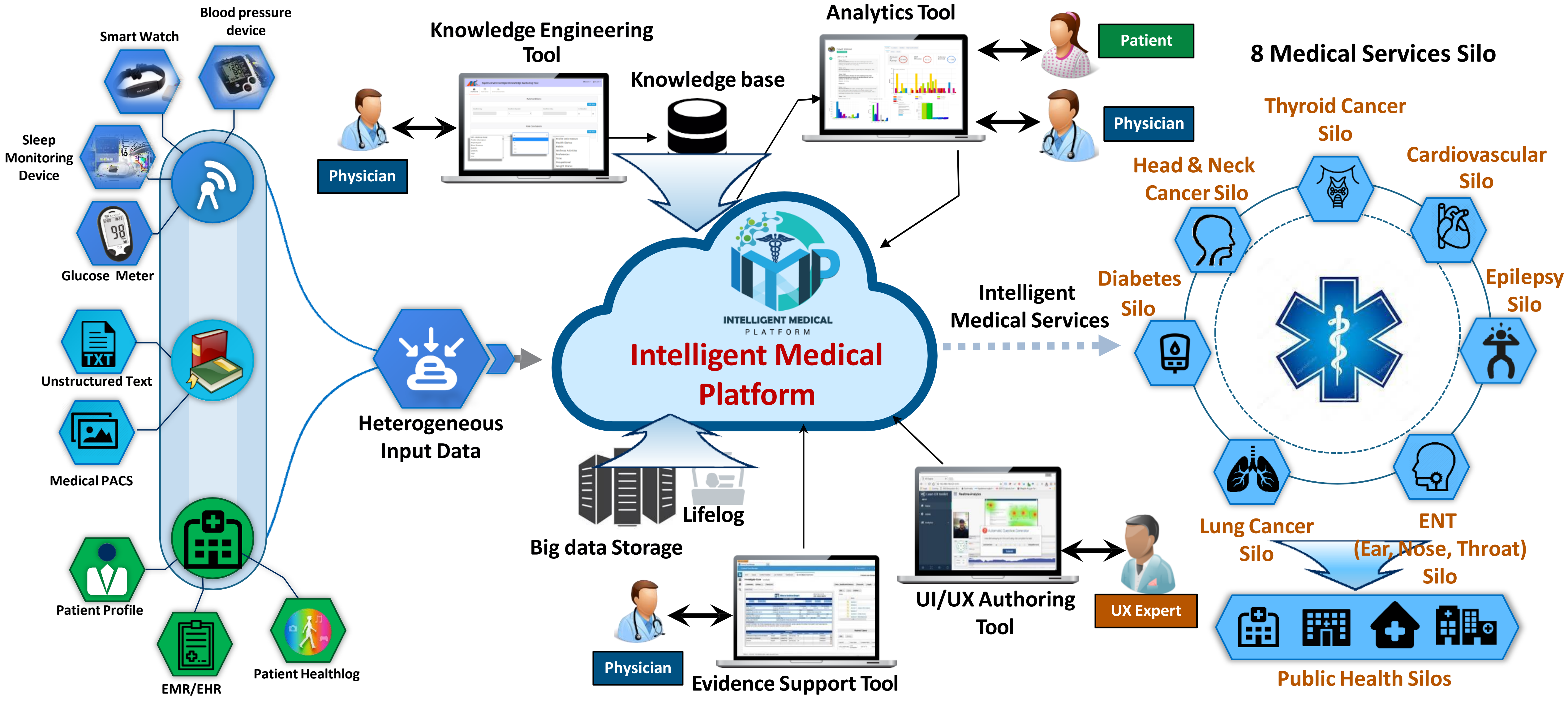
- Evidence support form PubMed
- Quality assessment retrieved documents



Physician

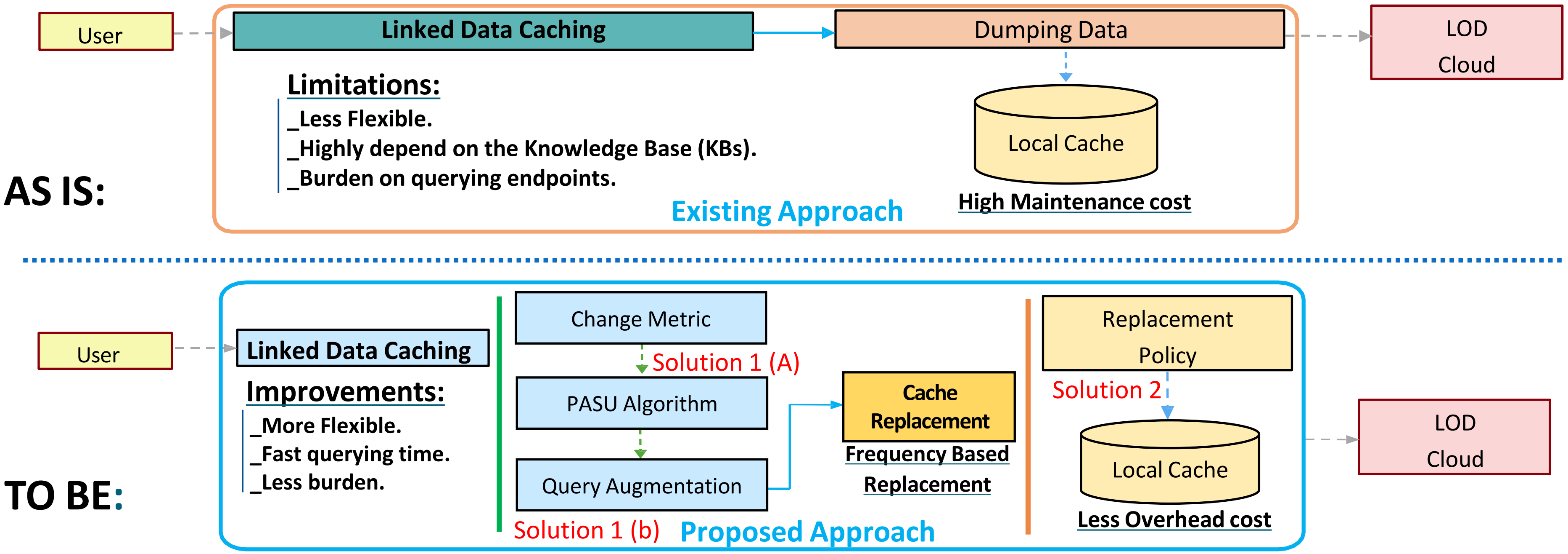
Easy to commercialize by providing development environment

Intelligent Medical Platform (IMP) for AI Doctor



Ph.D. Dissertation Work

❑ **Ph.D. Dissertation: A Cache Based Method to Improve Query Performance of Linked Open Data Cloud**



Total Publications (27)

First Author Publications (18)

- **Thesis Contributions**
  - Comprehensively utilize client-side Linked Data caching for better query performance.
    - **Solution 1(a):** Proposed change metric to quantify the evolution of Linked Open Data.
    - **Solution 1(b):** Proposed query augmentation to alleviate the burden on server.
    - **Solution 2:** Proposed frequency-based cache replacement to replace less valuable Cache Items.

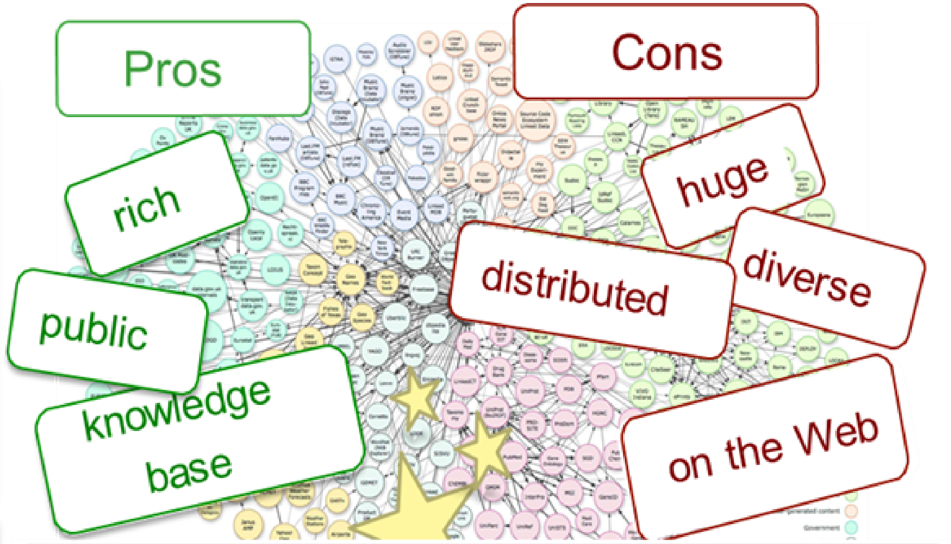
Background

Linked Data Dynamics

- Linked Open Data Cloud (LOD) is a distributed knowledge base on the web that handles a large number of requests from applications consuming these data [1,2].
- Understanding the evolution of the Linked Data Cloud (LOD) is important for applications [5,6].
  - e.g., Query Caching, Web Crawling, and knowledge graph search engines.

Querying

- Traditional ways of querying LOD are as follows:
  - Data Dumps [6].
  - Querying endpoints [7].



LOD: a rich, huge, diverse, public and distributed knowledge base on the Web.

Data Dumps

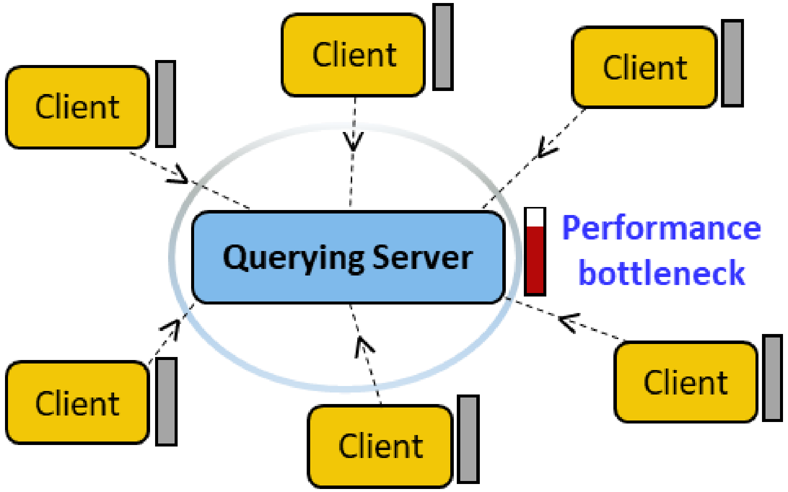
Cons: Dump the data locally and allows to setup own private querying endpoints.

- Out-of-date data
- No longer query the web
- Infrastructure cost

Querying endpoints

Cons: Public endpoints are often unreliable.

- Low availability (Downtime)
- High querying cost
- Hosting endpoints are expensive



Motivation

- To unlock the full potential of Linked data sources, we need flexible ways to query them [8].
  - Following benefits drive our research:
    - High-availability to the server.
    - High query performance

Limitations	Benefits	Limitations
- Low availability - Out-dated data - No longer query the web.	+ High availability + Decentralized caching + Low server cost	- Low availability - Hosting endpoints are expensive - High client cost
Data dumps *(Existing)*	**Proposed** *(Caching)*	**SPARQL endpoint** *(Existing)*

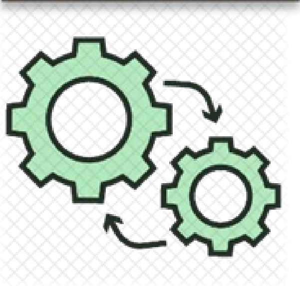
Linked Data Caching

(Existing) Server-side caching

+ Server manages request from cache
+ Dependent on the database
- Not flexible approach
- High server cost
- Low availability

(Proposed) Client-side caching

Methodology



I **Query Prefetching** — **A** **Similar query requests are served from cache**

I **Decentralized caching** — **A** **Higher hit rates**

I Idea A Advantage

Comparison with Existing Work

Categories	Methodologies	Existing work problems: Advantages	Existing work problems: Method	Existing work problems: Limitation	Overheads
Query Similarity & Prefetching (Challenge 1&2)	[SQC] Improving the performance of semantic web applications with SPARQL query caching [16]	• Cache complete triples query results. • Introduce a proxy layer to cache repeated query results	Structure based similarity	**Server side caching** Only consider repeated queries.	**High**
	[PFU] Proactive Policy for Efficiently Updating Join Views on Continuous Queries Over Data Streams and Linked Data [11]	• Proposed maintenance policy that update the cache prior to query execution	Content based similarity	**Server-side caching** Only update the local cache at system idle time.	**High**
	[CIR] Caching intermediate result of SPARQL queries [17]	• Adaptive cache to store intermediate results of SPARQL queries	Result based similarity	**Client-side caching** No cache replacement policy is introduced.	**High**
	[SDC] Semantic data caching and replacement [18]	• Proposed a semantic region-based caching and a distance measure to update cache	Distance based similarity	**Server-side caching** Only considered the structure similarity while creating a semantic region.	**High**
	[CAS] Towards content aware SPARQL caching for semantic web application [19]	• Introduced a query containment which evaluated whether a query can be answered from cache or not.	Content based similarity	**Client-side caching** Containment checking is computationally expensive task.	**High**
Cache Replacement (Challenge 3)	[GAW] Graph-aware, workload-adaptive SPARQL query caching [20]	• Work-load adaptive caching to reduce the SPARQL query response time	Result based similarity	**Server-side caching** Time based cache replacement	**High**
	[Autosparql] Let user query your knowledge base [21]	• Proposed machine learning approach to leverage the query processing.	Structure based similarity	**Server-Side Caching** The feature modeling approach in their work is time consuming	**High**
Query Similarity, prefetching & Cache Replacement	Proposed method	O (Alleviate burden on querying endpoints by identifying queries learnt from client historical patterns)	O (Distance based similarity & Frequency based cache replacement)	O (Local data Cache need to be updated during system idle time)	**Low**

Thesis Map

User

Proposed Solutions

Solution 1a		Solution 1b	Solution 2
Change Metric	Algorithm 1 & 2 AACP Algorithm PASU Algorithm	Algorithm 3 Query Augmentation	Algorithm 4 Cache Replacement
Problem: Linked Data Evolution **Result:** Quantify the changes occurs in the cloud such as addition and deletion of items	**Problem:** Change Prioritization **Result:** Identify the recent changes occurs in the cloud using weight function	**Problem:** Similar structure queries **Result:** Prefetches the result of similar queries and placed in cache for future queries	**Problem:** Replace cache **Result:** Frequency based cache replacement for replacing less valuable cache items

LOD Cloud

Outcomes

+ Low overheads
+ High Query performance

Challenges

C1: Linked data evolution

C 2: Query Similarity

C 3: Cache Replacement

Challenge 1, 2		Challenge 3	
Existing Approach – Unable to quantify the evolution of LOD. – Less flexible – Dump all the data locally.	**Existing Approach** – Unable to prioritize recent changes occurs in the cloud – Issued irrelevant queries – Performance bottleneck	**Existing Approach** – Only consider the structure similarity. – Aggravate the burden on query endpoint by issuing similar structure queries.	**Existing Approach** – The cost of cache maintenance is high – Introduced the time-based cache replacement – High overhead

Outcomes

– High Overheads
– Low query performance

Current Research Activities

❑ Exploring Computational Complexity of Ride-Pooling Problems

We report the computational complexity of real-world ride-pooling problems & trace the:

- search space sizes,
- computation times,
- ride-pooling performance,
- and properties of underlying shareability graphs.

To overcome the curse of dimensionality for real-size demand patterns, the utility-driven search space method is applied to effectively explore only attractive shared rides and avoid unnecessary searches for acceptable computation time

$$\Delta U = U^s - U^{ns} = \beta^c \lambda l + \beta^t \left(t - \beta^s \left(t^s + \beta^d t^d\right)\right)$$

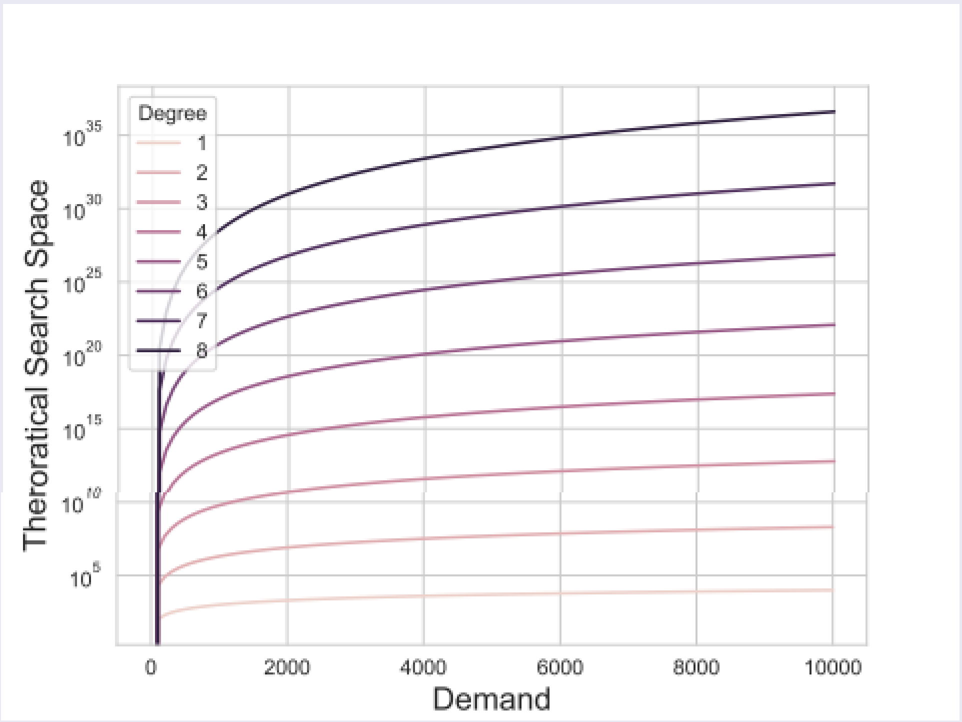
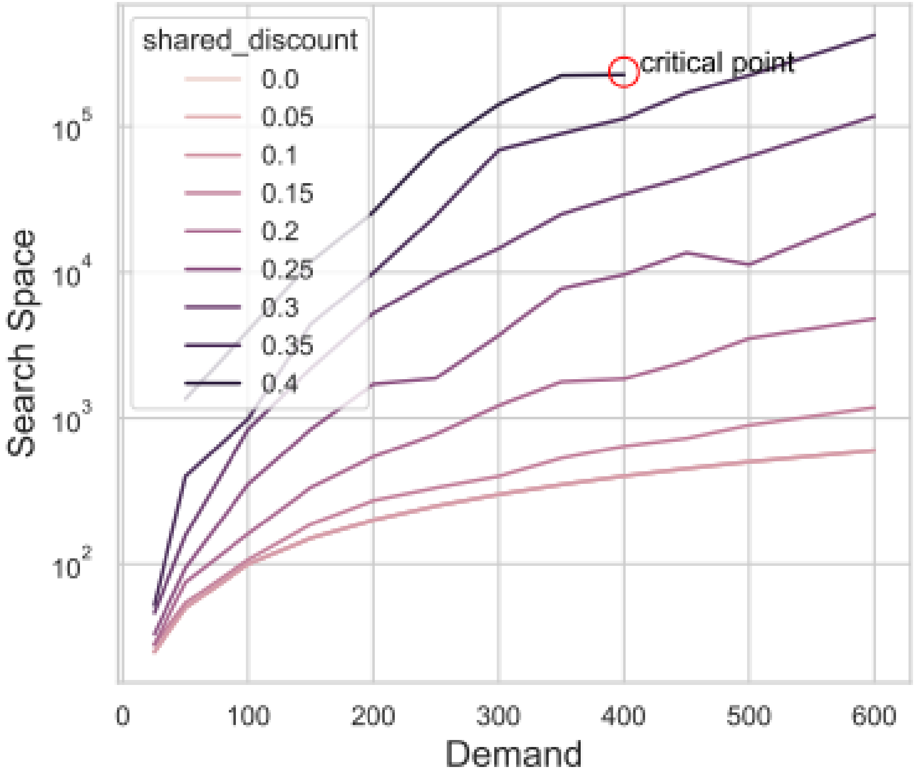


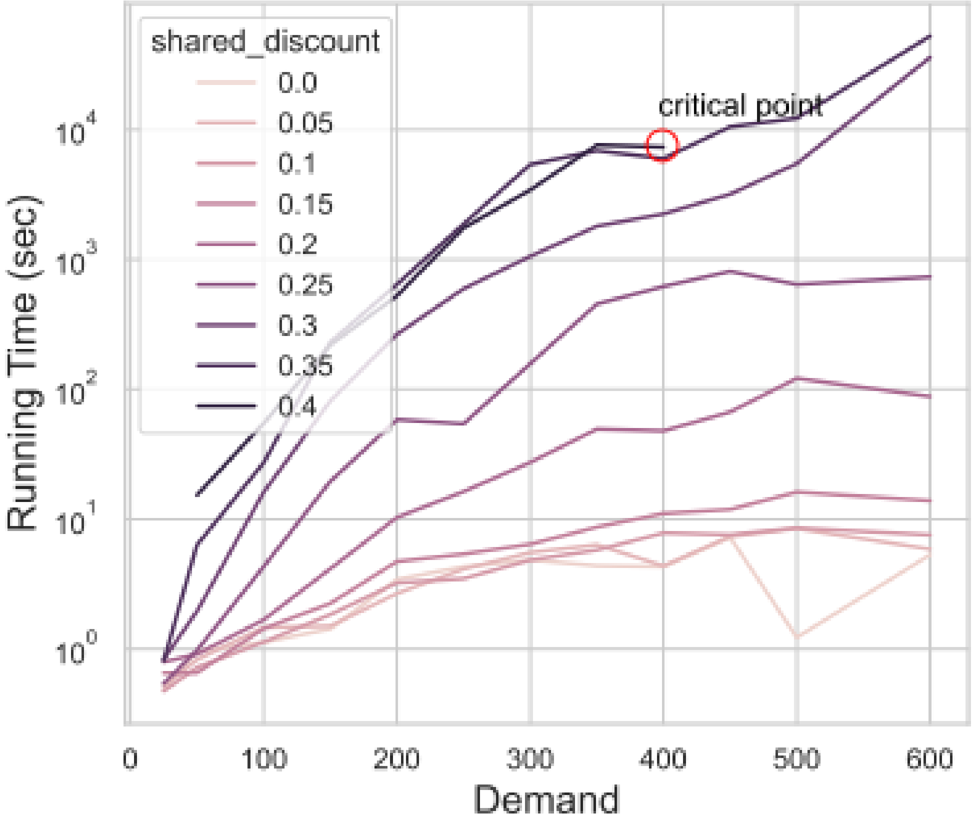
Figure: Theoretically computed search space of ride-pooling problems

Accepted*:* Exploring Computational Complexity of Ride pooling Problems TRB Annual Meeting, January 8-12, 2023, in Washington, DC

Ride-pooling problem complexity



(a) Number of feasible rides explored and (search-space) increases with the demand levels (x-axis) and the discount offered (line colours).



(b) Running time needed to solve ride-pooling problems. It grows significantly with the demand size (x-axis), yet much sharper growth is visible with increasing the discount (line colours).

The shared discount has the greatest impact on the running time and search space, until it reaches a critical point and from the computational complexity perspective it becomes intractable.

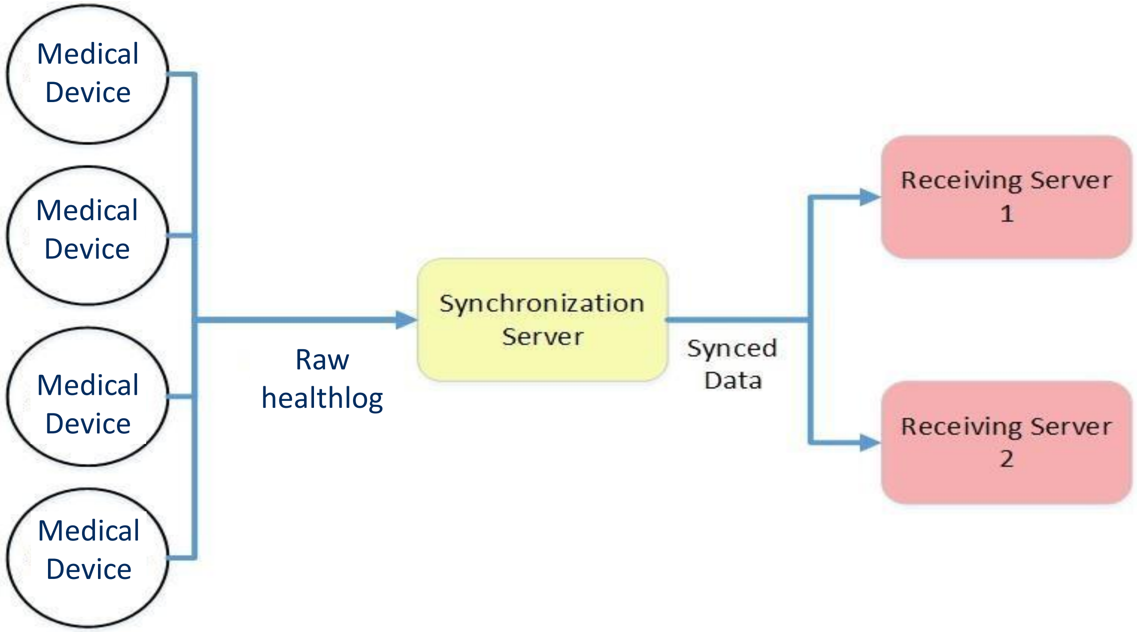
Thank you

Appendix

III - Lean UX Platform For User Experience Evaluation of any Digital Artifacts

Evaluation Environment

- Scenario
  - 4 medical devices sensors sending data to Synchronization Server
    - Initial sending period: randomly inside a window size.
    - Window size: 3 seconds.
    - Data size: randomly.
    - File attached: yes, 1KB.
  - Procedure
    - Synchronization Server syncs data before sending to 2 different receiving servers.
  - Receiving server
    - Implemented in nodeJS.
    - Receive synced json data, then parse it.
- Purspose: check that whether the synchronization works properly after refractoring or not.



Evaluation results

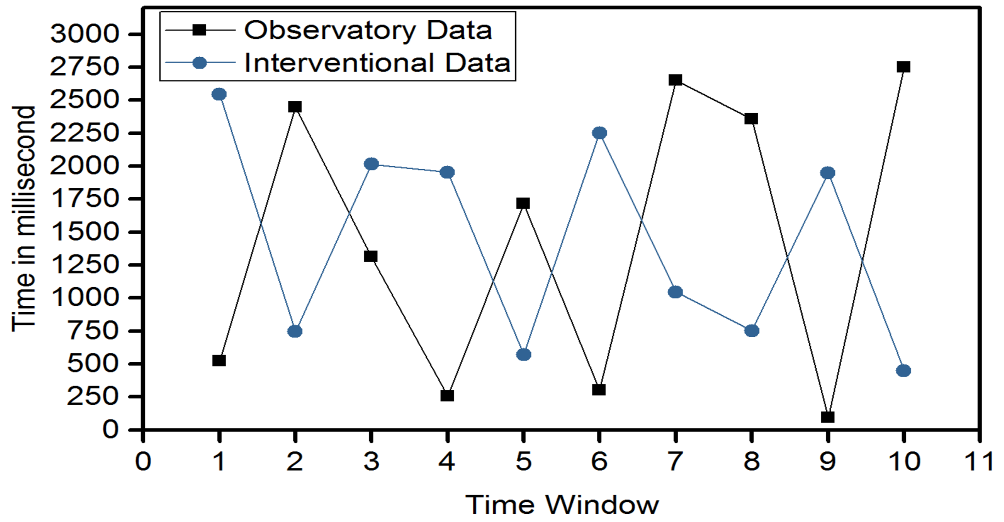


Fig 1. Showing the Data synchronization testing per time-window

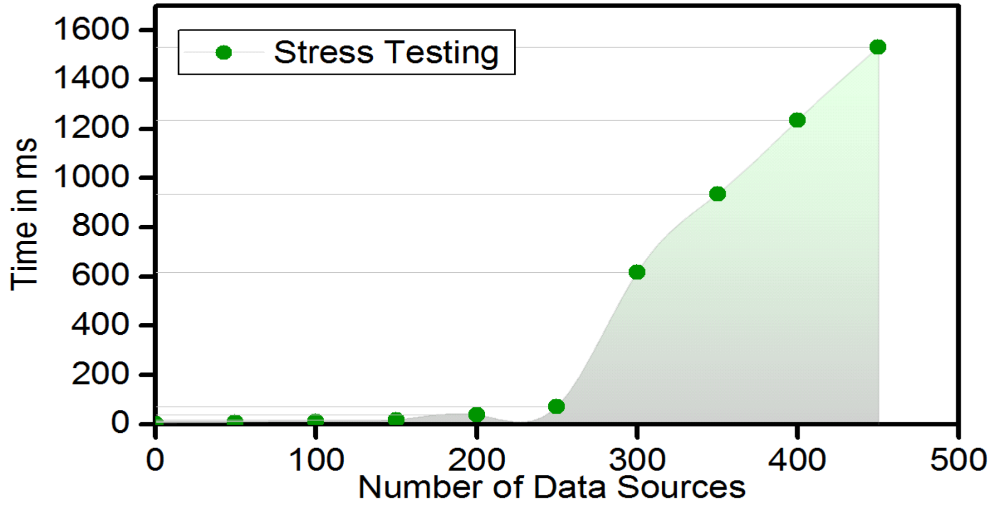


Fig 2. Stress Testing of Scalability

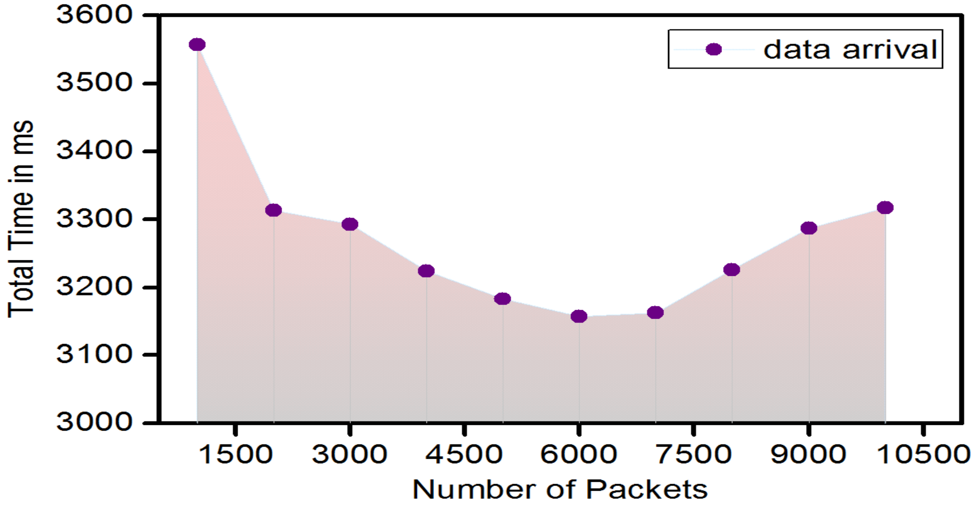


Fig 3. Performance Testing

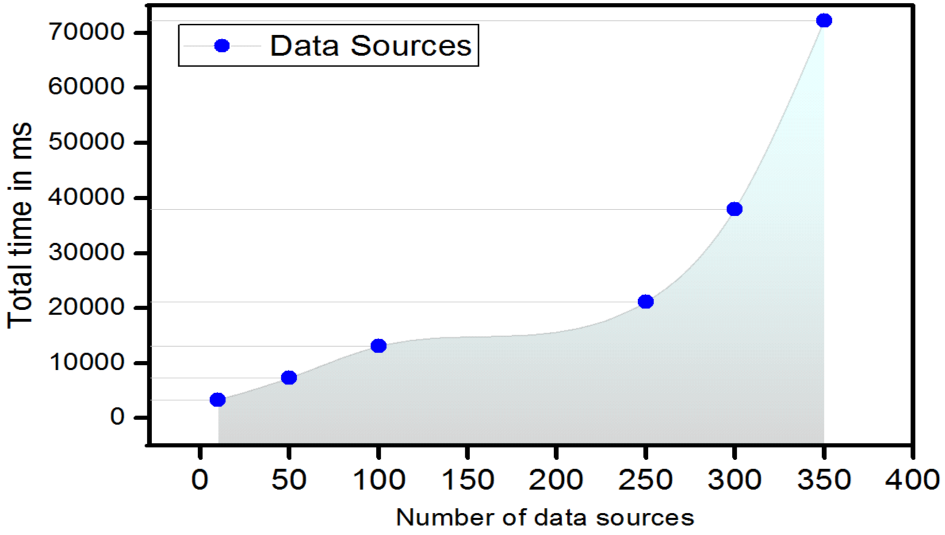
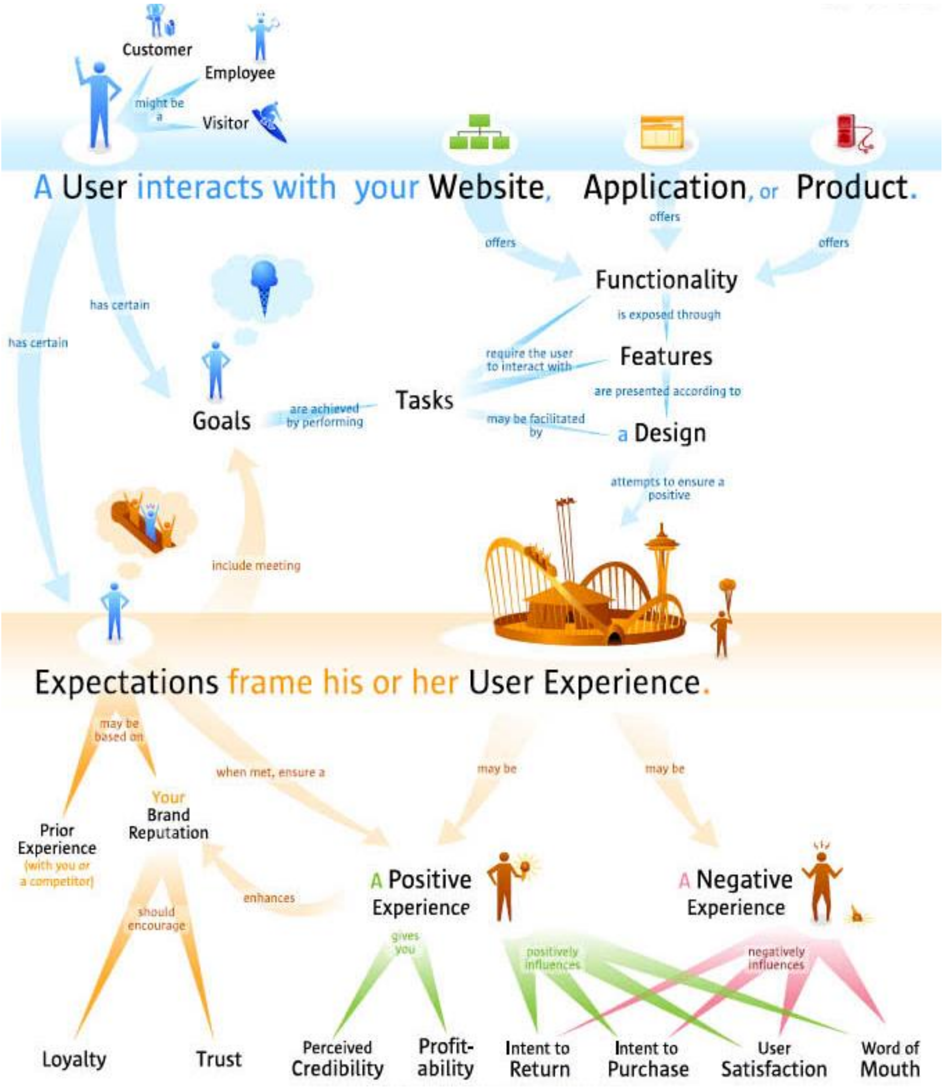


Fig 4. Scalability Testing



User Experience

- The overall experience a person has when interacting with any artifacts
  - Subjective, holistic, emotional, long-term

- The better your user experience, the more likely it is people will want to do business with you

How do you improve user experience?

- Measure how satisfied users are when they interact with your company

- Law, Effie L-C., and Paul van Schaik. "Modelling user experience–An agenda for research and practice." (2010): 313-322.

User experience (UX) evaluation

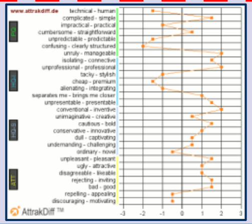
- User experience (UX) evaluation refers to a collection of methods, skills and tools utilized to uncover how a person perceives a digital artifacts before, during and after interacting with it.

- Different methods (**implicit and explicit**) and **technologies** used to **collected data** in order **measure** the certain **aspect** of **user experience** belongs to

When:	Before Usage	During Usage	After Usage	Over time
What:	Anticipated UX	Momentary UX	Episodic UX	Cumulative UX
How:	Imagining experience	Experiencing	Reflecting on an experience	Recollecting multiple periods of user

How to evaluate user experience?

Observation



self-reporting

subjective nature

Psychophysiological measurements

Existing methods - UX evaluation methods and weakness

Existing UX Evaluation Methods

Laborious

- Experience clip
- Repertory Grid Technique (RGT)
- Semi-structured experience interview
- 3E (Expressing Experiences and Emotions)
- Emotion Cards
- Experience Sampling Method (ESM)
- UX Curve
- AXE (Anticipated eXperience Evaluation)
- Day Reconstruction Method

High Fidelity Prototype

- 2DES
- Emotion Sampling Device (ESD)
- Day Reconstruction Method
- Long term diary study

Basic Emotions

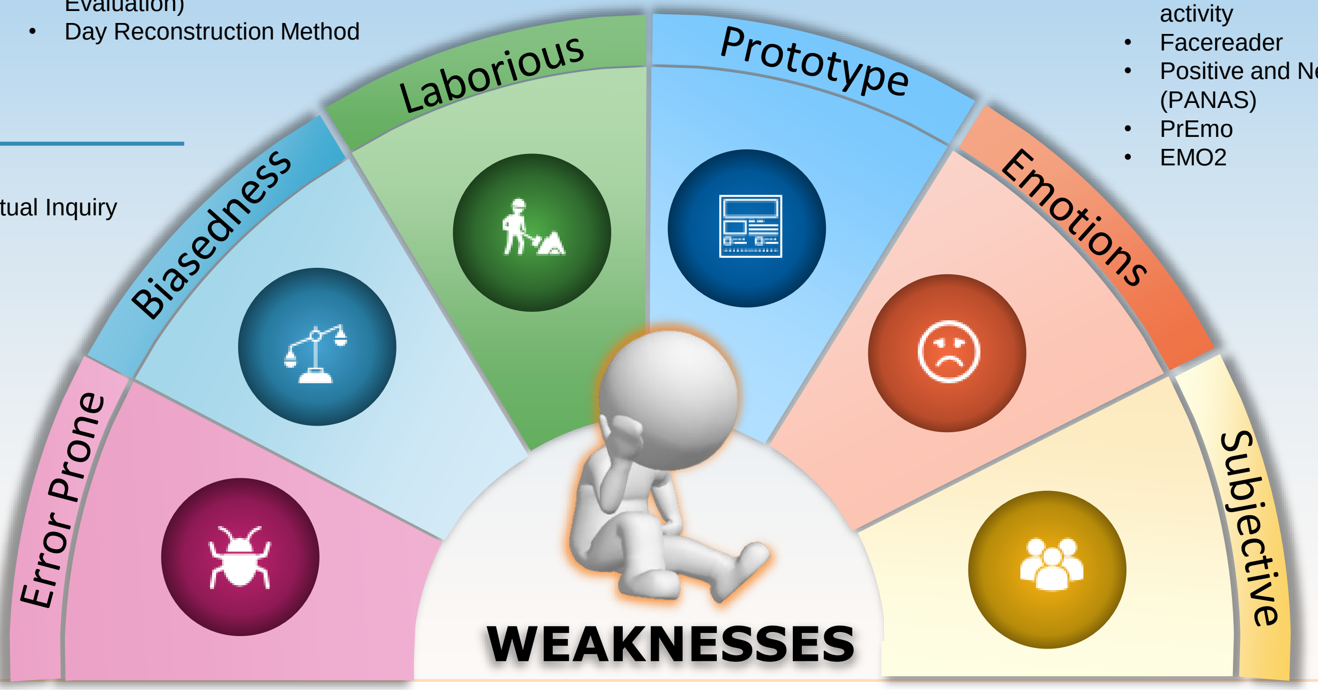
- Emotion Sampling Device (ESD)
- Differential Emotions Scale (DES)
- Physiological arousal via electrodermal activity
- Facereader
- Positive and Negative Affect Scale (PANAS)
- PrEmo
- EMO2

Biasedness

- EMO2
- Experiential Contextual Inquiry
- Immersion

Error Prone

- Emofaces
- Feeltrace
- Context-aware ESM
- 3E (Expressing Experiences and Emotions)
- Emotion Cards
- Experience Sampling Method (ESM)
- UX Curve
- AXE (Anticipated eXperience Evaluation)

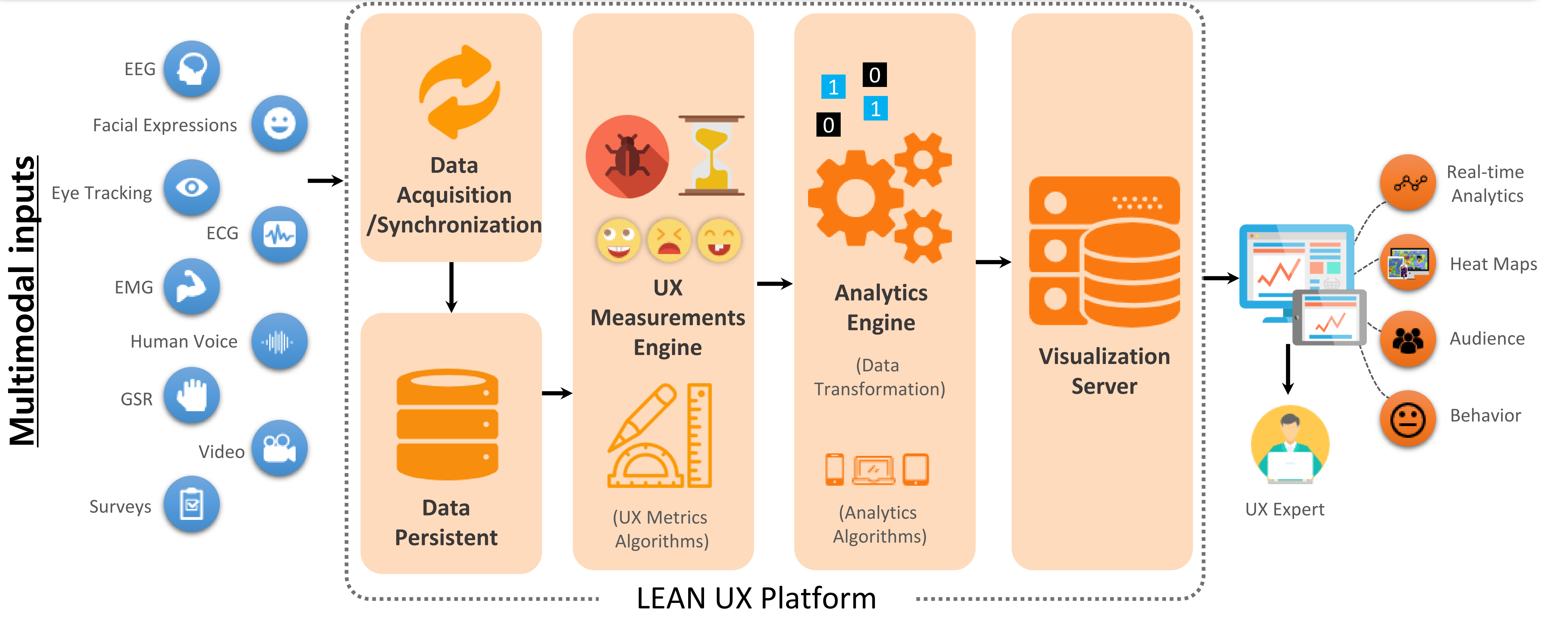


Subjective

- Geneva Appraisal Questionnaire
- Positive and Negative Affect Scale (PANAS)
- PrEmo
- ServUX questionnaire
- Aesthetics scale
- Affect Grid
- Attrak-Work questionnaire
- Exploration test
- Hedonic Utility scale (HED/UT)
- I.D. Tool
- Game experience questionnaire (GEQ)
- AttrakDiff
- Audio narrative
- SUMI

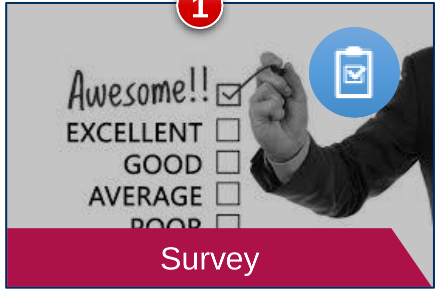
LEAN UX Platform Overview

"Lean UX core technologies and platform" that combines the different measurements by acquiring the complete picture of user emotional experience."



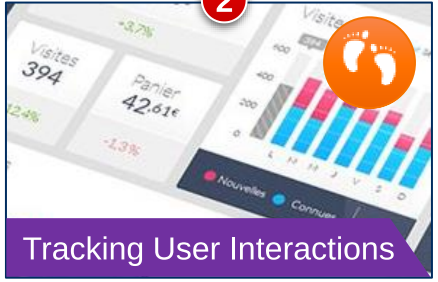
Core Technologies of Lean UX

Explicit measures



Sophisticated survey methodologies with advanced biometrics

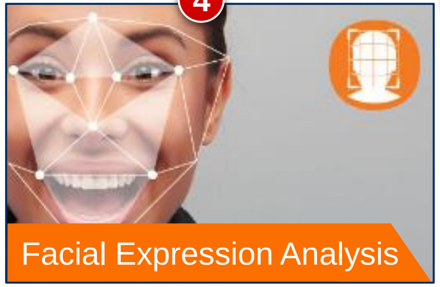
Observations measures



Track the user interaction while using the system

Implicit measures

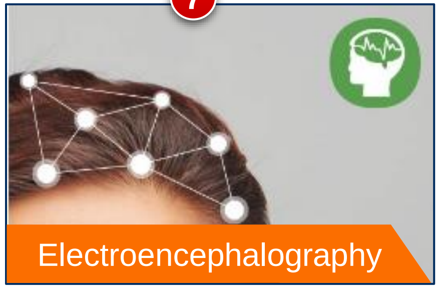
Multi-modal data acquisition and synchronization



Gain deeper insights into human emotional reactions

Detect emotional and motivational processes

Detect the drivers of attention in real life & lab environments



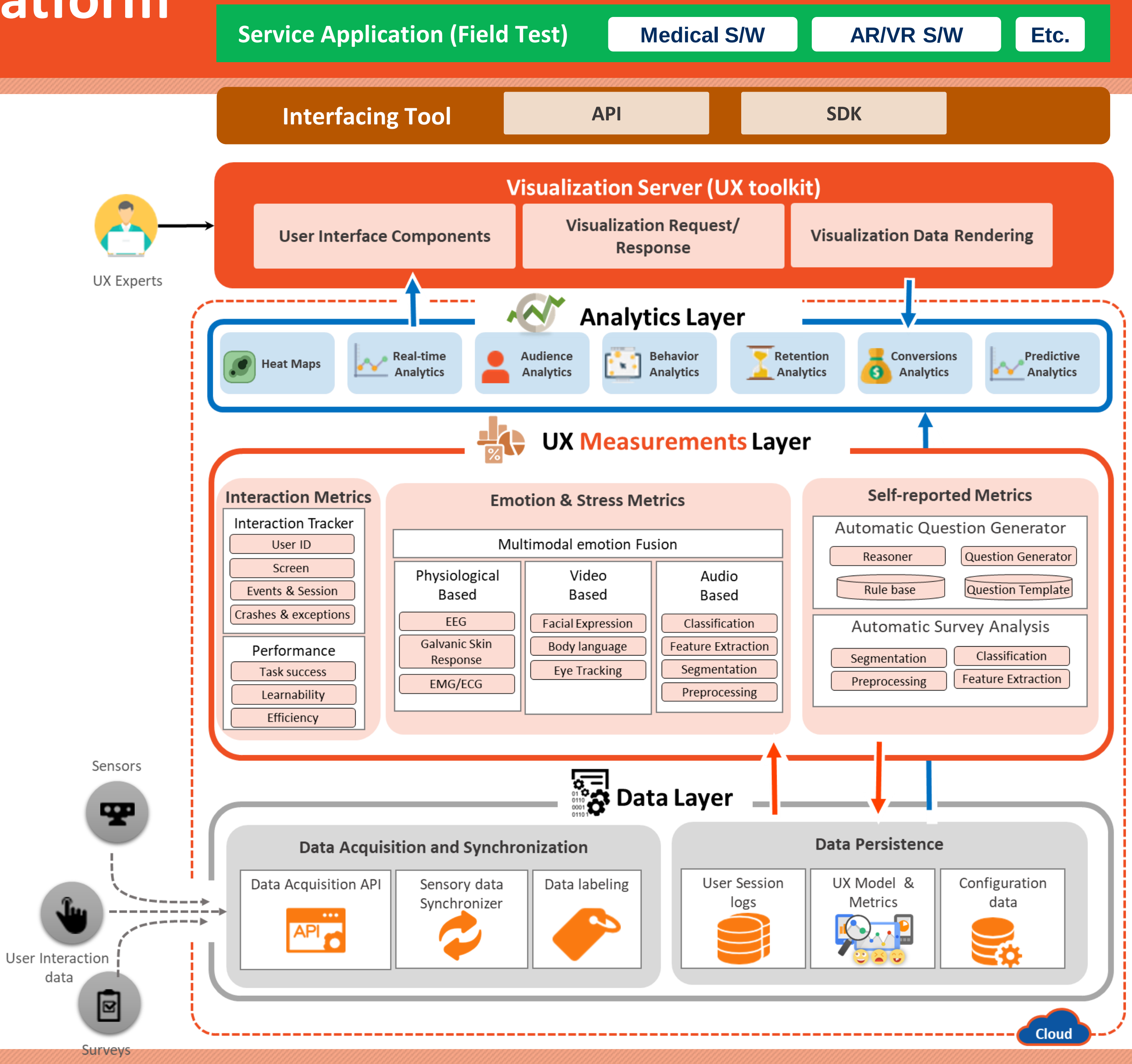
Detect emotional and motivational processes

Measure emotional arousal & stress by measuring changes in the conductivity of the skin

Creation of UX Model from online user reviews

All synchronized data streams is real time visualized in combination with stimuli. Individual or aggregated.

Lean UX Platform



Development environment