



LEEDS
BECKETT
UNIVERSITY

Citation:

Ramachandran, M (2019) Big Data SE vs. SE for Big Data. In: 5th International Conference on Data Science, 18 November 2019 - 21 November 2019, Belgrade. (Unpublished)

Link to Leeds Beckett Repository record:

<https://eprints.leedsbeckett.ac.uk/id/eprint/6387/>

Document Version:

Conference or Workshop Item (Presentation)

The aim of the Leeds Beckett Repository is to provide open access to our research, as required by funder policies and permitted by publishers and copyright law.

The Leeds Beckett repository holds a wide range of publications, each of which has been checked for copyright and the relevant embargo period has been applied by the Research Services team.

We operate on a standard take-down policy. If you are the author or publisher of an output and you would like it removed from the repository, please [contact us](#) and we will investigate on a case-by-case basis.

Each thesis in the repository has been cleared where necessary by the author for third party copyright. If you would like a thesis to be removed from the repository or believe there is an issue with copyright, please contact us on openaccess@leedsbeckett.ac.uk and we will investigate on a case-by-case basis.

Big Data SE vs. SE of BD Systems

Dr Muthu Ramachandran PhD FBCS Fellow of Advance HE, MIEEE, MACM
Principal Lecturer
Software Engineering Technologies and Emerging Practices (SETEP) Research
School of Built Environment, Engineering, and Computing
Leeds Beckett University
Email: M.Ramachandran@leedsbeckett.ac.uk



Process: Business Process Driven Service Development Lifecycle (BPD-SDL)



Methods and Design Principles: service components with soaML



Reference Architecture for big data (REF4BD)



Tools (SAS, Visual Paradigm, BonitaSoft, Bizagi Studio, Tabulea, Mathematica, Azure/ML)



SOSE4BD as a Service (SOSEaaS), BDaaS, Big Data Adoption Framework as a Service (BAaaS), Software Engineering Analytics as a Service (SEaaS), SE Prediction Model as a Service (SEPaaS), Bug Prediction as a Service with Azure/ML (MLaaS), BD Metrics as a Service (BDMaaS)



Adoption Models



Evaluation & Applications

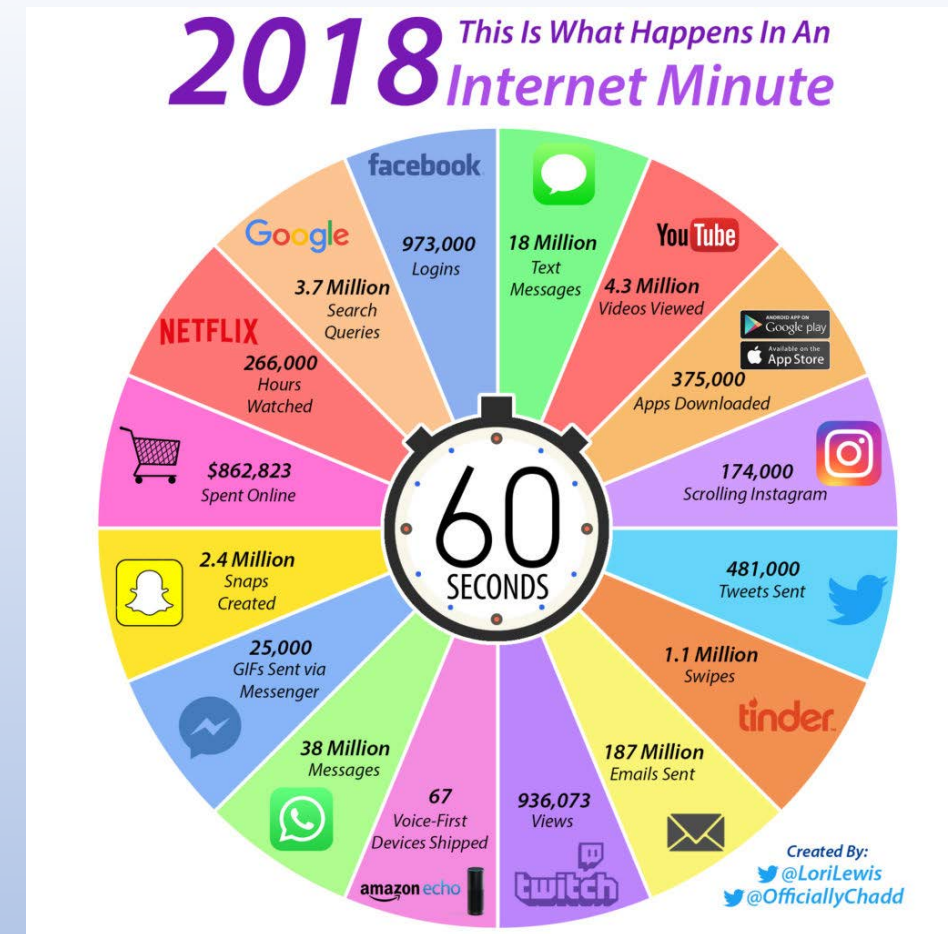
Research Motivation

- In this changing era of integrated IoT, Cloud, Real-time Big Data Stream (Social Media, Smart Cities, Smart Living, etc.) and services are to be Robust, Agile, Accessible and Available to its clients. For secured and guaranteed delivery of services, every big organization is shifting their service delivery model to Enterprise Service Bus (ESB). The following research questions are posed:
- How do we achieve Data Reuse, Reliability, Resiliency (3Rs) and Security, Accuracy and Availability (SAA)?
- How do we engineer a systematic approach to using and reusing software repositories, 50 years of software engineering knowledge and experience data for developing software and software as a service paradigm (**Big Data Software Engineering**)?
- How do we engineer a systematic approach with 50 years of software engineering approach to data science and to develop big data systems and services (**Software Engineering for Big Data**)?
- What are the design principles for a SOA driven reference architecture?
- What are services comprise reference architecture for big data systems?
- How to classify technologies and products/services of big data systems?
- How should software development teams integrate Data Analytics (Data Collection, Transformation, Analytics, and Improvement) into their software development process?
- What new roles, artifacts, and activities of BD process come into play?
- How do we integrate BD process of new roles, artifacts, and activities tie into existing agile or DevOps process?

BD Concepts

- Big data is characterized by 8V's: volume (large amounts of data), velocity (continuously processed data in real time), variety (unstructured, semi-structured or structured data in different formats and from multiple and diverse sources), veracity (uncertainty and trustworthiness of data), validity (relevance of data to the problem to solve), volatility (constant change of input data), and value (how data and its analysis adds value).
- Big data systems are software applications that process and potentially generate big data. Such applications receive and process data from various diverse (usually distributed) sources, such as sensors, devices, whole networks, social networks, mobile devices or devices in an Internet-of-Things. They process high workloads of data and handle high requests for data. The idea is to use large amounts of data strategically and efficiently to provide additional intelligence.

8Vs of big data



<http://www.visualcapitalist.com/internet-minute-2018/>

Velocity: BD requires real-time processing at varying intervals and may include stream as well as batch processing

Volume: BD provides a massive historical data over several time periods (years, months, weeks, days, etc)

Variety: The BD captured may be in variety of formats (multiple file files and multi-modal data) and may be structured and unstructured.

Veracity: The BD captured may contain unwanted data which requires extraction, transformation, and cleaning

Value: BD may contain very highly valuable data as well as not useful and it requires skilled data scientist to identify what to consider for analytical processing what to discard.

Characteristics of BD Systems Requirements

Spark is a cluster-computing framework, which means that it competes more with MapReduce than with the entire Hadoop ecosystem. For example, Spark doesn't have its own distributed filesystem, but can use HDFS. Spark SQL which integrates relational processing with the functional programming API of Spark. Querying data through SQL or the **Hive query language** is possible through Spark SQL. Spark SQL is a Spark module for structured data processing. It provides a programming abstraction called **DataFrames** and can also act as a distributed SQL query engine.

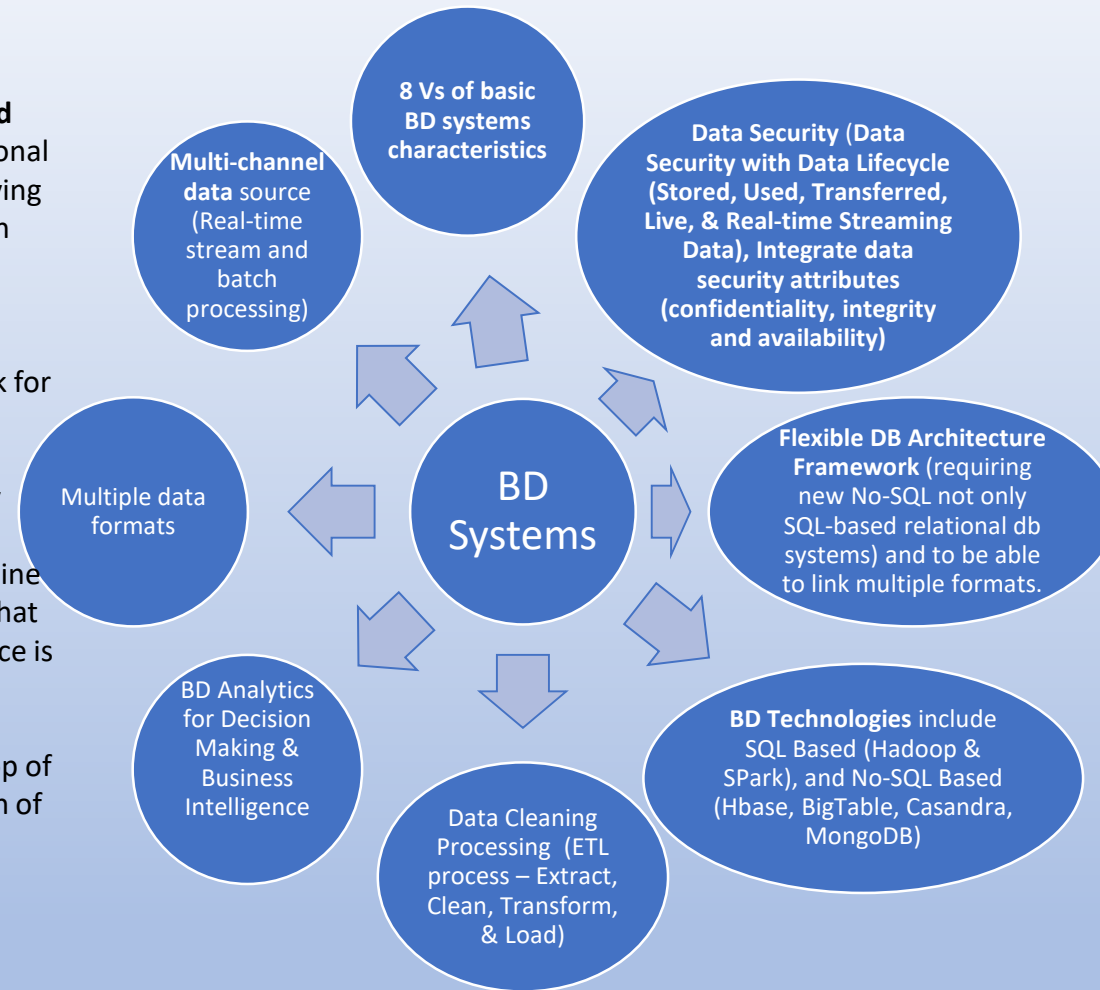
MongoDB is a NoSQL database, whereas Hadoop is a framework for storing & processing Big Data in a distributed environment. MongoDB is a document oriented NoSQL database. MongoDB stores data in flexible JSON like document format. You can easily map the documents to your applications.

Apache Cassandra is a NoSQL database ideal for high-speed, online transactional data, while Hadoop is a big data analytics system that focuses on data warehousing and data lake use cases. MapReduce is a programming paradigm for processing and handling large data sets.

Apache HBase is a column-oriented, NoSQL database built on top of Hadoop (HDFS, to be exact). It is an open source implementation of Google's Bigtable.

List of ETL Tools, <https://www.etltool.com/list-of-etl-tools/>

BD Analytics is the application of analysis, data, and systematic reasoning to make decisions, learn patterns of occurring to extract knowledge for improving process and business efficiency and cost. Analytics allows for summarising, filtering, modelling, and experimenting. **Tools and techniques include A/B testing, statistical modelling, machine learning, deep learning, natural language processing, and multi-linear subspace learning.**



Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs. The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part which is a MapReduce programming model. Hadoop splits files into large blocks and distributes them across nodes in a cluster.

Several research challenges of software engineering for developing big data systems

- Surveying the existing software engineering literature on applying software engineering principles into developing and supporting big data systems
- Identifying the fields of application for big data software systems
- Investigating the software engineering knowledge areas that have seen research related to big data systems
- Revealing the gaps in the knowledge areas that require more focus for big data systems development
- Determining the open research challenges in each software engineering knowledge area that need to be met
- **To be sustainable, we need an approach which is systematic, business-driven (supporting business-process and value driven), and based on established Software Engineering practices**

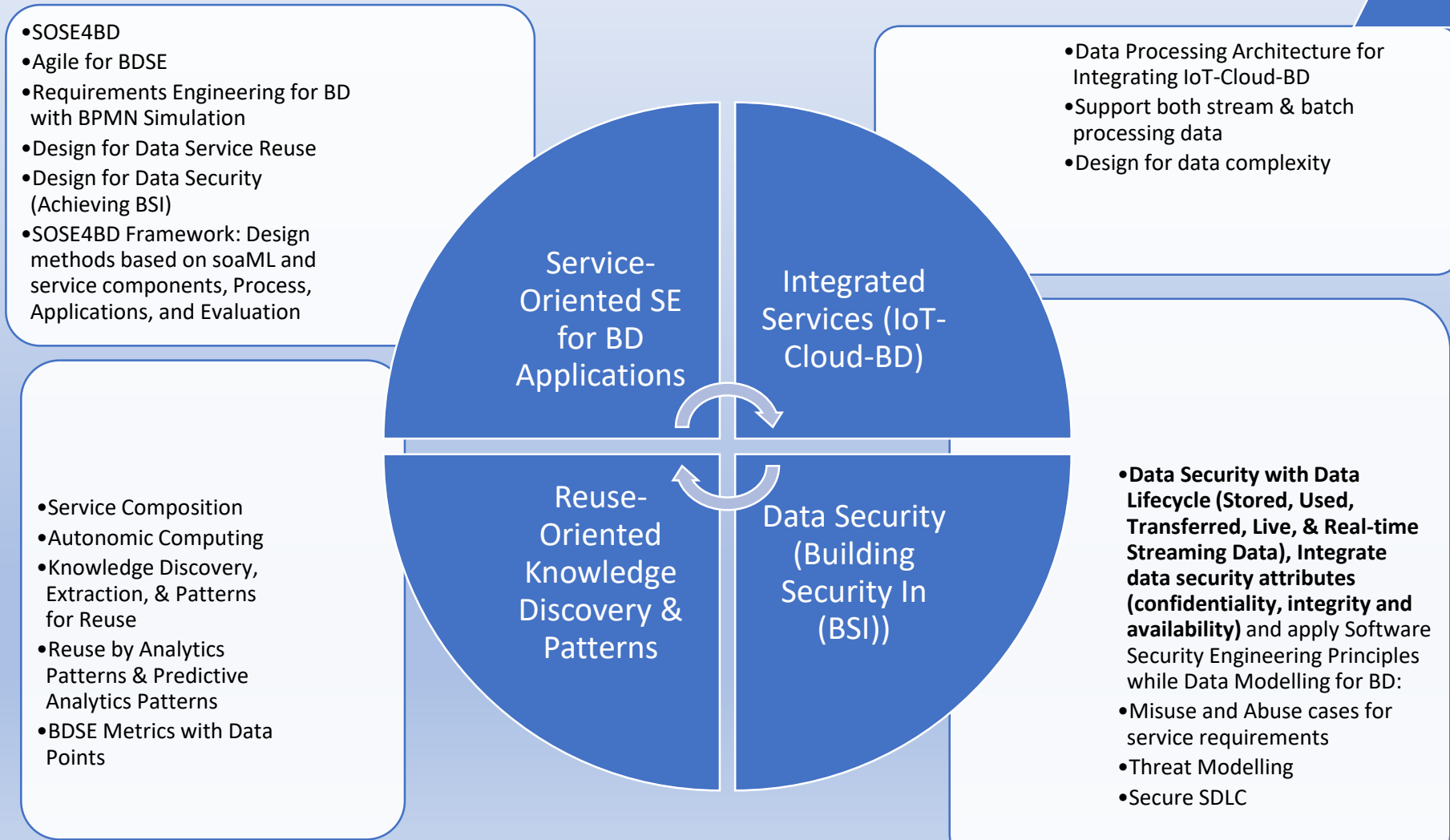
50 Years of SE

- 60 software development methodologies
- 50 static analysis tools
- 40 software design methods
- 37 benchmark organizations
- 25 size metrics
- 20 kinds of project management tools
- 22 kinds of testing and dozens of other tool variations.
- Minimum of 3000 programming languages software consisted, even though only 100 were frequently used. New programming languages are announced every 2 weeks, and new tools are out more than one in each month. Every 10 months new methodologies are discovered.
- Newly emerged **service computing paradigms (SOA, Web Services, Micro Services, Cloud SE, etc.)** and **established SE abstractions (Objects, classes, components, service components, virtualisation techniques, resource-oriented computing, and containers, etc.)**

Key Areas of Research Challenges

- **Software Engineering for Big Data** which can provide a systematic process for improving the development of big data systems. The process includes requirements gathering for BD, software architecture for BD, testing and debugging BD systems (performance, reliability, and security) where the logs of analysing 5V characteristics should be included, SE process for BD which could include CMMI, and finally Managing BD projects.
- **Big Data Software Engineering** is an area of research which should focus on utilising BD for the benefit of improving SE practices and to improve software production. The typical activities should include analytics for software engineering, data mining software repositories, visual analytics for software engineering, and self-adaptive systems which utilises data generated and self-learn.

Four Pillars of SE4BD Principles



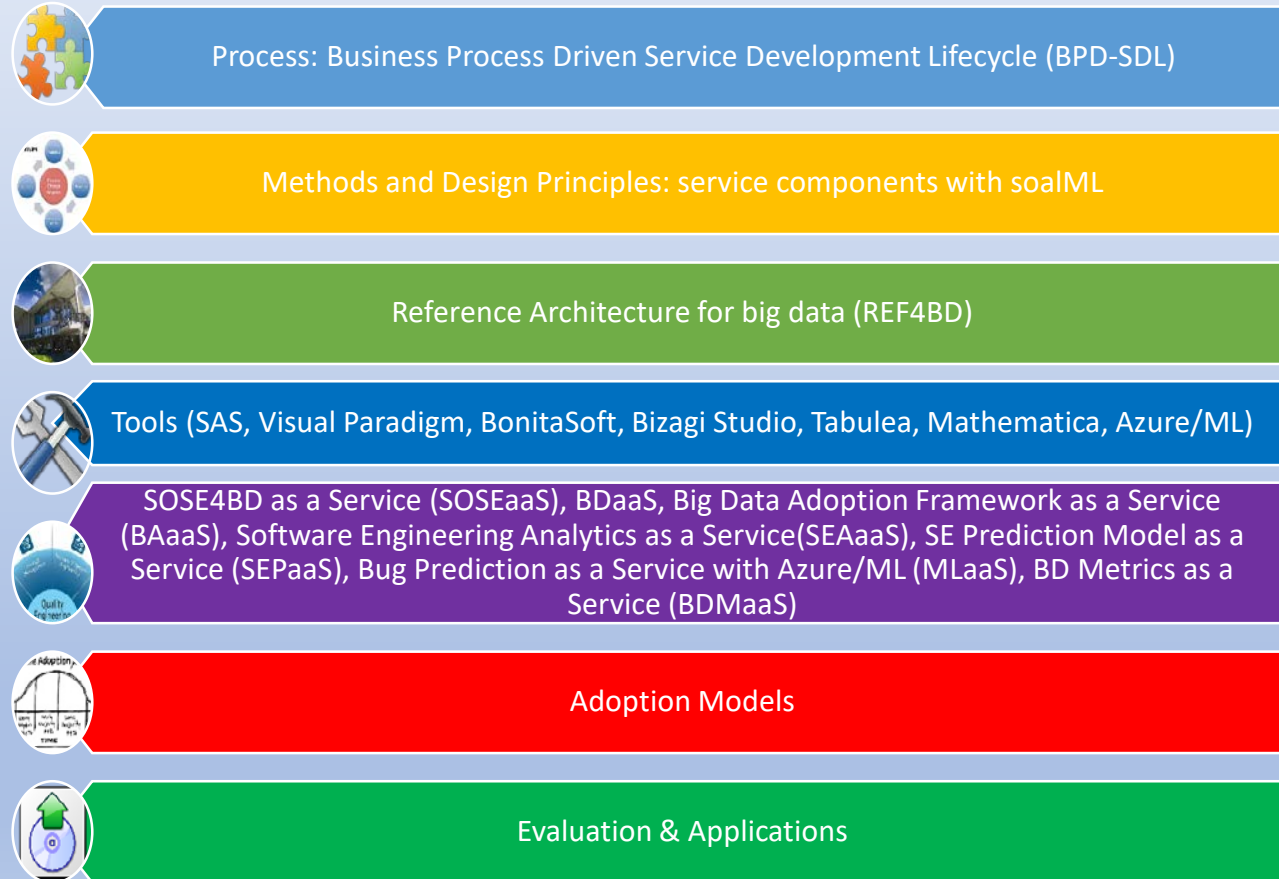
- Integrated Services (IoT-Cloud-BD)
- Data Security
- Reuse-Oriented Knowledge Discovery & Patterns
- Service-Oriented SE for BD Applications

Software Engineering Framework for Big Data Systems (SE4BD)

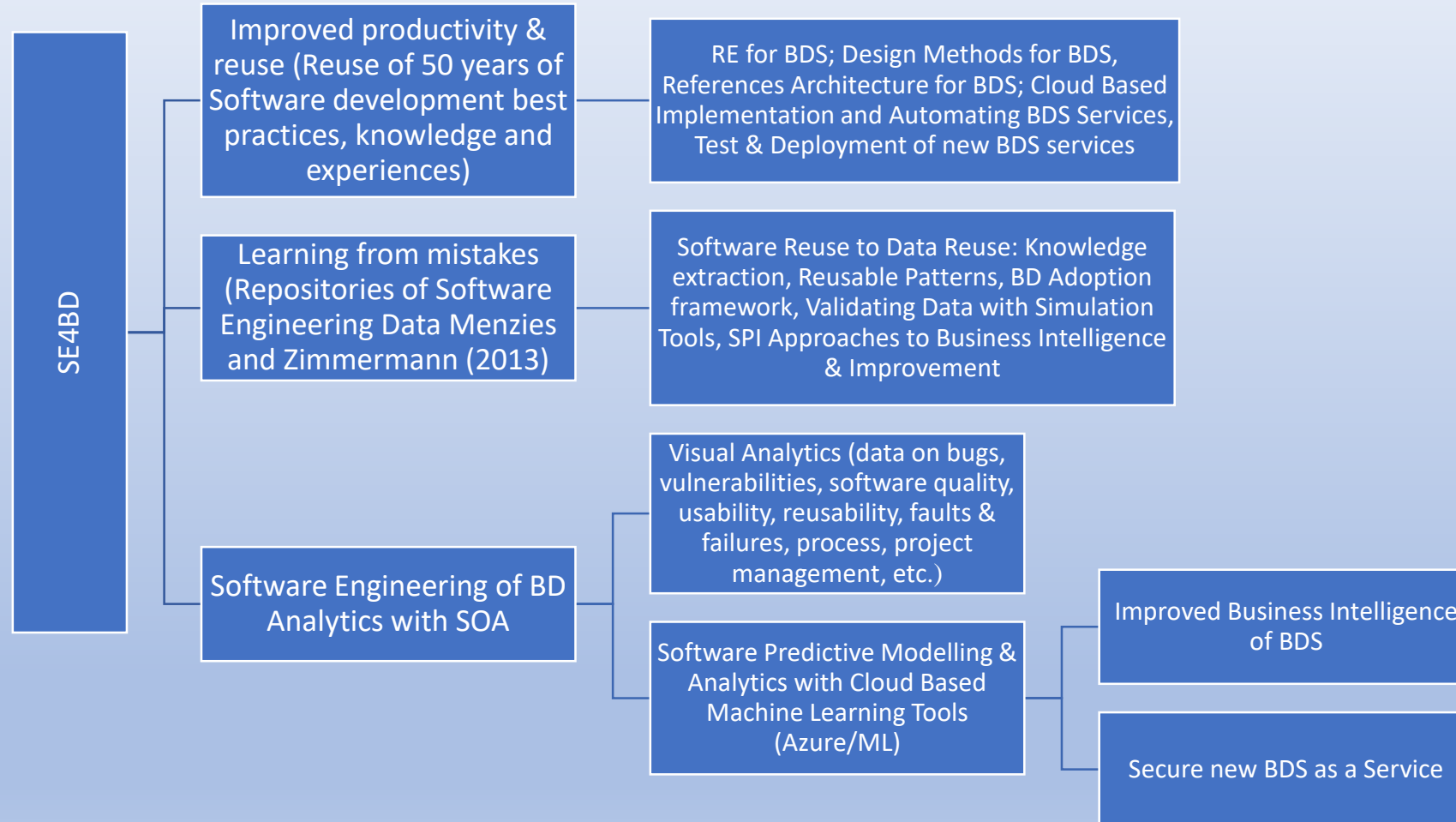
Framework: methods, process, reference architecture (REF4BD)
applications, adoption and evaluation

SE4BD Framework

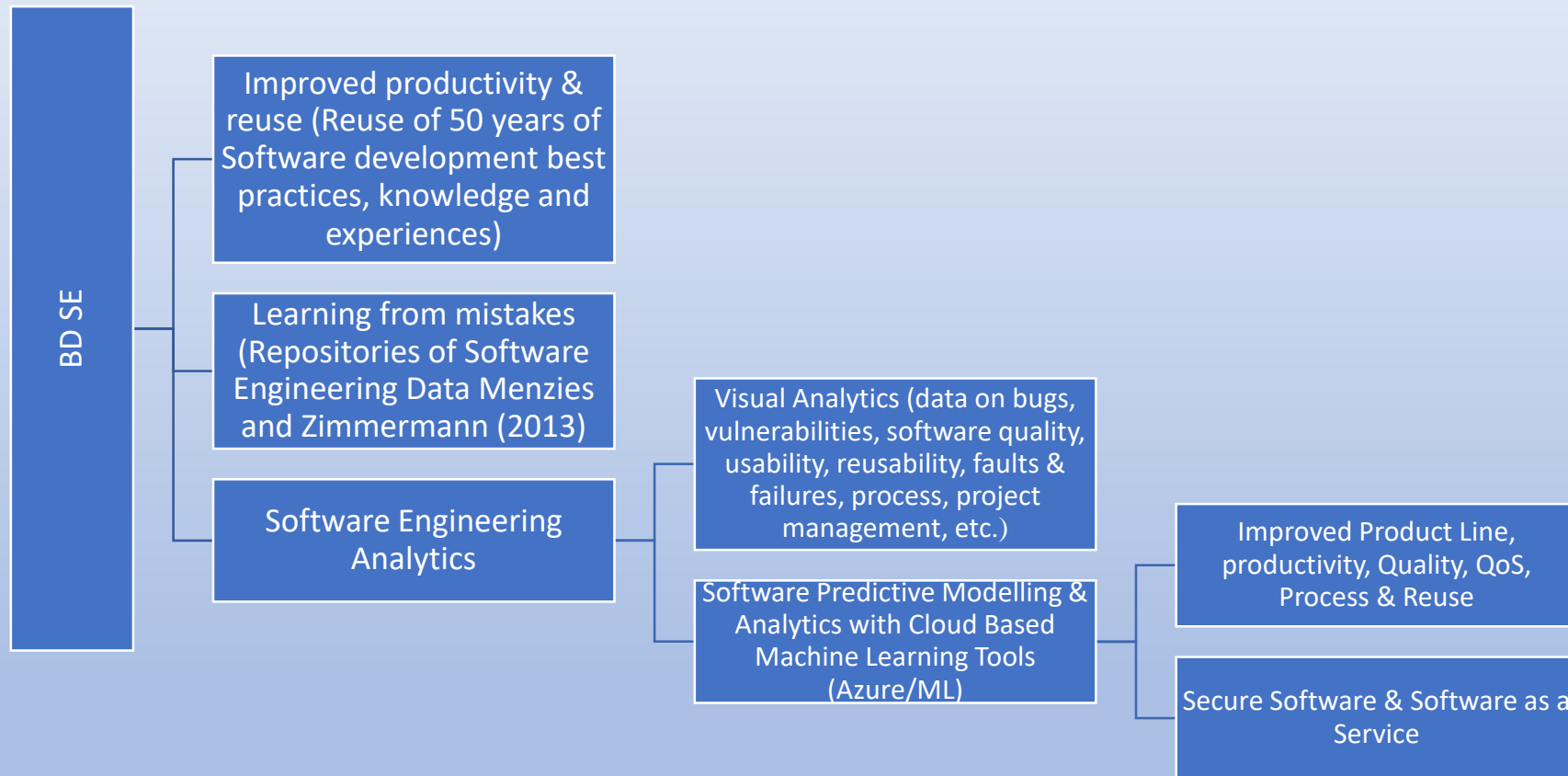
SE4BD is process-centric approach which provides process, Methods and design principles based on service components with soaML, reference architecture, tools, applications, adoption models, and evaluation



Benefits of SE Approach to BD Systems



Benefits of Big Data SE



Software Engineering Analytics

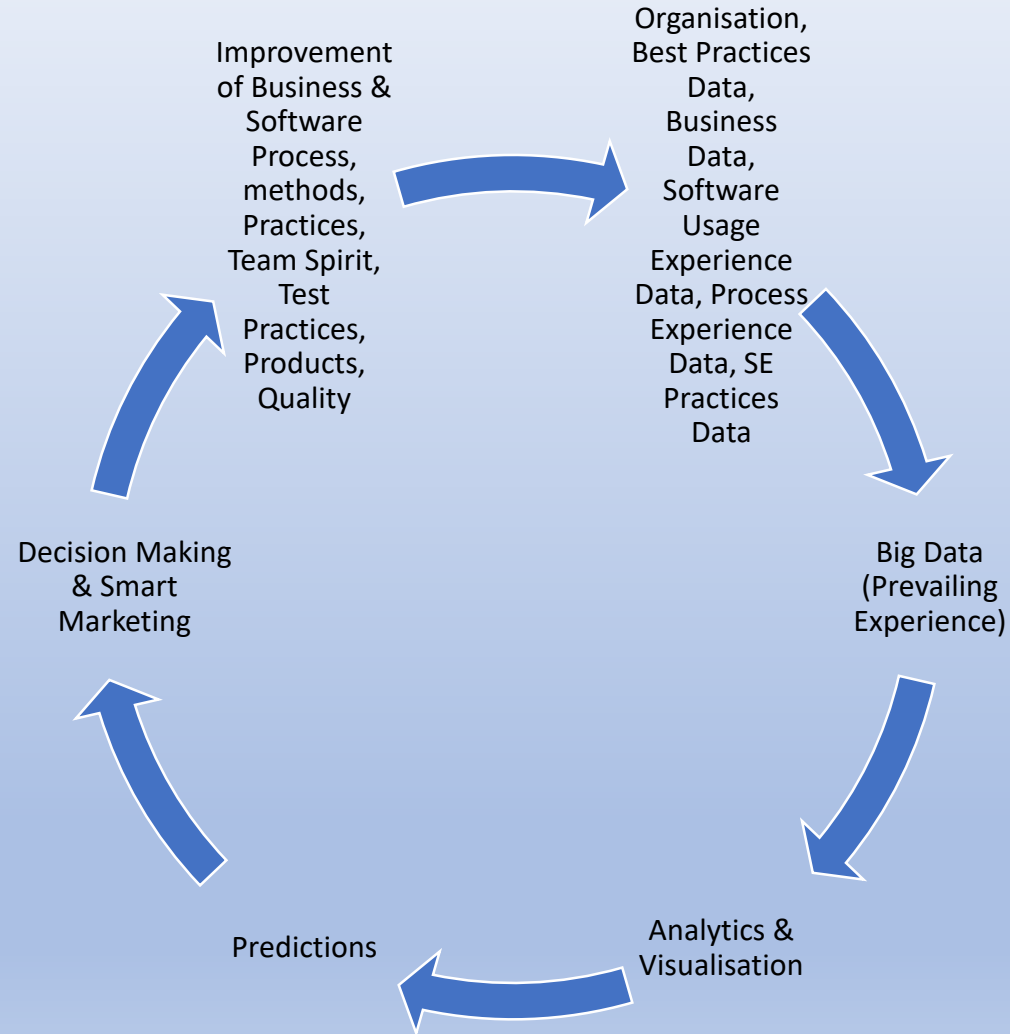
Data Science for Software Engineering



SE Analytics & Big Data SE

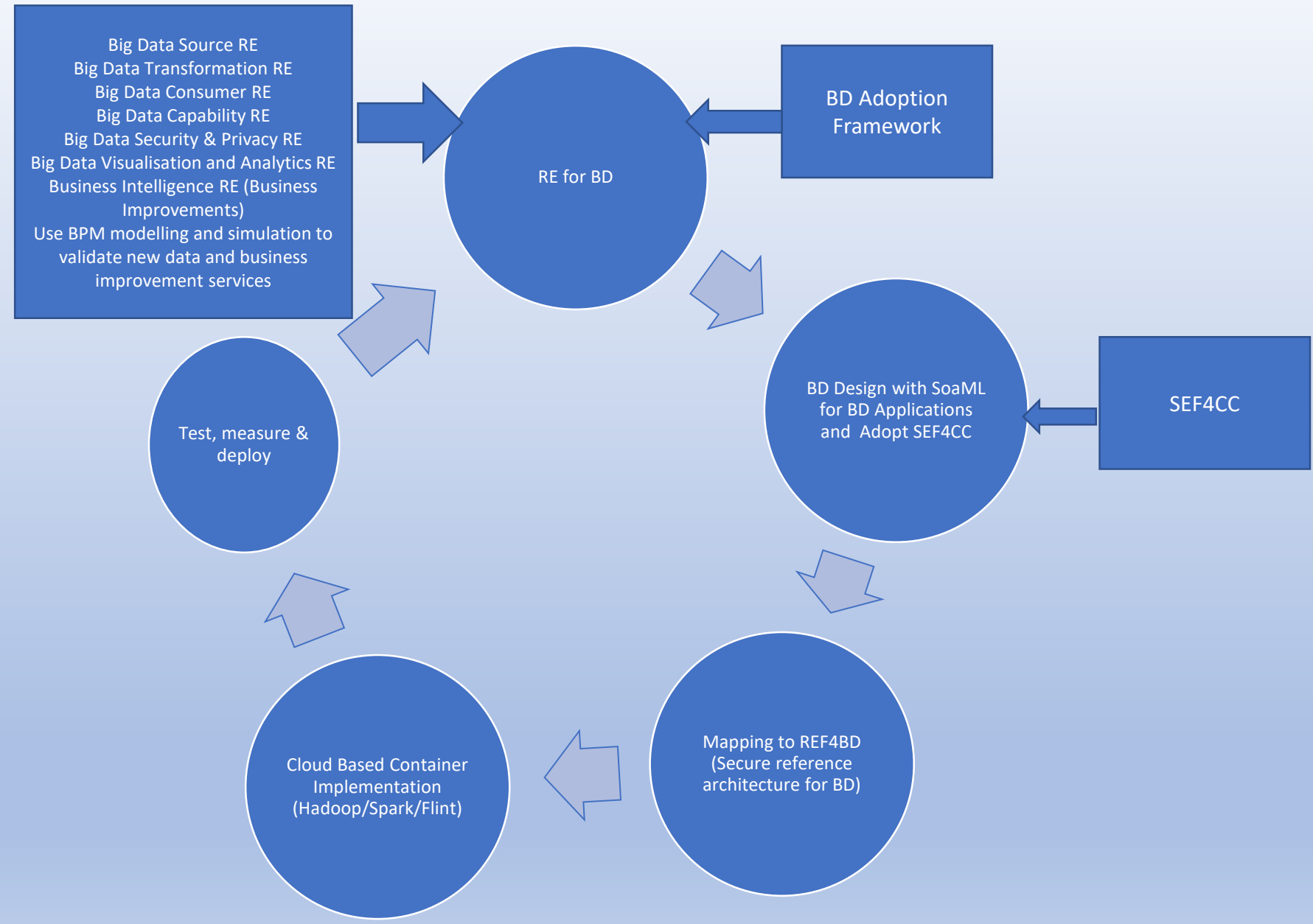
- Big data analytics, visualisation, and predictions has been useful and very popular research and applications in recent years. However, in the context of application of the big data practices to Software Engineering Analytics, there are some key questions need to be addressed:
 - How do we apply to software engineering analytics?
 - How do we collect and access SE experience data?
 - How do we apply them to decision making for business as well as the SE practices: process, methods, and technology?
 - How do we apply them for Software Process Improvement and Software Practices Improvement (SPI)
 - How do we apply them for software business process improvement?

SE for BD Analytics





SE4BD Lifecycle



Actionable SE Analytics Tools with SE Data Repositories

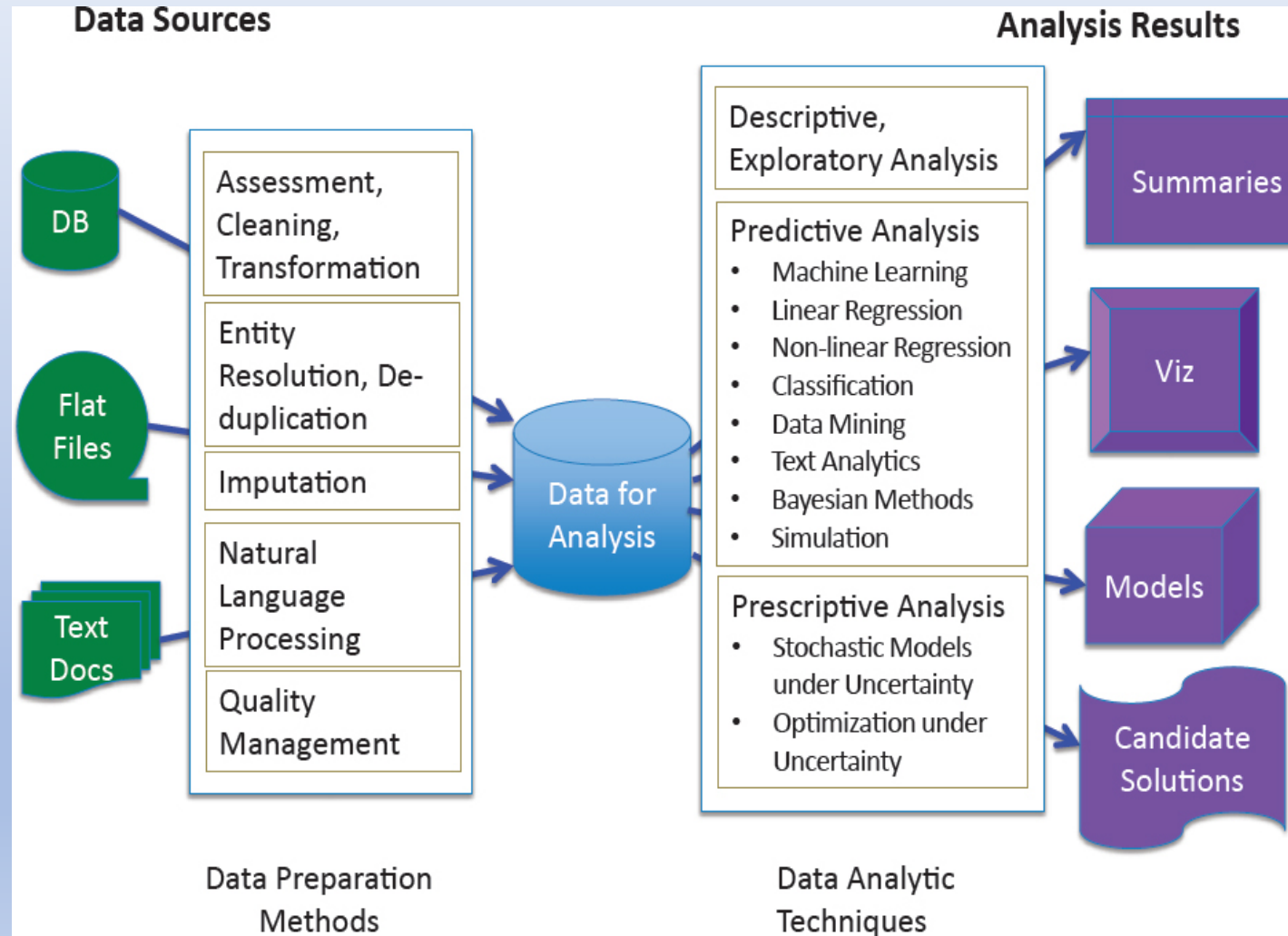
- Code coverage and visualisation tools
- Project Management & Monitoring Tools
- There are other tools to support management such as PROM, Hackystat which are capable of monitoring and to generate a statistical report of a software project. However, even after years of development, there is no major upgrade that can help predict management decisions (Zimmermann, 2013). Most of these tools were focused mostly more on collection rather than any critical analysis.
- Later, more tools were developed that realized the manager problems and created which focused data presentation rather than just collecting data. Tools like, Microsoft's Team Foundation Server and IBM's which keeps developers updated with modification, bugs and build results (Zimmermann, 2013). Other Project management tools such as Automated Project Office(APO) (Jones, 2017) also helps managers to come with the probable decision for the organization before or during the project.
- **Software Repositories:** As of late 2012, our Web searches show that Mozilla Firefox had 800,000 bug reports, and platforms such as Sourceforge.net and GitHub hosted 324,000 and 11.2 million projects, respectively.
- The PROMISE repository of software engineering data has grown to more than 100 projects and is just one of more than a dozen open source repositories that are readily available to industrial practitioners and researchers
- Jones, C 2018, Software Methodologies. [Electronic Resource] : A Quantitative Guide, n.p.: Boca Raton : CRC Press/Taylor & Francis
- Yang, Y. et al (2018) Actionable Analytics for Software Engineering, Actionable Analytics, Guest editors Introduction to Special Issue on Actionable Analytics for SE, IEEE Software, Jan/Feb 2018
- Menzies, T and Zimmermann, T (2013) Software Analytics: So What?," IEEE Software, vol. 30, no. 4, 2013
- **Applications: Bug Data Prediction Models, ML for SPI, and Agile Methods**

Repositories of software engineering data

Repository	URL
Bug Prediction Dataset	http://bug.inf.usi.ch
Eclipse Bug Data	www.st.cs.uni-saarland.de/softevo/bug-data/eclipse
FLOSSMetrics	http://flossmetrics.org
FLOSSMole	http://flossmole.org
International Software Benchmarking Standards Group (IBSBSG)	www.isbsg.org
ohloh	www.ohloh.net
PROMISE	http://promisedata.googlecode.com
Qualitas Corpus	http://qualitascorpus.com
Software Artifact Repository	http://sir.unl.edu
SourceForge Research Data	http://zerlot.cse.nd.edu
Sourcerer Project	http://sourcerer.ics.uci.edu
Tukutuku	www.metriq.biz/tukutuku
Ultimate Debian Database	http://udd.debian.org

Software Engineering Approaches for Data Science Applications

Data Science Process



Data Science Application 1:
Bug Data Prediction Models using
cloud based machine learning

<https://app.box.com/s/b1g4jsp4k7f9cof90an6dg1qju0vv9uh>

Application 2: Cloud Based Machine Learning Tool for Agile Method Decision Making

This tool is useful to know when making decision on choosing an Agile methods based on project size and constraints.

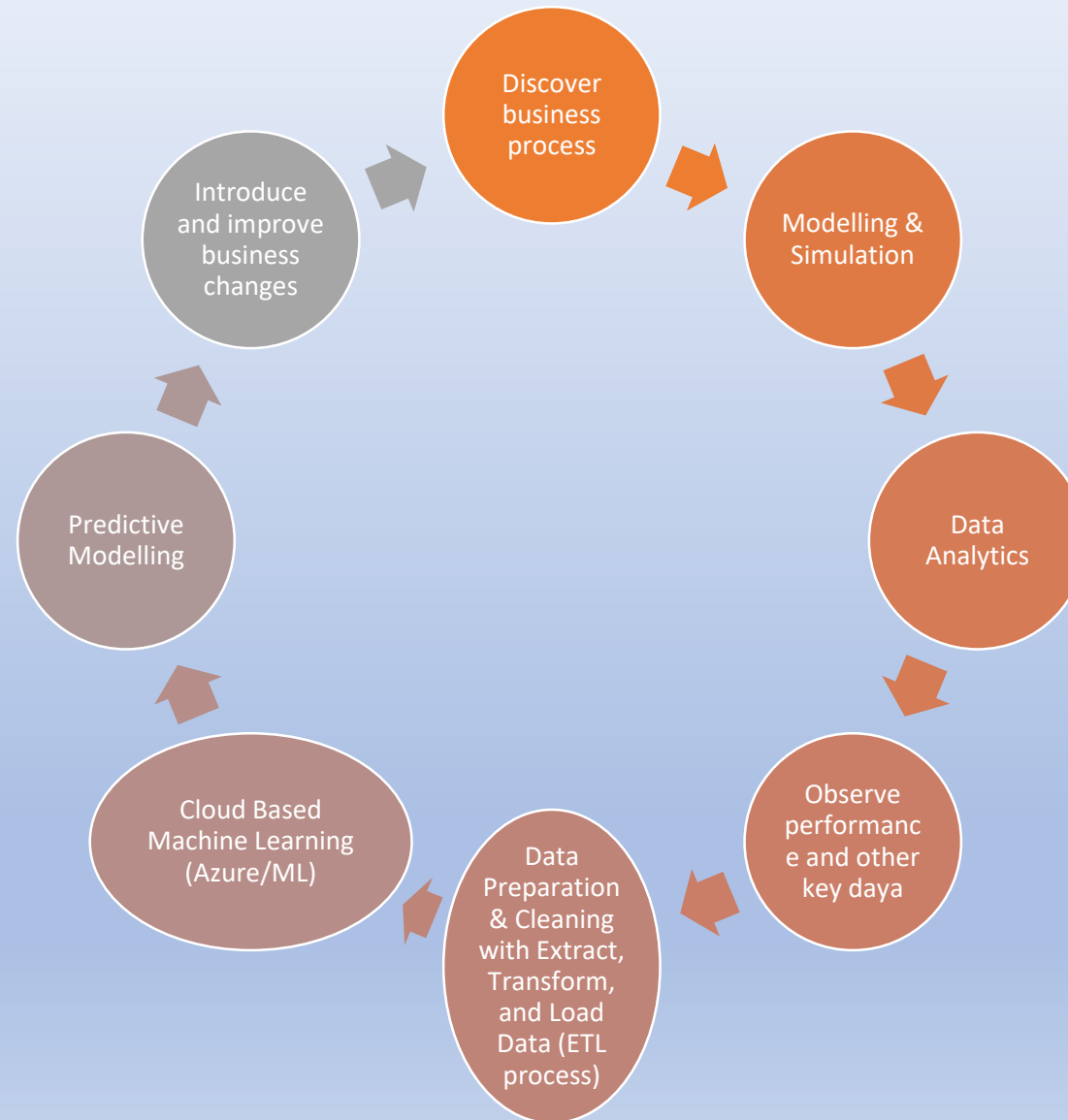
<https://app.box.com/s/7q8sjo0zf36qpsahoqv5ai28forv0vn>

Application 3: Cloud Based Machine Learning for Software Process Improvement

<https://app.box.com/s/6gd4zgimtz11014eavu3wv2c8cc1do6h>

Application 4: Process Mining and
Business Intelligence for Predictive
Modelling

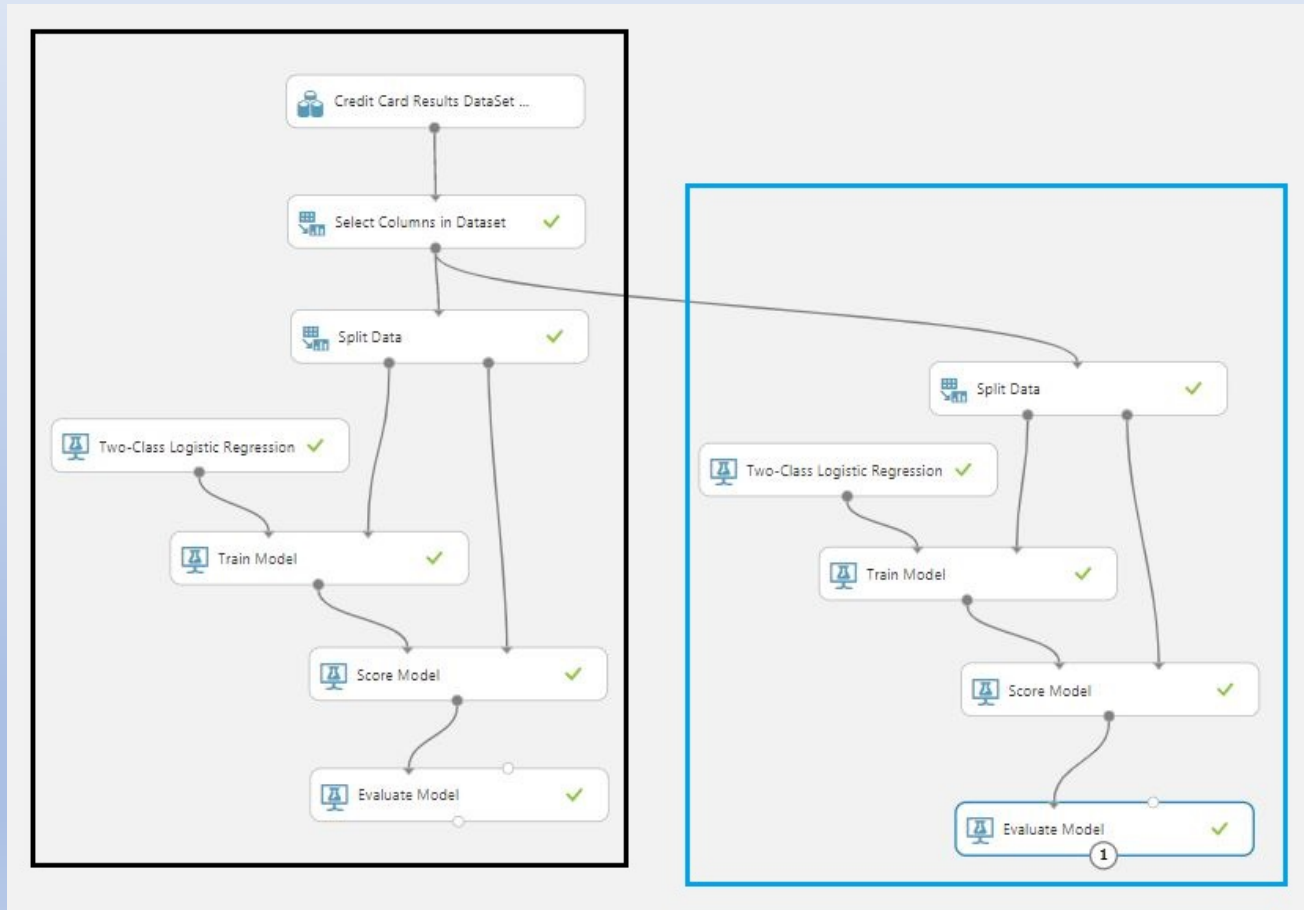
Process Mining and Business Intelligence with Machine Learning



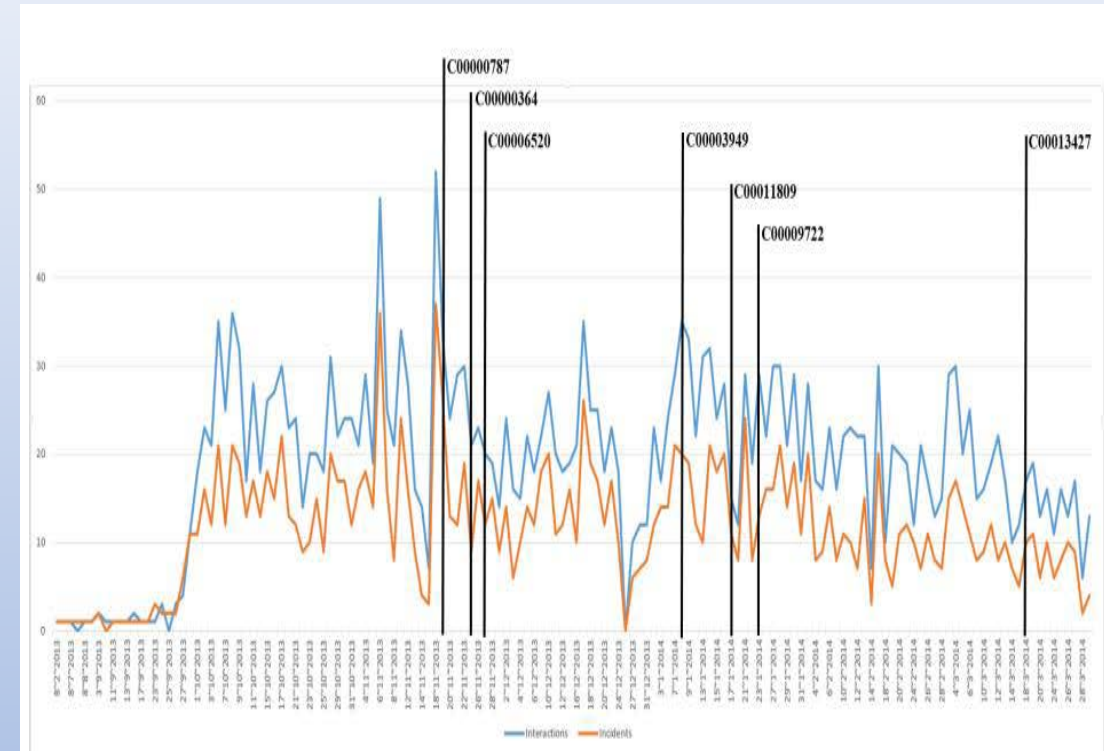
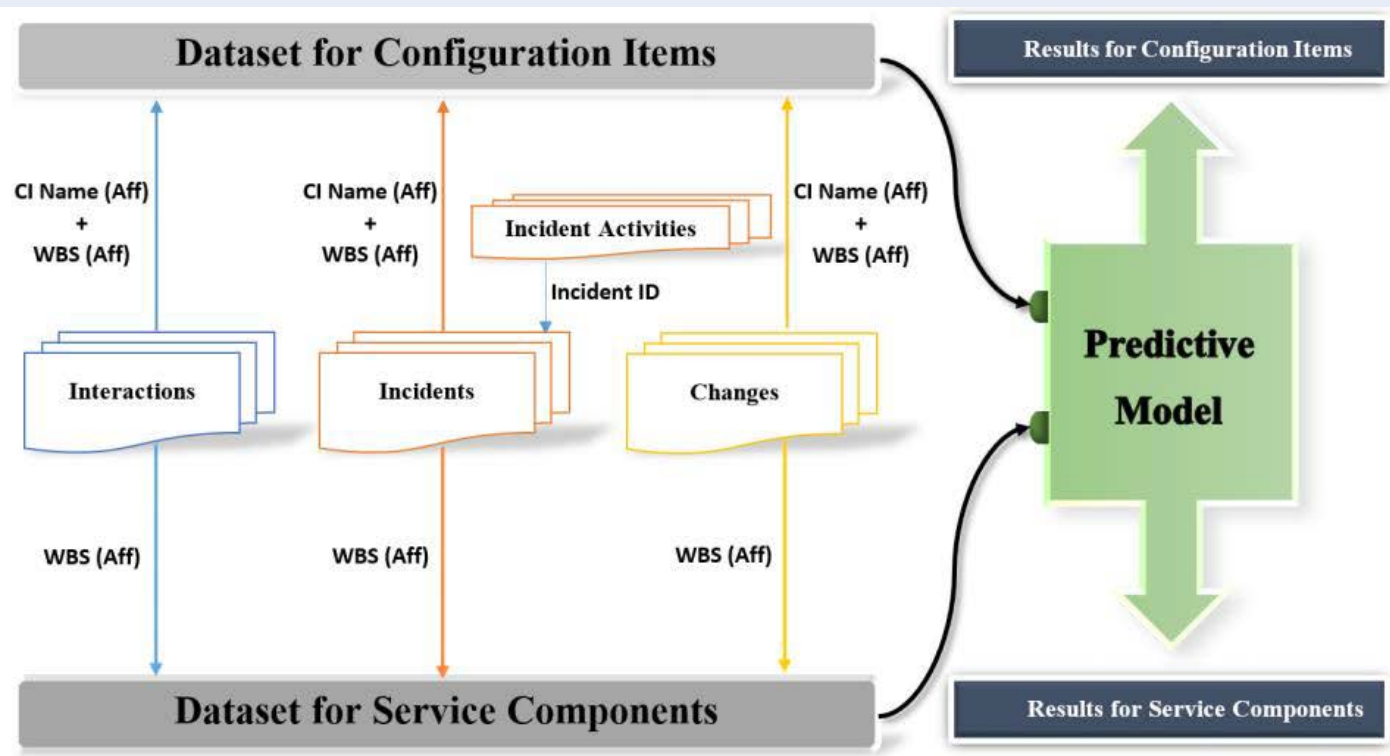
Benefits of Process Mining with Machine Learning

- Live data from the business services (event logs and performance data) to study the impact patterns
- Useful to study business and service patterns
- Useful to study business and service improvements
- Useful to improve change management process
- Reuse of services and data
- Improved QoS
- Predictive modelling for efficiency and QoS

Predictive Modelling for Loan Business Processes



Predictive Modelling for Service Desk Application



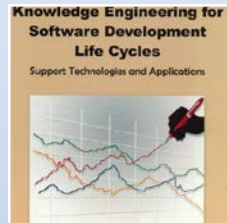
Occurrence of Interactions (blue) and Incidents (orange) in time with marked Changes (black) for Configuration Item "SBA000607" related to Service Component "WBS000263"

Software Engineering Framework for Service and Cloud Computing (SEF- SCC) for Developing Data Science Applications

SEF-SCC Framework Poster,

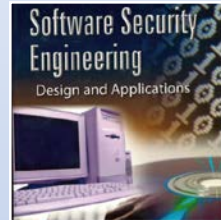
<https://app.box.com/s/u5fcktx687fy6qv2nhgzp93e5n9i7r8p>

SEF-SCC: Service-Security-Reuse – A Three Integrated Service Engineering Framework



Service Development

- Accuracy, Correctness & QoS Design Principles
- Service RE with BPMN and Simulation
- Service Design with Service Components (SoaML)
- Service Development (any platform)
- Service Testing & Deployment & Continuous Delivery



Software Security Engineering

- Building Security In (BSI), Resiliency, Fault-Tolerance Design Principles
- Service Security RE with Misuse & Abuse Use cases for all identified services
- Threat Modelling
- Design for Security
- Building Security In & Resiliency, Fault-tolerance,
- Software security testing



Service Reuse Engineering

- Design for Reuse & Design with Reuse, Composable, Scalable Design Principles
- Reuse RE (Commonality & Variability Analysis of secured requirements on selected BPMN & Secured use cases)
- Design for reuse approaches
- Reuse Development (Implementing composable services)
- Testing for reuse, composition & integration testing strategies



Software Engineering Lifecycle for Service and Cloud Computing (SEL4SCC)

Service Requirements

- Initial process models: Actors/roles/Workflows
- Detailed workflows
- Service Task modelling
- UI prototyping
- Process Simulation:
 - Configure Resources need for tasks
 - Load profiles in sec/min/days/no.of instances
 - Start the Process Simulation as a Service (PSSaaS)

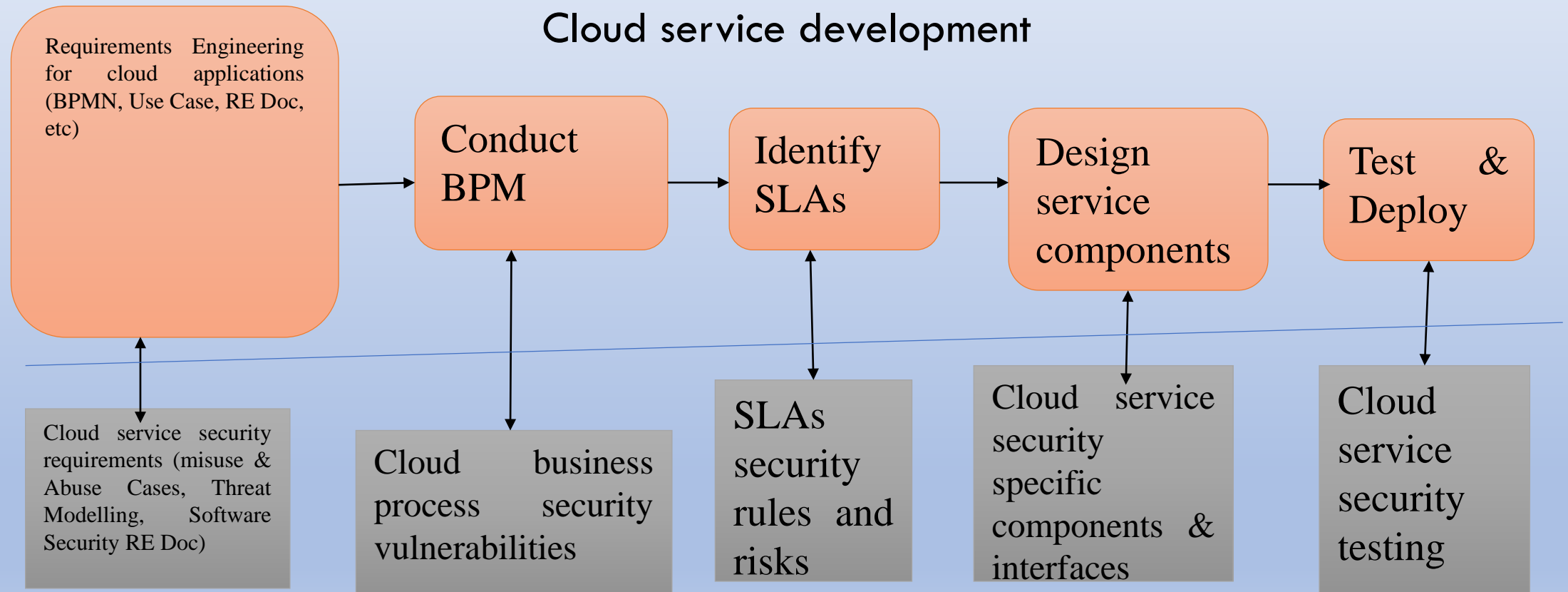
SOA Requirements with use case modelling, story cards, (Agile), Story Boards, CRC Cards, Feature-Oriented modelling

SOA Design with Service Component Models

SOA Implementation with SOAP/RESTful

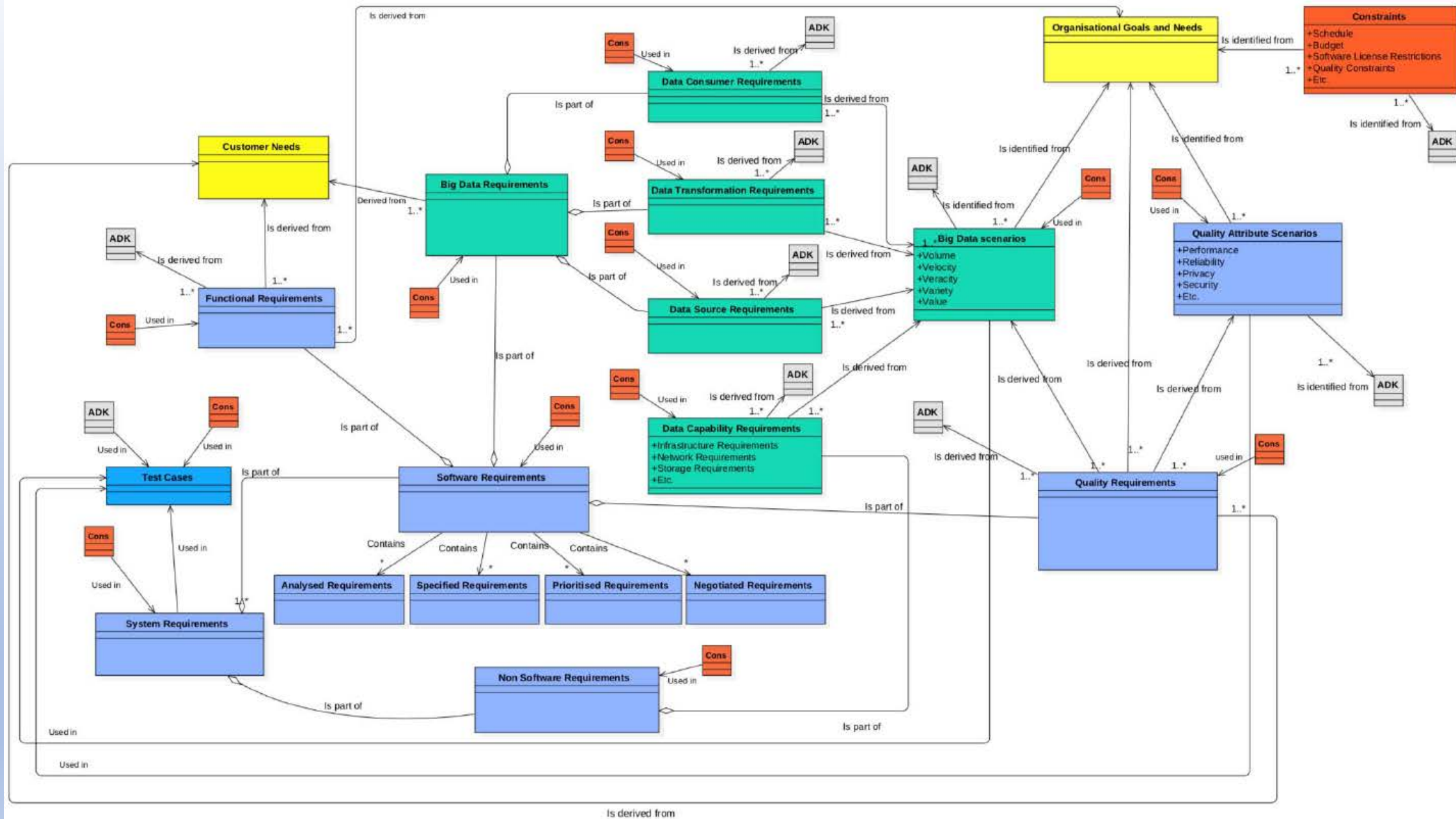
SOA Test & Deliver

Cloud service security development process with Building Security In (BSI) – Our Systematic Approach to developing secure services

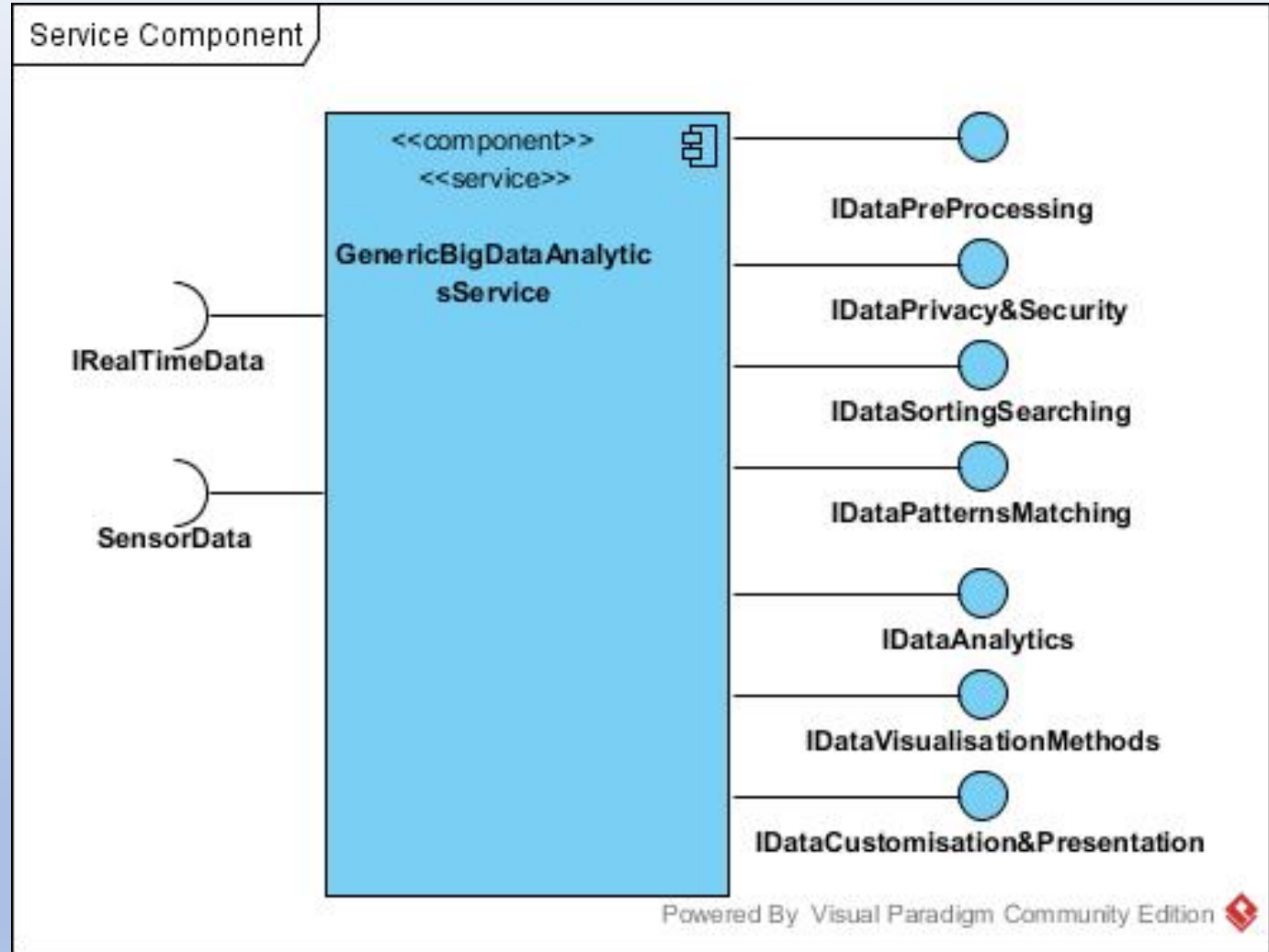


Build-In Security (BSI) – Cloud service development with build-in security

RE for BD

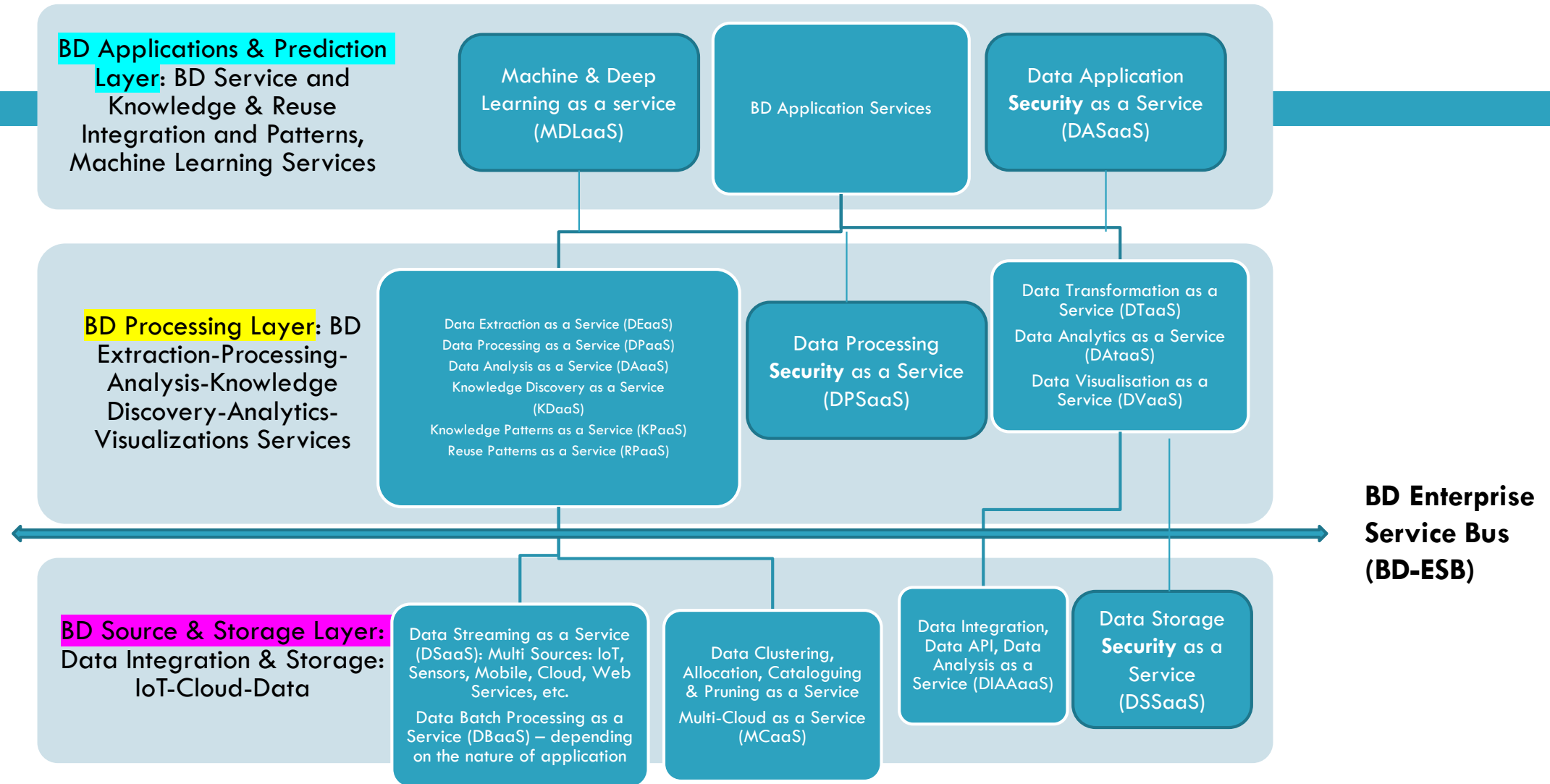


BD Service Component Model: Data Processing layer

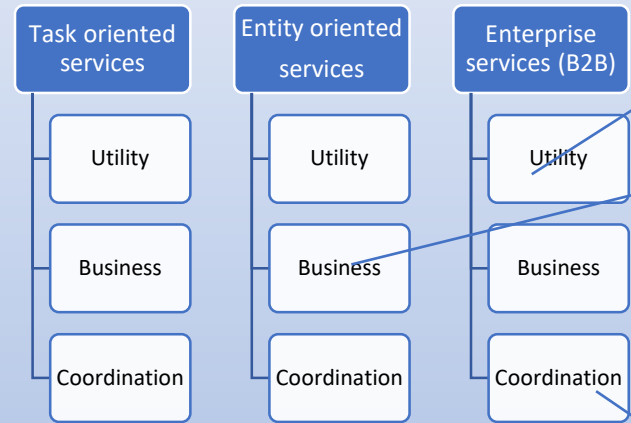


SEF4BD Reference Architecture for Big Data: Secure Service-Oriented SE for BD

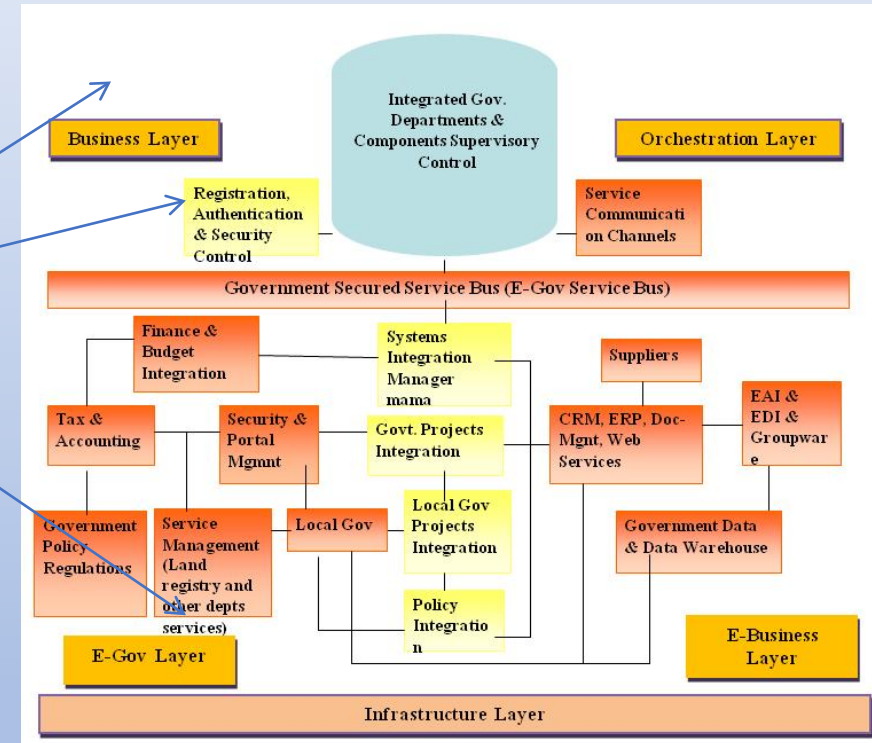
36



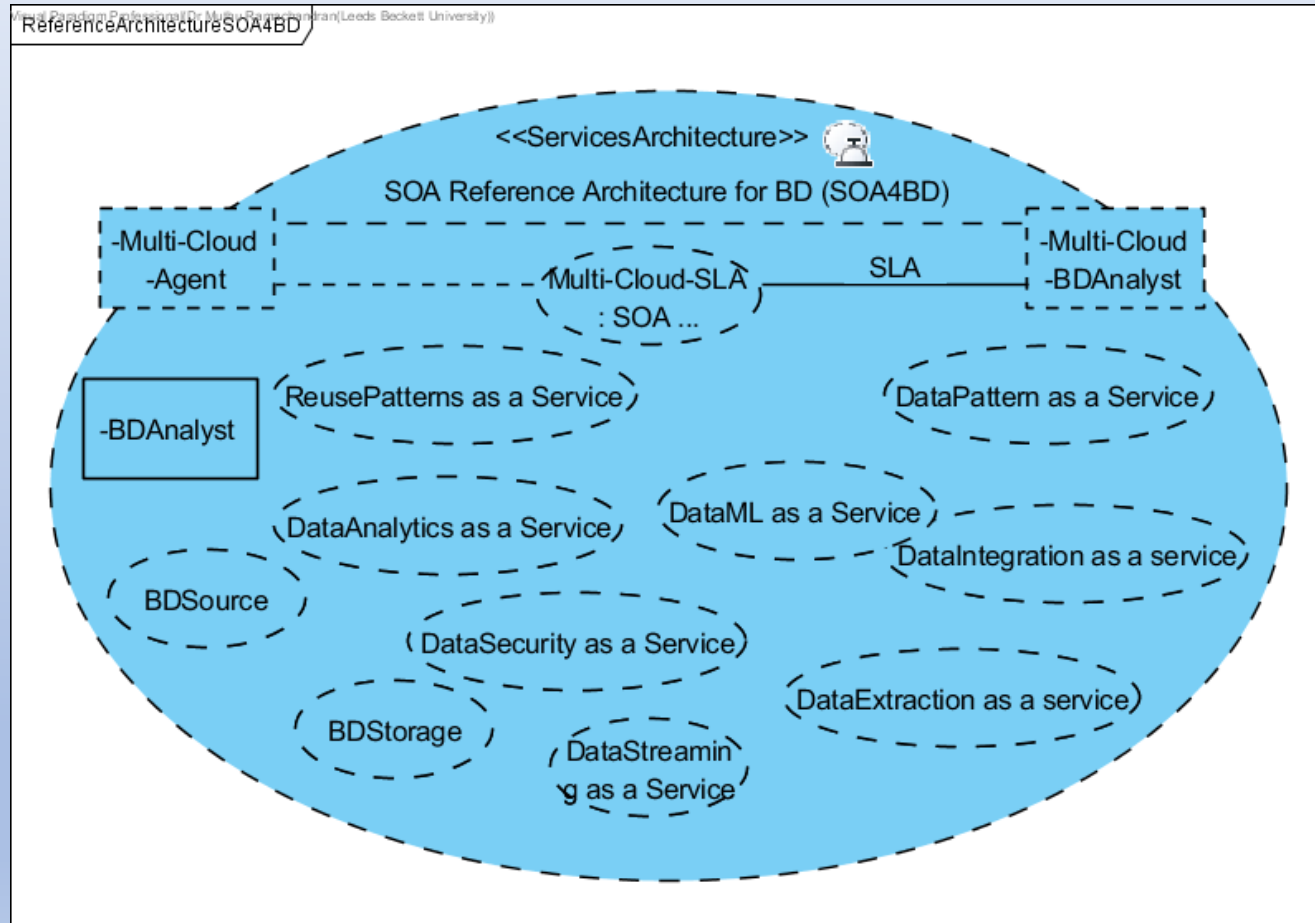
Mapping Services to SOA Design



As an Architect, you will need to categorise services therefore you will be able to place them in the appropriate architecture layers on the right



SOA Architecture (soaML design) for SEF4BD with REF4BD (Reference Architecture)

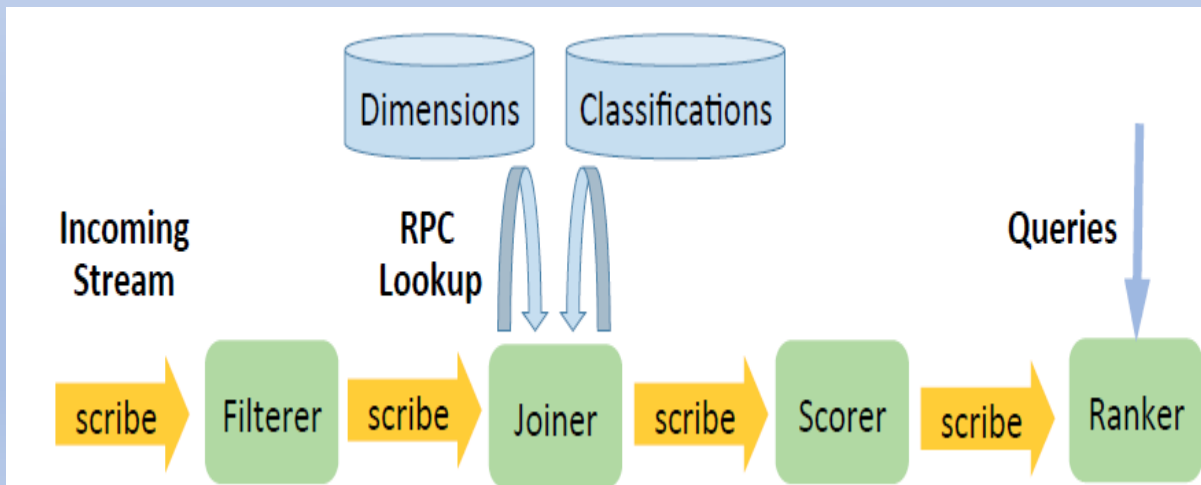
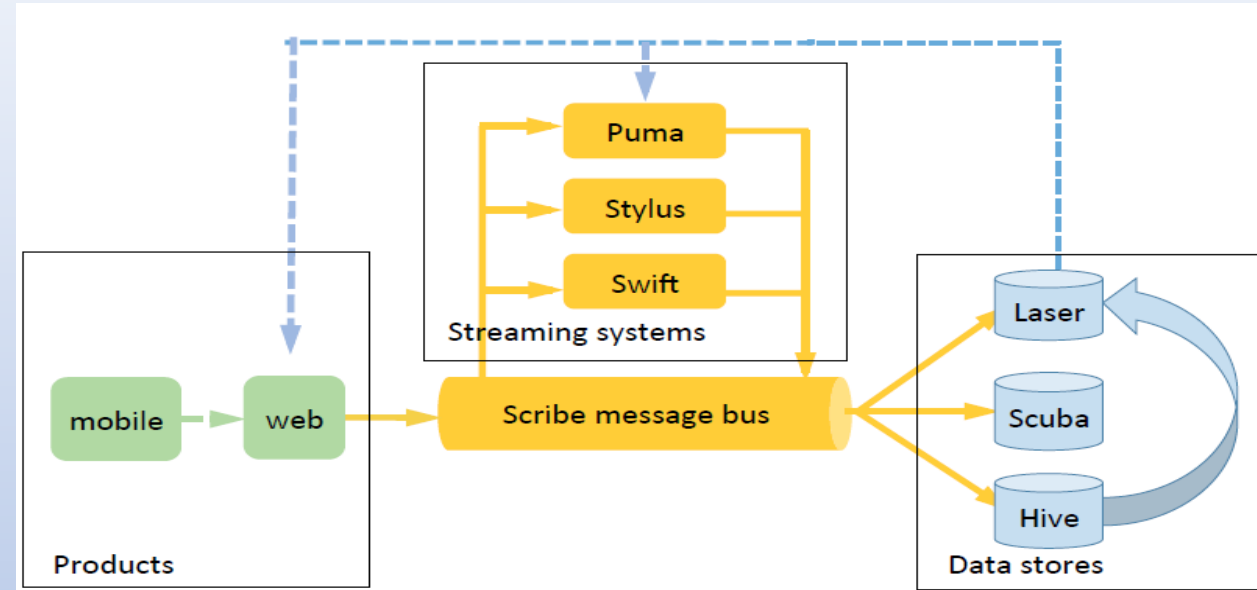


BPMN Simulation with REF4BD

Facebook Real-Time Big Data Analytics

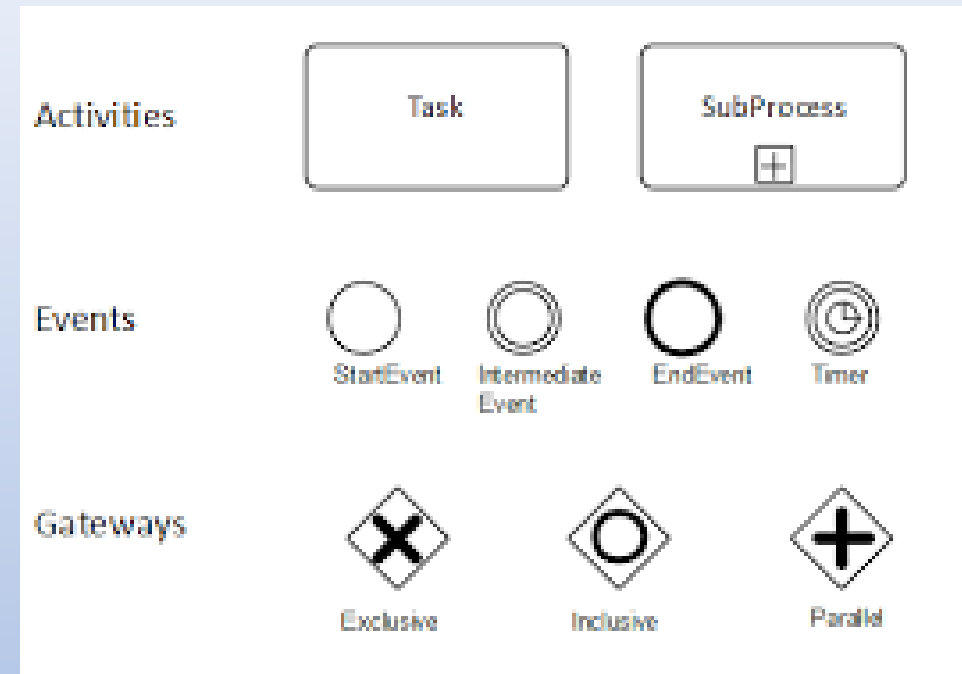
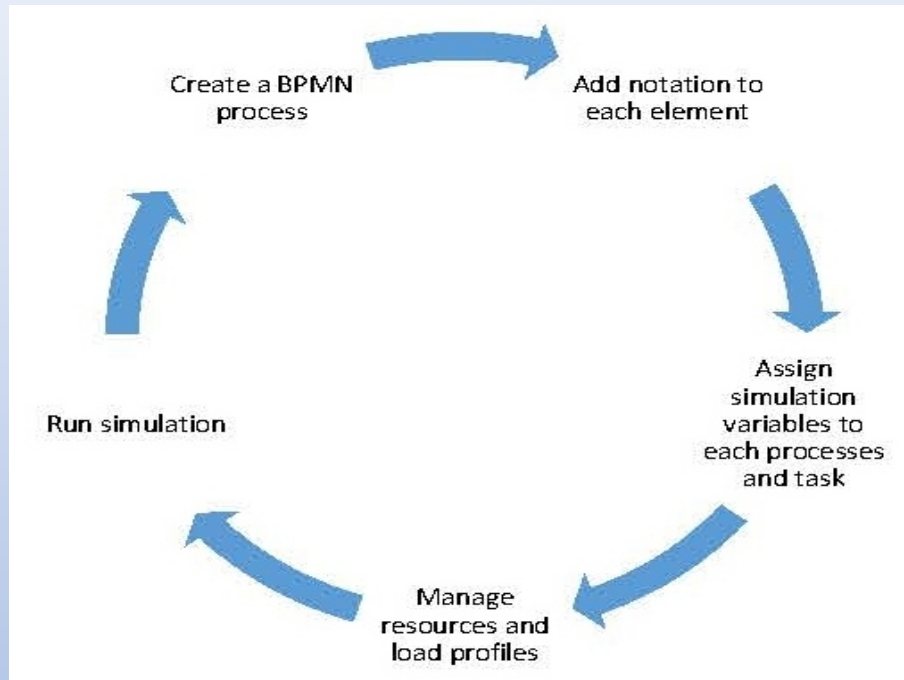
Facebook Real-time streaming services

Many companies have developed their own systems: examples include Twitter's Storm [28] and Heron [20], Google's Millwheel [9], and LinkedIn's Samza [4]. Facebook's used its own tools known as Puma, Swift, and Stylus stream processing systems. Facebook has identified important design decisions: **performance, fault tolerance, scalability, and correctness.**



An example streaming application with 4 nodes: this application computes "trending" events.

BPMN Framework for Validating the Reference Architecture (REF4BD)

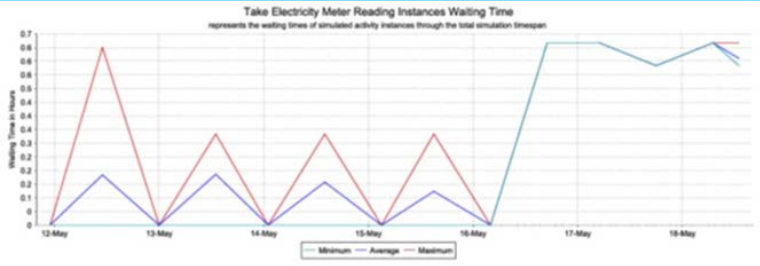
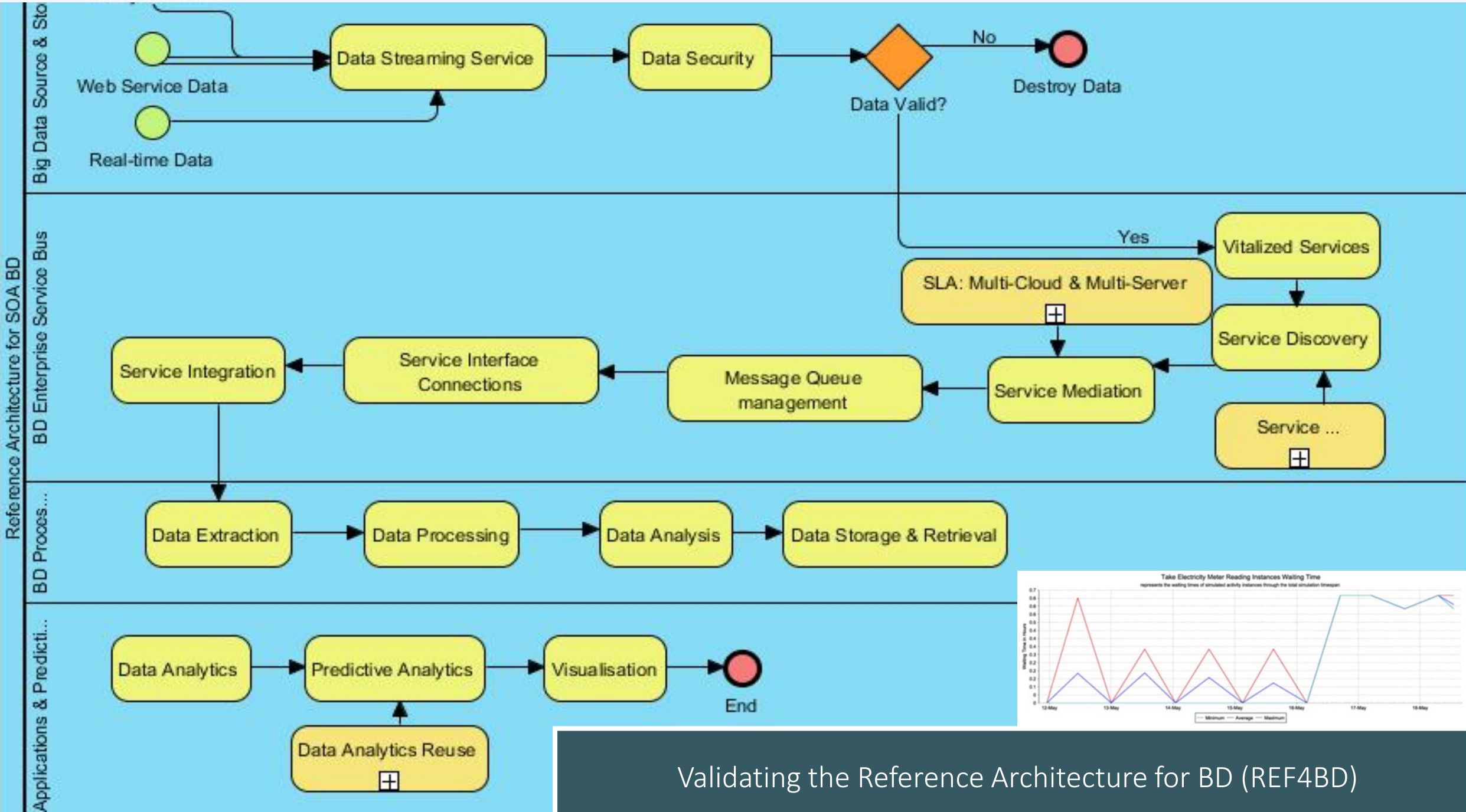


BPMN 2.0 modelling & simulation tools:

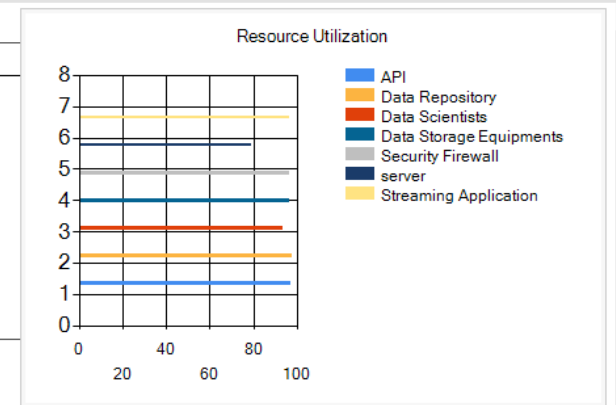
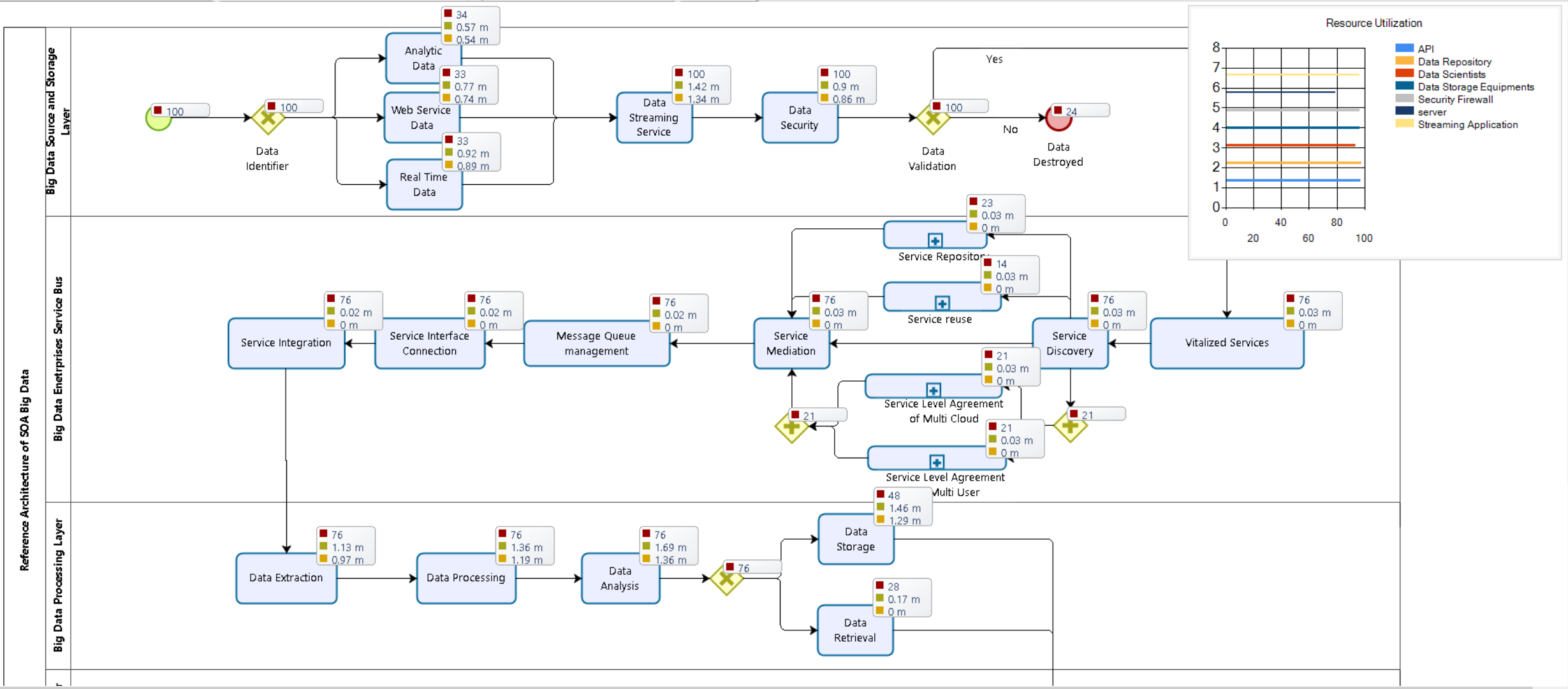
BonitaSoft 7.8, <https://www.bonitasoft.com/>

Visual Paradigm, <https://www.visual-paradigm.com/features/bpmn-diagram-and-tools/>

Bizagi Studio,
<https://www.bizagi.com/uk/products/bpm-suite/studio>



Validating the Reference Architecture for BD (REF4BD)



Results, Analysis, and Conclusion

- The results shows number of times a particular business service used to process that data, and time taken.
- In addition, BPMN 2.0 also shows a number of time each resource have been used such as API, Data Scientist (Human Tasks in BPMN), Data Repository, Servers, Firewall, Data Storage, etc.
- The results shows by implementing Facebook types of big data processing into REF4BD is more secure and uses resources efficiently than suing non-standard architectures. The efficiency result shows about 95% use of automated processing by API and Data Application (Service Components) services.
- Compared to Facebook streaming application which uses more filters which has extra-overheads and resources required whereas REF4BD is more predictable, and can achieve correctness, fault-tolerance, and scalability since it is standardised across all data process applications and services.

Summary and Questions

- SOA has emerged based on established software design principles of find-request-service paradigm suitable for service-oriented applications such as big data processing and analytics. Therefore, it is time to consider systematic and engineering approach to developing and deploying big data services as the data-driven applications and devices increasing rapidly.
- In this context, this paper proposed a software engineering framework and a reference architecture which is SOA based for big data applications' development. This paper also concluded with a simulation of a complex big data Facebook application with real-time streaming using BPMN simulation to study the characteristics before big data service design, development, and deployment. The simulation results demonstrated the efficiency and effectiveness of developing big data applications using the reference architecture framework for big data.
- To be sustainable, we need an approach which is systematic, business-driven (supporting business-process and value driven), and based on established Software Engineering practices

References

Gortan, I., Bener, A., and Mockus, A (2016) Software Engineering for Big Data Systems, Special Issue, IEEE Software, March/April 2016

Chen, G. J (2016) Real-time Data Processing at Facebook, ACM SIGMOD 2016 San Francisco, CA USA

SAKET NAVLAKHA AND ZIV BAR-JOSEPH (2015) Distributed Information Processing in Biological and Computational Systems, COMMUNICATIONS OF THE ACM | JANUARY 2015 | VOL. 58 | NO. 1

Big data of complex networks / edited by Matthias Dehmer, Frank Emmert-Streib, Stefan Pickl, Andreas Holzinger

Alessandro Fontana, Borys Wrobel (2013) Evolution and development of complex computational systems using the paradigm of metabolic computing in Epigenetic Tracking, Wivace 2013 - Italian Workshop on Artificial Life and Evolutionary Computation

Sommerville, I (2016) Software Engineering, 10th edition, Pearson

BCS (2004) The Challenges of Complex IT Projects, The report of a working group from The Royal Academy of Engineering and The British Computer Society

Ramachandran, M (2008) Software Components: Guidelines and Applications, Nova Science Publications

Ramachandran, M (2012) Software Security Engineering, Nova Science Publications

Ng, I et al (Eds) (2011) Complex Engineering Service Systems: Concepts and Research, Springer, London

Zanetti, S.M (2013) A COMPLEX SYSTEMS APPROACH TO SOFTWARE ENGINEERING, DSc thesis, ETH ZURICH

Jin, X., et al (2015) Significance and Challenges of Big Data Research, Big Data Research (2015) 59–64 <http://dx.doi.org/10.1016/j.bdr.2015.01.006>

Caldarelli, G and Vespignani, A (eds) (2007) Large Scale Structure and Dynamics of Complex Networks from information technology to finance and natural science, World Scientific Publishing Co. Pte. Ltd.

Cao, L.B (2017) Data Science: Challenges and Directions, COMMUNICATIONS OF THE ACM, 60 (8), August

Cao, L.B (2015). Metasynthetic Computing and Engineering of Complex Systems. Springer-Verlag, London, U.K.

Cady, F (2017) Data Science Handbook, Wiley

Wolfgang Karl Härdle, Henry Horng Lu, Xiaotong Shen, Shen Xiaotong eds. (2017) Handbook of Big Data Analytics

Bühlmann, Peter, et al. eds. (2017) Handbook of big data, CRC/Chapman & Hall

Albert Y. Zomaya, Sherif Sakr eds. (2017) Handbook of Big Data Technologies, Springer

Bernard Marr (2017) Data Strategy: How to Profit from a World of Big Data, Analytics and the Internet of Things, Kogan

Mohammed M. Alani, Hissam Tawfik, Mohammed Saeed, Obinna Anya (2018) Applications of Big Data Analytics: Trends, issues, & Challenges, Springer

Deng, Julia, Savas, Onur (2017) eds. Big Data Analytics in Cybersecurity, CRC

Tajunisha N., Sruthika P. (2016) Handbook On Big Data Analytics, Amazon Digital Services LLC, 2016

