FACULTY OF SCIENCE AND ENGINEERING

School Of Computing

COMP8760: Enterprise Application Integration

Group Assessment

Semester 1, 2024

Assignment 3 Part A

Group 2	Student ID
Safal Kafle	47474548
Jacqueline Dela Rosa	47901306
Ojaswi Kafle	47987731
Shila Parajuli	47530960

Table of Contents

Executive Summary

In a strategic move to boost operational efficiency and strengthen its market position, CAKE LABS, a subsidiary of Sysco and a leading provider of integrated software solutions for the restaurant industry, embarked on a transformative project by integrating AdroitLogic UltraESB into its main operational systems. This decision was made to handle the volume of daily transactions and tackle issues related to system scalability, security, and data management. By integrating cloud technology into its platform used by more than 600,000 restaurants CAKE LABS has enhanced the dining experience, for both restaurant owners and customers alike.

The primary objective of deploying AdroitLogic UltraESB was to centralize authentication processes, enhance the management of API requests, and streamline data manipulation and payload aggregation across various application interfaces, including mobile and web platforms. The project was catalyzed by the need to handle approximately three million daily API requests efficiently, which vary in complexity and size and are critical for the seamless operation of point-of-sale systems, guest management, and other integrated services offered by CAKE LABS.

The integration has led to significant improvements across several key business areas:

- **Increased Revenue:** The improved effectiveness and rapidity of handling transactions have empowered restaurant customers to boost their sales figures during busy periods, directly leading to increased revenue.
- **Cost Avoidance**: By automating and centralizing critical processes, CAKE LABS has notably decreased the expenses related to interventions and system upkeep, thereby reducing operational costs.
- **Improved Service Quality:** The reliability of the new system guarantees increased uptime and quicker response times, enriching user satisfaction and experience.

1. Introduction

CAKE LABS, an innovative leader in the restaurant software solutions industry and a notable subsidiary of Sysco has consistently aimed to harness technology to streamline operations and enhance the dining experience. Recognizing the evolving demands of over 600,000 restaurant clients, ranging from small family-run establishments to major chains and fine dining venues across the United States, CAKE LABS has committed to continual improvement of its technological infrastructure. The integration of AdroitLogic UltraESB into CAKE LABS' systems marks a pivotal advancement in the company's strategy to optimize its service offerings through cutting-edge software solutions.

The primary purpose of this report is to provide a comprehensive analysis of the reengineering processes undertaken by CAKE LABS through the integration of the AdroitLogic UltraESB. It aims to detail the motivations behind the upgrade, the challenges the integration seeks to address, and the subsequent benefits realized by both CAKE LABS and its clients. Furthermore, the report serves as a documented reflection on the project's execution and outcomes, offering insights that may guide future technological implementations and strategic decisions.

The scope of this report covers the systematic process of integrating AdroitLogic UltraESB into CAKE LABS' existing infrastructure, focusing on the adjustments made to the core operational systems and the resultant enhancements in service delivery. The scope extends to evaluating the strategic impact of integration on business operations, customer satisfaction, and financial performance (Chi and Gursoy, 2009). It also examines the scalability, security, and efficiency improvements that align with broader business objectives and customer needs (Chi and Gursoy, 2009).

Rationale for AdroitLogic UltraESB Integration

CAKE LABS faced significant challenges in handling an ever-increasing volume of data transactions driven by its diverse service offerings, including CAKE POS (Point of Sale), CAKE Guest Manager (Waitlist Management), and CAKE Connect (Reporting and Analytics). The existing system's limitations in scalability, security, and efficiency hindered CAKE LABS' ability to effectively manage three million daily API requests, impacting customer service and operational reliability.

The integration of AdroitLogic UltraESB was strategically chosen to address these challenges through its robust capabilities in high-performance request handling, flexible integration, and superior data manipulation. This middleware solution not only promised to enhance the overall architecture but also to bring about significant improvements in process automation, data security, and system responsiveness—core aspects critical to maintaining CAKE LABS' competitive edge in a fast-paced industry.

2. Project Context

CAKE LABS operates in the restaurant industry, providing a cloud-based software platform that transforms restaurant management with features like point-of-sale systems, guest management, and data analytics. Based in Redwood City, California with operations in Colombo, Sri Lanka, and Austin, Texas, CAKE LABS blends innovation from Silicon Valley with engineering expertise from, around the world.

CAKE LABS, an innovative subsidiary of Sysco, the world's largest food service company, is a Fortune 500 entity that caters to over 425,000 restaurants globally with food products, kitchen supplies, and more. Within this network, CAKE LABS plays a role by offering cutting-edge technological tools aimed at enhancing efficiency and customer satisfaction in the restaurant industry through advanced cloud computing solutions. These solutions are tailored to bring about a change. The platform provides an approach to restaurant management by integrating all aspects seamlessly.

2.1 Significant Processes Affected

The implementation of UltraESB had an impact on the authentication, authorization, and API management procedures.

- **Authentication and Security:** The centralization of authentication processes was crucial for ensuring secure and consistent access controls across platforms.
- **API Management:** The optimization of how API requests are processed, directed, and overseen aimed to enhance response times and the reliability of services.
- **Data Manipulation:** Improving the capability to conduct real-time data conversions and combine payloads was essential for tailored customer interactions and operational reporting.

These operations formed the core framework of CAKE LABS' services influencing everything from handling transactions at the point of sale, to providing real time data streams that shape customer service decisions.

2.2 Importance - IRACIS Framework

- **Increased Revenue:**
  - Enhanced Service Delivery: Faster and more reliable system responses improve customer satisfaction, encouraging repeat business and potentially increasing patronage (Wysocki, 2019).
  - Upselling Opportunities: With more reliable data and system performance, restaurants can better implement promotional activities and upsell during peak times, directly boosting revenue (Wundengba, 2020).

- **Avoidance of Costs:** The centralized handling of authentication and API requests reduced the need for redundant system components and decreased the likelihood of costly data breaches (Wysocki, 2019).
  - Reduced Downtime: The robustness of UltraESB minimizes system outages, thus avoiding revenue loss due to operational disruptions.
  - Efficiency Gains: Streamlined processes reduce the need for manual intervention, lowering labor costs and operational overhead (Wundengba, 2020).

- **Improved Service:** Implementation of a robust, scalable solution enhanced the overall user experience by reducing downtimes and service interruptions.
    - Customer Experience: Quick and accurate service, from order placement to payment processing, enhances the overall customer experience (Wundengba, 2020).
    - Operational Accuracy: Improved data handling and transaction processing accuracy reduce errors, ensuring that both customer orders and inventory management are handled flawlessly (Wysocki, 2019).

2.3 Deliverables

- **Integrated UltraESB Platform**: A fully functional enterprise service bus that centralizes and manages all API interactions, data security, and system communications (Suzic, 2016).
- **Updated System Documentation**: Comprehensive documentation detailing the new system architecture, configurations, and operational guidelines (Karaman et al., 2012).
- **Training Programs**: Training sessions and materials for system users and IT staff to ensure they can fully leverage the new system capabilities (Karaman et al., 2012).

2.4 Metrics

- **System Uptime/Availability**: It measures how often the system is up and running smoothly for users without any downtime (Lee et al., 2007).
- **Response Time**: This involves keeping track of how it takes for the system to handle requests to improve upon benchmarks, after integration (De Langhe and Puntoni, 2016).
- **Customer Satisfaction Scores**: Regular surveys are conducted to assess user satisfaction regarding the system's reliability and responsiveness (Lee et al., 2007).
- **Cost Savings**: Analyzing the savings achieved through operations and reduced manual work (Rust, Moorman and Dickson, 2002).
- **Revenue Growth**: Monitoring changes in revenue before. After integration, evaluate the impact of service delivery and system performance (Rust, Moorman and Dickson, 2002).

These deliverables and metrics provide tangible goals for CAKE LABS to aim for during and after the integration process, ensuring the project's success is measurable and aligns with the organization's strategic objectives.

3. As-Is Process Mind Map

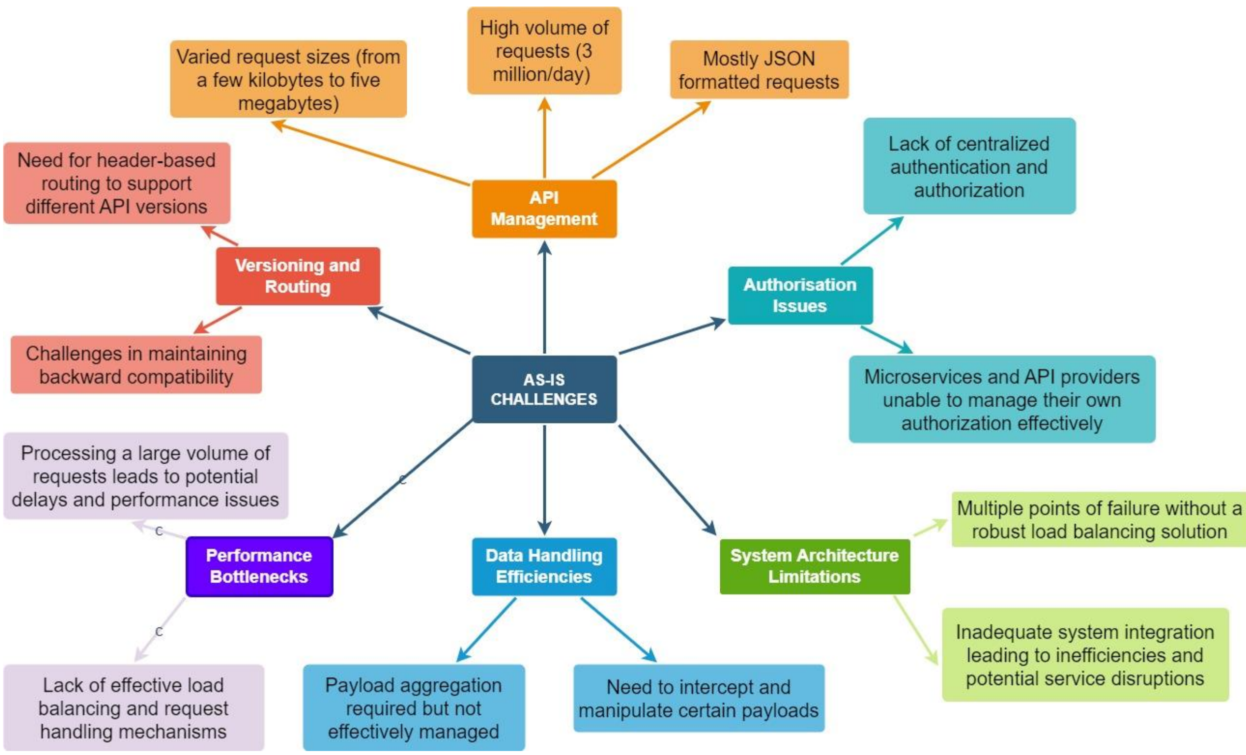

Figure 1.0 *– AS-IS Process Mind Map for CAKE LABS*

The above illustrated 'As-Is' process mind map for CAKE LABS effectively provides a comprehensive visualization of the operational challenges faced by CAKE LABS before the integration of AdroitLogic UltraESB. Each mind map node represents a critical area where inefficiencies and bottlenecks were prevalent, significantly impacting overall system performance and user experience. The following detailed elaboration further explores these nodes to provide a deeper understanding of the existing challenges:

3.1 API Management Challenges

High Volume of Requests: Processing approximately three million daily API requests puts immense pressure on CAKE LABS' infrastructure. The systems in place were frequently overwhelmed during peak usage times, leading to slow processing speeds and increased transaction failure rates, which directly affected restaurant operations and customer service.

Varied Request Sizes: The variation in request sizes from a few kilobytes to as much as five megabytes necessitated a flexible and powerful processing capability. The existing infrastructure struggled to handle large data payloads efficiently, often resulting in increased latency and higher rates of errors.

Versioning and Routing: Effective management of multiple API versions was a significant challenge. The system lacked sophisticated routing mechanisms, complicating the delivery of services across different versions, and leading to compatibility issues. This not only made system upgrades cumbersome but also risked potential data mismatches and operational inconsistencies.

3.2 Authorization Issues

Lack of Centralized Authentication and Authorization: The decentralized approach to authentication and authorization was a major vulnerability. It exposed the system to security risks, such as unauthorized access and data breaches, as consistent security protocols were not enforced across all interfaces and services.

Microservices and API Providers: The inability of microservices and individual API providers to manage their authorization mechanisms effectively created gaps in security and operational integrity. This fragmentation made enforcing uniform security measures difficult, complicating user access and data protection management.

3.3 Performance Bottlenecks

Handling Large Volumes: The infrastructure's inability to efficiently process high volumes of requests led to bottlenecks that severely impacted user experience. Delays in processing and performance issues were frequent, undermining the reliability of CAKE LABS' software solutions.

<u>Ineffective Load Balancing</u>: Before the integration, the system did not employ effective load balancing strategies, resulting in uneven server load distribution. This often led to server overloads and downtimes, detrimental to service availability and responsiveness.

3.4 Data Handling Inefficiencies

<u>Payload Aggregation</u>: Essential for operational reporting and data analytics (Limmer and Dressler, 2009), payload aggregation was poorly managed. This inefficiency not only reduced the accuracy of the data insights provided to restaurant owners but also hindered the responsiveness of the system to real-time data queries.

<u>Need to Intercept and Manipulate Payloads</u>: The existing system's limited capability to intercept and manipulate data payloads restricted the flexibility to customize responses based on dynamic data conditions. This was particularly problematic in adapting to user-specific needs and in performing real-time data adjustments.

3.5 System Architecture Limitations

<u>Multiple Points of Failure</u>: The design had vulnerabilities due to insufficient load-balancing measures. This raised the likelihood of system failures causing disruptions to operations and putting data at risk.

<u>Inadequate System Integration</u>: The fragmented approach to integrating the system resulted in inefficiencies and service interruptions. Inadequate coordination among software components and services hindered data transmission and communication leading to delays in operations and jeopardizing data accuracy.

3.6 Implications of As-Is Pain Points

The operational challenges identified in the 'AsIs' process mind map for CAKE LABS had wide-ranging implications that affected various aspects of the business, from technical performance to customer satisfaction and financial stability. These implications underscore the necessity of the technological overhaul provided by the integration of Adroit Logic UltraESB. Below is an in-depth analysis of the effects these pain points had on the organization:

3.6.1 Impaired Customer Experience

Service Delays and Disruptions: The inability to efficiently process high volumes of data, especially during peak periods, often resulted in significant service delays. For a business operating in the fast-paced restaurant industry, any delay can lead to a direct loss of sales as customers face long waiting times or choose alternate venues. This not only impacts immediate revenue but also affects the reputation of the establishments using CAKE LABS' solutions.

Inconsistent User Experience Across Platforms: Inadequate API management and version control resulted in varying experiences on platforms such as mobile and web. This lack of consistency might puzzle users, affecting their satisfaction levels and possibly resulting in customer retention rates and unfavorable reviews (Kleinberg, Sendhil Mullainathan and Raghavan, 2023).

3.6.2 Operational Inefficiencies

High Maintenance and Operational Costs: Managing a decentralized authentication system and handling multiple versions of APIs without proper routing mechanisms required substantial manual effort and resources. This not only increased operational costs but also diverted technical resources away from innovation and improvement, stifling growth and adaptation to new market demands.

Resource Drain on IT Departments: Frequent system downtimes and the need for manual interventions to manage load balancing and data integrity issues placed a significant burden on IT staff. The constant need to troubleshoot and maintain the existing system could lead to burnout and reduced productivity among the technical staff.

3.6.3 Security Vulnerabilities

Increased Risk of Data Breaches: The decentralized way of verifying identities and the lack of API security measures made the system vulnerable to unauthorized access and data breaches. In today's world, where safeguarding data privacy and security is crucial, these weaknesses could result in not only financial harm but also legal consequences and harm to the company's image.

Compliance Risks: There is also a risk of noncompliance with industry rules and regulations like GDPR or PCI DSS (Elluri, Nagar and Joshi, 2018), for CAKE LABS if they do not uphold a security protocol depending on the type of data they handle and store.

3.6.4 Hindered Business Growth

Scalability Constraints: The existing system's limitations in handling varied request sizes and high volumes of transactions restricted CAKE LABS' ability to scale its operations efficiently. This hindered the company's ability to expand its customer base or add new functionalities without risking further degradation of service quality.

Lost Business Opportunities: Operational and service delivery issues could lead to lost business opportunities, as potential new clients might opt for competitors with more reliable and efficient systems. Existing customers might also reconsider renewing contracts if they perceive that the technology does not meet their growing needs.

These identified pain points highlight the critical operational challenges faced by CAKE LABS. The implications of these issues were multifaceted, affecting not only the technical performance but also customer satisfaction and operational costs. The necessity for a scalable, secure, and efficient system was evident, driving the decision to integrate AdroitLogic UltraESB. This integration aimed to address these foundational challenges by enhancing the system architecture, streamlining API management, improving data handling capabilities, and consolidating authentication processes into a centralized framework.

The detailed understanding of these 'As-Is' pain points, as elaborated from the mind map, provided a clear directive for the subsequent reengineering processes. Each enhancement was strategically planned to mitigate these issues, thereby ensuring a more robust, responsive, and efficient operational infrastructure moving forward.

4. To-Be Process Model

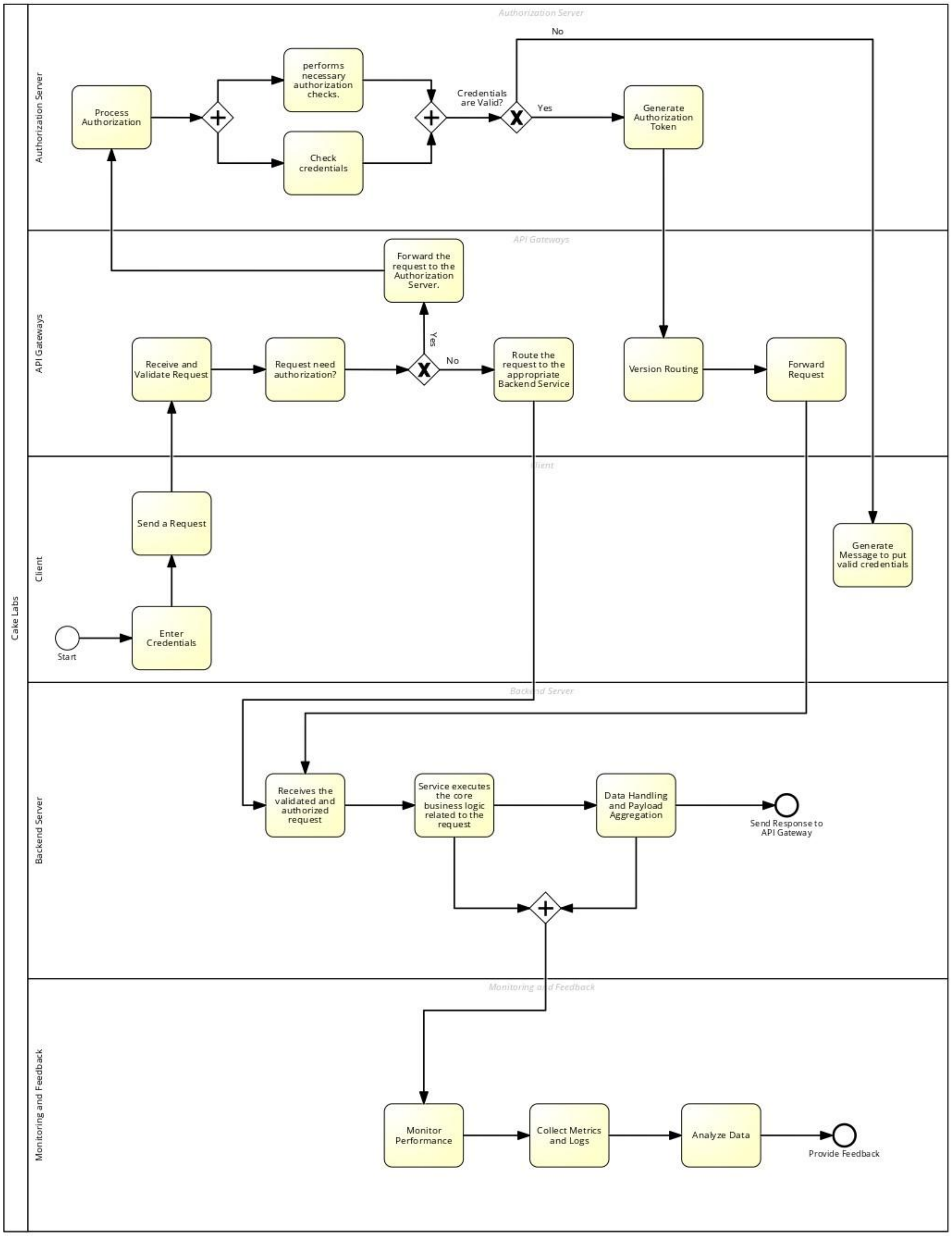

Figure 2.0 – *BPMN 2.0 for CAKE LABS*

4.1 Process Improvements Commentary

Post-integration, CAKE LABS enjoyed streamlined operations with a centralized authentication mechanism and more effective API management capabilities, enabling:

Centralized Authentication and Authorization:

Before: Decentralized systems leading to inconsistent security practices and potential vulnerabilities.

After: Implementing a centralized authentication system guarantees security measures across all platforms thereby minimizing the chances of entry and safeguarding against data breaches (Kim and Lee, 2017).

Efficient API Management

Before: Struggled with high volumes of varied requests, causing delays and failures.

After: The implementation of UltraESB facilitates the smooth handling of up to three million requests per day, with dynamic load balancing ensuring even distribution and processing efficiency (Suzic, 2016). Header-based routing allows seamless management of multiple API versions, ensuring backward compatibility and smoother transitions during upgrades (Suzic, 2016).

Enhanced Data Handling

Before: Inefficiencies in payload aggregation and manipulation, leading to data integrity issues.

After: UltraESB's robust data manipulation capabilities ensure real-time, accurate processing and aggregation of data, improving the quality of insights and operational decision-making (Suzic, 2016).

Improved System Scalability and Reliability

Before: Limited scalability and multiple points of failure, risking frequent downtimes.

After: The architecture now includes an active configuration with UltraESB instances and Elastic Load Balancer (Rahman, Iqbal and Gao, 2014), significantly enhancing the system's ability to scale dynamically and ensuring high availability and fault tolerance (Iyer, Gupta and Johri, 2005).

Continuous Monitoring and Feedback

Before: Insufficient monitoring and feedback systems resulted in delays in identifying and addressing issues promptly.

After: The implementation of monitoring and feedback mechanisms enables the real-time tracking of performance detection of obstacles and proactive resolution of problems. This enhances system dependability and facilitates the enhancement of procedures (Lopez-Pena et al., 2020).

Comprehensive Load Balancing

Before: There were issues with the system due to a lack of load balancing, which could cause interruptions in services.

After: The Elastic Load Balancer helps spread out traffic among UltraESB instances making sure each instance handles a fair share of the workload and reducing the chances of one instance getting overwhelmed. This setup boosts reliability during failures and keeps services up and running smoothly. (Rahman, Iqbal and Gao, 2014).

Seamless Integration and Interoperability

Before: Issues with the system integration caused disruptions in services. Made operations less efficient.

After: The smooth blending enabled by UltraESB aids in accommodating data formats and communication standards fostering interaction among diverse services. This promotes collaboration and harmony, among all elements minimizing interruptions, in integration processes and boosting effectiveness overall (Chen, Doumeingts and Vernadat, 2008).

Improved Security and Compliance

Before: Security measures vary widely among systems posing difficulties in meeting industry regulations and standards.

After: Centralized authentication and improved security protocols, CAKE LABS guarantees compliance with industry regulations and standards creating a space for data transactions. This helps minimize the chances of compliance issues and related penalties.

Enhanced User Experience

Before: Issues, with delays and variations, in how API responses are delivered have impacted user contentment.

After: The enhanced API. Efficient data processing results, in response times and increased reliability ultimately elevating the user experience (Suzic, 2016). Customers enjoy more dependable interactions, with the platform resulting in satisfaction and improved customer retention.

Proactive Incident Management

Before: A responsive method, to handling incidents with challenges, in identifying and resolving problems.

After: The system for monitoring and feedback enables the management of incidents. It helps in detecting and resolving problems before they affect users, reducing downtime, and ensuring service quality is upheld.

Robust Payload Manipulation

Before: The mishandling of the payload resulted in data processing and difficulties with integration.

After: UltraESB’s advanced features, for handling data payloads enable the transformation of data ensuring that incoming information is consistently structured and accurately handled (Limmer and Dressler, 2009). This helps maintain data reliability and facilitates integration between systems.

Service Delivery

Before: Inconsistent service delivery was caused by processes and inefficiencies.

After: The efficient and dependable service delivery is guaranteed by the organized procedures and strong system structure. CAKE LABS' clients benefit from a top-notch service experience thanks to the API management load balancing and real-time data processing.

The To-Be process model does not only tackle the issues found in the as-is process but also sets the groundwork for a stronger, expandable, and more effective system at CAKE LABS. By centralizing authentication, optimizing API management, improving data handling, and implementing monitoring and load balancing mechanisms CAKE LABS is now ready to manage its increasing demands and offer top-notch service to its customers. The incorporation of UltraESB has changed how operations are handled ensuring that CAKE LABS stays adaptable, secure, and able to support long-term growth. These enhancements establish a foundation for innovations and growth opportunities positioning CAKE LABS as a frontrunner in the restaurant software sector.

4.2 Justification for IRACIS

Increased Revenue

Enhanced Transaction Processing: By enhancing the processing of the high volume of transactions restaurants can swiftly and precisely handle orders resulting in heightened sales, during busy periods.

Better Customer Experience: Providing quicker and dependable service fosters customer loyalty. Increases the likelihood of repeat business ultimately driving revenue growth.

Avoidance of Costs

Reduced Downtime: The robust architecture minimizes system outages, thereby avoiding revenue loss and costs associated with downtime.

Operational Efficiency: Automating data handling and API management reduces the need for manual interventions, lowering labor costs and freeing up resources for strategic initiatives.

Improved Service

Consistent Security and Data Integrity: Centralized authentication and efficient data processing ensure that all interactions are secure and reliable, enhancing overall service quality.

User Satisfaction: Faster response times and seamless service delivery across platforms improve user satisfaction, fostering loyalty and positive word-of-mouth.

5. Implementation

5.1 SaaS Functionality Description

AdroitLogic UltraESB mediates incoming API calls to backend systems, offering robust authentication, payload manipulation, and API version management. This top-tier SOA solution integrates, develops, deploys, and manages multiple services on a common platform, moderating costs, accelerating IT implementations, and reducing complexity. Organizations can align business needs by selecting suitable ESBs for specific service domains and reusing applications via web services. UltraESB provides a rich, interactive environment for developers to build federated services rapidly without detailed J2EE or Java knowledge. It supports virtualization, extends SOA functionalities, and offers easy mediation, extensive connectivity, an integrated development framework, single-point manageability, flexibility, performance optimization, business adaptability, secure routing, and API adaptability (Bhadoria, Chaudhari and Tomar, 2017).

5.2 Interfaces and Integration

As depicted in Figure 3.0, applications are connected indirectly via an Enterprise Service Bus (ESB) rather than being directly linked to one another. The ESB handles all the inherent logic required for the structures to interact and integrate (Aziz et al., 2020).

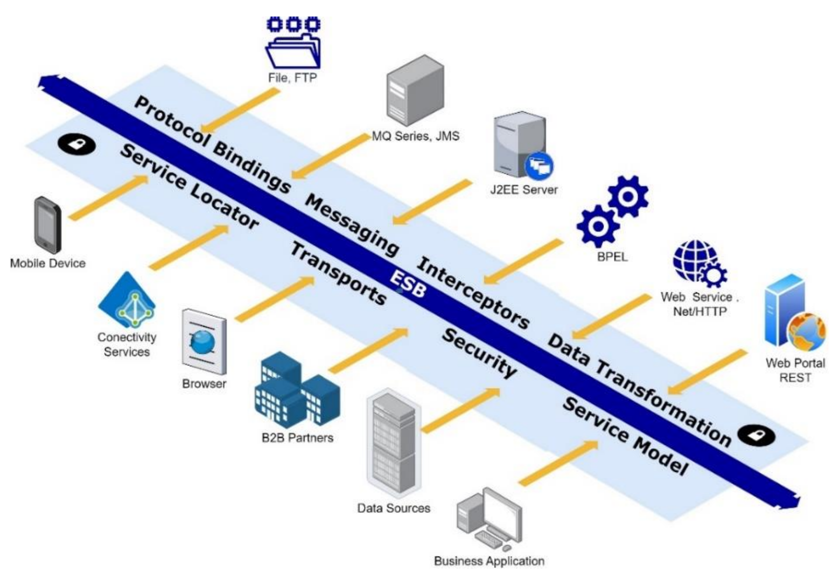

***Figure 3.0** – ESB Architecture (Aziz et al., 2020).*

This proposed ESB architecture enables CAKE LABS to manage its complex and high-volume transactional environment effectively, supporting its strategic objectives of operational efficiency and enhanced service delivery.

The objectives include centralizing authentication processes, enhancing API request management, streamlining data manipulation and payload aggregation across various application interfaces (mobile and web platforms), efficiently handling approximately three million daily API requests, and ensuring seamless operation of point-of-sale systems, guest management, and other integrated services.

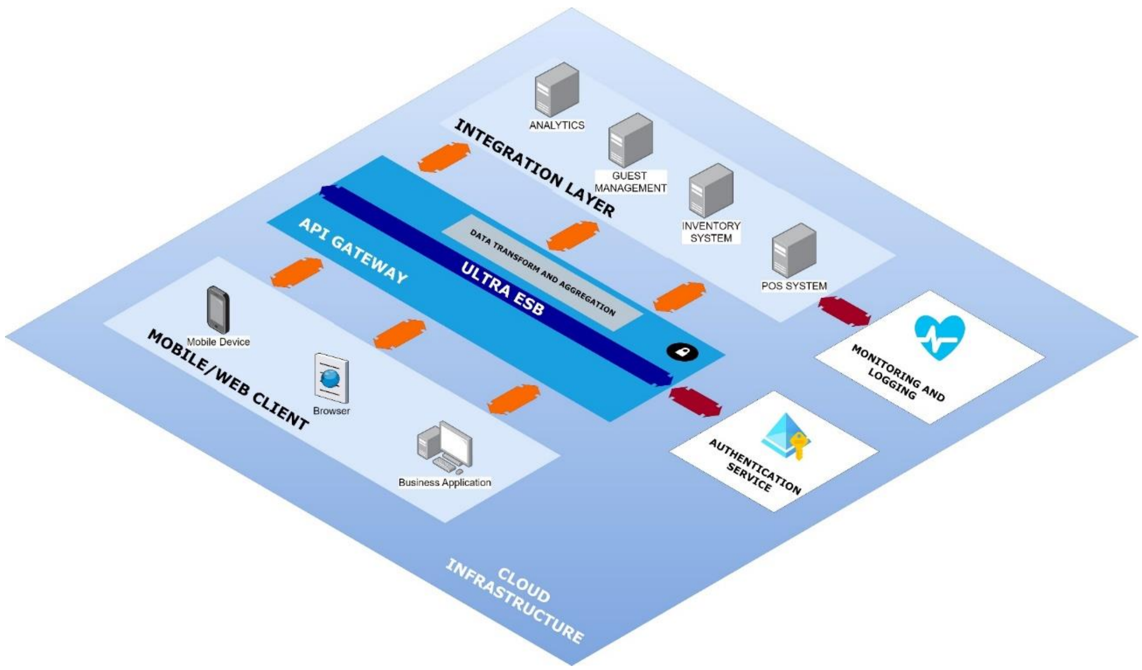

Figure 4.0 *Proposed Architecture ESB for Cake Labs*

These external interfaces interact with the system through the API Gateway.

- Mobile Applications
- Web Applications
- Third-party Integrations

5.2.1 Architecture Components

AdroitLogic UltraESB: The core component responsible for managing and routing API requests, providing centralized authentication, and handling data transformation.

- Service Orchestration: Coordinates between different services to fulfill complex business processes.
- Message Transformation: Converts messages between different formats as required by different services.
- Protocol Mediation: Supports communication over various protocols (HTTP, JMS, etc.).

Cloud Infrastructure: Utilizes cloud technology for scalability and reliability, ensuring high availability and disaster recovery.

API Gateway: Acts as the entry point for all API requests, providing security, load balancing, and routing to appropriate services.

- Request Routing: Directs incoming requests to appropriate services.
- Load Balancing: Distributes the load across multiple instances of services.
- Security: Provides SSL termination, rate limiting, and IP whitelisting.

Authentication Service: Centralized service for handling user authentication and authorization.

- User Authentication: Handles user login and session management.
- Authorization: Ensures users have appropriate permissions for requested actions.

Data Management Layer: Manages data storage, retrieval, and manipulation, ensuring data consistency and integrity.

- Database Cluster: Manages persistent data storage with high availability and redundancy.
- Cache Layer: Provides fast access to frequently used data, reducing database load.
- Data Transformation Services: Handle complex data manipulations and aggregations.

Monitoring and Logging: Provides real-time monitoring, logging, and alerting for all system activities to ensure operational transparency and quick issue resolution.

- Monitoring Tools: Provide real-time insights into system performance and health.

- Logging Framework: Captures detailed logs of all transactions for audit and debugging purposes.
- Alerting System: Sends notifications for critical events and potential issues.

Operational Workflow

Enterprise service buses (ESBs) offer several core functionalities, including uncoupling, transport protocol conversion, high availability and scalability, message transformation, routing, and security (Aziz et al., 2020).

The process begins with API request initiation, where a request is made from a mobile or web application. The request passes through the API Gateway, which handles initial security checks and load balancing. It is then forwarded to the Authentication Service to verify user credentials. Once authenticated, UltraESB orchestrates the request, involving multiple backend services as needed. The Data Management Layer performs necessary data transformations and manipulations. A response is generated and sent back through the API Gateway to the originating application. Throughout this process, monitoring and logging are conducted for transparency and issue resolution.

5.2.2 Recommendations

Choosing the appropriate ESB applications is vital for swiftly integrating diverse applications and expediting market entry. Key considerations include usability, supportability, functionality, moldability, expandability, enterprise maintenance, community support, connector availability, cost, and licensing flexibility. By carefully evaluating these criteria, organizations can optimize their integration processes while minimizing overhead costs and ensuring scalability (Aziz, et. al, 2020).

5.3 Feedback and Monitoring Loop

The feedback and monitoring for AdroitLogic UltraESB encompasses several critical parameters. It begins with tracking message counts to evaluate communication effectiveness among service components, assessing both successful deliveries and failures. Data transfer and time complexity are monitored to optimize service interaction efficiency, focusing on state, value, and behavior using distribution functions. Monitoring adherence to ESB-provided service patterns ensures seamless cloud messaging integration. Heap memory usage is closely monitored for efficient resource handling and instance creation. Daemon thread counts are observed to support user threads effectively, while CPU load monitoring evaluates overall performance and uses load-balancing techniques. Cache memory usage in JVM is optimized to avoid unnecessary garbage collection, enhancing cloud system reliability. Continuous interface efficiency monitoring ensures optimal service handling across network/cloud environments, fostering ongoing performance improvements (Bhadoria, et. al, 2018).

Parameters	Description
Message Count (Successful)	Communication Messages float among service components
Message Count (Failed)	Number of failed messages
Message Count (Total)	Total Number of messages
Active Threads Counts	Supports in concurrency and integration of services
Daemon Thread Counts	Determine schedule task is planned or not
Minimum Response Time	Minimum time incurred in process and response back
Average Response Time	Average time incurred in process and response back
Maximum Response Time	Maximum time incurred in process and response back
Memory Allocation	Total allotment of memory
Memory Usage	Actual Usage of Memory

Figure 5.0 - *Performance analysis of the Adroit Logic Ultra ESB as detailed in (Chaudhari, Robin Singh Bhadoria and Prasad, 2016).*

According to Manik (2021), performance evaluation of enterprise service buses (ESBs) like AdroitLogic UltraESB relies on several key metrics to assess system effectiveness in delivering services. Central to this evaluation are response time, throughput, resource utilization (RU), and system availability. Response time, defined as the interval between user request and system response, is critical for measuring system responsiveness. Throughput, typically measured in Transactions Per Second (TPS), quantifies the system's processing capacity over time, reflecting its efficiency in handling transactional workloads. Resource utilization (RU) assesses the efficient use of system resources during service delivery, crucial for optimizing performance and scalability.

Meanwhile, system availability measures the percentage of time the system remains operational to meet user demands, indicating reliability and accessibility (Manik, 2022). These metrics collectively provide a comprehensive view of UltraESB's performance in terms of responsiveness, productivity, and utilization, essential for evaluating its operational efficiency and service delivery capabilities.

6. Conclusion

The deployment of AdroitLogic UltraESB, at CAKE LABS has led to enhancements in API management, data security, and overall system resilience. By tackling the challenges concerning scalability and security, CAKE LABS has not only boosted its service delivery but also established a solid platform that supports future growth and expansion. Significant accomplishments include:

- Decreased response times for API calls improving the customer experience.
- Enhanced system uptime ensuring dependable service availability.
- Strengthened security landscape through centralized authentication protocols.

The strategic incorporation of AdroitLogic UltraESB has positioned CAKE LABS as a pioneer in innovation within the restaurant management software industry. Looking ahead, it is crucial for CAKE LABS to continuously monitor, assess, and adapt its infrastructure to meet the evolving needs of its clientele, uphold its competitive edge in the market and win the race.

By following the strategies engraved in this report and persistently utilizing cutting-edge technology solutions CAKE LABS can secure growth and service excellence. This report is evidence of how technology integrations can revolutionize business operations and steer industry leadership. The report details the implementation phase with a grasp of the aspects of the project encompassing UltraESB’s functional capabilities and system architecture.

CAKE LABS' effective integration of UltraESB showcases how technology is utilized to address data management challenges, within the food sector. This development platform empowers CAKE LABS to deliver services to a clientele in the food industry ensuring business growth and ongoing prosperity.

References – TASK A1

Aziz, O., Farooq, M.S., Abid, A., Saher, R. and Aslam, N. (2020). Research Trends in Enterprise Service Bus (ESB) Applications: A Systematic Mapping Study. *IEEE Access*, 8, pp.31180–31197. doi:https://doi.org/10.1109/access.2020.2972195.

Bhadoria, R.S., Chaudhari, N.S. and Tharinda Nishantha Vidanagama, V.G. (2018). Analyzing the role of interfaces in enterprise service bus: A middleware epitome for service-oriented systems. *Computer Standards & Interfaces*, 55, pp.146–155. doi:https://doi.org/10.1016/j.csi.2017.08.001.

Bhadoria, R.S., Chaudhari, N.S. and Tomar, G.S. (2017). The Performance Metric for Enterprise Service Bus (ESB) in SOA system: Theoretical underpinnings and empirical illustrations for information processing. *Information Systems*, 65, pp.158–171. doi:https://doi.org/10.1016/j.is.2016.12.005.

Chaudhari, N., Robin Singh Bhadoria and Prasad, S. (2016). Information Handling and Processing Using Enterprise Service Bus in Service-Oriented Architecture System. *2016 8th International Conference on Computational Intelligence and Communication Networks (CICN)*. doi:https://doi.org/10.1109/cicn.2016.88.

Chen, D., Doumeingts, G. and Vernadat, F. (2008). Architectures for enterprise integration and interoperability: Past, present and future. *Computers in Industry*, 59(7), pp.647–659. doi:https://doi.org/10.1016/j.compind.2007.12.016.

Chi, C.G. and Gursoy, D. (2009). Employee satisfaction, customer satisfaction, and financial performance: An empirical examination. *International Journal of Hospitality Management*, 28(2), pp.245–253. doi:https://doi.org/10.1016/j.ijhm.2008.08.003.

De Langhe, B. and Puntoni, S. (2016). Productivity Metrics and Consumers' Misunderstanding of Time Savings. *Journal of Marketing Research*, [online] 53(3), pp.396–406. doi:https://doi.org/10.1509/jmr.13.0229.

Elluri, L., Nagar, A. and Joshi, K.P. (2018). *An Integrated Knowledge Graph to Automate GDPR and PCI DSS Compliance*. [online] IEEE Xplore. doi:https://doi.org/10.1109/BigData.2018.8622236.

Iyer, L.S., Gupta, B. and Johri, N. (2005). Performance, scalability and reliability issues in web applications. *Industrial Management & Data Systems*, 105(5), pp.561–576. doi:https://doi.org/10.1108/02635570510599959.

Karaman, A.D., Cobanoglu, F., Tunalioglu, R. and Ova, G. (2012). Barriers and benefits of the implementation of food safety management systems among the Turkish dairy industry: A case study. *Food Control*, 25(2), pp.732–739. doi:https://doi.org/10.1016/j.foodcont.2011.11.041.

Kim, H. and Lee, E. (2017). *Authentication and Authorization for the Internet of Things* .

Kleinberg, J., Sendhil Mullainathan and Raghavan, M. (2023). The Challenge of Understanding What Users Want: Inconsistent Preferences and Engagement Optimization. *Management Science*. doi:https://doi.org/10.1287/mnsc.2022.03683.

Lee, S.M., Lee, H., Kim, J. and Lee, S. (2007). ASP system utilization: customer satisfaction and user performance. *Industrial Management & Data Systems*, 107(2), pp.145–165. doi:https://doi.org/10.1108/02635570710723787.

Limmer, T. and Dressler, F. (2009). Flow-based Front Payload Aggregation. *The 4th IEEE LCN Workshop on Network Measurements (WNM 2009)*. doi:https://doi.org/10.1109/lcn.2009.5355213.

Lopez-Pena, M.A., Diaz, J., Perez, J.E. and Humanes, H. (2020). DevOps for IoT Systems: Fast & Continuous Monitoring Feedback of System Availability. *IEEE Internet of Things Journal*, 7(10), pp.10695–10707. doi:https://doi.org/10.1109/jiot.2020.3012763.

Manik, L.P. (2022). Performance Factors Effect on the Performance Metrics of the Enterprise Service Bus. *International Journal of Computing and Digital Systems*, 11(1), pp.107–115. doi:https://doi.org/10.12785/ijcds/110108.

Rahman, M., Iqbal, S. and Gao, J. (2014). *Load Balancer as a Service in Cloud Computing*. [online] IEEE Xplore. doi:https://doi.org/10.1109/SOSE.2014.31.

Rust, R.T., Moorman, C. and Dickson, P.R. (2002). Getting Return on Quality: Revenue Expansion, Cost Reduction, or Both? *Journal of Marketing*, 66(4), pp.7–24. doi:https://doi.org/10.1509/jmkg.66.4.7.18515.

Suzic, B. (2016). *User-centered security management of API-based data integration workflows*. [online] IEEE Xplore. doi:https://doi.org/10.1109/NOMS.2016.7502993.

Wundengba, C. (2020). *IRACIS – A Roadmap to Business Intelligence ROI*. [online] Wundef.com. Available at: https://wundef.com/iracis-a-roadmap-to-business-intelligence-roi/ [Accessed 16 May 2024].

Wysocki, R.K. (2019). *Effective Project Management: Traditional, Agile, Extreme.* 8th ed. Wiley.

Task A2: Trello Design

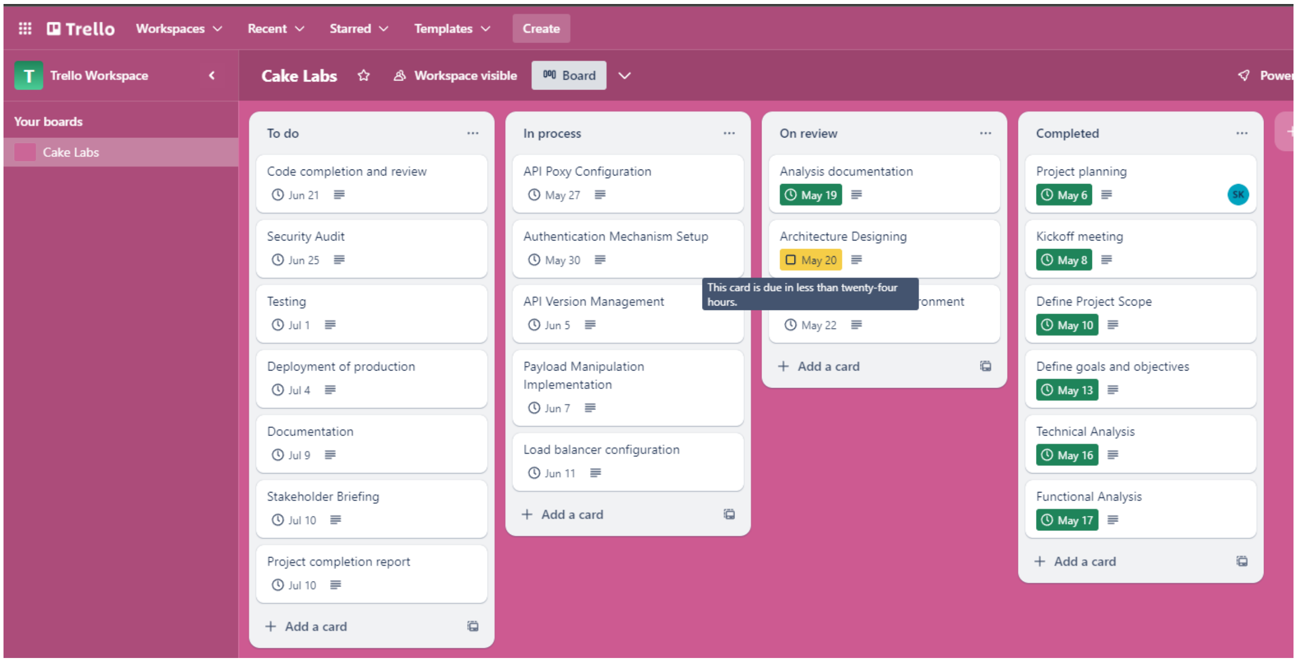

Fig 1: Trello Board Overview

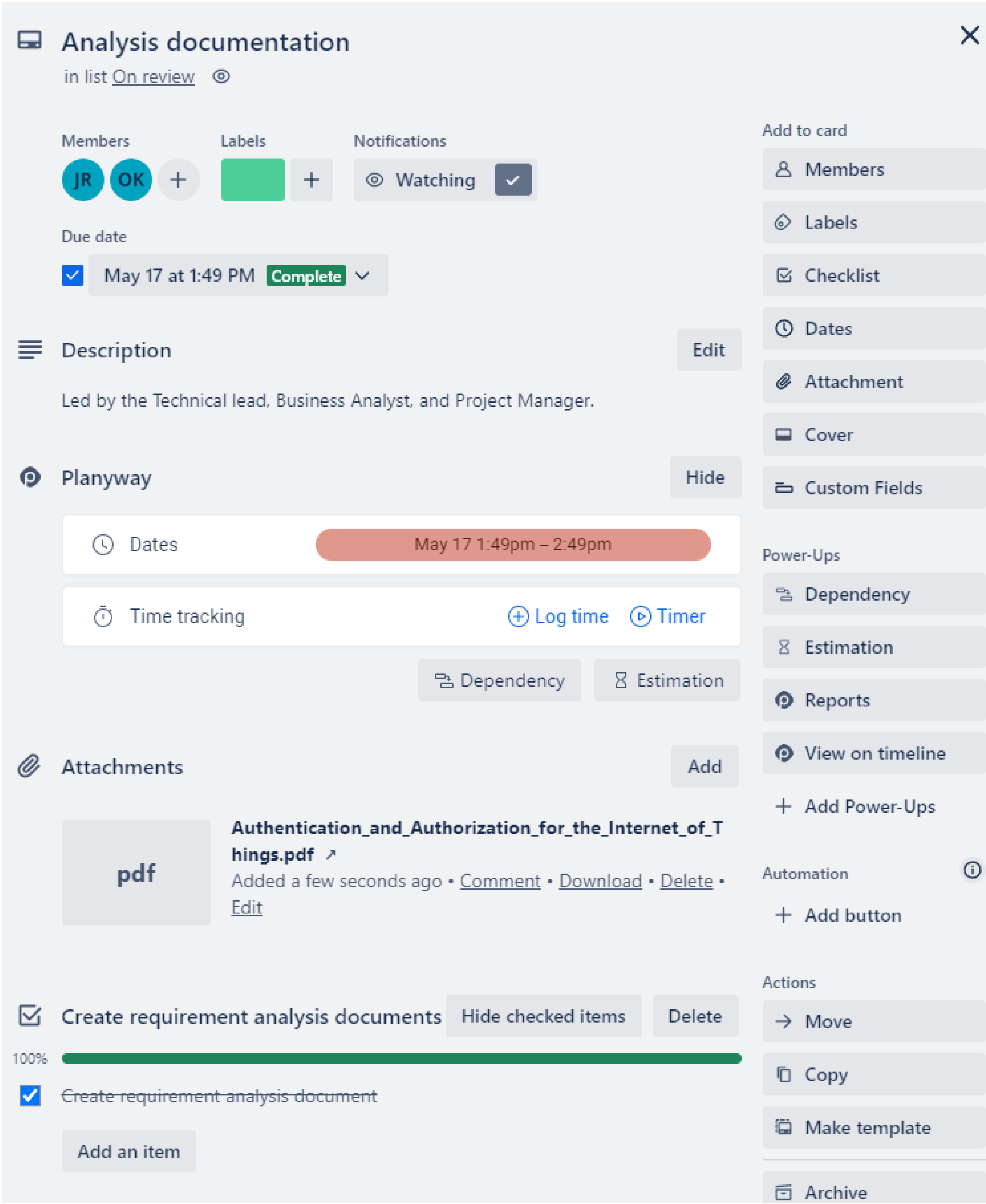

Fig 2: Trello Features for Each Task

Board Structure:

Our Trello board for Cake Labs is structured with four lists; To Do, In Progress, Under Review and Done. Each list signifies a phase of the project cycle ensuring visibility into the progress of tasks.

1. **To Do:**

This list comprises tasks that have not yet commenced. Examples include:

- Code Completion and Review: Handled by the technical lead and developers to ensure the final code meets all functional requirements.
- Security Audit: Conducted by the security analyst to confirm adherence to security standards.
- Testing: Performed by the QA engineer to verify system performance under loads.
- Deployment of Production: Managed by the technical lead and development team post-testing.
- Documentation: Handled by the business analyst and technical lead
- Stakeholder Briefing: Led by the project manager, technical lead, and business analyst.
- Project Completion Report: Compiled by the project manager.

2. **In Process:**

This list contains tasks in progress such as:

- API Proxy Configuration: Handled by the development team.
- Authentication Mechanism Setup: Implemented by the development team.
- API Version Management: Managed by the development team for API version control.
- Payload Manipulation Implementation: Executed by the development team to handle data formatting.
- Load Balancer Configuration: Configured by the technical lead for traffic management.

3. **On Review:**

Tasks in this list are under review to ensure they meet the required standards:

- Analysis Documentation: Created by the technical lead, business analyst, and project manager.
- Architecture Designing: Developed by the technical lead and business analyst.
- Setup Development Environment: Organized by the technical lead and development team.

4. **Completed:**

This section includes tasks that have already been completed:

- Project Planning: Conducted by the project manager.
- Kickoff Meeting: Led by the project manager.
- Define Project Scope: Outlined by both the project manager and business analyst.
- Define Goals and Objectives: Established by the project manager.
- Technical Analysis: Performed by the technical lead and developers.
- Functional Analysis: Conducted by the business analyst.

Roles Assigned and Rationale:

Roles are assigned based on each team member's expertise to ensure efficiency and capitalize on strengths. For instance:

- Technical Lead: Manages tasks such as code review and load balancer configuration.
- Security Analyst: Conducts security audits, for compliance assurance.
- QA Engineer: Carries out testing to validate system performance.
- Business Analyst: Oversees documentation. Functional analysis purposes.
- Project Manager: Oversees the planning briefs stakeholders. Ensures the project is completed.
- Development Team: Configures APIs, establishes authentication mechanisms, and implements payload manipulation.

Task Distribution:

Team members are assigned tasks to encourage smooth workflow and collaboration. Each task is delegated based on skills and responsibilities to ensure timely completion. For instance, technical tasks are handled by the lead and developers while business-related tasks are overseen by the project manager and business analyst.

Additional Features:

The Trello board includes several features to boost functionality and productivity:

- Labels: Tasks can be categorized according to priority and status.
- Checklists: Tasks can be broken down into steps, for progress tracking.
- Due Dates: Ensure that tasks are finished on time.
- PowerUps: Integrations like calendar view and automation enhance visibility and streamline processes.
- Comments and Attachments: Facilitate communication and document sharing within each task card.