

|  |
| --- |
| Solution Architecture Document  Global Identification Integration: Product and Location Management System (PLMS). |
| X-Customer-PLMS |

| REVISION HISTORY | | | | | |
| --- | --- | --- | --- | --- | --- |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1 | Initial Document | Enrique Zetina | 15-04-2024 |  | 30-04-2024 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

| Related Artifacts | |
| --- | --- |
| Ref. | Name |
|  |  |
|  |  |

Contents

[Global Identification Integration: Product and Location Management System 1](#_Toc165285639)

[1 Executive Summary 4](#_Toc165285640)

[2 Solution Approach 5](#_Toc165285641)

[2.1 Solution overview and how key goals are achieved 5](#_Toc165285642)

[2.1.1 Key Goals Achievement 5](#_Toc165285643)

[2.1.2 High-Level Technical Architecture 5](#_Toc165285644)

[2.2 Architecture components 5](#_Toc165285645)

[2.3 Integration with external systems 6](#_Toc165285646)

[2.4 List of platforms, TECHNOLOGIES, and tools 7](#_Toc165285647)

[2.5 Deployment approach 7](#_Toc165285648)

[2.6 Dependencies, Risks and Assumptions 8](#_Toc165285649)

[2.6.1 Dependencies 8](#_Toc165285650)

[2.6.2 Risks 8](#_Toc165285651)

[2.6.3 Assumptions 8](#_Toc165285652)

[3 Delivery Approach 9](#_Toc165285653)

[3.1 Used methodology 9](#_Toc165285654)

[3.2 Estimations 9](#_Toc165285655)

[3.3 Timeline, team structure 10](#_Toc165285656)

[3.3.1 Timeline 10](#_Toc165285657)

[3.3.2 Team Structure 10](#_Toc165285658)

[3.4 Pricing including the total cost 11](#_Toc165285659)

# Executive Summary

EPAM's proposed offering for X-Customer is a robust, scalable, secure, and user-friendly web-based application that meets and exceeds the extensive list of requirements as set out in the project scope. Understanding the complexity of X-Customer's current business environment and the ultimate project goals, we are confident that our solution will effectively service the current state while also accommodating room for future growth.

The solution will accommodate the management of Global Identification Numbers (GINs) for Products, Location Numbers (LNs) for Locations, and streamline the access of data associated with these entities while strictly validating each action against X-Customer standards and business rules. We have designed a comprehensive, modular approach structured to include integrated layers for managing GINs, LNs, and Data Access modules in a cohesive manner.

The proposed solution, while being highly functional and efficient, also places notable emphasis on intuitive user interface design and experience, making it straightforward for users of varying technical proficiency to navigate and operate. The primary objective is to create a tool that not only offers powerful features but is also easy to use.

Our project approach is methodical and strategic, divided into distinct stages of Requirements and Design, Development and Testing, Implementation, and Post-production Support and Transition. This life-cycle approach allows for systematic execution, continuous control, and timely communication with stakeholders, ensuring alignment with X-Customer's strategic objectives throughout the project duration.

Our solution is designed to be future-ready, prepared to accommodate potential expansions such as add-on modules for managing and sharing data beyond GIN or LN. It will also demonstrate a nimble ability to adapt to increasing data-sharing privileges to the field level.

We hold the competence to deliver this project on budget and within the stipulated timeframe leveraging our tested and proven project management methodologies. We understand X-Customer's need for fiscal propriety and the solution has been designed to respect budgetary constraints while still delivering superior functionality and performance.

In summary, this architecture document proposes a sophisticated, well-designed solution that comprehensively addresses X-Customer’s needs while creating an effective, user-friendly system for managing GIN and LN. It embodies a future-focused approach that anticipates and is ready to adapt to growth, thus aligning with X-Customer's strategic goals and providing significant operational value.

# Solution Approach

## Solution overview and how key goals are achieved

The proposed solution is a scalable, secure, and robust web-based application leveraging cutting-edge technologies and industry best practices. It's designed to manage Global Identification Numbers (GINs) for Products and Location Numbers (LNs) for Locations, together with streamlining related data access. It complies with X-Customer standards and business rules and uniquely integrates key functions to meet the business's evolving dynamics.

### Key Goals Achievement

The solution is fundamentally built around X-Customer’s key goals, which include robust functionality, future scalability, intuitive user interface, and comprehensive data access.

* **Robust functionality:** The application incorporates the functionality needed for role-based access control system, a workflow engine, database record locking, among many others. It ensures absolute data integrity and security across the process.
* **Future scalability:** The architecture of the application is developed with expandability in mind. It is flexible to add new modularity, thus catering to the future growth of the organization aligning with its dynamic needs.
* **Intuitive User Interface:** Ease of usage is paramount in our design strategy. We leverage advanced UI/UX principles to create a user-friendly interface with intuitive navigation and real-time data display, meeting the requirement of real-time responses.
* **Comprehensive data access:** The proposed solution is engineered with tools to import and export data records in diverse formats, reinforced by a strong validation process for data consistency. Multiple channels notifications have also been embedded for superior user engagement.

### High-Level Technical Architecture

The architecture of the solution is based on a three-tier architecture model:

* **Presentation Layer:** This is the user interface module that will provide interactive access to the application.
* **Business Logic Layer:** This encompasses all the application's business logic, including the role-based access system, workflow engine, record locking, notifications, etc.
* **Data Access Layer:** This consists of the data access functionalities, advanced search functionality, and the database itself.

The architecture promotes modularity, maintainability, and scalability, tailored to handle high workloads and extensive data. The service-oriented approach ensures easy integration with other systems and facilitates a high degree of reusability.

By combining our technological skill set with a deep understanding of X-Customer's business needs, our solution approach delivers a powerful, scalable, and user-friendly application that not only meets the current requirements but also provides the capacity to meet future challenges. We are committed to delivering a solution that caters to both functional and non-functional requirements while ensuring the highest level of data integrity and security.

## INFRASTRUCTURE / DEPLOYMENT VIEW Integration with external systems

The proposed solution is built to interoperate seamlessly with X-Customer's existing software ecosystem. Here's an outlook on how it integrates with external systems:

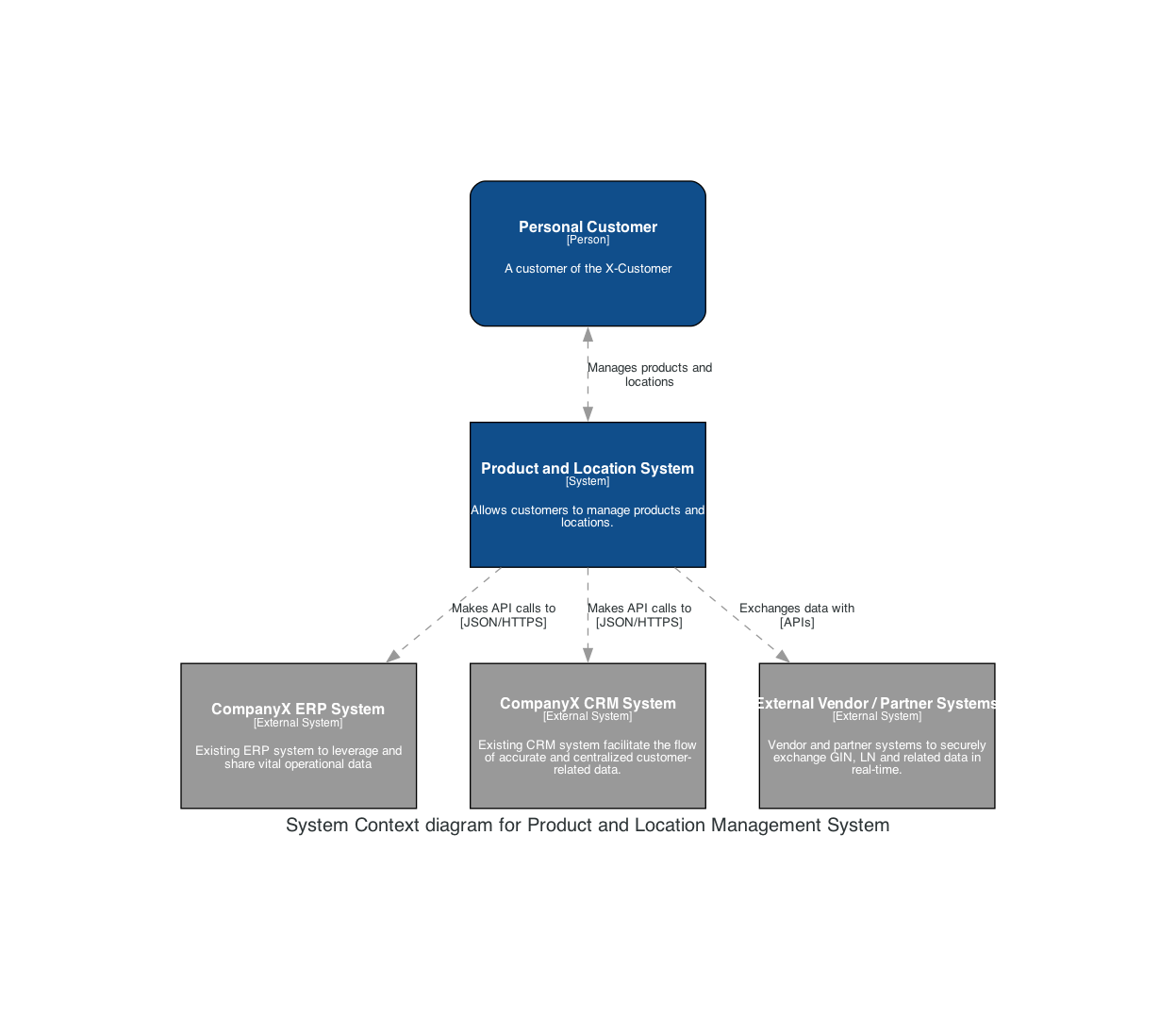
* **Enterprise Resource Planning (ERP) Integration:** Our solution will be integrated with the existing ERP system to leverage and share vital operational data, facilitating efficient business management.
* **Customer Relationship Management (CRM) Integration:** By integrating our solution with the CRM platform, we will facilitate the flow of accurate and centralized customer-related data, promoting customer satisfaction and revenue enhancements.
* **Data Warehouse Integration:** Through this integration, we intend to support advanced analytics and reporting by efficiently feeding data into the Data Warehouse system.
* **External Vendor and Partner Systems:** Catering to the growing inter-dependency needs, the solution will offer APIs to securely exchange GIN, LN and related data with external vendor and partner systems in real-time.
* **Data Migration Tools:** We'll ensure that the solution integrates with your data migration tools, easing the transition from older systems.
* **Security Systems:** The web-based solution will be fully compatible with X-Customer's existing security infrastructure including firewalls, intrusion detection/prevention systems, encryption and key management systems, and antivirus solutions.

The precise nature and extent of these integrations will depend on the specific software used by X-Customer and functional requirements. Our technical team will work closely with the client’s IT team to ensure seamless and secure integration.

## List of platforms, TECHNOLOGIES, and tools

* **Platform**: Google Cloud Platform (GCP) - Leveraging its scalability, reliability, and security.
* **Programming Languages:** 
  + Python - For server-side development due to its flexibility, compatibility with GCP.
  + Java - For microservices development and complex server-side operations, again compatible with GCP.
* **Front-End Technologies:** 
  + HTML/CSS - foundational languages for building web pages.
  + JavaScript (React.JS or Angular.JS) - These powerful libraries/frameworks will ensure the development of a responsive, interactive, and intuitive user interface.
* **Database:** Google Cloud SQL – Managed relational database service for smooth data handling.
* **Data Warehouse:** Google BigQuery - Handles real-time analytics and reporting on large datasets.
* **Authentication & Authorization:** Google Identity Platform – For managing secure, simple sign-up and sign-in.
* **Storage:** Google Cloud Storage - For storing files and other essential data in various formats.
* **Version Control:** Git/GitHub - To manage and control versions and source code.
* **CI/CD:** Google Cloud Build – Enables automated build tests and deployments.
* **Containerization:** Docker - for creating lightweight, stand-alone, and executable software packages that include everything needed to run a piece of software.
* **Container Orchestration:** Google Kubernetes Engine (GKE) - For managing containerized applications.
* **API Gateway:** Google Cloud Endpoints - To manage, secure, and monitor APIs.
* **Event/Messaging System:** Google Pub/Sub - Implementing an event-driven architecture and setup communication between services.
* **Infrastructure Monitoring:** Google Cloud Monitoring - To monitor the performance of applications and maintain logs.
* **Error Reporting:** Google Cloud Error Reporting - Keeps track of runtime errors and exceptions.
* **Load Balancing:** Google Cloud Load Balancing - To distribute network or application traffic across many resources.
* **Logging:** Google Cloud Logging - To store, search, analyze and alert on log data and events.
* **Data Integration:** Google Cloud Data Fusion - To integrate, clean, and transform raw data into useful insights.
* **API Management:** Google Cloud's Apigee - To design, secure, and scale API ecosystems.
* **Planning & Collaboration:** JIRA & Confluence - For project management and collaboration.

### System Context diagram

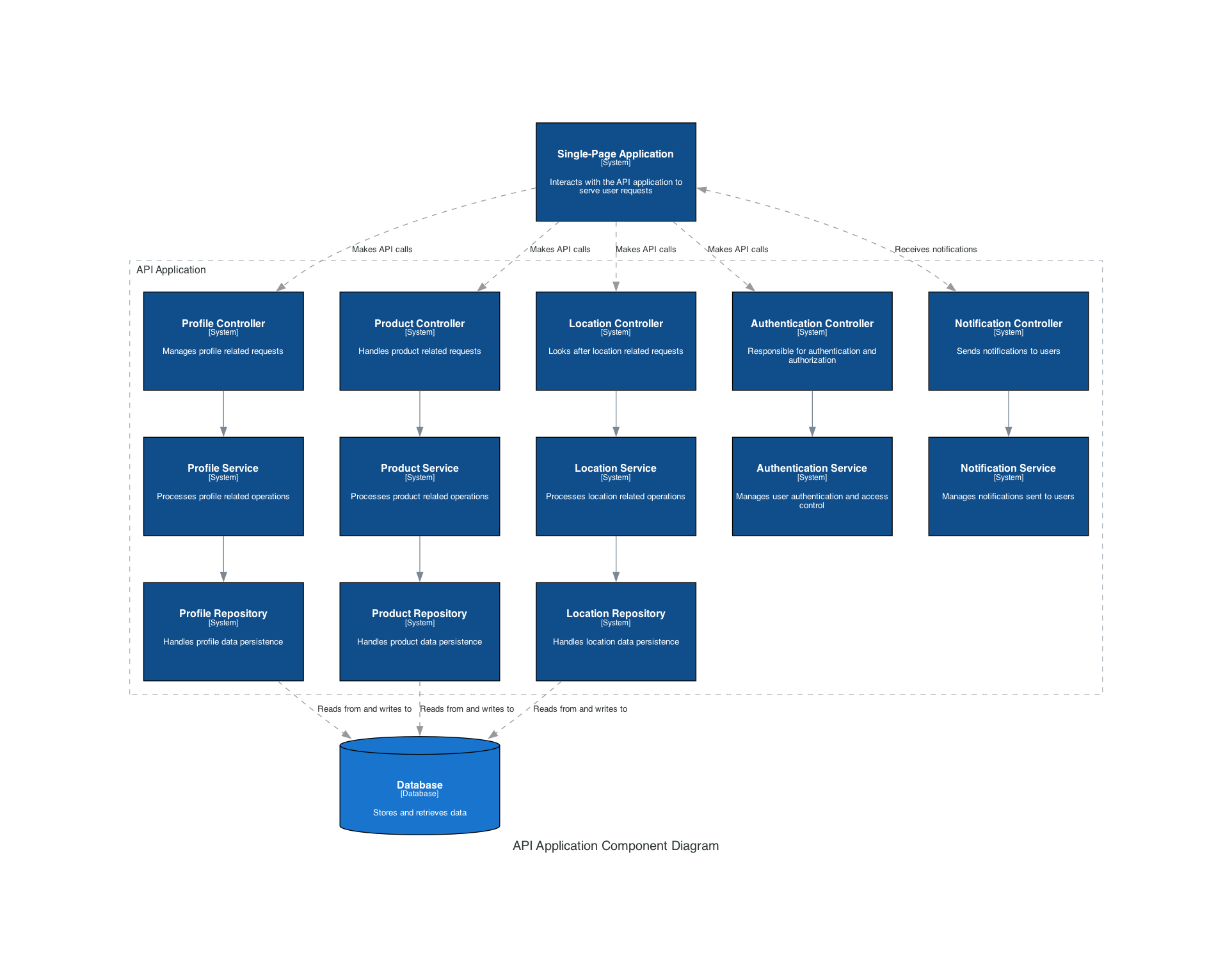


### Container diagram

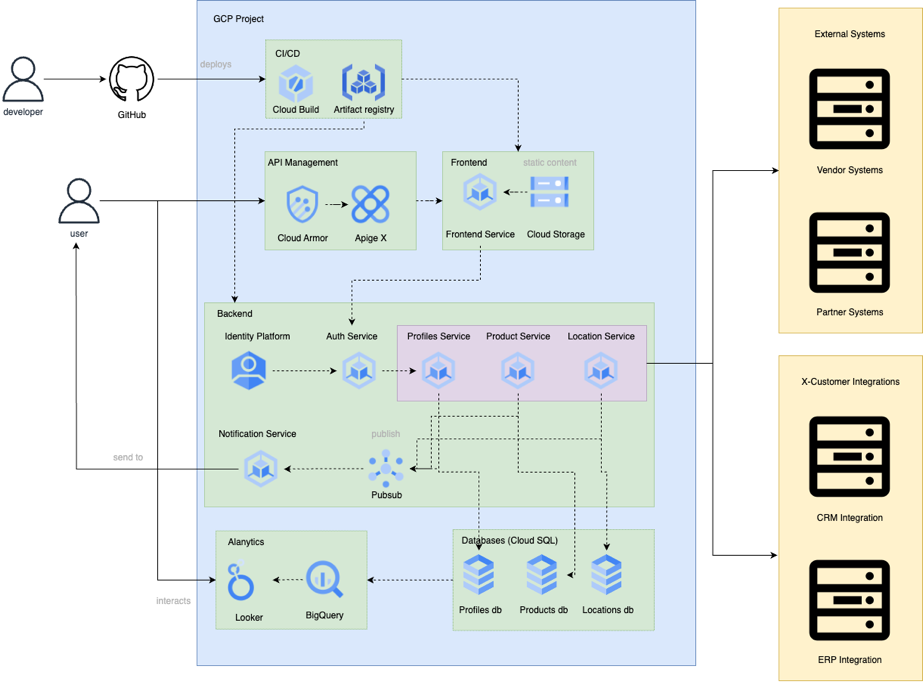
A diagram of a company's flowchart

Description automatically generated

### Component diagram



### Deployment diagram



## Deployment approach

The deployment approach leverages the principles of Continuous Integration and Continuous Deployment (CI/CD) to ensure smooth transitions from development to production. The approach guarantees quick, reliable, and automated releases with minimal human intervention.

* **Development and Testing**: The developers initially work in a local environment where they utilize Google Cloud SDK. Code, once committed, triggers the CI/CD pipeline using Google Cloud Build or a similar tool, which ensures that all tests pass and code quality standards are met.
* **Staging**: Upon successful trial, the code is then deployed to a staging environment that closely mirrors the production environment. This stage allows for intensive testing without affecting the live system. Performance, load, and security tests are directed here to ensure the upcoming version's stability and readiness.
* **Blue/Green Deployment**: In the production environment, we aim to adopt a Blue/Green deployment strategy. In this scenario, we have two production environments, Blue (the live environment) and Green (an idle replica). Once we're ready to release a new version, we deploy it on the green environment. After thorough validation, we switch the traffic to the green environment, making it the new live or blue environment. The advantage here is near-zero downtime and an instantaneous rollback if something goes wrong.
* **Rollout**: In some cases where changes are significant and may impact users, we would opt for a phased rollout, progressively increasing the percentage of users handling the new version.
* **Monitoring and Logging**: Post-deployment, monitoring, and logging, using Google Cloud Monitoring and Logging, are vital to ensure system health and proactively detect any anomalies.
* **Backups and Recovery**: To safeguard against unexpected failures or data loss, we schedule automatic backups using Google Cloud storage solutions. We also plan to test our backup and recovery procedures regularly to ensure data safety.
* **Documentation**: Post-deployment, we will provide detailed deployment and system documents for the IT team. The documentation will contain specifics about the deployment process, application performance tuning, and other operational tasks.

This well-structured, automated deployment approach will ensure we release the most stable iterations of the application with the least disruption to your operations.

## Dependencies, Risks and Assumptions

### Dependencies

* **Access to Existing Systems:** Successful integration of the proposed solution with current systems depends on necessary access rights and complete information about the existing systems.
* **Stakeholder Availability**: The availability of key stakeholders for timely input, decisions, and validation is vital for the project's success.
* **Infrastructure**: The assumption that the existing hardware and network infrastructure are adequate to support the new web application's requirements.

### Risks

* **Change in Requirements**: Unanticipated changes to the project requirements may lead to delays, increased costs, or possibly rework.
* **Data Sensitivity**: Risk of data breaches and leaks due to the sensitivity of the data involved. We will mitigate this through stringent security measures.
* **Technical Constraints**: Occurrence of unforeseen technical or integration issues might delay the planned timelines.
* **Resource Availability**: Project success can be jeopardized if crucial team members become unavailable during critical project phases.

### Assumptions

* **Project Responsibilities:** It is assumed that roles and responsibilities will be clearly defined and followed by the project team members.
* **Technical Support:** X-Customer IT team will provide necessary support as required, especially while integrating with existing systems.
* **Training:** X-Customer will provide the required training for end-users and properly communicate any system changes.
* **Readiness to Change:** The organization, its processes, and its people are ready for the changes the new system will introduce. Resistance to implementation from staff might pose a significant risk to the project.

The above-defined dependencies, risks, and assumptions outline the potential constraints that may affect the project timeline or outcome and helps in outlining a viable mitigation plan. We intend to closely monitor all project-related aspects, perform regular risk assessment, and take proactive measures to counteract potential adverse impacts.

# Delivery Approach

## Used methodology

The delivery approach leverages Agile Scrum methodology considering its effectiveness in managing complex projects, encouraging collaboration, and allowing for rapid adjustments.

* **Sprint Planning:** We organize the project into smaller, manageable units called 'sprints', usually spanning two weeks. Each sprint begins with a planning meeting where the team commits to complete a set of user stories from the product backlog.
* **Daily Stand-ups:** Short daily meetings are held to update the team on each member's accomplishments, future tasks, and to identify any roadblocks or risks. This keeps the team aligned and proactive in problem resolution.
* **Development:** Developers begin working on the user stories. The focus is on producing working software at the end of each sprint, fostering a constant pace of delivery.
* **Testing:** Continuous testing is performed alongside development to ensure early identification and rectification of defects.
* **Sprint Review and Retrospective:** At the end of each sprint, a meeting is held to evaluate the work done conducted during the sprint and to discuss any potential improvements for the next sprints.
* **Product Iteration:** The product is improved incrementally sprint by sprint, integrating feedback after each delivery. This allows for frequent validation with stakeholders, which ensures alignment with expectations.
* **Repeat:** This cycle repeats until all the requirements are met.

The Agile Scrum approach is flexible and easily adapts to changes. It supports close collaboration between the development team and the stakeholders, ensuring that the product increment matches the customer's requirements and expectations. It also enables constant inspecting and adapting, helping improve both the development process and the quality of the product.

## Estimations

* **Project Initiation and Planning:** Approximately 2 weeks will be spent on project initiation activities, which includes finalizing the project plan, setting up communication and reporting protocols, and formalizing the governance structure.
* **Requirements Gathering and Analysis:** This phase involves detailed interviews with stakeholders and potentially end-users. We anticipate this to take about 3 weeks to a month.
* **Solution Design:** The creation of detailed design documents, including system architecture, database design, and user interface design, may take approximately 1-2 months.
* **Solution Development:** Given the complexity of the application and the need to ensure high quality, we anticipate the development phase to last around 3-4 months.
* **Integration & System Testing:** Integrating the components and performing end-to-end system testing would take around 1-1.5 months.
* **User Acceptance Testing (UAT):** Once the system has passed internal testing, it will be tested by actual users. This phase could take about 2 weeks.
* **Deployment and Stabilization:** Actual deployment to the production environment and initial stabilization would take around 2 weeks.
* **Post-Production Support and Transition:** We plan to provide post-production support for a period of 2 months.

Given these estimates, we foresee the complete project life cycle from initiation to post-production support to last approximately **9-12 months**. These timelines are provided as an estimate and could change based on several factors including change in requirements, productivity in development, delays in feedback, among others. We will continuously monitor the project to identify any risk to these timelines at an early stage and take appropriate mitigation measures.

## Timeline, team structure

### Timeline

A graph with different colored lines

Description automatically generated

Given the high-level estimates described earlier, the following tentative timeline is proposed:

* **Project Initiation and Planning:** Weeks 1-2
* **Requirements Gathering and Analysis:** Weeks 3-6
* **Solution Design**: Weeks 7-14
* **Solution Development:** Weeks 15-30
* **Integration & System Testing:** Weeks 31-38
* **User Acceptance Testing (UAT):** Weeks 39-40
* **Deployment and Stabilization:** Weeks 41-42
* **Post-Production Support and Transition:** Weeks 43-50

### Team Structure

* **Project Manager (PM):** 1 - The PM will co-ordinate all aspects of the project and will be your main point of contact.
* **Business Analysts (BAs):** 2 - The BAs will work together closely with your team to ensure that all business requirements are captured and translated into effective functional specifications.
* **Solution Architect (SA):** 1 - Our SA will be responsible for creating the optimal system design that meets all your business needs, while ensuring the system remains robust and scalable.
* **UI/UX Designers:** 2 - Our designers will work closely with the BAs to create an intuitive, user-friendly interface for the application.
* **Developers:** Depending on the specific requirements and the programming languages involved, we propose a team of 6 developers. This could include 2 each of Python, Java and Node.js developers.
* **Quality Assurance (QA) Specialists:** 3 - QA team will be tasked with conducting extensive testing on the developed application to ensure it meets all functional requirements and adheres to quality standards.
* **Infrastructure Specialist:** 1 - This expert will manage the deployment and operational environment of the solution.

This brings the total to a team of 16 members. This is a well-resourced, cross-functional team that should enable us to maintain high throughput, meet the proposed timeline, and deliver a high-quality product. Please note that this is based on our current understanding and the final team numbers might need adjustments based on the actual project progress and uncovered complexities.

### Pricing including the total cost.

For this estimate, we'll consider a simplified scenario with the costs for each environment as following based on their size and uptime:

* **Development (Dev) Environment**: 50% of Production resources, 60% uptime
* **Staging (Stg) Environment:** 75% of Production resources, 60% uptime
* **User Acceptance Testing (UAT) Environment**: 100% of Production resources, 60% uptime
* **Production (Prod) Environment**: 100% of resources, 100% uptime





The above estimates are indicative and represent the cost of the various services that the application will use over time, from the current year to the projected increase by 2024.