

From Workflow Mapping to Autonomous Selling: Building a Process-Centric Reseller Sales & Management Application

Rodrigo Castro Lima

Master's Thesis

Supervisor at FEUP: Prof. Nuno Soares



Master in Industrial Engineering and Management

2025-06-26

Abstract

This dissertation addresses the operational inefficiencies and coordination challenges faced by *Company X*, a rapidly scaling SME operating in Portugal’s energy services sector. Although part of a broader corporate group using Salesforce CRM, *Company X* historically relied on fragmented and informal workflows involving email, WhatsApp, spreadsheets and manual document handling across its reseller network. These practices led to duplicated efforts, inconsistent data quality, lack of visibility and customer dissatisfaction, particularly during sales and onboarding processes. To address this, the research project pursued a full-cycle digital transformation, centered on business process modeling and the development of a tailored mobile-first platform, *X-App*.

The methodology combined Business Process Management (BPM) principles with modern software engineering practices. BPMN 2.0 was used to formally model the existing process landscape, surface coordination breakdowns and categorize pain points across four operational stages. Drawing on recent literature, the platform architecture adopted a modular and serverless design, integrating Firebase for data and identity services and Salesforce for centralized CRM operations. The system supported distinct user roles and enabled real-time data synchronization and workflow automation through custom backend logic.

The resulting system significantly improved process transparency, reduced redundant data entry and streamlined document handling. By digitizing key reseller interactions — including invoice submission, proposal status tracking and document validation — the platform reduced manual backoffice workload and elevated service consistency. While this dissertation focuses on current implementation outcomes, it also explores a forward-looking vision in which autonomous AI agents could replicate reseller workflows and further automate sales operations.

Resumo

Esta dissertação aborda as ineficiências operacionais e os desafios de coordenação enfrentados pela *Empresa X*, uma PME em rápida expansão que atua no setor energético em Portugal. Apesar de integrar um grupo empresarial mais amplo que utiliza *Salesforce* como sistema de CRM, a Empresa X desenvolveu práticas de trabalho fragmentadas e informais, baseadas em email, *WhatsApp*, folhas de cálculo e gestão manual de documentos na sua rede de revendedores. Estes métodos resultaram em retrabalho, inconsistência na qualidade dos dados, falta de visibilidade e insatisfação dos clientes, sobretudo durante os processos de venda e *onboarding*. Para enfrentar estes problemas, o projeto de investigação implementou uma transformação digital completa, centrada na modelação de processos de negócio e no desenvolvimento de uma plataforma móvel personalizada, a *X-App*.

A metodologia combinou princípios de *Business Process Management* (BPM) com práticas modernas de engenharia de software. Recorreu-se à notação BPMN 2.0 para modelar formalmente os processos existentes, identificar falhas de coordenação e categorizar os principais pontos críticos ao longo de quatro fases operacionais. Com base na literatura especializada, foi concebida uma arquitetura modular e *serverless*, integrando *Firebase* para serviços de dados e autenticação e *Salesforce* como sistema central de CRM. A plataforma foi desenhada para suportar diferentes perfis de utilizador e permitiu a sincronização de dados em tempo real, bem como a automatização de tarefas através de lógica de *backend* personalizada.

O sistema resultante melhorou significativamente a transparência dos processos, reduziu a duplicação de dados e agilizou a gestão documental. Ao digitalizar interações críticas com os revendedores — nomeadamente a submissão de faturas, o acompanhamento de propostas e a validação de documentos — a plataforma diminuiu a carga de trabalho administrativa e aumentou a consistência do serviço prestado. Embora esta dissertação se concentre nos resultados da implementação atual, apresenta também uma visão prospetiva, na qual agentes autónomos baseados em IA poderão replicar as funções dos revendedores e automatizar ainda mais os processos comerciais.

Acknowledgments

I would like to begin by expressing my sincere gratitude to my supervisor, Professor Nuno Soares, for his guidance and continued support throughout the development of this work. His insight was instrumental in shaping both the structure and depth of this project.

I also extend my thanks to the Upgraide team for their collaboration and technical contributions during the project's development. I'm especially grateful to João, Francisco, and Salvador for giving me the opportunity to work on such a challenging and exciting project. To my colleagues Bernardo, Gonçalo, Miguel, and Francisco — thank you for the companionship and the all-around great vibes. Special thanks to Afonso and Bruno, my office space partners, for putting up with my music and helping me through various challenges along the way.

I appreciate Company X for the opportunity and support in developing this project within their operational context.

To my family — especially my mother Dafne, father Rui, and brother Lourenço — thank you for your unwavering support, encouragement and patience throughout this academic journey. Your belief in me has been a constant source of strength and motivation to always aim higher.

I also wish to thank all my university professors and classmates for accompanying me on this journey. It has been a deeply rewarding learning and personal growth experience.

Looking back on these five years, I genuinely feel proud of the path I've taken. There were plenty of highs and lows, but I wouldn't change a thing. Moving on isn't easy — and truthfully, closing this chapter hits harder than I expected. Still, whenever I feel that weight, I try to hold onto a quote that's always stuck with me: "Don't cry because it's over, smile because it happened."

Rodrigo Castro Lima

"You only ever grow as a human being if you're outside your comfort zone."

Jonny Wilkinson

Contents

1	Introduction	1
1.1	Project Context and Motivation	1
1.2	The Company X Project in Upgrade AI	2
1.3	Project Objectives	2
1.4	Project Methodology	3
1.5	Structure of the Dissertation	4
2	Literature Review	7
2.1	Business Process Modeling and Workflow Improvement	7
2.2	CRM Integration and Customization	10
2.3	Leveraging Modern Technology Stacks for Operational Efficiency	12
2.4	AI-Driven Automation and Intelligent Systems	14
3	Organizational Context and Problem Framing	17
3.1	Organizational Context	17
3.1.1	Market Position and Commercial Strategy	17
3.1.2	Organizational Structure	18
3.2	Description of the Existing Workflow	19
3.3	Problem Identification and Process Gaps	23
3.4	Strategic Targeting of Process Redesign	27
3.5	Rationale for Platform Development	31
4	Solution Architecture and Implementation	35
4.1	Overview and Design Principles	35
4.2	Tech Stack Summary	38
4.3	Feature Catalogue	40
4.3.1	Authentication and User Management	40
4.3.2	Service Submissions	42
4.3.3	Opportunity and Proposal Management	44
4.3.4	Message and Notifications	46
4.3.5	Client Management	47
4.4	Integration Architecture	48
4.4.1	Salesforce CRM Synchronization	49
4.4.2	Firebase Service Integration	51
4.4.3	Cross-Service Communication	53
4.5	Security Architecture	54
4.6	Deployment and Operational Practices	55

5	Conclusion	57
5.1	Summary of Work Conducted	57
5.2	Contributions and Impact	58
5.3	Limitations	59
5.4	Future Work	59
5.5	Final Remarks	61
	References	63

Acronyms and Symbols

AI	Artificial Intelligence
API	Application Programming Interface
BaaS	Backend-as-a-Service
BPM	Business Process Modeling
BPMN	Business Process Model and Notation
BPR	Business Process Reengineering
CDN	Content Delivery Network
CI/CD	Continuous Integration / Continuous Deployment
CLI	Command Line Interface
CRM	Customer Relationship Management
ERP	Enterprise Resource Planning
FaaS	Function-as-a-Service
FCM	Firebase Cloud Messaging
FEUP	Faculdade de Engenharia da Universidade do Porto
ID	Identifier
ISO	International Organization for Standardization
IT	Information Technology
JWT	JSON Web Token
LLM	Large Language Model
ML	Machine Learning
NLP	Natural Language Processing
OCR	Optical Character Recognition
OAuth	Open Authorization
PEM	Privacy-Enhanced Mail
PKCE	Proof Key for Code Exchange
REST	Representational State Transfer
SDK	Software Development Kit
SLA	Service-Level Agreement
SME	Small and Medium-sized Enterprise
UI	User Interface
UID	User Identifier
VM	Virtual Machine

List of Figures

1.1	Gantt chart outlining project phases and timeline	4
3.1	Complete BPMN diagram of the reseller workflow	20
3.2	Workflow diagram for Phase 1	21
3.3	Workflow diagram for Phase 3	22
3.4	Impact–Effort Matrix used to prioritize intervention areas based on diagnostic findings	28
4.1	Architecture of the X-App system	36
4.2	Reseller home page interface	41
4.3	Admin home page interface	41
4.4	Multi-step service submission form used by resellers to initiate structured onboarding requests within the <i>X-App</i>	43
4.5	Administrative review interface where submissions are analyzed and approved or rejected	43
4.6	Relationship between Salesforce objects	45
4.7	Administrative messaging panel displaying real-time communication with resellers, including message history, file attachments and live updates	46
4.8	Reseller dashboard displaying assigned opportunities and their statuses	47
4.9	Proposal details and associated document submission interface	48
4.10	Comparison of Salesforce authentication flows for admins and resellers	49

List of Tables

2.1	Essential BPMN symbols and their functions	8
2.2	Cost comparison between VM and serverless architectures at different scales (Eivy, 2017)	12
2.3	The Twelve-Factor App principles	14
3.1	Synthesis of core operational problems	26
3.2	Mapping of addressed problems to platform solution mechanisms	32
4.1	Core components of the X-App technology stack	39
4.2	Firestore collections and access scope	52

Chapter 1

Introduction

This introductory chapter provides the foundational context for the project. It begins by outlining the operational challenges faced by Company X and the motivations behind the intervention. It then situates the project within the broader scope of Upgrade’s applied AI practice, introduces the project’s objectives and methodology and concludes with an overview of the dissertation’s structure. Together, these elements set the stage for the technical and organizational analysis that follows.

1.1 Project Context and Motivation

In the context of growing digital maturity within sales-driven organizations, the effective coordination of decentralized workflows has emerged as a key operational challenge. This is especially true in environments where commercial activity is conducted through external partners or independent agents, such as resellers. While this model offers scalability and market reach, it often leads to fragmented communication, inconsistent data handling and limited process traceability, particularly when operational flows rely on informal tools.

The company studied in this project, hereafter referred to as **Company X**¹, operates under such a model. As a commercial division of the XYZ Group, it distributes energy services through a network of independent resellers who work on a commission-based structure. Although the broader group uses Salesforce as its Customer Relationship Management (CRM) system, Company X developed its own informal workflows outside this structure, relying heavily on informal coordination between field agents and back-office teams. As the reseller network scaled, these informal processes became increasingly unmanageable, creating delays, rework and operational risk.

This project was conceived in response to that reality. Its core motivation was to redesign Company X’s commercial workflow into a structured, traceable and digitalized process that could

¹A pseudonym is used throughout this dissertation to protect the confidentiality of the client organization and its parent group, here referred to as “XYZ Group”.

support business growth without incurring proportional administrative burden. Beyond digitalization, the goal was also to build a technological foundation capable of integrating AI-driven features, thus paving the way for intelligent process assistance and future automation.

1.2 The Company X Project in Upgraide AI

This project was developed at *Upgraide*, a Portuguese artificial intelligence consulting company founded by two engineering students from FEUP. Upgraide specializes in building tailored AI-powered solutions that drive real-world impact, helping businesses become more competitive through intelligent automation and digital transformation. Rather than relying on generic software products, the company follows a bespoke, engineering-centric approach—designing and implementing AI systems that are deeply aligned with each client’s processes, objectives and operational constraints.

Upgraide’s work spans a variety of sectors, including energy, logistics and industrial services, with a strong emphasis on creating modular and scalable systems that integrate seamlessly into existing workflows. The company also invests in AI training programs for client teams, enabling organizations to build internal capabilities and sustain innovation independently. With its focus on measurable outcomes and agile deployment, Upgraide aims to place artificial intelligence at the core of business strategy and operations.

The client for this project was *Company X*, a Portuguese energy reseller operating under a decentralized sales model. Independent agents, referred to as resellers, are responsible for identifying potential clients, collecting utility documents and submitting them to an internal back-office team. This team then prepares offers using supplier platforms and manages the documentation cycle until the contract is activated by the provider.

Prior to the intervention, Company X’s entire process, from lead acquisition to contract activation, was managed through informal tools such as WhatsApp, Excel and local storage. Salesforce CRM was used exclusively for audit compliance, not for real-time operational control. This fragmented approach led to duplicated data entry, inconsistent documentation standards and a lack of visibility across the sales lifecycle.

The project documented in this dissertation aimed to address these structural inefficiencies. To that end, a new multi-platform digital system — *X-App* — was designed and developed to centralize reseller operations, integrate securely with Salesforce, and provide both field agents and back-office personnel with a unified workflow for managing commercial activities. The application not only improves operational control but also lays the foundation for intelligent process automation, enabling future extensions such as AI-driven agent systems.

1.3 Project Objectives

The project was conceived to address structural inefficiencies in Company X’s sales operations, which at the time relied heavily on informal tools and manual coordination. The absence of a cen-

tralized system led to duplicated effort, limited process visibility and inconsistent documentation quality. As the organization aimed to scale its reseller network, these constraints posed a serious barrier to sustainable growth and operational control.

To resolve these challenges, the project was guided by a set of strategic objectives, each targeting a critical area of intervention:

- **Digitize and structure reseller workflows :** Replace ad hoc communication and fragmented processes with a formalized digital workflow, improving coordination, reducing errors and enabling full lifecycle traceability across all sales stages.
- **Develop a unified platform accessible across devices :** Create a responsive, cross-platform system compatible with both mobile and desktop environments. This ensures resellers can operate effectively from the field while back-office teams manage validation and administration centrally.
- **Ensure integration with Salesforce CRM :** Connect the new application to XYZ Group's existing Salesforce system in a way that avoids redundant data entry while preserving data integrity and audit compliance. This objective required the use of robust API integrations and secure authentication flows.
- **Standardize documentation submission and validation :** Introduce structured data input and file submission flows to reduce the incidence of incomplete or incorrect documentation. This minimizes rework, streamlines internal validation and speeds up overall process cycles.
- **Lay the foundation for intelligent features :** Architect the system with modularity and extensibility in mind, enabling future additions such as AI-powered agents, automated document verification and engagement analytics without requiring fundamental reengineering.

These objectives were formulated through a process-centric diagnostic approach, supported by BPMN modeling and an Impact–Effort prioritization framework. They reflect both immediate operational needs and long-term innovation goals aligned with Company X's ambition to scale its decentralized sales model sustainably.

1.4 Project Methodology

The project followed an iterative, process-centric development methodology. It began with a comprehensive mapping of the existing workflow using BPMN standards, supported by informal interviews and document analysis. This was followed by a diagnostic phase where operational inefficiencies were identified and categorized.

Key problems, including redundant data entry, fragmented communication and inconsistent document validation, were then prioritized for intervention. The architectural design of the platform was developed in parallel with a modular implementation roadmap using modern cloud technologies.

The development of *X-App* was structured into eight overlapping phases over 4.5 months, as shown in Figure 1.1. The project began with Requirements Analysis and a Diagnostic Phase to map inefficiencies and understand operational needs.

Architecture Design and Prototyping followed in parallel. The design phase included literature review and the definition of core principles such as modularity, mobile-first access and system integration. A working prototype was built to validate the solution with the client before implementation began.

Core Implementation and Salesforce Integration ran through March and April. Internal Testing took place in May to stabilize features and resolve integration issues. In parallel, the Final Documentation phase focused on preparing the dissertation and consolidating technical deliverables. The phased overlap of design, implementation and validation ensured iterative progress and early detection of integration or workflow issues.

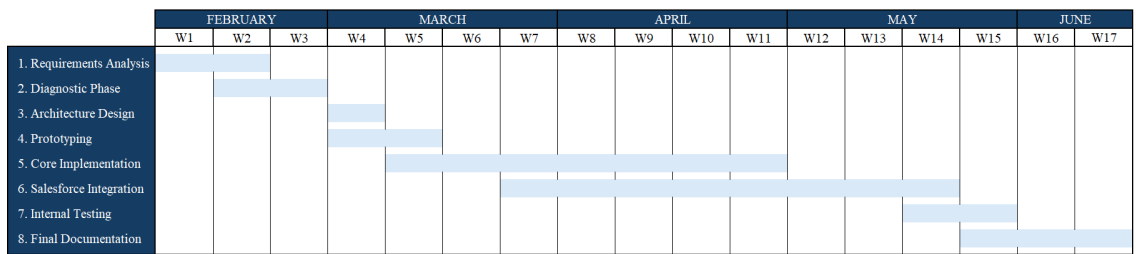


Figure 1.1: Gantt chart outlining project phases and timeline

1.5 Structure of the Dissertation

This dissertation is structured into five chapters, each dedicated to a specific dimension of the project and collectively forming a coherent narrative on the digital transformation of workflows within Company X.

Chapter 2 presents the literature review, which explores foundational concepts related to business process modeling, CRM integration strategies, modern cloud technology stacks and the evolving role of artificial intelligence in enterprise automation. This chapter establishes the theoretical and methodological underpinnings that informed the project’s architectural and strategic decisions.

Chapter 3 examines the organizational context in which Company X operates. It details the company’s existing workflow, modeled using BPMN standards, and identifies the key inefficiencies, coordination gaps and rework patterns that prompted the need for intervention. The chapter concludes with the rationale for developing a dedicated digital platform as a response to the operational constraints identified.

Chapter 4 is dedicated to the solution architecture and the implementation strategy of the proposed system. It presents the core design principles, details the selected technology stack and describes the key features of the *X-App*. The chapter also covers the security architecture, integration mechanisms with Salesforce and deployment practices that enabled the platform to function reliably in a decentralized sales environment.

Finally, Chapter 5 concludes the dissertation by synthesizing the outcomes of the project and discussing potential future developments. Particular emphasis is placed on the long-term vision of evolving the system toward an intelligent, autonomous agent capable of performing key reseller functions, thereby enabling deeper automation of commercial workflows.

Chapter 2

Literature Review

The following chapter reviews the academic and technical foundations that informed the design and implementation of the *X-App* platform. It synthesizes relevant literature across four key domains: business process modeling and workflow optimization, CRM integration and customization, modern cloud-native technology stacks and AI-driven automation. These sections establish the conceptual and methodological underpinnings that guided both the problem framing and the solution architecture developed in this project.

2.1 Business Process Modeling and Workflow Improvement

A business process is commonly defined as a structured and measurable set of activities designed to produce a specific output for a particular customer or market. These activities are often sequenced and cross-functional boundaries, involving multiple departments, roles or systems. Processes may be operational, managerial or strategic, but in all cases, they are the primary mechanisms through which organizations deliver value (Davenport, 1993).

Historically, organizations were structured around functional silos, emphasizing departmental goals and responsibilities. However, organizational thinking has increasingly shifted toward process-oriented approaches, emphasizing cross-functional workflows that directly impact customer outcomes. This shift was catalyzed by paradigms such as Business Process Reengineering (BPR), introduced by Hammer and Champy (1993), who emphasized the radical redesign of core processes to achieve dramatic improvements in performance. Today, process thinking underpins methodologies such as Lean, Six Sigma and continuous improvement frameworks, all of which rely on visibility and standardization of processes to drive operational excellence (George, 2002; Imai, 1986).

In this context, Business Process Modeling (BPM) emerges as a critical practice in Industrial Engineering and Management. It provides a structured approach to representing and optimizing workflows, enabling organizations to improve their operational efficiency, reduce variability and align activities with strategic goals (Dumas et al., 2018).

Business Process Model and Notation (BPMN) has emerged as the de facto standard for modeling organizational processes. Designed to be both technically precise and accessible to business stakeholders, BPMN provides a graphical language for representing workflows, decisions and actors in a unified manner (Dumas et al., 2018). Its wide adoption is attributed to its dual value: it facilitates communication between technical and non-technical participants while also enabling automation and simulation through standard-compliant process engines. BPMN is codified under the ISO/IEC 19510:2013 standard (International Organization for Standardization, 2013), further supporting its widespread adoption in enterprise environments requiring traceability and auditability (vom Brocke et al., 2016). The primary symbols used throughout this thesis to model the reseller workflow are summarized in Table 2.1, following BPMN 2.0 conventions.

Symbol	Name	Description
	Start Event	Marks the entry point of a process
	End Event	Represents the completion of a process
	Task	A specific unit of work or action performed
	Gateway	Used for decision points (e.g., XOR, AND)
	Sequence Flow	Indicates the order in which activities are performed
	Message Flow	Denotes communication between separate process participants
	Timer Event	Represents a time-based condition that triggers a process transition

Table 2.1: Essential BPMN symbols and their functions

Several studies underscore the importance of formalized process modeling in enabling digital transformation. Gabryelczyk et al. (2023) analyzed over 290 BPM use cases and found that organizations increasingly adopt Business Process Management not only for documentation, but as a strategic lever for improving transparency, responsiveness and operational performance. This strategic shift is reinforced by vom Brocke et al. (2016), who argue that BPM must be context-aware and aligned with organizational goals to drive real impact. The practical benefits of this approach are further supported by Teixeira et al. (2024), who applied the BPM lifecycle in a case study within the energy sector. Their findings show that formal modeling of previously informal and semi-structured workflows enabled data-driven analysis and guided targeted performance improvements, including selective automation and dashboard-based monitoring.

In contrast, many organizations still rely on informal tools such as spreadsheets, email and messaging applications to coordinate complex workflows. While such tools offer flexibility, they

often lead to fragmentation of knowledge, lack of traceability and communication inefficiencies. Wiśniewski et al. (2019) proposed a methodology for transitioning from spreadsheet-based documentation to formal BPMN models, emphasizing the loss of semantic structure and process coherence that occurs in informal representations.

Coordination Theory, proposed by Malone and Crowston (1994), is one of the most widely cited theoretical explanations for inefficiencies in informal workflows. The theory emphasizes that organizational processes require mechanisms to manage dependencies between tasks, such as shared resources, task sequencing or producer-consumer relationships. In decentralized workflows, where task outputs must seamlessly pass from one actor to another, the absence of formal coordination mechanisms leads to handoff failures, duplicated work and delays. BPMN plays a critical role in surfacing and managing these dependencies explicitly, making it an effective tool for minimizing coordination costs in fragmented environments (Malone and Crowston, 1994).

Within Small and Medium Enterprises (SMEs), the adoption of BPM poses both opportunities and challenges. Prior studies have shown that while SMEs can benefit from process modeling through increased productivity and responsiveness, they also face significant barriers such as limited organizational resources, low process maturity and resistance to formalization. These challenges are amplified in environments where operations rely on ad hoc coordination and lack standardized process documentation. According to vom Brocke et al. (2016), organizational context, including size, available resources and cultural support for BPM, plays a decisive role in the success of BPM initiatives, requiring tailored approaches for smaller firms with constrained capabilities.

Process mapping also plays a crucial role in supporting continuous improvement methodologies. As noted by Faria (2024), process models not only serve as documentation artifacts but also as analytical tools for identifying inefficiencies, diagnosing bottlenecks and evaluating redesign alternatives. This aligns with Lean thinking, particularly the principle of *Poka-yoke*, which emphasizes designing error-proof processes by preventing mistakes or surfacing them immediately for correction (Shingo, 1986). When properly implemented, BPMN models make process handovers, delays and redundancies visible, providing the foundation for these built-in quality mechanisms and reducing rework through early error detection.

Moreover, the utility of BPMN extends beyond mere visualization. It provides the groundwork for advanced techniques such as process simulation, performance analysis, and eventually automation through digital platforms. Recent research has emphasized BPMN's adaptability to decentralized execution environments, including implementations via blockchain-based smart contracts and multi-party collaborative workflows (Mendling et al., 2018). These capabilities strengthen BPMN's suitability for modern SMEs, particularly those operating with remote or distributed teams. This adaptability is particularly relevant given the architectural choices in Chapter 4, where lightweight, serverless infrastructure supports real-time coordination in a distributed reseller network.

In summary, the literature clearly supports the transition from informal, loosely defined workflows to structured, model-driven process management as a critical enabler of operational visi-

bility, efficiency, and digital maturity. The use of BPMN enables not only cross-functional understanding and stakeholder alignment but also lays the foundation for embedded coordination mechanisms, error prevention, and scalable digital execution. These capabilities are particularly valuable in decentralized, high-volume workflows such as the reseller onboarding and validation processes modeled and redesigned in this thesis.

2.2 CRM Integration and Customization

Customer Relationship Management (CRM) systems are central to managing interactions between businesses and their clients, offering structured environments for sales tracking, customer service, marketing automation and performance analysis (Buttle, 2009). Initially perceived primarily as IT systems supporting sales and service functions, modern CRM platforms have evolved into strategic tools encompassing operational, analytical, and customer-centric dimensions. Buttle (2009) explicitly distinguishes between strategic, operational and analytical CRM, highlighting how platforms such as Salesforce now support process automation, campaign management, data-driven segmentation and end-to-end service workflows. This evolution is particularly significant in distributed sales environments, where centralized CRM systems must interface with multiple user-facing platforms and back-office systems.

Modern CRM platforms have transformed from static data repositories into extensible, workflow-driven systems that support embedded business logic and enable end-to-end process automation (Almotairi, 2010). This shift holds particular relevance in distributed sales environments, where centralized CRM systems must integrate seamlessly with diverse front-end interfaces and back-office operations, an architectural challenge directly addressed in this thesis. However, for small and medium-sized enterprises (SMEs), this evolution also introduces considerable barriers. As detailed by Nguyen and Whelan (2012), many SMEs struggle with CRM customization due to limited IT capabilities, constrained budgets and misalignment between generic CRM functionalities and their domain-specific requirements. These findings underscore the importance of aligning CRM architecture with organizational characteristics to achieve successful adoption and long-term value realization.

From a software engineering perspective, CRM customization entails tailoring existing systems through declarative configurations or programmatic extensions to align with specific business needs. This process requires rigorous requirements engineering to avoid misalignment between the organization's operational logic and the platform's capabilities. As noted by Lenarduzzi and Fucci (2019), poor requirements elicitation and lack of formal specification during early development stages can lead to what is known as *requirements debt*, which is a form of technical debt that accumulates when informal, vague or incomplete requirements are implemented prematurely. This debt often manifests as maintenance overhead, integration inconsistencies and elevated long-term costs due to rework and suboptimal system behavior. These challenges are particularly relevant in digital transformation initiatives involving CRM integration within complex operational ecosystems, as explored in this thesis.

One of the key barriers in CRM integration is the synchronization of data across heterogeneous systems while maintaining data consistency, responsiveness, and security. As highlighted by Ruivo et al. (2014), integration between CRM and ERP systems is particularly complex due to the need for both system and process integration. Challenges commonly arise from misaligned data models, lack of standardization and integration logic that spans beyond internal systems to include customers and external platforms. These issues frequently lead to delays, rework and limited return on IT investment.

A modular, API-first architectural approach is increasingly recognized as the preferred method for integrating CRM systems with external applications. By exposing well-defined RESTful interfaces and supporting event-based messaging patterns, such architectures reduce coupling and simplify extensibility and testing (Nordli et al., 2023; Lazzari and Farias, 2022). Middleware layers, often implemented as API gateways, further support robust integration by abstracting authentication, logging, transformation and error handling, while facilitating secure communication with diverse platforms ranging from mobile apps to AI services (Nordli et al., 2023).

Middleware plays a crucial role in bridging CRM systems with external platforms. In the case of Salesforce, while its API ecosystem is mature, implementing bi-directional synchronization with external platforms often requires server-side middleware to handle authentication, queuing, transactional consistency, data transformation and robust error handling (Kumar and Rao, 2021). This middleware serves as a crucial abstraction layer for managing both API rate-limits and protocol translation, thereby enabling more resilient and scalable CRM integrations. This pattern is reflected in modern CRM architectures, which increasingly rely on modular middleware services to ensure flexibility and responsiveness across distributed components.

Moreover, integration is not solely about data exchange. As He and Erturk (2018) argue, well-integrated CRM systems can reduce process fragmentation and ensure that critical business logic is consistently enforced across different interfaces. This is particularly important in decentralized sales ecosystems, where external actors such as independent resellers interact with the CRM indirectly via customized workflows or mobile interfaces.

A notable challenge with off-the-shelf CRM user interfaces is their generic design, which may not suit all user roles. Independent resellers, for instance, often require streamlined, domain-specific workflows that are not supported by standard CRM frontends. Choudhury and Madan (2021) suggest that in such cases, developing custom interfaces that connect to the CRM backend through secure APIs offers greater control over the user experience, improves usability and increases stakeholder adoption. These recommendations are echoed in recent architectural studies of CRM platforms integrating AI and analytics, where interface customization plays a key role in ensuring alignment between user needs and backend capabilities (Berezin, 2025). This design principle guided the development of tailored interfaces in the CRM-integrated platform presented in this thesis.

In summary, the literature emphasizes the strategic importance and technical complexity of CRM integration and customization. Modern CRM platforms offer the potential for deep operational alignment, but realizing this potential requires structured approaches to integration, robust

middleware, and tailored user experiences. These insights frame the relevance of CRM systems in enabling digital transformation in sales-oriented organizations.

2.3 Leveraging Modern Technology Stacks for Operational Efficiency

Cloud computing has radically transformed the way digital platforms are designed, developed and deployed. Among the most impactful paradigms in recent years is *serverless computing*, which abstracts server management away from developers and enables them to focus on business logic while delegating infrastructure concerns to the cloud provider (Jonas et al., 2019). This paradigm is often implemented through *Function-as-a-Service* (FaaS) models that allow code to be triggered in response to events, scaling automatically and only consuming resources when executed (Baldini et al., 2017).

Recent studies have reinforced the benefits of serverless architectures for Small and Medium Enterprises (SMEs) and startups, particularly in terms of development speed and operational efficiency. Eivy (2017) highlights that serverless computing is especially effective for event-driven modular systems, such as those common in CRM-integrated platforms and automated workflows. However, he also cautions that serverless is not a one-size-fits-all solution, as its economic viability varies significantly with scale. As illustrated in the case study below (Table 2.2), while serverless infrastructure proves cost-effective at low traffic volumes, it can become substantially more expensive than traditional VM-based architectures under high-demand scenarios. This reinforces its suitability for SMEs and prototype-stage systems where workloads remain controlled and modular.

Hits Per Second	VM Infrastructure Cost	Serverless Cost
150	\$200/month	\$167/month
30,000	\$18,000/month	\$55,000/month

Table 2.2: Cost comparison between VM and serverless architectures at different scales (Eivy, 2017)

Building on these findings, Eismann et al. (2021) conducted a comprehensive study titled "*Serverless Applications: Why, When, and How?*", which synthesizes academic and industry perspectives to provide actionable guidance for serverless adoption. Rather than treating serverless as a universal solution, the authors offer a structured framework around three essential questions that help determine its suitability:

- **Why?** Serverless platforms reduce operational overhead by abstracting infrastructure management, accelerate development through simplified deployment pipelines, and follow a pay-per-use model—making them particularly cost-efficient for irregular or low-volume workloads.

- **When?** Serverless is ideal for event-driven systems or modular architectures where scalability and rapid iteration are priorities. It is especially advantageous during early-stage development or in scenarios with unpredictable usage patterns.
- **How?** Successful adoption involves designing stateless functions, using event-based triggers, and ensuring observability via logging and monitoring. Avoiding pitfalls such as vendor lock-in or limited execution time requires architectural discipline and planning.

These principles reflect many of the implementation decisions in this thesis, where serverless functions were used to handle user authentication, file uploads and integration with external APIs, providing scalability and maintainability with minimal infrastructure complexity.

Complementary to serverless computing is the emergence of *low-code* and *Backend-as-a-Service* (BaaS) platforms. These technologies provide developers with prebuilt components for common backend services such as authentication, storage and notifications. Platforms such as Firebase exemplify this trend by allowing rapid prototyping and production deployment with minimal backend code (Käss et al., 2022).

While serverless and low-code stacks offer clear benefits for accelerating development, particularly for SMEs, they also introduce important trade-offs. These include vendor lock-in, limited execution time for functions and challenges in debugging distributed event-driven logic (Eismann et al., 2021; Eivy, 2017). In cases involving compute-intensive or specialized workflows, traditional architectures may still offer better performance and control. Moreover, the simplicity and abstraction provided by low-code platforms can lead to architectural decisions that lack scalability or robustness if not designed with long-term maintainability in mind.

Nonetheless, the literature broadly agrees that when paired with sound engineering practices, serverless and low-code architectures can significantly improve productivity and maintainability, especially in process-oriented applications (Käss et al., 2022; Jonas et al., 2019). To guide such implementations, many developers turn to established architectural guidelines like the *Twelve-Factor App* methodology (Wiggins, 2017), which outlines best practices for building cloud-native applications that are scalable, portable, and maintainable.

Originally formulated by Wiggins (2017) during his time at Heroku, the Twelve-Factor App methodology has become a de facto standard for designing cloud-native applications, emphasizing declarative configuration, strict separation of concerns and adherence to portability principles. Its recommendations align closely with the design patterns encouraged by serverless and BaaS platforms. In the context of this thesis, backend practices such as API-first design, modular function deployment, and environment-based configuration were directly influenced by this framework. The complete set of principles is summarized in Table 2.3.

Factor	Purpose
1. Codebase	Maintain a single codebase with multiple deploys to ensure consistency.
2. Dependencies	Explicitly declare and isolate dependencies to simplify builds and deployments.
3. Config	Store configuration in the environment to separate code from environment-specific settings.
4. Backing Services	Treat attached services as replaceable resources.
5. Build, Release, Run	Separate the build, release, and run stages for reproducible deployments.
6. Processes	Execute the app as one or more stateless processes to enhance scalability and portability.
7. Port Binding	Export services via port binding for self-contained service deployments.
8. Concurrency	Scale out via the process model to support diverse workloads.
9. Disposability	Maximize robustness with fast startup and graceful shutdown.
10. Dev/Prod Parity	Keep development, staging and production environments as similar as possible.
11. Logs	Treat logs as event streams, sending them to a centralized log management system.
12. Admin Processes	Run administrative/management tasks as one-off processes in the same environment.

Table 2.3: The Twelve-Factor App principles

Overall, modern technology stacks, such as serverless computing and low-code platforms, provide compelling advantages for SMEs seeking to accelerate digital development. When combined with API-first design principles and robust middleware integration, these technologies enable the implementation of complex workflows while minimizing infrastructure overhead. As a result, they are particularly well-suited for process-oriented platforms that aim to increase operational efficiency and advance digital maturity.

2.4 AI-Driven Automation and Intelligent Systems

Artificial Intelligence (AI) has emerged as a transformative force in the digitalization of business operations, enabling organizations to augment decision-making, streamline processes and

automate complex tasks. Within the context of business process management and workflow improvement, AI is increasingly integrated not merely as an analytical tool but as a strategic enabler of intelligent automation (Davenport and Ronanki, 2018).

Intelligent automation refers to the integration of AI technologies, such as machine learning (ML), natural language processing (NLP) and computer vision, into automation platforms to support decision-making and task execution. As emphasized by Ransbotham et al. (2020), organizations that lead in AI adoption do not treat AI as a peripheral tool but embed it deeply within their operational workflows, using it to transform core business processes.

A growing body of literature highlights the value of AI-powered systems in enhancing productivity and accuracy in enterprise environments. Classic ML models, such as classification, clustering and regression, continue to underpin use cases like lead scoring and document classification (Shmueli et al., 2016). However, the recent surge in Large Language Models (LLMs) has introduced powerful capabilities for processing unstructured data and performing contextual reasoning, particularly in areas such as customer interaction, document analysis and process summarization (Bommasani et al., 2022; Dwivedi et al., 2023). These capabilities are being actively leveraged across different stages of the customer lifecycle, including lead generation, engagement and post-sale support, as illustrated in the examples below:

- **Lead Generation and Qualification:** Conversational AI and generative models can initiate personalized interactions across digital channels, capturing prospect data and evaluating lead quality through behavioral patterns and predefined criteria. As outlined by Deveau et al. (2023), these systems are transforming top-of-funnel sales operations by automating early engagement, dynamically identifying high-quality leads and tailoring outreach content at scale, thereby significantly improving lead conversion efficiency.
- **Sales Engagement and Transactional Support:** Humanized chatbots can manage mid-funnel interactions such as personalized follow-ups, service nudges and transactional assistance. Hildebrand and Bergner (2019) demonstrates that effective conversational design, particularly turn-taking and personalization, boosts trust and purchasing decisions, making these bots valuable in high-volume, engagement-intensive sales environments.
- **Customer Service and Retention:** NLP-enhanced chatbots support customer retention by delivering real-time, personalized assistance, handling repetitive queries, and escalating complex requests to human agents when necessary. As detailed by Ojha (2024), such systems improve response efficiency, elevate user satisfaction, and are increasingly being adopted across industries such as banking, e-commerce and telecommunications.

In the context of SMEs, intelligent systems can compensate for limited human resources by enabling scalable and responsive automation. However, deploying AI in such environments requires architectural discipline, transparency and a focus on explainability to ensure trust and effective user adoption (Guidotti et al., 2018; Sivarajah et al., 2017).

Although the AI landscape is rapidly evolving, the integration of intelligent components into business workflows remains most effective when grounded in clear objectives, data availability and maintainable deployment strategies. As will be demonstrated in later chapters, aligning intelligent automation with business goals is key to delivering measurable impact and supporting sustainable growth.

Chapter 3

Organizational Context and Problem Framing

Before proposing a technical solution, it was essential to understand the organizational setting in which it would operate. This chapter outlines the structural characteristics, commercial logic and coordination dynamics that shaped Company X's operational reality at the time of the project. It details the company's business model and its reliance on a decentralized reseller network, highlighting the coordination challenges and manual dependencies that arose from this setup.

Through a structured analysis of existing workflows, the chapter identifies systemic inefficiencies and recurring process failures across all stages of client onboarding. These issues were formally documented using BPMN diagrams and consolidated into a set of core problems categorized by impact and feasibility. This diagnostic phase laid the foundation for prioritizing interventions and ultimately informed the design of the digital platform described in Chapter 4.

3.1 Organizational Context

Before detailing the specific workflows and technical constraints that shaped the project's direction, it is essential to contextualize Company X within its operational environment. This section outlines the company's business model and stakeholder structure that defined its working reality prior to this project's development. Understanding the informal processes and communication breakdowns that emerged during its initial growth is fundamental to adequately frame the problems addressed in this thesis.

3.1.1 Market Position and Commercial Strategy

Company X is a commercial division of the XYZ Group, a Portuguese business group with over 25 years of experience in telecommunications, insurance and energy. Presently, Company X operates exclusively in the energy sector, focusing on the commercial distribution of electricity and solar services to both residential and commercial clients. This is made possible through partnerships with the leading energy providers in Portugal.

The company's go-to-market strategy is structured around a decentralized sales model supported by a network of independent resellers. These resellers operate under a commission-based incentive scheme and are responsible for prospecting new clients, collecting essential documentation and initiating the supply request process. In turn, Company X supports these field activities through a centralized backoffice team, which is in charge of validating submitted information, generating supplier proposals and managing the flow of documentation throughout each client process.

Resellers are compensated through a monthly commission defined at the proposal stage. The value of this commission depends on the tariff negotiated for the client and the expected duration of the contract, which typically spans five years. Since commission levels are inversely related to the tariff offered, the company must strike a balance between market competitiveness and reseller incentives.

At the time of this project, Company X maintained a grid of approximately 50 active resellers, with the strategic goal of doubling this number by the end of the year. This projected growth reinforced the importance of building operational processes that could support an increasingly distributed and information-dependent business model.

3.1.2 Organizational Structure

Operational activity within Company X is distributed across three primary stakeholders: resellers, backoffice personnel and external energy suppliers. Each plays a critical role in the lifecycle of a client process, from lead generation to service activation.

- **Resellers** – These field agents are responsible for identifying potential clients, collecting relevant documentation and submitting this material to backoffice staff for validation and proposal generation.
- **Backoffice** – The internal operations team verifies the information received from resellers, simulates tariff scenarios using a custom Excel model and interacts with the dedicated supplier's systems to generate contract proposals. This team is also responsible for document management, process follow-up and overall reseller support.
- **Suppliers** – External service providers who validate the final documentation and activate the contract once all conditions are met.

These stakeholder roles operate through loosely coordinated workflows developed organically during the company's early stages. Although the broader XYZ Group uses Salesforce as a centralized CRM system, Company X's day-to-day operations developed independently, relying on informal tools such as WhatsApp, email, Excel spreadsheets and local device storage. Consequently, once a client process is completed internally, the backoffice team is responsible for manually entering the relevant information into Salesforce to ensure that group-wide records remain consistent.

Communication between resellers and the backoffice team is typically carried out through non-centralized channels, with documents received in multiple formats and manually stored in local folders by individual team members. Tracking the progress of client processes depends on back-and-forth exchanges and personal notes, as there is no shared interface for centralized status monitoring. Moreover, resellers lack independent access to process updates and must inquire manually through these informal channels. While this system can be sustainable on a small scale, it does demand significant coordination and manual oversight, revealing a critical lack of integration and standardization that became a defining characteristic of the organization's operational model at the time.

3.2 Description of the Existing Workflow

The sales and onboarding workflow at Company X is a multistage process involving three primary actors: the reseller, the backoffice team and the energy provider. This structure supports the decentralized acquisition of new clients while centralizing validation, simulation and contract preparation tasks within the internal team. Although the client provides essential documentation and makes the final decision, they are not modeled as an operational actor, as all interactions are intermediated by the reseller and therefore fall out of scope for this analysis.

The complete process, from initial client contact to contract activation, unfolds across four logically cohesive stages. Each stage is defined by a specific operational goal, actor responsibility and transition logic. These stages involve frequent document exchanges and rely heavily on informal tools such as WhatsApp, email and local folders, resulting in coordination friction and limited process visibility.

A complete BPMN diagram of the workflow is presented in Figure 3.1. This model adheres to BPMN 2.0 standards and provides a formalized representation of the process, highlighting the interactions between internal actors across clearly defined swimlanes. It incorporates both sequence and message flows to capture task transitions and inter-actor communication, as well as data artifacts and a boundary timer event to model time-sensitive operations, such as proposal expiration.

The diagram was developed based on direct observation of operational routines, informal interviews with back-office staff and the review of internal process documentation, ensuring that the representation accurately reflects the as-is state of the organization's workflows. It served as a foundational tool for diagnosing inefficiencies and identifying rework patterns, enabling the formulation of concrete opportunities for process optimization.

The following subsections provide a detailed walkthrough of each operational phase, highlighting the specific actions, tools and coordination challenges encountered throughout the workflow lifecycle.

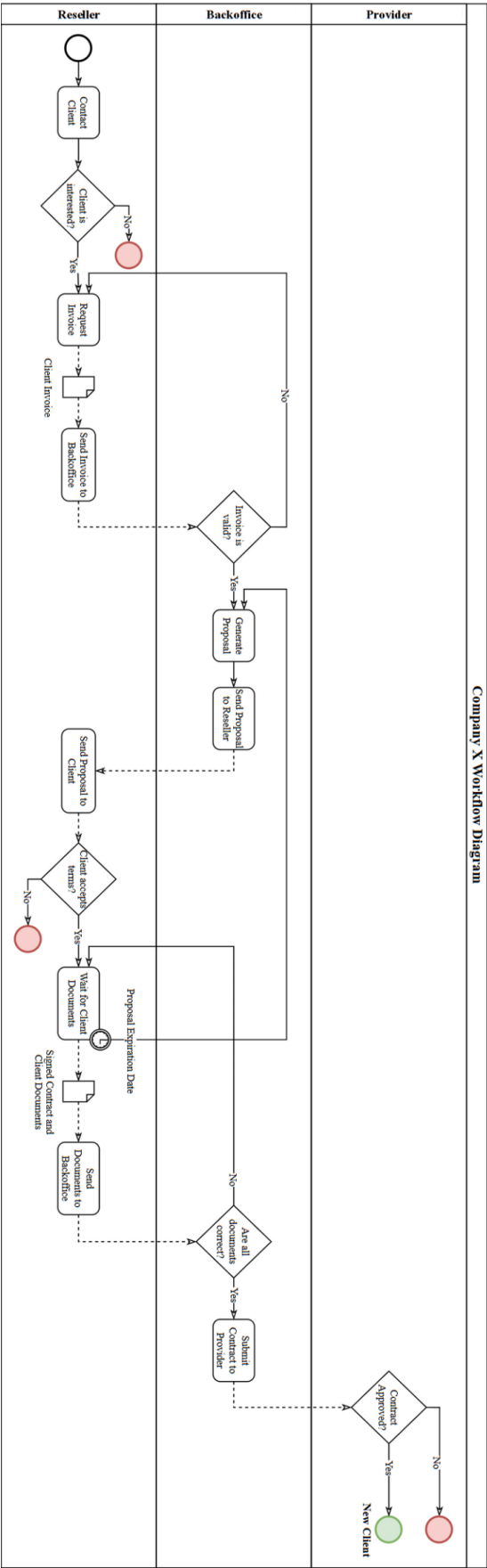


Figure 3.1: Complete BPMN diagram of the reseller workflow

Phase 1 – Client Identification and Invoice Collection

The first phase of Company X’s operational procedure begins with the reseller identifying and engaging a potential client. Company X strategically selects its resellers from among small businesses with existing commercial activity in sectors such as telecommunications, information technology or energy, thereby leveraging their pre-established customer networks for lead generation. While the responsibility for prospecting lies almost exclusively with the reseller, Company X does also provide lead support through internal mechanisms not covered in the scope of this analysis.

Once a prospective client is identified, the reseller is tasked with collecting essential onboarding information, most notably a recent utility invoice. This document is critical to enabling a contract simulation further down the process. However, the reseller is not required to assess the document’s technical validity, such as its legibility, date or completeness, as these checks are handled by the backoffice team. The reseller’s role is limited to obtaining and transmitting the invoice, with the possibility of repeating the request if quality issues arise.

All communication in this phase is carried out through informal and unstructured channels, mainly WhatsApp or email, without the use of standardized intake procedures. This decentralized approach introduces friction and variability into the process and serves as a precursor to the coordination and traceability issues that persist throughout the entire workflow.

Figure 3.2 illustrates the sequence of tasks and document exchanges involved in this initial stage.

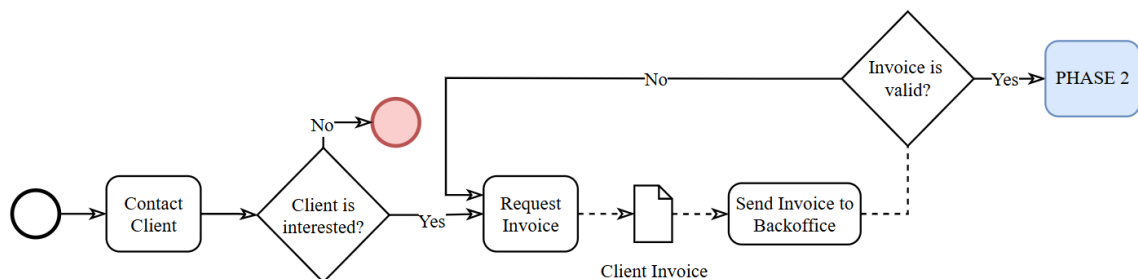


Figure 3.2: Workflow diagram for Phase 1

Phase 2 – Proposal Generation

After receiving a valid invoice, the backoffice team proceeds to analyze the client’s current energy contract by manually extracting key parameters such as contracted power, tariff type and historical consumption. This information is fed into an internal Excel-based simulation tool designed to calculate the client’s cost baseline and evaluate the potential for savings through a contract migration.

This evaluation is critical, as reseller commissions are inversely proportional to the discount applied to the client, meaning the lower the client's tariff, the smaller the reseller's reward. Company X must therefore balance price competitiveness with the economic viability of the proposal, ensuring that the incentive structure remains attractive without compromising margin targets.

When a viable opportunity is identified, the backoffice generates a contract draft through the provider's dedicated platform and prepares a formal proposal document. These materials are forwarded to the reseller, who is tasked with presenting the offer to the client and managing follow-up communication. The absence of an automated tracking mechanism for proposal delivery or client response creates a high dependency on reseller initiative, often requiring the internal team to manually confirm progress and maintain momentum throughout this stage.

Phase 3 – Document Validation and Signature

Once the proposal has been presented to the client, the decision to proceed rests entirely with them. In cases where the client rejects the offer, the process is typically discontinued, given that there is limited room for renegotiation, as the proposal is already structured to reflect the most competitive conditions available within Company X's internal thresholds.

If the client accepts the terms, they are required to provide a signed contract along with supporting documentation, namely proof of identity, proof of residence and any additional documents stipulated by the energy provider. These materials must be submitted within a limited time window, as proposals are sensitive to changes in market parameters. Failure to respond in a timely manner results in the expiration of the initial offer, requiring the backoffice to reprocess the opportunity and regenerate a new proposal, effectively restarting this phase.

Upon receiving the documents, the backoffice performs a thorough validity check to ensure that all materials are current and compliant with supplier requirements. If any document is missing, outdated, or otherwise non-compliant, the backoffice notifies the reseller and requests resubmission, creating additional delays and administrative effort. Only once the full documentation set is validated does the process advance to the final stage, where the completed package is submitted to the supplier for contract activation.

Figure 3.3 details the key decision points and document exchanges involved in the document validation and signature stage, including expiration triggers and rework scenarios.

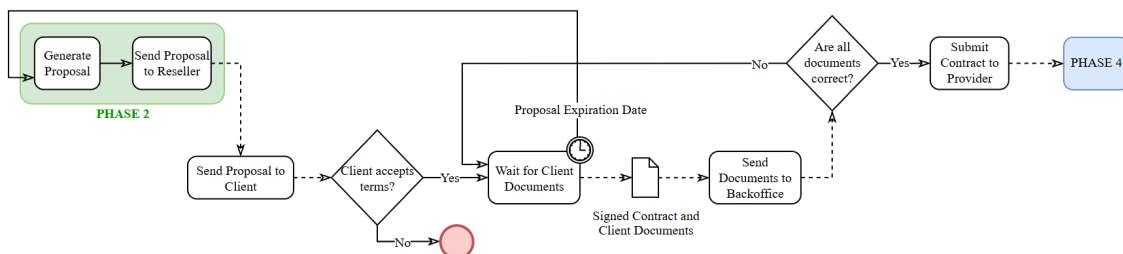


Figure 3.3: Workflow diagram for Phase 3

Phase 4 – Contract Activation

After validating the full documentation, the backoffice submits the package to the selected energy provider. The provider then reviews the application and issues a decision. If approved, the contract is activated and the reseller receives a commission. If rejected, the process is rarely restarted, as denials typically stem from structural electrical incompatibilities or client-side issues that are not easily resolved.

Additionally, to maintain consistency with group-level records, the backoffice performs a manual insertion of the client data into Salesforce, the CRM system used by the XYZ Group. This step, while necessary, introduces redundancy and increases administrative workload.

3.3 Problem Identification and Process Gaps

Based on the process mapping outlined in the previous section, a series of structural weaknesses and operational inefficiencies were identified across each stage of Company X's sales workflow. The following breakdown highlights the most critical pain points per phase, focusing on recurring issues such as delays, rework cycles, inconsistent data handling and lack of process observability.

Phase 1 – Client Identification and Invoice Collection

This stage relies entirely on informal and decentralized communication streams, typically carried out via messaging applications such as WhatsApp. While this approach offers flexibility and leverages the reseller's proximity to their client base, it presents substantial coordination challenges for the backoffice. As the volume of simultaneous client processes grows, tracking, validating and organizing new submissions and active workflows becomes increasingly unmanageable. The lack of formal intake procedures or standardized submission formats increases the likelihood of receiving outdated or illegible invoices, which often forces rework and delays. Additionally, the absence of structured document channels results in considerable context-switching effort on the backoffice side, as employees must manually navigate disparate messages, rename files and sort unstructured data sources. These low-value administrative tasks consume time and reduce operational efficiency.

Although Company X delegates all client-facing communication to resellers, thereby reducing internal workload, this model creates a logistical overhead in terms of asynchronous coordination and follow-up. The absence of a shared system to signal responsibility or confirm handoffs further contributes to potential bottlenecks, as accountability becomes difficult to enforce. While this delegation remains strategically beneficial, it also reinforces the need for structured intake channels and validation mechanisms to ensure process scalability.

Phase 2 – Proposal Simulation

This stage revolves around processing utility invoices to simulate new contract offers. After receiving a valid invoice, the backoffice must manually extract key data points, such as contracted

power, consumption and tariff details, from documents that vary significantly in format and quality. These values are entered into an internal Excel-based simulation tool used to calculate the client's cost baseline and produce a tailored offer. Although this workflow is functionally sound, it is highly manual and disconnected from surrounding systems, increasing the likelihood of transcription errors and making it difficult to scale without adding headcount.

Once the offer is generated, the same client data must be manually re-entered into Salesforce to ensure alignment with group-level CRM requirements. This duplication of effort (inputting the same structured data into two unlinked systems) represents the most critical inefficiency in the entire workflow. It violates core principles of information management and workflow design by introducing unnecessary touch points, increasing the risk of inconsistencies and diverting staff time toward non-value-added activities. From a Lean perspective, this step constitutes pure process waste, as it neither transforms the data nor adds operational value, and it is likely to become exponentially costlier as volumes increase.

In addition to these structural inefficiencies, the process also relies heavily on the tacit knowledge of backoffice personnel. Without standardized processing logic or embedded business rules, the quality of the final proposal may vary between operators. This makes outcomes harder to reproduce and exposes the company to knowledge continuity risks in cases of personnel turnover.

A further challenge lies in commercial decision-making. Company X must balance client savings with reseller commissions, which are inversely correlated. Yet, this trade-off is made on an ad hoc intuition-driven basis, with no formal decision support tool or profitability model. As a result, pricing consistency is limited, and margin management becomes reactive rather than strategic.

Finally, because proposals are delivered informally by resellers, the backoffice lacks visibility into the status of each offer. Whether the proposal was received, reviewed or declined remains unknown unless manually reported. This absence of structured engagement data not only complicates opportunity tracking but also undermines Company X's ability to learn from historical trends and improve conversion performance over time.

Phase 3 – Document Validation and Signature

In this phase, the reseller is responsible for collecting and submitting the signed contract along with supporting documents, typically including proof of identity, proof of residence and any additional files required by the energy provider. All submissions occur through informal channels and are reviewed manually by the backoffice. Despite its routine nature, this phase introduces significant coordination overhead and contributes disproportionately to workflow delays.

A key issue is the absence of any pre-validation mechanism or structured intake checklist. As a result, files are frequently missing, expired or improperly formatted. Each discrepancy triggers asynchronous back-and-forth exchanges between the reseller and the backoffice, increasing cycle time without adding value. These rework loops are especially problematic in a context with limited automation or submission controls.

Compounding this issue is the lack of process visibility or queue management logic. All client files are treated equally, regardless of urgency or progress status, and there is no systematic method to prioritize in-progress cases, track delays or categorize stalled submissions. Although Salesforce provides automated reminders for proposal expiration, these alerts are not integrated into a broader workflow management system that tracks real-time document status or guides resolution steps.

Finally, the absence of a structured audit trail undermines traceability. Document corrections and re-submissions are not versioned or time-stamped, making it difficult to track the history of a case or extract insights for process improvement. This gap poses challenges not only for internal performance analysis but also for potential compliance requirements in regulated environments.

Phase 4 – Contract Submission and External Finalization

In the final stage of the workflow, the backoffice submits the validated contract and supporting documents to the energy provider, typically through a proprietary supplier portal. This step marks the formal handoff of client data to external systems for contract activation. However, Company X has no direct control or visibility into the internal workflows or processing timelines of its supplier partners.

This external dependency introduces a latency that can adversely affect the client experience. Once documentation is submitted, the backoffice cannot accurately estimate when the contract will be activated, nor can it proactively update resellers or clients about the status. Delays on the supplier's end are often perceived by clients as a reflection of Company X's reliability, which undermines brand trust and can jeopardize the success of the onboarding process. The absence of a monitoring mechanism or escalation protocol compounds this problem, as the company must rely on manual portal checks or reactive communication to manage expectations and detect service issues.

Although this phase involves minimal internal complexity, its downstream nature makes it disproportionately impactful from a reputational and service assurance standpoint.

This detailed process walkthrough exposed a series of structural inefficiencies, rework patterns and coordination failures across all operational stages. To consolidate this analysis, Table 3.1 presents a synthesis of the most critical pain points identified during the process mapping effort. Each entry is categorized according to the stage in which it manifests, the nature of the problem and the underlying root cause.

ID	Stage	Problem Description	Type	Root Cause
A	Cross-cutting	Informal and decentralized communication channels across all stages	Coordination	No centralized platform or task routing logic
B	Phase 1	Inconsistent invoice quality and rework due to unstructured intake	Data Integrity	Lack of standardized submission format or validation
C	Phase 1	Coordination overhead caused by asynchronous messaging	Workflow Friction	No intake dashboard or submission management tool
D	Phase 2	Manual extraction of invoice data across heterogeneous formats	Scalability	No OCR pipeline or structured data capture mechanism
E	Phase 2	Duplicate data entry into internal databases and Salesforce	Redundancy	Lack of system integration or unified backend
F	Phase 2	Lack of engagement tracking after proposal delivery	Visibility	No read receipt, delivery confirmation or analytics system
G	Phase 2	Proposal pricing depends on intuition rather than formal logic	Decision Support	No optimization model to balance margin vs. commission
H	Phase 3	High rate of document rejection and resubmission cycles	Rework	No pre-validation of submission package or live checklist
I	Phase 3	No priority queue or triage system for document validation	Process Management	Flat intake logic regardless of urgency or status
J	Phase 4	No visibility or control over supplier-side activation delays	Service Assurance	External system with no feedback loop or SLAs

Table 3.1: Synthesis of core operational problems

This classification provides a structured baseline for evaluating potential interventions. While

not all problems carry equal weight, their systemic interconnections make it clear that piecemeal solutions may be insufficient. In the following section, we apply an impact-effort matrix to prioritize which issues should be addressed through digitalization and system redesign, forming the foundation for the platform architecture discussed in Chapter 4.

3.4 Strategic Targeting of Process Redesign

Building on the diagnostic mapping presented earlier, the ten core operational issues identified in Table 3.1 were further evaluated to determine which merited prioritization for intervention. Not all problems carry the same operational weight, nor are they equally feasible to resolve - some present clear opportunities for immediate improvement, while others are more complex or fall outside the organization's sphere of control. To support a rational and defensible scoping of potential reforms, each issue was analyzed in terms of its estimated resolution effort, both technical and organizational, and its anticipated impact on workflow efficiency, data integrity and business scalability.

This assessment is grounded in established frameworks from Lean Thinking and Business Process Management (BPM), which emphasize the importance of addressing high-friction, high-return bottlenecks as a prerequisite for sustainable process improvement and automation.

The resulting classification is presented in the **Impact–Effort Matrix** (Figure 3.4), which positions each problem in one of four strategic quadrants: *Quick Wins*, *Big Bets*, *Fill-ins* or *Avoid*. These categories provide the basis for a structured prioritization of interventions. What follows is a detailed justification for each classification, presented by quadrant. For every problem, the nature of the issue is revisited, a plausible solution is outlined and the reasoning behind its placement in the matrix is made explicit.

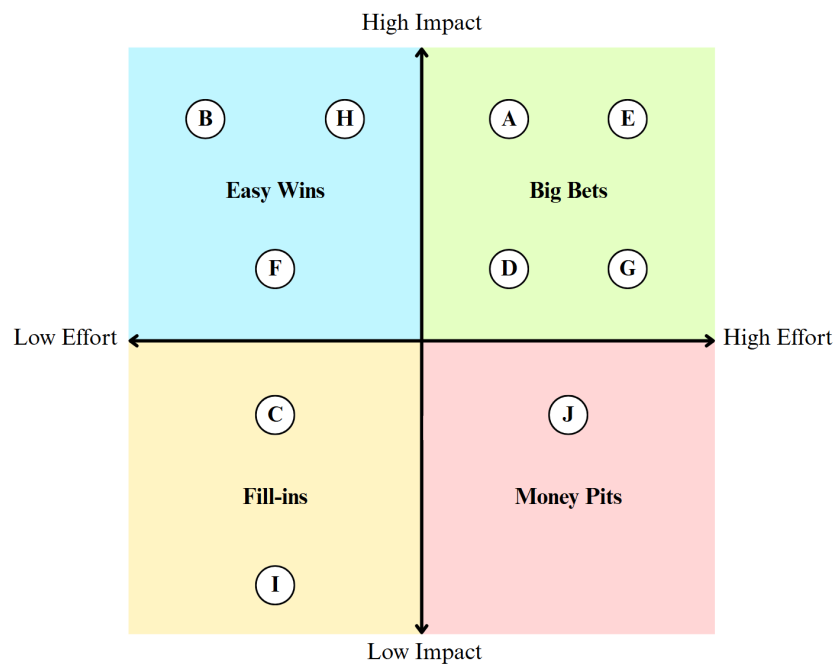


Figure 3.4: Impact–Effort Matrix used to prioritize intervention areas based on diagnostic findings

Quick Wins (High Impact, Low Effort)

These issues present substantial operational inefficiencies that could be addressed through targeted, low-complexity interventions. They are well-suited for early stages of process reform, offering meaningful improvements without requiring major organizational restructuring or long development cycles.

Unstructured Intake of Utility Invoices (Problem B) – The absence of a standardized method for submitting client utility invoices has led to frequent rework caused by incomplete or invalid documents submitted through informal channels. One possible intervention would involve establishing a structured submission mechanism with mandatory fields and basic file validation to ensure that incoming documentation meets predefined quality standards. This type of approach could significantly improve data quality and reduce administrative friction with minimal technical effort.

Lack of Engagement Tracking for Proposals (Problem F) – The current workflow lacks a mechanism for determining whether proposals sent to clients are being viewed, ignored or acted upon. This absence of feedback limits visibility into the sales pipeline and constrains the ability of staff to proactively follow up. A potential solution could involve implementing basic read-status indicators or delivery confirmations that provide insight into client engagement without introducing significant technical complexity.

High Rate of Document Rejections in Final Validation (Problem H) – In the final stage of the workflow, client documents are often rejected due to missing information, expired identification or non-compliant formats. These errors cause unnecessary delays and repeated back-and-forth between resellers and the backoffice. A feasible mitigation strategy would be to introduce a pre-submission checklist and apply simple validation rules at the point of intake to catch errors before they propagate downstream. This intervention could be implemented with minimal changes to existing procedures and would meaningfully reduce cycle time and rework.

Big Bets (High Impact, High Effort)

A second group of problems was categorized as Big Bets due to their foundational importance to long-term scalability and operational control. These issues require significant technical effort or organizational change to resolve, and although typically not suited for early-stage implementation, they are critical to enabling sustainable process transformation.

Informal and Decentralized Communication Channels (Problem A) – One of the most pervasive issues in the workflow is the use of informal communication platforms such as WhatsApp and email to exchange critical client information between resellers and the backoffice. This results in fragmented records, reduced traceability and inconsistent follow-ups. A structured messaging system would centralize communication and tie each conversation to a specific client process, improving auditability and coordination. However, implementation involves not only developing a chat infrastructure and notification logic, but also managing user adoption and replacing deeply ingrained habits. The combination of engineering effort and behavioral change makes this problem substantially complex.

Duplicate Data Entry into Internal Databases and Salesforce (Problem E) – During the proposal stage, client information must be entered both into an internal workflow tracker and again into Salesforce to satisfy audit and reporting requirements. This redundancy leads to inefficiencies and increases the risk of inconsistencies across systems. One possible intervention would be to establish a secure, bidirectional data synchronization between the internal workflow and Salesforce, using API middleware to ensure schema compatibility and token-based authentication. While technically feasible, such an integration would require managing cross-system consistency, handling access control and ensuring resilience against synchronization errors, making it a complex but potentially high-value improvement.

Manual Extraction of Invoice Data (Problem D) – The process of manually extracting data from utility invoices is both time-consuming and prone to human error. Automating this step would involve integrating an OCR (Optical Character Recognition) pipeline capable of parsing unstructured invoice formats and extracting key parameters such as tariff, power and consumption. Although the long-term benefits are clear — reducing processing time and improving accuracy — OCR implementation requires text recognition tooling, validation logic and exception handling, especially for edge cases and poor-quality documents.

Intuition-Based Pricing Decisions (Problem G) – The decision to approve a client proposal currently relies on human intuition to balance commission levels with client discounts. This introduces inconsistency and prevents structured margin management. A solution would involve formalizing business logic into a rule-based decision system or, in future iterations, integrating predictive analytics to evaluate profitability trade-offs based on historical deal data. The feasibility of such an approach depends on the availability and quality of past proposal records, as well as the organization's capacity to operationalize data-driven decision rules over time.

Fill-ins (Low Impact, Low Effort)

This category includes problems that are easy to address from a technical perspective but have limited strategic or operational significance when considered in isolation. While these items may offer marginal process improvements, they were not prioritized for early-stage implementation given their minimal contribution to overall workflow performance.

Coordination Overhead Caused by Asynchronous Messaging (Problem C) – Resellers and backoffice staff currently coordinate through asynchronous channels, often leading to delays or redundant follow-ups. While this introduces mild workflow friction, its root cause overlaps substantially with Problem B. The lack of structured intake procedures is the primary factor generating coordination overhead, as it forces manual clarifications and follow-ups. By introducing standardized intake mechanisms, much of the operational noise attributed to asynchronous messaging is inherently reduced. In this sense, Problem C represents a secondary effect of a higher-priority root cause. This rationale aligns with dependency modeling principles in BPM literature, which suggest downstream friction points should be deprioritized when their upstream drivers are being directly addressed.

Absence of Priority Queue or Triage Logic for Document Validation (Problem I) – All client submissions are currently handled on a first-come first-served basis, with no prioritization logic based on urgency, reseller tier or document completeness. Although this occasionally introduces inefficiencies, it does not represent a structural bottleneck, particularly at current process volumes. A lightweight triage mechanism could be implemented using tag-based filtering or urgency flags in the admin interface, but its benefits are marginal without broader workflow automation or higher transaction throughput.

Avoid (Low Impact, High Effort)

One of the problems identified during the process mapping was excluded from the scope of proposed interventions due to its limited impact on internal operations and the high effort required to address it. The issue lies largely outside the organization's sphere of influence, and resolving it would require changes to external systems or third-party coordination.

Lack of Visibility and Control Over Supplier-Side Activation Delays (Problem J) — Once a client's contract documentation is submitted to the external energy provider, Company X has no control over the processing time, nor access to real-time status updates. This creates uncertainty in the final phase of the onboarding workflow and may negatively affect client perceptions. However, resolving this would require integration with external supplier systems, service-level agreements or direct coordination channels, all of which involve a level of effort and dependency management that far exceeds the internal scope of this project. Given that this step occurs after all internal operations are complete and offers no feasible path for technical intervention, the problem was excluded from the intervention strategy.

3.5 Rationale for Platform Development

The decision to develop a custom digital platform was not an a priori assumption, but rather the logical conclusion of a structured diagnostic process. As detailed in the previous sections, Company X's operational workflow revealed systemic inefficiencies across all phases of the process. These inefficiencies, ranging from document rework to information asymmetry, fragmented communication and redundant data entry, could not be sustainably resolved through isolated procedural changes or the temporary adoption of off-the-shelf tools.

Instead, what emerged from the analysis was a clear pattern - the core problems stemmed from the absence of a shared, structured digital layer that could unify the different actors, standardize interactions and provide real-time visibility across the client life cycle. Company X did not lack software per se; Salesforce was already in use for CRM purposes. However, there was no integrated interface where resellers could operate efficiently and where the backoffice could coordinate, validate and govern the overall workflow. The problem was not one of individual tool selection, but of missing process infrastructure.

Given this context, the development of a custom platform became the most appropriate and strategic response. Conceptually, the platform was designed to serve two primary functions: first, it had to empower resellers to initiate and track client processes with clarity and consistency, enabling them to submit proposals, upload documents and receive feedback without needing to rely on informal communication channels or repeated back-and-forth exchanges; second, it had to give the backoffice team a centralized interface through which to validate submissions, manage priorities, enforce compliance standards and maintain synchronization with the organization's broader CRM ecosystem.

More concretely, the platform was scoped to resolve the high-priority problems identified in the Impact-Effort Matrix through tightly integrated design decisions. The issue of unstructured document intake (Problem B) was addressed by implementing a guided submission flow that collects all required files and client data in a standardized format. The lack of engagement tracking for proposals (Problem F) was mitigated through a proposal status tracking interface, enabling resellers to verify whether offers were delivered, opened or declined. Frequent document rejections

(Problem H), typically caused by missing or invalid uploads, were reduced through embedded validation steps and pre-submission checklists. Informal and fragmented communication (Problem A) was replaced by an internal messaging module linked to each client record, ensuring that all exchanges are contextually anchored and easily traceable.

By resolving Problems A and B, the platform also indirectly eliminates the coordination overhead associated with asynchronous messaging (Problem C), as resellers now operate within structured submission workflows and have access to centralized communication channels. Finally, the critical inefficiency of duplicate data entry into Salesforce (Problem E) was addressed by enabling proposal approvals and CRM synchronization from within a unified interface, reducing rework and ensuring consistency with group-level audit requirements.

Table 3.2 summarizes the specific operational problems addressed by the platform and the mechanisms through which each was resolved.

Problem	Problem Description	Solution Mechanism in Platform
A	Informal communication channels	Integrated in-app messaging module tied to each client process
B	Unstructured intake of utility invoices	Guided submission flow with mandatory fields and validation
C	Coordination overhead from async messaging	Resolved via structured submissions and centralized messaging
E	Duplicate data entry into Salesforce	Direct proposal approval with automatic data sync to CRM
F	Lack of proposal engagement tracking	Proposal status tracking interface with reseller visibility
H	Document rejection due to missing/invalid uploads	Pre-submission checklist and inline validation during uploads

Table 3.2: Mapping of addressed problems to platform solution mechanisms

Crucially, this solution was designed to be accessible across both mobile and desktop environments. This decision was central, since most resellers operate in the field, often using their phones to manage client interactions, while the backoffice works from fixed locations. Ensuring full cross-platform functionality was therefore a structural requirement, not an implementation detail. The platform needed to accommodate both modalities from the outset, with a consistent user experience and synchronized access to data.

By consolidating fragmented workflows into a single operational layer, the platform also provides a foundation for future capabilities. While the immediate goal is to standardize and control

core operational flows, the system is intentionally modular. In later iterations, it could support automated decision support, AI-powered assistant agents, predictive analytics based on proposal and client data or rule-based triage systems to prioritize work based on urgency and strategic value. None of these would be feasible without first establishing a unified, traceable and process-aware foundation.

In short, the platform is not just a technological solution, but a structural intervention. It is the process layer that Company X was missing - a system designed not just to digitize tasks, but to coordinate and govern them at scale. Its development was grounded in the diagnostic, shaped by the organization's specific needs, and guided by a commitment to resolving not just symptoms, but root causes.

Chapter 4

Solution Architecture and Implementation

This chapter presents the technical response to the operational challenges outlined in Chapter 3. Based on the diagnostic findings, it became clear that fragmented communication, document inconsistency, manual data entry and lack of process observability were not isolated incidents, but systemic issues stemming from the absence of a structured digital layer. To address this, the project developed a modular, cloud-native platform, *X-App*, designed to serve as a centralized process infrastructure connecting field resellers, backoffice staff and external systems like Salesforce.

The following sections detail the platform’s architecture, technology stack and core functional modules. Each design choice is tied directly to the problems diagnosed in the existing workflow and guided by a clear set of design principles prioritizing scalability, traceability and security. Together, these components form a coherent system tailored to the operational context of Company X and designed for long-term extensibility.

4.1 Overview and Design Principles

The *X-App* was conceived as a direct response to the systemic inefficiencies documented in Chapter 3. The diagnostic phase revealed a set of recurring operational issues across the reseller onboarding and proposal lifecycle, including unstructured document intake, fragmented communication channels, repeated document rework and duplicate data entry into disconnected systems. These problems not only increased process overhead but also constrained Company X’s ability to scale its operations or enforce consistent compliance across a growing reseller network.

Rather than attempt to resolve these challenges through isolated procedural changes or disconnected tooling, the analysis pointed to a deeper structural gap: the absence of a centralized, process-oriented digital layer. As argued in the previous chapter, the logical solution was to develop a platform with two fundamental purposes. First, it would support resellers in conducting compliant, trackable commercial interactions from their smartphones. Second, it would equip the backoffice with the tools to validate, monitor and synchronize data flows across internal and

external systems. The aim was not merely to digitize individual tasks, but to introduce a shared operational interface capable of governing the end-to-end reseller process.

This chapter presents the resulting technical solution and its architectural logic. It outlines the system's structural design, core platform components, integration mechanisms with Salesforce and layered security strategy. The implementation reflects a modular and cloud-native design, purposely built to resolve the pain points identified in the diagnostic phase while remaining extensible enough to support future developments such as AI-powered assistant agents and predictive workflow features.

X-App adopts a three-tier architecture, comprising a client-side interface, a backend service orchestration layer and external system integrations. This approach follows standard procedure of decoupling the presentation layer from the backend logic, improving maintainability and enabling platform-agnostic development. A diagrammatic representation of this architecture is provided in Figure 4.1.

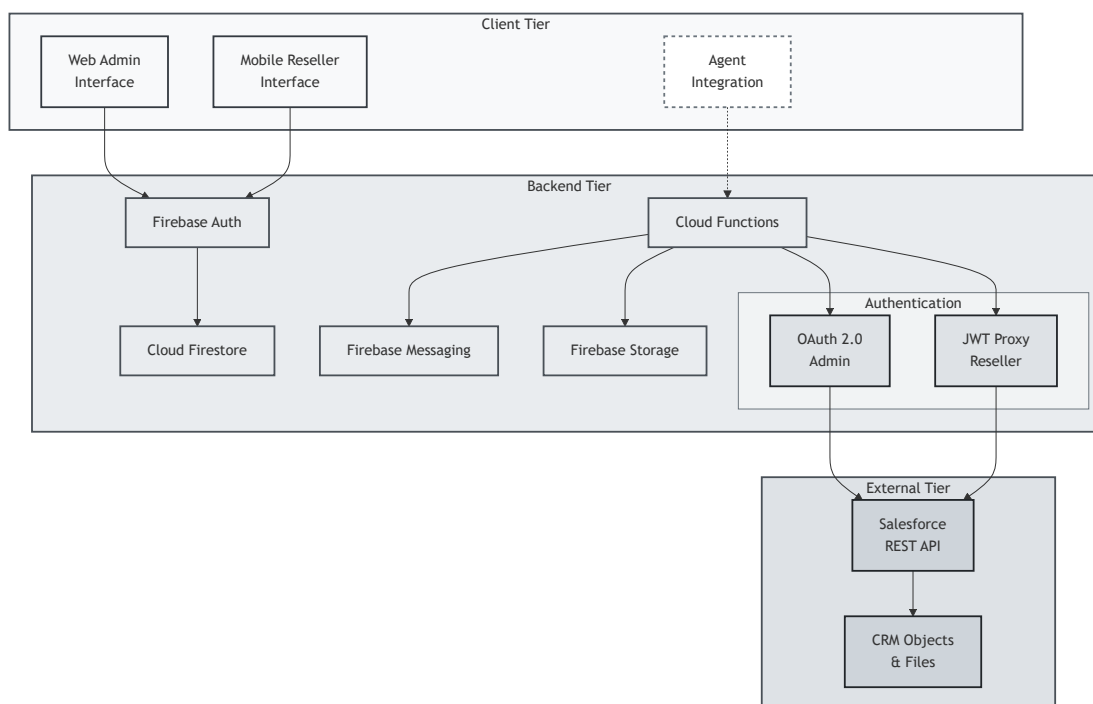


Figure 4.1: Architecture of the X-App system

The frontend tier consists of web and mobile compatible pages developed using Flutter. Both interfaces are unified under a responsive design model that adapts across devices and screen sizes. Components are built using a shared widget library to ensure consistency and reduce redundancy. The mobile-first approach was adopted to support resellers who primarily operate via smartphones, while still ensuring that administrative interfaces remain robust on desktop.

The backend logic is fully reliant on Firebase's serverless platform. Firebase Authentication provides secure user login and token management, Firestore Database serves as the real-time document store supporting the communication channel and transactional workflows, and Firebase

Storage handles document uploads, particularly for proposals and client verification documents.

Business logic is encapsulated in Cloud Functions, which can be triggered by both user actions and backend events. These functions are responsible for handling Salesforce API communication, playing a key part in integrating the company's CRM into this platform. The serverless model was selected to minimize infrastructure management overhead, reduce operational complexity for a lean development team and allow effortless scaling during peak usage.

External service integration is centered on Salesforce, which remains the system of record for all operational data. Other services, such as Firebase Messaging and analytics tools, extend the platform's communication and monitoring capabilities.

The platform's architecture was guided by a set of design principles that reflect both its technical constraints and operational goals. These principles ensure alignment with long-term scalability, ease of maintenance and role-based control over sensitive business processes.

1. **Single Source of Truth** – Salesforce maintains full ownership of business data, while Firestore offers local caching and user-centric data operations. This reduces synchronization conflicts and ensures business continuity.
2. **Mobile-First Responsive Architecture** – Built with Flutter, the UI layer offers a consistent experience across web, iOS and Android. This design simultaneously supports the needs of the reseller grid and central backoffice, which are fundamentally distinct, though not diametrically opposed.
3. **Role-Based Access Control** – Clear distinct permission layers between admin and reseller functions. Firebase custom claims and Firestore rules are used in tandem to control both UI and backend access.
4. **Microservices Architecture** – Business logic is divided across isolated Cloud Functions, each scoped to a single domain-specific task. This modular structure improves maintainability and testability while also contributing to platform security. Sensitive credentials are stored as encrypted environment variables and are only accessible within the execution context of server-side functions. By design, these secrets are never exposed to the frontend or client-side code, ensuring strict privilege boundaries and minimizing the surface area for potential security breaches.
5. **API-First Integration** – The application enforces a clean separation between the frontend interface and backend logic through a strict API-first design. All business operations are encapsulated in callable or REST-exposed endpoints, allowing for seamless communication between system components while abstracting implementation details. This architecture not only simplifies cross-platform consistency but also lays the groundwork for future extensibility. By maintaining well-defined API boundaries, the platform becomes compatible with additional clients, including third-party systems or intelligent agents capable of querying, submitting and managing commercial workflows autonomously.

In summary, the architecture of *X-App* is purposefully modular and scalable, directly aligning with present operational requirements and future ambitions. It reflects deliberate engineering trade-offs aimed at balancing performance and flexibility, offering a technically sound and context-appropriate solution tailored to the company's operational landscape.

4.2 Tech Stack Summary

X-App implements a cloud-native technology stack optimized for agility, modularity and long-term maintainability. The frontend is developed using Flutter (SDK 3.7.0+), a UI framework built in Dart that supports simultaneous deployment across web and mobile platforms from a single codebase. Flutter was selected over alternatives such as React Native due to its superior integration with Firebase, strong support for native performance across both iOS and Android and its unified widget rendering engine. Both Flutter and Firebase are maintained by Google, ensuring ecosystem-wide compatibility, with reliable updates and coherent development tooling across the application's components.

The backend architecture is structured around Firebase's serverless service suite. Firestore is used as the NoSQL document database, offering real-time synchronization and offline access, both critical features for supporting a geographically distributed reseller network. Firebase Authentication manages user sign-in and role-based access, while Firebase Cloud Storage handles all document uploads, such as invoices submitted with service proposals. Business logic is encapsulated in Firebase Cloud Functions, implemented in TypeScript and deployed on the Node.js runtime (v22), allowing for modular, event-driven computation. The application's web version is hosted via Firebase Hosting, which provides a global CDN and automatic SSL provisioning for secure and low-latency delivery.

This serverless stack was chosen over traditional monolithic or containerized architectures due to its reduced operational complexity, built-in scalability and tight integration with frontend tooling. Alternatives such as Supabase or custom Node.js backends were considered, but Firebase offered the most seamless ecosystem alignment, particularly given the need for real-time updates and Salesforce integration.

Salesforce serves as the system of record for all commercial and client-related data. Integration is performed through direct REST API calls (v58.0), with administrative access authenticated via OAuth 2.0 and reseller access managed through server-side JWT bearer tokens. This dual-mode model ensures secure, role-based data access while maintaining compliance with internal governance policies.

In addition, the application integrates a set of third-party libraries that streamline development and enforce architectural best practices. Riverpod is used for reactive state management due to its declarative architecture and strong support for testability. GoRouter handles declarative navigation logic, Dio manages HTTP communication and Flutter Secure Storage provides encrypted local token handling. The complete list of technologies and their justifications is presented bellow (Table 4.1).

Component	Technology	Purpose
Frontend Framework	Flutter SDK 3.7.0+	Cross-platform UI for web and mobile
Programming Language	Dart	Application logic and interface components
Backend Logic	Firebase Cloud Functions (TypeScript, Node.js v22)	Event-driven backend services and business logic
Database	Firestore	Real-time NoSQL database with offline support
Authentication	Firebase Authentication	User sign-in, session handling and role-based access
File Storage	Firebase Storage	Upload and management of documents and images
Web Hosting	Firebase Hosting	Static hosting with SSL and global CDN
CRM Integration	Salesforce REST API	Synchronization of client and sales data
OAuth Provider	OAuth 2.0	Secure admin authentication via Salesforce
Token Management	JWT (server-side)	Secure reseller access to Salesforce resources
State Management	Riverpod	Declarative and testable state architecture for Flutter widgets
Routing	GoRouter	Declarative navigation between app screens
HTTP Networking	Dio	API request handling and error management
Secure Local Storage	FlutterSecureStorage	Encrypted storage of authentication tokens and credentials

Table 4.1: Core components of the X-App technology stack

Firebase's free-tier pricing allows for cost-effective operation during early-stage usage, particularly suitable for pilot environments and limited-scale deployments. Importantly, this affordability does not compromise scalability, since the architecture can scale vertically and horizontally as user demand increases, without requiring major changes or vendor migration.

At the time of writing, mobile deployment to the Play Store and App Store has not yet been executed due to time constraints within the dissertation development period. Nonetheless, the

application's Flutter architecture includes full support for both Android and iOS devices. Platform-specific configuration files, asset pipelines and build procedures have been finalized, ensuring that once developer accounts and store metadata are prepared, the deployment process can proceed using standard distribution pipelines via Google Play Console and Apple App Store Connect.

While the selected stack offers significant advantages in terms of productivity and scalability, it also introduces trade-offs. Firebase's proprietary infrastructure entails a degree of vendor lock-in, and the debugging of serverless functions, especially in asynchronous workflows, can be more opaque than in traditional hosted environments. These limitations were considered acceptable in light of the project's lean structure and the pressing need for rapid development.

Overall, the selected technology stack reflects a deliberate alignment between engineering efficiency and operational pragmatism. Its integration of Google's unified development tools, native CRM synchronization and low infrastructure overhead results in a robust and scalable foundation for both current workflows and future agent-based extensions.

4.3 Feature Catalogue

This section presents the core functional modules that make up the *X-App* platform. Each feature was designed to address specific operational inefficiencies identified during the diagnostic phase (see Chapter 3) and is tightly integrated with the platform's backend architecture and CRM synchronization logic. The catalogue is organized around five key modules: Authentication and User Management, Service Submissions, Opportunity and Proposal Management, Message and Notifications, and Client Management. Together, they operationalize the platform's process-centric vision, enabling secure and scalable execution of reseller workflows across the platform's interfaces.

4.3.1 Authentication and User Management

Although not directly linked to a standalone inefficiency in the diagnostic matrix, the authentication and user management system forms the foundational layer of the *X-App* platform. It governs how users access the system, how permissions are enforced and how role-based logic determines the entire application experience. Beyond ensuring secure access, it functions as a dynamic router, deciding whether a user is directed into the administrator console or the reseller interface.

X-App enforces a strict two-role model. All users are assigned either the `admin` or `reseller` role at the time of account provisioning. These roles are mutually exclusive and have access to entirely distinct workflows and interface components. Upon successful authentication, each user is presented with a role-specific interface optimized for their operational context.

Resellers interact with a mobile-optimized dashboard that supports proposal submission, document upload, real-time messaging and status tracking (Figure 4.2). In contrast, administrators access a web-based control panel for managing users, validating submissions, generating Salesforce opportunities and handling CRM-linked metadata (Figure 4.3).



Figure 4.2: Reseller home page interface

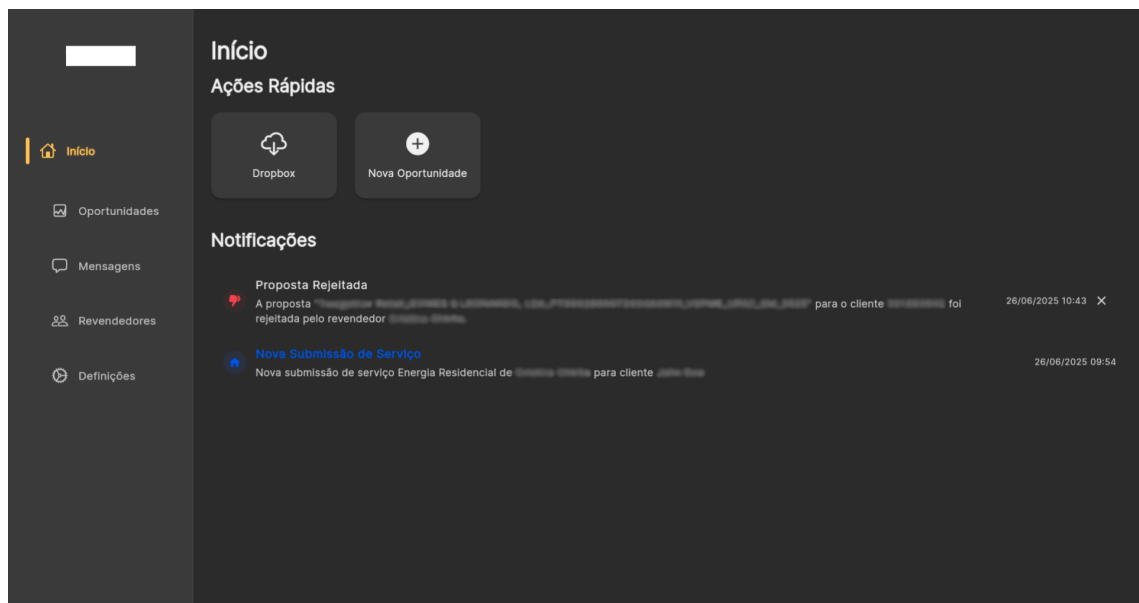


Figure 4.3: Admin home page interface

Account creation follows a closed provisioning model, where only administrators can create new users. This is done through a secure admin interface that links new users to existing Salesforce entities via synchronization scripts. Each reseller account is required to include a valid `salesforceId`, which ensures CRM alignment, allows record filtering based on ownership, and is used later to fetch opportunities and calculate commissions.

Role-based access control is enforced through Firebase Authentication and encoded as custom claims in each user's session token. These claims define frontend permissions and gate access to backend logic via Firestore security rules and Cloud Functions. The authentication system also ensures identity protection through the use of encrypted token storage mechanisms such as `flutter_secure_storage` and web-based equivalents.

The authentication layer is more than a security mechanism — it serves a foundational functional role in *X-App*. By enforcing a strict separation between administrator and reseller roles, it enables the operation of two distinct digital workspaces, each tailored to the specific responsibilities and workflows of its users. This structure not only enhances usability and compliance, but also provides the architectural basis for controlled and traceable role-specific interactions across the entire platform.

4.3.2 Service Submissions

One of the key pain points identified in the diagnostic phase was the unstructured intake of utility invoices (Problem B), which led to frequent rework and inconsistent data quality. The service submission module directly addresses this issue by introducing a standardized and verifiable workflow for initiating commercial opportunities within the *X-App* platform. It serves as the system's primary intake mechanism, transforming informal client referrals into structured and reviewable submissions.

The submission workflow is implemented as a multi-step guided form within the reseller interface, as illustrated in Figure 4.4. It begins with the selection of the service category and the classification of the client as either residential or commercial. Based on these inputs, the reseller is prompted to select the appropriate service provider from a predefined list. Resellers then input the required client details, including the client's name, tax identification number (NIF), email address and phone number. A supporting document, specifically a recent utility invoice, is mandatory and must be uploaded directly through the interface. These files are then stored in Firebase Cloud Storage and their download URLs are saved in Firestore as part of the submission metadata.

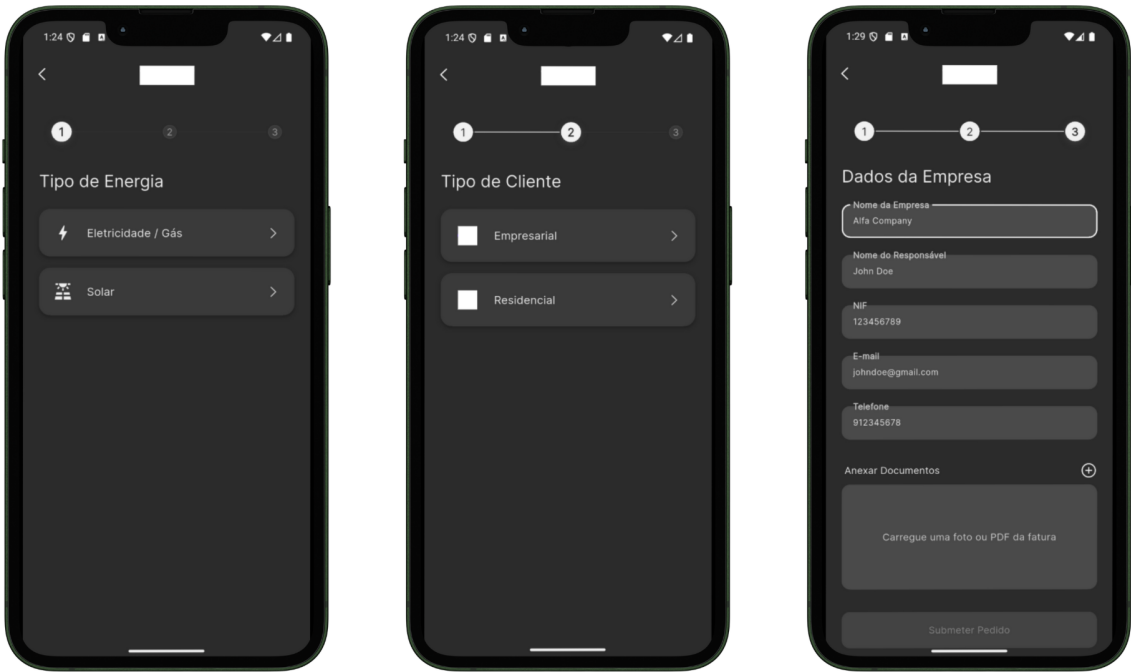


Figure 4.4: Multi-step service submission form used by resellers to initiate structured onboarding requests within the *X-App*

Each submission is stored as a document in the `serviceSubmissions` collection within Cloud Firestore and automatically assigned a `pending_review` status. This record is immediately surfaced in the administrative dashboard, where back-office personnel can inspect the data, preview attached files and complete any additional metadata required for Salesforce compatibility, as shown in Figure 4.5.

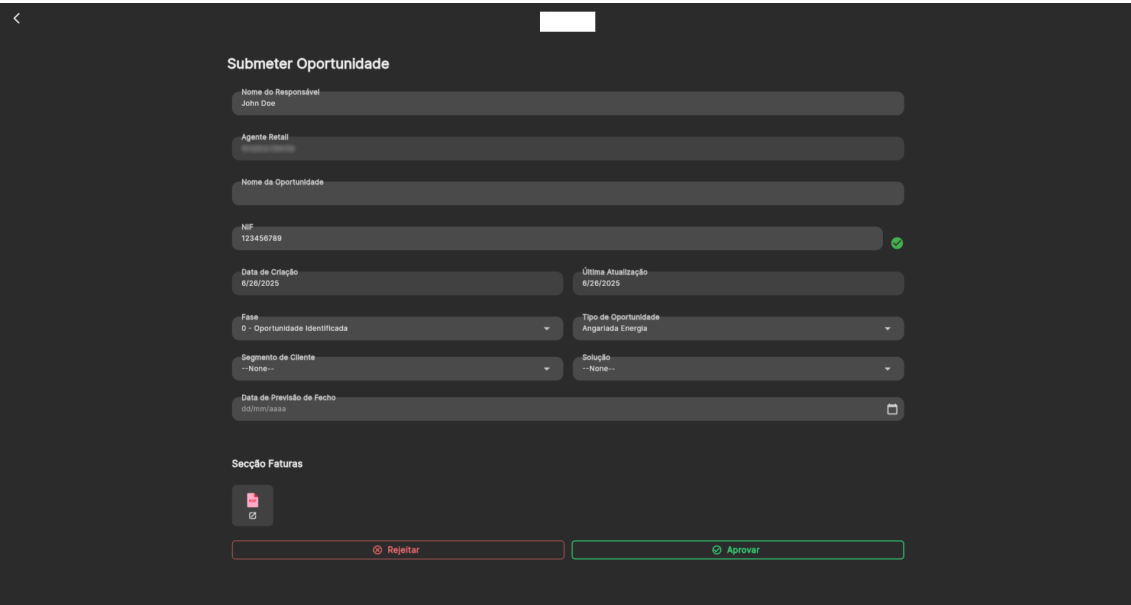


Figure 4.5: Administrative review interface where submissions are analyzed and approved or rejected

As part of this workflow, the system allows for NIF field validation against existing Salesforce accounts. Once the record is deemed complete and accurate, administrators can manually trigger a Cloud Function (`createSalesforceOpportunity`) to create the corresponding Opportunity and Proposal records within Salesforce. This ensures that only verified submissions are escalated to the CRM, preserving data integrity and minimizing downstream rework.

The result of each review is recorded in the submission's `status` field and communicated to the reseller through in-app notifications. Rejected submissions may include metadata explaining the rejection reason, while approved ones are cross-referenced with their Salesforce Opportunity ID, ensuring traceability and auditability.

This module plays a foundational role in operationalizing the platform's process logic. By enforcing structured input and introducing a review-based control layer, it addresses one of the most persistent sources of inefficiency in the previous workflow. Its implementation bridges the gap between informal reseller practices and formalized CRM operations, enabling scalable and compliant lead onboarding.

4.3.3 Opportunity and Proposal Management

The Opportunity and Proposal Management module governs the end-to-end process by which administrative users convert new service submissions into formal CRM records within Salesforce. Its primary purpose is to address one of the most critical inefficiencies identified in Chapter 3: the redundant and error-prone manual entry of client and proposal data into Salesforce by back-office staff (Problem E). This bottleneck was not only time-consuming but also introduced frequent inconsistencies between operational records and the CRM system. By automating the creation of Opportunities and Proposals through a structured workflow, this module eliminates duplication, ensures data integrity and establishes Salesforce as the single source of truth for all commercial records.

Following the approval of a service submission, an opportunity is created, with each opportunity being able to contain one or more associated proposals. Proposals are created through a dedicated administrative workflow that allows the back-office to define pricing terms, validity periods, service types and required documentation. This process includes the integration of Customer Premises Equipment (CPE) records. Through the `createCpeForProposal` function, each proposal can be linked to one or more CPE entries that describe technical and commercial attributes such as consumption estimates, fidelization periods and equipment specifications. This modular structure supports flexible quoting and service configuration, which is particularly important in scenarios where a single client holds multiple energy contracts, each requiring a distinct technical configuration.

The hierarchical relationship between Salesforce objects (Opportunity, Proposal and associated `CPE_Proposal`) is visually represented in Figure 4.6, highlighting how multiple CPEs can be grouped under a single commercial proposal.

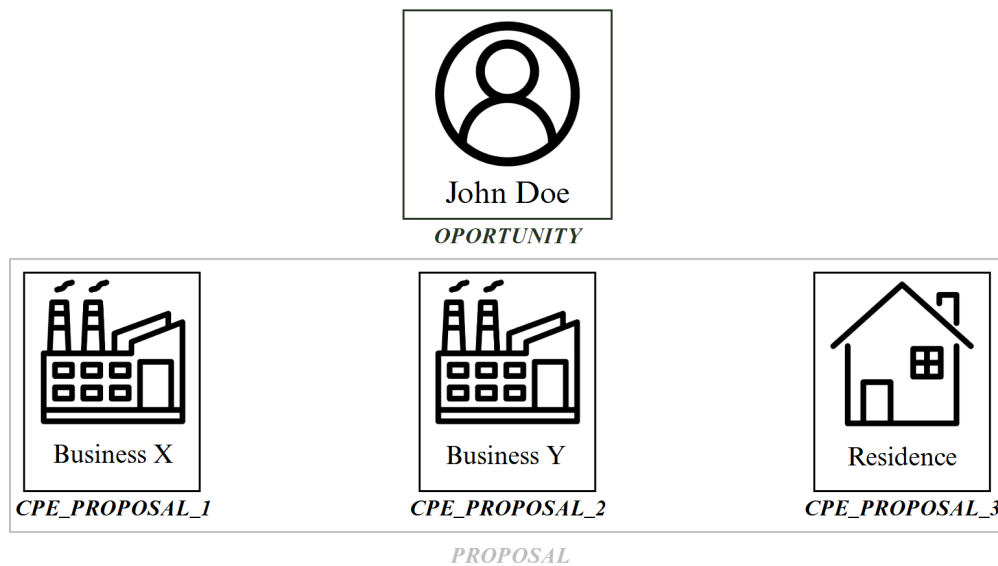


Figure 4.6: Relationship between Salesforce objects

Once a proposal is prepared and approved by the administrator, the reseller can initiate the formal acceptance workflow via the `acceptProposalAndUploadDocs` function. This process includes client-side document validation, automated file upload to Salesforce and downstream record creation in Salesforce.

Each proposal is assigned a status reflecting its lifecycle stage: *Enviada*, *Em Aprovação*, *Aceite*, *Não Aprovada* or *Cancelada*. These statuses follow a predefined schema established in Salesforce, ensuring consistency with existing CRM workflows and compliance standards. The application does not persist status values locally in Firestore; instead, it fetches the current status from Salesforce on-demand, treating it as the single source of truth. Any status update triggered within the platform is propagated directly to Salesforce through secured Cloud Functions. To maintain stakeholder awareness, status changes trigger automated push notifications via Firebase Messaging.

The administrative dashboard consolidates opportunity and proposal records, providing functionality for searching and filtering. Filters include reseller ID, client type, service category and Salesforce status. The interface allows seamless navigation between linked entities, such as transitioning from a service submission to its corresponding opportunity and attached proposals. Integrated document handling enables file preview and upload via Salesforce APIs. Resellers are granted read-only access to their associated opportunities and proposals through secure Cloud Function endpoints, including `getResellerOpportunities`, `getResellerProposalDetails` and `getResellerOpportunityProposals`.

Ultimately, this module transforms validated service submissions into structured CRM opportunities and manages the complete sales lifecycle. By embedding real-time synchronization and modular configuration capabilities, *X-App* directly resolves multiple structural inefficiencies in the reseller onboarding workflow and provides a robust operational interface for opportunity tracking and sales execution.

4.3.4 Message and Notifications

This module directly addresses Problem A from Chapter 3: the reliance on informal communication channels such as WhatsApp and email, which previously fragmented conversations, delaying decision-making and hindering traceability. To mitigate these issues, *X-App* introduces a structured, in-platform messaging system that facilitates real-time communication between resellers and the back-office team.

Each reseller is automatically assigned a dedicated one-to-one conversation channel upon first use of the chat feature. This channel remains persistent throughout their activity on the platform and supports text messages, as well as image attachments or general file uploads. Messages are stored in Firestore using a hierarchical model, with each conversation as a document and associated messages as subcollection entries. This structure supports real-time synchronization across all user devices. Notifications are dispatched via Firebase Cloud Messaging whenever new messages are posted and the recipient is inactive, ensuring timely visibility while minimizing disruption.

Figure 4.7 shows the administrative view of the messaging panel, where incoming messages are organized by reseller and displayed with real-time indicators and attachment previews.

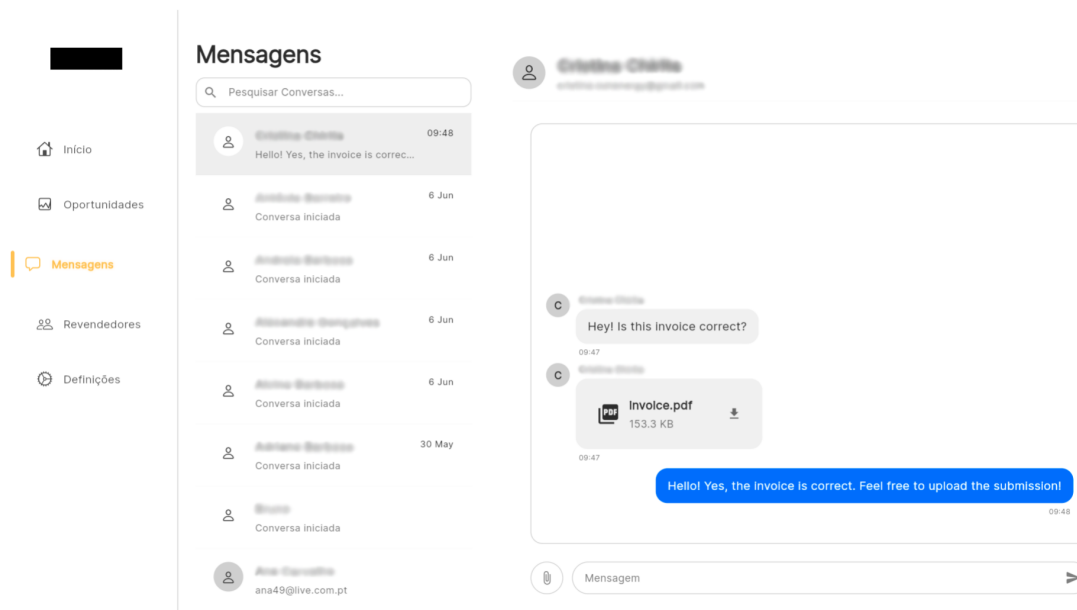


Figure 4.7: Administrative messaging panel displaying real-time communication with resellers, including message history, file attachments and live updates

The chat interface includes a file preview widget that allows users to quickly inspect uploaded media and documents directly within the conversation thread, without requiring external downloads. This way, the interface provides a familiar and accessible messaging experience for both resellers and administrators. All interactions are governed by strict Firestore security rules that limit access to verified conversation participants. Backend services manage notification delivery

and scheduled data cleanup, ensuring both real-time responsiveness and long-term compliance with internal data retention policies.

By centralizing communication within the platform, this module eliminates the operational ambiguity of external messaging apps and strengthens overall process transparency, particularly during high-stakes proposal reviews or documentation handovers.

4.3.5 Client Management

This module addresses two critical problems identified in Chapter 3: the lack of engagement tracking for proposals (Problem F) and the high rejection rate in final validation due to inconsistent documentation (Problem H). By equipping resellers with real-time visibility and structured client follow-up tools, the platform ensures proactive proposal handling and standardized document compliance.

The client management module enables resellers to oversee their assigned commercial opportunities throughout the lifecycle of engagement. It provides a filtered dashboard that lists Salesforce-linked opportunities associated with the reseller's Salesforce ID. Figure 4.8 shows the interface used by resellers to track active opportunities, each annotated with visual status indicators that reflect the current stage of the associated proposal. These icons provide at-a-glance clarity on whether a proposal is awaiting reseller action, under review by the back-office or requires no immediate intervention, thereby supporting efficient prioritization and timely follow-up.

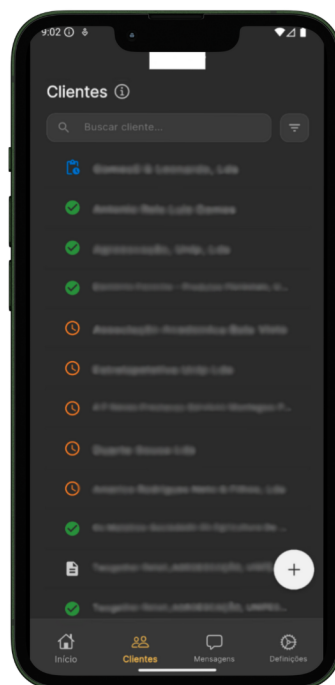


Figure 4.8: Reseller dashboard displaying assigned opportunities and their statuses

Each dashboard entry displays essential client metadata, including name and current proposal

status. Search and filtering features allow resellers to quickly locate specific clients or opportunities based on name or lifecycle stage, supporting timely follow-up and case prioritization.

Within each opportunity, resellers can access detailed proposal information, including pricing terms and validity periods. When proposals reach actionable states, the reseller is prompted to either accept or reject the proposal on behalf of the client. Acceptance triggers the second phase of the workflow: structured document submission. Figure 4.9 shows the proposal details view and the integrated interface for uploading validation documents required for CRM finalization.

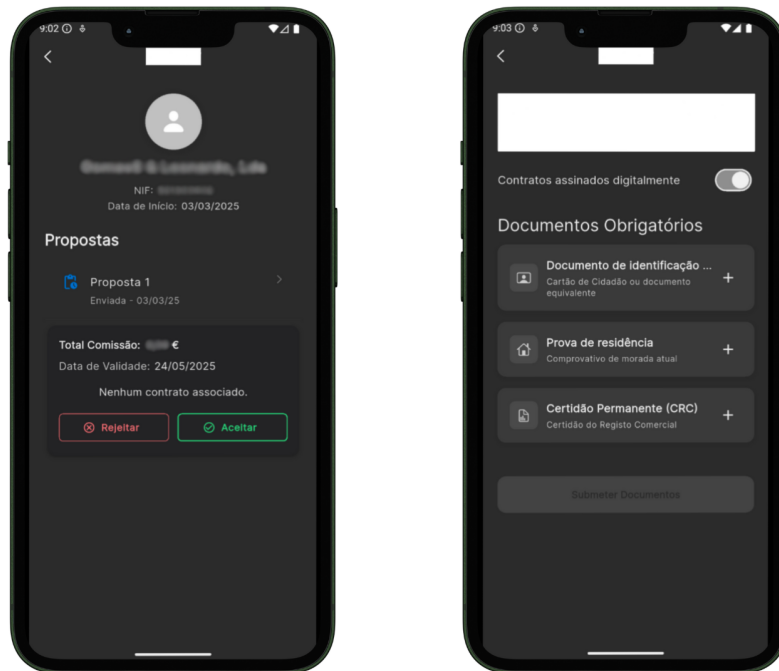


Figure 4.9: Proposal details and associated document submission interface

Uploaded files are securely relayed to Salesforce via backend Cloud Functions, which verify file types and structure before attaching them to the appropriate Opportunity record using Salesforce's `ContentDocumentLink` objects. This mechanism ensures both traceability and compliance with internal validation requirements.

In summary, this module empowers resellers to take an active and compliant role in the proposal lifecycle. It replaces back-and-forth communication with real-time data access, supports structured document validation and ensures synchronization with Salesforce workflows, directly mitigating two of the most persistent operational pain points observed in the original workflow analysis.

4.4 Integration Architecture

The integration architecture of *X-App* defines how the platform connects its internal frontend and backend systems with external services, most notably Salesforce CRM. This integration layer was critical to resolving key operational pain points identified in Chapter 3, including redundant data

entry (Problem E), fragmented workflows (Problem A) and limited visibility into process status (Problem F). To ensure reliability and process traceability, the platform implements a combination of real-time data coordination via Firebase and role-differentiated synchronization with Salesforce through controlled API access. The following subsections outline how these patterns are implemented in practice.

4.4.1 Salesforce CRM Synchronization

X-App integrates directly with Salesforce as its primary CRM system, enabling secure, real-time synchronization of commercial data. This integration addresses a core operational pain point identified in Chapter 3, the need for redundant manual entry of client and proposal information into Salesforce by the back-office team (Problem E). By replacing these ad hoc processes with secure and programmatic synchronization, the platform ensures data consistency across systems while reducing coordination overhead and rework. The architecture supporting this integration is built around a dual-authentication model, a centralized Connected App and a suite of server-side functions that mediate all interactions with Salesforce.

Dual Authentication Model

To ensure strict role-based access to CRM data, the platform implements two separate authentication flows: one for administrative users and one for resellers, based on OAuth 2.0 and JWT Bearer Flow, respectively. These dual flows, illustrated in Figure 4.10, underpin the system's integration with Salesforce by enforcing differentiated access paths aligned with each user's operational role.

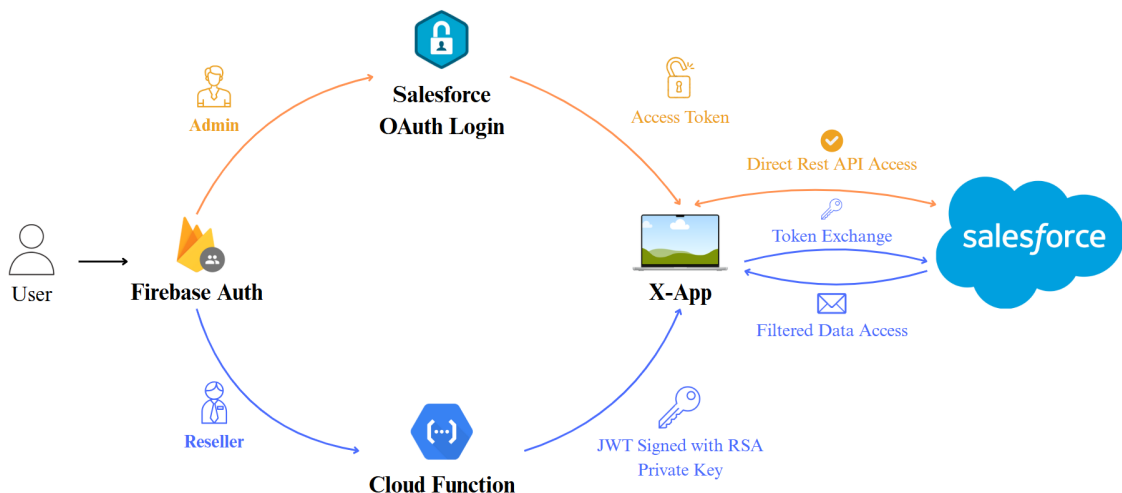


Figure 4.10: Comparison of Salesforce authentication flows for admins and resellers

Administrative users authenticate via the OAuth 2.0 Authorization Code Flow with PKCE. This interactive process redirects users to the Salesforce login portal for secure credential entry. Upon successful login, an authorization code is exchanged for access and refresh tokens, which are stored locally and grant full access to the Salesforce REST API. This method is used exclusively

by back-office personnel to perform privileged operations, such as opportunity creation, account management and document uploads.

Resellers, by contrast, never interact directly with Salesforce. Instead, their access is mediated through a secure JWT Bearer Flow. After authenticating with Firebase, a backend Cloud Function issues a JWT signed with an RSA private key (stored securely in encrypted environment variables). This JWT is exchanged server-side for a scoped Salesforce access token. The resulting token permits only limited, read-only access to objects explicitly associated with the reseller's Salesforce ID, such as their opportunities and proposals.

Conceptually, the two flows can be illustrated through the metaphor of access control. OAuth operates like a VIP badge, granting full entry into a secured environment after identity verification. Conversely, JWT Bearer Flow resembles a concierge system, where the user makes a request that is verified and fulfilled by a trusted proxy, without ever accessing the secure area themselves.

This design enforces the principle of least privilege and significantly reduces security risk. No sensitive credentials are ever exposed to the client. All reseller interactions are funneled through validated Cloud Functions, ensuring full auditability and compliance with internal governance policies.

Together, these authentication models support a balanced architecture, providing flexibility for administrators while enforcing strict constraints on reseller activity.

Connected App Foundation

Both authentication flows are anchored on a Salesforce Connected App, which acts as the secure entry point for all third-party API interactions. This app is configured with the following elements:

- **X.509 Certificate Upload** – The public certificate derived from the JWT signing key is uploaded to Salesforce, enabling it to verify the authenticity of incoming JWTs.
- **OAuth Scopes** – Including `api`, `offline_access` and `refresh_token` for OAuth-based admin access and minimal scopes for JWT-based reseller access.
- **Pre-Authorization Setup** – Access is restricted to “Admin approved users” and dedicated integration profiles defined in the org.
- **Profile Binding** – JWT tokens are issued in the context of a system-level integration user, whose Salesforce profile determines visibility and access scope.

This configuration provides a robust framework for maintaining controlled access to Salesforce data, ensuring that both admin and reseller interactions occur within predefined boundaries. Setting up the Connected App required direct configuration within the Salesforce CRM environment, a technically involved but essential process when enabling third-party platform integration. It forms the backbone of the platform's authentication and authorization architecture, enforcing strict identity control and role-based access to CRM data.

Integration Patterns and API Usage

Salesforce API interactions are handled via the `jsforce` Node.js library, embedded within Firebase Cloud Functions. These functions perform operations on both standard and custom Salesforce objects, including `Oportunidade`, `Proposta` and `CPE_Proposta`.

Document exchange is facilitated via Salesforce's `ContentDocument` and `ContentVersion` APIs. When a proposal is accepted, required documentation is uploaded to Firebase Storage, then relayed to Salesforce through backend functions. Files are linked to their corresponding Opportunity records via the `ContentDocumentLink` object, ensuring bidirectional traceability and CRM compliance.

Security and Compliance

All sensitive secrets (including the Salesforce consumer key, client secret and JWT private key) are stored as encrypted environment variables in Firebase. These values are never exposed to the client or frontend environment. Access to Cloud Functions is gated by Firebase Authentication and validated using custom claims, ensuring that only authenticated users can trigger integration logic.

API rate limits, session lifetimes and error responses are centrally monitored. OAuth tokens are refreshed using long-lived refresh tokens stored locally on the client device, while JWT tokens are regenerated on-demand via backend services.

By encapsulating Salesforce access within authenticated and audited Cloud Functions, *X-App* guarantees strict compliance with internal security requirements, minimizing privilege exposure and ensuring that all CRM interactions are both traceable and revocable.

4.4.2 Firebase Service Integration

The *X-App* backend architecture is built upon the Firebase ecosystem, leveraging its suite of services to orchestrate real-time data management, secure identity verification, push notification delivery, file storage and scalable business logic. Firebase operates as the core coordination layer, ensuring responsive user experiences across platforms while maintaining strict access controls and integration consistency.

This Firebase-driven integration layer was critical for addressing the fragmented operational model outlined in Chapter 3, particularly by enabling real-time process visibility (Problem F), centralized messaging and unified document workflows (Problem A).

Authentication and Authorization

User identity is managed through Firebase Authentication, which issues ID tokens used for access control throughout the system. These tokens are evaluated in Cloud Functions and Firestore

security rules to enforce permission boundaries. Role-based access is implemented through Firebase custom claims, automatically set via the `handleUserRoleChange` Cloud Function, ensuring dynamic role enforcement without exposing sensitive business logic on the client side.

Unlike the external Salesforce integration flows described previously, Firebase’s internal authorization is used for backend function gating and role-conditional UI rendering. Custom claims are used to check the user role, assigning distinct permissions within the app’s administrative and reseller interfaces.

Firestore Data Architecture and Access Control

Cloud Firestore serves as the primary NoSQL database, structured around five core collections: `users`, `conversations`, `serviceSubmissions`, `notifications` and `providers`. Each collection uses hierarchical document structures, such as nested `messages` subcollections under `conversations` and `files` under `providers`.

Granular access control is enforced through Firestore security rules. For instance, conversation access rules verify the user’s UID against the conversation’s `resellerId`, ensuring that only participants can read or write messages. Administrative users are granted full read/write access across collections, while resellers operate within tightly scoped permissions based on their authenticated identity.

Collection	Purpose	Access Scope
<code>users</code>	User profiles, roles, FCM tokens	Admins only
<code>conversations</code>	Reseller-admin chat threads	Admin + Reseller
<code>serviceSubmissions</code>	Proposal submissions by resellers	Admin + Owner Reseller
<code>notifications</code>	Push notification records	Admin + Owner Reseller
<code>providers</code>	Available energy service providers	Read-only to all roles

Table 4.2: Firestore collections and access scope

Cloud Functions and Backend Orchestration

The backend logic is implemented through over 40 Cloud Functions. These functions support critical operational domains such as user management, Salesforce integration, notification delivery and secure file transfer between Firebase and external systems. Each function is scoped to a specific task, promoting modularity, maintainability and strict access control.

Administrative functions verify elevated permissions, while reseller-targeted endpoints ensure data isolation by comparing Salesforce IDs before returning data.

To ensure system resilience, functions are configured with controlled timeouts, structured error propagation and automated retries for transient failures. UI-level guards are used throughout Flutter components to prevent post-unmount state access after asynchronous operations.

Hybrid File Storage Model

X-App employs a dual file storage strategy. Files relevant to internal app operations, such as chat images, provider documents and invoice uploads, are stored in Firebase Storage. In contrast, formal documentation destined for CRM storage is uploaded to Salesforce through Cloud Functions using base64 encoding. These files are not stored in base64 locally, encoding is only applied during upload. Each document is linked to Salesforce records via `ContentDocumentLink` relationships, maintaining traceability and compliance with business requirements.

Notification System and Messaging Integration

Push notifications are implemented through Firebase Cloud Messaging (FCM), triggered by back-end events and synchronized with persistent notification records in Firestore. User documents include an array of `fcmTokens` to support multi-device delivery, and invalid tokens are automatically purged based on FCM response codes.

Each notification follows a standardized metadata structure that records sender, recipient, related object and timestamp. This persistent notification layer allows users to view their notification history across devices, while Cloud Functions such as `sendMessageNotification` handle real-time delivery orchestration.

Viewed holistically, Firebase acts as the central nervous system of *X-App*, coordinating user authentication, access control, data storage and inter-service communication. Its tight integration with the Flutter frontend and ability to orchestrate secure Cloud Functions ensures that operational workflows remain scalable by design.

4.4.3 Cross-Service Communication

The *X-App* architecture implements secure cross-service communication through Firebase Cloud Functions, which serve as authenticated intermediaries between the Flutter frontend and Salesforce CRM. All Salesforce interactions follow a synchronous request-response pattern using HTTPS callable functions under Firebase Functions v2. This design explicitly avoids Firestore-triggered or event-driven flows, ensuring predictability and traceability.

Each function performs strict input validation and role-based access control, using Firebase Authentication and custom claims to gate permissions. Request payloads are transformed to comply with Salesforce's `SObject` schema, including field mapping (e.g., from `snake_case` to `camelCase`) and type normalization for dates and decimals.

Error handling is implemented through structured responses. When Salesforce tokens expire, functions respond with a `sessionExpired` flag, allowing the client to gracefully reauthenticate. Limited retries are used for transient failures, but no exponential backoff or retry queues are implemented.

Function configurations reflect operational complexity: 60 seconds for standard API calls, 120 seconds for file handling and up to 540 seconds and 1 GiB memory for complex multi-step operations. These constraints ensure reliable performance without over-provisioning.

In summary, this communication model supports centralized backend enforcement, secure data exchange and real-time error signaling. It plays a critical role in addressing many of Chapter 3's problems, by ensuring operational traceability and eliminating undocumented side-channel workflows.

4.5 Security Architecture

Security was identified from the outset as a non-negotiable requirement by Company X, given the platform's role in handling sensitive client data, contractual documents and CRM credentials. As such, security considerations were embedded into every stage of the system's design and implementation. The resulting architecture ensures that all user interactions, data access patterns and cross-system integrations are governed by strong guarantees of integrity, confidentiality and compliance. The system's security model is structured around two primary boundaries: identity enforcement within Firebase and scoped CRM access via Salesforce. This section outlines the key strategies used to manage authentication, enforce role-based access, protect data and secure third-party API integrations.

The platform uses Firebase Authentication as its centralized identity provider. Upon user login, a cryptographically signed ID token is issued, which encapsulates user identity and role metadata. These tokens are automatically refreshed and verified on every backend operation, ensuring that access decisions are made using current credentials. Role assignments are managed via Firebase custom claims and evaluated both in Firestore rules and in backend Cloud Functions. This enables fine-grained permission checks without requiring additional reads or exposing role logic to the frontend.

To propagate access boundaries across system components, Salesforce authentication follows a dual-flow strategy. Administrators use OAuth 2.0 with PKCE to initiate secure login sessions, which produce short-lived access and refresh tokens stored locally on the device. Mobile clients use `FlutterSecureStorage` for encrypted token storage, while web clients rely on `sessionStorage` or `SharedPreferences`. Resellers never handle Salesforce credentials directly, since their access is mediated by a server-side JWT Bearer Flow, where a Cloud Function signs a JWT using a private RSA key and exchanges it for a scoped Salesforce token. This design guarantees that CRM access for resellers is read-only, filtered by Salesforce ID which never exposed to the client side.

All sensitive credentials, such as the Salesforce JWT private key and OAuth secrets, are stored as encrypted environment variables within Firebase. These are accessed only at runtime by privileged backend functions, with private key material loaded and transformed into PEM format solely during signing operations. Session expiration and error conditions are monitored through specific

Salesforce response codes and the frontend is notified via structured error responses that prompt secure reauthentication flows.

Access to Firestore data is governed by declarative security rules that verify user claims before permitting reads or writes. Rules use reusable helper methods to control access to sensitive collections such as service submissions, chat conversations and user metadata. Nested subcollections inherit parent-level security policies, ensuring that access control remains consistent across the document hierarchy.

API security is enforced by making all sensitive operations callable only through Firebase Cloud Functions, which act as controlled intermediaries between the frontend and external systems like Salesforce. Every function performs layered validation: first by verifying the caller's Firebase ID token and role claims, then by checking input structure and document ownership. Functions return standardized error responses and prevent information leakage by sanitizing Salesforce error messages before relaying them to the client.

This layered approach ensures that *X-App* maintains full control over sensitive operations. By combining platform-native security primitives with architectural separation of roles and credentials, the system achieves a high level of integrity while remaining lightweight and maintainable.

4.6 Deployment and Operational Practices

The deployment strategy of *X-App* reflects a deliberate balance between operational complexity and organizational scale. Rather than adopting a fully automated CI/CD pipeline, the system follows a structured manual deployment process using Firebase CLI tools. This decision aligns with the adopted project lean structure and the platform's moderate release cadence, favoring clarity and control over automation overhead.

Deployments are managed through well-defined procedures that coordinate updates to the Flutter web frontend, backend Cloud Functions and Firestore rules. Local testing, access validation and smoke testing are performed prior to each release, ensuring functional stability and compliance with platform dependencies. The Salesforce integration layer, in particular, requires careful sequencing of token updates and endpoint availability, which is more reliably handled in a manual process.

While the deployment process is manual, it incorporates quality gates such as Firestore rule validation and backend permission testing. Risk is mitigated through built-in Firebase rollback features: version history for hosting, function-level rollbacks and CLI-based rule restoration. This infrastructure provides sufficient safety for a system of this scope, without the need for custom recovery pipelines.

Operational resilience is further supported by a layered error handling approach. Authentication failures, token expirations and quota-related interruptions are captured and surfaced to the frontend through structured Firebase error codes. A conservative retry policy is applied to critical operations like token refreshes and file downloads, preventing infinite loops or unnecessary load. Additionally, pseudo-circuit-breaker behavior is implemented client-side to temporarily halt

function calls after repeated failures, ensuring predictable performance even under partial system degradation.

Real-time responsiveness is achieved through Firestore snapshot listeners managed with Riverpod's `StreamProvider`. This architecture supports chat, notifications and status updates without polling, while avoiding unnecessary overhead by selectively pausing listeners during app backgrounding. External CRM data is fetched via manual refresh calls to accommodate Salesforce's rate-limited, eventually consistent APIs.

This strategy exemplifies the principle of “appropriate technology”: the deployment process is intentionally simple and reliable, especially suited for a small development team managing a modular, cloud-native system. It establishes a reproducible foundation that enables safe iteration and leaves room for future automation once the operational context justifies it.

Chapter 5

Conclusion

Following the development and deployment of the *X-App* platform, this chapter assesses the solution’s operational alignment, system-level impact and broader strategic implications. While the platform was initially scoped to resolve a targeted set of workflow inefficiencies, its implementation revealed structural shifts in how Company X conducts and governs its reseller-driven operations. This chapter provides a critical reflection on the platform’s benefits and limitations, both technical and organizational, and outlines the conditions necessary to realize its full potential.

The discussion is structured into four sections. First, the achieved benefits are reviewed in terms of process alignment, usability and operational coherence. Second, limitations are acknowledged, including current system gaps and organizational barriers to adoption. Third, future development paths are explored, ranging from incremental automation to the speculative deployment of autonomous AI resellers. Finally, the chapter concludes with strategic reflections on the role of intelligent systems in SME process transformation.

5.1 Summary of Work Conducted

This project addressed the systemic inefficiencies present in Company X, a reseller-based energy company operating under a decentralized commercial model. Prior to this intervention, the company relied on informal communication channels, unstructured document exchanges and fragmented data handling practices. These issues led to redundant data entry across disconnected systems and reduced process visibility, ultimately impairing scalability.

To resolve these challenges, the project adopted a process-centric approach grounded in business process modeling. Existing workflows were formally mapped using BPMN 2.0, enabling a structured diagnosis of coordination breakdowns and compliance risks. A set of critical inefficiencies was then prioritized using an Impact–Effort Matrix, serving as a roadmap for targeted technical interventions.

From this foundation, a modular and scalable digital platform — *X-App* — was designed and implemented. The platform architecture leverages Flutter for cross-platform frontend development and Firebase for backend orchestration, integrating services such as Authentication, Firestore,

Cloud Functions, Storage and Messaging. Seamless synchronization with Salesforce CRM was achieved through a secure dual-authentication model, enabling real-time access to commercial data.

The system's core features were tightly aligned with the process problems identified during the diagnostic phase. These include: structured service submissions, opportunity and proposal management, real-time chat, push notifications, document upload with traceability, and reseller dashboards reflecting CRM-linked process statuses.

The resulting platform reflects a layered architectural logic, enforcing clear separation between data, business logic and presentation. Its modular design ensures maintainability while enabling future extensibility — positioning Company X for continued digital growth and the adoption of intelligent, process-aware workflows.

5.2 Contributions and Impact

X-App delivered concrete value across both organizational and operational dimensions. By replacing fragmented and informal workflows with a centralized and traceable system, the platform significantly reduced administrative burden, improved data quality and shortened process cycles. The structured intake of service requests minimized rework stemming from incomplete documentation, while integrated communication tools eliminated the need for external messaging platforms.

The system also established a single operational interface for both resellers and back-office personnel. This alignment improved coordination and role clarity, enabling real-time visibility into case statuses and enhancing stakeholder accountability. Through automated synchronization with Salesforce, the platform eliminated the most redundant aspects of data handling, namely the manual transcription of client and proposal data across disconnected systems.

Importantly, the architecture created a foundation for future scalability. Built-in support for mobile and desktop access ensures usability across user profiles, while modular backend services and cloud scalability support increased process volumes without proportional increases in headcount.

Nevertheless, it is important to note that the impact of the platform could not be quantitatively assessed during the time frame of this dissertation. At the time of writing, the application had not yet been fully deployed to the reseller network, particularly on mobile devices, where most reseller activity is expected to occur. Consequently, it was not possible to gather usage data or analyze key performance indicators such as submission volume, rejection rates, turnaround times or proposal acceptance rates.

To conduct a robust evaluation in future stages, it will be necessary to track how the system affects core operational outcomes. These include reductions in document rework, acceleration of process cycles, the proportion of CRM records created automatically rather than manually and actual reseller engagement with platform features. Additionally, capturing user feedback on interface usability and perceived transparency will be critical to understand how the platform influences day-to-day reseller behavior and process adherence.

While the present work demonstrates clear architectural improvements and strong alignment with stakeholder needs, its ultimate effectiveness should be confirmed through post-deployment measurement. A structured monitoring effort will not only validate the platform's process alignment but also inform iterative improvements and optimization over time.

5.3 Limitations

While the platform successfully addressed several high-priority inefficiencies, certain limitations remain that constrain full automation and scalability. Document validation and pricing logic continue to depend on human decision-making, limiting the system's autonomy during critical stages of the workflow. The current implementation also lacks optical character recognition (OCR), meaning that invoice data must still be interpreted manually by back-office personnel during the proposal preparation phase.

In addition, the platform's reliance on external systems, particularly Salesforce and third-party supplier portals, introduces integration constraints that prevent true end-to-end automation. For instance, the application cannot currently monitor the real-time status of contract activations once the process is handed off to an external energy provider, which limits downstream process visibility and automation potential.

On the organizational front, adoption remains an ongoing challenge. Transitioning users from informal habits to structured workflows requires not only a user-friendly interface but also sustained onboarding efforts, training sessions and behavioral reinforcement, especially for field agents accustomed to flexible and unstructured communication tools. Without proactive change management, even well-designed systems risk under utilization or resistance during early deployment phases.

5.4 Future Work

Building upon the solid process and technical foundation established in this project, several strategic extensions are possible. These range from transformative initiatives, such as replacing human resellers with fully autonomous agents, to more incremental enhancements involving automation, integration and data-driven decision support. The following subsections outline the most promising directions.

Autonomous AI Reseller Agent

One of the most ambitious and potentially transformative extensions of the platform lies in replacing the human reseller role with a fully autonomous AI agent. Rather than merely assisting resellers, this agent would emulate their entire workflow, from lead generation to contract submission, operating as a virtual commercial representative of the company.

Conceptually, this agent would be capable of initiating contact with prospective clients, collecting and validating necessary documentation, submitting service proposals and monitoring each

opportunity's lifecycle through to contract activation. It would interact with existing systems through pre-defined APIs and operate asynchronously, managing multiple client processes in parallel.

Strategically, the agent offers significant advantages. It would reduce the dependency on human resellers, mitigate onboarding and training costs, and standardize the quality and consistency of client interactions. Moreover, it could operate 24/7, scale elastically with demand and provide comprehensive audit trails.

From a business perspective, deploying an AI reseller could unlock new market segments, enable micro-targeted client outreach campaigns and extend the company's operational footprint without increasing headcount. It would also allow for greater experimentation in commercial strategy, as the agent's behavior could be dynamically adjusted based on conversion data, seasonal trends or product performance.

While technically feasible, the implementation of such an agent involves challenges. It requires robust orchestration of multiple capabilities, such as lead qualification, document handling, client interaction and CRM updates, coordinated over long-running processes. A modular design is essential, with frameworks like LangGraph recently proving particularly well-suited for this task due to their support for persistent and stateful workflows. Nevertheless, the immediate priority should be to define clear agent boundaries, ensure compliance and transparency, and gradually introduce autonomous behaviors through carefully scoped pilot programs.

This AI reseller concept moves beyond incremental automation. It represents a structural shift in how the company can operate commercially, leveraging intelligent systems not just to support humans, but to assume core business functions directly.

Incremental Enhancements and Intelligent Automation

In parallel with long-term automation goals, several other avenues for improvement can enhance the platform's operational intelligence, efficiency and integration maturity.

A key priority is the automation of pricing decisions. Currently, the trade-off between client savings and reseller commissions is handled through ad hoc human judgment. Future iterations could incorporate decision support models or rule-based agents that analyze historical deal data to recommend margin-optimal proposals. These agents could simulate different pricing strategies and automatically detect anomalies or suboptimal commissions.

Document processing is another area with significant potential for automation. By integrating OCR-based invoice parsing tools, the platform could eliminate the need for manual data extraction and validation. This would reduce rework, shorten processing times and ensure greater consistency in proposal preparation. In high-volume environments, such automation would be critical for maintaining process velocity without increasing administrative workload.

Finally, intelligent performance monitoring tools could support internal management functions. Predictive dashboards and AI-driven analytics could highlight at-risk opportunities, flag under-performing workflows and surface strategic interventions to optimize team efficiency. These

tools would support data-informed decision-making across both operational and commercial dimensions.

Together, these incremental enhancements and the broader vision of an autonomous AI reseller outline a clear future trajectory: a transition from digital support toward intelligent orchestration of commercial workflows, fully aligned with the company's growth and scalability objectives.

5.5 Final Remarks

This dissertation has demonstrated how structured process modeling, when combined with modern cloud architecture and integration practices, can transform fragmented operational environments into coherent and scalable platforms. The results delivered by *X-App* go beyond basic digitalization, they establish a foundation for intelligent, auditable and eventually autonomous workflows in reseller-driven organizations.

Although the platform was developed under tight time and resource constraints, it presents a robust and extensible solution capable of supporting both current operations and future innovations. By aligning structured process logic with technological agility, the project exemplifies how industrial engineering and software development can converge to drive sustainable growth and lay the groundwork for intelligent system augmentation.

References

- Almotairi, M. (2010). CRM architecture and implementation approaches. *International Journal of Computer Applications*, 4(7):30–35.
- Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., Mitchell, N., Muthusamy, V., Rabbah, R., Slominski, M., and Suter, P. (2017). Serverless computing: Current trends and open problems. *Research Advances in Cloud Computing*, pages 1–20.
- Berezin, S. (2025). Architecting scalable AI-CRM systems: Design patterns, infrastructure, and performance optimization. *International Journal of Engineering and Computer Science*, 14(06).
- Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., et al. (2022). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- Buttle, F. (2009). *Customer Relationship Management: Concepts and Technologies*. Elsevier.
- Choudhury, A. and Madan, A. (2021). Effective CRM strategies for SMEs in the age of APIs. *International Journal of Information Management*, 58:102314.
- Davenport, T. H. (1993). *Process Innovation: Reengineering Work through Information Technology*. Harvard Business School Press.
- Davenport, T. H. and Ronanki, R. (2018). Artificial intelligence for the real world. *Harvard Business Review*, 96(1):108–116.
- Deveau, R., Griffin, S. J., and Reis, S. (2023). AI-powered marketing and sales reach new heights with generative AI. *McKinsey & Company*.
- Dumas, M., Rosa, M. L., Mendling, J., and Reijers, H. A. (2018). *Fundamentals of Business Process Management*. Springer, 2 edition.
- Dwivedi, Y. K., Hughes, L., Ismagilova, E., Aarts, K., Coombs, C., Crick, T., Duan, Y., Dwivedi, A., Edwards, J. S., Eirug, A., et al. (2023). So what if chatgpt wrote it? multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. *International Journal of Information Management*, 71:102642.
- Eismann, S., Scheuner, J., Wittern, E., Leitner, P., Bezemer, C.-P., Grohmann, J., and Schulte, S. (2021). Serverless applications: Why, when, and how? *IEEE Software*, 38(1):32–39.
- Eivy, A. (2017). Be wary of the economics of “serverless” cloud computing. *IEEE Cloud Computing*, 4(2):6–12.
- Faria, J. (2024). Process modeling fundamentals. Lecture slides, Faculdade de Engenharia da Universidade do Porto.

- Gabryelczyk, R., Sipior, J. C., and Biernikowicz, A. (2023). Motivations to adopt BPM in view of digital transformation: A review of use cases. *Information Systems Management*, 40(1):1–10.
- George, M. L. (2002). *Lean Six Sigma: Combining Six Sigma Quality with Lean Speed*. McGraw-Hill, New York.
- Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5):1–42.
- Hammer, M. and Champy, J. (1993). *Reengineering the Corporation: A Manifesto for Business Revolution*. HarperBusiness.
- He, S. and Erturk, E. (2018). Study on a high-integrated cloud-based customer relationship management system. *arXiv preprint arXiv:1812.09005*.
- Hildebrand, C. and Bergner, A. (2019). Using chatbots to boost sales. *NIM Marketing Intelligence Review*, 11(2):36–41.
- Imai, M. (1986). *Kaizen: The Key to Japan's Competitive Success*. McGraw-Hill, New York.
- International Organization for Standardization (2013). ISO/IEC 19510:2013 - Business Process Model and Notation (BPMN). Available at: <https://www.iso.org/standard/62652.html>.
- Jonas, E., Schleier-Smith, J., Sreekanti, V., Tsai, C.-C., Khandelwal, A., Pu, Q., Shankar, V., Carreira, J., Krauth, K., Gonzalez, J. E., and Stoica, I. (2019). Cloud programming simplified: A Berkeley view on serverless computing. *arXiv preprint arXiv:1902.03383*.
- Kumar, A. and Rao, S. (2021). API-driven salesforce integration security and performance. *International Research Journal of Modernization in Engineering Technology and Science*, 07(04):7821–7825.
- Käss, S., Strahringer, S., and Westner, M. (2022). Drivers and inhibitors of low code development platform adoption. In *2022 IEEE 24th Conference on Business Informatics (CBI)*, pages 196–205.
- Lazzari, L. and Farias, K. (2022). Os efeitos da arquitetura dirigida a eventos na modularidade de software: Um estudo exploratório. *arXiv preprint arXiv:2110.14699*.
- Lenarduzzi, V. and Fucci, D. (2019). Towards a holistic definition of requirements debt. *arXiv preprint arXiv:1907.10887*.
- Malone, T. W. and Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119.
- Mendling, J., Weber, I., van der Aalst, W., vom Brocke, J., Cabanillas, C., Daniel, F., Debois, S., Ciccio, C. D., Dumas, M., Dustdar, S., et al. (2018). Blockchains for business process management - challenges and opportunities. *ACM Transactions on Management Information Systems*, 9(1).
- Nguyen, H. M. and Whelan, E. (2012). Understanding customer relationship management (CRM) technology adoption. In *Proceedings of the European, Mediterranean & Middle Eastern Conference on Information Systems (EMCIS)*, pages 1–13. EMCIS.

- Nordli, E. T., Haugeland, S. G., Nguyen, P. H., Song, H., and Chauvel, F. (2023). Migrating monoliths to cloud-native microservices for customizable SaaS. *Information and Software Technology*, 160:107230.
- Ojha, R. (2024). From algorithms to conversations: The influence of natural language processing on chatbot innovation. *International Journal of Contemporary Research in Multidisciplinary*, 3(6):21–27.
- Ransbotham, S., Kiron, D., LaFountain, B., and Khodabandeh, S. (2020). Expanding AI’s impact with organizational learning. *MIT Sloan Management Review*, 62(1).
- Ruivo, P., Mestre, A., Johansson, B., and Oliveira, T. (2014). Defining the ERP and CRM integrative value. *Procedia Technology*, 16:704–709.
- Shingo, S. (1986). *Zero Quality Control: Source Inspection and the Poka-Yoke System*. Productivity Press.
- Shmueli, G., Bruce, P. C., and Patel, N. R. (2016). *Data Mining for Business Analytics: Concepts, Techniques, and Applications in R*. Wiley, first edition.
- Sivarajah, U., Kamal, M. M., Irani, Z., and Weerakkody, V. (2017). Critical analysis of big data challenges and analytical methods. *Journal of Business Research*, 70:263–286.
- Teixeira, A. R., Ferreira, J. V., and Ramos, A. L. (2024). Optimization of business processes through BPM methodology: A case study on data analysis and performance improvement. *Information*, 15(2):499–517.
- vom Brocke, J., Zelt, S., and Schmiedel, T. (2016). On the role of context in business process management. *International Journal of Information Management*, 36(3):486–495.
- Wiggins, A. (2017). The Twelve-Factor App. <https://12factor.net/>.
- Wiśniewski, P., Kluza, K., Kucharska, E., and Ligęza, A. (2019). Spreadsheets as interoperability solution for business process representation. *Applied Sciences*, 9(4):748.