Resident Portal

Technology and Architecture Requirements

1 CONTENTS

1 Contents 2
2 Figures 3
3 Introduction 5
4 Scope 5
5 Technology Market Scan 6
5.1 Assumptions 6
5.2 Our approach 6
5.3 Technology Considerations 6
5.4 Digital Experience Platforms 8
5.5 Content Management System 9
5.6 Resident User Web Front-end Technology 10
5.7 Campaign Engine Technology 10
5.8 Mobile Front-End Technology 12
5.9 Chat Front-end Technology 13
5.10 Conversational API 14
5.11 Next Best Action (NBA) Engine Technology 15
5.12 Feature Management 16
5.13 Payments 17
6 Technology Recommendations 17
6.1 User Interfaces 17
6.2 Digital Experience Platforms 18
6.3 Content Management System 18
6.4 Big Foot Bot / AI 19
6.5 Feature Management 19
6.6 Data Lake 20
6.7 Payments ..... **Error! Bookmark not defined.**
6.8 Recommendations Summary 20
7 Platform Architecture 21
7.1 Platform Architecture - Amazon Centric AI Approach 21
7.2 Platform Architecture - Microsoft Centric AI Approach 22
7.3 Platform Architecture – Color Key 22
8 Key Components 23
8.1 The User Interfaces 23

8.2 Big Foot Bot / AI 24
9 Alignment to Journeys and Key Capabilities 28
9.1 Walking Andrew's Journey with the Tech - Part 1 28
9.2 Walking Andrew's Journey with the Tech - Part 2 29
10 Back-Office Recommendations 29
10.1 Back-Office Tools 29
10.2 Integrated Back-Office Capabilities 29
11 Agency Integration Strategy & Architecture 30
11.1 Easy Wins 30
11.2 Service Integration Levels 32
11.3 Service Integration Tiers Journey Walkthrough – Reserving a campsite 35
11.4 Agency Capability Cohort Strategy 36
11.5 Technology Enablers 37
11.6 Shared Services - Resident Portal Owned Services 37
11.7 Agency Owned Services 39
11.8 Feature Management 39
11.9 Agency Kit 39
12 Data Strategy, Architecture, Privacy, & Security 40
12.1 Data Sharing & Data Security 40
12.2 Centralizing the "Core" Resident Profile 43
12.3 Separating Concerns: Account, Profile, and Verification 43
12.4 Data Architecture 44
12.5 User-Centric Data Management Experiences 45
12.6 Adjust the Data Strategy Over Time 47
13 Cost Projection 47
14 Appendix 48
14.1 References 48
14.2 Appendix A: Citations 48

2 FIGURES

Figure 1: Platform Architecture - Amazon Centric AI Approach 21
Figure 2: Platform Architecture - Microsoft Centric AI Approach 22
Figure 3: Platform Architecture - Color Key 22
Figure 4: Big Foot Bot Architecture 24
Figure 5: Determining which Agency Service 25

Figure 6: Walking Andrew's Journey with the Tech - Part 1 28
Figure 7: Walking Andrew's Journey with the Tech - Part 2 29
Figure 8: MyTN - New Resident Booking a Campsite - 1 of 2 30
Figure 9: MyTN - New Resident Booking a Campsite - 2 of 2 31
Figure 10: TN.gov --> TNStateParks.com - Resident Booking a Campsite 31
Figure 11: Service Integration Tiers 32
Figure 12: Tier 1 – User Experience 36
Figure 13: Agencies Exchanging Data - 1 to Many 41
Figure 14: Agencies Exchanging Data - Not Many to Many 41
Figure 15: Agencies Exchanging Data - Many to Many Via the Resident 42
Figure 16: Disambiguating Account from Profile 44
Figure 17: Opting In (“App Store model”) 46

3 INTRODUCTION

This document addresses the technology requirements for the Resident Portal. It highlights essential features of the technology and outlines any required hardware, software, or infrastructure specifications. Additionally, it identifies and proposes the key strategic and architectural requirements.

4 SCOPE

The capabilities were evaluated at a high level to assess their overall fit, alignment, gap analysis, and risks associated with the chosen technologies. As a result of this analysis, the user interfaces for residents (high priority) and Big Foot Bot (largely unknown) were identified as focal points for this document.

The Agency Integration Strategy & Architecture was recognized as a key deliverable and an important topic for discussion. Additionally, back-office capabilities and strategies emerged as another area for exploration and were included in this document.

This document does not aim to outline a high-level architecture for each capability. Further analysis, design, and architecture will be required for each group as the capabilities are prioritized and selected for enablement, integration, and implementation.

5 TECHNOLOGY MARKET SCAN

The Resident Portal solution will require a collection of integrated solutions to support the vision and full suite of capabilities WaTech intends for the resident experience. This Technology Market Scan investigates and evaluates technology products currently available in the market for consideration as potential components for the WA Resident Portal ecosystem. For the sake of brevity, only the top evaluated contenders are included in this document.

Please refer to the list of considered technologies are included in the Market Scan Matrix: WARP Market Scan - Competitive Capability & Feature - Final - 2024-10-14-.xlsx.

5.1 ASSUMPTIONS

The Technology Market Scan work accounts for the following assumptions:

A. It is unlikely that a single out of the box technology platform will support the full suite of Resident Portal capabilities.
B. This evaluation does not drill down into specific user experiences or flows for every Resident Portal capability; any off-the-shelf solution will always come with compromises regarding end-user experience.
C. There are additional Washington State projects and initiatives that intersect may impact these recommendations; not all of them may be known at this time. Known decisions are:
    a. Okta will be used for identity management and relative to the roadmap timelines will be assumed to be in place
    b. Databricks will be used for data governance, to share data between agencies and the Resident Portal, and to transform raw data to formatted data for consumption.
    c. Amazon Connect will be used as a contact center and entry point for the Resident Portal.
    d. Service Now for internal ticketing / IT Service Management.
D. Some technology platforms currently utilized by WaTech are open to change.
    a. Drupal – Content Management System

5.2 OUR APPROACH

Technologies presented in the following sections of this Market Scan that have a UI or interactive aspect include a summary usability evaluation that identifies key considerations or barriers that technology choice could have on the end portal development and user experience. As with the technology evaluations themselves, this assessment is not at the component or functional flow level.

5.3 TECHNOLOGY CONSIDERATIONS

Here are the things we considered while evaluating the technology:

Citeria	Considerations

Use by other States / Large Companies	• Is the technology used by other states in their solutions? • Is the technology used by large companies? • What is the scale of its use in those applications?
Pros/Cons	• Top level pros/cons known about the technology within the industry.
Alignment to State of Washington Architecture Principles and Nonfunctional Domain Requirements	• Security • Privacy • Interoperability • Accessibility • Reliability & Availability • Performance • Scalability • Maintainability • User experience • Legal
AI tools & integration support	• Does the technology have built-in support for AI integration(s)? • Do the technology's AI tools enable aggregation from external sources? • Can the technology be extended to support embedded AI modules or components? • Can the technology be packaged into a module? • Can the technology's content be made available for AI training data.
Compatibility	• Does the technology leverage the same skill set or technology stack as other technologies? • Does the technology require specialized or proprietary formats? If yes, how does it import or export data? • Does the technology allow for customization? • "codeless" or "low code"? • Robust and well documented APIs?
Industry Standard	• Is non-proprietary support available? • Does the technology have a wide talent pool? • Is the technology open source? If so, how is it maintained? • Is the technology proprietary? If so, what is the stability of the owning company. • Is the technology mature, cutting edge, or bleeding edge?
Functionality	• How "good" is the primary feature(s)? • Are there any key Sub Features?
Infrastructure and hosting	• Is the technology SAAS, self-hosted, or optional? • Continuous integration • Continuous deployment • Compatible with containers
Testability	• Can the technology be quality tested in pre-production environments? • incorporated into build pipelines?
Pricing	• Licensing • Use restrictions • Cost

5.4 Digital Experience Platforms

Technologies that offer shared user experiences across multiple application domains. The technologies considered in this category offer capabilities beyond a typical CMS. The exploration in this category is focused on discovering if an off the shelf solution could fit all (or most) of our requirements.

5.4.1 Liferay

Liferay is an enterprise-grade platform primarily used for building web portals and intranets. It offers a wide range of features to create, manage, and deliver digital experiences for various business needs. It handles a wide range of content, streamlines business workflows, integrates with enterprise services, and offers APIs and SDKs for building custom applications. Customizes the look and feel of sites using themes and layout templates. Extends functionality through custom modules and plugins.

- URL: https://www.liferay.com/

5.4.2 Alfresco Platform

Alfresco is an open-source content management and business process management platform. It handles a wide range of content, streamlines business workflows, integrates with enterprise services, and offers APIs and SDKs for building custom applications.

- URL: https://www.hyland.com/en/products/alfresco-platform

5.4.3 Jahia

Jahia is a digital experience platform (DXP) that integrates content management, personalization, and digital asset management to deliver tailored digital experiences across multiple channels. It combines the capabilities of a traditional CMS with advanced personalization and data management features, making it suitable for enterprises looking to enhance their digital presence and engage users more effectively. It handles a wide range of content, integrates with enterprise services, included a campaign engine, and offers APIs and SDKs for building custom applications.

- URL: https://www.jahia.com/

5.4.4 Service Now

ServiceNow is a cloud-based platform that provides a wide range of enterprise services, focusing on automating and streamlining workflows, service management, and business operations. It is widely used by organizations for IT Service Management (ITSM) and Customer Service Management (CSM), but its capabilities extend to other domains.

- URL: https://www.servicenow.com/now-platform.html

5.4.5 Portal Application Evaluation

- Liferay's Resident facing UI: Modern UI layouts and a large variety of out of the box components, from forms to charts. React, Angular, and Vue can be utilized to build custom widgets in the Liferay toolkit. Back of the house workflows for building forms and pages leverage similar UI components as Resident facing. Unclear how easily customizable the UI components are beyond branding & color.
- Alfresco's Resident facing UI: The default UI and components kit contains the expected basics in a mundane style. The Theme can be updated in a limited way for branding and color. Customization beyond this does not seem to be supported. The documentation does not hint at any advanced (e.g., charts, wizards, or highly animated interaction) capabilities out of the box. Supports through JavaScript the addition of custom build components. Back of the house workflows are document management / CMS centric and are not robust for managing Resident facing UI components or flows.

- Jahia's Resident facing UI: Would be build using React.js. UI would be largely custom-built. However, Jahai has integrations with pre-built "Moonstone" library in Storybook (UI library code management). Storybook has Figma plug-ins for linking code to design files. Back of house creation of screen layouts needs sufficient knowledge of both design tools and coding.
- Service Now's Resident facing UI: The default UI and components kit contains the expected basics in a mundane style. Customization is limited. React, Angular, and Vue can be utilized to build custom widgets in the Service Now toolkit. Service Now is able to run in a headless mode, providing access to resources via a rest api.

5.5 Content Management System

A Content Management System (CMS) is a software platform that allows users to create, manage, and modify digital content without needing specialized technical knowledge. Traditional CMS platforms typically combine content creation with the frontend display, meaning they handle both the management of content and its presentation on a website. However, with the rise of omni-channel digital experiences, headless CMS systems have gained popularity. A headless CMS decouples the back-end content management from the frontend display, providing content via APIs that can be served across various channels, such as websites, mobile apps, and chat bots. This flexibility allows for greater control over how and where content is delivered. Platforms like Contentful and Strapi specialize in headless content management, while Drupal also offers a headless mode and content APIs, making it a versatile choice for modern, omni-channel content distribution.

5.5.1 Drupal

Drupal is an open-source, traditional content management system (CMS) designed for building and managing web applications. It excels in flexibility, scalability, and advanced content management.

- URL: https://www.drupal.org/

5.5.2 Contentful

Contentful is a cloud-based headless content management system (CMS) designed for creating, managing, and delivering digital content across multiple channels. It provides a flexible and scalable solution for content management with a focus on API-driven delivery.

- URL: https://www.contentful.com/

5.5.3 Strapi

Strapi is an open-source headless content management system (CMS) designed for creating, managing, and delivering digital content. It provides a flexible and scalable solution for content management with a focus on API-driven delivery.

- URL: https://strapi.io/

5.5.4 CMS Evaluation

- Drupal is currently in use by WaTech; no evaluation needed.
- Contentful's Resident facing UI: Contentful has a Studio that contains its out of the box components, which can be extended or added to via the Contentful SDK. Components need modification to meet WCAG contrast at minimum; out of the box modifications option are very limited and include only light styling.Back of house: though UI is JavaScript, it is built for Next.js. Builder flows are intended for developers.

- Strapi's Resident facing UI: Strapi 5 has a Design System set up in the Atomic Design paradigm. It is intended to be used with React.js. All UI would be custom built with a jump off point of existing code for the UI building blocks.Back of house: has an "admin panel" for layout and content management, but it appears rather limited and most of the flows are intended for developers

5.6 Resident User Web Front-end Technology

5.6.1 React

React is an open-source JavaScript library developed by Facebook for building user interfaces, particularly single-page applications (SPAs). Its strong community support, extensive ecosystem, and ability to facilitate cross-platform development with React Native make it a popular choice for building modern, high-performance user interfaces.

- URL: https://react.dev/

5.6.2 Angular

Angular is an open-source framework created by Google for crafting dynamic web applications. It offers a powerful platform for building single-page applications (SPAs), emphasizing scalability and maintainability. It has a strong community backing.

- URL: https://angular.dev/

5.6.3 Vue.js

Vue.js is a progressive, open-source JavaScript framework used for building user interfaces and single-page applications (SPAs). It is designed to be adaptable, allowing developers to integrate it into projects incrementally.

- URL: https://vuejs.org/

5.6.4 Web Front-end Evaluation

- React is extremely malleable and robust for UI. There are many existing component libraries and/or Atomic design libraries that can be leveraged. React.js is cross-platform for Android, iOS, and web. Best for dynamic web apps. Light weight code performs well on mobile.
- Angular uses component-based TypeScript paradigm. There is also an open-source JavaScript front-end framework developed and managed by Google's Angular team that has a solid number of UI libraries, but not as rich a set of options as React. It is best for Single-page applications with intricate workflows. Angular is known for poor performance on mobile.
- Vue blends HTML and JS, with a large official component library, but a much smaller field for additional component libraries. Vue.js does not have an official solution for mobile app development. Best for simple and fast applications without the need for complexity.

5.7 Campaign Engine Technology

A campaign engine is a software tool used for multi-channel communication, involving email, social media, mobile, web, and other channels. The engine helps automate and streamline the process.

5.7.1 Salesforce Marketing Cloud

Salesforce Marketing Cloud is a comprehensive marketing automation and customer engagement platform designed to help businesses create personalized marketing campaigns across multiple channels, such as email,

SMS, social media, advertising, and push notifications. Salesforce Marketing Cloud is available as both a stand-alone product and a part of the Salesforce suite.

- URL: https://www.salesforce.com/marketing/

5.7.2 Amazon Pinpoint

Amazon Pinpoint is a cloud communications platform that enables businesses and organizations to engage with their users through multiple communication channels, including email, SMS, push notifications, and voice messages. It is designed to deliver personalized, targeted messages at the right time, with analytics and tracking to monitor engagement and effectiveness. With its strong integration into the Amazon AWS ecosystem, Amazon Pinpoint is a strong choice for businesses looking to provide seamless and secure communication experiences to their users.

- URL: https://aws.amazon.com/pinpoint/

5.7.3 Microsoft Dynamics 365 Marketing

Microsoft Dynamics 365 Marketing, a part of the Microsoft Dynamics 365 suite, offers a comprehensive set of tools for creating, executing, and managing personalized marketing campaigns across multiple channels including email, SMS, events, and social media. It can automate customer journeys and leverage deep insights from customer data. With its seamless integration into the Microsoft ecosystem, it is ideal for organizations that require an enterprise-level marketing automation platform.

- URL: https://www.microsoft.com/en-us/dynamics-365#overview

5.7.4 Microsoft Azure Communication Services

Microsoft Azure Communication Services (ACS) is a cloud communications platform for adding communication capabilities to your applications. ACS offers a wide range of tools to enable voice, video, chat, and SMS communication across multiple channels. It's lighter weight than Dynamics 365 Marketing, lacking campaign management. With its strong integration into the Microsoft Azure ecosystem, ACS is a strong choice for businesses looking to provide seamless and secure communication experiences to their users, without the overhead of a full campaign engine. ACS is best suited for real-time communication.

- URL: https://azure.microsoft.com/en-us/products/communication-services

5.7.5 Twilio

Twilio is a cloud communications platform that enables businesses to integrate communication functionalities such as voice, SMS, video, email, and chat into their applications. It provides APIs that allow developers to build and customize communication workflows for various use cases, including customer support, marketing campaigns, notifications, and authentication services. Twilio is known for its scalability, reliability, and ease of integration, allowing companies to build communication solutions without needing to manage telecommunications infrastructure themselves.

- URL: https://www.twilio.com/en-us

5.7.6 Campaign Engine Evaluation

- While Salesforce Marketing Cloud is a powerful and comprehensive marketing automation platform, its high costs and complexity, makes it prohibitive. It requires significant setup, integration, and customization efforts, often needing specialized teams or developers, which can delay the return on investment. For companies with simpler marketing needs, more affordable and user-friendly alternatives offer a better fit without the steep learning curve.

- Twilio is versatile for custom communication solutions across multiple channels. It has great API support. It has higher upfront setup effort and lacks campaign management. Twilio offers a suite of products, but they are not bundled into a single "suite" package. Instead, Twilio provides individual APIs and services that businesses can integrate based on their needs. The products of Twilio's Communications Platform as a Service (CPaaS) can be used together to build a comprehensive, omnichannel communication solution.
- Amazon Pinpoint has strong campaign management, personalization, and segmentation for marketing and notifications.
- Azure Communication Services has comprehensive real-time communication (voice, video, chat). It integrates well with Microsoft Teams. It lacks email, push notifications, and campaign management.
- Microsoft Dynamics 365 Marketing offers a full suite of tools for managing complex omnichannel campaigns, personalized customer journeys, and lead nurturing, all integrated with Dynamics 365 CRM. It's an excellent fit for businesses with advanced marketing needs and teams looking for AI-driven insights to optimize campaigns. Pricing is a major concern, as this suite may overkill the need. More cost analysis is needed to fully consider this technology.

5.8 Mobile Front-End Technology

5.8.1 React

Using React for a mobile-friendly website involves leveraging its component-based architecture and responsive design principles to create adaptable, reusable components for various screen sizes. Implementing a mobile-first approach ensures that the core features work seamlessly on smaller screens before scaling up for larger devices. By using responsive libraries like Material-UI or React Bootstrap and optimizing performance through techniques like lazy loading and code splitting, developers can enhance load times and improve the user experience.

- URL: https://react.dev/

5.8.2 React Native

React Native is an open-source framework developed by Facebook for building cross-platform mobile applications using JavaScript and React. It allows developers to write code once and deploy it on both iOS and Android platforms, leveraging native components for performance and user experience.

- URL: https://reactnative.dev/

5.8.3 .NET MAUI

.NET MAUI (formerly Xamarin) is a cross-platform framework developed by Microsoft for building native applications with a single codebase, leveraging .NET's extensive ecosystem and modern development tools. Its strengths include unified APIs, native performance, and a rich set of UI components. It is integrated with Microsoft's ecosystem and supported by strong tooling and community resources.

- URL: https://dotnet.microsoft.com/en-us/apps/maui

5.8.4 Flutter

Flutter is an open-source UI framework developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. It uses the Dart programming language and provides a rich set of customizable widgets and tools for creating high-performance, visually attractive applications.

- URL: https://flutter.dev/

5.8.5 Mobile Front-End Evaluation

- React is a strong choice for building mobile-friendly websites due to its component-based structure, responsive design support, and performance optimizations. It works well for creating responsive web apps and Progressive Web Apps (PWAs), offering native-like experiences in mobile browsers. React allows access across both mobile and desktop browsers without needing an app, ensuring broader reach and a single codebase for web and mobile-friendly versions.
- React Native uses as its foundation React with Native UI. The tech allows the freedom to stay with the default platform's styling or adapt it to the desired UI. It is a strong choice if the team is already used to JS and React. Some consider it to have inefficient performance if the UI is complex.
- .NET MAUI is based on C# for creating native mobile apps for Android, iOS, macOS, and Windows from a single codebase. It is supported by a .Net community created components. The framework is known for unhelpful error messages, warnings, and problems with animations and mouseovers.
- Flutter uses Dart as its code base. It has its own customizable widgets to control look and behavior across every platform. It also comes with a set of out-of-the-box widgets based on the Material (Android) and Cupertino (iOS) designs. Best for teams already familiar with Dart and for ultra lightweight applications.

5.9 Chat Front-end Technology

The user interface component of an application that users interact with directly (in the form of chat) in real time or near real time. This may be embedded in another aspect of application.

5.9.1 Intercom

Intercom is a customer communication platform that excels in multi-channel messaging, advanced targeting, and automation. It integrates chat, email, and other communication tools to provide a unified experience for interacting with customers.

- URL: https://www.intercom.com/

5.9.2 Zendesk Chat

Zendesk Chat is a live chat software solution that excels in real-time communication, automation, and integration. It is designed to facilitate real-time communication between businesses and their customers Zendesk Chat provides all core chat functionalities such as real-time messaging, chatbots, automated triggers, and reporting. Zendesk Chat can function effectively as a standalone solution without requiring the full range of Zendesk's customer service tools.

- URL: https://www.zendesk.com/

5.9.3 Drift

Drift is a conversational marketing and sales platform that focuses on helping businesses engage with potential customers through real-time chat and automated messaging. It aims to streamline the lead generation and sales process by facilitating immediate interactions and personalized communication.

- URL: https://www.drift.com/

5.9.4 Chat Front-end Evaluation

- Intercom has less dashboard features than others, the chat itself is considered to have strong knowledge-base integrations, and have a clean modern design with a customizable interface (default is free floating chat bubble). Back of the house set-up is modern, but the builder flow is not intuitive

- While Zendesk Chat is considered to have a feature-rich integration capability, the chat itself is known to have an outdated user interface. Little to no customization. It is still one with all the necessary features.
- Drift has a customizable and familiar "messenger box" UI. It handles video. It does not handle multi-lingual support. The back of the house reporting UI is also modern and customizable.
- Amazon Connect, currently selected by WaTech, will continue to be utilized for the chat front-end as part of the contact center.

5.10 Conversational API

In a traditional call center, agents would sit in cubicles with landline phones and computers to look up information from various siloed databases. Modern organizations often use contact center platforms instead. The entry point into the contact center will be handled by a combination of self-service and Artificial Intelligence (AI) Bots. The list of technologies below are candidates for providing the AI Bots.

5.10.1 Nuance Mix (acquired by Microsoft)

Nuance Mix is a comprehensive conversational AI and voice platform that excels in advanced natural language understanding, speech recognition, and speech synthesis. Its strengths include omnichannel support, customization and flexibility, seamless integration with business systems, scalability, performance, and robust analytics.

- URL: https://www.nuance.com/index.html

5.10.2 Microsoft Azure Bot Service with Speech

Microsoft Azure Bot Service with Speech is a cloud-based platform that integrates speech capabilities into conversational AI applications, enabling bots to handle voice interactions alongside text. It leverages Azure Speech Services to provide advanced speech recognition and synthesis features. Its strengths include accurate speech-to-text and text-to-speech conversion, real-time translation, voice biometrics for security, seamless integration with the Azure ecosystem, and flexibility in customization.

- URL: https://azure.microsoft.com/en-us/products/ai-services/ai-bot-service

5.10.3 Dialogflow CX (by Google Cloud)

Dialogflow CX is a conversational AI platform developed by Google Cloud for building and managing sophisticated chatbots and virtual agents. Its strengths include a visual flow builder, robust state management, complex conversational flows, high accuracy in intent recognition, advanced natural language understanding, and multi-channel interactions.

- URL: https://cloud.google.com/dialogflow

5.10.4 Amazon Lex

Amazon Lex is an AWS service that enables developers to build chatbots and voice assistants using natural language understanding (NLU) and automatic speech recognition (ASR). It allows for conversational interfaces that can handle multi-step interactions, making it ideal for customer service bots, virtual assistants, and IVR systems. Lex integrates easily with other AWS services and can be deployed across web, mobile, and messaging platforms.

- URL: https://aws.amazon.com/lex/

5.10.5 Conversational API Evaluation

- Nuance Mix is a highly specialized conversational AI platform known for its advanced speech recognition and domain-specific natural language understanding (NLU), ideal for industries like healthcare and finance. It provides powerful voice interaction capabilities and customizable language models, making it best suited for complex voice-driven systems. However, it can be more complex and expensive compared to other platforms, targeting enterprise-level solutions.
- Azure Bot Service, integrated with Azure Cognitive Services, provides powerful voice and text-based bots for enterprise applications. It offers seamless integration with the Azure ecosystem and supports scalable conversational AI solutions with high-level security and compliance. It's well-suited for businesses already using Azure, though it may require more technical expertise to set up.
- Google Dialogflow is a user-friendly conversational AI platform known for its powerful NLP engine and broad cross-platform deployment, supporting chatbots and voice assistants across messaging platforms and devices. It offers multilingual support and easy integration with Google services, making it a great fit for customer support and voice-enabled applications. Its simplicity makes it accessible, but advanced customization may be more limited compared to enterprise solutions.
- Amazon Lex is an AWS service for building voice and text chatbots, using the same technology as Amazon Alexa for ASR and NLU. It integrates deeply with the AWS ecosystem, making it a cost-effective choice for customer service and transactional bots within AWS environments.

5.11 Next Best Action (NBA) Engine Technology

An advanced decision-making system that recommends the most appropriate action to take. A Next Best Action (NBA) engine is a system that uses machine learning (ML), artificial intelligence (AI), and advanced analytics to recommend the most relevant or optimal action for an individual user at a specific point in time. The purpose of an NBA engine is to enable personalized experiences and guide interactions in a way that maximizes engagement and satisfaction.

5.11.1 Microsoft Personalizer

Microsoft Personalizer is a powerful AI-driven recommendation engine designed to provide real-time, personalized, context-aware suggestions and content recommendations. It leverages reinforcement learning to optimize decision-making in real time, ensuring that it continually improves as it gathers more interaction data, making it highly effective for optimizing user experiences and driving engagement. By integrating easily with the Azure ecosystem, Personalizer offers a flexible, scalable, and efficient way to deliver context-aware personalization in dynamic environments.

- URL: https://azure.microsoft.com/en-us/products/ai-services/ai-personalizer

5.11.2 Amazon Personalize

Amazon Personalize is an AI-powered personalization service from Amazon Web Services (AWS) that allows businesses to deliver real-time, personalized recommendations to their users. By simplifying the complexities of machine learning and leveraging AWS infrastructure, it enables developers to create custom recommendation systems without the need for extensive machine learning expertise.

- URL: https://aws.amazon.com/personalize/

5.11.3 Pega Customer Decision Hub

Pega Customer Decision Hub is a powerful decisioning and analytics platform that excels in real-time, AI-driven decision-making and personalization. Its strengths include advanced analytics, a unified customer profile, omnichannel engagement, dynamic personalization, and customizable decision strategies.

- URL: https://www.pega.com/

5.11.4 Next Best Action Evaluation

- Microsoft Personalizer is an AI-driven service that delivers real-time, personalized recommendations using reinforcement learning. It adapts to user behavior over time and integrates seamlessly with Azure, making it ideal for real-time decision-making in apps.
- Amazon Personalize is a machine learning service for personalized product recommendations and targeted marketing, using the same technology as Amazon.com. Highly integrated with AWS, it's easy to use and suitable for businesses wanting personalized experiences without deep ML expertise.
- Pega Customer Decision Hub is a real-time AI decisioning engine that delivers next-best-action recommendations across all customer touchpoints. It excels in omnichannel marketing and complex customer journeys, making it ideal for enterprises needing tailored, multi-channel decisioning.

5.12 Feature Management

5.12.1 LaunchDarkly

LaunchDarkly is a feature management platform that allows development teams to manage and control feature rollouts using feature flags. It enables businesses to decouple feature deployment from releases, meaning you can deploy code to production but choose when and to whom to make features available. This allows for safe, gradual rollouts, targeted feature releases, A/B testing, and the ability to quickly roll back problematic features without redeploying code, all while improving deployment agility and reducing risk.

- URL: https://launchdarkly.com/

5.12.2 Microsoft.FeatureManagement Namespace

The Microsoft.FeatureManagement namespace is a .NET library that allows developers to implement feature flags within their applications to control which features are enabled or disabled at runtime. It supports conditional logic and feature filters for targeted rollouts, such as user-specific or percentage-based feature activation. This library integrates with Azure App Configuration but can also be used independently for managing feature flags locally in .NET applications.

- URL: https://learn.microsoft.com/en-us/dotnet/api/microsoft.featuremanagement?view=azure-dotnet

5.12.3 Optimizely Feature Experimentation

Optimizely Feature Experimentation is a platform that allows teams to manage feature flags and conduct A/B testing and multivariate experiments on new features in real time. It enables developers to toggle features on or off for specific user segments, run experiments to determine which version of a feature performs better, and make data-driven decisions without needing to redeploy code. With granular targeting, audience segmentation, and deep performance analytics, Optimizely helps businesses optimize user experiences and measure the impact of new features before fully rolling them out to all users.

- URL: https://www.optimizely.com/products/feature-experimentation/

5.12.4 Feature Management Evaluation

- LaunchDarkly is ideal for large-scale teams needing enterprise-grade feature management with real-time control, scalability, and deep integration options, making it suitable for complex deployment environments.
- Microsoft.FeatureManagement is well-suited for .NET developers, offering flexibility to manage feature flags locally or through Azure, with a strong focus on conditional logic and Azure integration, though it can

also be adapted for non-Azure environments, by leveraging app settings with AWS Parameter Store or AWS Secrets Manager to manage the flags centrally within AWS.

- Optimizely Feature Experimentation is ideal for data-driven teams, offering advanced analytics and segmentation for precise feature control and real-time performance measurement. It's powerful for optimizing user experiences but may be more complex and costly for those not focused on experimentation.

5.13 Payments

A review of six popular payment processing services found that they are all relatively comparable in functionality and price; in most cases Enterprise pricing tiers are available which could be a deciding factor. While Tyler Technologies has a history in serving government, we're leaning towards a more light-weight solution in Square based on market trends and the fact that WA state has had a rocky relationship with Tyler Technologies in the past.

Provider	Relative Transaction Fee	Flat Transaction Fee
Stripe	0.03%	$0.30
PayPal	0.03%	$0.30
Square	0.03%	$0.30
Tyler Technologies	0.03%	$0.30
Authorize.Net	0.03%	$0.30
Adyen	0.03%	$0.13

6 Technology Recommendations

6.1 User Interfaces

6.1.1 Omnichannel Contact Center

WaTech has selected Amazon Connect as the foundational platform for the omnichannel contact center. Amazon Connect is a competitive solution for domain, offering features tailored for diverse customer interactions.

Note: WaTech Amazon Connect expert has recommended leveraging the campaign engine for SMS communication instead of Amazon Connect for this function.

6.1.2 Campaign Engine

Amazon's campaign engine stands out as a competitive choice in the market. While Microsoft offers two strong contenders in this space, they appear to fall short when compared to their counterparts. As a strategic recommendation, Launch suggests aligning the campaign engine with Amazon if it emerges as the leader within the AI ecosystem. Should Amazon not take the forefront, Twilio presents a strong alternative worth considering. The following platforms are recommended based on their respective ecosystems:

- Amazon Pinpoint for Amazon ecosystem
- Twilio for Microsoft ecosystem

Note: The digital experience platforms typically include campaign engines. This may require reevaluation if a decision is made to pursue a digital experience platform in the future.

6.1.3 Web & Mobile

We recommend using React for both web and mobile due to create a mobile friendly website. This ensures efficient development with shared logic across platforms and a seamless user experience on both web and mobile.

6.2 Digital Experience Platforms

Digital experience platforms (DXPs) encompass a wide range of capabilities across various domains, including content management systems (CMS), customer relationship management (CRM), profile management, back-office systems, AI integration, and workflows. Should we decide to procure a digital experience platform, it will be necessary to revisit areas of overlap with our proposed architecture.

6.2.1 Liferay

Liferay stands out as a potential platform solution that aligns with the vision and high-level architecture of the Resident Portal. Liferay is designed to utilize React for creating custom widgets that can be paired with its native widgets, achieving the desired customization and design goals of the Resident Portal. The capability to reuse widgets is another attractive feature that supports replication of similar functionalities across different agency services. Liferay's capabilities with user profiles are impressive and an accelerator for the Resident Portal's capabilities in that space. Back-office and some feature management (A/B testing) support are integrated as well. Liferay is currently leading in the digital experience platform evaluation.

6.2.2 Jahia

Jahia offers a compelling platform solution that aligns with the vision and high-level architecture of the Resident Portal. Designed to fully leverage React for its user interface, Jahia can meet the customization and design goals of the Resident Portal. Jahia's capabilities to reuse widgets, profiles, feature management (A/B testing), and back-office support make it a promising accelerator for recreating similar functionality across multiple agency services. Jahia is currently a close second in the digital experience platform evaluation.

6.2.3 Service Now

Service Now is an industry leader in IT Service Management (ITSM) and customer service management. As we enable Tier 1 Service Integrations, the back-office will become a key factor for agencies to interact with service requests. Utilizing Service Now in headless mode can significantly accelerate the back-office capabilities. If we opt not to pursue other digital experience platforms, Service Now presents itself as an attractive alternative for back-office solutions.

6.2.4 Alfresco

We could not get hands on time with Alfresco and their team. At this time, we do not have enough information on Alfresco and therefore cannot recommend the platform as a part of the solution.

6.3 Content Management System

When evaluating potential content management systems (CMS), Drupal and Contentful emerge as the leading contenders. The race is close enough that we recommend keeping Drupal as the CMS platform. The overhead involved in transitioning to Contentful outweighs the potential benefits. Therefore, maintaining the existing infrastructure with Drupal is deemed the most efficient course of action.

Note: The digital experience platforms typically include campaign engines. This may require reevaluation if a decision is made to pursue a digital experience platform in the future.

6.4 Big Foot Bot / AI

To achieve the best results, it's crucial to select a comprehensive AI ecosystem rather than piecing together disparate solutions. Amazon and Microsoft stand out as industry leaders, offering robust, end-to-end AI frameworks that seamlessly integrate with a wide range of services and tools. Both platforms are well-suited for building the AI components of the Resident Portal, offering scalability, security, and extensive support. A deeper evaluation of each will be essential to identify the optimal choice, as either would provide a strong foundation for AI-driven innovation in the Resident Portal.

- Amazon AWS: Amazon offers a robust suite of solutions for each required technology, including essential capabilities for content ingestion. Additionally, Amazon Connect, which WaTech has already chosen integrates seamlessly within the ecosystem. These strengths position AWS with a slight advantage over its competitors.
- Microsoft Azure: Microsoft also provides solutions for each necessary technology, though it lacks a solution for content ingestion. As a result, separate technologies will need to be sourced to address content ingestion requirements. Nevertheless, Microsoft Azure's AI solutions and overall capabilities ensure that it remains a strong contender in this landscape.

6.4.1 AI Components

The following AI components are recommended as part of the Resident Portal's framework:

- Intelligent Search
- Large Language Model (LLM)
- Next Best Action (NBA) Engine
- Conversation API
- Content Ingestion

6.4.2 Content Ingestion

6.4.2.1 Drupal Connector & Indexer

To ensure efficient content ingestion, we recommend aligning the campaign engine with the winner of the AI ecosystem.

- Amazon: Amazon provides a Drupal Connector & Indexer.
- Microsoft: Microsoft requires a third-party tool to deliver the content from Drupal to Blob storage for indexing.
    - We recommend using the Raytion Drupal Connector.

6.4.2.2 Web Crawler & Indexer

- Amazon: Amazon also provides a Web Crawler & Indexer.
- Microsoft: Microsoft requires a third-party tool to deliver the web content to Blob storage for indexing.
    - We recommend Website Crawler by BA Insight.

6.5 Feature Management

LaunchDarkly is the industry leader for feature management. We recommend LaunchDarkly to enhance our feature management and improve our continuous delivery process.

This tool allows for decoupling of deployments from releases, enabling controlled rollouts, instant feature toggling, and safer, faster feature releases. By using LaunchDarkly, we can better manage risks and deliver features to production with increased flexibility and confidence.

We were not able to reach LaunchDarkly for enterprise pricing details.

Next Steps: Contact LaunchDarkly to inquire about enterprise pricing options.

6.6 Data Lake

WaTech's Data Architect recommends a data lake for shared data and Resident Portal data in general. Amazon, Microsoft, and Databricks all have strong offerings. Further investigation will be required to make an informed selection.

6.7 Recommendations Summary

6.7.1 User Interfaces

	User Interface Recommendations
Web & Mobile Front End	React
Contact Center	Amazon Connect (Preselected by WaTech)
Campaign Engine	[TBD: Amazon Pinpoint | Twilio]
Digital Experience Platform	[TBD: Liferay | Jahia | Service Now]

6.7.2 Big Foot Bot / AI

	Amazon Centric AI Approach	Microsoft Centric AI Approach
Large Language Model (LLM)	Amazon Bedrock	Azure OpenAI (includes GPT-3/4 & Codex)
Next Best Action Engine	Amazon Personalize	Azure Personalizer
Conversational API	Amazon Lex	Azure Bot Service with LUIS
Intelligent Search	Amazon Kendra	Azure Cognitive Search
Web Crawler	Amazon Kendra Web Crawler	BA Insight Website Crawler & Azure Cognitive Search
Drupal Connector & Indexer	Amazon Kendra Drupal connector	Raytion Drupal Connector & Azure Cognitive Search

7 PLATFORM ARCHITECTURE

This section outlines the various technology components of the platform. The diagrams below illustrate how these components integrate and work together to create a cohesive architecture, providing a comprehensive overview of the platform's structure and functionality.

7.1 PLATFORM ARCHITECTURE - AMAZON CENTRIC AI APPROACH

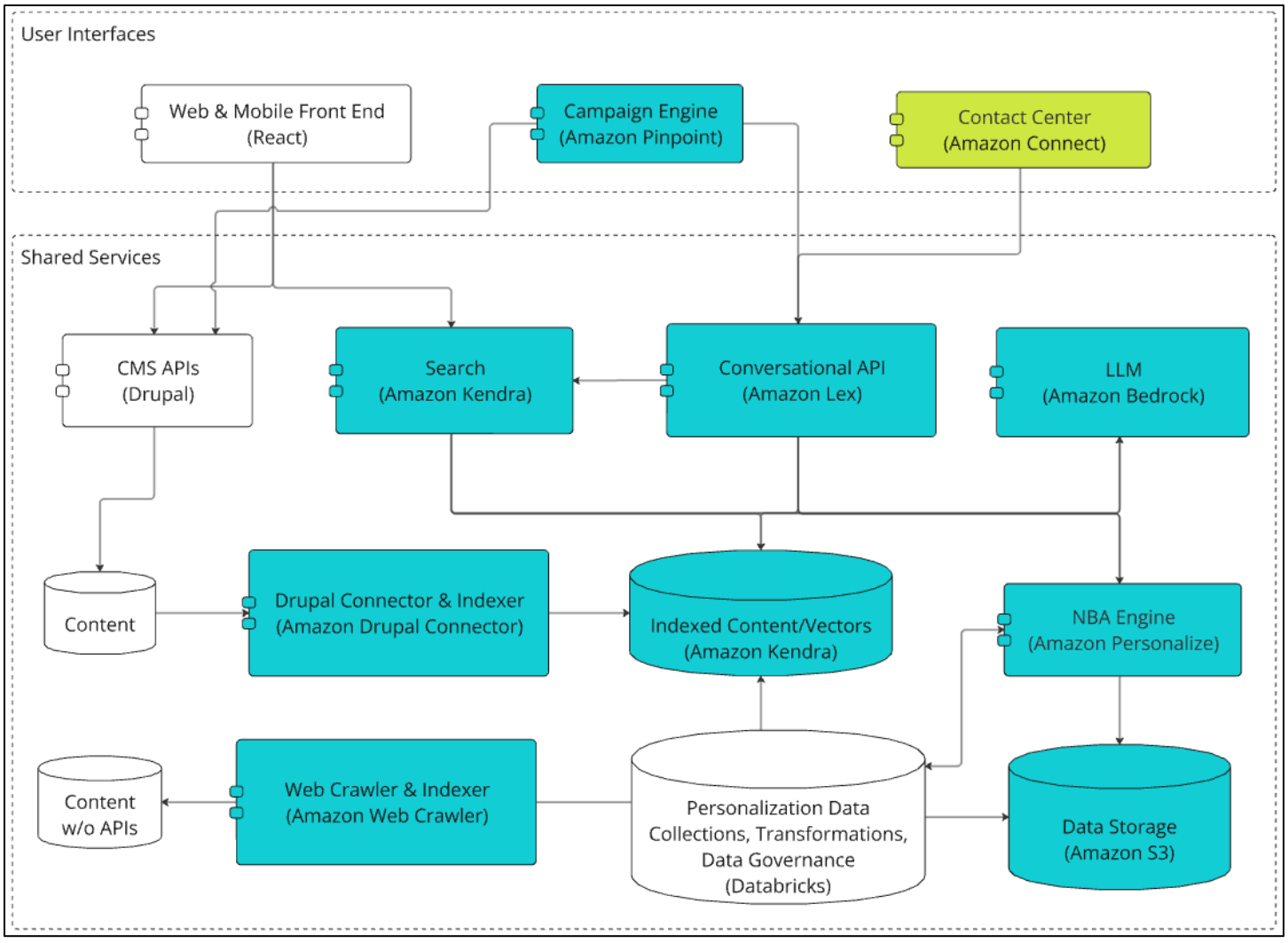


Figure 1: Platform Architecture - Amazon Centric AI Approach

7.2 Platform Architecture - Microsoft Centric AI Approach

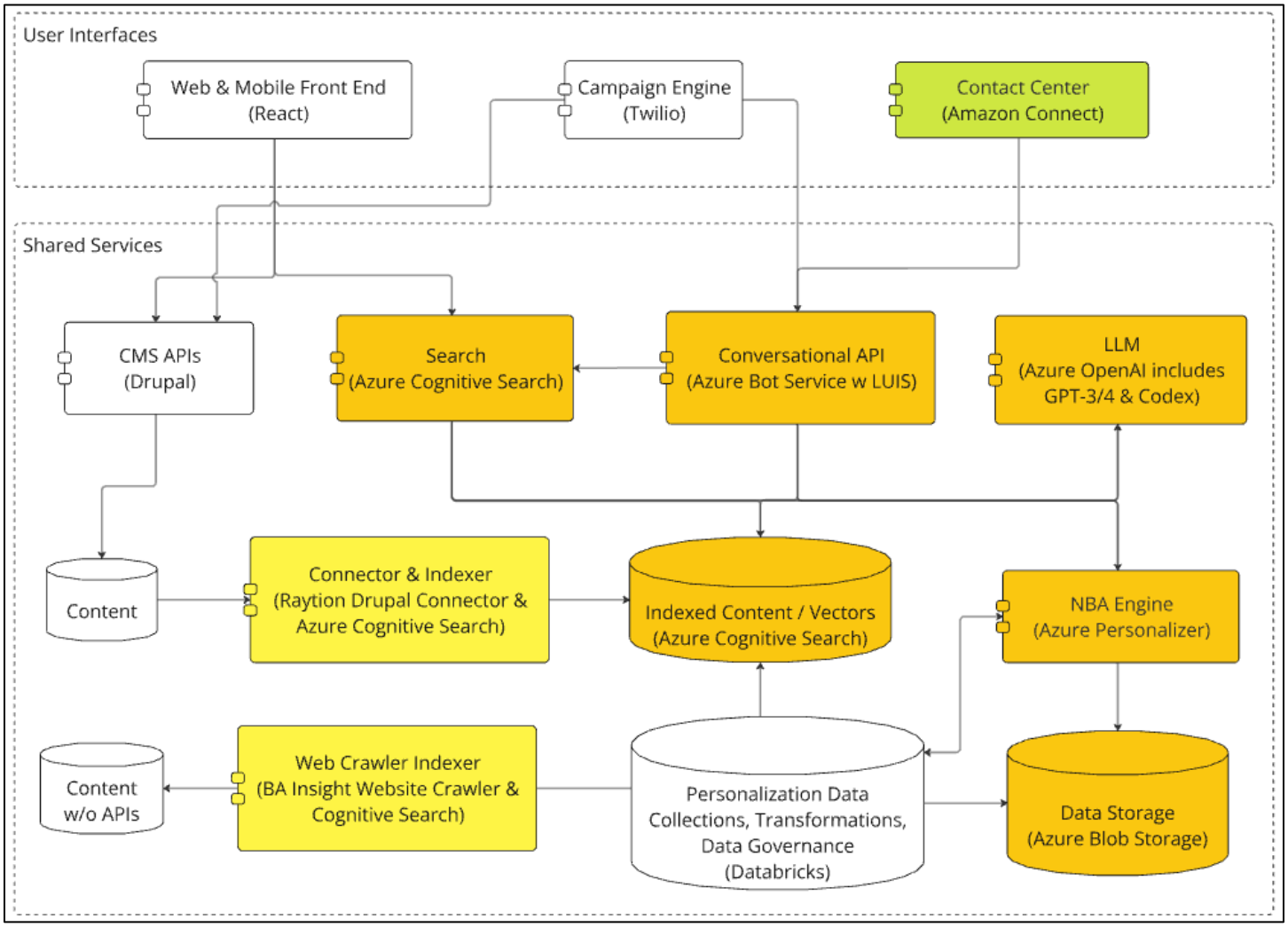


Figure 2: Platform Architecture - Microsoft Centric AI Approach

7.3 Platform Architecture – Color Key

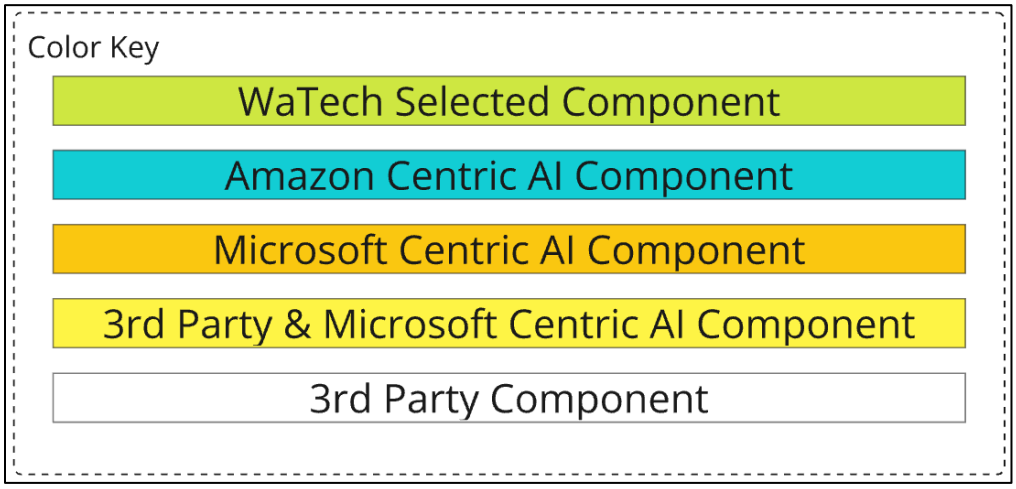


Figure 3: Platform Architecture - Color Key

8 KEY COMPONENTS

8.1 THE USER INTERFACES

The web & mobile frontends, along with the campaign engine and contact center integrate with our AI to foster a coherent, connected omnichannel experience across agencies.

8.1.1 Omnichannel Contact Center (Amazon Connect – Preselected by WaTech)

Provide residents with easy access to government services via phone, video calls, chat, and email. Integrate with Big Foot Bot (our AI Bot) for intelligent self-service options or routing to the correct service regardless of agency.

Requirements:

8.1.1.1 Enable multi-channel support (phone, video calls, chat, email).

8.1.1.2 Implement intelligent routing based on resident inquiries.

8.1.1.3 Integrate with Big Foot Bot.

8.1.2 Campaign Engine (Amazon Pinpoint | Twilio)

Deliver personalized messages, reminders, and updates through email, SMS, and push notifications to residents based on their service interactions and preferences.

Requirements:

8.1.2.1 Manage campaigns for email, SMS, and push notifications.

8.1.2.2 Personalize messages based on user behavior, profile, and interaction history.

8.1.2.3 Ensure compliance with communication preferences and data privacy regulations.

8.1.2.4 Integrate with Big Foot Bot.

8.1.3 Web & Mobile Front-ends (React)

Create a modern and responsive web front-end interface using React to ensure a smooth, interactive user experience. The web front-end also provides device agnostic, and friendly access to the Resident Portal.

Requirements:

8.1.3.1 Implement a component-based architecture for UI consistency and reusability on various devices (mobile, tablet, desktop).

8.1.3.2 Integrate with back-end services, including Big Foot Bot.

8.1.3.3 Ensure responsiveness on various devices (mobile, tablet, desktop).

8.1.3.4 Ensure accessibility on various devices (mobile, tablet, desktop).

8.1.3.5 Ensure functionality on various devices (mobile, tablet, desktop).

8.1.3.6 Ensure look and feel on various devices (mobile, tablet, desktop).

8.1.3.7 Implement security best practices for frontend.

8.2 Big Foot Bot / AI

Big Foot Bot is the face of the services that leverage AI to provide the resident's an unrivaled experience with personalized content, intuitive process flows, personalized service recommendations, intelligent search with further enhanced results, and seamless interaction across multiple channels. For the best user experience, it is recommended to have multiple personas for the chat bot. The main bot should be general purposed, able to handle a breadth of topics. Specialized bots, dedicated to in depth focus on a specific area(s), could take the form of separate bot personas or assistants to be called on by the main bot.

8.2.1 Big Foot Bot Architecture

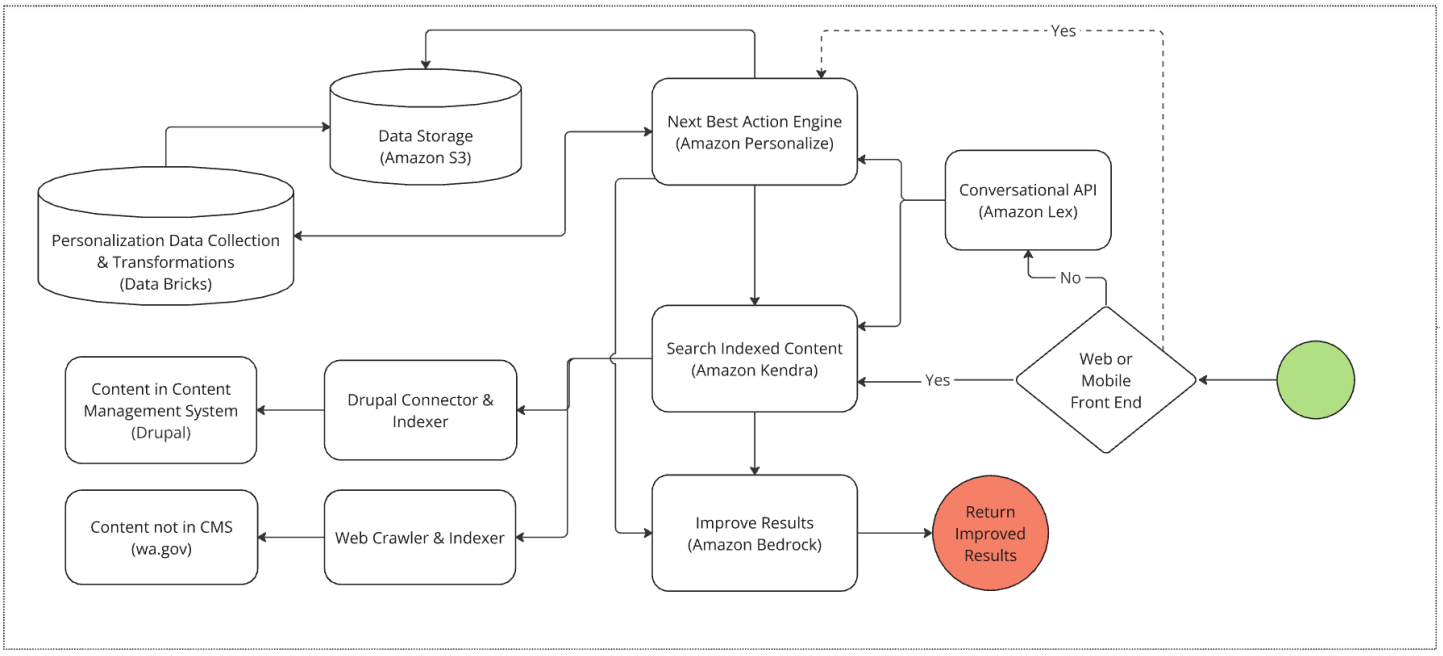


Figure 4: Big Foot Bot Architecture

8.2.2 Determining which Agency Service

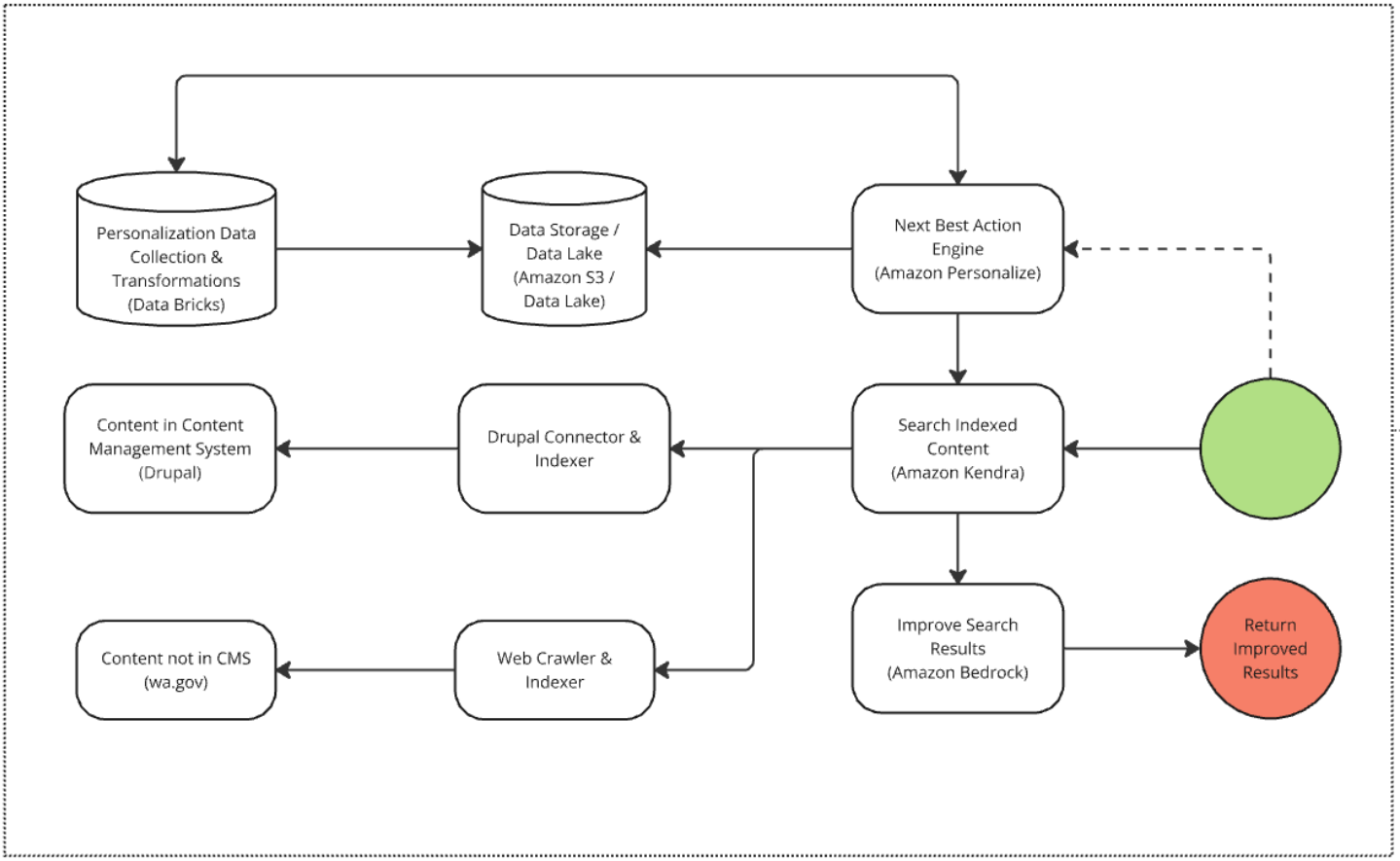


Figure 5: Determining which Agency Service

8.2.3 Conversational Interfaces / Chat Bots (Amazon Lex | Azure Bot Service with LUIS)

Enable residents to interact with chatbots and voice assistants to access information, make appointments, apply for services, or complete transactions through natural language.

Requirements:

8.2.3.1 Support for natural language processing (NLP) for chatbots and voice assistants.

8.2.3.2 Integration with Amazon Connect for conversational IVR (Interactive Voice Response).

8.2.3.3 General Purpose AI Bot(s): Provide versatile, multi-function bots that handle a wide range of common tasks or inquiries.

8.2.3.4 Specialized AI Bots: Provide specialized bots, offering in-depth assistance and domain knowledge for particular and / or complex government functions.

8.2.3.5 Minimize persona switching between AI Bots.

8.2.3.6 Voice Assistants: Provide a Conversational Bot for Voice dialogue

8.2.4 Generative AI / Large Language Models (Amazon Bedrock | Azure OpenAI includes GPT-3, GPT-4, & Codex)

Use AI models to generate relevant content, summarize documents, answer complex inquiries, and provide personalized guidance based on resident needs, questions, and context.

Requirements:

8.2.4.1 Leverage pre-trained models for content generation and query resolution.

8.2.4.2 Summarize long documents or sets of search results, providing users with concise and understandable results.

8.2.4.3 Contextual Answer Generation: generate clear, relevant, and cohesive answers to complex questions by synthesizing information from search results instead of just returning raw content.

8.2.4.4 Ensure privacy and data compliance when processing user information.

8.2.5 Real-Time Personalization / Next Best Action Engine (Amazon Personalize | Azure Personalizer)

The Next Best Action (NBA) Engine delivers tailored recommendations and services based on resident behavior and preferences, ensuring each user's experience is customized.

Requirements:

8.2.5.1 Deliver real-time personalized content and service recommendations.

8.2.5.2 Analyze user behavior and adapt content dynamically.

8.2.6 Centralized Content Management / Headless CMS (Drupal)

Manage content centrally through a headless CMS that can distribute information to the user interfaces. The CMS will provide content to the UIs. The content from the CMS will also be indexed, making it available for Big Foot Bot in the form of intelligent searches.

Requirements:

8.2.6.1 API-first approach to allow easy integration with the User Interfaces, Big Foot Bot, and other services.

8.2.6.2 Connector for seamless integration with indexing and search tools (Kendra, web crawler).

8.2.7 Intelligent Search (Amazon Kendra | Azure Cognitive Search)

Provide an AI-driven search functionality that helps residents quickly find information across various government services and documents. The intelligent search will leverage the indexed content provided by the web crawler and headless CMS indexer.

Requirements:

8.2.7.1 Natural Language Understanding (NLU): Interpret questions and inquires phrased in everyday language and provide the most relevant results.

8.2.7.2 Contextual Relevance: Rank results based on their relevance to the search query, considering metadata and content.

8.2.7.3 Document Understanding: Extract answers from within documents and provide specific passages, not just entire files, as search results.

8.2.8 Dynamic Content Discovery / Web Crawler & Web Crawler Indexer (Amazon Web Crawler | BA Insight Website Crawler & Cognitive Search)

Utilize a web crawler to scan and discover agency web content, ensuring up-to-date information is displayed in the Resident Portal even when the content is not hosted in the CMS. The indexer organizes and tags the data so it can be easily searched.

Requirements:

8.2.8.1 Traverse agency websites not hosted in the CMS to retrieves new or updated information and sends this data to the web crawler indexer.

8.2.8.2 Include content from the web crawler in the overall search index and search results.

8.2.9 CMS Connector & Indexer (Amazon Drupal Connector | Raytion Drupal Connector & Azure Cognitive Search)

The CMS Connector and an Indexer work together to ensure that content from the Content Management System (CMS) is accessible, up-to-date, and easily searchable within the Resident Portal. The CMS Connector communicates with the CMS through APIs, fetching content and ensuring that any new content or changes in the CMS are recognized. The CMS Indexer is responsible for taking the content retrieved by the CMS Connector and indexing it, so that it can be efficiently included in search results.

Requirements:

8.2.9.1 Fetch Content: The CMS Connector pulls content and metadata from the CMS, ensuring it has the latest relevant content. The CMS Connector ensures that any new content or updates in the CMS are available for indexing.

8.2.9.2 Deliver Content to the Indexer: The CMS Connector sends the content to the CMS Indexer.

8.2.9.3 Index Content: The CMS Indexer analyzes the content provided by the CMS Connector. It organizes, tags, and categorizes the content, building an efficient search index.

8.2.9.4 Continuous Updates: As the CMS Connector monitors the CMS for new content or changes, it informs the CMS Indexer, which then updates the search index to reflect these changes in real time or at regular intervals.

8.2.9.5 Search Inclusion: Include content from the CMS in the overall search index and search results.

9 ALIGNMENT TO JOURNEYS AND KEY CAPABILITIES

The user journeys were walked as an exercise to analyze alignment with key capabilities and technical direction. A sample of that exercise is depicted below. The resulting insight helped validate the key focus areas and identify enigmatic areas / black boxes to demystify.

9.1 WALKING ANDREW'S JOURNEY WITH THE TECH - PART 1

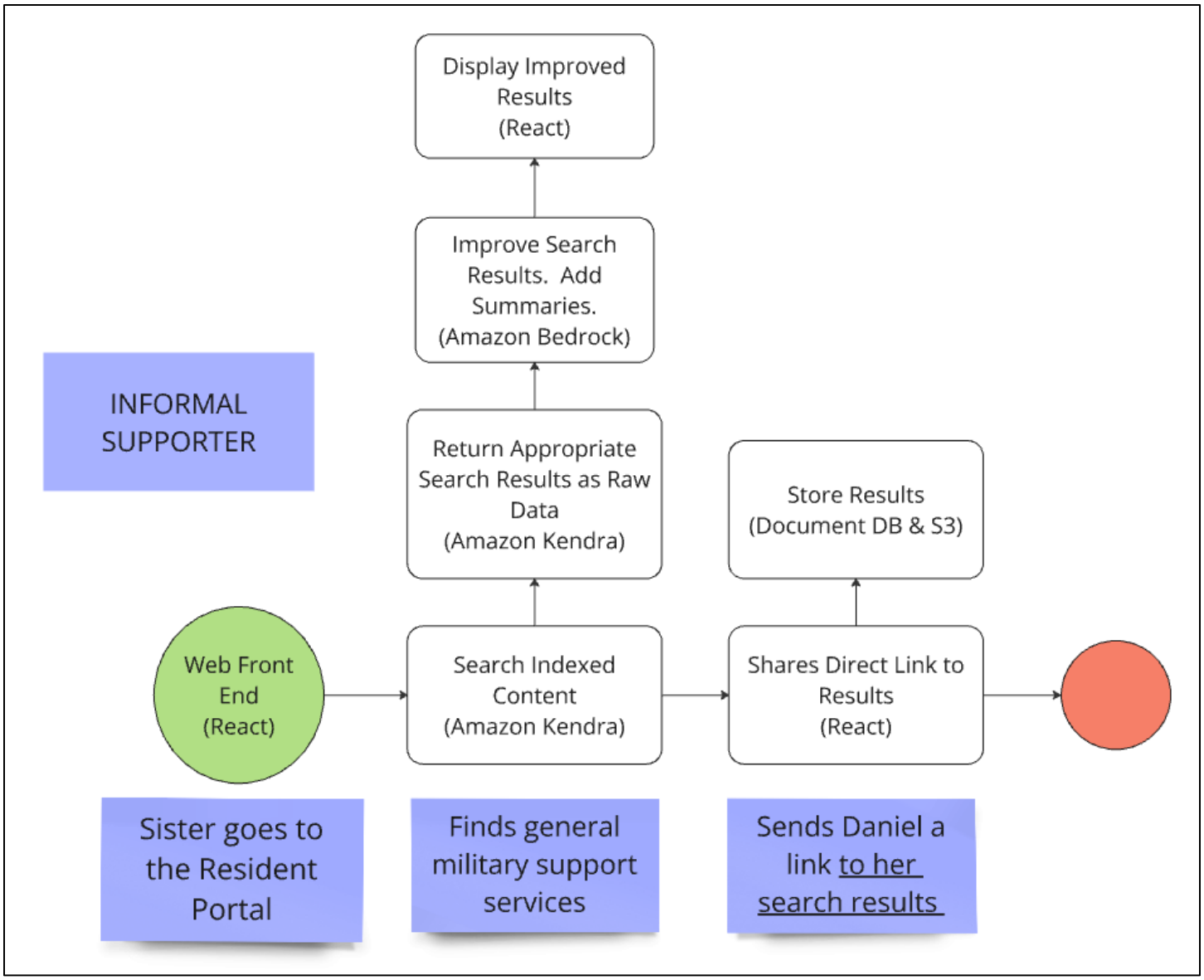


Figure 6: Walking Andrew's Journey with the Tech - Part 1

9.2 WALKING ANDREW'S JOURNEY WITH THE TECH - PART 2

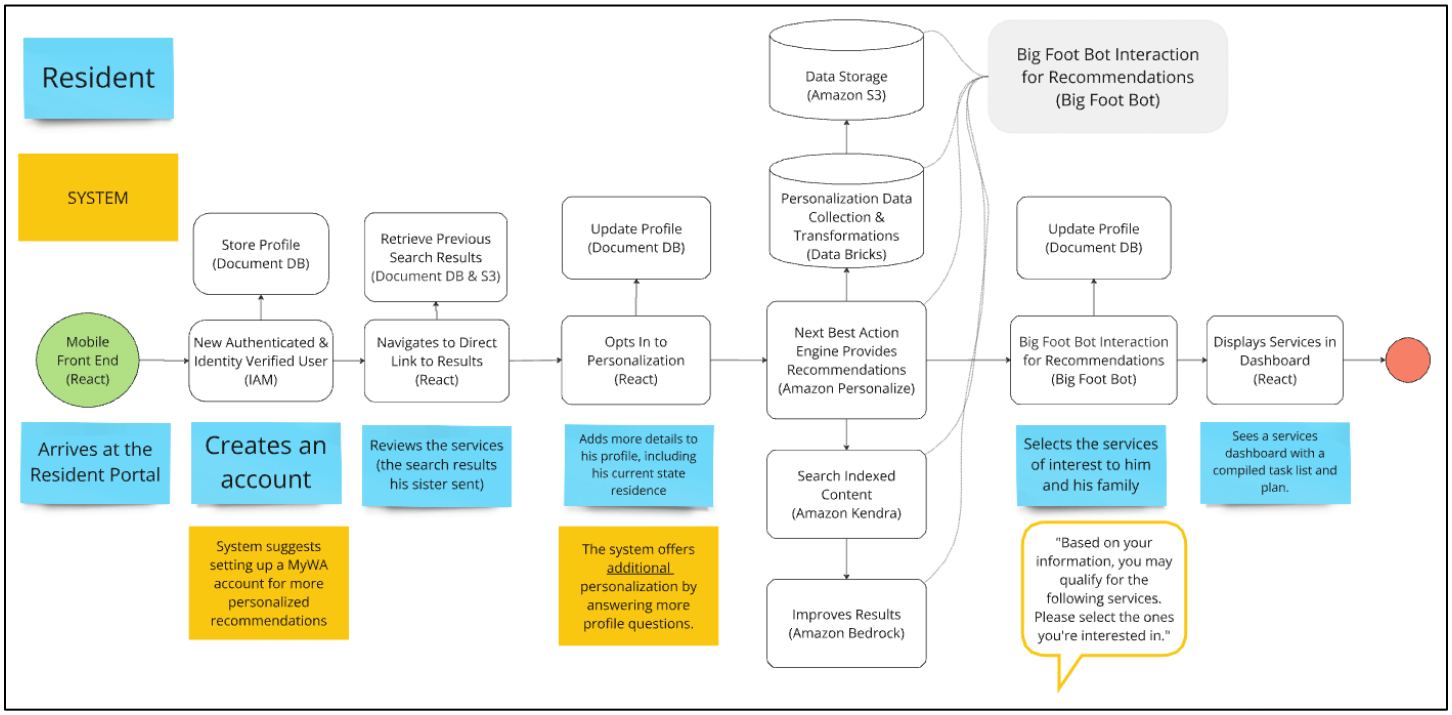


Figure 7: Walking Andrew's Journey with the Tech - Part 2

10 BACK-OFFICE RECOMMENDATIONS

The back-office for handling administrative tasks for the system. Additional technology requirements may be needed to support key administrative tasks.

10.1 BACK-OFFICE TOOLS

Leverage native back-office tools before investing in custom solutions whenever possible, especially in the early stages of development / implementation. The primary focus should be on the Resident Portal. Invest in the Back-Office where necessary and as time permits. The back-office needs to be fully functional but does not need all the bells and whistles on day one. Identify, prioritize, and plug gaps in the native tools where necessary, while prioritizing nice-to-haves. For example: utilize the native tools of the CSM to manage content.

10.2 INTEGRATED BACK-OFFICE CAPABILITIES

Include back-office capabilities in the Agency Capability Cohort. The functionality should be implemented as a combination of tech enablers and implementation in the portal, like other capabilities. Define the super user / back-office functionality. Establish specific roles and permissions. Leverage IAM for role-based access control. For example: a back-office capability to enable an agent to schedule an appointment for a resident, should be included, prioritized, and potentially implemented (depending on priority) in Agency Capability Cohort enabling the resident's similar capability.

11 Agency Integration Strategy & Architecture

This section discusses a strategic approach to enabling capabilities and integrating agencies and their services into the Resident Portal.

11.1 Easy Wins

Some states have resident portals that are essentially pretty link farms. Contrary to initial bias, and certainly depending on the presentation and execution, if done right this approach can actually create satisfying user experiences. Tennessee, for instance, has a mobile app, MyTN, with a hybrid approach between integrated application functionality and deep linking to agency websites. MyTN allows residents to navigate through the many agency services, then seamlessly direct link to the desired agency's website via the app. The look and feel gives the impression of a cohesive experience, regardless of the level of integration.

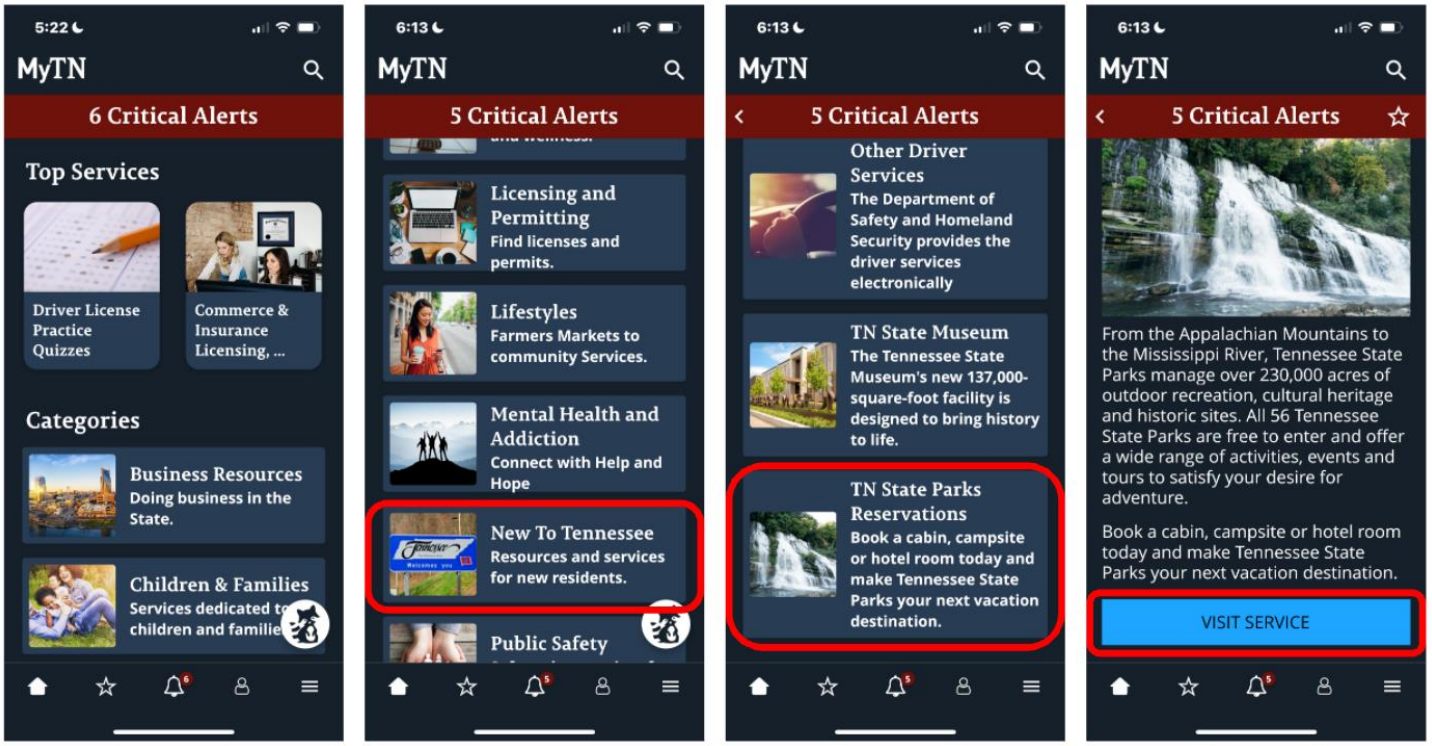


Figure 8: MyTN - New Resident Booking a Campsite - 1 of 2

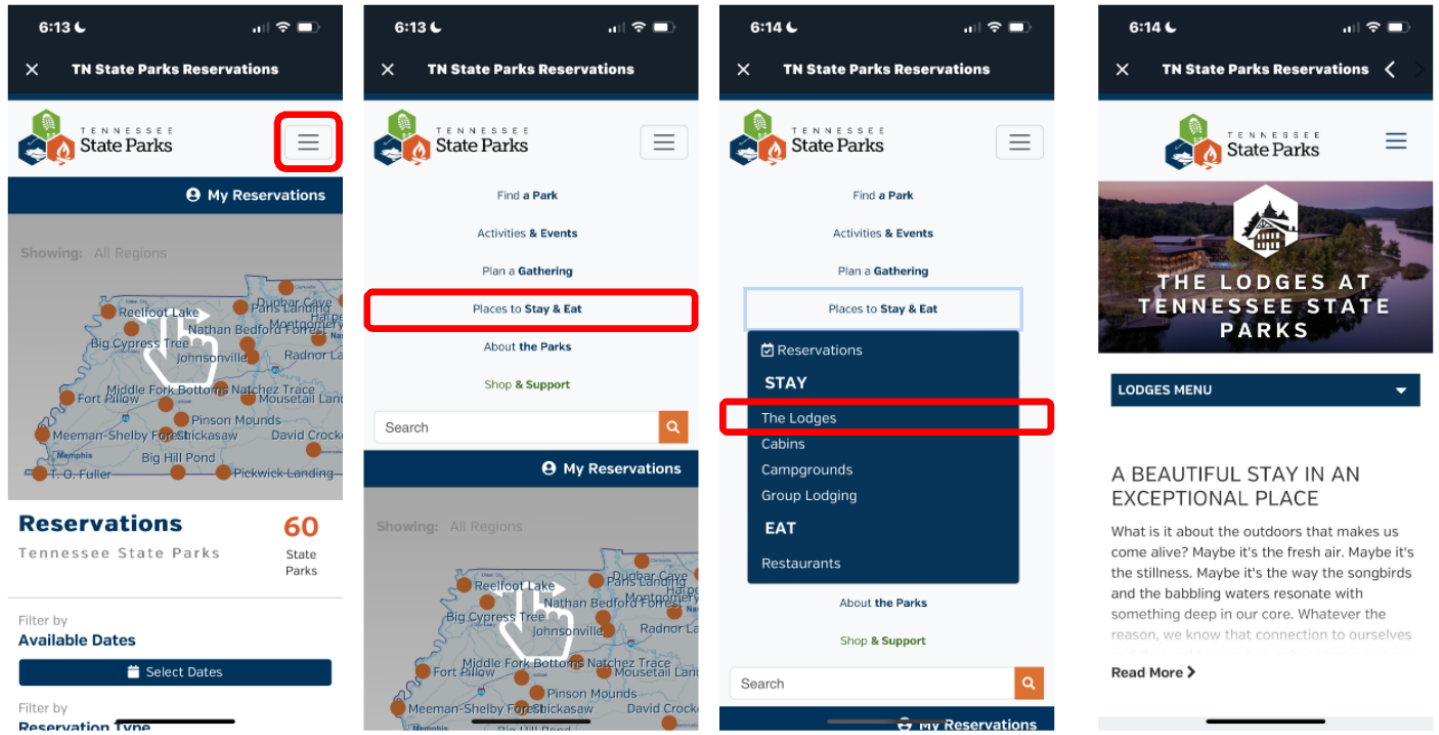


Figure 9: MyTN - New Resident Booking a Campsite - 2 of 2

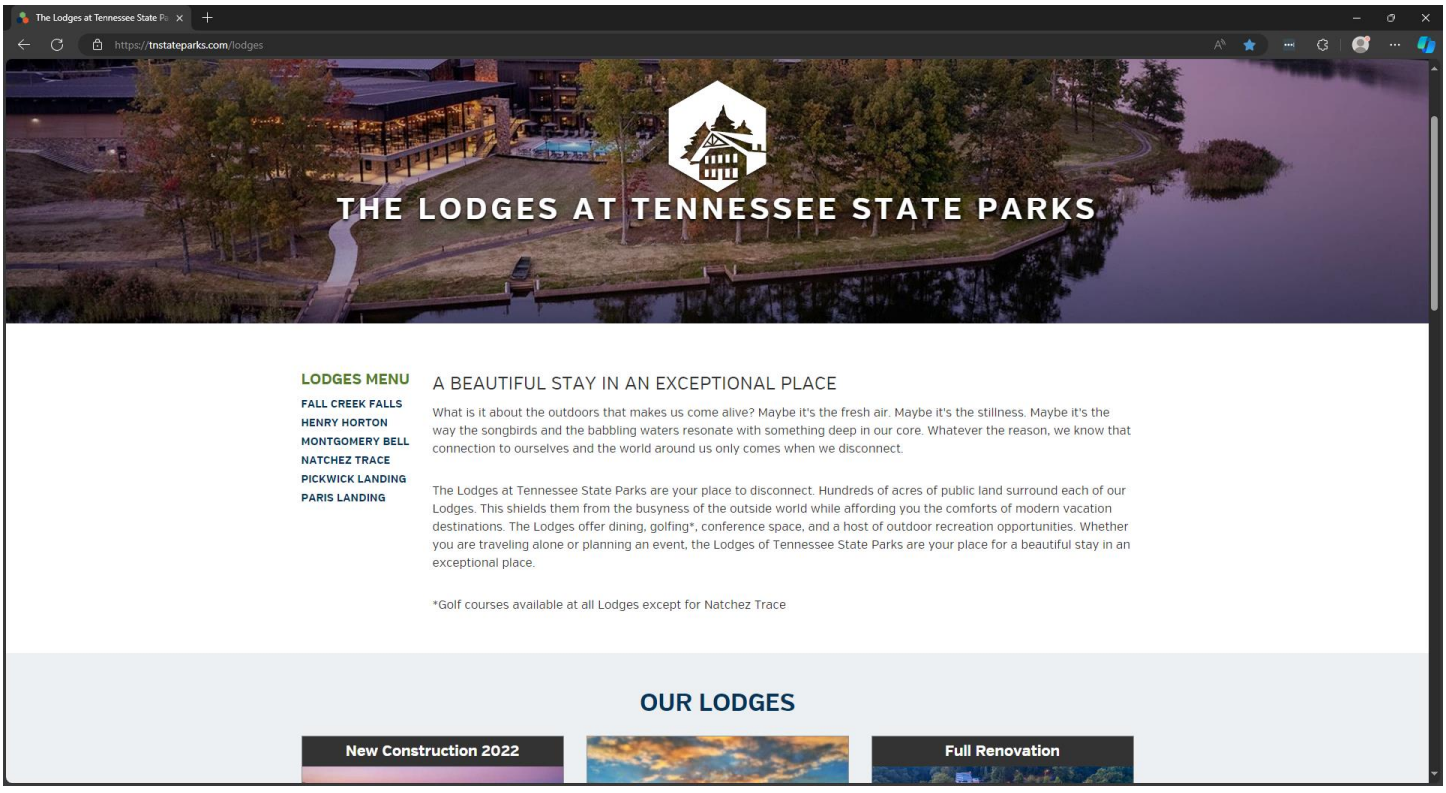


Figure 10: TN.gov --> TNStateParks.com - Resident Booking a Campsite

Learning from other states and quickly delivering an initial Resident Portal with these capabilities would be a quick win for both residents and the portal.

11.1.1 Complete List of Services with Direct Links

Having the list of agency services in a single location is a win. The Resident Portal can benefit by providing a comprehensive list of agency services. Presentation will shape the perception. Pairing the full list of services with an intelligent service finder to help residents locate the desired one would complete the offering.

11.1.2 Shared Portal Navigation

Some states failed to implement shared navigation and themes across their agencies. While themes are cosmetic, they contribute to the feeling of a unified experience. A lack of shared navigation completely disrupts this immersion. The Resident Portal could benefit by encouraging agencies to adopt shared navigation with a direct link back to the portal. A shared theme or color palette would be the finishing touch to enhance the unified experience.

11.2 Service Integration Levels

The integration levels refer to how a specific service is provided by the agency. Since agencies offer various services, each one can have a different level of integration and can be added to the portal at different times. Understanding the service integration levels will help WaTech and the agency determine expectations for including services in the Resident Portal.

11.2.1 Service Integration Tiers

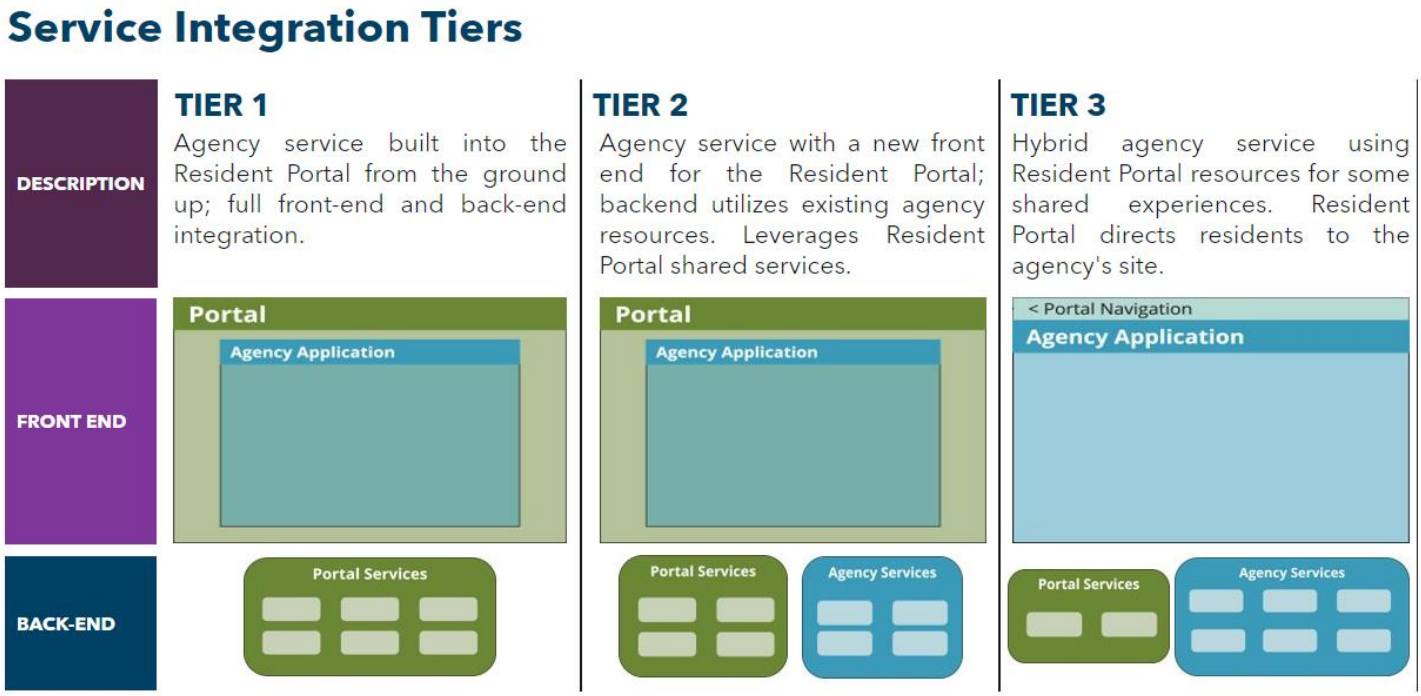


Figure 11: Service Integration Tiers

11.2.2 Tier 1 - 100% Portal Integration

This level of integration describes an agency service is built directly into the Resident Portal. For this level of service, the service's front-end and back-end are both fully integrated with the Resident Portal and subscribe to

the portal's established strategies and initiatives. This is the most suitable level of integration for new, innovative, and visionary services. Technology Enablers and Cohort Implementation are required at this level of integration.

Some examples of services that may seek this level of integration include:

- A service that is currently managed only through manual or paper processes.
- A service run by legacy system that needs modernization or a complete upgrade.
- An innovative service that does not currently exist.
- A new service.
- A service owned by an agency that desires to adopt the strategies and initiatives of the Resident Portal.

11.2.2.1 Funding and developing Tier 1 agency services

Despite being built directly into the Resident Portal, the staffing, financing, and development of a Tier 1 service can be managed through several different arrangements depending on what is politically and budgetarily feasibly at the time. A few examples include:

- WaTech could entirely fund, staff, and build the service, engaging with the service is being developed for as if they were a client.
- WaTech could staff and build the service while the agency the service is being developed for funds its development through the agency's budget or grants.
- An agency could fund, staff, and build the service under WaTech guidance and specifications; the completed service would be hosted by WaTech on the Resident Portal infrastructure.

Ideally the backend is abstracted away to an API interface registered with the API Gateway. With this approach, there will be little difference between the high-level view of a Tier 1 and Tier 2 service integration.

11.2.3 Tier 2 - Fully Integrated Front-end & Agency Back-end

This level of integration describes an agency service that features a new front-end specifically designed for the Resident Portal. The back-end service utilizes existing agency resources and is registered with the Resident Portal, providing a set of minimum capabilities. Services at this integration level should provide a Tier 1 user experience for residents. While innovative services can exist at this tier, they may become cumbersome as new capabilities are added. Resident Portal Technology Enablement is required for this level of integration, but Resident Portal Technology Implementation may not be required, as most of the implementation is managed on the agency's side. Services at this tier should share status updates, payment options, and utilize other shared services to provide residents with a uniform Resident Portal experience.

Some examples of services that may seek this level of integration include:

- A modernized service that is currently available as a microservice.
- A complex service that would require significant investment to be rewritten.
- A service that could benefit from added capabilities.
- A service owned by an agency that desires to adopt the strategies and initiatives of the Resident Portal.
- An intermediate solution for a service marching towards a Tier 1 integration.

11.2.3.1 Key difference between Tiers 1 and 2:

The key difference between Tiers 1 and 2 is the back end. Tier 1 and 2 service integrations should provide residents with a consistent user experience through the Resident Portal. However, the back-end processing differentiates the two tiers: Tier 1 service integrations handle everything via the Resident Portal; Tier 2 service integrations hand

the back-end processing to an agency-managed back-end system. The handoff should be coordinated via an API interface registered with the API Gateway.

11.2.3.2 Funding and developing Tier 2 agency services

Like Tier 1 services, WaTech and the agency can manage Tier 2 service implementation through various arrangements, collaborating to ensure adherence to Resident Portal strategies for a cohesive user experience. The agency will need to encapsulate its service offering into a single interface, registered with the API Gateway. Modifications may be required to meet the Resident Portal's user experience standards, such as sharing status via the Share Status API.

11.2.4 Tier 3 - Hybrid Integration

This level of integration describes an agency service that is a hybrid. Some aspects of the agency service utilize Resident Portal resources and / or provides residents some aspects of a shared experience.

For example:

- The service may use the Resident Portal for processing payments.
- The service may share status information with the Resident Portal, allowing residents to see status updates and receive alerts, notifications, and reminders.

Resident Portal Technology Enablement is required for this level of integration. However, Resident Portal Technology Implementation (depending on capability) may not be required, as most of the implementation is managed on the agency's side.

Some examples of services that may seek this level of integration include:

- A modernized service that is currently available as a microservice.
- A complex service that would require significant investment to be rewritten.
- A service that could benefit from additional capabilities.
- A service owned by an agency that desires to adopt the strategies and initiatives of the Resident Portal.
- An intermediate solution for a service marching towards a Tier 1 or Tier 2 integration.

11.2.4.1 Key difference between Tiers 2 and 3:

The key difference between Tiers 2 and 3 is the front-end / user interfaces. Tier 2 service integrations rely on the Resident Portal for user interactions with the resident, while the back-end is handled primarily by the agency. The agency handles both, the front-end and the back-end, A few shared Resident Portal services are utilized to improve resident value in key areas.

11.2.4.2 Funding and developing Tier 3 agency services

Tier 3 services differ significantly from Tier 1 and Tier 2, as agencies retain ownership of the technology residents use to access, apply for, and process services or benefits. At this tier, agencies enhance existing functionality by leveraging Resident Portal shared services via an API or component, ideally providing residents with additional capabilities, such as shared status, and / or payments, etc.

Examples include:

- Notifications and status updates on a resident's Resident Portal Dashboard regarding the processing of a Tier 3 service (e.g. reminders from DOL that one of the tabs for one of their vehicles is expiring.)
- Links to documents made available for residents in the Resident Portal for quick access but are hosted externally (e.g. a direct link to a PDF of the resident's Discover Pass.)
- A widget that lets a resident use their Resident Profile data to complete an application on an agency website (e.g. let a resident use their Resident Portal address when applying for unemployment benefits.)
- A widget that let's a resident use the Resident Portal payment processor when completing an application on an agency website.
- An agency using data stored in the Resident Portal to verify if the resident qualifies for a service on their website without requiring the resident to re-upload the same documentation or re-enter data.
- A shared software library or service an agency can leverage to reduce the risk of errors caused by making the same calculations across different repositories.

The Resident Portal roadmap assumes that developing Tier 3 integrations is a collaborative effort, where WaTech may be expected to fund the development of the underlying capability, while the agency funds the changes they need to make to take advantage of those capabilities. As with Tiers 1 and Tiers 2 the exact arrangement regarding funding, staffing, and implementation will need to be negotiated.

11.3 Service Integration Tiers Journey Walkthrough – Reserving a campsite

11.3.1 Tier 1 – User Experience

The visual below provides a brief walkthrough of Tier 1 service integration, highlighting the user experience of reserving a campsite through the fully integrated Resident Portal.

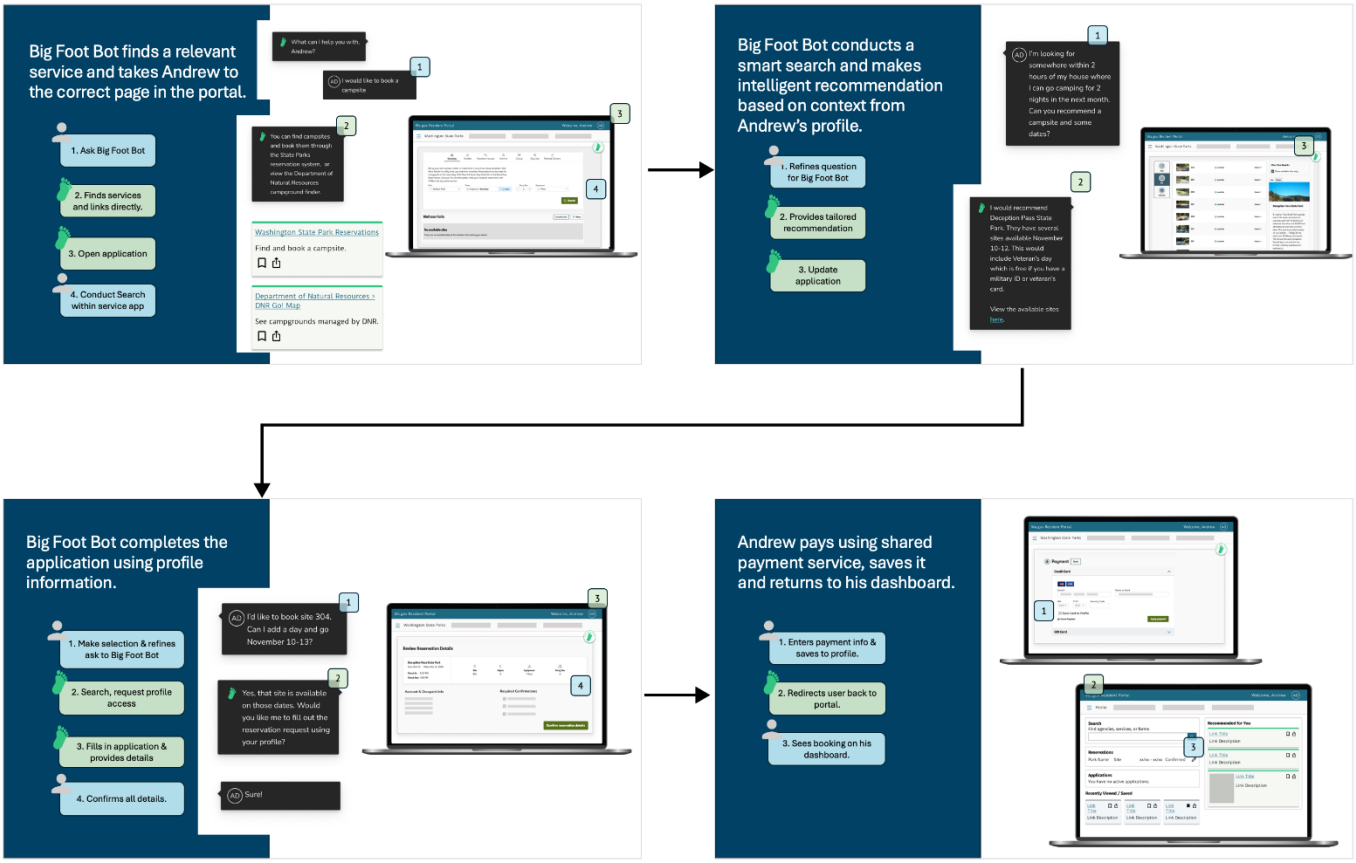


Figure 12: Tier 1 – User Experience

11.3.2 Service Integration Tiers Journey Walkthroughs – Reserving a campsite

Detailed walkthroughs of each service integration tier can be found in this deck. The walkthroughs showcase the user experience and the behind the scenes view of activities that interact with the Resident Portal vs directly with the Agency.

WaTech - Service Integration Tier Walkthroughs.pptx

11.4 Agency Capability Cohort Strategy

Unlocking capabilities in the Resident Portal will be driven in large by the agencies as they integrate their services into the portal. The level of integration of those services will determine the needs of the Resident Portal to enable and / or implement the functionality. Building capabilities into the Resident Portal, without consideration for agency integration would likely waste resources and opportunities. Most capabilities in the Resident Portal will only be relevant to the residents as they correlate with Agency services. Hence, we drafted an Agency Capability Cohort Strategy to guide planning, scope definition, enablement, implementation, and rollout of capabilities closely coupled with Agency Integrations.

11.4.1 Cohort Scope Planning

Identify a collection of capabilities and partner agencies to collaborate on their development. Select agencies for the cohort that share similar interests for a set of capabilities. WaTech will push for capabilities to align with long term Resident Portal roadmap goals. Agencies will push for capabilities of interest to their agency and provided services. WaTech and the Agencies will need to collaborate, compromise, and generate win/win scenarios for the

Resident Portal and Agencies. Enabling capabilities in the Resident Portal without agencies onboarding to utilize those capabilities would fail to maximize value for the value for the residents. Feature Management is recommended and should be considered for each phase of the cohort.

11.4.2 Cohort Scope Definition

WaTech and the selected partner agencies decide which capabilities are the most practical and highest value. Agencies may bring to the table new capabilities that were not originally on the cohort agenda. This should be encouraged, since it indicates that the agencies are invested. WaTech should solicit or request prior notification of agency introduced capabilities, while preparing for the cohort. The selected capabilities are then uniquely prioritized. Capabilities that are deprioritized or at the low end of the priority list may get dropped during the time bound implementation phase.

11.4.3 Cohort Implementation

The desired integration level of the agencies' service determines rather the Resident Portal needs to implement or simply enable the capability. Tier 1 Integrations will require a full implementation. Other tiers may only require enablement (depending on the capability). WaTech and the partner agencies collaborate on the development, enablement, implementation, validation, and deployment of the selected capabilities. The implementation phase should be time bound. Incomplete capabilities should be listed as a high priority capability for the next cohort, although still subject to reprioritization.

11.4.4 Cohort Rollout

WaTech and partner agency staff should receive training on the new capabilities. Once implementation is complete, the new capabilities should be gradually rolled out (in accordance with the feature management plan) and legacy technologies sunset. Resident Portal and agency service finders and listings should point to the new services.

11.5 Technology Enablers

Some infrastructure and / or shared services may be needed to unlock certain capabilities. For example, to display service status information to the resident:

- The Resident Portal needs to be informed of the service's status.
- The agency needs a mechanism to inform the Resident Portal of the service's status.
- The resident needs a way to view the status information.

To unlock the capability mentioned in the above example, a shared service or API will likely need to be provided by WaTech to enable the agency to send status info and the portal to receive it. A separate API would provide the data to the interfaces for resident consumption.

Technology enablers will be identified in the early phase of each Agency Capability Cohort, based on the capability and its service integration level for each agency. Potential capabilities have been analyzed at a high level to ensure the proposed architecture is able to handle a breadth of the capabilities without any major surprises. A more in-depth architecture analysis is needed during each Agency Capability Cohort to ensure alignment, provide low level direction, design, and architecture.

11.6 Shared Services - Resident Portal Owned Services

The resident portal offers a suite of shared services designed to provide a unified experience for residents.

- Shared payment system - provides a shared payment system that supports various payment methods. This allows residents to pay for agency services through a single, secure system.
- Shared transaction status service – provides unified tracking of agency service requests and transactions. The transaction status service provides a centralized way for residents to track the status of their service requests and transactions across all agencies integrated with the portal.
- Omni-channel call center service – provides a unified support service that allows residents to connect with Big Foot Bot and agencies across multiple communication channels.

11.6.1 Payments Shared Service (Square Payments):

The Resident Portal should incorporate a secure, user-friendly payment gateway integration, providing a payment system that supports various payment methods. This allows residents to pay for agency services through a single, secure system.

Requirements:

11.6.1.1 Provide online transactions.

11.6.1.2 Provide in person transactions.

11.6.1.3 Provide manually entered transactions.

11.6.1.4 Provide transactions utilizing saved data.

11.6.1.5 Provide invoice transactions.

11.6.1.6 Provide reports for accounting and auditing.

11.6.1.7 Ensure PCI compliance for secure processing of payment information.

11.6.2 Shared transaction status service

The Resident Portal should provide unified tracking of agency service requests and transactions. The service will enable capabilities for residents to track the status of their service requests and transactions across all agencies integrated with the portal.

11.6.3 Omni-channel call center service

The Resident Portal should provide a unified support service that allows residents to connect with Big Foot Bot and agencies across multiple communication channels.

11.6.4 Identity and Access Management (IAM) - Okta Authentication

Provide seamless user authentication across multiple government services with Okta's Identity Cloud. IAM typically handles Single Sign On (SSO), Multi-factor Authentication (MFA), Identity Verification (IDV) integration, role-based access control, and account recovery for secure access to the Resident Portal. *Note: IAM requirements are managed by WaTech's IAM initiative.*

11.6.5 Identity Verification (IDV)

Identity verification is the process of confirming that an individual is who they claim to be. It is used to ensure secure access to services, prevent fraud, and comply with regulations. The verification process often involves collecting and validating personal information, documents, or biometrics (e.g., ID cards, passports, facial recognition), and may use online databases or real-time checks to confirm the user's identity. *Note: IDV requirements are managed by WaTech's IAM IDV initiative.*

11.7 Agency Owned Services

Agency Owned Services will be the handshake for the Resident Portal to handoff service requests to the agency for Service Integration Tiers 1 and 2. Note that the agency will still own the service (and the data regarding that service), even if implemented by WaTech. Agency Owned Serviced need to be registered with WaTech's API Gateway.

11.8 Feature Management

Feature management is crucial for the Resident Portal because it provides control, flexibility, and responsiveness in managing the portal's functionality and user experience.

11.8.1 Controlled Feature Rollouts

The Resident Portal is intended to serve all of WA. With a large user base and multiple agencies, launching new features can be risky. Feature toggles allow for incremental deployment, allowing new features to be rolled out to smaller user groups, tested, and then gradually expanded to the entire user base.

11.8.2 Minimized Risk and Enhanced Stability

If a new feature causes issues or breaks functionality, it can be quickly turned off without the need for rolling back an entire release, reducing downtime and improving the overall stability of the portal. The ability to deploy features to a subset of users allows for controlled beta testing (A/B testing), early feedback collection, real-world validation, and ensures features work as expected before the broader release.

11.8.3 Personalization

The Resident Portal caters to various audiences – residents, businesses, tourists, government agencies and employees, etc. Feature management enables user segmentation, allowing features to be toggled on or off based on user role, location, or specific needs, ensuring each user has access to the most relevant services. Feature management allows the portal to be tailored to user behavior and preferences, providing a personalized experience.

11.8.4 Accessibility

Accessibility features, such as language support, text-to-speech, or font size adjustments, can be toggled based on user preference, ensuring inclusivity and compliance with accessibility standards.

11.9 Agency Kit

A personalized Agency Kit that includes training materials, prototypes, and functioning code examples can greatly facilitate the onboarding of agencies into the Resident Portal.

11.9.1 Personalization

Focusing on tailored training materials (knowledge guides, templates, tools, and examples) that meet the specific needs of a targeted agency or group of agencies (potentially withing an Agency Capability Cohort) will accelerate adoption, increase efficiency, and result in better agency integrations. The agency kit will reduce the learning curve, speed up implementation, and steer consistency across services from different agencies. The tailored training materials allow the agencies to dive straight into relevant use cases without filtering through generalized content.

11.9.2 Cohesive Service Experience

The agency kit can ensure that services within the Resident Portal maintain consistent UI/UX guidelines, ensuring users experience a uniform look and feel across agencies and services.

11.9.3 Shared Service Prototypes

By offering functioning code examples and prototypes specific to the shared services of Resident Portal for the cohort, the kit can provide reusable components directly aligned with the agency's integration.

11.9.4 Building Resident Trust and Familiarity

The agency kit that's tailored to a specific cohort will help agencies create services that directly align with user expectations. This builds the resident's trust and familiarity with the Resident Portal.

12 Data Strategy, Architecture, Privacy, & Security

Today, resident data is fragmented across agencies; each agency captures and stores resident data such as addresses, phone numbers, and birth dates separately, resulting in a disjointed resident experience:

- Residents must manage contact information such as their address or phone number at each individual agency,
- Residents need to upload and share copies of identity-related documentation multiple times to receive services,
- Residents can accidentally create multiple SAW accounts each connected to a different subset of agencies
- Getting information about application status or finding documents (e.g. copies of professional licenses, Discover Pass, vehicle tab renewals, etc.) requires that residents track that information at each individual agency.

The Resident Portal data strategy aims to consolidate core "commonly used" resident data into a centralized Resident Profile stored and managed by WaTech.

12.1 Data Sharing & Data Security

Creating a consolidated Resident Profile is controversial proposition. Multiple federal laws, state laws, and state policies restrict how and when resident data can be handled and shared between government entities and employees. State agencies are rightfully cautious about sharing resident data between agencies due to the risk of penalties if they inadvertently violate these laws and policies.

More importantly, surveys and interviews indicate that WA residents themselves have serious concerns about how their data may be stored, shared, and used.

- Residents worry that their data will not be properly secured.
- Residents worry that data will be shared "just because" (e.g. DoL employees seeing their SNAP benefits.)
- Residents worry that their data being abused or used to discriminate against them.
- However, agencies need access to resident data for necessary business functions.

For the Resident Portal to be successful, Washington State must adopt a robust data sharing and protection strategy that puts residents in control of what data is shared between agencies and how it is shared. This is especially true of Personally Identifying Information (PII).

12.1.1 Applicable Federal and Washington State Laws

- Interlocal Cooperation Act: Data requests (RCW 39.34.240)
- WA Technology Solutions: Information technology governance (RCW 43.105.054)
- WA Technology Solutions: Office of privacy and data protection (RCW 43.105.369)

- HIPAA, FERPA, CJIS, Gramm-Leach-Bliley Act (GLBA), 42 CFR, Chapter 1, Subchapter A, Part 2

12.1.2 Applicable Washington State Policies

- Enterprise Data Standards Framework (EA-01-02-S was 182.10.10)
- Online File Storage Guidance (EA-01-01-G was 171.01.G)
- Data Center investments policy (EA-02-03-S was 184)
- Metadata Standard (DATA-01-02-S was 187.10)
- Data Sharing Policy (SEC-08 was 141.10(4.2))
- Data Classification Standard (SEC-08-01-S was 141.10(4.1))
- Privacy and Data Protection Policy (DATA-03)
- Securing Information Technology Assets (SEC-01 was 141)
- Washington State Agency Privacy Principles

12.1.3 Data Sharing Agreements Between State Agencies

Washington State law and policies require that agencies establish explicit written data sharing agreements between each other to manage the sharing of a resident's personally identifiable information.

Establishing a standard one-to-many data sharing agreement between WaTech and each agency when onboarding agencies to the portal would allow WaTech to share data that residents submit directly to WaTech (via the Resident Portal) with agencies.

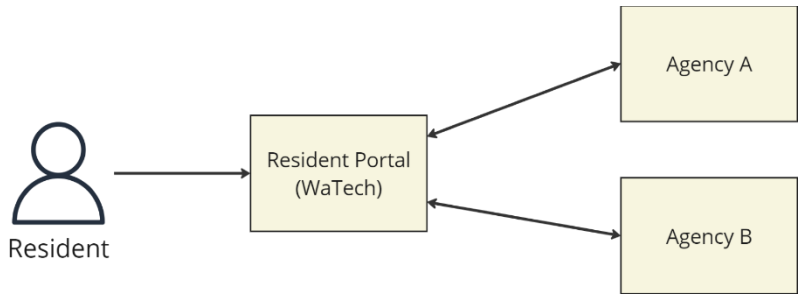


Figure 13: Agencies Exchanging Data - 1 to Many

This one-to-many data sharing agreement also allows the Resident Portal to display resident data owned by an Agency if the resident opts-in. However, WaTech can NOT share data about a resident they receive from an Agency with another Agency without a separate explicit data sharing agreement between those agencies, especially for data that falls into categories 3 and 4 (see Data Classification Standard (SEC-08-01-S was 141.10(4.1).)

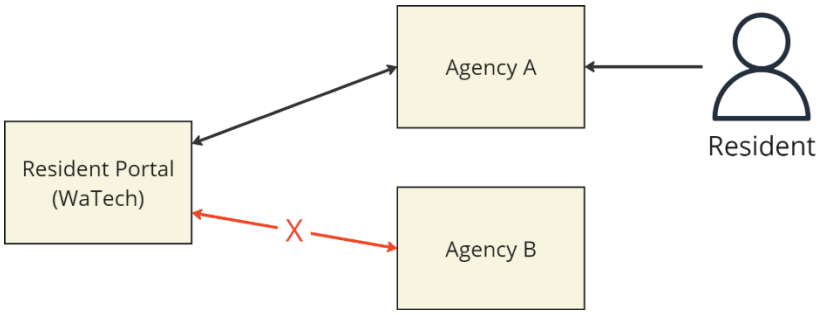


Figure 14: Agencies Exchanging Data - Not Many to Many

12.1.4 Gaining Residents' Consent Builds Trust

While agencies can create exceptions to data sharing policies in the event of legitimate business needs, if the goal is to increase resident trust in government, the Resident Portal should strive to gather active consent from residents to share their data using User-Centric Data Management Approaches.

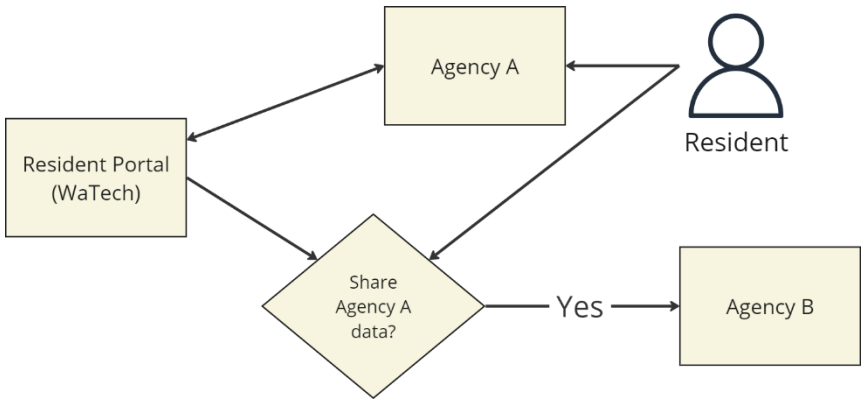


Figure 15: Agencies Exchanging Data - Many to Many Via the Resident

- Agencies can freely ask residents for information about themselves; data provided by a resident directly to an Agency does NOT require a data sharing policy.
- The Resident Portal can explicitly request permission from the Resident to share data with another Agency without a data sharing agreement between these agencies; this request must be explicit about what data will be shared and how it will be used.
- State and Federal Laws may add additional restrictions on data that can be shared across Agencies and the Resident Portal; exceptions cannot be created to circumvent these laws (e.g. HIPAA)
- Outside of limited circumstances where Agencies need specific information about a resident to evaluate a qualification for a service (e.g. firearm registration status,) the vast majority of data sharing transactions will be between WaTech and an Agency in order to manifest Agency data within the Resident Portal or as part of a Centralized Resident Profile.

12.1.5 Anonymized Data Across Agencies

Even in situations where we plan to anonymize and aggregate the data of multiple residents (e.g. recommending a service to a resident based on that service's usage by others from their same zip code,) residents must be informed and consent to the use of their data for these sorts of activities, especially if the data can be directly linked back to an individual (Category 3 or Category 4), such as their name, location, age, etc.

12.1.6 The Right to Data Erasure

Increasing concern around the collection, storage, and use of personal data by large tech companies has given rise to an interest in a "right to be forgotten," with the European Court of Justic solidifying it as a "human right" in a case against Google in 2014. More accurately described as a "right to data erasure," since then, many US states have implemented or considered laws that give individuals the right to request a copy or the removal of all data an entity has on them unless that entity can provide a reasonable business justification for retaining it.

The Resident Portal will need to clearly communicate to residents what data can be deleted; the Resident Portal can only delete data stored within the Resident Portal and some Resident Portal data may need to be retained for a specific justified legal reason.

12.2 Centralizing the "Core" Resident Profile

Consolidating commonly used data (e.g., addresses, phone numbers, emails) into a centralized Resident Profile simplifies data management across state agencies, helps residents better manage and track what agencies have access to their data, and allows state agencies to reduce resources spent on common tasks across agencies (e.g. Identity Verification.)

While, due to a combination of laws, policies, and legacy technology constraints, it is unlikely that the Resident Profile will ever encompass the totality of all data state agencies collect and store on residents, the Resident Profile is well positioned to serve as a central source for "core" (i.e. commonly used resident data points) and as a hub onto which external data sources can connect.

1. Determining what data constitutes "core" resident data requires evaluating each piece of data as agencies are onboarded; generally, "core" resident data should include data commonly used across many agencies, such as a Resident's name, contact information, and documents used for identity verification.
2. In addition to "core" resident data, the Resident Profile will include data WaTech stores on behalf of agency services built entirely into the resident portal (see Tier 1: 100% Portal Integration); this data cannot be exposed within the Resident Portal or shared across Agencies without an explicit data-sharing agreement.
3. Agencies will continue to collect and own data for specific business purposes; ownership and use of this data in Resident Portal will vary and requires collaboration with the relevant agencies to make those decisions:
    a. Some data may be aggregated and displayed within the Resident Portal, but the source of record remains with the agency.
    b. Some data may be collected by an Agency and stored in the Resident Portal to support asynchronous interactions between that Agency and Resident Portal.
    c. Some data may be collected by an Agency but saved to the Resident Profile and owned by Resident Portal with the intent to be shared with other Agencies.
    d. Some data may be collected by an Agency but remain isolated to that agency if there is no reasonable business need or laws preventing it from being shared with the Resident Portal or other Agencies.
    e. Agencies that choose to migrate their backend processes into the portal (Integration Tier 1) may require additional safeguards to ensure that while the Resident Profile stores the data on their behalf, access to and ownership of that data remains in the hands of the relevant Agency (see 2 above.)

12.3 Separating Concerns: Account, Profile, and Verification

Multiple concurrent initiatives at WaTech concern themselves with capturing, storing, and managing resident data, and the overlap between these initiatives and how they fit into the broader in-progress Enterprise Data Governance strategy is still being defined.

It is important to recognize that each of these areas of concern: Identity Access Management (IAM-Okta), Identity Verification (IDV-TBD), Identity Management (IDM), and service delivery to support the development of APIs that exchange data between agencies and the Resident Portal are different with distinct areas of responsibility.

IAM and IDV serve a very specific function, and as a rule the data they steward should be limited to the bare minimum necessary to serve to serve a specific function.

While in many cases a Resident's Profile (including IDM) will be tied to their Resident Account, there are many cases where multiple Residents' Profiles may be managed directly under a single Resident Account.

Examples:

1. Parents who both manage the profiles of their underage dependent children
2. A resident granted power of attorney for their aging parent who does not have Resident Portal Account.
3. A resident may have a separate Business Profile accessed through their Resident Portal Account.

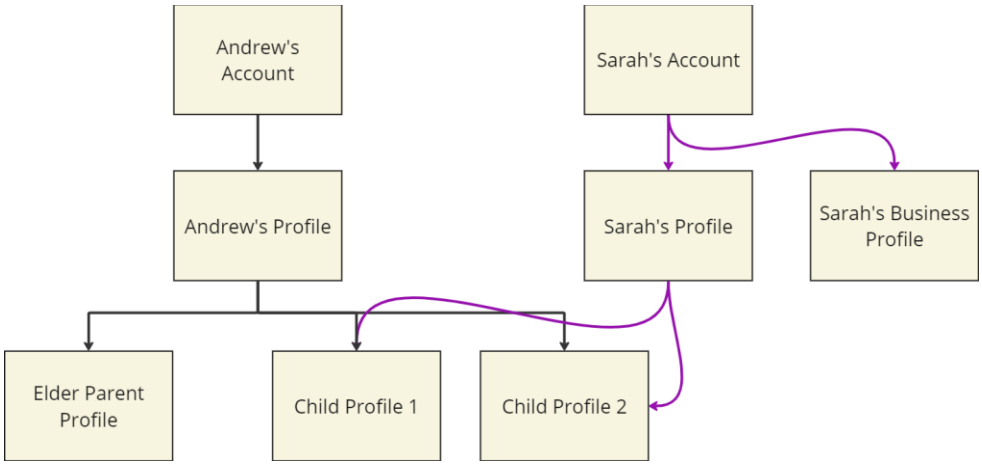


Figure 16: Disambiguating Account from Profile

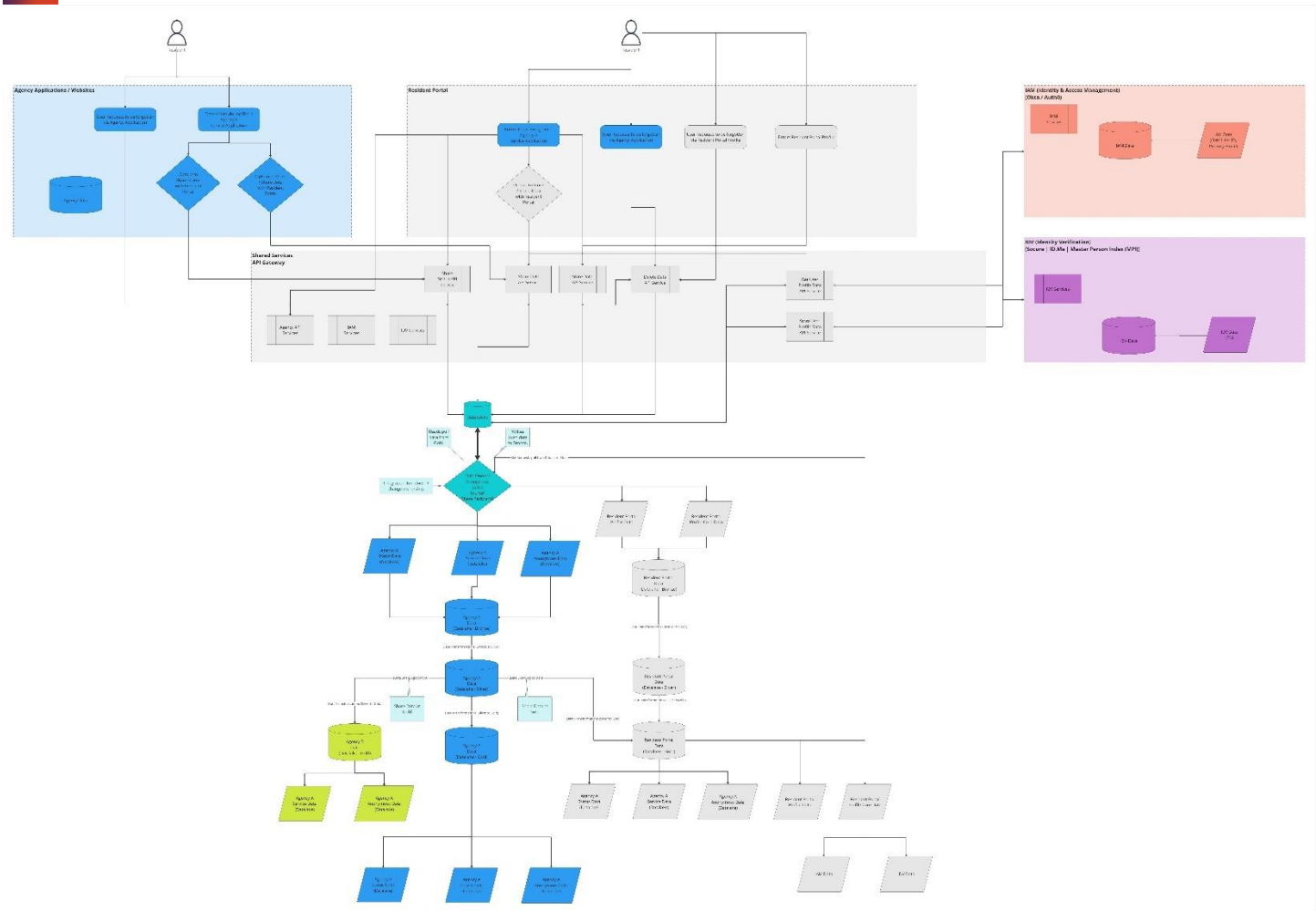
Additional analysis is needed to develop clear policies and a resident experience that supports residents as they go through major life changes including, but not limited to:

1. A child dependent transitioning to an adult and taking ownership of their profile.
2. A divorce that requires two residents to separate their households and re-evaluate access to other associated profiles.
3. A parent or guardian blocking access from someone else (e.g. abusive partner, etc.)
4. A resident granting access to a portion of their profile to a professional supporter, such as an attorney, for the purpose of providing a specific service.
5. A resident delegating access to an employee to manage information related to their business.

Agency administrators and staff acting on behalf of a Resident for a specific service

12.4 Data Architecture

The full Data Architecture diagram is here: WARP Technology & Architecture - Data Architecture Diagram.pdf



12.5 User-Centric Data Management Experiences

Beyond the data architecture, building a user experience that gives residents of their data as they interact with the Resident Portal and various agencies can help build resident trust and avoid legal liability. The user experience should strive to provide residents with a balance of privacy and convenience. As trust in the portal increases or the state of WA state technology becomes less decentralized (federated), some of these privacy strategies could be deprecated or replaced with a more centralized experience.

12.5.1 Tie data collection to a specific value proposition

When defining a data collection strategy, most organizations take a "the more data the better" perspective, often collecting and storing data "just in case" they have an unknown future use for it. However, the more data an organization collects, the greater the risk for the organization if that user data gets compromised.

Furthermore, collecting "irrelevant" information can erode an end user's trust in the organization, increase user friction, and increase the risk they will abandon the experience. Therefore, the Washington State Portal should adopt these strategies for data collection:

- Avoid asking residents for data not necessary to their immediate task.

- Example: Don't require that a resident provide both their email address and phone number if you only need 1 communication channel.

- Use opt-in experiences when collecting data for "value add" purposes (i.e. data that isn't necessary for completing a required task.)

    - Example: Don't collect a resident's income or non-critical details on account creation, instead offer separate "service matcher" experience that lets residents opt-in to volunteering information that helps them find new services.

- Clearly articulate the benefit to the resident in providing their information to the organization.

    - Example: Tell the resident that providing their address would enable discovery of local benefits. Another benefit would be consolidation of potential discrepancies between agencies (with resident approval).

- Avoid storing data that doesn't need to be stored.

    - Example: If a resident fills out a survey stating that they have multiple dependents, only store the information necessary for fulfilling a specific service.

12.5.2 Opting In: Service as "App" Model

Proactively giving residents, the opportunity to choose when and how their data will be shared will increase their trust in the app and in their interactions with government agencies.

One approach to doing so would be to think of the Resident Portal a bit like an "app store," where each individual service is treated as "an app" that users add to their portal (like how SAW users "add" services to their profile today.)

Just like when installing a new app on their iPhone or Android device, when a resident finds a new agency service using the Service Finder, the Resident Portal could inform that resident of what data that service is requesting permission to use, and the resident could be given the opportunity to grant or deny access.

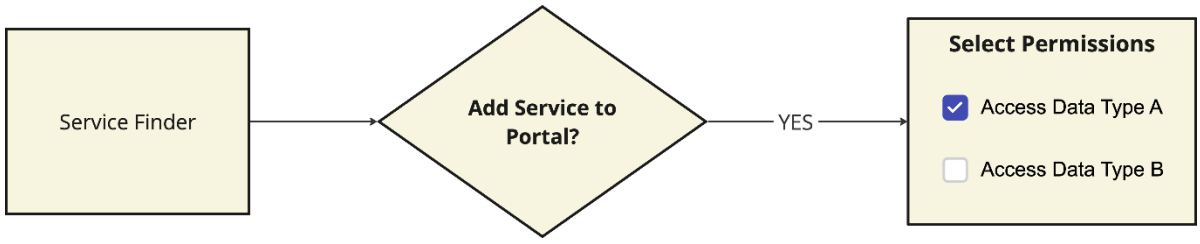


Figure 17: Opting In ("App Store model")

12.5.3 Opting-in: Permissions Manager

Continuing with the "App Store" concept, the Resident Portal would also offer residents a permissions management where a resident can view, edit, and revoke permissions they have granted to a Service or Individual in the case of delegation.

To help users manage permissions there may be regular reminders and notifications for them to review permissions. Some permissions could be set to expire if not renewed.

12.5.4 Opting-in: Use Resident Profile Option

Agencies in the Tier 3: Hybrid integration tier will continue to collect and store resident data on their own technology separate from the Resident Portal, continue to perpetuate a disjointed experience where common data (e.g. addresses, phone numbers, etc.) needs to be managed in multiple systems.

The Resident Portal could encourage residents to consolidate their data through convenience by providing agencies with an API or an embedded component that gives residents the option use their Resident Profile to apply for a service rather than entering the data manually, and the resident could be given the choice to choose whether to use their Resident Profile for some or all of their transaction.

Many possible front-end experiences can support this sort of option:

- Example: Password Keepers like LastPass can store a user's contact information, credit cards, and account details for many different websites. When a user visits a site with information stored in LastPass, they are given an option to have LastPass fill in the relevant fields. In this user experience, the data is entered at the push of a button but isn't directly shared from Last Pass with the website.
- Example: Many services give user the opportunity to choose one of several payment options. Like choosing between Apple Pay or entering your Credit Card information, residents could be presented with the option to "Use my Resident Profile" as a 1-click choice.

12.5.5 Opting-in: Saving data back to the Resident Portal

This exchange of data between agencies and WaTech can go both ways. The primary reason residents share their data with a government entity is because they are attempting to access a specific service. This is the ideal time to present the Resident Profile as a benefit and to encourage a resident to save the data they're entering to their profile to use in the future.

If using this experience, be explicit about which data is getting shared to their profile WaTech should also consider what data should and shouldn't get saved back to the Resident Profile; if other agencies don't need the resident's VIN number for their service, there is no reason to collect and save it to the Resident Profile (information stored within an agency's back-end can still be displayed within the Resident Portal using an API instead.)

12.5.6 Review and Submit

This is something WA state agencies already do generally, but Residents should always be presented with a "review" screen to review what's been entered so that they can verify its accuracy before it is submitted.

12.6 Adjust the Data Strategy Over Time

As trust in the portal increases or the state of WA state technology becomes less decentralized (federated) some aspects of this strategy will be replaced with an increasingly centralized solution. As these decisions are made, ensuring the collection and use of this data remains transparent to residents is critical to retaining resident trust.

13 Cost Projection

Cost projection for the one-time project cost and forecasted M&O over a 5-year period.

- WARP Resident Portal Future-State 5yt TCO 2024-10-15.xlsx

14 APPENDIX

14.1 REFERENCES

14.1.1 Market Scan Spreadsheet:

- WARP Market Scan - Competitive Capability & Feature - Final - 2024-10-14-.xlsx

14.1.2 Miro Board

- https://miro.com/app/board/uXjVKu5uzMo=/?share_link_id=652914647499

14.1.3 Cost Projection

- WARP Resident Portal Future-State 5yt TCO.xlsx

14.1.4 Visualization Deck

- WaTech - Service Integration Tier Walkthroughs DRAFT_v2-0_2024_10_10.pptx

14.1.5 Contentful Conversation:

- Contentful_Conversation_2024_10_03.mp4
- Contentful_Conversation_chat_2024_10_03.txt
- Contentful_Conversation_transcript_2024_10_03.transcript.vtt.txt

14.1.6 Jahia Demo

- Demo_ WaTech-Jahia_2024-10-01.mp4

14.1.7 Liferay Demo

- Demo_WaTech_Liferay_2024_10_03.mp4

14.2 APPENDIX A: CITATIONS

https://www.openai.com/chatgpt.
Technology Descriptions. Accessed September - October, 2024.
https://docs.alfresco.com/process-automation/latest/model/interfaces/
YouTube. “Liferay CoS | Stylebooks 201 - Goodbye Themes!” Accessed September 18, 2024.
https://www.youtube.com/watch?v=Pp4aZqAFCZ8
Jahai Academy. “About Jahia.” Accessed September 18, 2024.
https://academy.jahia.com/documentation/jahia/jahia-8/end-user/authoring-content-in-jahia/about-jahia
Jahai Academy. “Jahia Under the Hood.” Accessed September 18, 2024.
https://academy.jahia.com/documentation/jahia/jahia-8/developer/extending-and-customizing-jahia-ui/jahia-ui-under-the-hood#basic-design-system-layout-components
Contentful. “Experiences - What Are Experiences.” Accessed September 18, 2024.
https://www.contentful.com/developers/docs/experiences/what-are-experiences/
Vercel. “Strapi Design System.” Accessed September 18, 2024.
https://design-system-git-main-strapijs.vercel.app/?path=/docs/getting-started-welcome--docs
Strapi.io. “Development Docs - Templates.” Accessed September 18, 2024.
https://docs.strapi.io/dev-docs/templates
Nethues. “React vs Angular: Choose the Right Framework to Uplift UX” Accessed September 18, 2024.

https://www.nethues.com/blog/react-vs-angular/#:~:text=Angular%20is%20ideal%20for%20building,interfaces%20with%20frequently%20changing%20data
Radix. “React vs Angular: Which JS Framework to choose for Front-end Development?” Updated: Aug 31, 2024. Accessed September 18, 2024.
https://radixweb.com/blog/react-vs-angular
Radix. “React vs Vue: Comparison of the Best JavaScript Frameworks at a Glance!” Updated: Aug 28, 2024. Accessed September 18, 2024.
https://radixweb.com/blog/react-vs-vue
Alokai. “Vue vs React: Which one to choose in 2024?” June 20, 2024. Accessed September 18, 2024.
https://alokai.com/blog/vue-vs-react
Simplilearn. “G Flutter vs. React Native: Which One to Choose in 2024?” Accessed September 18, 2024.
https://www.simplilearn.com/tutorials/reactjs-tutorial/flutter-vs-react-native#what_is_react_native
TheDroidsOnRoids. Flutter vs. React Native – Which is Better for Your Project in 2024?” Accessed September 18, 2024.
https://www.thedroidsonroids.com/blog/flutter-vs-react-native-comparison
UXCam. “NET MAUI vs React Native - How to Choose the Right Framework.” June 3, 2024. Accessed September 18, 2024.
https://uxcam.com/blog/net-maui-vs-react-native/
Kanini. “.NET MAUI vs. Flutter vs. React Native: A Comprehensive Comparison” updated on: May 18, 2024. Accessed September 18, 2024.
https://kanini.com/blog/net-maui-vs-flutter-vs-react-native/
HelpCrunch. “Intercom vs Zendesk: Key Differences & Best Choice for 2024.” Aug 13, 2024. Accessed September 18, 2024.
https://helpcrunch.com/blog/zendesk-vs-intercom/#:~:text=Zendesk%20vs%20Intercom%3A%20functionality,-The%20Intercom%20versus&text=However%2C%20they're%20crafted%20for,support%20representative%20can%20dream%20about
Zapier. “Zendesk vs. Intercom: Which is better?.” February 10, 2023. Accessed September 18, 2024.
https://zapier.com/blog/zendesk-vs-intercom/
G2. “Compare Intercom and Zendesk Support Suite.” Accessed September 18, 2024.
https://www.g2.com/compare/intercom-vs-zendesk-support-suite
Customerly.io. “Intercom vs Drift: comparison between the famous live chat tools.” Sep 16, 2024. Accessed September 18, 2024.
https://www.customerly.io/blog/intercom-vs-drift/#:~:text=Drift%20and%20Intercom%20are%20two,Drift%20is%20more%20sales%2Dfocused
G2. “Compare Drift and Intercom.” Accessed September 18, 2024.
https://www.g2.com/compare/drift-vs-intercom
HelpCrunch. “Drift vs Intercom: Which Tool is Better?” Sep 18, 2023. Accessed September 18, 2024.
https://helpcrunch.com/blog/intercom-vs-drift/
Greyling, Cobus. Medium. “A Review of Nuance Mix Conversational AI Platform.” April 12, 2022. Accessed September 19, 2024.
https://cobusgreyling.medium.com/a-review-of-nuance-mix-conversational-ai-platform-4c822b535676
Microsoft. Learn. “Tutorial: Voice-enable your bot.” August 28, 2024. Accessed September 19, 2024.
https://learn.microsoft.com/en-us/azure/ai-services/speech-service/tutorial-voice-enable-your-bot-speech-sdk
George, Arun. UX Design. “Designing Voice Experience.” September 28, 2017. Accessed September 19, 2024.
https://uxdesign.cc/voice-user-experience-design-and-prototyping-for-mere-mortals-ef080c843640
George, Arun. UX Planet. “UX Design Process for Chatbots — A Crash Course.” October 9, 2019. Accessed September 19, 2024.
https://uxplanet.org/design-process-for-chatbots-under-5-minutes-98f17d925449

Microsoft. Learn. “Tutorial: Voice-enable your bot.” August 28, 2024. Accessed September 19, 2024. https://learn.microsoft.com/en-us/azure/ai-services/speech-service/tutorial-voice-enable-your-bot-speech-sdk
MyTN screenshots of Booking a Campsite. MyTN. October 2024. https://www.mytn.gov/
The Lodges at Tennessee State Parks. Tennessee state Parks. October 2024. https://tnstateparks.com/lodges