WEB BASED ANNUAL REPORTING SYSTEM

WASHINGTON STATE
HOUSING FINANCE
COMMISSION

Opening doors to a better life

intellectyx

Table of Contents

INTRODUCTION

This document has been created to outline the current system design and architecture for Web Based Annual Reporting System by Intellectyx. This product has been developed as a web-based annual reporting solution that includes the portals of multifamily, & homeownership. The system allows on-site manager, property managers and owners to submit the occupant/loan details to their funders on yearly basis.

Revision History

All suggestions, comments, and requests for changes to this document should be directed to the document owner.

Date of Revision	Section Revised	Nature of Revision

Stakeholder Approvers

The following are the Stakeholder Approvers of this document.

Stakeholder Approvers	Representing Area	Date Approved

Reviewers

This section is used to track the people who review this document to verify its completion.

Reviewers	Representing Area
Lily Caroline	

PURPOSE

The purpose of the document is to explain the current system design and technical architecture of WBARS which facilitates the reporting submission with funders.

This document intends to help the reader determine how the system has been structured at the highest level.

PROJECT SCOPE

Washington State public funders have implemented an online reporting system, WBARS, used by owners and managers to report data from their affordable multifamily rental projects. Currently several major public funders have adopted the system and participation is growing throughout Washington State. The public funders include Washington State Housing Finance Commission, State Department of Commerce (Housing Trust Fund), City of Seattle, King County, Snohomish County, City of Tacoma, and City of Spokane. The system satisfies program and contract reporting requirements and tracks most of the main components of project performance (operation indicators, income, expense, status of reserves, tenancy, etc.)

The Web-Based Annual Reporting System (WBARS) is the way to submit a Combined Funder Annual Report (Table 1,2,3,4) along with a Report Cover Sheet.

- Report Cover Sheet – General report information & and report overview text from contractors.
- Table 1 – Resident/Household details (move-in, cert dates, income, rent household size)
- Table 2 & 3 – Demographics, special need counts for the year.
- Table 4, 4(a) & 4(b) – Financial data for the calendar year (project income, expenses, reserves)

Each Report Table is submitted separately. A history of all submittals is tracked in the system and viewable on the Report Cover Sheet. The levels of the Hierarchy are:

- On-Site Managers (if applicable) - submit to the PROPERTY MANAGER
- Property Managers—submit to the CONTRACTOR/OWNER
- Contractor/Owners—submit to the FUNDER.

ARCHITECTURE

The current system is implemented in a MERN stack-based system with support in handling real-time tasks like notifications, application tracking, etc. The system is also extended with REST APIs for both backend and partner calls.

DEFINITIONS

Tech Stack	Description
React.js	An open-source web application library maintained by Facebook.
Redux	Redux is an open-source JavaScript library for managing application state. It is most used with libraries such as React for building user interfaces.
Material UI	Material UI is a free and open-source front-end library for designing websites and web applications. It contains React Component-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional third party extensions. Unlike many web frameworks, it concerns itself with front-end development only.

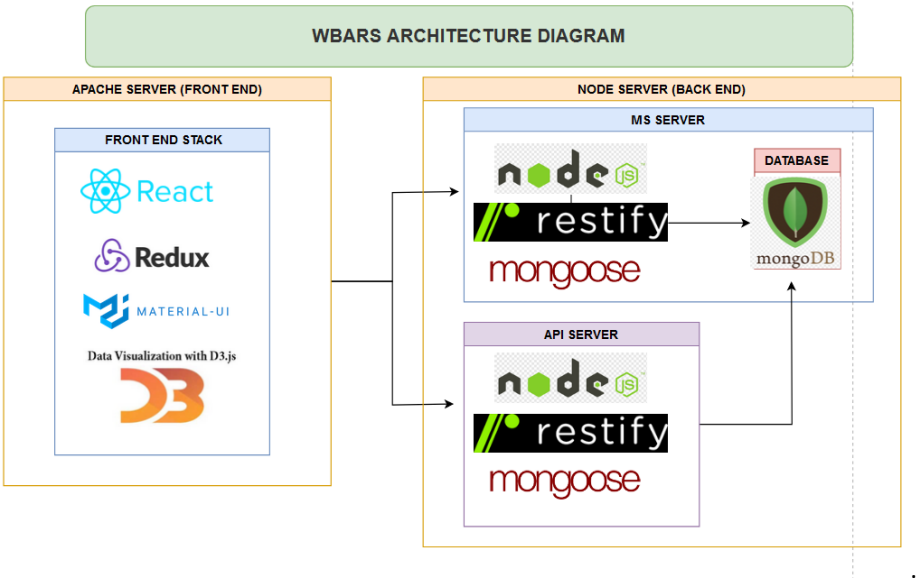
Node.js	A popular open-source JavaScript-based framework/platform built on Google Chrome's JavaScript V8 Engine.
D3.js	D3.js is a JavaScript library for producing dynamic, interactive data visualizations in web browsers. It makes use of the widely implemented SVG, HTML5, and CSS standards.
MongoDB	MongoDB is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas.
Webpack	Webpack is an open-source JavaScript module bundler. Webpack takes modules with dependencies and generates static assets representing those modules.
Babel	Babel is a JavaScript compiler
Microservices	It is an architectural style that structures an application as a collection of loosely coupled services, which implement business capabilities.
Jest	Jest is a JavaScript testing framework designed to ensure correctness of any JavaScript codebase. It allows you to write tests with an approachable, familiar and feature rich.

ABBREVIATIONS

Abbreviations	Description
JSON	JavaScript Object Notations
SDK	Software Development Kit
UI	User Interface
URL	Universal Resource Locator
API	Application Program Interface
MVC	Model View Controller
HTTPS	It is a communications protocol for secure communication over a computer network, website, Internet or URL
TCP/IP	A computer networking model and set of communication protocols used on the internet and similar computer networks, including the Transmission Control Protocol (TCP) and the Internet Protocol (IP).
AWS	Amazon Web Services
CI/CD	Continuous Integration / Continuous Deployment

APPLICATION ARCHITECTURE

Below is a high-level overview of current application architecture.

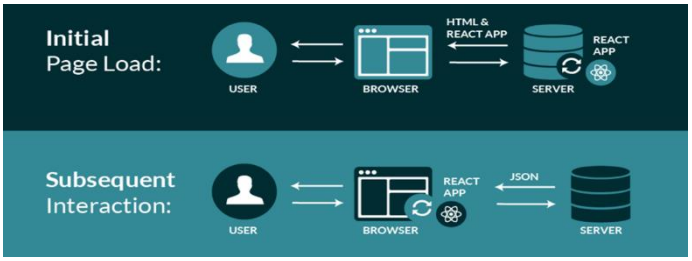


The current architecture is designed for leveraging the following capabilities –

- Portability and reusability of code across channels
- Versatility of changing the theme and look and feel of the application with minimal changes to the backend/architecture.
- Clean separation between front end and backend technologies
- Faster response time by maximizing the use of cache and minimizing the amount of the data transferred through the network.
- Ease of replacement for each layer in the architecture as the technology changes

CLIENT – SERVER ARCHITECTURE FLOW

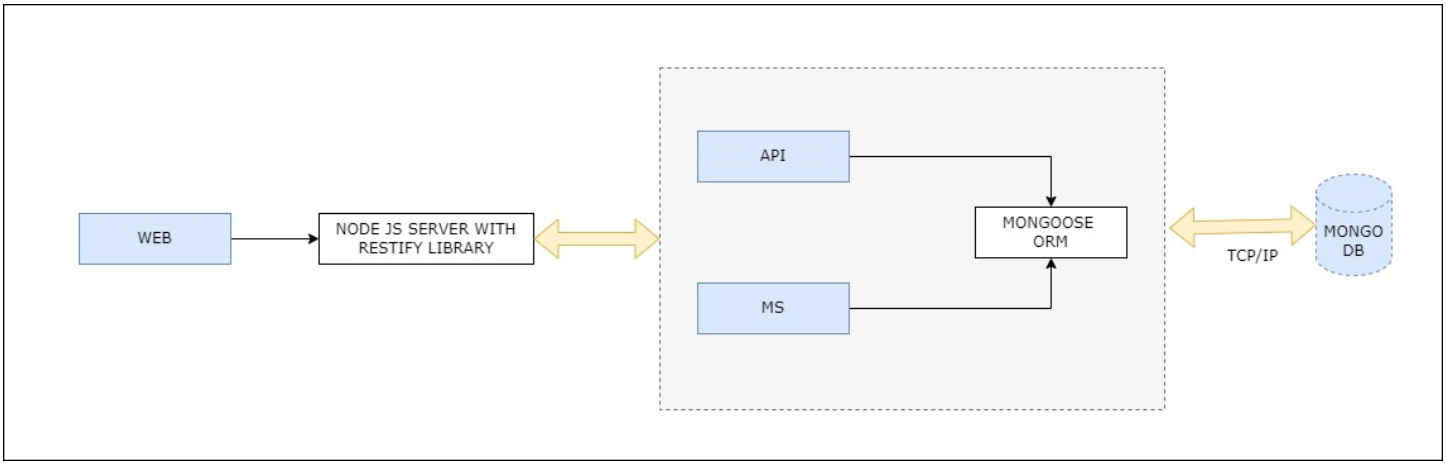
- All request to the frontend is served from the Rest API Web Service.
- React JS and Material UI is responsible for rendering the template/pages in the frontend.
- The pages contain a dynamic transactional data (remote read) and static information that is rendered on the local.
- Services acts as a client-side API responsible for accessing the API and feeding it to the store and in turns to the store again.
- Store refresh the view for update the data while receiving data from the server.
- Each component contains specific events and variables matched with the context of the component.
- CSS is used to theme the UI Component. At the very high level a generic style guide is applied across the components, however each individual component has its own stylesheet.



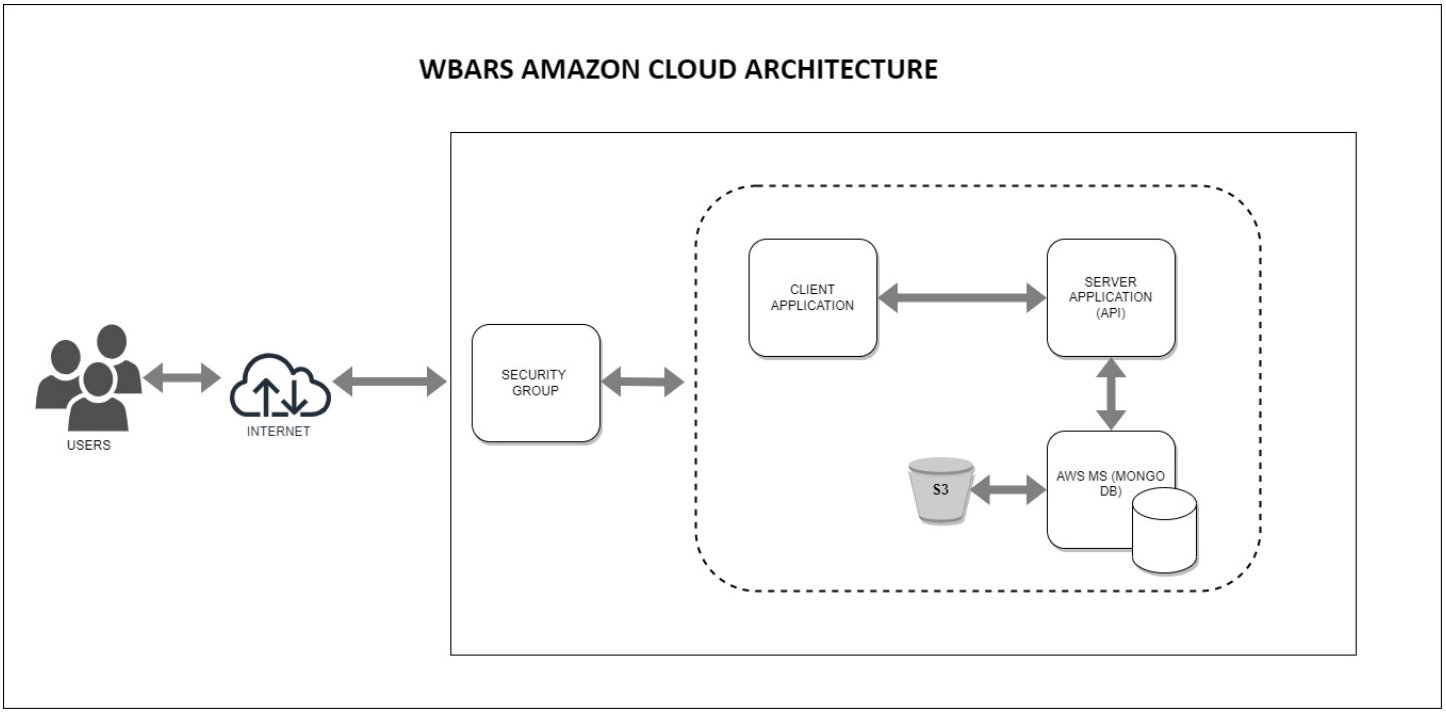
All CSS, UI Components and Service Module are cached in the browser cache for performance and optimization.

API LAYER – NODE JS + RESTIFY

As an asynchronous event driven JavaScript runtime, Node has designed to build scalable network applications. Restify.js Framework development is used because of its flexibility, simplicity, extensibility, and performance. The software is built with Server side and with Restify.js and Node.js together, the application is built with APIs for single-page, multi-page, hybrid mobile and web apps support. Mongoose has been used as a Mongo Driver with Restify.js



CLOUD ARCHITECTURE



A three-tier architecture with AWS and Apache server in the front end is used with separating the application into three logical layers: presentation, application logic, and data storage. Here's a breakdown of each tier along with how AWS services and Apache server can be utilized:

Front-end layer – The Front layer is constructed using ReactJS. This layer provides the interactive UI to the web portal, and it is separated from the server layers, this separation will allow the application to have multiple interfaces (or views). The application state is managed through Redux for ReactJS. Redux action delegates request/response processing to the service layer.

API SERVER Layer - Modules in the assistance layer are answerable for business rationale, taking care of assignments, for example, clients the board, information the executives, occasions dealing with, and so on, A module in the help layer processes the REST API and solicitations the comparing administration in the help layer for managing the mentioned task.

MS SERVER Layer - The server layer is developed considering NODE JS and MONGODB is used as an information base. The server additionally gets a sense of ownership by overseeing content, clients, and notices. The JWT (JSON Web Token) is dealt with the verification and approval.

The above-mentioned instance types & architecture can handle multiple concurrent requests with security applied across multiple levels around Data protection, Identity and Access management, Infrastructure Protection and Threat detection, continuous monitoring.

SECURITY FEATURES

The current platform features help to control data access, operations, and development within WBARS Profile. Key capabilities include:

- Authorization: This is the most widely recognized situation for utilizing JWT. When the client is signed in, each resulting solicitation will incorporate the JWT, permitting the client to get to courses, administrations, and assets that are allowed with that token.

- Information Exchange: JSON Web Tokens are used to safely communicating data between parties. Since JWTs can be marked — for instance, utilizing public/private key matches — you should rest assured the senders are who they say they are. Furthermore, as the mark is determined utilizing the header and the payload, you can likewise check that the substance hasn't been altered. Authentication with accounts:

- Access control with privilege sets: You define permissions that determine levels of access to your custom app. You can define as many privileges sets as needed.

- Data encryption on the Network: Its enabled with HTTPS connection to secure network transport between client and server requests. The system offers the ability to add a layer of security to your data at rest in the cloud, providing scalable and efficient encryption features.

- Server monitoring and administration: Logging and Notification has been configured and enabled on the server side. We receive the email notifications on any error that happens on the server.

- Monitoring tools record performance statistics over time so that usage patterns can be identified. Monitoring agents record selected metrics at set intervals and store the resulting data in a time-series format.

- Nagios is the monitoring tool that has been utilizing to monitor the WBARS system the following metrics in the server –

    - Current Load
    - Current Users
    - MongoDB Connect

- Root Partition
- Total Processes
- Memory Usage
- Rooy Partition
- Total Process
- Zombie process
- Total Process
- Website Status
- SSL Monitoring

- Tracing records system events, such as an HTTP request from a client. In distributed tracing, details captured about the event include the path of the request across multiple services/applications, along with metrics about the request such as latency at each step of the way.

AWS SECURITY

Data on the AWS platform is always encrypted in transit. The application has leveraged the built-in tools provided by AWS to create an encrypted file system that encrypts all data (including documents) and metadata at rest using an industry-standard AES-256 encryption algorithm. An encrypted file system has designed to handle encryption and decryption automatically and transparently. AWS's secure network also has built-in mechanisms to protect against distributed denial-of-service (DDoS) attacks. It isolates networks, ensures the confidentiality of data, and actively works to combat DDoS attacks to ensure such attacks do not bring down AWS services.

AWS MONITORING:

- CPU utilization
- Network in (bytes)
- Network out (bytes)
- Disk Read
- Disk Write
- CPU Credit Usage

In addition to taking advantage of the basic DDoS protections automatically enabled in the platform, we have utilized the built-in service, *AWS Shield* which provides always-on detection and automatic inline mitigations that minimize application downtime and latency. Using the AWS Shield Standard, we have defended our web application against the most common, frequently occurring

network and transport layer DDoS attacks. The AWS Shield along with Amazon CloudFront and Amazon Route 53 provides comprehensive protection against all known infrastructure (Layer 3 and 4) attacks.

LANGUAGES SUPPORTED

Portal	**Languages**
WBARS	Worldwide English.

BROWSER & COMPATIBILITY

The WBARS portal can support the following browsers with current versions and any high-level versions.

S. No	Browser Name
1	Safari 17.3 minimum
2	Chrome 121 minimum
3	Microsoft Edge 121
4	Firefox 122