



### About the Client

Builds systems and solutions pertaining to the property market and land ownership in British Columbia (BC), Canada

Is responsible for easing the process of enforcing statutory laws by digitizing the process and workflow with software that has user-friendliness and security



### Business Needs

Create a landowner transparency registry for residents in British Columbia

Provide robust security for the application, infrastructure, and secure PII handling at rest and transit

Implement a reduction of property prices in BC with the help of a systematic registry

Implement automatic detection of security vulnerabilities and threats in the application and on the libraries

Manage the IT system for the network

Manage fault tolerance by developing key business workflows so that application functions smoothly despite a failure

Automate scripts to benchmark and measure performance at regular intervals

Adjusted capacity to server traffic based on load scalability

Maintain IaC infrastructure as a code

Configuration and management of alerts on CloudWatch

#### SUPPORT REQUIREMENTS

Oversee requirements for Elasticsearch

Monitor health of the servers and applications on AWS Cloud

Create and manage the CI/CD pipeline



### Our Solutions

#### TECHNICAL MECHANISM

Performed infrastructure automation using Terraform and Ansible

Designed microservices (with its dedicated database) that evenly distributed heavy traffic functionality while having less service-to-service communication

Created an end-to-end project with microservices and event driven architecture running on Kubernetes in AWS Cloud

Created infrastructure on AWS with fine grain security

Designed fault-tolerant critical business workflows using a messaging system

Managed Enterprise search using Elasticsearch with zero-downtime and high parallel (with RabbitMQ and POD auto-scaling) bulk data sync

Implemented highly scalable services and APIs

Elasticsearch indexes are designed to achieve high performance with less data duplication

Implemented CI/CD with highly scalable job runners in GitLab

Set up a branching strategy to support parallel development by teams spread across the globe



Handled the back-office application via AWS Cognito with role-based access rights

Built a self contained system (SCS) for automated enterprise grade microservice

Added API design in SwaggerHub to easily drive API strategy across multiple integrators

Administered the database with auto-rotating credentials at runtime

#### SUPPORT ACTIVITIES

Ensured PII is tokenized and encrypted to a separate store at rest and transit with provision to change encryption key on a need basis

Sent a daily shift report

Sent monthly status reports

Actively handled tickets, alerts, and critical ticket updates

### Technology Stack

#### OPERATING SYSTEMS



#### SCRIPTING



#### MONITORING TOOLS



#### CLOUD



#### SECURITY



#### DEVOPS TOOLS



#### SERVER ENVIRONMENT



#### FRAMEWORK/LANGUAGE/TOOLS



#### ARCHITECTURE

Microservices based architecture with event driven workflows

### Business Impact

Created the landowner transparency registry from scratch in a **short time**, thus ensuring **no delays or penalties**

Built a **one-click automatic system** to push any new release seamlessly

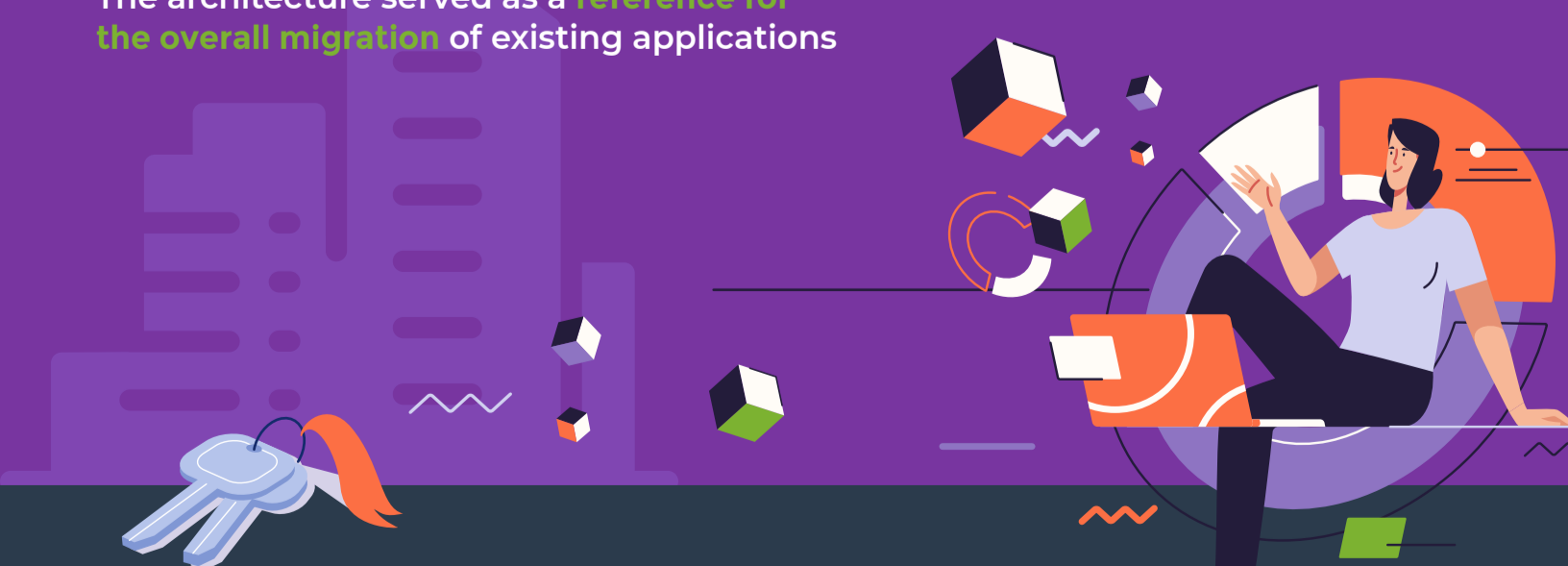
Currently **maintaining 2 registries** for smooth operations

Effectively **carried-out synchronization** between 2 registries smoothly

**Reduced bug leakage** significantly

Effectively **lowered cost and usage** for **AWS dev account**

The architecture served as a **reference** for the **overall migration** of existing applications



### Cybage Cloud Solutions



End-to-end management for smooth process flow and enhanced productivity on the cloud network