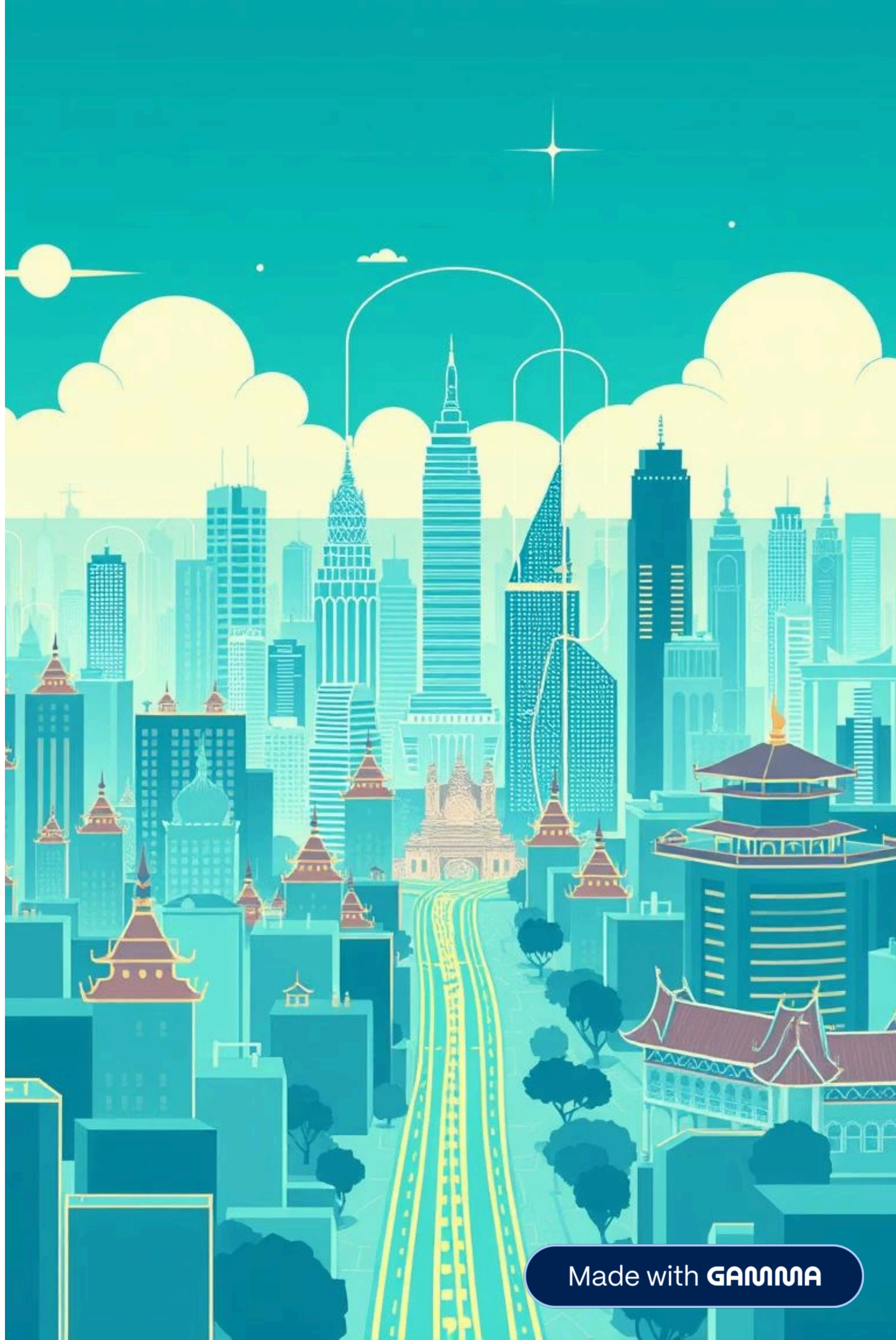
# Population Management System – Full Platform Architecture

Empowering Cambodia's Future with a Modernized Digital Identity Platform



# Project Overview: Building Cambodia's Digital Foundation

This national platform aims to modernize citizen identification and service delivery.

## Purpose

Establish a robust, secure, and efficient digital population management system for Cambodia.

## Scope

Comprehensive system covering citizen-facing services and registrar operations, integrating with national databases.

## Key Objectives

- Enhance data accuracy and integrity.
- Improve accessibility for citizens.
- Streamline administrative processes.
- Ensure high levels of security and privacy.

# Frontend Architecture: Engaging Citizens and Registrars

Our frontend is designed for both public accessibility and robust registrar functionality.

## Public Site (Citizen-Facing)

- **Technology:** Next.js
- **Features:** Information dissemination, service requests, status checks.
- **Goal:** User-friendly interface for all citizens.

## Registrar Portal

- **Technology:** BlazorWasm
- **Functionality:** Offline drafting, bilingual UI support (Khmer & English).
- **Goal:** Empower registrars with efficient data management tools.

# Backend Architecture: The Powerhouse of the Platform

A multi-language backend ensures scalability, performance, and specialized task handling.

### Node.js TSX API

**Core API:** Handles JWT authentication and multi-tenancy for secure data access.

### PHP 4.x Native

**Legacy Integration:** Manages existing critical functionalities and data.

### Python 3.12.7

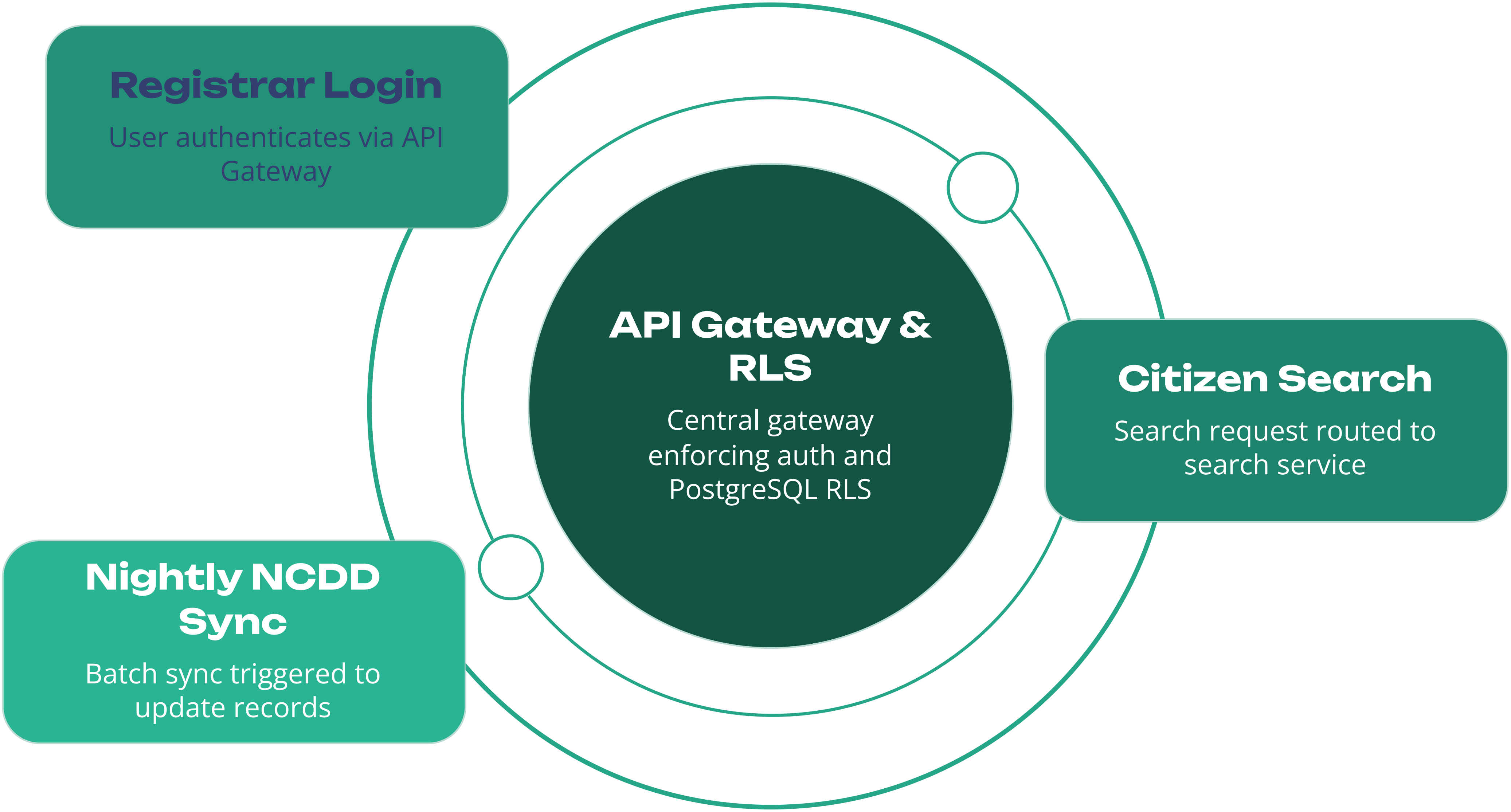
**Specialized Tasks:** NCDD synchronization, data cleaning, and PDF generation.

### BlazorServer API .NET 10

**Real-time:** Role management and real-time communication via SignalR.

# Runtime Data Flows: Ensuring Seamless Operations

Data moves securely and efficiently through the system, guided by robust protocols.



### Registrar Login

User authenticates via API Gateway

### API Gateway & RLS

Central gateway enforcing auth and PostgreSQL RLS

### Citizen Search

Search request routed to search service

### Nightly NCDD Sync

Batch sync triggered to update records

# Component-to-File Mapping: Navigating the Codebase

Understanding the project structure is key to efficient development and maintenance.

## Frontend Files

```
/frontend
├──── public/
│    ├──── index.html
│    └──── assets/
├──── src/
│    ├──── components/
│    ├──── pages/
│    │    ├──── citizen/
│    │    └──── registrar/
│    ├──── App.js
│    └──── index.js
└──── package.json
```

## Backend Files

```
/backend
├──── node-api/
│   ├──── src/
│   │   ├──── controllers/
│   │   └──── services/
│   └──── package.json
├──── php-legacy/
│   ├──── public/
│   └──── src/
├──── python-scripts/
│   ├──── ncd_sync.py
│   └──── pdf_gen.py
└──── blazor-api/
├──── Controllers/
└──── Services/
```

# Onboarding Value: Accelerating Adoption and Impact

A well-structured platform architecture delivers tangible benefits for teams and stakeholders.

### Unified Architecture

Provides a coherent system landscape, reducing complexity and increasing interoperability.

### Reduced Onboarding Time

New developers and partners can quickly understand and contribute to the project due to clear structure.

### Improved Maintainability

Modular design allows for easier updates, bug fixes, and feature enhancements.

### Enhanced Security

Consistent security practices across all components protect sensitive citizen data.

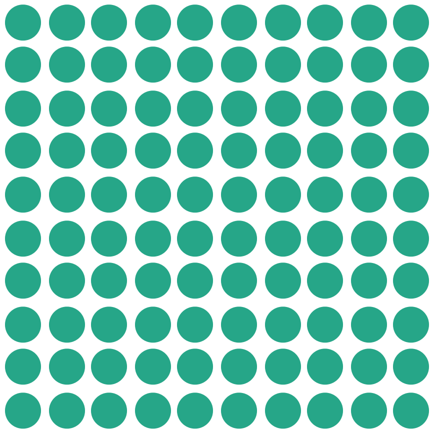
# Security Measures: Protecting Citizen Data

Data protection is paramount, implemented through multiple layers of security.

- **JWT Authentication:** Secure access tokens for all API interactions.
- **Role-Based Access Control (RBAC):** Granular permissions ensure users only access necessary data.
- **PostgreSQL Row-Level Security (RLS):** Enforces data isolation directly at the database level.
- **Data Encryption:** All sensitive data is encrypted at rest and in transit.
- **Regular Audits:** Ongoing security assessments and penetration testing to identify and remediate vulnerabilities.

# Scalability and Performance: Growing with Cambodia's Needs

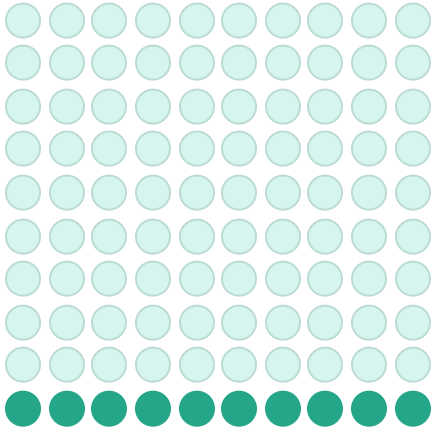
The platform is engineered to handle increasing demand and evolving requirements.



## 99.9%

### Uptime Target

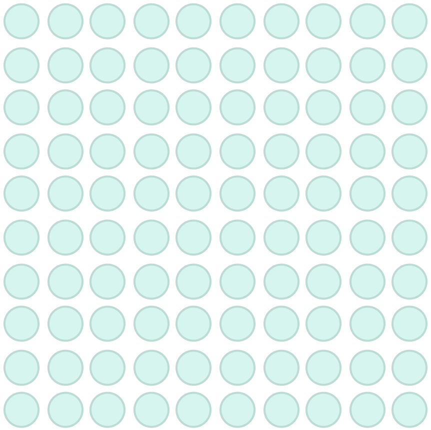
Ensuring continuous availability for critical services.



## 10K+

### Concurrent Users

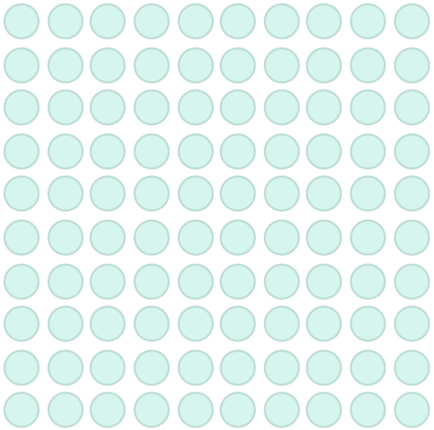
Designed to support high volumes of simultaneous interactions.



## Modular

### Architecture

Allows independent scaling of components to meet specific load demands.



## Optimized

### Databases

PostgreSQL RLS contributes to efficient query execution.

# Key Takeaways and Next Steps

## Key Takeaways

- **Robust Foundation:** Modern architecture supports national digital transformation.
- **User-Centric Design:** Tailored for citizens and registrars alike.
- **Secure & Scalable:** Built for long-term reliability and growth.
- **Collaborative Future:** Designed for easy integration and ongoing development.

## Next Steps

- **Pilot Program:** Initiate pilot testing in selected regions.
- **Training Workshops:** Conduct comprehensive training for registrars.
- **Feedback Integration:** Establish channels for continuous improvement.
- **Phased Rollout:** Plan for nationwide implementation.