AGN-DB

Sebastian Gonzales-Portillo, Carlos Rodriguez,
Ethan Lichtblau, Zain Khalid

WHAT IS AGN-DB?

- AGN-DB is a comprehensive catalog containing millions (~7.5 million) of data points about supermassive blackholes and active galactic nuclei, crucial for astrophysics research.
- Originally created by Astrophysicists and computer scientists at the University of Miami and Yale University in collaboration with NASA and the NSF.

WHY IS AGN-DB IMPORTANT?

- Provides a centralized location for astrophysical data.
- Facilitates collaboration and data sharing among research.
- Used to drive data analysis, research publications, and discovery.

Current System Issues:

- The website deteriorated over time and became non-functional due to legacy code, lack of documentation, and poor programming practices.
- “Database” schema is highly impractical with zero relational database design fundamentals, making it hard to scale or optimize queries.
- Performance bottlenecks in DB querying and potential security vulnerabilities.

Project Approach:

- Fix, update, and optimize existing codebase, implementing modern technologies when applicable.
- Implement best practices in documentation, testing, and security.
- Containerize with Docker for consistent deployment.
- Leverage MariaDB queries for high-performance data retrieval.

FUNCTIONAL REQUIREMENTS, USER FACING

01

Query Interface Implementation

- Allows Researches to input various parameters such as coordinates, measurement types, date ranges, to find astrophysical data
- Validates user input to prevent errors or malicious queries
- Handles large datasets efficiently

02

Data Download Functionality

- Enables users to export queried data in different formats (CSV, JSON, FITS, etc.)
- Ensures data integrity and seamless researcher workflow

03

User Interface Navigation

- A consistent, intuitive interface design that organizes main sections like Query, Results, Downloads, Documentation, and Admin Console
- Minimizes user confusion and error-prone action

FUNCTIONAL REQUIREMENTS, ADMIN FACING

01

Database Administration Controls

- Secure login and restricted access
- Database maintenance tasks like index rebuilding and performance checks
- Real-time data about DB resource usage and performance

02

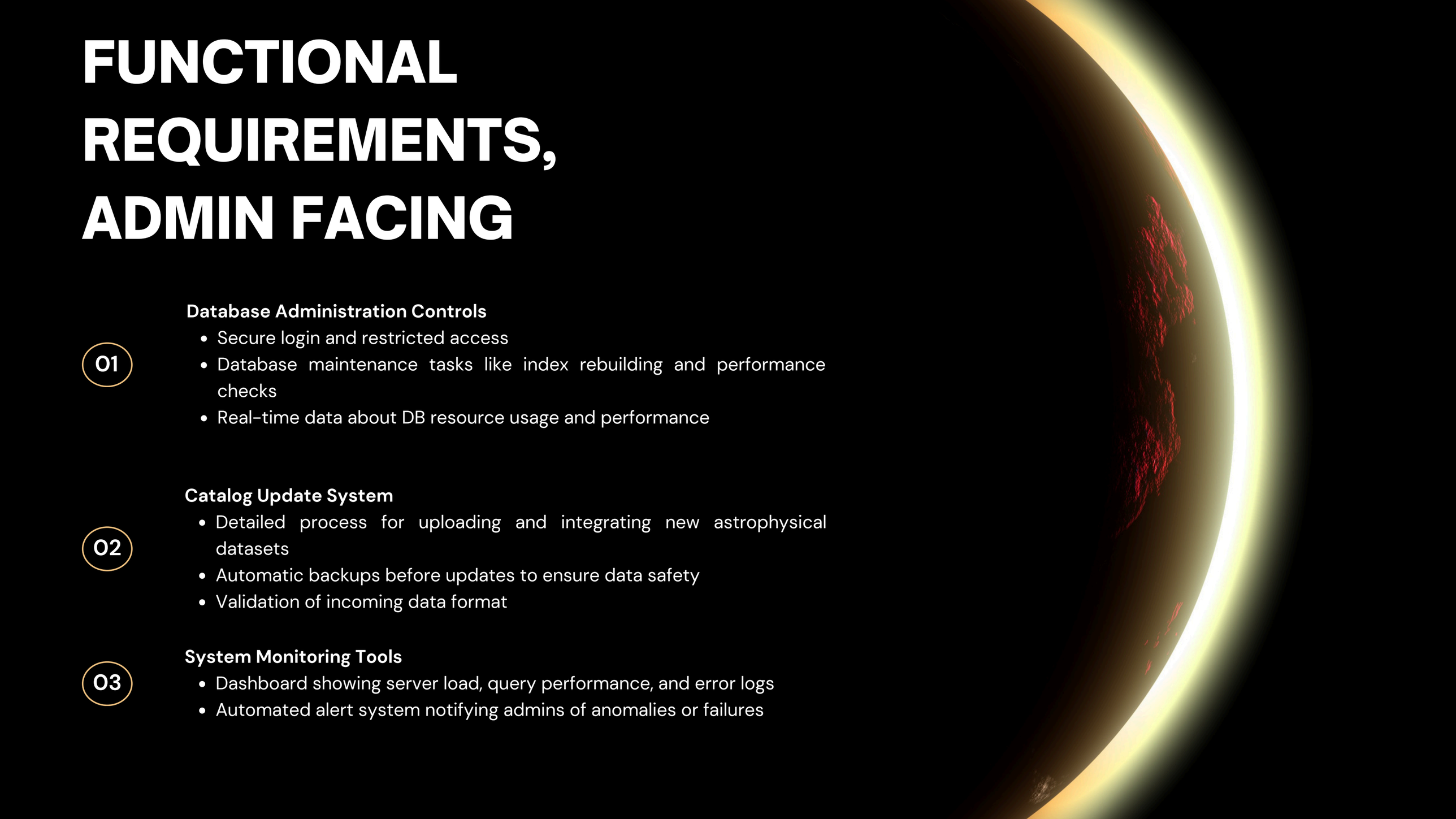
Catalog Update System

- Detailed process for uploading and integrating new astrophysical datasets
- Automatic backups before updates to ensure data safety
- Validation of incoming data format

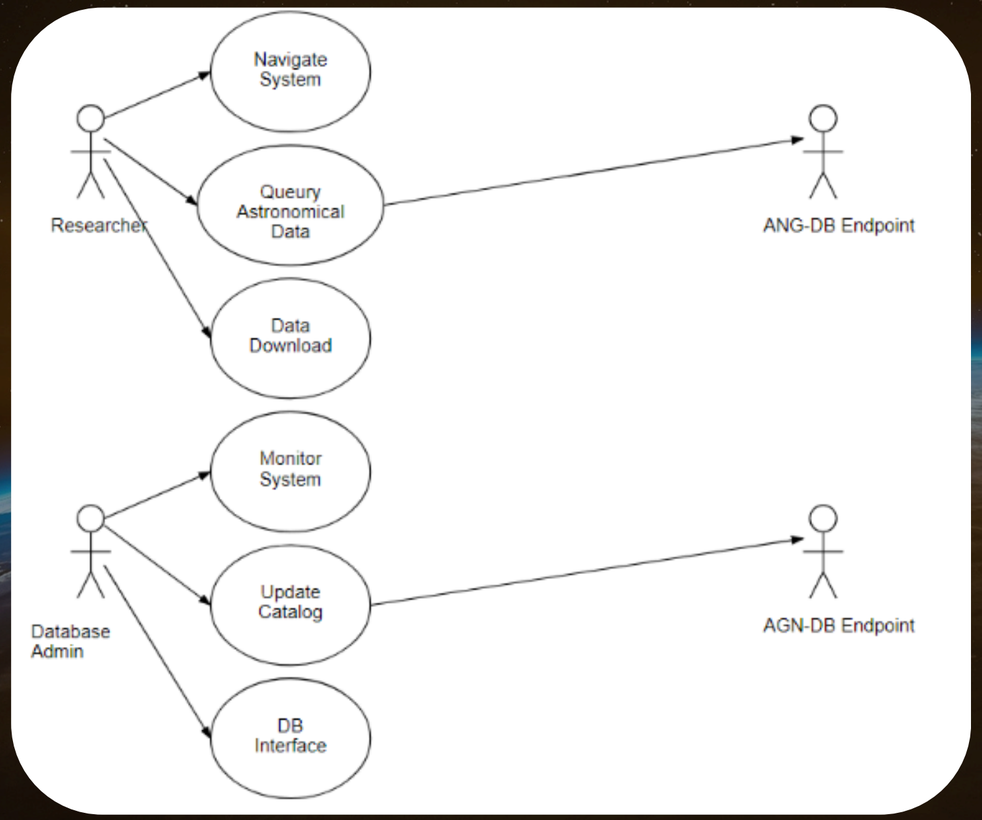
03

System Monitoring Tools

- Dashboard showing server load, query performance, and error logs
- Automated alert system notifying admins of anomalies or failures



USE-CASES



- **Navigate System:** provides an intuitive interface design with organized sections.
- **Query Astrophysical Data:** implements the query interface with parameter inputs and validation.
- **Data Download:** enables export functionality with multiple format options.
- **Monitor System:** delivers real-time performance data and automated alerts to the user.
- **Update Catalog:** manages the process for integrating new datasets with validation.
- **DB Interface:** handles secure access and database maintenance tasks.

NON-FUNCTIONAL REQUIREMENTS

DATABASE PERFORMANCE

- Queries must return results rapidly, even with millions of rows
- Indexing, caching, and optimized schema

SECURITY IMPLEMENTATION

- Prevent SQL injection via prepared statements and input sanitization
- Employ secure authentication/authorization frameworks for all roles

SYSTEM AVAILABILITY

- Target high uptime to serve an international research community
- Deploy redundancies or fallbacks servers if possible

NON-FUNCTIONAL REQUIREMENTS (CONTINUED)

ADMINISTRATIVE ACCESS & SECURITY

- Multi-factor authentication for critical accounts
- Role-based access control to separate read/write privileges

LOGGING & AUDIT TRAILS

- Keep logs of all changes for accountability
- Store logs in a secure location, maintain them for required duration

BACKUP & RECOVERY

- Automatic backups scheduled daily/weekly
- Documented procedures for restoration

SYSTEM CONSTRAINTS

Technical Constraints:

Frontend Framework

React.JS

Middleware Framework

Express.JS

Database Management System

MariaDB (MySQL fork)

Network Constraints:

Security Implementation

Administrative Access Security

SQL Injection Prevention

ACoRN Network Integration

Platform/Deployment Constraints:

Server Infrastructure:

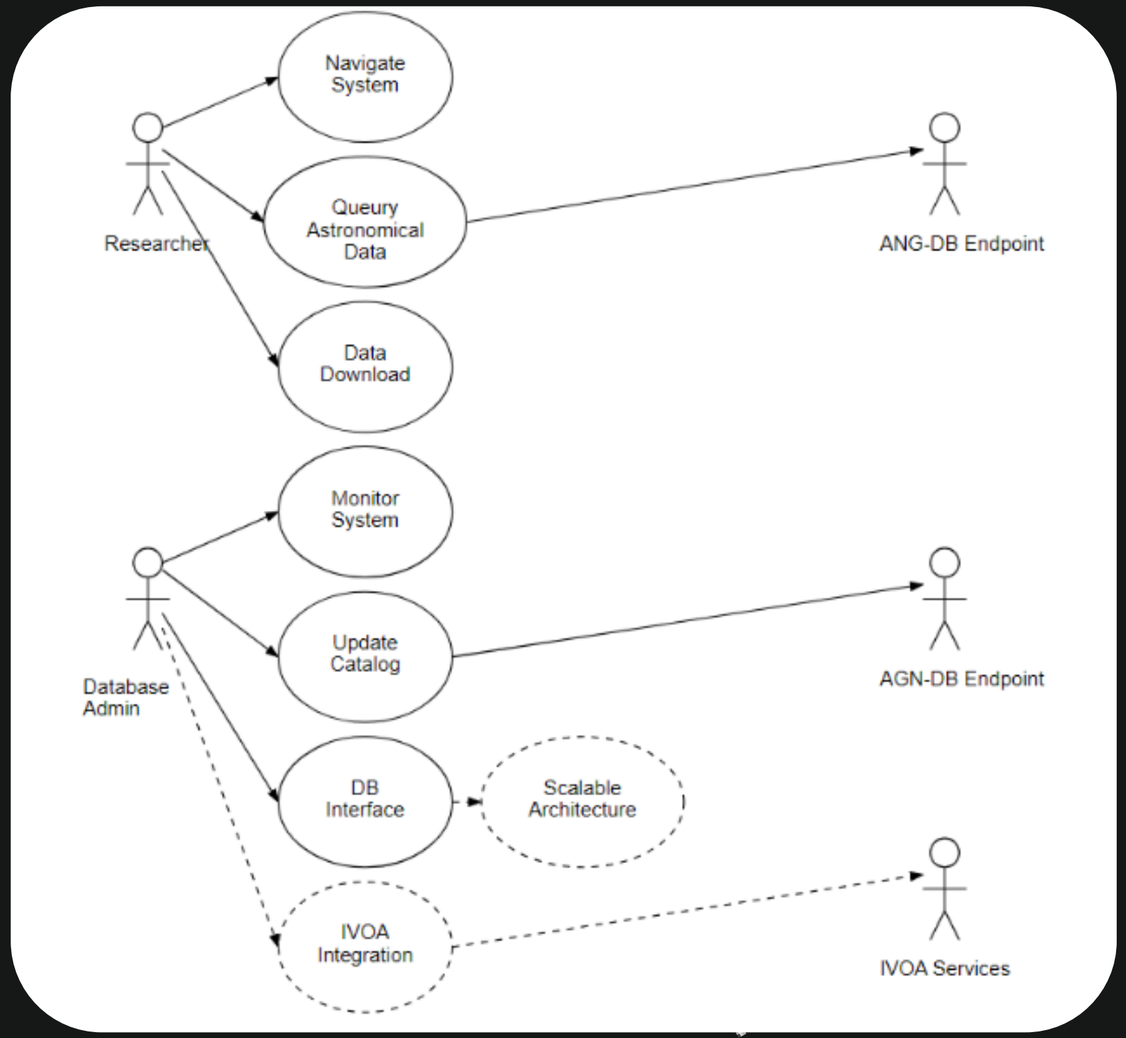
Maserati server Deployment

Docker Containerization

Evolutionary Requirements

01 Scalable Data Architecture

02 IVOA Integration



THANK YOU!