

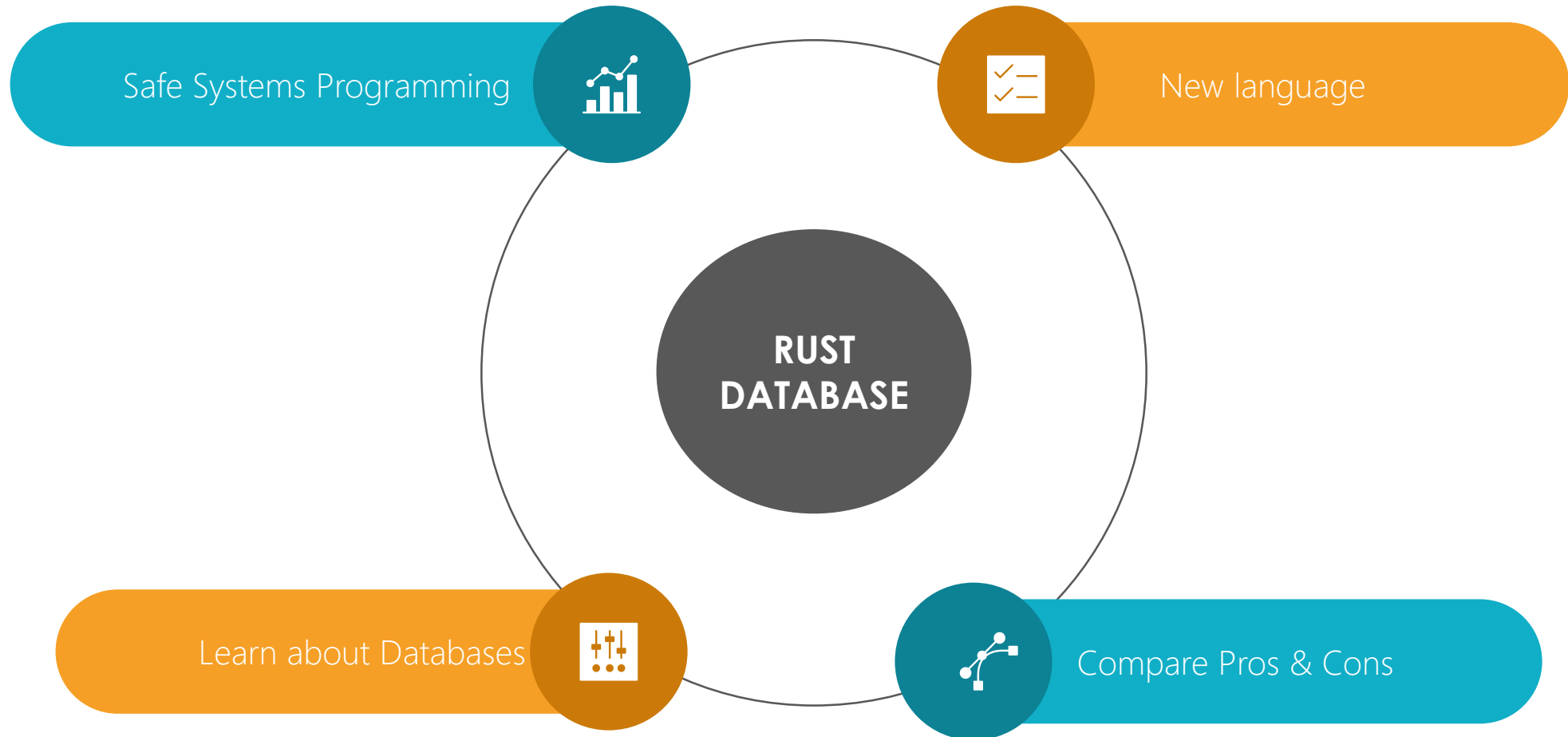


# Building a Database in Rust

John McMenemy

Supervisor: Robert Atkey  
Second Marker: Daniel Thomas

# Project Aims & Objectives



# Related Work

## ROCKSDB

Provided insight on database structure and design: how to manage the RAM, storage and using a log-file.

## ACID & TRANSACTIONS

Learned about database keywords and language. Understood what ACID standards are and transactions.

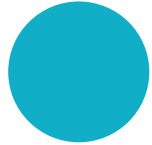
## LOG STRUCTURE DESIGN

An interesting database design choice where data is never overwritten on the disk. It is one of many structural designs.

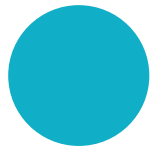
## BTREES

A data structure that minimizes disk access time. It is important in the context of databases as speed is critical (if reliability isn't affected).

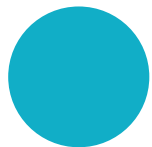
# Specification



Being able to access the database through a network.



Being able to store, modify and delete values through typed commands.



Being able to get a list of values and keys in the database.



Have measures in place to prevent data loss (crash, reboot etc.)



Have multiple concurrent users.



Being able to access the database through the web.

# Progress, Testing & Evaluation

---

As of now I have a database that can support concurrent users and have a working value store operation and key listing. It is tested through the Windows Telnet service.

At the moment, testing is done by myself through my machine and while demonstrating it to my supervisor.

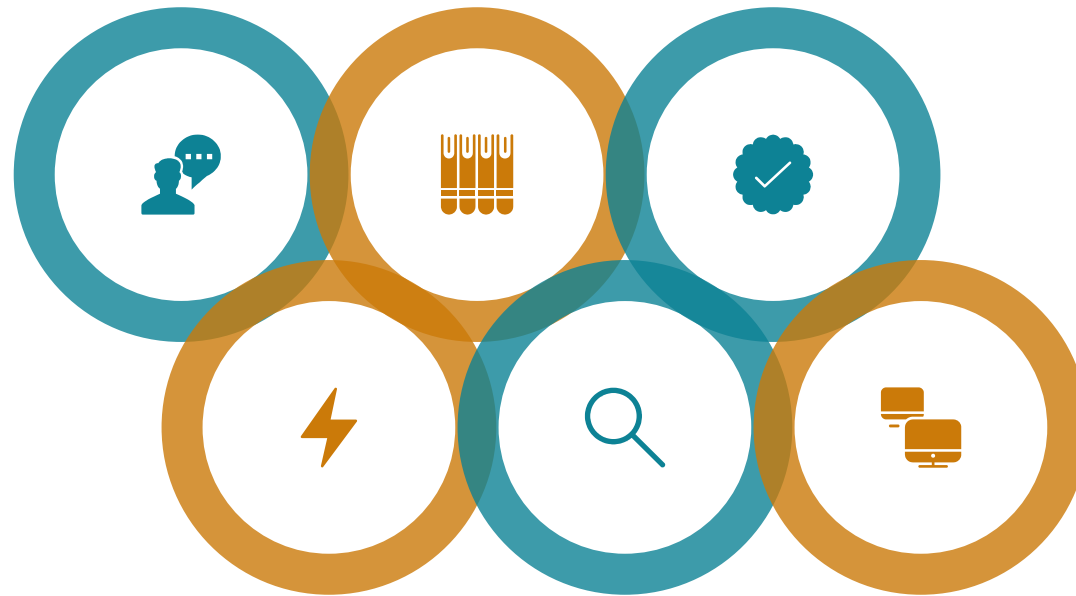
In the future I aim to have automated tests done.  
e.g. stress testing

# Development Approach

1. I broke down the development into manageable steps.

2. I read a Beginner's Rust book while implementing a simple String parser and manipulator.

3. Then I learned about asynchronicity and the Tokio library to learn Networking.



4. I learned how to make a simple webserver with Threads and Workers.

5. I made a simple asynchronous database with a single value that could be modified and accessed by multiple users.

6. I started working on a full key-value implementation with BTrees, multiple operations and concurrency.

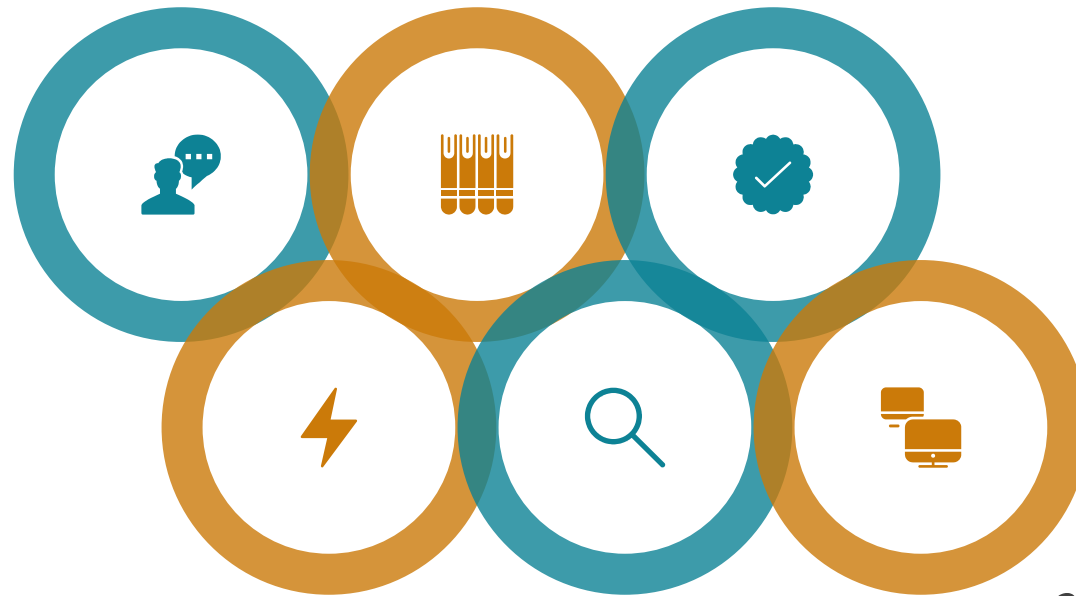
# Future Plans

16/01

1. Finish modularizing the DB operation functions and command parsing.

2. Implementing automated testing functionality for the DB.

3. Add Operation logging.



30/01

4. Implement writing to the storage drive.

5. Add Crash recovery functionality.

27/02

6. Implement a Web access API.



**Thank You**