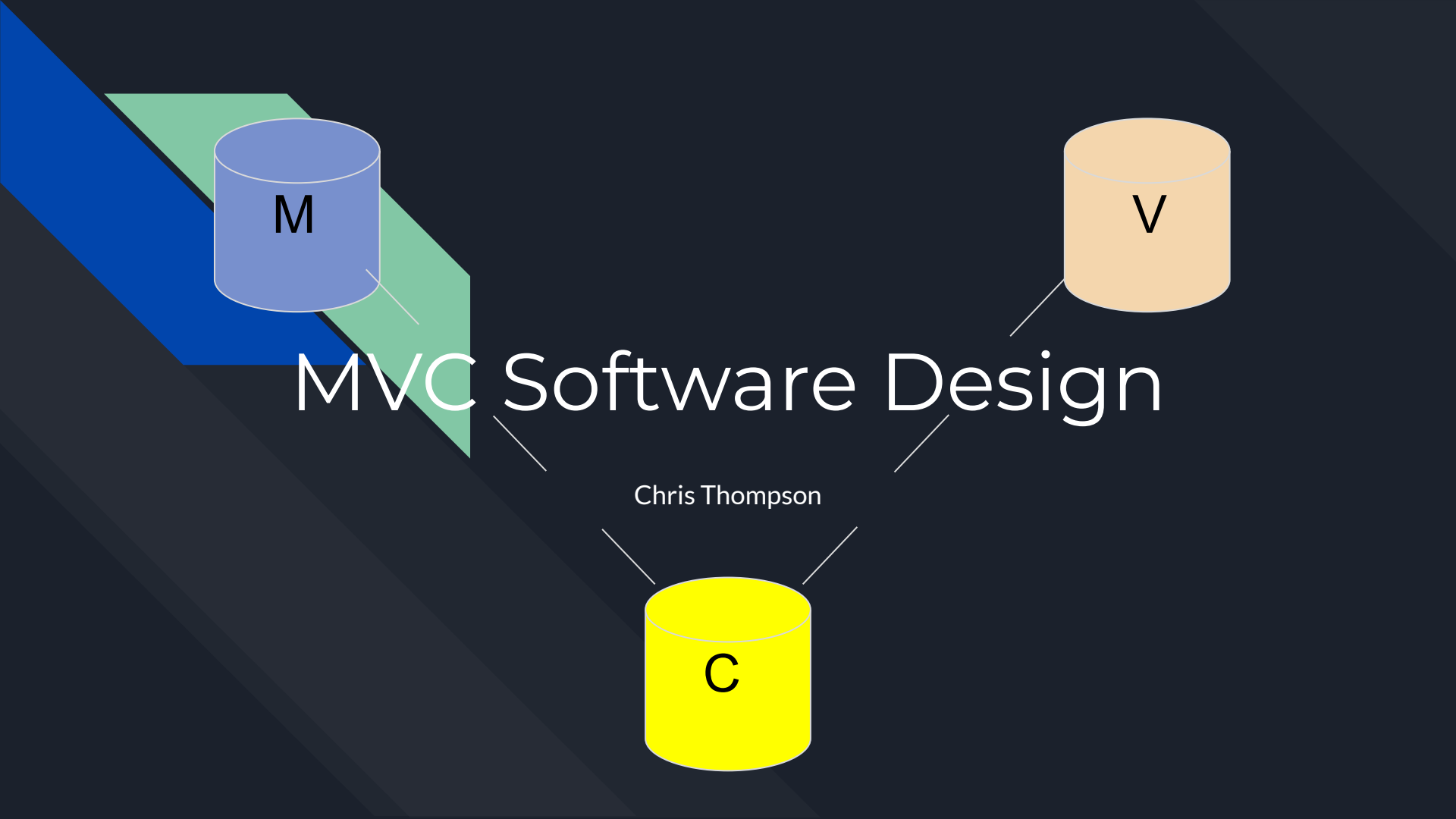


# MVC Software Design

Chris Thompson

# What does MVC Stand For and what is it for?

M: Model

V: View

C: Controller

MVC is used when building FullStack software or applications that require communication between different frameworks and services. It creates “separation of concerns” within our backend code. Each segment of code is separated into different files.

# What is the Model?

Per the MDN, the Model “Manages data and business logic” and “Defines data structure (e.g. updates application to reflect added item)”.

In simpler terms, our Model is what communicates with our database. It can Create (Post), Read (Get), Update(Put) or Delete elements from our database and then send the information back to us.

The Model has 1 job, and it reports everything directly to it’s boss;

> “The Controller”.

# What are some features or tools used by the Model?

Our Model can use **Schemas** (blueprints to create objects from our database content) .

**Object Data Models** (ODMs) allow for us to avoid using the databases' native languages. Some are specific to a particular database, while others can be used across various databases. Examples:

- Mongoose for MongoDB,
- Waterline for Sails, Redis, MySQL, LDAP, MongoDB and Postgres,
- Bookshelf for PostgreSQL, MySQL and SQLite3

# What is the View?

While it could be a 25 season daytime talk show featuring Whoopi Goldberg, we are talking about software. Per the MDN, the View “Handles layout and display”.

This is everything that our user will see (a.k.a. The DOM (again, we are talking about software, not a crazy thief that drives fast cars)).

The View can be .ejs code, handlebars and more.

It communicates exclusively with it’s boss; The Controller.

The controller sends and receives information to, and from, the View to manage what the user sees.

# What is the Controller?

You would think that the big boss would be someone like Shao Kahn, Agent Smith or Bill Lumbergh, but it's intentions are far less sinister or selfish.

Per the MDN, the Controller "Routes commands to the model and view parts" and "Contains control logic".

The controller is the glue that holds our application together. It communicates with our Model and our View and allows them to work together. It is kind of like a USPS or Amazon shipment hub where everything goes to be sorted and then sent out.

# Why is this important?

The software world is complex and always evolving. If a business decides to change how it handles a part of their application's processing, it could become a very large overhaul. Utilizing MVC allows for an application to have one of these processes changed into a new language or framework without having to change the entire codebase.

For example: If a an application decides to start using SQL instead of MongoDB, they would ONLY need to replace the Model code.

Same goes for if a software using EJS decides to switch to Handlebars, they can simply change the code for the View.

# Sources

MDN: https://developer.mozilla.org/en-US/docs/Glossary/MVC

https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose