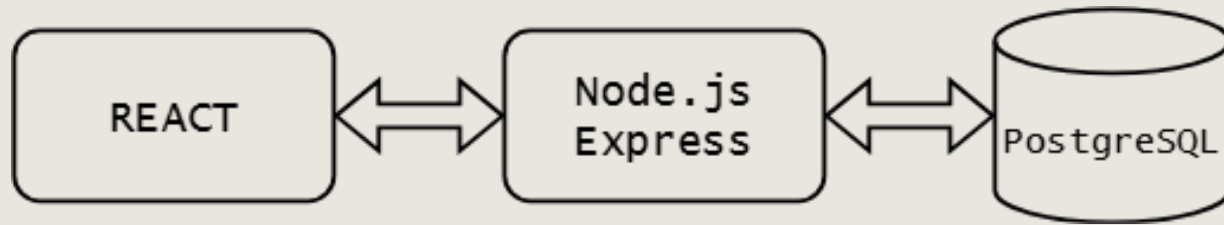




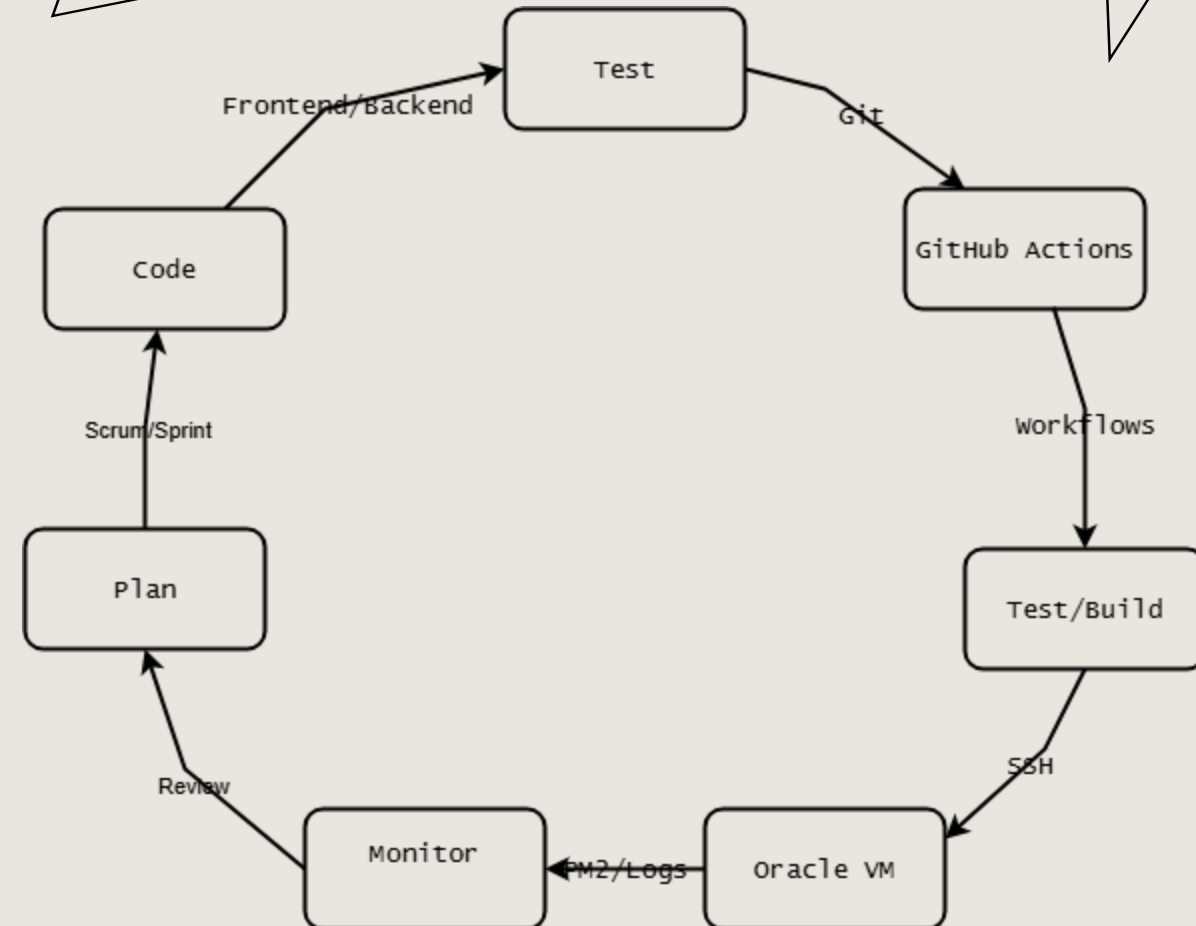
MovieTok

MARTIN NEGIN
SAMU LYHTY
TOPIAS PERÄLÄ
VEIKKA KOSKINEN

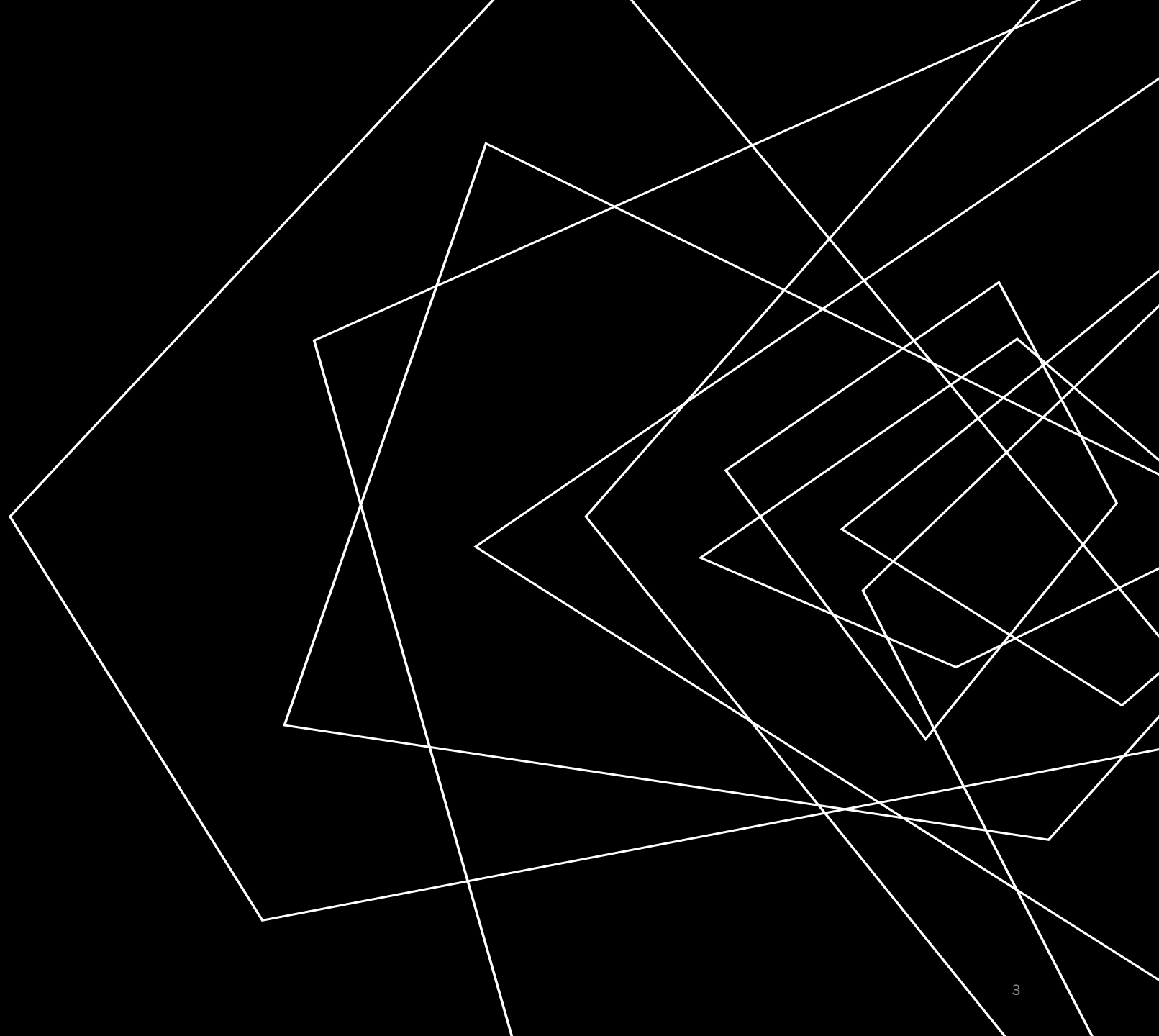
PROJECT ARCHITECTURE



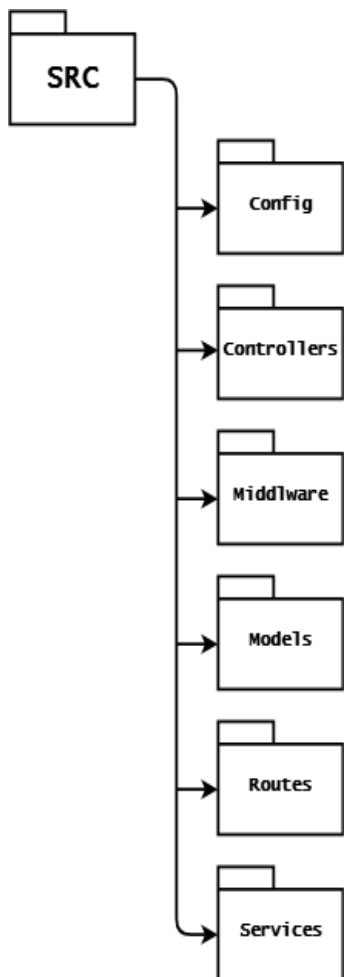
- **CI/CD**
 - Automated building, Automated testing, Version control
- **Backend**
 - Node.js, PostgreSQL
 - Frameworks: ExpressJS, bcrypt, JWT Auth, dotenv, XML2JS
 - Testing: Chai, Mocha
- **Frontend**
 - Node.js, React, Vite, Axios, TailwindCSS, PostCSS
- **Oracle**
 - NGINX Web Proxy frontend, Ubuntu server



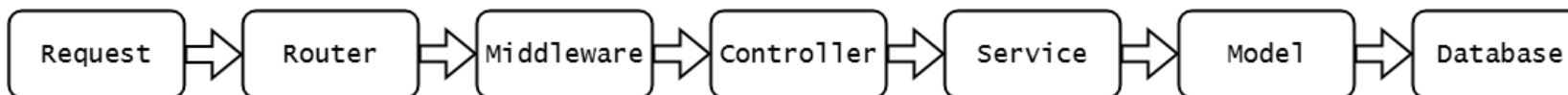
BACKEND



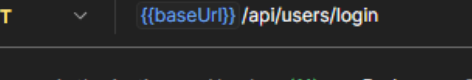
APPLICATION PROGRAM INTERFACE (API)



- **Config:** Database connections and environment settings management.
- **Controllers:** HTTP request/response handling and validation.
- **Middleware:** Authentication, CORS, error handling.
- **Models:** Database interaction with raw SQL queries.
- **Routes:** API endpoint definitions with authentication requirements.
- **Services:** Business logic and data processing.



Endpoint example



The screenshot shows a REST client interface. The top bar indicates a **POST** request to the endpoint `{{baseUrl}}/api/users/login`. Below the bar, there are tabs for **Params**, **Authorization**, **Headers (11)**, **Body** (which is selected and underlined), and **Scripts**. Under the **Body** tab, there are four radio button options: **none**, **form-data**, **x-www-form-urlencoded**, and **raw** (which is selected). The **raw** body contains the following JSON:

```

1  {
2    "email": "Teppo@Movietok.fi",
3    "password": "123456"
4  }

```

```

1
2      "message": "Login successful",
3      "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  

4          eyZpcjC160cwIdXNlcmhhbnUwIOUzXBwb1RlC3RoIiwiaWwiOiJIUzE  

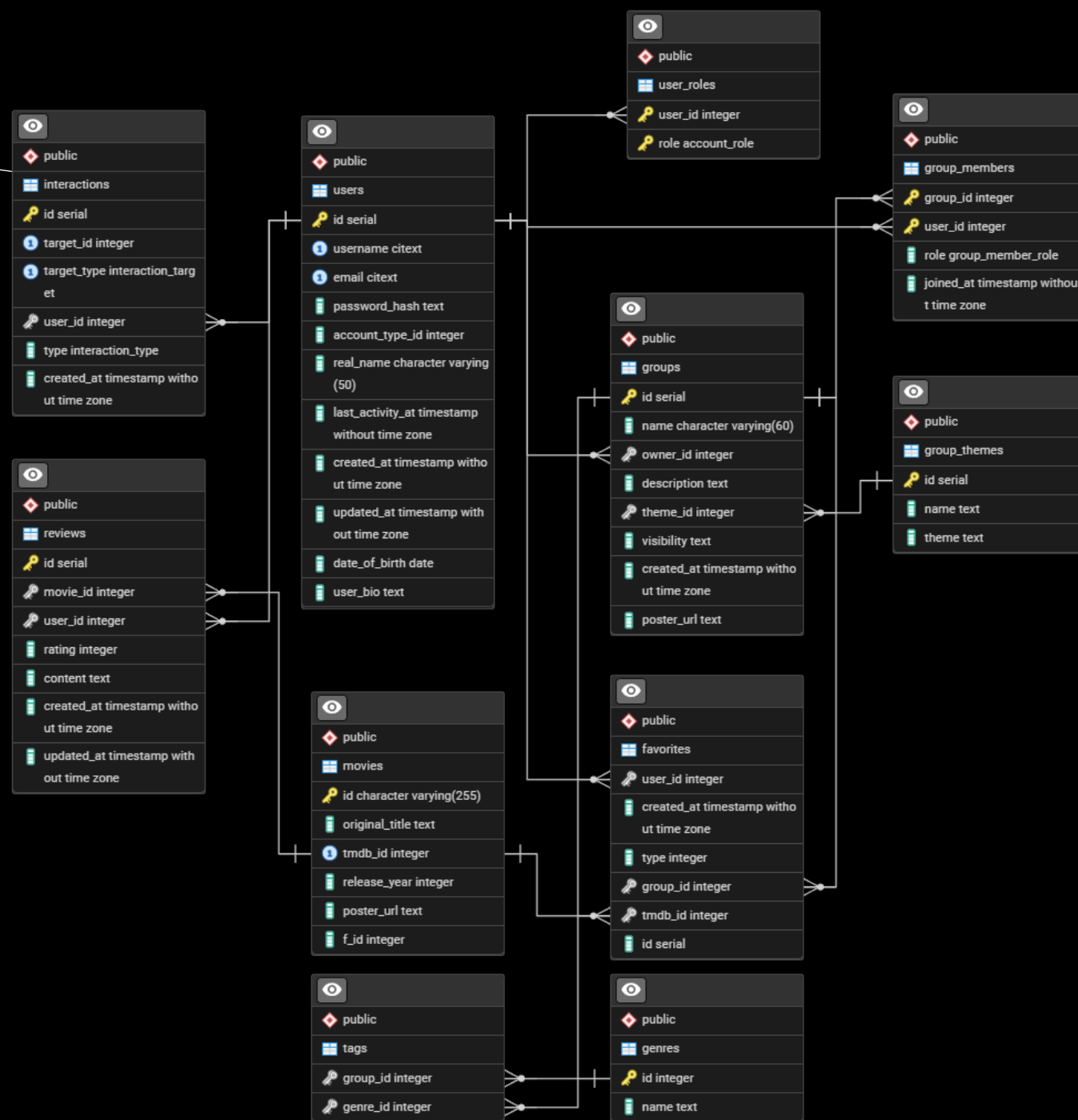
5          bGNBNzSpZkYyYVY5bSIsImhhdCI6MTkxOTkwTUo3MiwiaWF0IjoxNDU5OT  

6          MTCyFQ.GjWTtjt2S69pyic4BDDBDNg6K3rSVvD24PXC9dkZo",
7      "user": {
8        "id": 8,
9        "username": "TeppoTesti",
10       "email": "Teppo@Movietok.fi",
11       "real_name": null,
12       "user_bio": null,
13       "created_at": "2025-09-29T19:36:38.436Z",
14       "updated_at": "2025-09-29T19:36:38.436Z",
15       "review_count": "0",
16       "average_rating": null,
17       "total_likes": "0",
18       "total_dislikes": "0",
19       "aura": "e"
20     }
21   }

```

DATABASE

- **Complex relationships**
 - Many to many relationships
 - Self-referencing relationships
 - Multi-level foreign key constraints
- **Advanced features**
 - Three-tier favorites system
 - Social interaction tracking
 - Role-based permissions
- **Performance optimizations**
 - Strategic indexing
 - Normalized structure
 - Composite keys



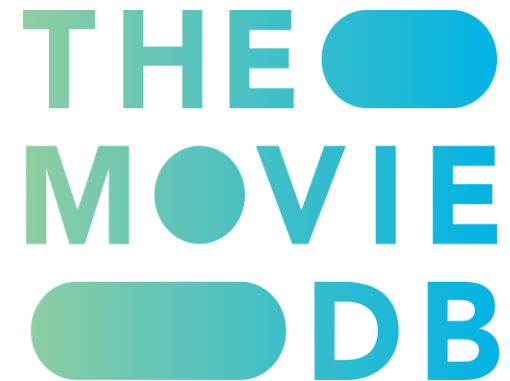
SECURITY

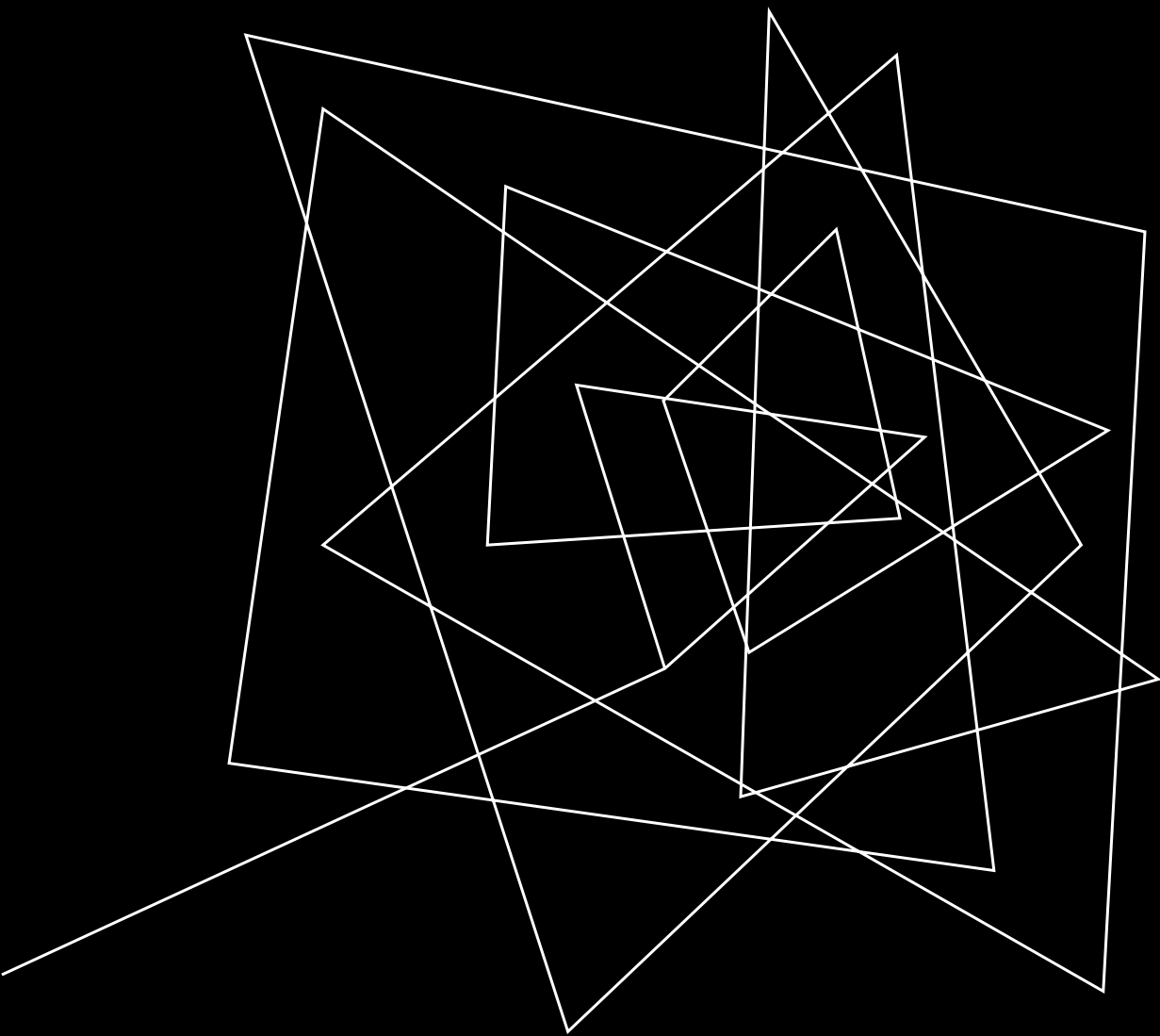


- **JWT Token System** with environment-specific expiration
- **Hybrid Authentication**
 - Public users get limited data
 - Authenticated users get full access
- **Input validation** and SQL injection prevention
- **Password Security** with bcrypt hashing
- **Sanitized Error Responses**
- **Database Security**
 - Connection pooling
 - Timeout settings
 - SSL configuration

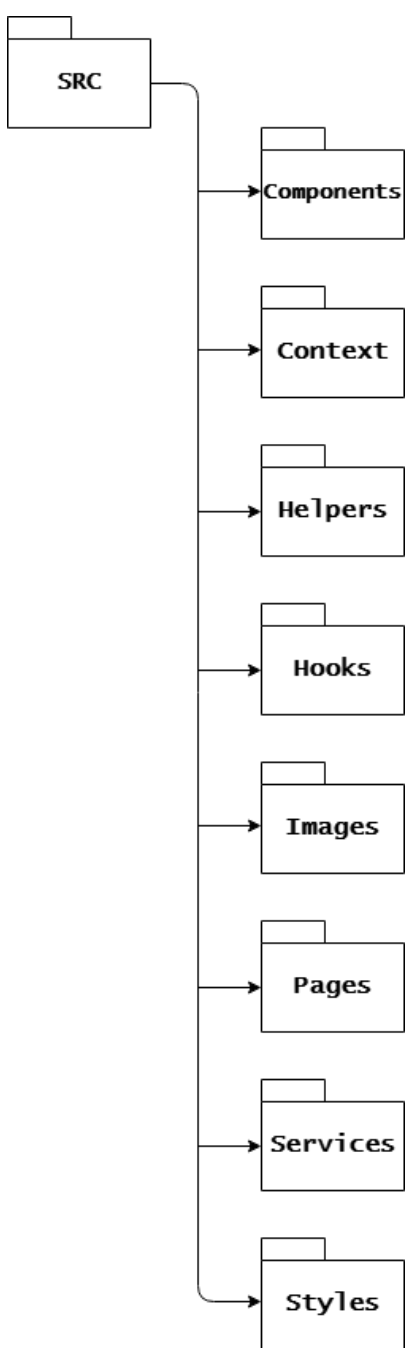
THIRD-PARTY SERVICE INTEGRATION

- **TMDB Integration:** Global movie database with 4+ endpoints for fetching data.
- **Finnkino API:** Finnish cinema showtimes and theatre data.
- **API Response Optimization:** Single requests return complete data.
- **Error handling:** Graceful fallbacks when external APIs fail
- **Rate limiting:** Preventing API abuse and quota management





FRONTEND



SRC

Codebase is structured in the following manner to allow ease of use in scalability and reuse of components.

Codebase structure of the /src directory

/components includes reusable UI components, such as buttons

/context includes global app state

/helpers includes utility functions

/hooks includes event handling functions

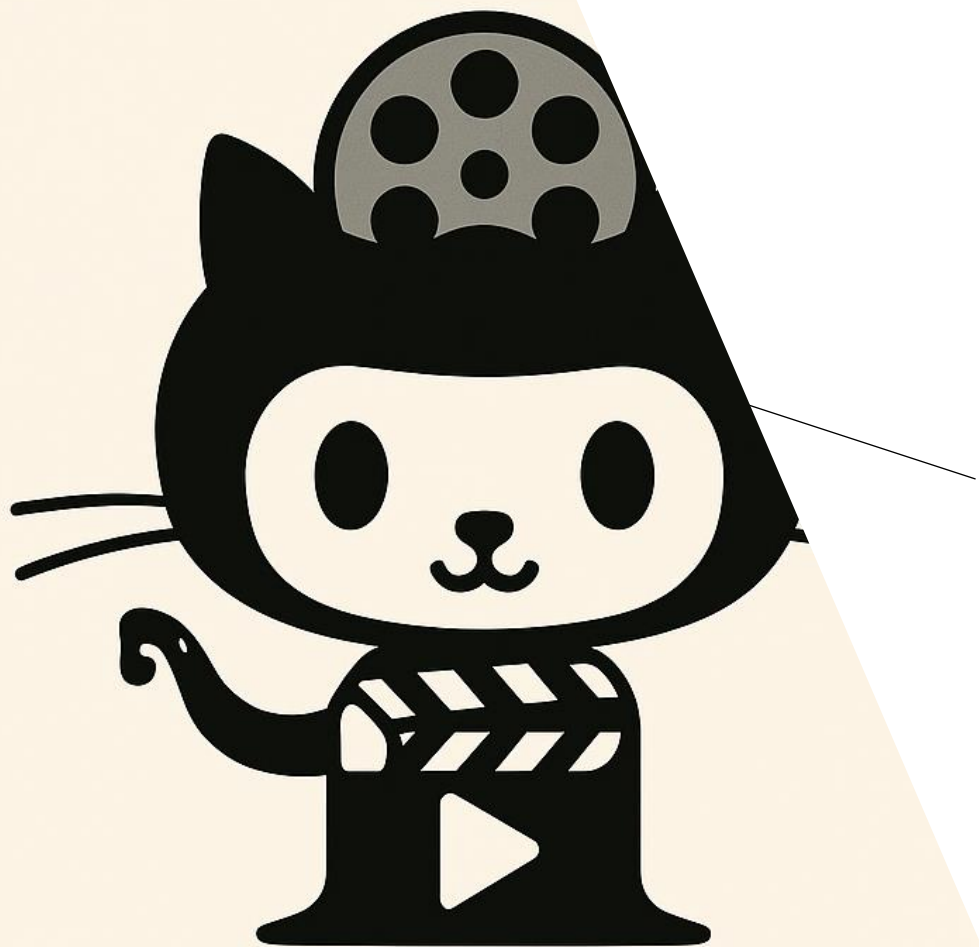
/images includes media used

/pages includes complete page files

/services API interaction layer

/styles includes modules for visual styling

/app.jsx and main.jsx



MovieTok

CREDITS

Our Team of four

- Topias Perälä - Focused on frontend: Homepage, groups, reviews, browse pages and their detail page functions
- Veikka Koskinen – Focused on frontend: architecture, navbar/footer about etc, profile, settings, signup/login, favorites, showtimes
- Samu Lyhty – Focused on backend: TMDB, Groups, Reviews & User management endpoints. Database structure, connections and data managing.
- Martin Negin – Focused on backend: Finnkino, Favorites, User management, Interactions & Reviews endpoints. Database structure & connections. Frontend: Finnkino integration. CI/CD Automation to Oracle server. Scrum master.

Week	35	36	37	38	39	40	41	42	Total	
Martin	28	26	25	32	28	38	30	30	237	Hours
Samu	18	24	29	30	34	32	31	30	228	Hours
Topias	10	15	20	20	35	40	40	15	195	Hours
Veikka	15	26	35	28	24	43	45	20	236	Hours

A series of white, thin, overlapping geometric lines on a black background, forming a complex, abstract pattern on the left side of the slide.

THANK YOU

Proceeding to demo