# Motivation

Share details on steps we would like to follow and confirm that items in *Future Considerations* section are valid and we can dive deeper into them. If they are not valid please confirm that we can think about already existing frameworks/libraries usage for React(as in document bellow)/VueJS/Angular to implement ERC-3643 functions or share your own vision on this topic.

In the nut-shell our initial idea is to create React hook (+component) to be easily integrated into DApp implemented with `@usedapp` framework.

# Dev Setup

Technical steps for preparing a Git monorepo for project using TypeScript. The focus will be on leveraging advanced TypeScript features such as "references" and "composite" options, with the goal of creating modular packages that align with potential future NPM module publication. The key modules will include a non-framework-specific implementation of ERC-3643 standards, particularly centered around the ABI(s), under the package name `@erc-3643/core`.

## Repository Structure

Our project's monorepo structure may be organized as follows:

```
project-root/
|-- packages/
| |-- @erc-3643/core/
| |-- @erc-3643/react-useDApp/
| |-- other-packages/
|-- src/
|-- .gitignore
|-- tsconfig.json
|-- package.json
|-- README.md
```

## TypeScript Configuration

To set up TypeScript with the "references" feature, `tsconfig.json` may be as follows:

```
{
  "compilerOptions": {
  "baseUrl": "./",
  "declaration": true,
  "strict": true,
    // others...
  },
  "references": [
    { "path": "./packages/@erc-3643/core" },
    { "path": "./packages/@erc-3643/react-useDApp" }
  ]
}
```

## `@erc-3643/core` Package

The core package will encapsulate ERC-3643 related implementations. The initial focus will be on the ABI implementation. Additional generic functions that could be utilized across modules will be progressively added.

### Package Structure

```
@erc-3643/core/
|-- src/
| |-- abi.ts
| |-- generic-functions.ts
|-- package.json
|-- tsconfig.json
```

## `@erc-3643/react-useDApp` Package

This package will specialize in providing framework-specific hooks for the applications build with `@usedapp`, capitalizing on the capabilities of the `@erc-3643/core` package.

### Package Structure

```
@erc-3643/react-useDApp/
|-- src/
| |-- hooks/
| |-- erc3643-hooks.ts
|-- package.json
|-- tsconfig.json
```

# Future Considerations

As the project evolves, the following aspects require further exploration and development:

- **Non-Framework-Specific Implementation**: As per requirements. Develop strategies to integrate the `@erc-3643/core` module into React, VueJS, and Angular projects without requiring separate framework-specific implementations.
- **Connectivity and Event Handling**: Implement features like automatic refresh on new blocks, wallet changes, or network changes. Consider utilizing `ethers.js` library to facilitate this functionality.

# Conclusion

By adhering to the outlined structure and utilizing advanced TypeScript features, we'll be well on our way to creating a robust monorepo with modular packages that adhere to open-source best practices. The division between the core package and the framework-specific package sets a strong foundation for scalability and flexibility in ERC-3643 related project.