Разработка ПРИЛОЖЕНИЙ

Занятие #5.

ПЛАН

01 Повтор

02 Context

03 Redux

04 Практика

Прогресс

- ★ create-react-app, основы, npm
- ★ компоненты, jsx
- ★ git
- ★ Состояния, переменные, props
- ★ useState()
- ★ Стилизация
- ★ useEffect(), useRef()
- ★ Отладка
- ★ Маршруты, React Router

- ❑ Redux, useContext()
- ❑ Firebase и API
- ❑ Тест, деплой
- ❑ БЭМ, цикл проекта, figma
- ❑ PWA
- ❑ Улучшения, библиотеки
- ❑ React + TS
- ❑ Next.js

01 Повтор

Вопросы

- Зачем нужны маршруты на фронтенде?
- Какой компонент управляет маршрутами в реакт?
- Как сделать “правильную” ссылку?

ДЗ прошлой лекции

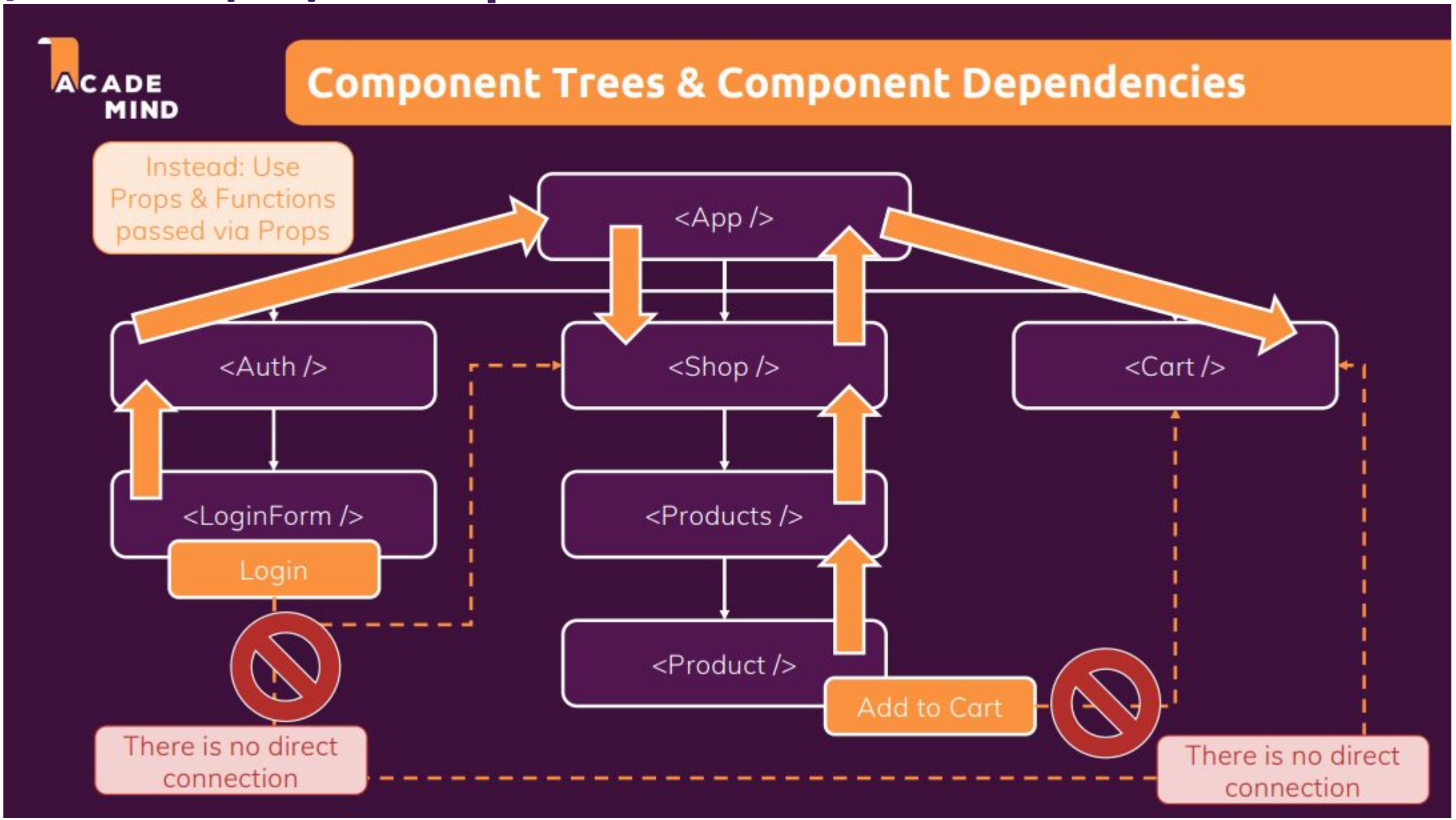
- Добавить функцию удаления кота
- Добавить страницу кота без переменных (универсальную)
- Добавить навигацию с помощью children в react
- Почитать про localStorage

02 Context

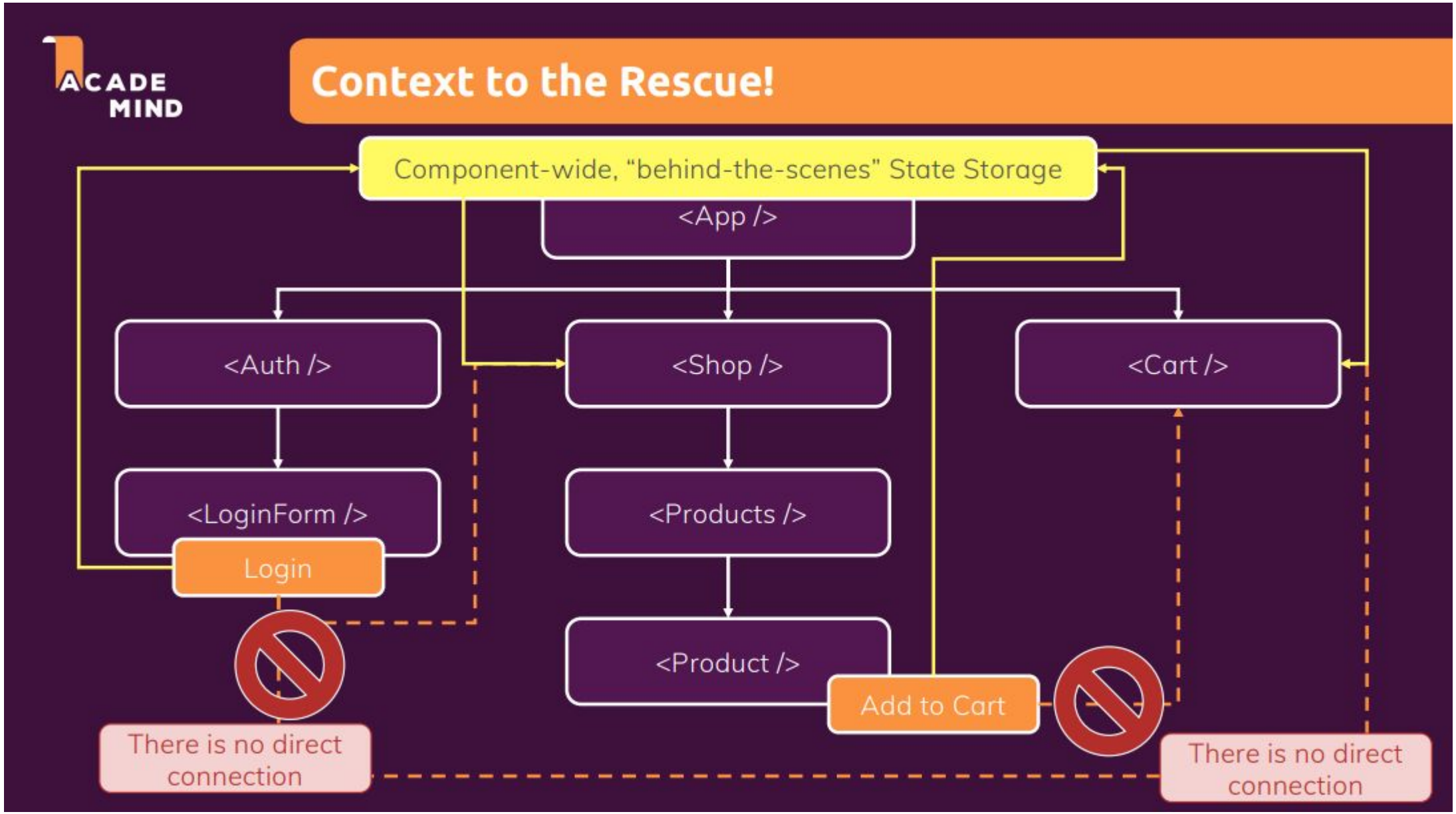
Как управлять состояниями?

- useState
- **React Context**
- Redux

Цепочки props и передача состояния



Единое хранение состояния



React Context

- React.createContext()
- <[Context].Provider/>
- useContext()

React Context

Файл с контекстом, например context/app-context.js

```
import React from "react";

const AppContext = React.createContext({
    isLoggedIn: false,
    loginHandler: () => {},
    logoutHandler: () => {},
})

export default AppContext;
```

React Context

App.js

```
import AppContext from "./context/app-context";
…
return (
    <AppContext.Provider
      value={{
        isLoggedIn: isLoggedIn,
        loginHandler: () => {setIsloggedIn(true)},
        logoutHandler: () => {setIsloggedIn(false)},
      }}
    >
      <RouterProvider router={router} />
    </AppContext.Provider>
  );
```

React Context

Компонент, обращающийся к стейту

```
import AppContext from "../../context/app-context";

export default () => {
  const ctx = useContext(AppContext);
  return (
    <>
      <Button onClick={ctx.loginHandler}>Login</Button>
      <Button onClick={ctx.logoutHandler}>Logout</Button>
    </>
  );
};
```

Практика

React Context. Недостатки

- Не оптимизирован для частого изменения состояния (раз в секунду и чаще)
- Неудобен для сложных стейтов

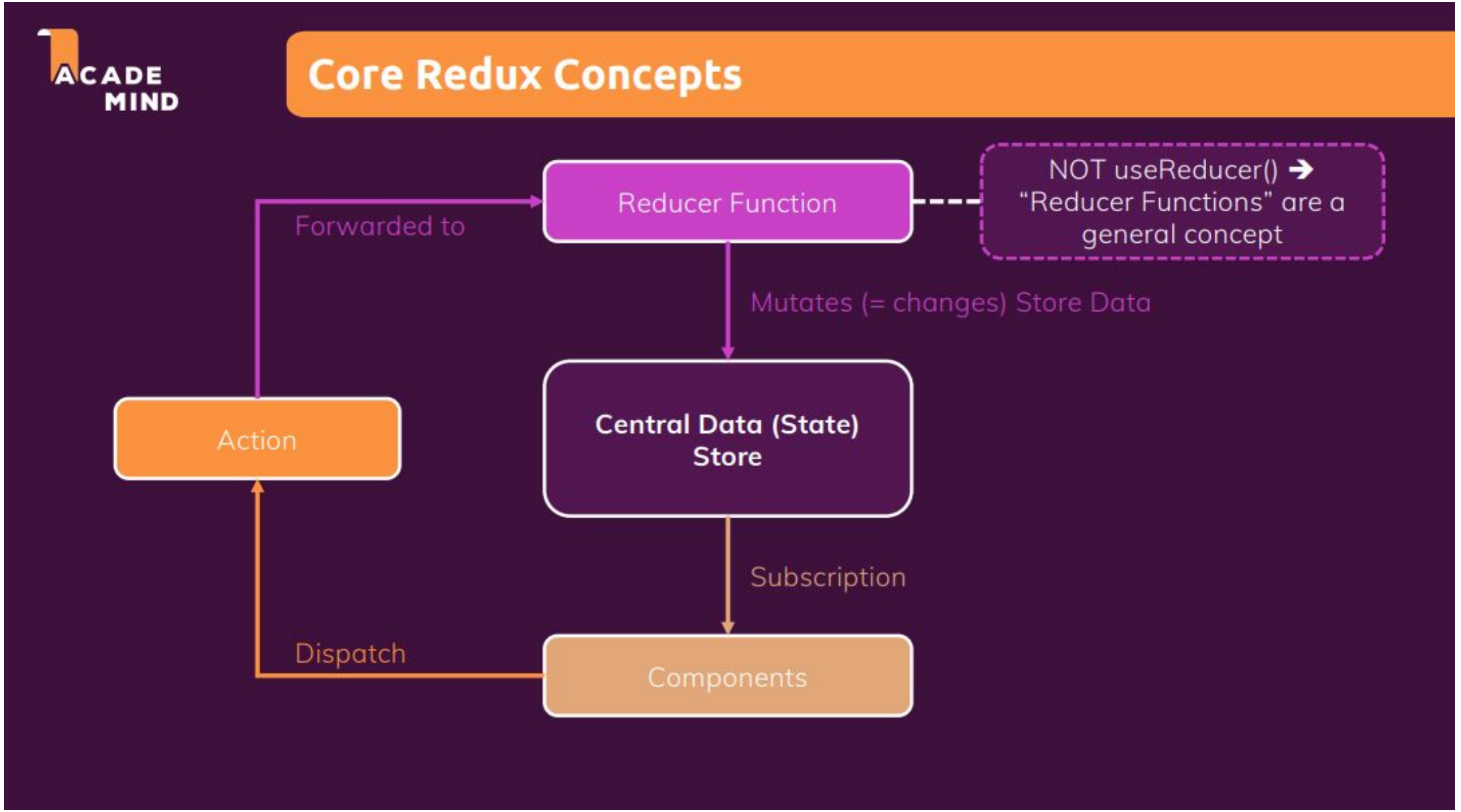
03 Redux

Redux

Система управления состояниями на уровне приложения

- Более приспособлен для комплексных стейтов
- Продуманная структура

Redux


ACADEMIND
Core Redux Concepts
Reducer Function
NOT useReducer() ➔
"Reducer Functions" are a general concept
Forwarded to
Mutates (= changes) Store Data
Action
Central Data (State) Store
Subscription
Dispatch
Components

Понятия

- Store
- Компоненты
- Reducer
- Подписка на изменение

Redux далее

- https://redux.js.org/tutorials/quick-start
- начать с createStore() - “устаревший”
- продолжить с Toolkit

ДЗ

1. **Имитировать авторизацию и доступ к закрытой информации (через разные данные или разные страницы)**
    a. разработать форму авторизации (локально)
    b. создать маршруты и разные данные (или страницы)
    c. реализовать на React Context

    или
2. **То же на Redux**