



**ISOLATED  
COMPONENTS**

**DRIVEN  
DEVELOPMENT**

**WHAT**

**WHEN**

**HOW**

WFSO



# ANDREI PFEIFFER

Timișoara / RO

**Code Designer**

**UI Engineer for Web & Mobile**



Code Designer



Co-Organizer



[andreipfeiffer.dev](https://andreipfeiffer.dev)

**WHAT**

# DEVELOPMENT & DEBUGGING TECHNIQUE

**Component  $\rightarrow$  f(props)**

**WHEN**

COMPONENTS LIBRARY

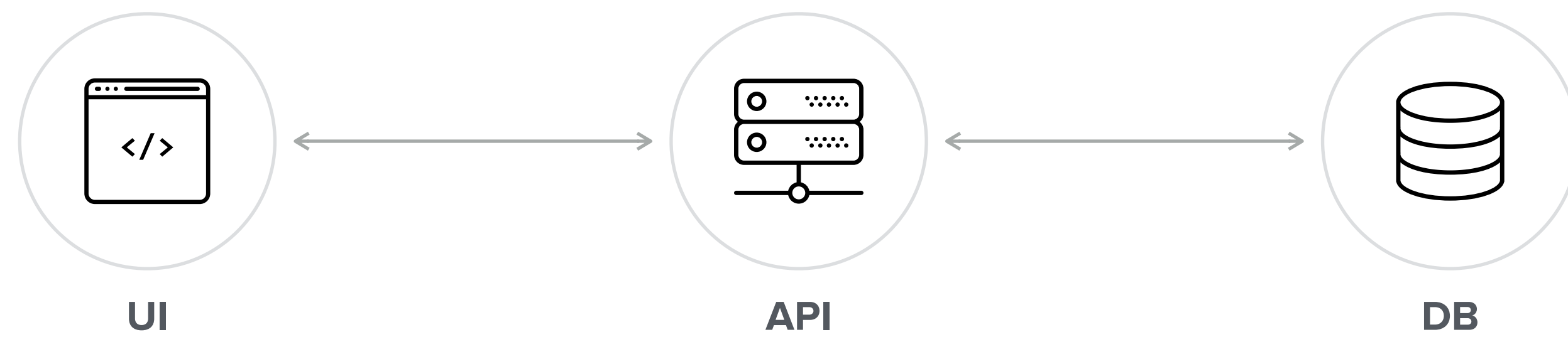
**WE NEED A PLACE TO  
RENDER & USE THEM**

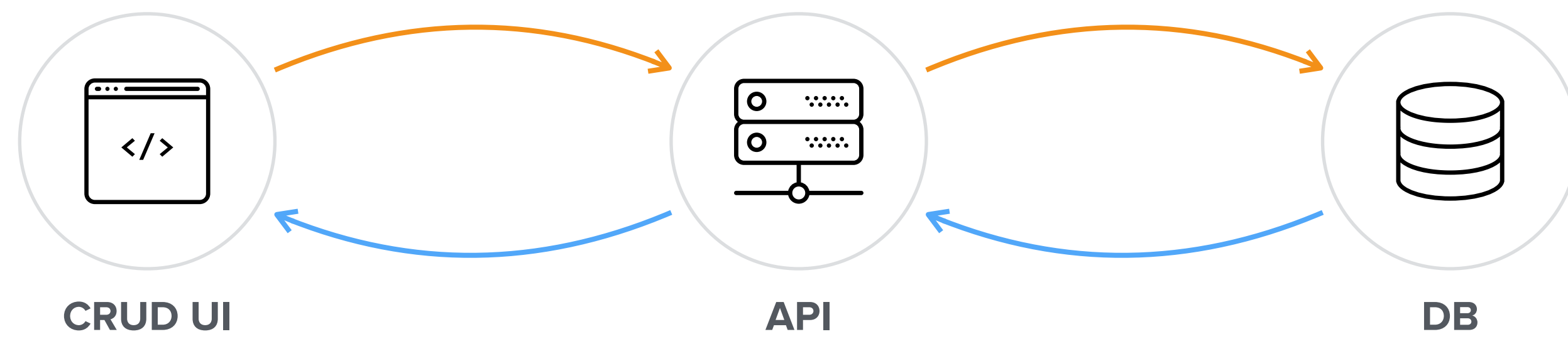




READ-ONLY UIs

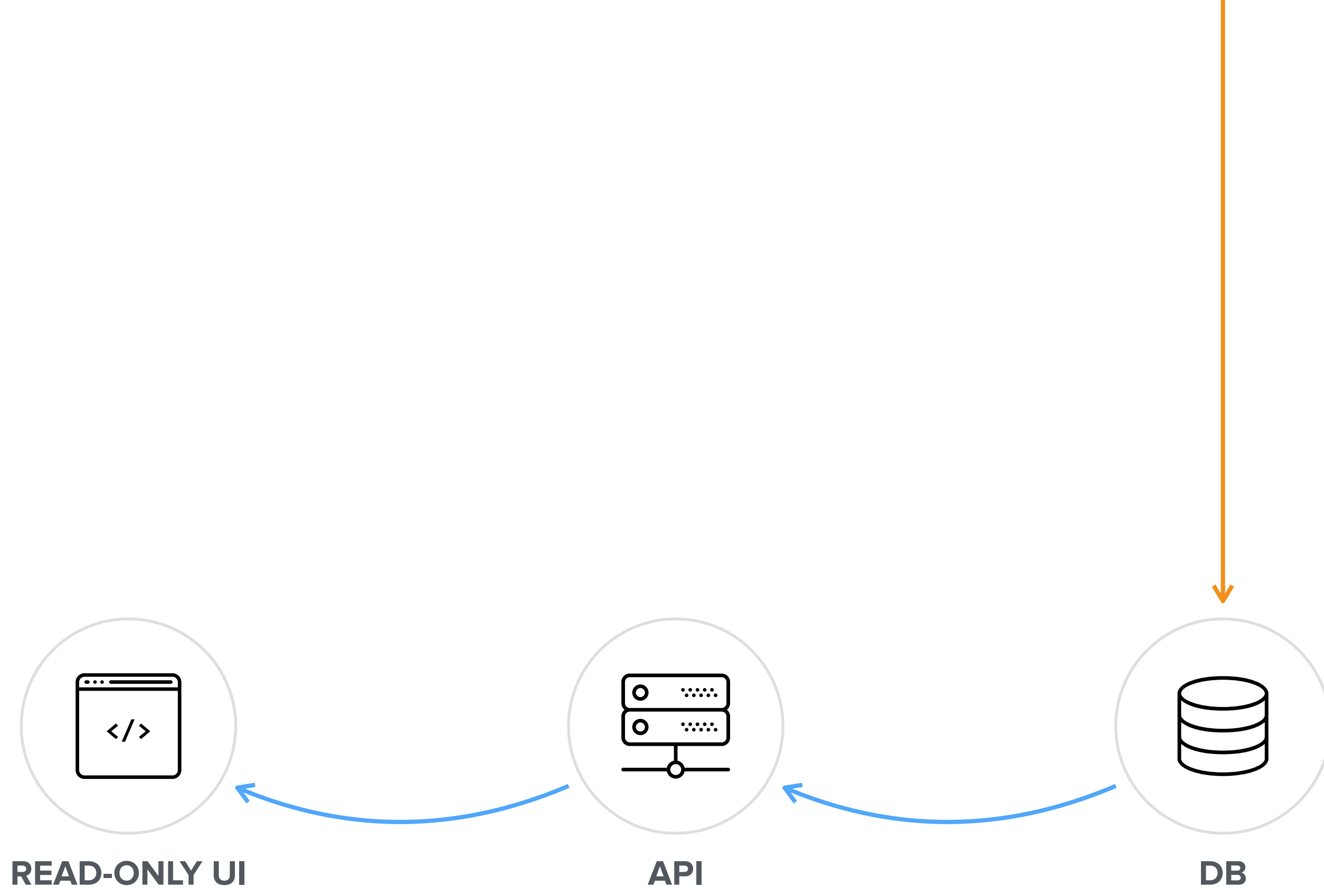
2

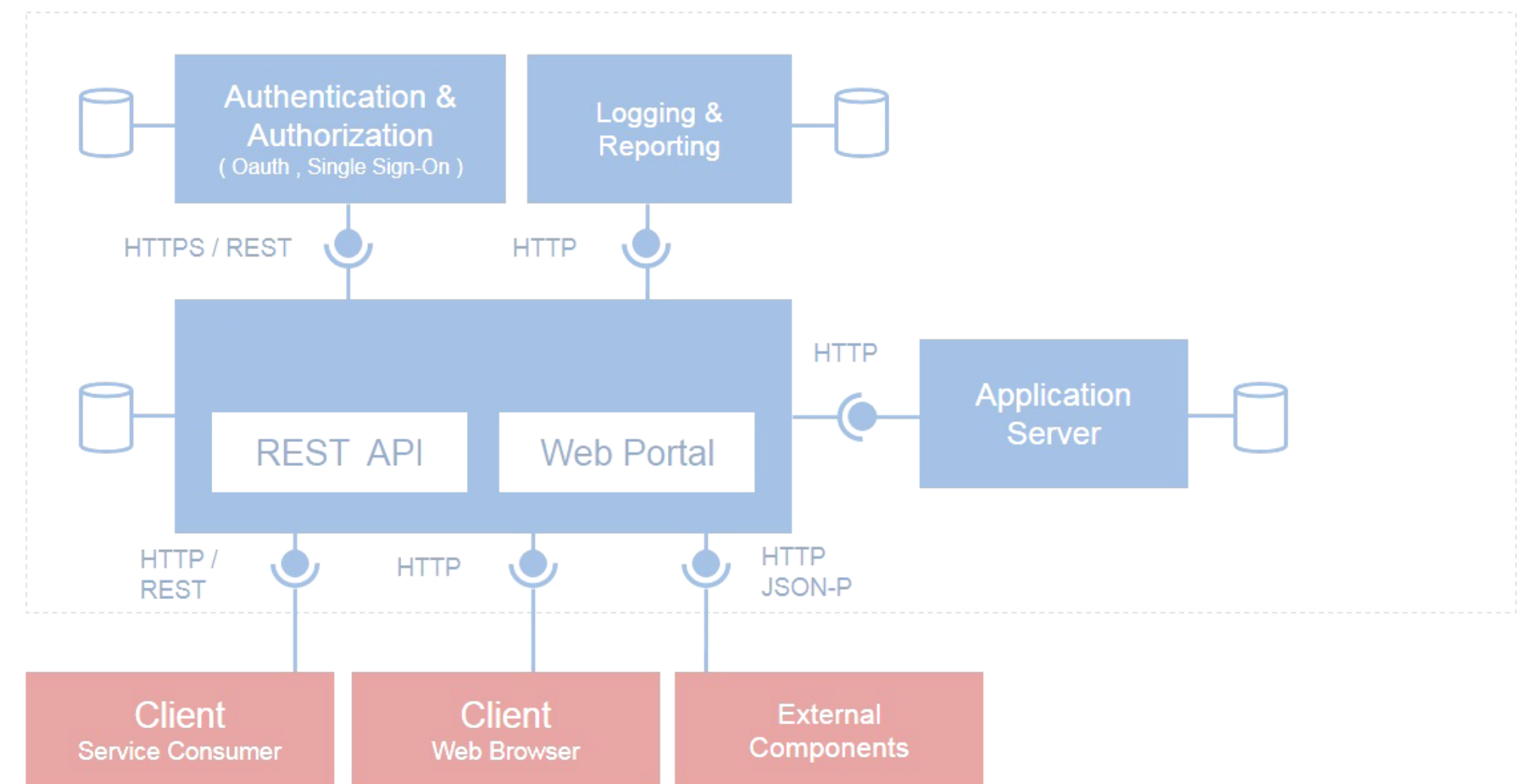




← READ OPERATIONS

→ WRITE OPERATIONS



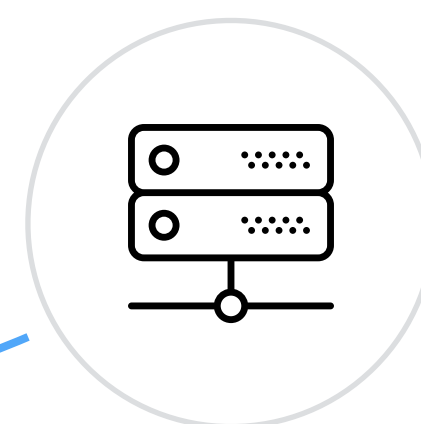


 Google Analytics

 **GitHub**



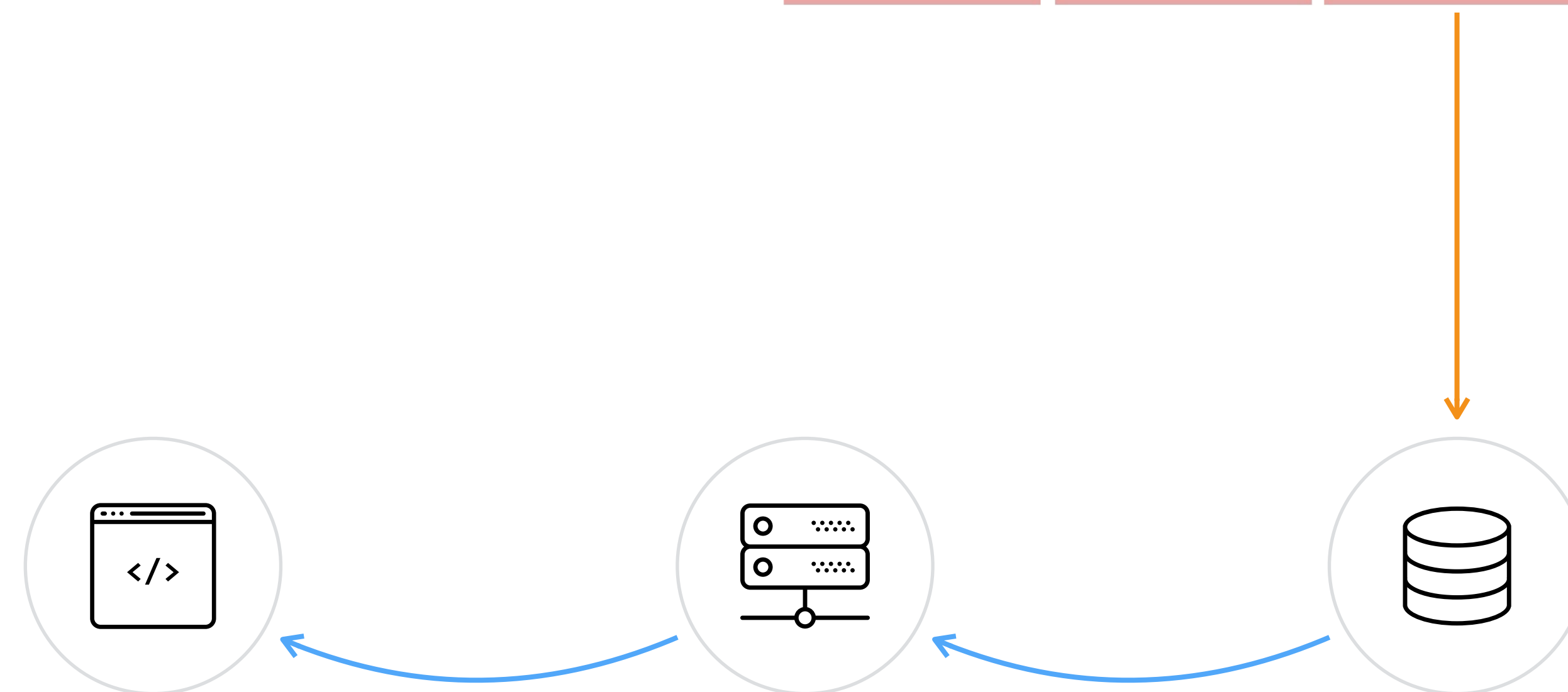
**READ-ONLY UI**



**API**



**DB**



**DEVELOPING READ-ONLY USER INTERFACES  
IS PROBLEMATIC**



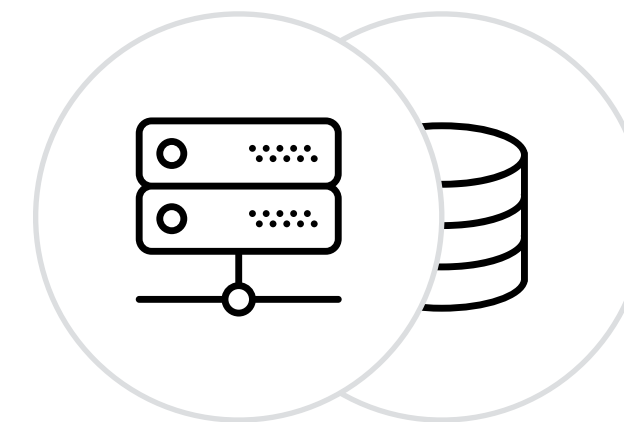
UNSTABLE

WEB APIs

3



**UI**

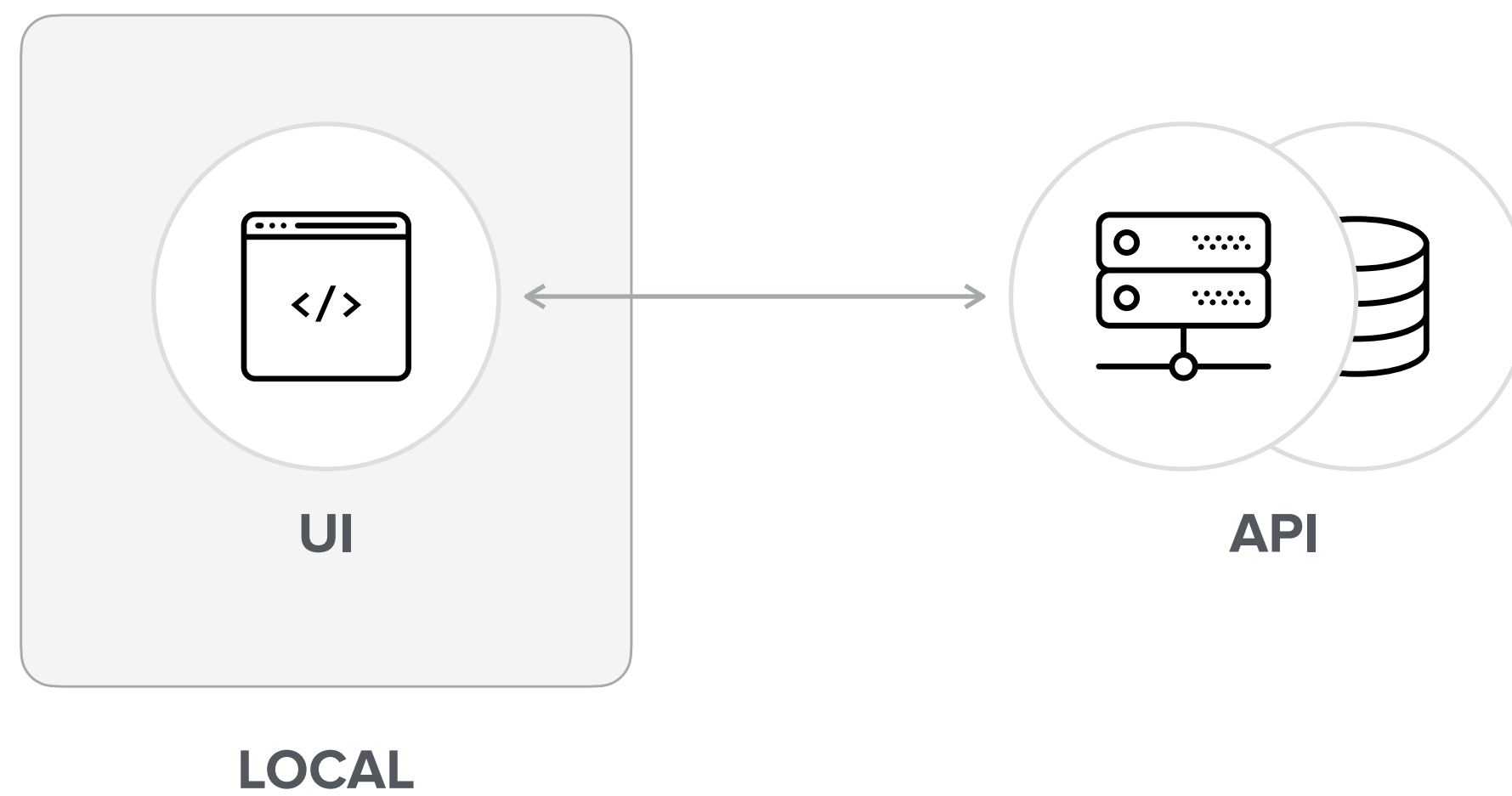


**API**



# 1. REMOTE DEVELOPMENT SERVER

---



**NO LOCAL API SETUP**

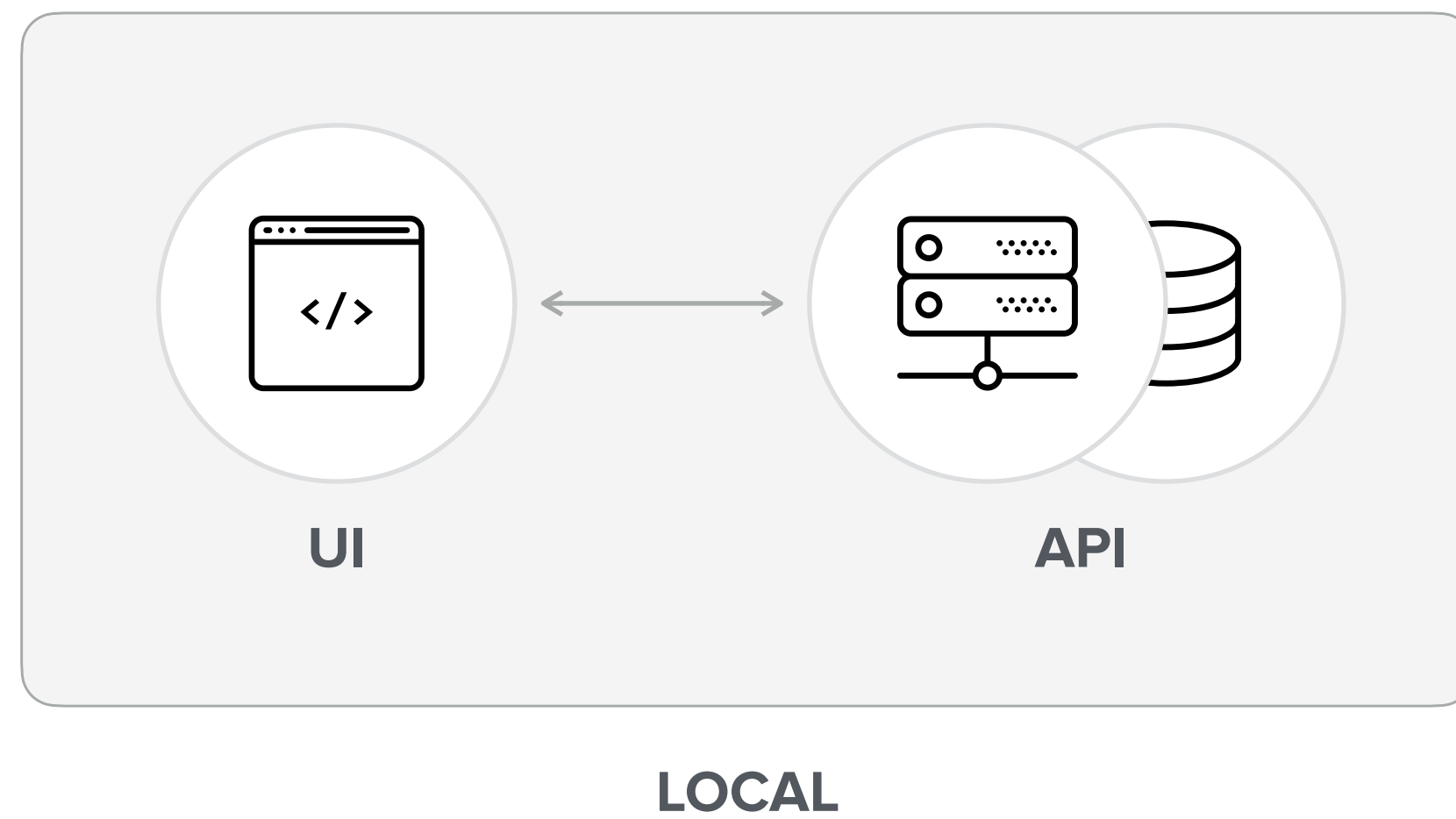
**SHARED DATA**

**INFRASTRUCTURE DEPENDENCY**

**ONLINE ONLY**

## 2. LOCAL API SETUP

---



OFFLINE

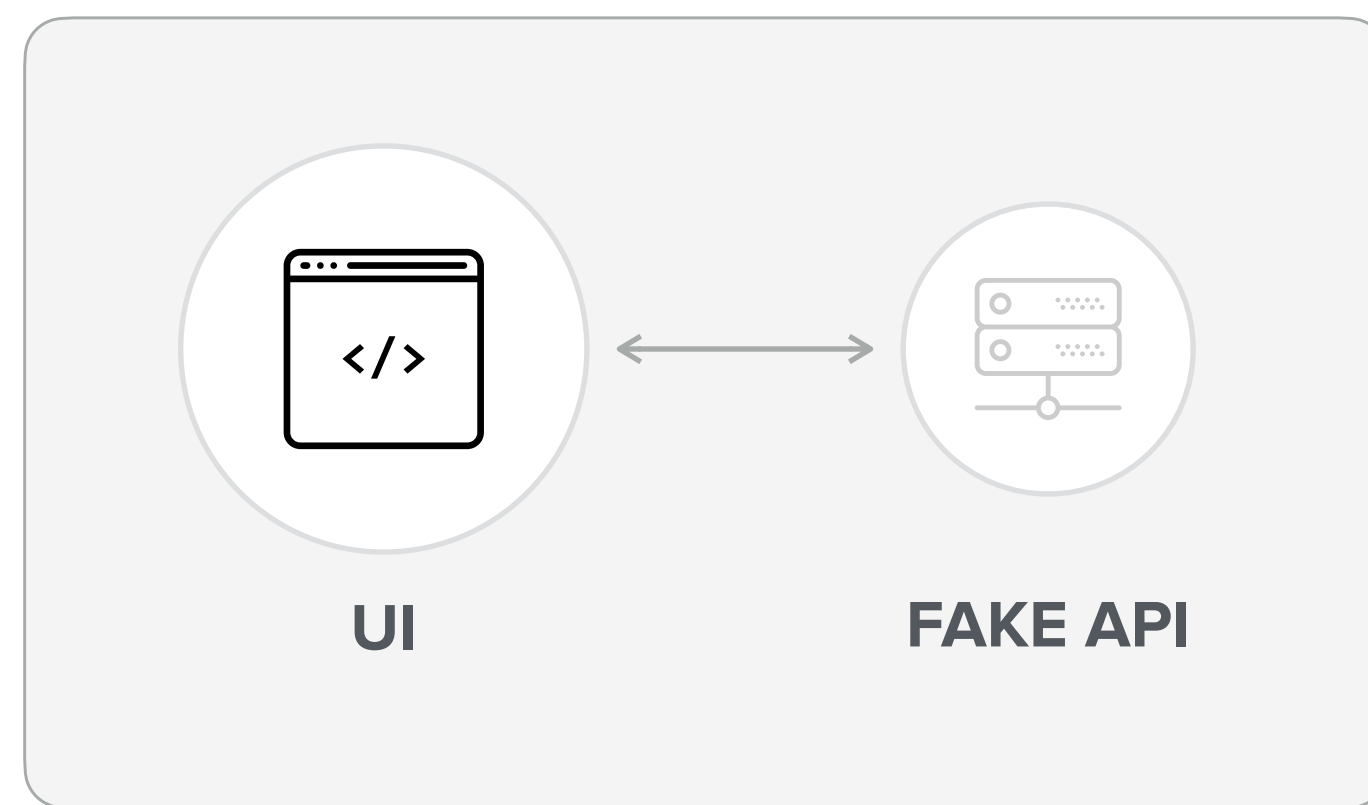
EASILY SWITCH BRANCHES

NON-TRIVIAL DATA SYNC

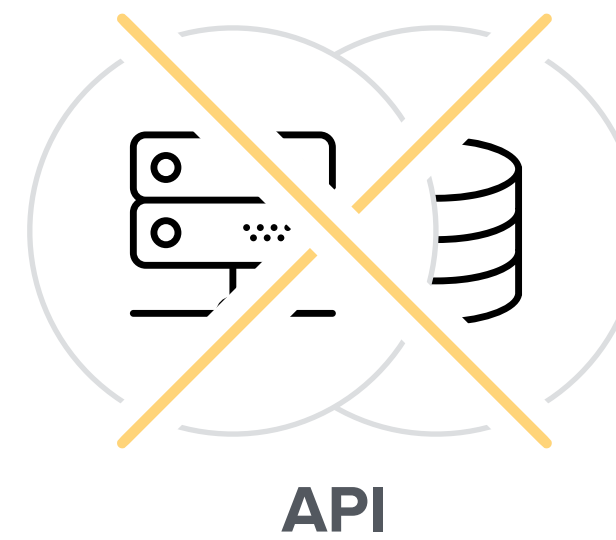
SYSTEM SETUP / MIGRATIONS

3rd PARTY SERVICES MOCKS

### 3. FAKE API STUBS



LOCAL



API

OFFLINE

EASY DATA SETUP

MAINTENANCE COST

**INTEGRATING WITH A WEB API  
IS NOT ALWAYS TRIVIAL**

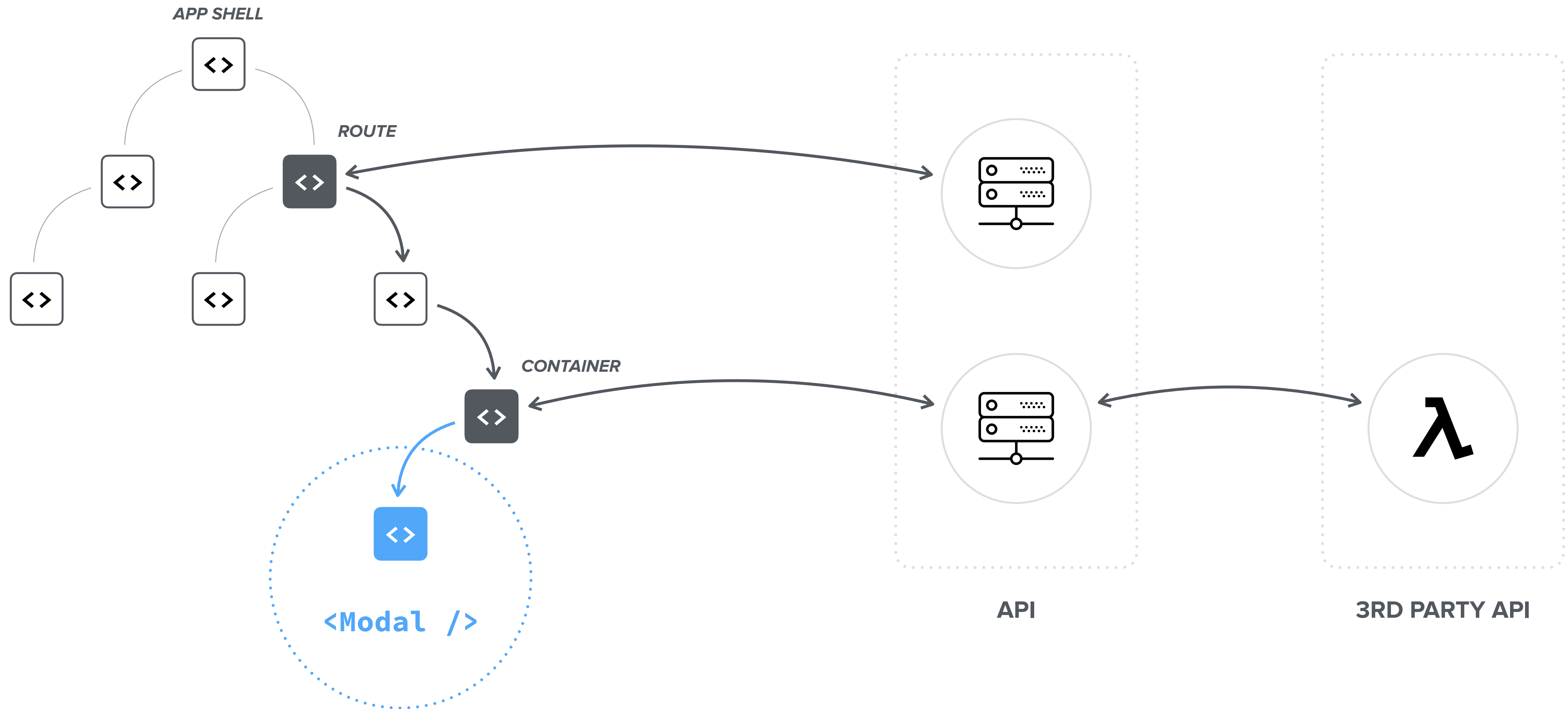


DATA DEPENDENT

COMPONENTS

4

# DATA DEPENDENT COMPONENTS



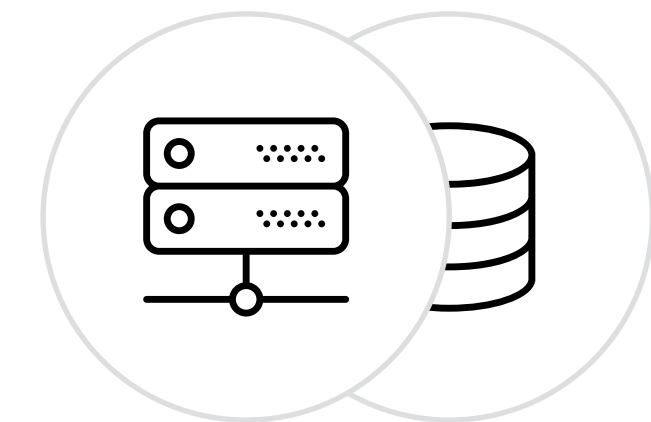
# UI ISOLATION



# UI ISOLATION



UI



API

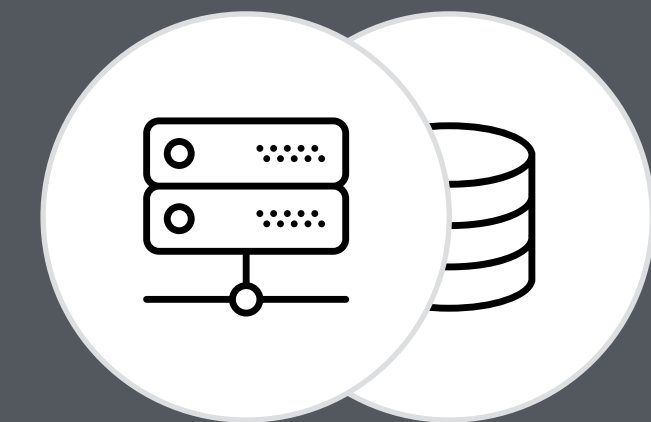
DB



# UI ISOLATION



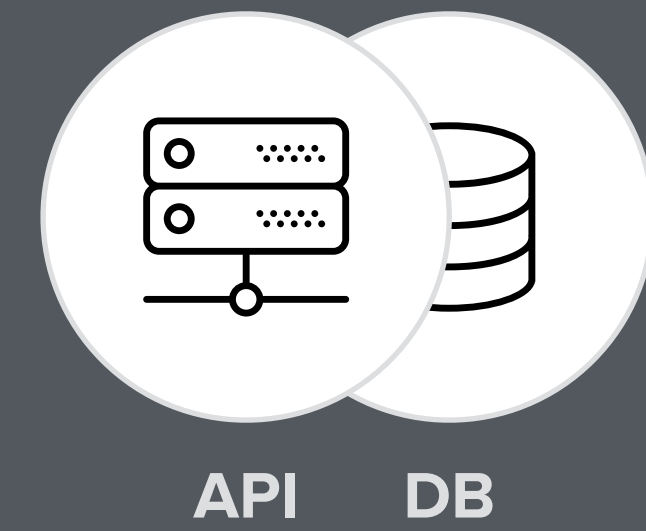
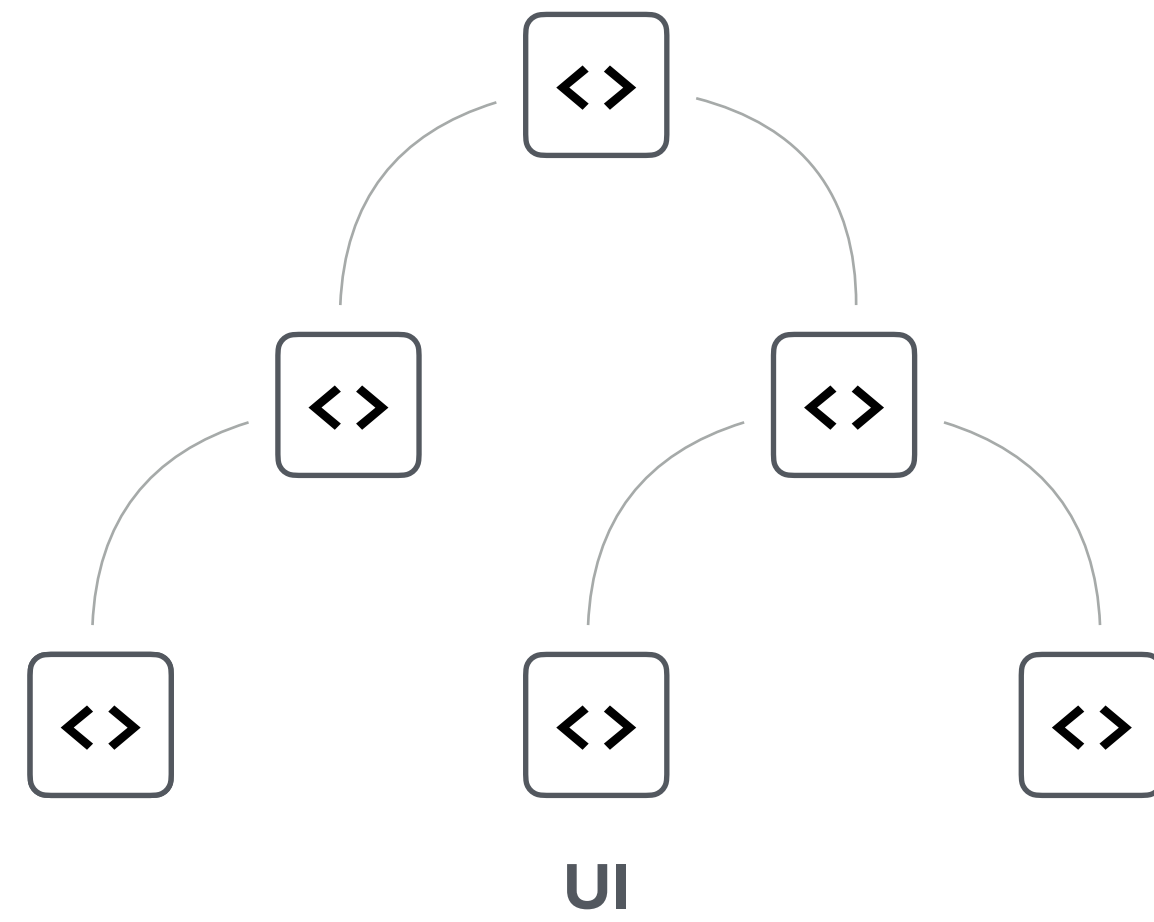
UI



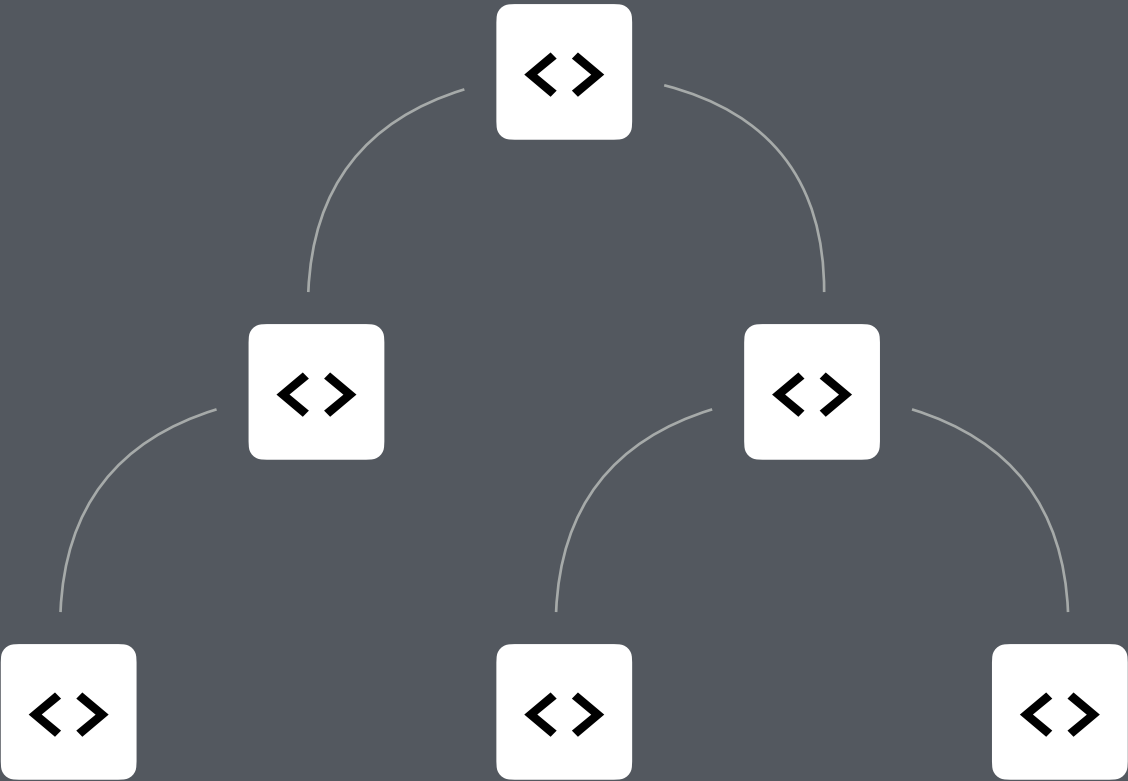
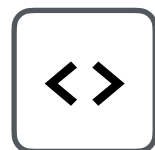
API

DB

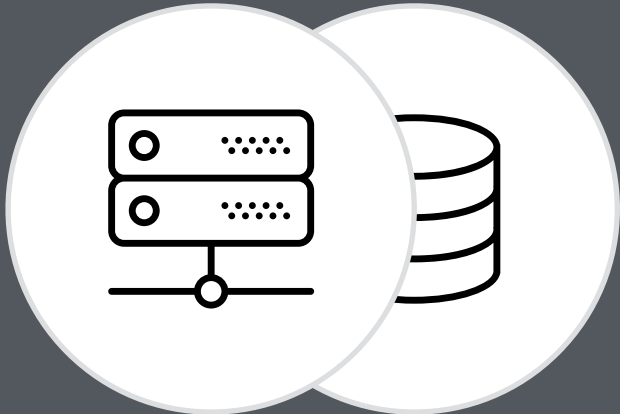
# UI ISOLATION



COMPONENT  
ISOLATION



UI

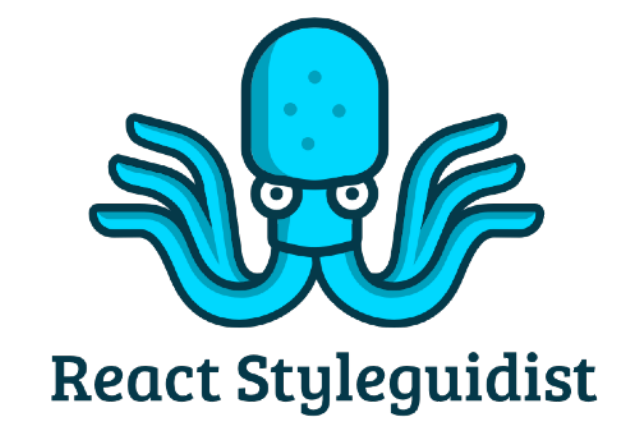


API DB

**HOW**

**BUILD YOUR OWN**

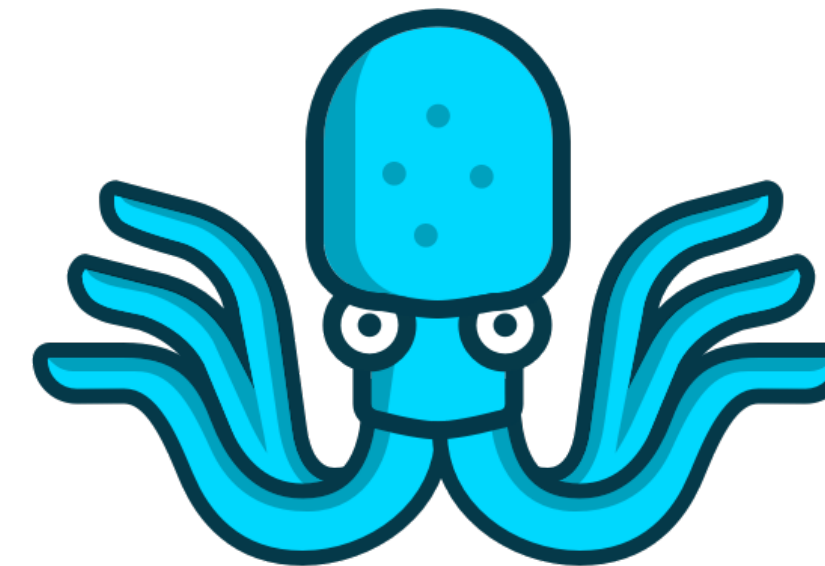
*andreipfeiffer.dev*





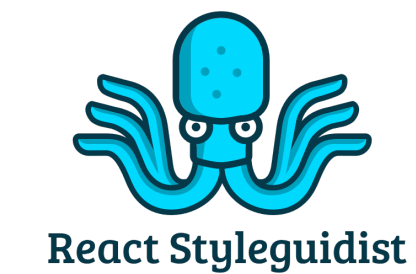
**Storybook**

*OPEN SOURCE TOOL FOR  
BUILDING UI COMPONENTS AND PAGES IN ISOLATION*



**React Styleguidist**

*ISOLATED REACT COMPONENT DEVELOPMENT ENVIRONMENT  
WITH A LIVING STYLE GUIDE*



---

**// src/components/Button.stories.jsx**

```
import { Button } from "../Button";

export default {
  component: Button,
  title: "components/Button",
};

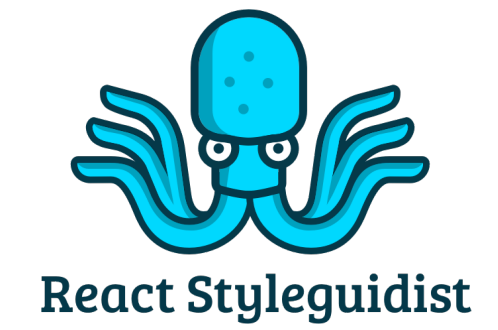
const Template = (args) => <Button {...args} />;

export const Primary = Template.bind({});
Primary.args = { primary: true, children: "Submit" };
```

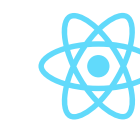
**// src/components/Button.md**

```
```jsx
<Button primary>Submit</Button>
```
```





OFFICIAL INTEGRATIONS



WRITING EXAMPLES



PLUGINS

YES

-

PRIMARY PURPOSE

DEVELOPMENT & DOCS

DOCUMENTATION

TYPESCRIPT SUPPORT

FULL

LIMITED



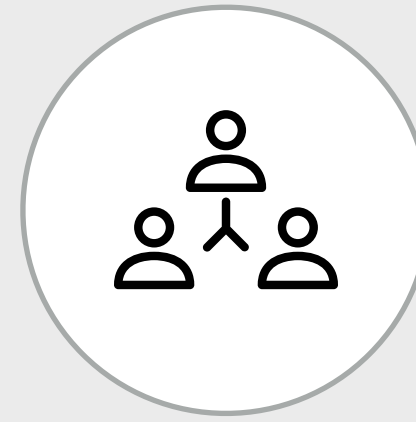
**INCREMENTAL**

**ADOPTION**



## **LOCAL SETUP**

USE IT WHEN NEEDED



## **CHECK IN REPO**

SHARED WITH THE TEAM



## **STATIC BUILD**

PUBLISHED ONLINE

1

2

3

**ADDED**

**BENEFITS**

```
export class Project extends React.Component {
  render() {
    return (
      <Loading isLoading={() => this.props.isFetchingProject}>
      {
        this.state.noProjectFound ? (
          <div>
            No project found for ID <strong>{this.props.match.params.projectId}</strong>
            <br />
            <Link to={`/projects/new`}>Add a new project</Link>
          </div>
        ) : ''
      }
    )
  }
  {
    this.props.project ? (
      <AppHeader
        items={['my-projects', 'project', 'analysis']}
        criteria={this.props.match.params}
        projectId={this.props.projectId}
        project={this.props.project}
        analysis={this.props.analysis}
      />
      <Badge color="grey" variant="outline" inline>
        Inactive
      </Badge>
      <Text secondary>
        Enforced only
        &nbsp;
        <Popup
          trigger={<Icon circular name="question" size="small" />}
          content='Only enforced rules with issues found will fail your build.'
        />
        <Checkbox
          toggle
          checked={this.state.filter.status === 'enforced'}
          onChange={(e, d) => {
            this.setState({ filter: { ...this.state.filter, status: d.checked ? 'enforced' : 'all' } })
          }}
        />
      </Menu.Item>
      <Menu.Item fitted position="right">
        With issues only
        <Checkbox
          toggle
          checked={this.state.filter.onlyIssues}
          onChange={(e, d) => {
            this.setState({ filter: { ...this.state.filter, onlyIssues: d.checked } })
          }}
        />
      </Menu.Item>
      {this.props.readOnly ? '' : (
        <Menu.Item fitted position="right">
          <div>
            <Button.Group basic>
              <Button
                toggle
                active={this.state.editEnabled}
                onClick={this.toggleEditMode}
              >
                Edit
              </Button>
              {this.addRuleButton()}
            </Button.Group>
          </div>
        </Menu.Item>
      )}
    </Menu>
  </AppHeader>
) : ('')
}

{
  this.props.project ? (
    <Container>
      <Card.Group itemsPerRow={3} stackable doubling>
        {
          this.renderRules()
        }
      </Card.Group>
    </Container>
  ) : ('')
}
</Loading>
)
}
}
```





















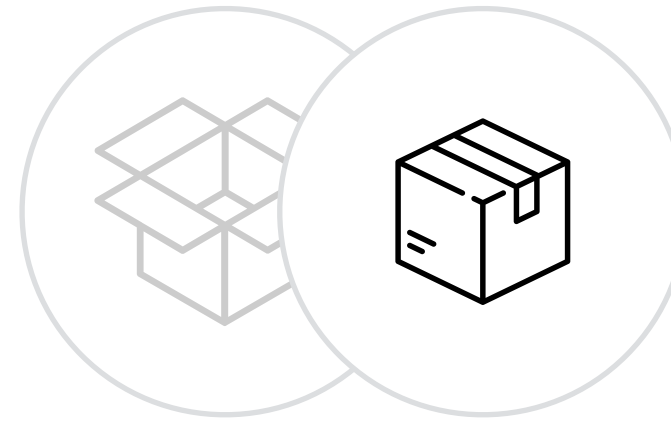
FOCUS ON **COMPONENT'S API**



# SHARED **DOCUMENTATION**



# **INTERACTIVE PLAYGROUND**



# **CONTAINER / PRESENTATIONAL**

COMPONENT SEPARATION



# **COMPONENTS LIBRARY**

STYLE GUIDE / DESIGN SYSTEM

# CREDITS

Icons from **flaticon.com**

by Good Ware, itim2101, Freepik, Smashicons

Images from **Pixabay**

by Free-Photos, Ilona Ilyés, CarlosAlcazar

THANK YOU



*andreipfeiffer.dev*