ISOLATED COMPONENTS DRIVEN DEVELOPMENT

WHAT

WHY

HOW

WHO

ANDREI PFEIFFER

[e-spres-oh]

CODE DESIGNER

TECHNICAL WRITER

tim.js

revo·js

ORGANIZER

WHAT

DEVELOPMENT TECHNIQUE

```
component = f(props)

f(props) === // unit
```

WHY

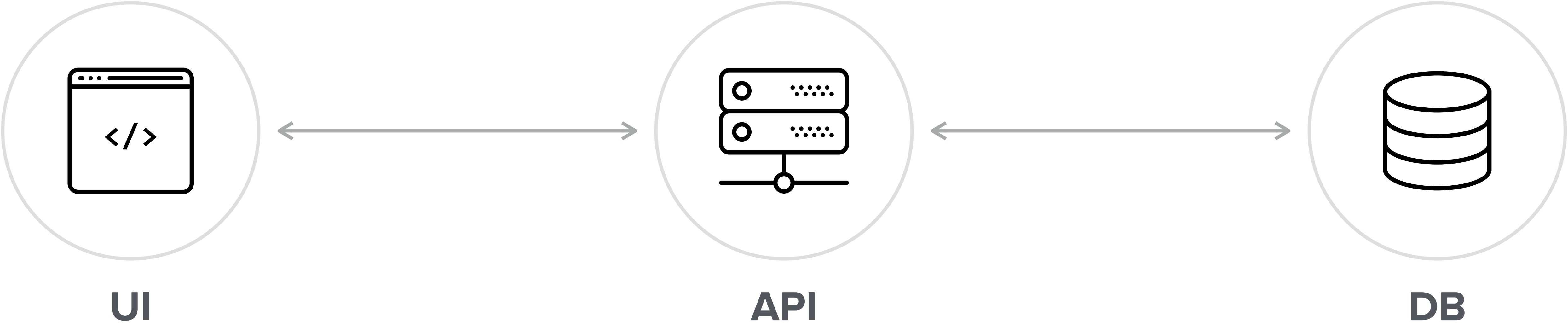
COMPONENTS LIBRARY

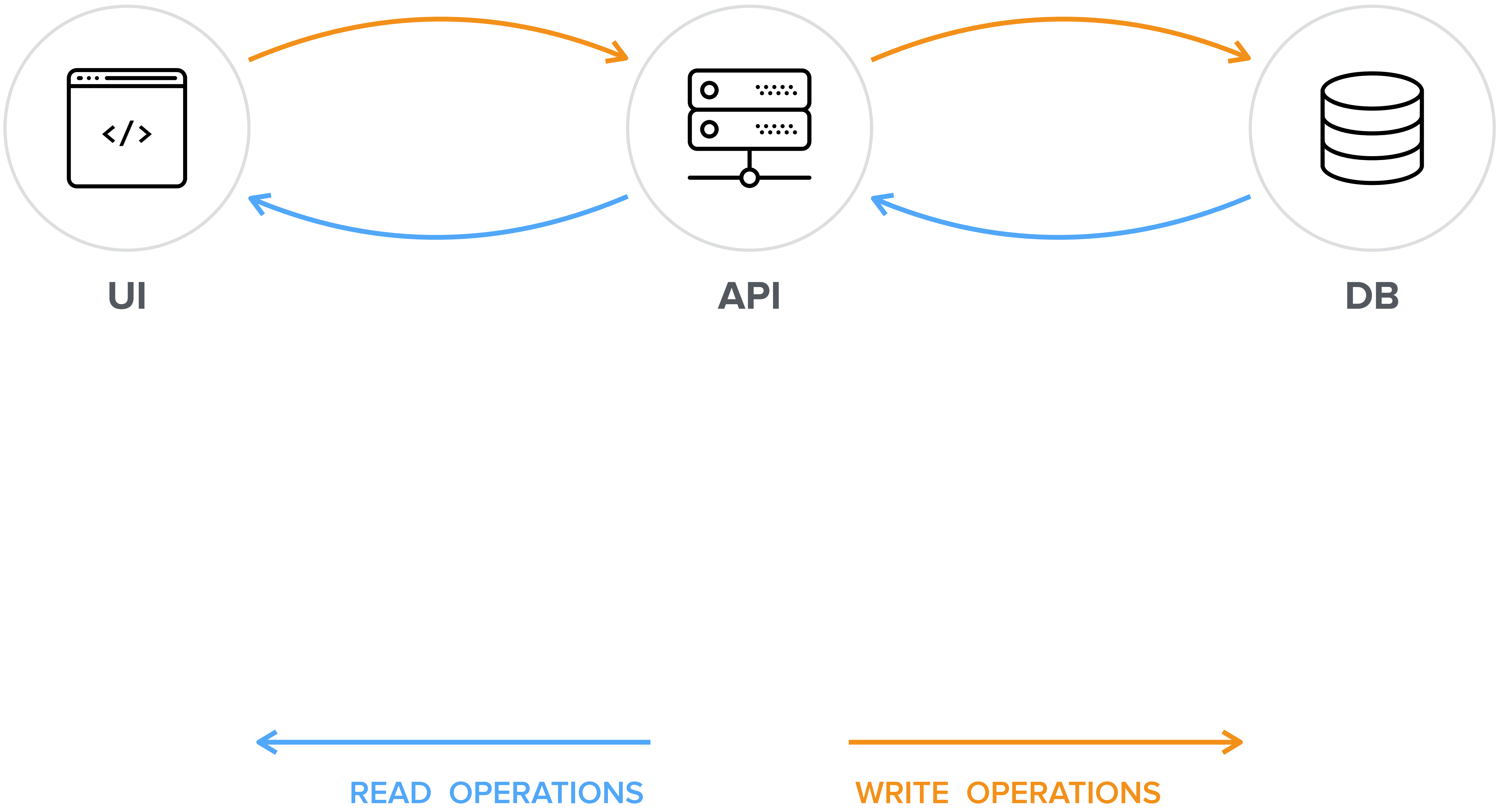
NEED A PLACE TO RENDER & USE THEM

1

CRUD UIs

2


UI
API
DB



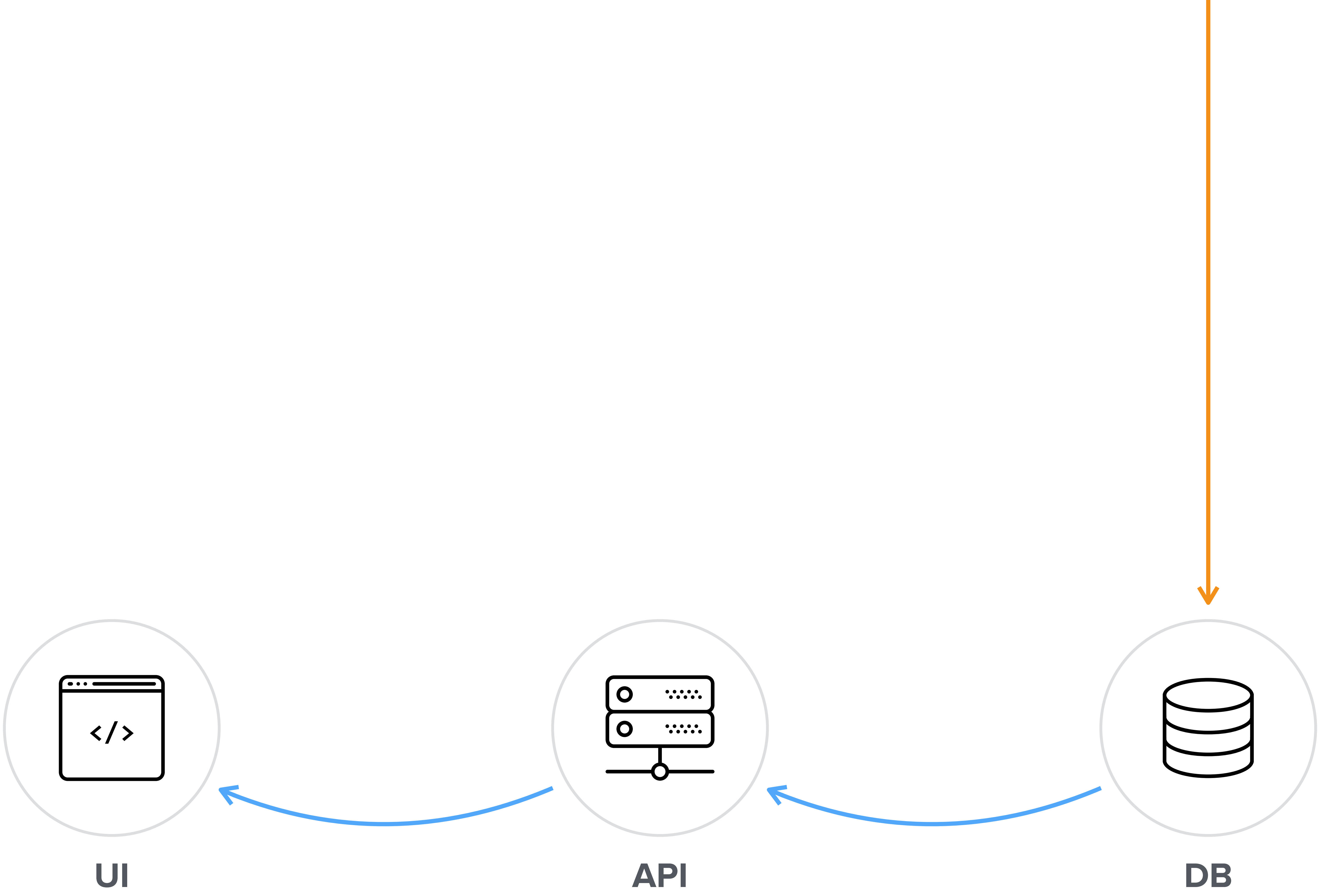
UI

API

DB

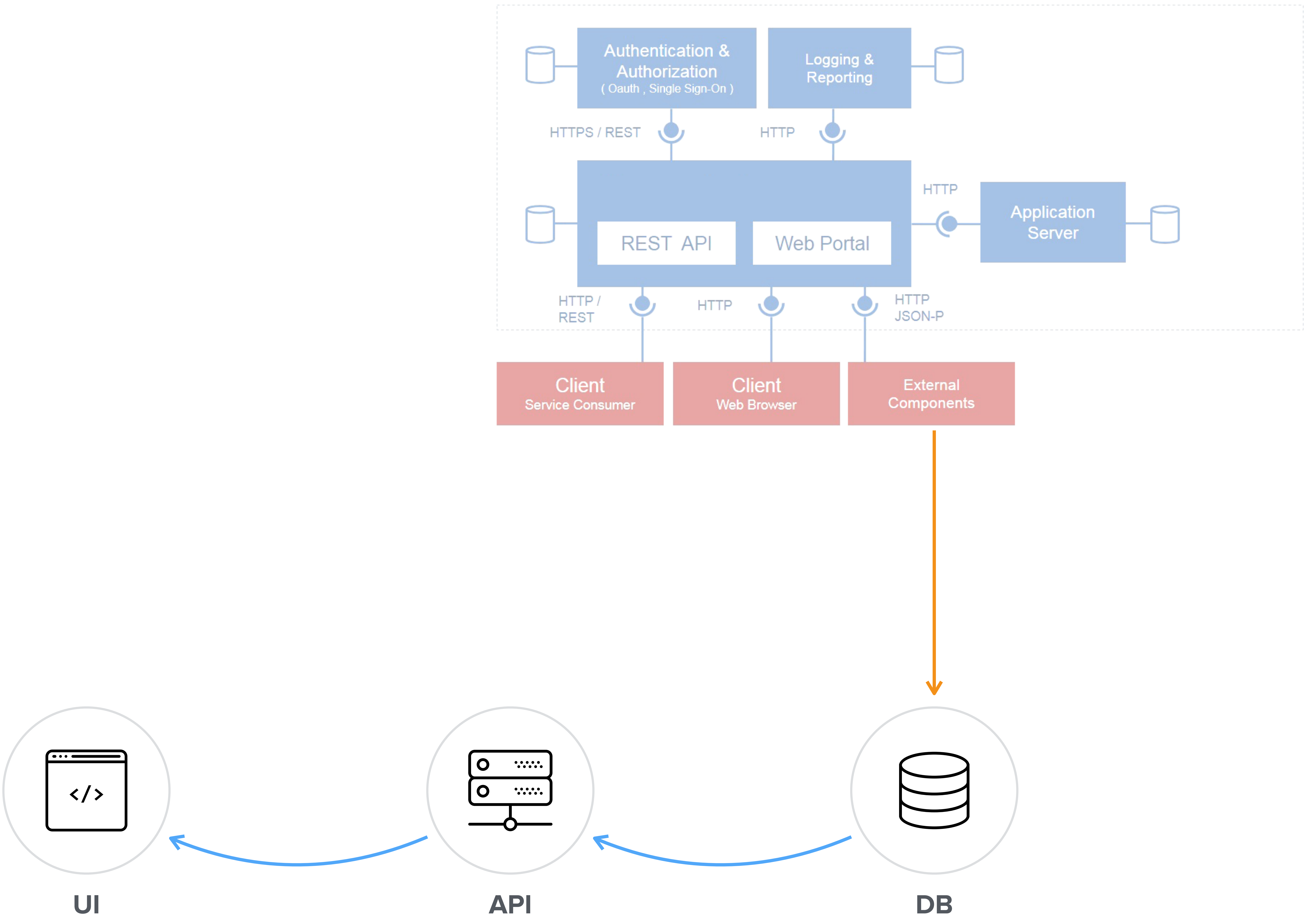
READ OPERATIONS

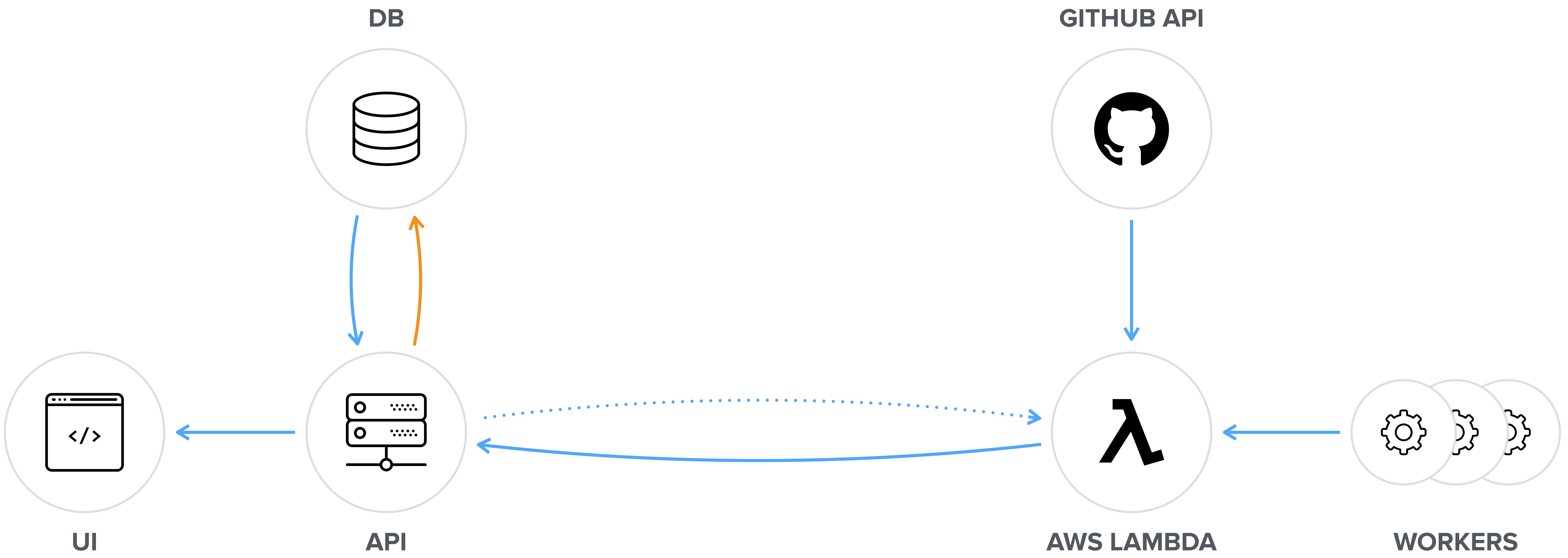
WRITE OPERATIONS

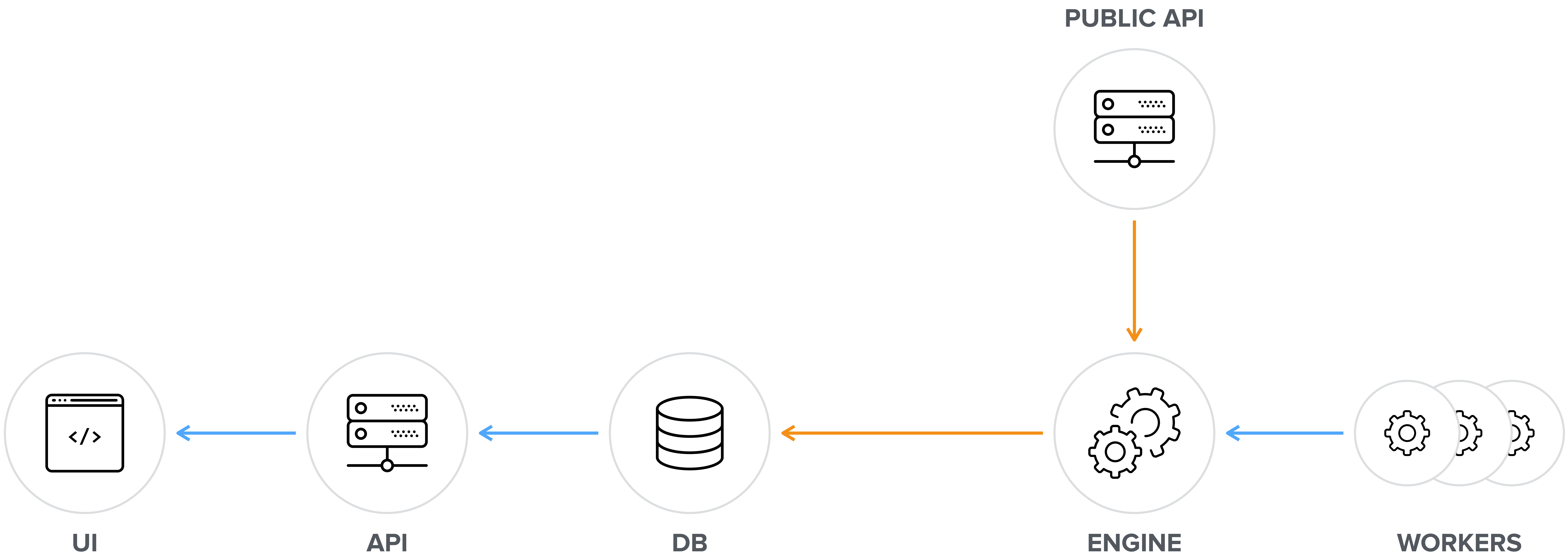

UI
API
DB

Google Analytics

GitHub




DB
GITHUB API
UI
API
AWS LAMBDA
WORKERS


PUBLIC API
UI
API
DB
ENGINE
WORKERS

DEALING WITH COMPLEX USER INTERFACES

… IS COMPLEX

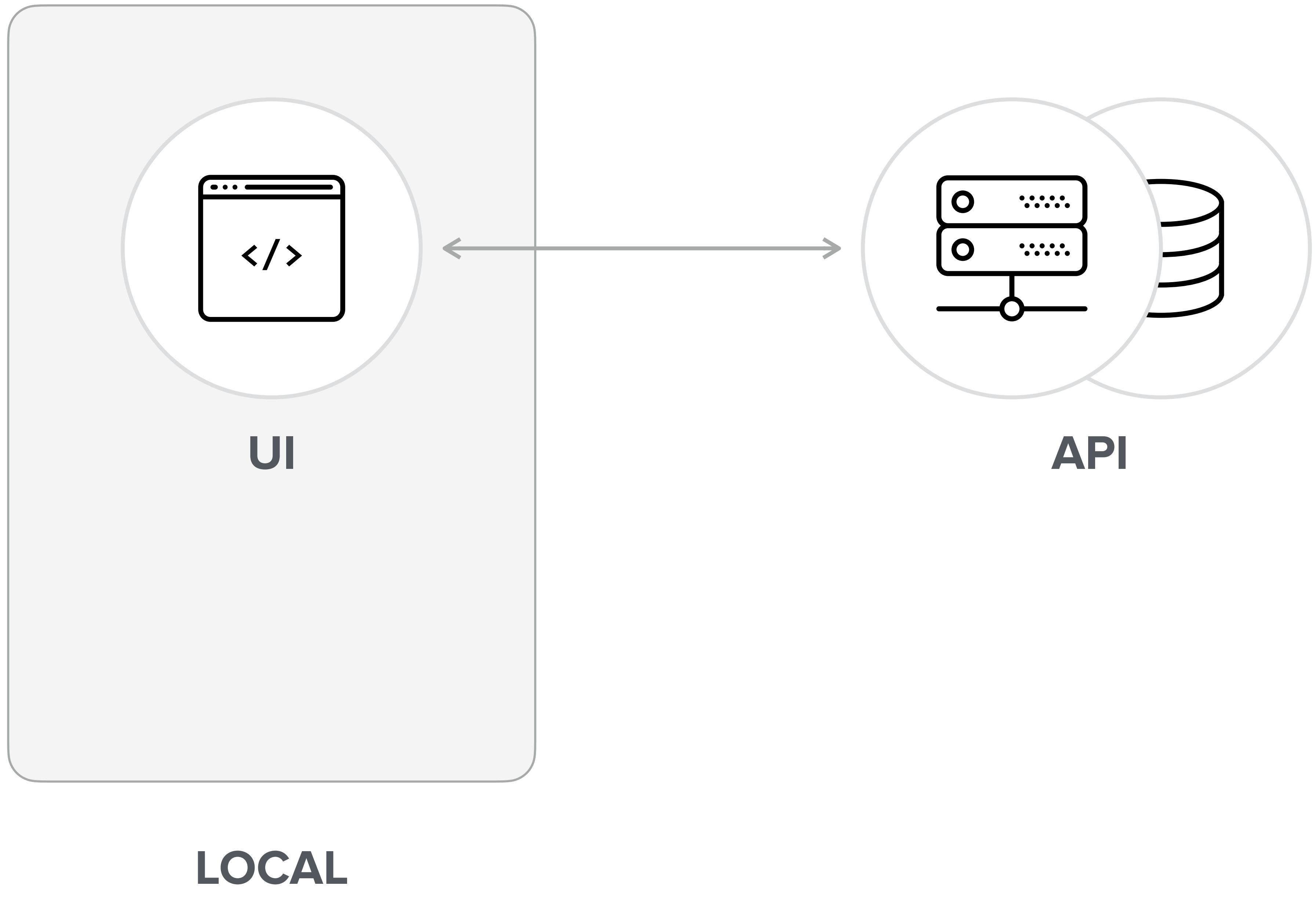
UI API CONNECTION

3

UI

API

1. REMOTE CONNECTION TO DEV API



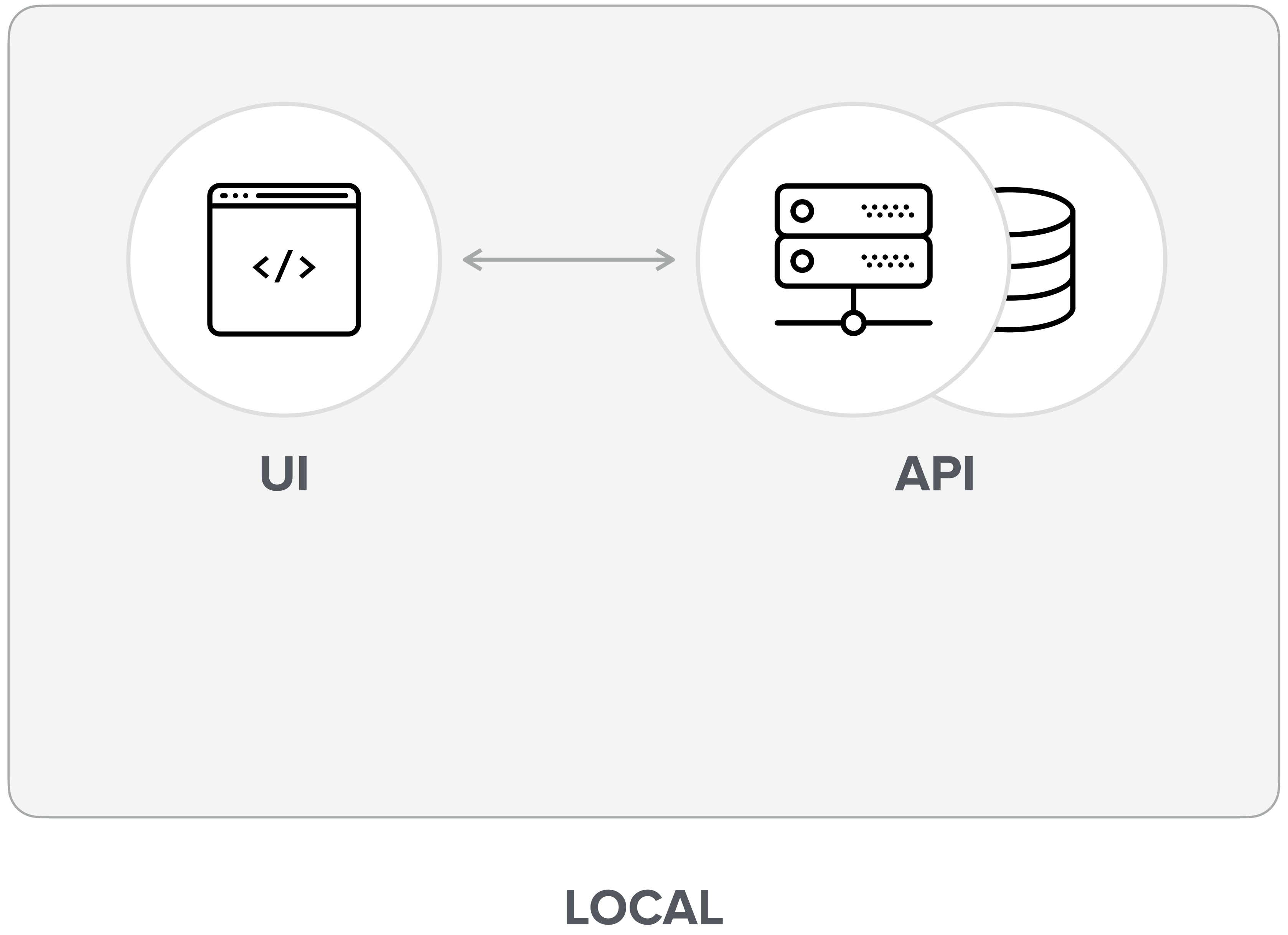
NO LOCAL API SETUP

SHARED DATA

INFRASTRUCTURE DEPENDENCY

ONLINE ONLY

2. LOCAL API SETUP



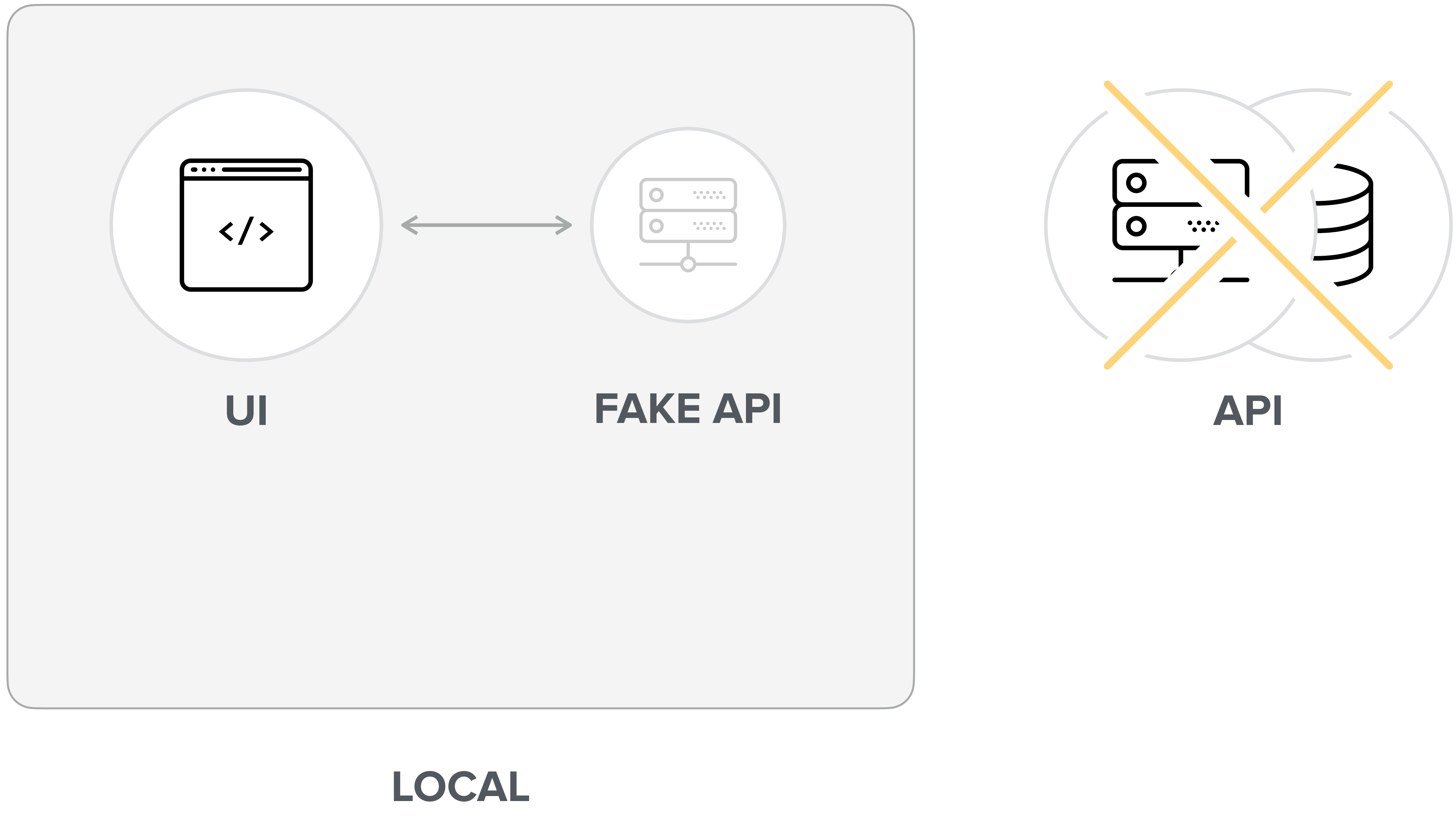
OFFLINE

EASILY SWITCH BRANCHES

NON-TRIVIAL DATA SYNC

SYSTEM SETUP / MIGRATIONS

3. FAKE API STUBS



OFFLINE

EASY DATA SETUP

NON-TRIVIAL SCHEMA UPDATES

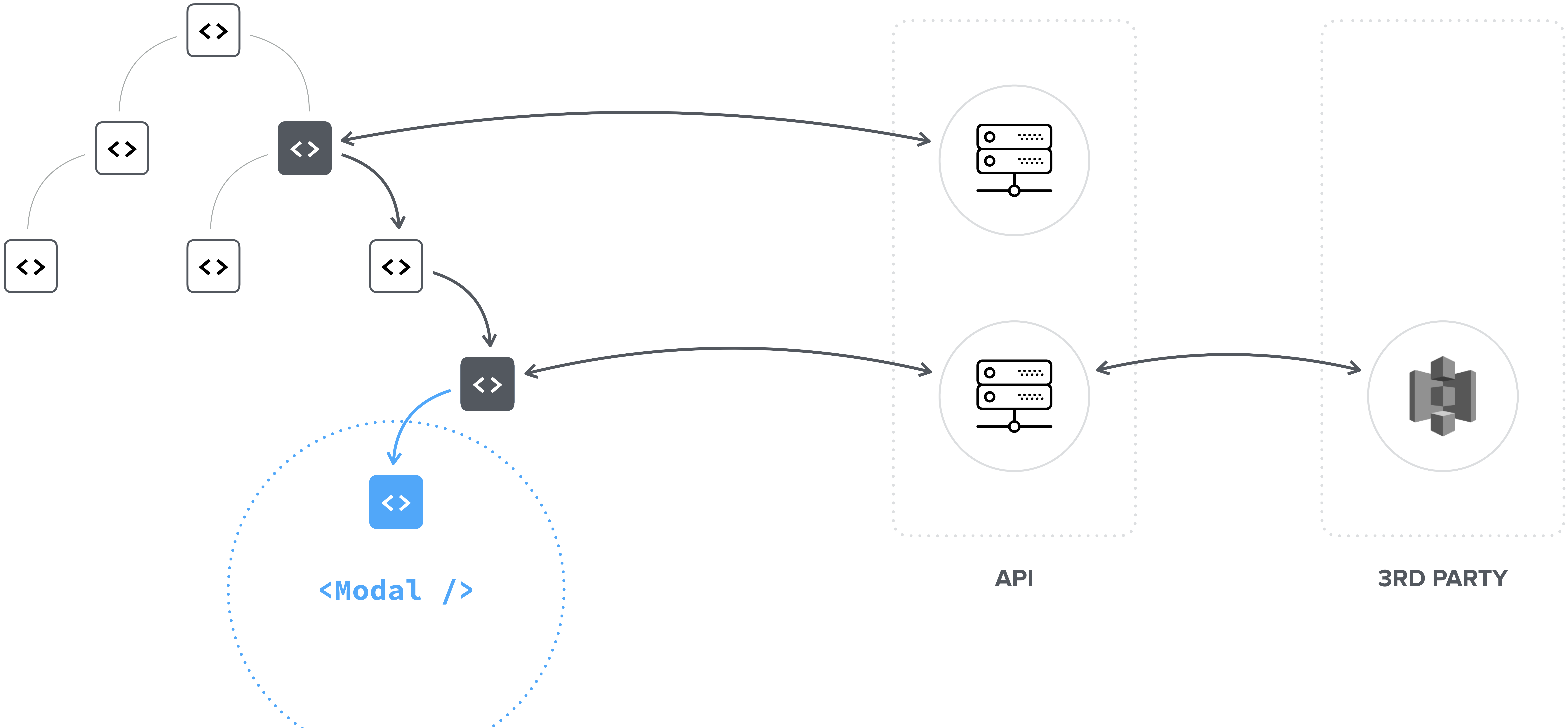
INTEGRATING WITH YOUR API

… IS NOT TRIVIAL

DATA DEPENDENT COMPONENTS

4

DATA DEPENDENT COMPONENTS



COMPONENT ISOLATION

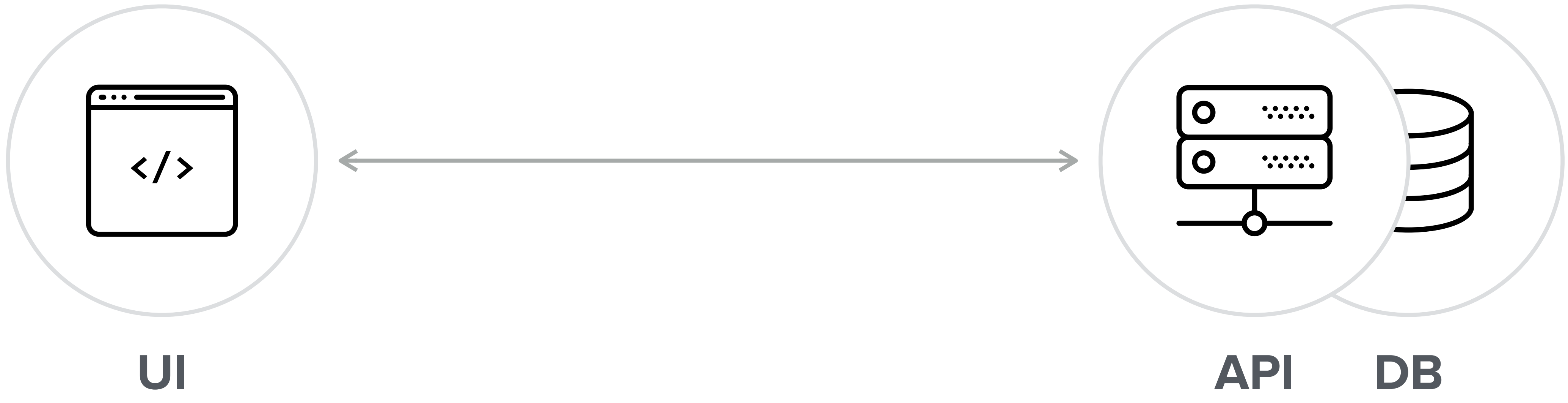
RENDER THE COMPONENT OUTSIDE THE APPLICATION

```
<Modal src={image} />
```

```
<Modal loading={true} />
```

" …like Unit Testing "

UI ISOLATION



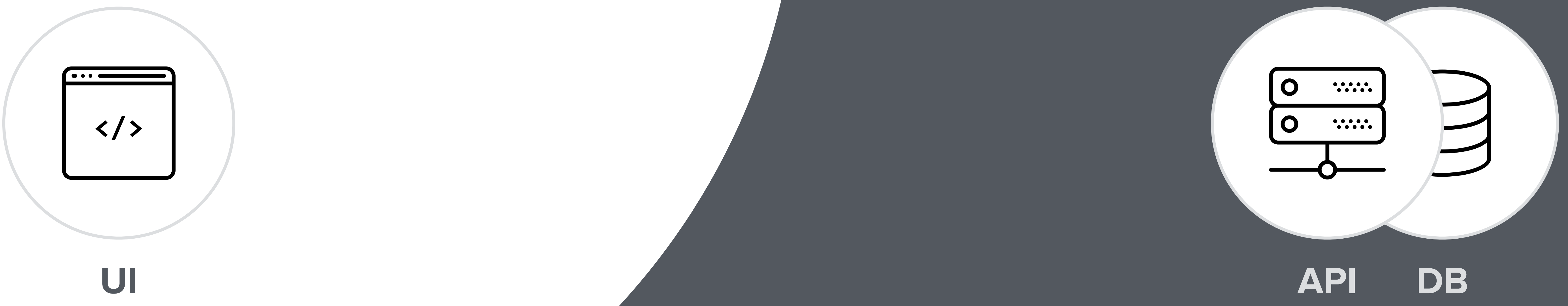
UI ISOLATION

UI

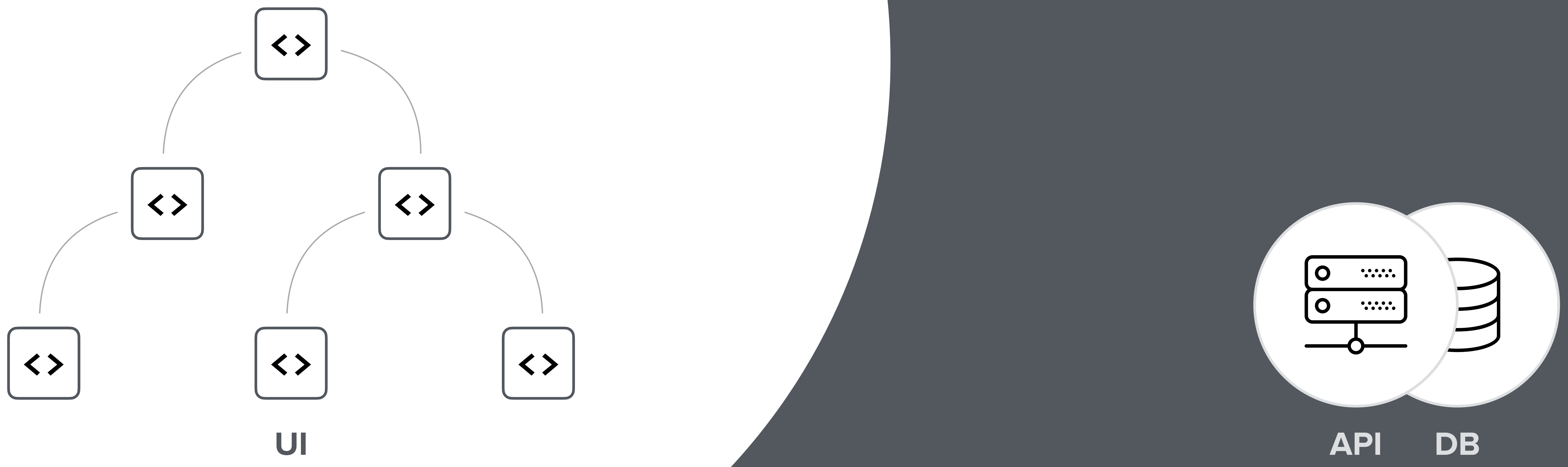
API

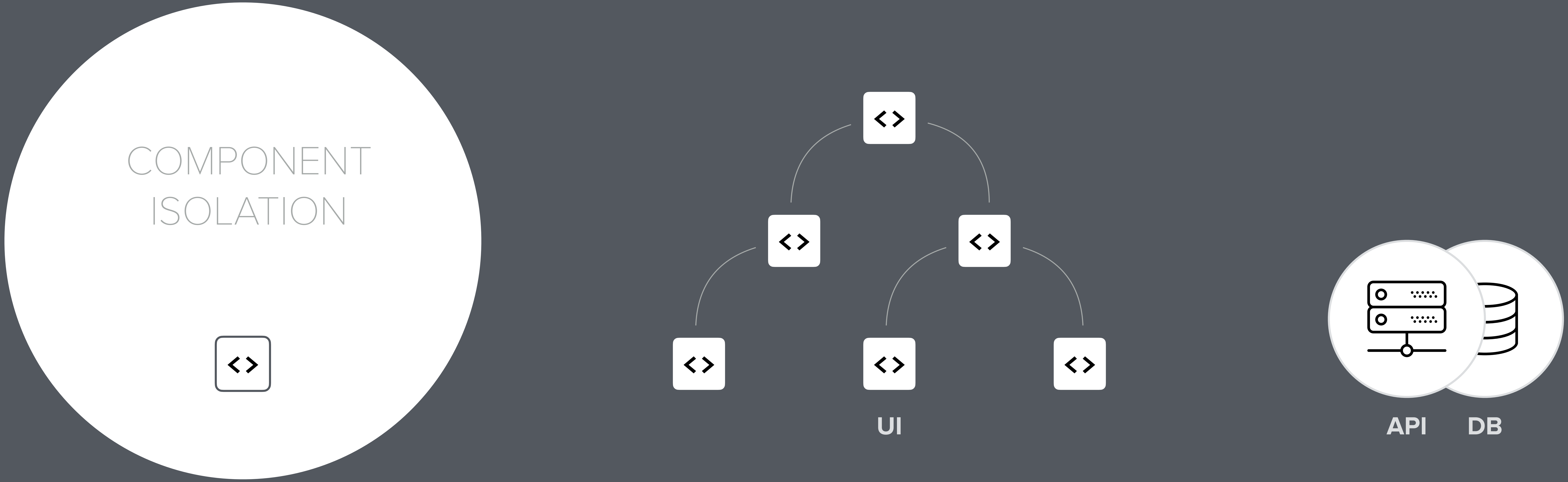
DB

UI ISOLATION



UI ISOLATION





COMPONENT ISOLATION

UI

API

DB

HOW

Storybook

docz

React Cosmos

React Styleguidist

Storybook

OPEN SOURCE TOOL
FOR DEVELOPING UI COMPONENTS IN ISOLATION

React Styleguidist

ISOLATED REACT COMPONENT DEVELOPMENT ENVIRONMENT
WITH A LIVING STYLE GUIDE

Storybook

```
// .storybook/config.js

import { configure } from "@storybook/react";
// include any other global styles
import "bootstrap/bootstrap.min.css";

function loadStories() {
  // specify stories files
  require("./stories/index.js");
}

configure(loadStories);
```

React Styleguidist

```
// styleguide.config.js

module.exports = {
  // specify which components to include
  components: "src/components/**/*.js",
  // include any other global styles
  require: ["bootstrap/bootstrap.min.css"],
};
```

Storybook

```
// .storybook/stories/Modal.js

import React from "react";
import { storiesOf } from "@storybook/react";

import Modal from "../../src/components/Modal";

storiesOf("Components", module)
  .add("Modal", () => (
    <>
      <Modal src={/* data */} />
      <Modal loading />
    </>
  ));
```

React Styleguidist

~~~~
// src/components/Modal.md

```jsx
<>
  <Modal src={/* data */} />
  <Modal loading />
</>
```
~~~~

	Storybook	React Styleguidist
INTEGRATIONS	ember 5 ⚛ A V RIOT	⚛ V
WRITING EXAMPLES	JAVASCRIPT	MARKDOWN
PLUGINS	YES	-
CUSTOM LAYOUT	-	YES
MAIN PURPOSE	DEVELOPMENT	DOCUMENTATION
TYPESCRIPT SUPPORT	FULL	LIMITED

INCREMENTAL ADOPTION

LOCAL SETUP

USE IT WHEN NEEDED

1

CHECK IN REPO

SHARED WITH YOUR TEAM

2

STATIC BUILD

PUBLISHED ONLINE

3

```
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}
              activeStep='analysis'
            >
              <Badge color="grey" variant="outline" inline>
                Inactive
              </Badge>
              <Menu secondary>
                <Menu.Item fitted position="right">
                  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>
    )
  }
}
```

```
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}
          activeStep='analysis'
        >
          <Badge color="grey" variant="outline" inline>
            Inactive
          </Badge>
          <Menu secondary>
            <Menu.Item fitted position="right">
              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' } })
                }}
              />
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}
          activeStep='analysis'
        >
          <Badge color="grey" variant="outline" inline>
            Inactive
          </Badge>
          <Menu secondary>
            <Menu.Item fitted position="right">
              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' } })
                }}
              />
```

```jsx
<Badge
  color="grey"
  variant="outline"
  inline
>
  Inactive
</Badge>
```

FOCUS

BETTER COMPONENT API

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**

THANK **YOU**

andreipfeiffer.dev