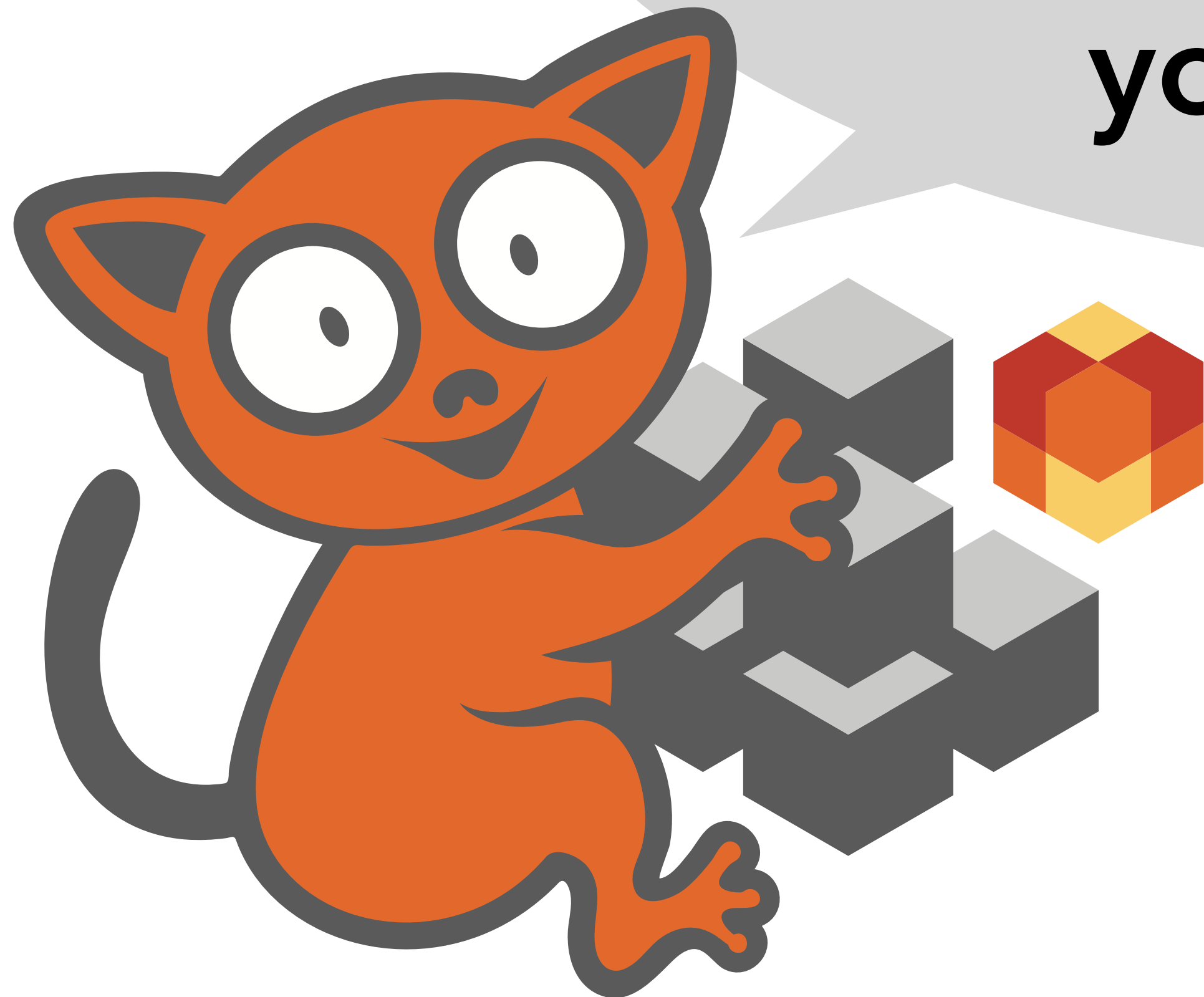




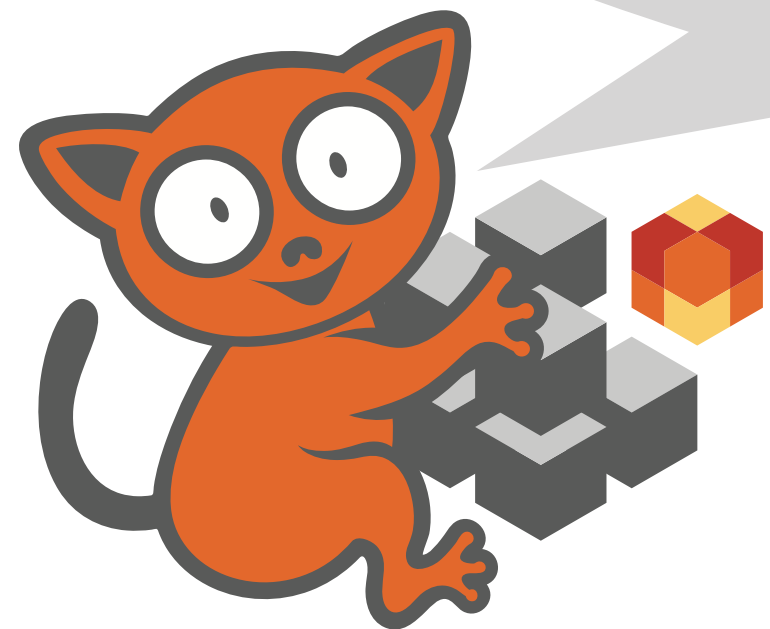
**How to implement E2EE in
your app in 50 minutes**



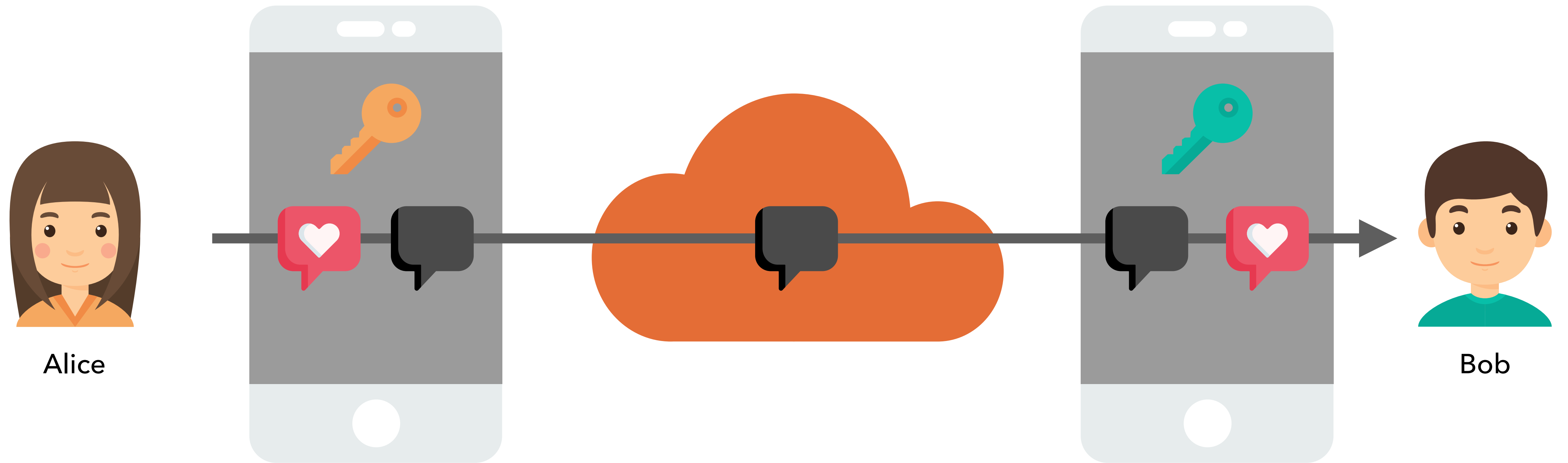
Henri Binsztok

WALLIX

The Vision



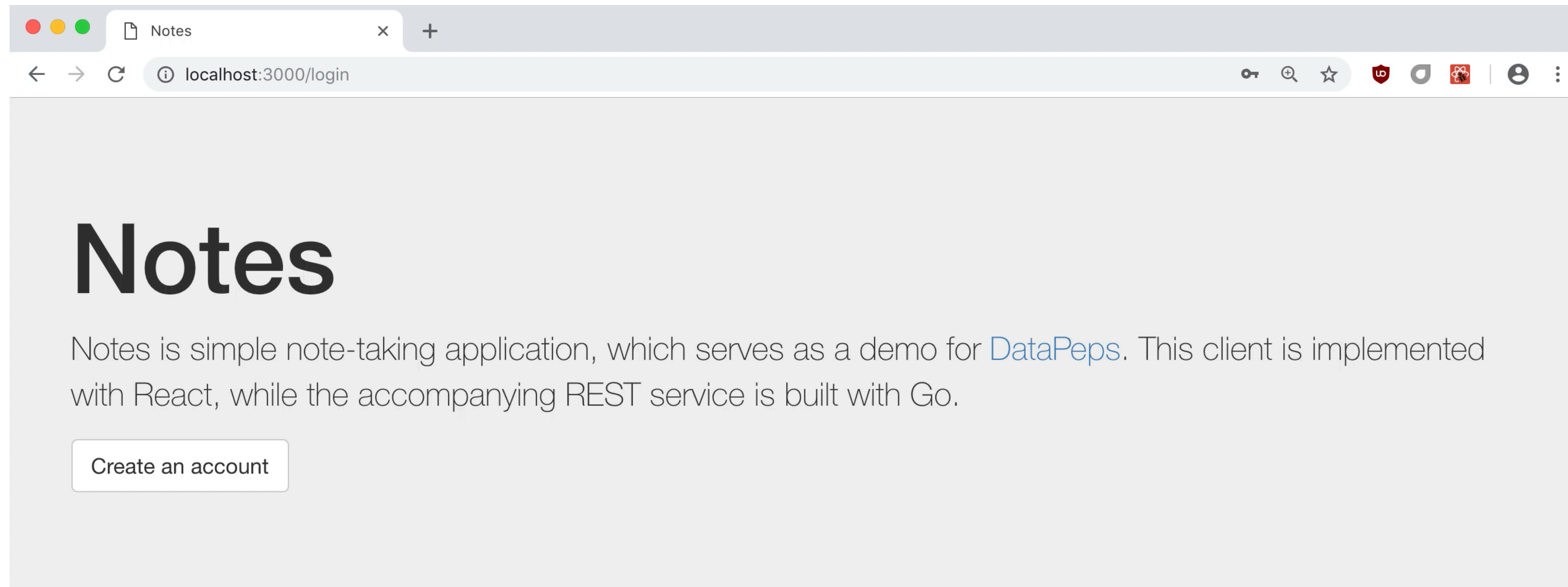
End-to-End Encryption



Alice

Bob

Our Demo App



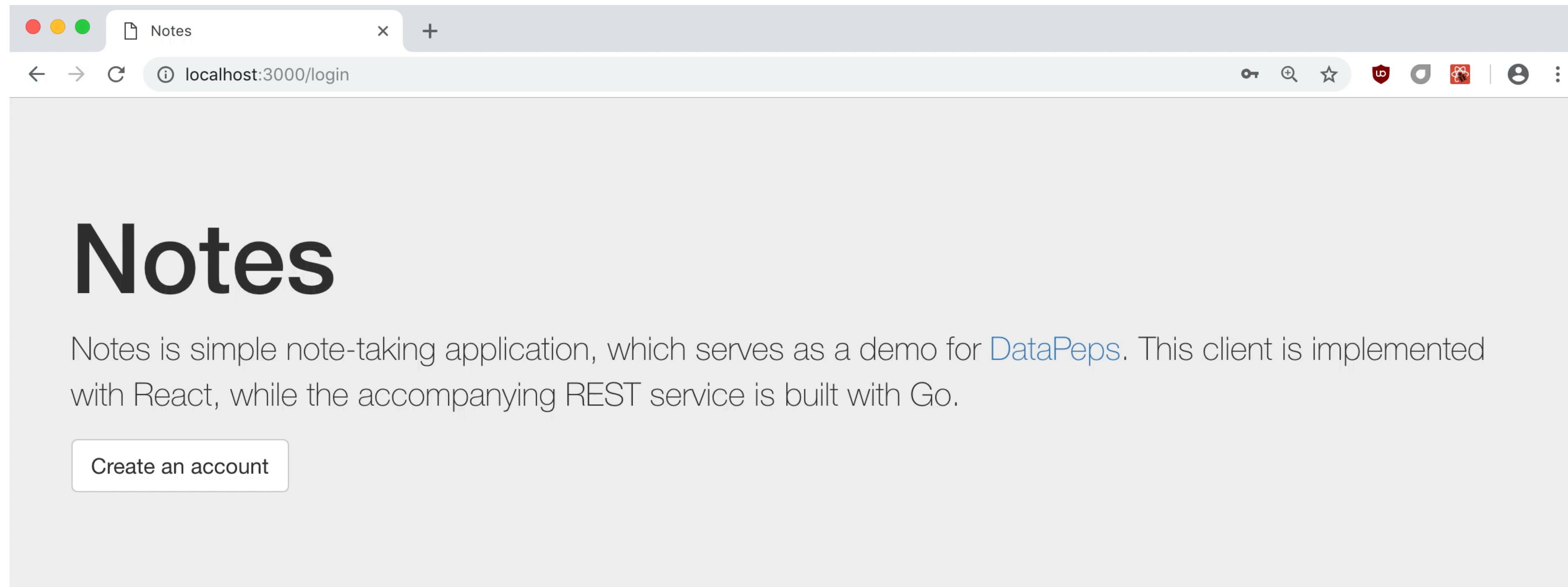
Login

Username

Password

Login

Our Demo App



Login

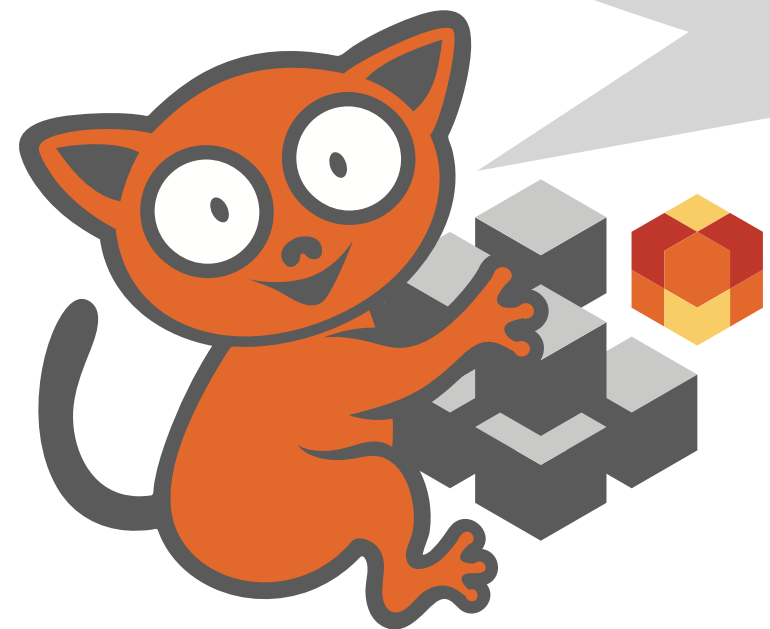
Username

Password

Login

The Tour

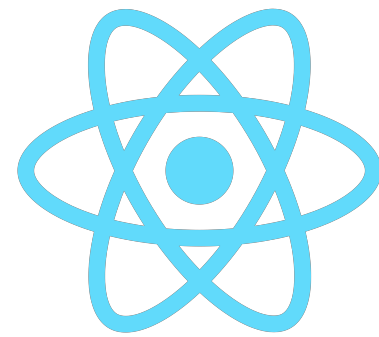
Notes Frontend



Front-end Technologies



JavaScript



React



Redux



Bootstrap

Small project: 1200 LoC

package.json

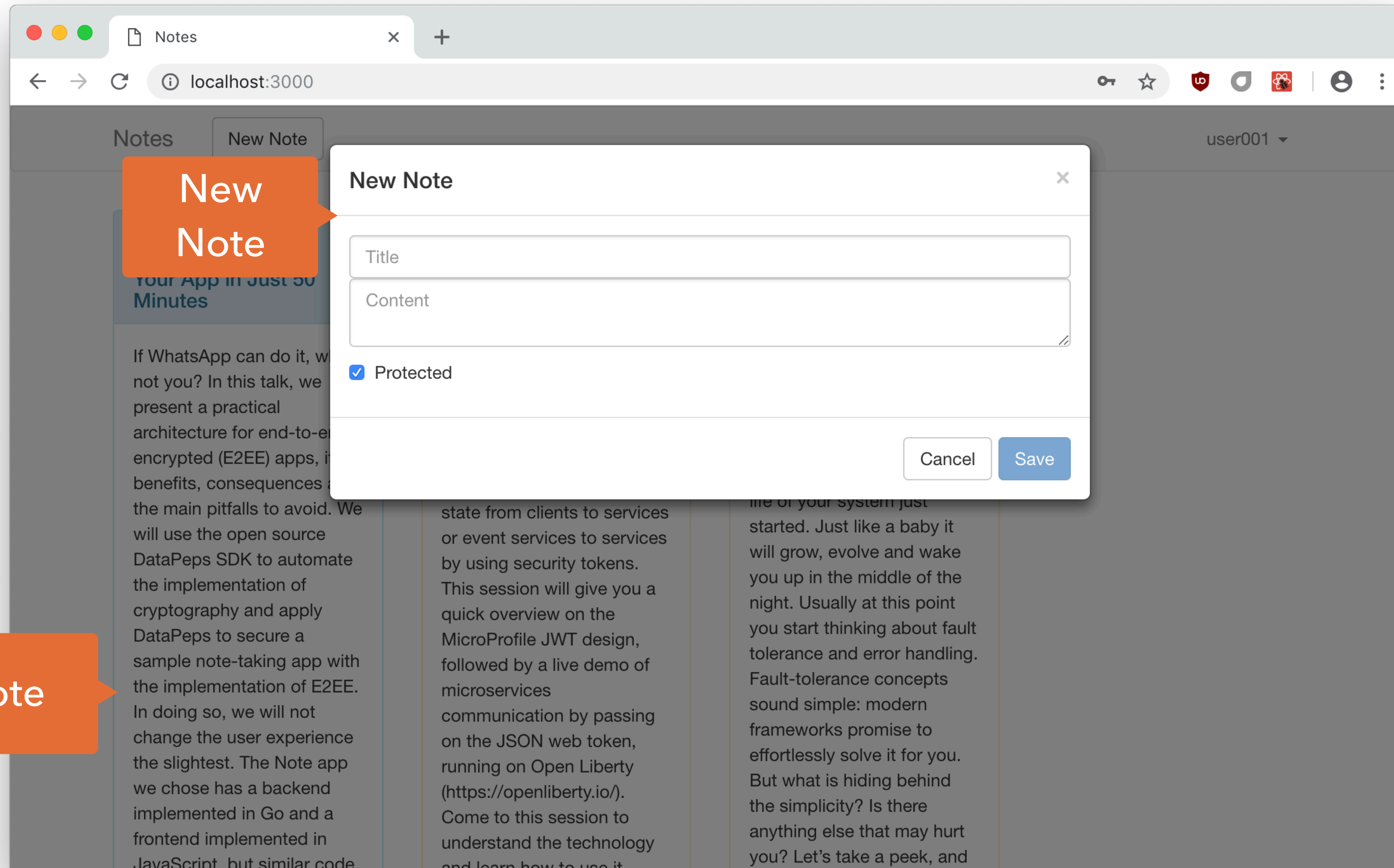
```
"devDependencies": {  
  "react-scripts": "^2.0.5"  
},  
"dependencies": {  
  "history": "^4.7.2",  
  "long": "^4.0.0",  
  "react": "^16.3.1",  
  "react-bootstrap": "^0.32.4",  
  "react-dom": "^16.3.1",  
  "react-redux": "^5.0.7",  
  "react-router-bootstrap": "^0.24.4",  
  "react-router-dom": "^4.3.1",  
  "redux": "^3.5.2",  
  "redux-logger": "^3.0.6",  
  "redux-thunk": "^2.3.0"  
},
```

Authentication

actions/auth.js (partial)

```
function login(username, password) {  
  return async dispatch => {  
    try {  
      const user = await  
authService.login(username, password);  
      dispatch(success(user));  
      history.push("/");  
    } catch (error) {  
      dispatch(failure(error));  
    }  
  }  
};
```


Components Overview



Note Component: Display a Note

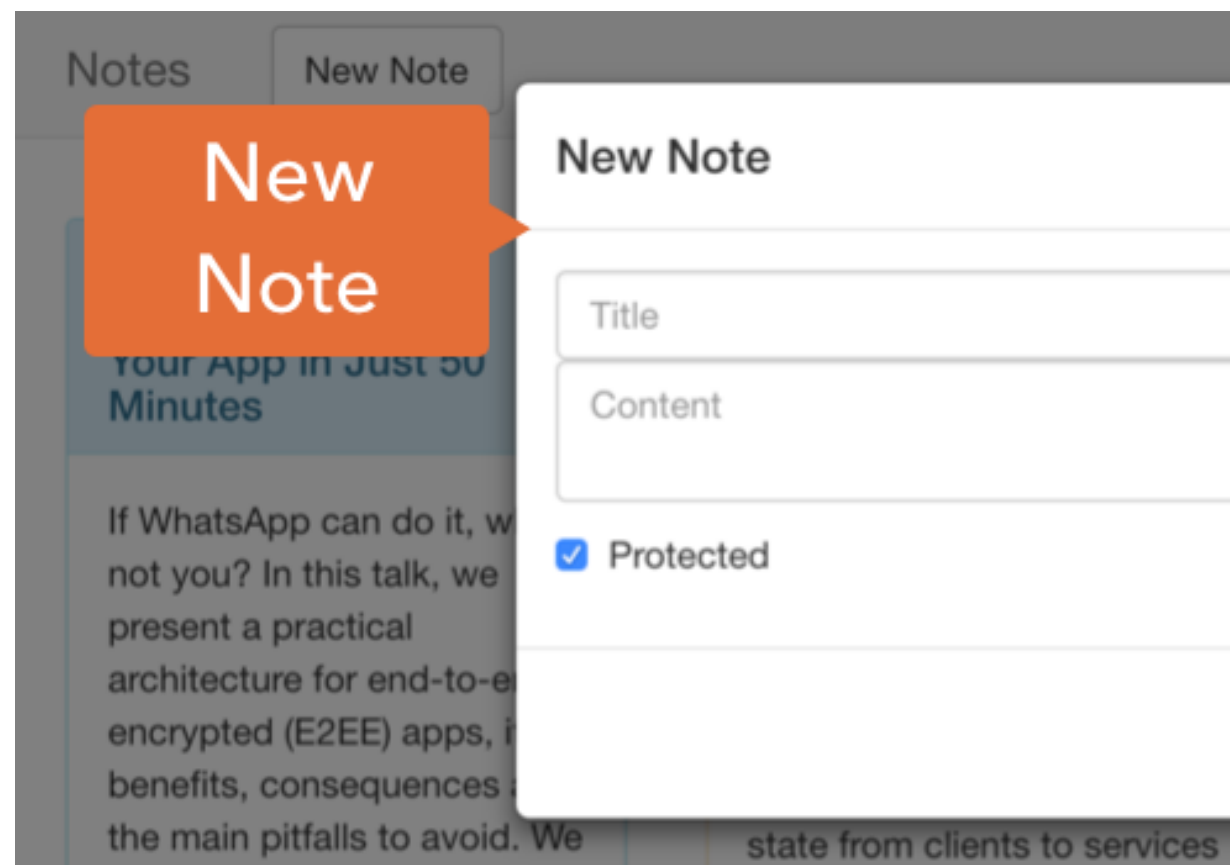
Implement End-to-End Encryption in Your App in Just 50 Minutes

If WhatsApp can do it, why not you? In this talk, we present a practical architecture for end-to-end encrypted (E2EE) apps, its benefits, consequences and the main pitfalls to avoid. We will use the open source DataPeps SDK to automate the implementation of cryptography and apply DataPeps to secure a sample note-taking app with the implementation of E2EE. In doing so, we will not change the user experience

components/note.js (partial)

```
class Note extends React.Component {  
  
  render() {  
    const { Title, Content } = this.state;  
    return (  
      <Panel>  
        <Button  
          onClick={() => deleteNote(ID)}>  
          &times;  
        </Button>  
        <Panel.Heading>  
          <Panel.Title>  
            {Title}  
          </Panel.Title>  
        </Panel.Heading>  
        <Panel.Body>  
          {Content}  
        </Panel.Body>  
      </Panel>  
    );  
  }  
}
```

New Note Component



components/newnote.js (partial)

```
render() {  
  ...  
  <Modal>  
    <Modal.Header>  
      <Modal.Title>New Note</Modal.Title>  
    </Modal.Header>  
    <Modal.Body>  
      <Form>  
        <FormControl name="title" onChange={this.changeTitle}/>  
        <FormControl name="content" onChange={this.changeContent}/>  
      </Form>  
    </Modal.Body>  
    <Modal.Footer>  
      <Button onClick={this.onAddNote} type="submit">  
        Save  
      </Button>  
    </Modal.Footer>  
  </Modal>  
  ...  
}
```

onAddNote() and posting to server

components/newnote.js (partial)

```
async onAddNote() {  
  let title = this.state.title;  
  let content = this.state.content;  
  this.props.addNote(title, content);  
}
```

event

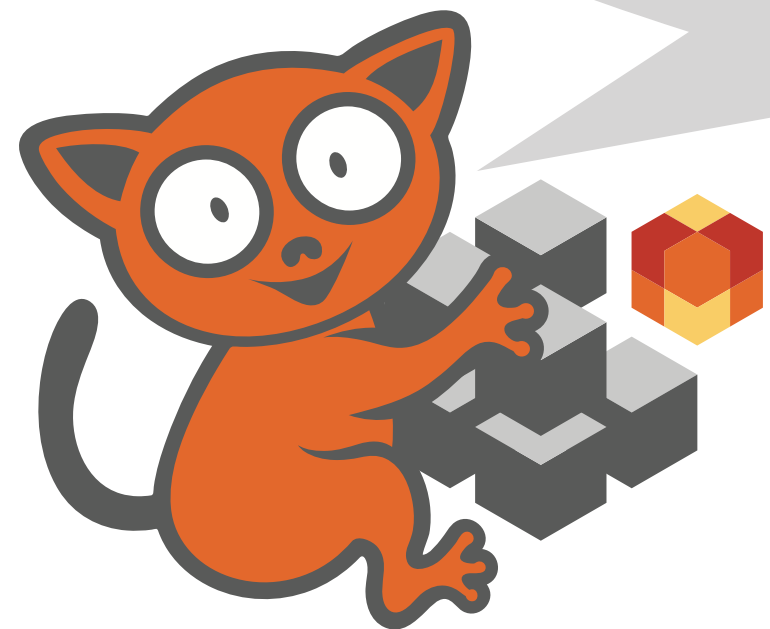
actions/notes.js (partial)

```
function postNote(note) {  
  return async dispatch => {  
    try {  
      const response = await notesService.postNote(note);  
      dispatch(success(response.noteID));  
      dispatch({ ...note, id: response.noteID });  
    } catch (error) {  
      dispatch(failure(error));  
    }  
  };  
}
```

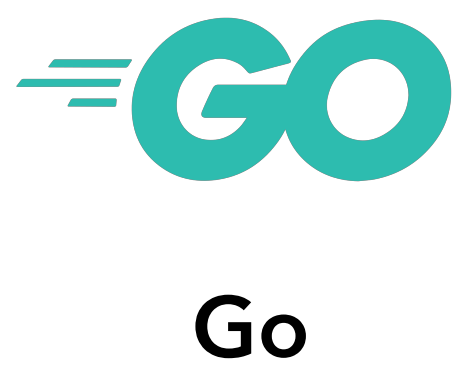
noteID from
service

The Tour

Notes Backend

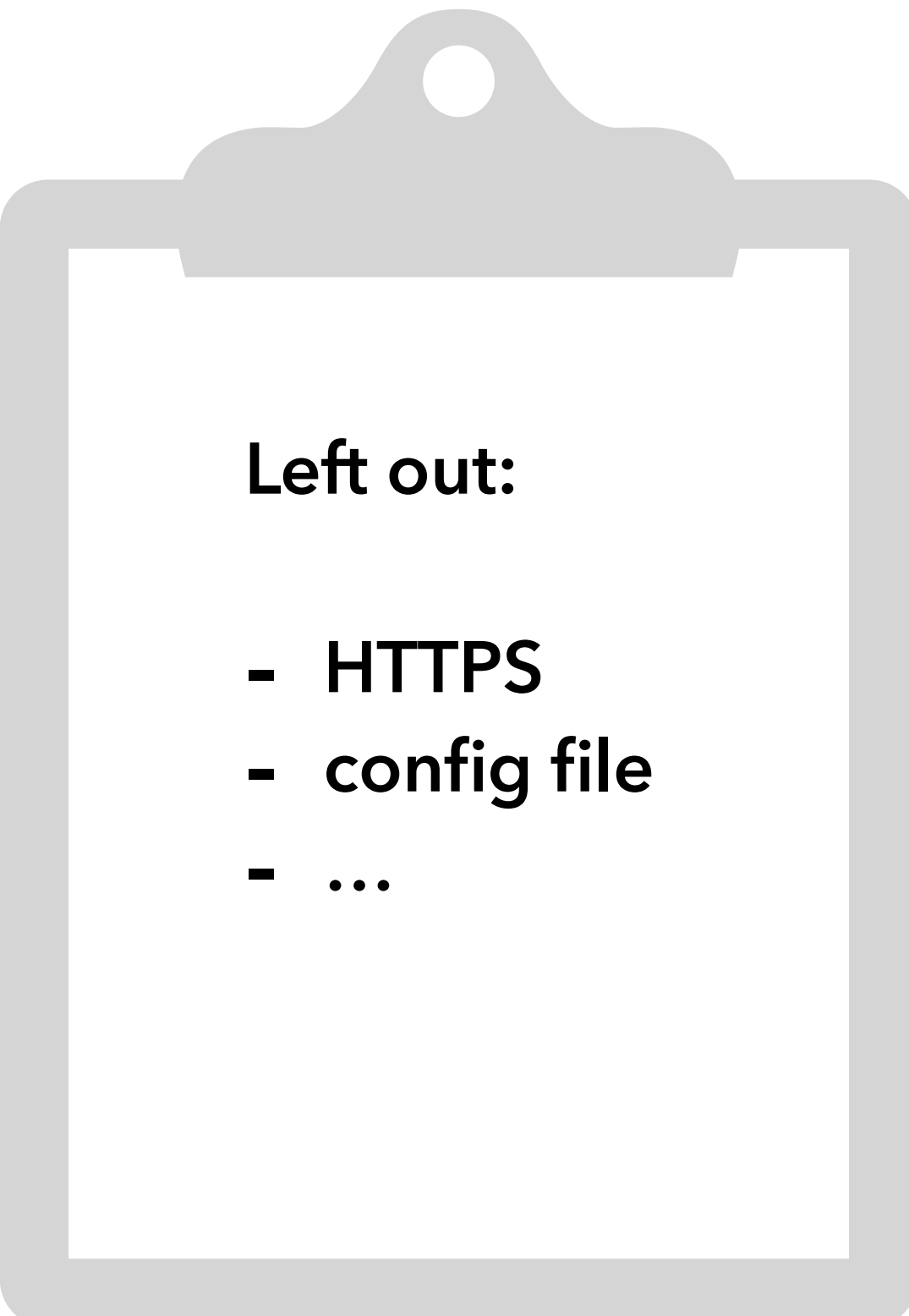


Technologies



Less than 600 LoC

including more than 200 test



Endpoints

```
demo-note-server — ./demo-note-server /Users/henri/git/demo-note-server — demo-note-server — 135x24
henri@Edinburgh ~/g/demo-note-server (master) [130]> ./demo-note-server
[GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.

[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env:   export GIN_MODE=release
- using code:  gin.SetMode(gin.ReleaseMode)

[GIN-debug] POST    /subscribe          --> main.(*Env).subscribeHandler-fm (4 handlers)
[GIN-debug] POST    /login              --> github.com/appleboy/gin-jwt.(*GinJWTMiddleware).LoginHandler-fm (4 handlers)
[GIN-debug] GET     /auth/refresh_token --> github.com/appleboy/gin-jwt.(*GinJWTMiddleware).RefreshHandler-fm (4 handlers)
[GIN-debug] POST    /auth/update/password --> main.(*Env).subscribeHandler-fm (5 handlers)
[GIN-debug] GET     /auth/notes         --> main.(*Env).noteListHandler-fm (5 handlers)
[GIN-debug] GET     /auth/notes/:id    --> main.(*Env).noteGetHandler-fm (5 handlers)
[GIN-debug] POST    /auth/notes         --> main.(*Env).notePostHandler-fm (5 handlers)
[GIN-debug] PATCH   /auth/notes/:id    --> main.(*Env).notePostHandler-fm (5 handlers)
[GIN-debug] DELETE  /auth/notes/:id    --> main.(*Env).noteDelete-fm (5 handlers)
[GIN-debug] POST    /auth/share/:id/:with --> main.(*Env).noteShareHandler-fm (5 handlers)
[GIN-debug] GET     /auth/share/notes  --> main.(*Env).getSharedNotes-fm (5 handlers)
[GIN-debug] GET     /ping               --> main.(*Env).httpEngine.func1 (4 handlers)
[GIN-debug] Environment variable PORT is undefined. Using port :8080 by default[GIN-debug] Listening and serving HTTP on :8080
```

Sharing, yeah!

JWT

jwt.go

```
func makeAuthenticator(...) {  
    ...  
    login = getParam(...)  
  
    var query Login  
    e.db.First(&query, "username = ?",  
    login.Username)  
  
    if login.password == query.Password {  
        return &Login{  
            Username: userID,  
        }, nil  
    }  
  
    ...  
}
```

JWT TOKEN



eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4
gRG9lIiwiaXNjb2N0IjoiPSYXNrXCjTwXyr4BsezDI1AVTmud2fU4

User management

Features

- ✓ Subscribe
- ✓ Login
- ✓ Refresh token
- ✓ Update password

If we want to use later the password for something else than authentication, we'll need a new one

	id	created_at	updated_at	deleted_at	username	password
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	2018-11-15 17...	2018-11-15 17...	NULL	user002	abcdefg

worst case!

Bonus features

✓ Soft delete 

	id	created_at	updated_at	deleted_at	title	content	owner
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	2018-11-15 17...	2018-11-15 17...	<i>NULL</i>	Implement End-...	If WhatsApp ca...	user002
2	2	2018-11-15 17...	2018-11-15 17...	2018-11-15 17...	First Major Rele...	Blah blah blah	user002

Bonus features

✓ Soft delete 

	id	created_at	updated_at	deleted_at	title	content	owner
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	2018-11-15 17...	2018-11-15 17...	NULL	Implement End...	If WhatsApp ca...	user002
2	2	2018-11-15 17...	2018-11-15 17...	2018-11-15 17...	First Major Rele...	Blah blah blah	user002



Getting the notes



WHERE owner={...}



```
notes.go

func (e *Env) noteListHandler(...) {
    ...
    owner := getOwner(c)

    err := e.db.Where("owner = ?", owner).Find(&notes).Error
    if err != nil {
        c.JSON(http.StatusUnauthorized, gin.H{"err": err})
        return
    }

    c.JSON(http.StatusOK, gin.H{"notes": notes})
}
```

Getting the notes



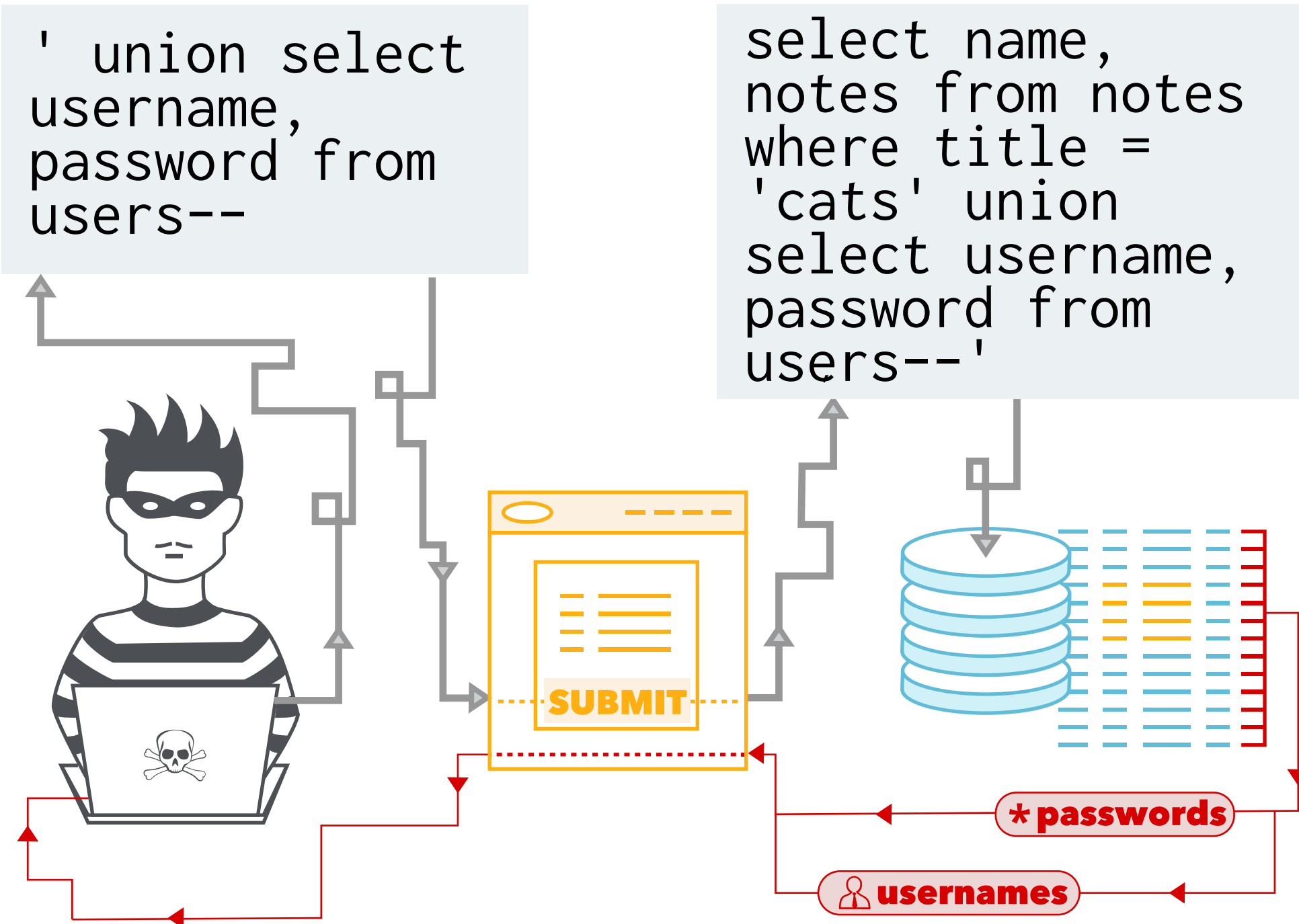
```
notes.go

func (e *Env) noteListHandler(...) {
    ...
    owner := getOwner(c)

    err := e.db.Where("owner = ?", owner).Find(&notes).Error
    if err != nil {
        c.JSON(http.StatusUnauthorized, gin.H{"err": err})
        return
    }

    c.JSON(http.StatusOK, gin.H{"notes": notes})
}
```

Problem: SQL Injections



Who hacked Facebook?

A researcher found that someone had been harvesting employee passwords since July, but its security team thinks it's no big deal.

Violet Blue, @violetblue
04.29.16 in Security

Late last week, a hacker named Orange Tsai wrote about how he hacked into Facebook under the aegis of its bug bounty program. A bug bounty is when a company pays hackers for vulnerabilities they find, providing the company with real-world threat testing outside the scope of its security team.

But Tsai found much more than a bug. He discovered that another hacker had been in the company's systems for around eight months, grabbing employee usernames and passwords -- and probably more.

In his post [How I Hacked Facebook, and Found Someone's Backdoor Script](#), Tsai

SQL Injection Prevention Cheat Sheet

OWASP Cheat Sheets

Last revision (mm/dd/yy): 02/6/2018

Introduction

[hide]

- 1 Introduction
- 2 Primary Defenses
 - 2.1 Defense Option 1: Prepared Statements (with Parameterized Queries)
 - 2.2 Defense Option 2: Stored Procedures
 - 2.3 Defense Option 3: White List Input Validation
 - 2.4 Defense Option 4: Escaping All User-Supplied Input
 - 2.4.1 Database Specific Escaping Details
 - 2.4.1.1 Oracle Escaping
 - 2.4.1.1.1 Escaping Dynamic Queries
 - 2.4.1.1.2 Turn off character replacement
 - 2.4.1.1.3 Escaping Wildcard characters in Like Clauses
 - 2.4.1.1.4 Oracle 10g escaping
 - 2.4.1.2 MySQL Escaping
 - 2.4.1.3 SQL Server Escaping
 - 2.4.1.4 DB2 Escaping
 - 2.4.2 Hex-encoding all input
 - 2.4.3 Escaping SQLi in PHP
- 3 Additional Defenses
 - 3.1 Least Privilege

Problem: Cleartext Data (and passwords)



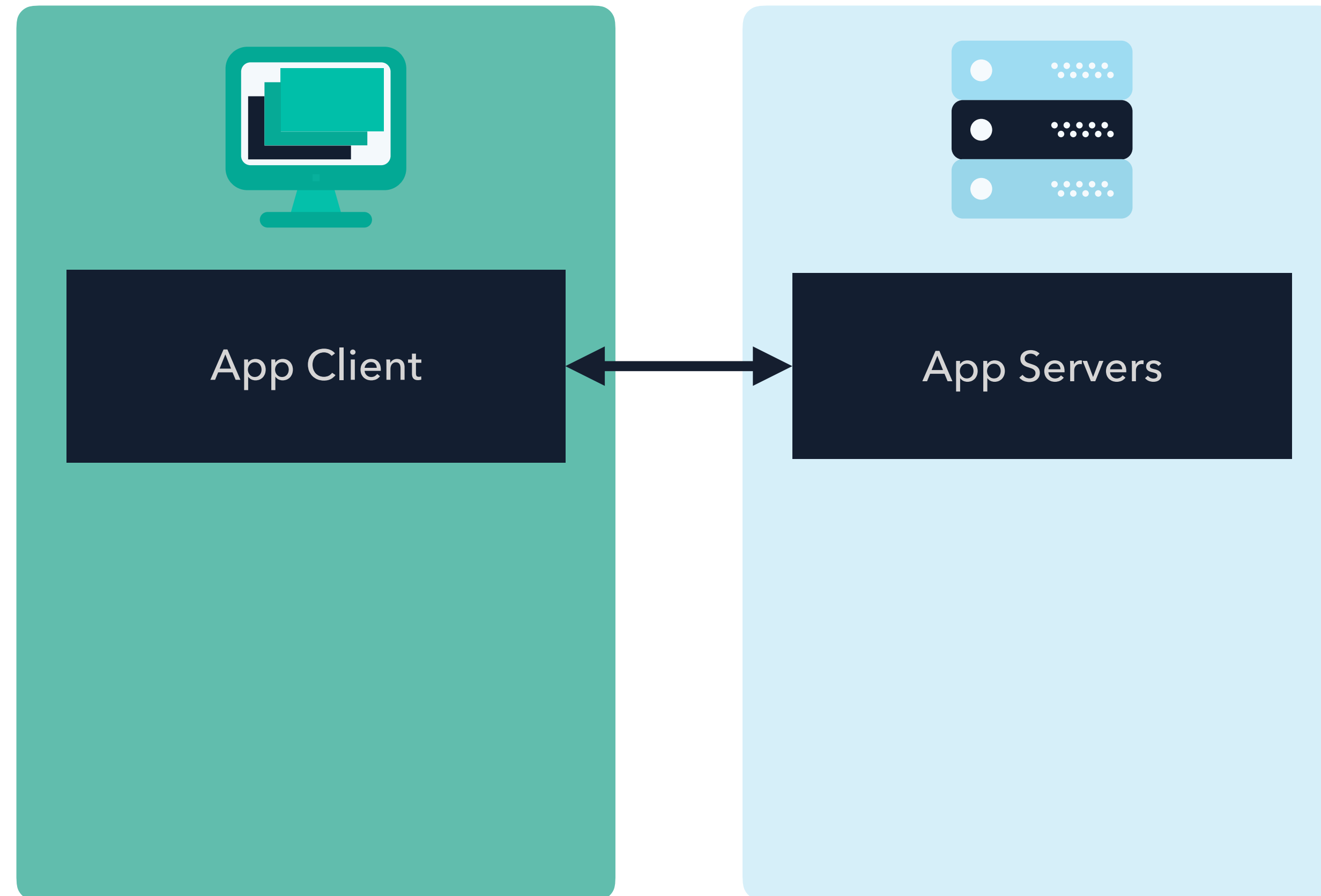
and let's not forget





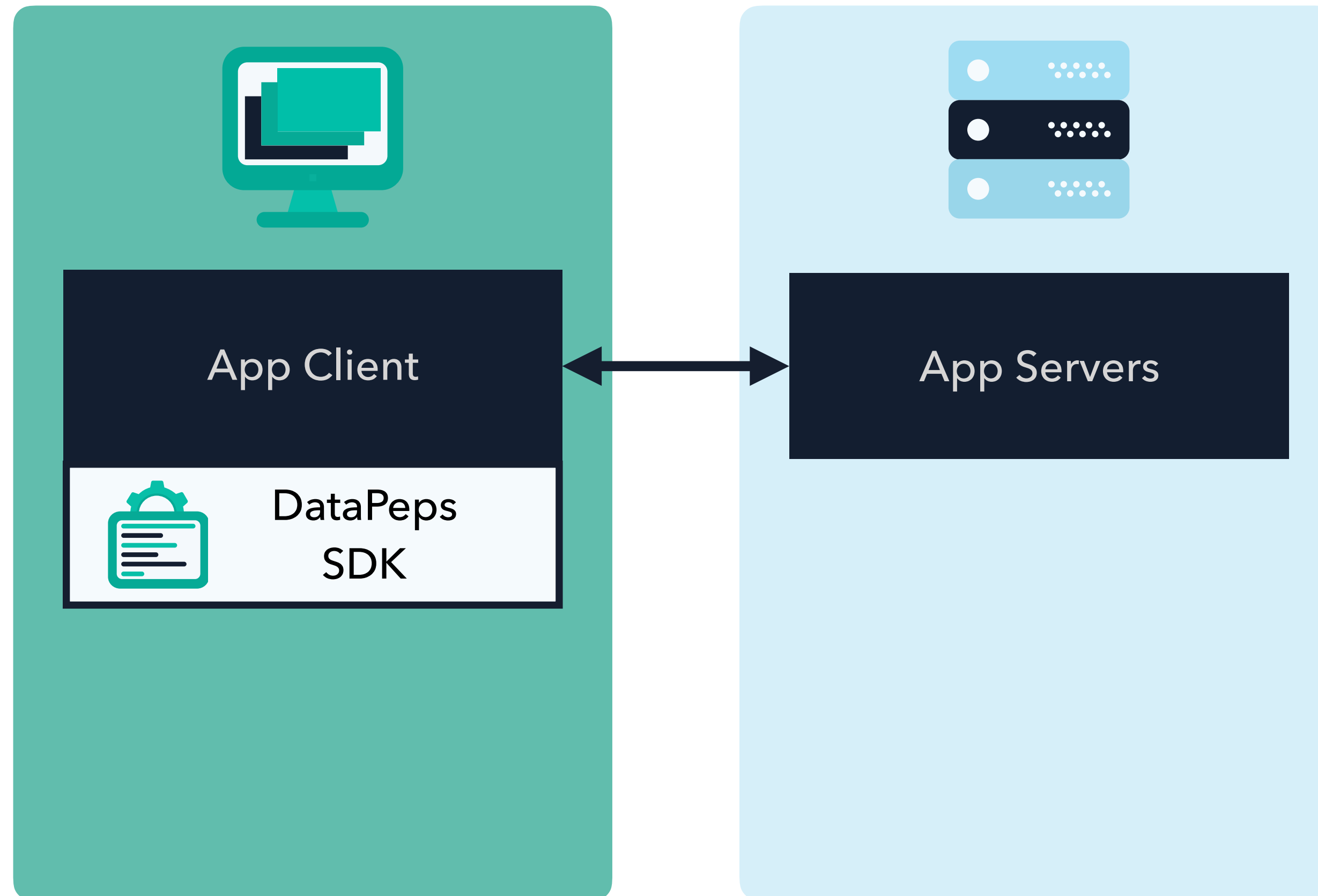
The Solution

DataPeps



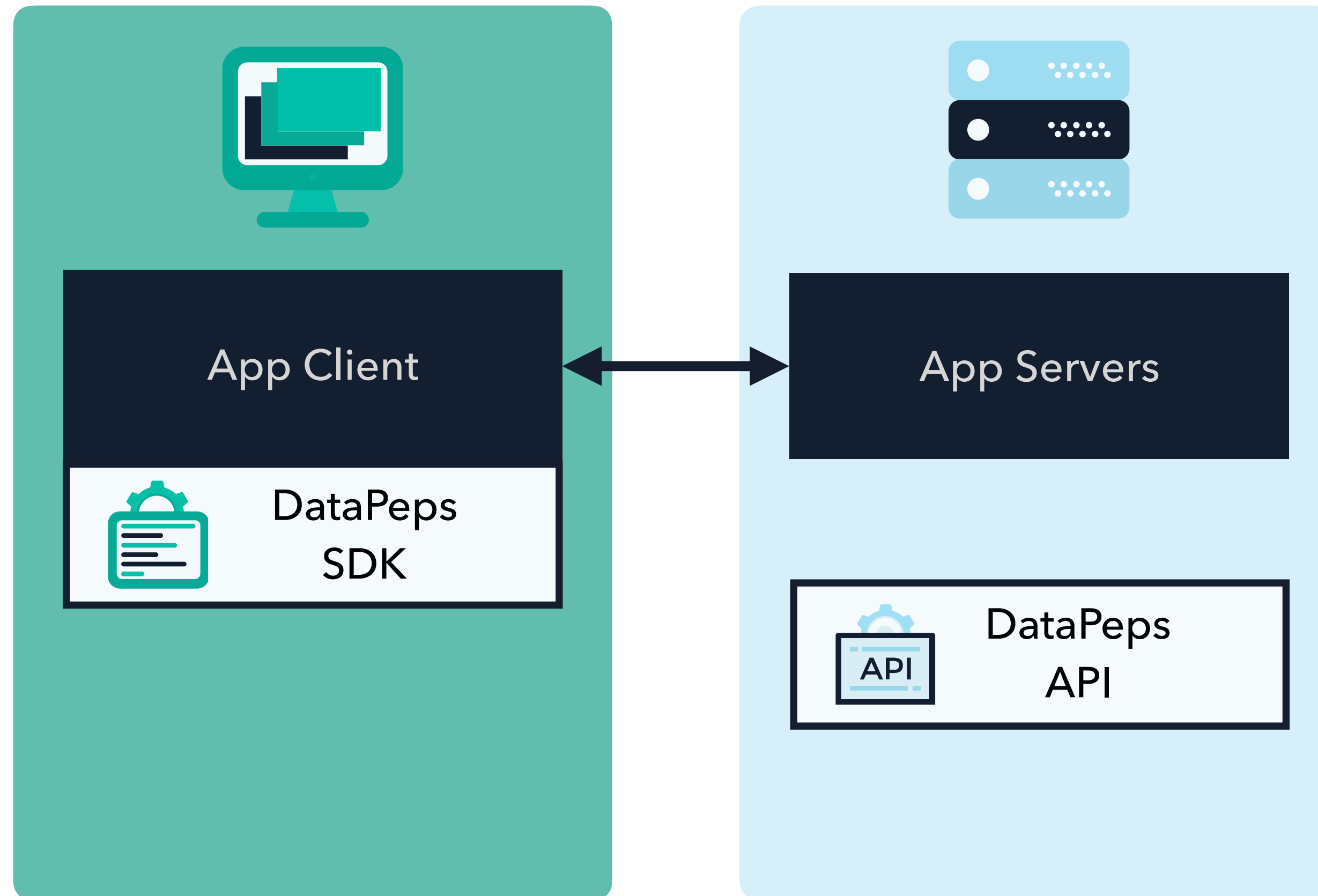
DataPeps

Dev(Ops) integrate
DataPeps SDK into a Client
App (web, desktop, ...)



DataPeps

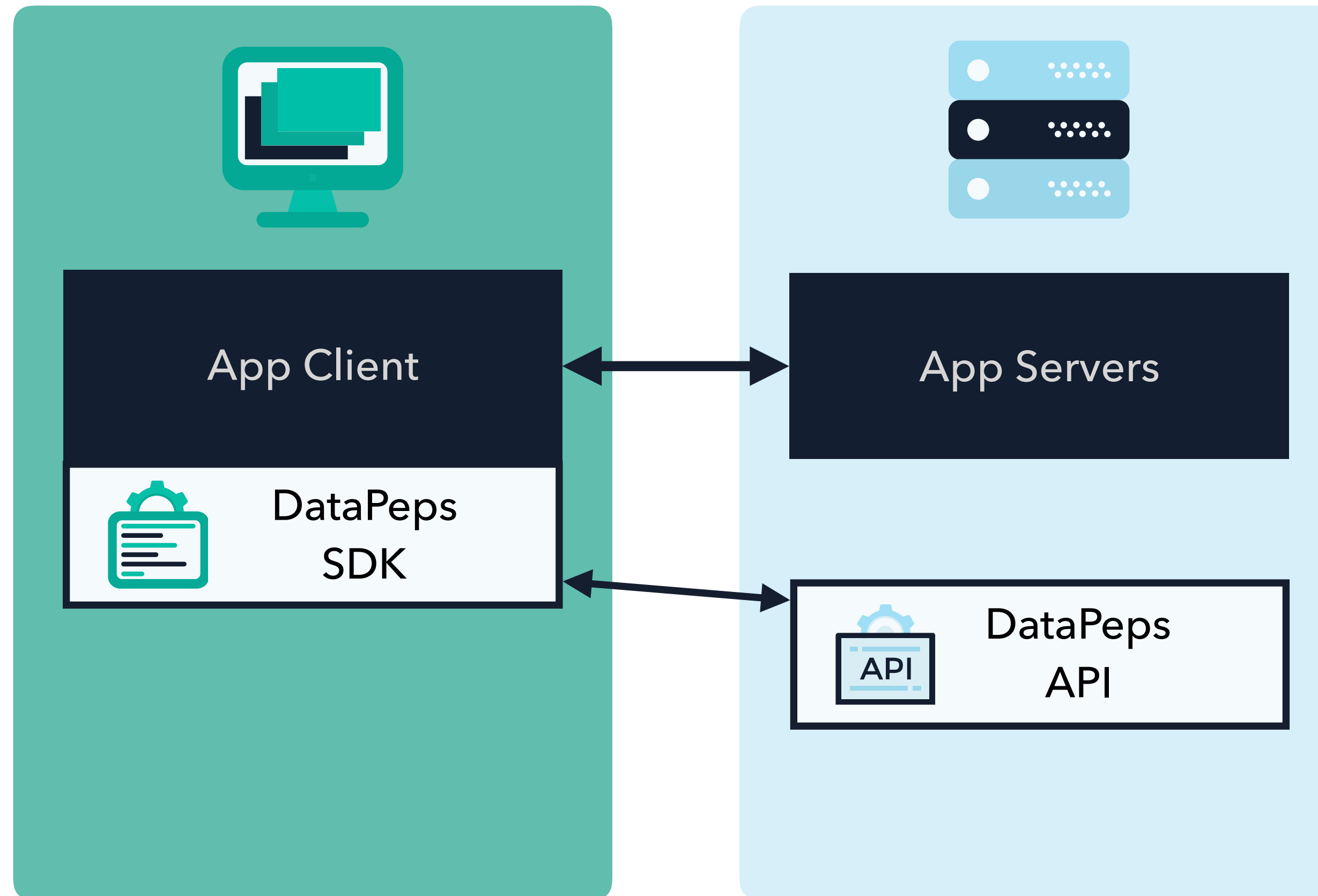
Dev(Ops) integrate
DataPeps SDK into a Client
App (web, desktop, ...)



The DataPeps API service
can run in the cloud or on-
premises

DataPeps

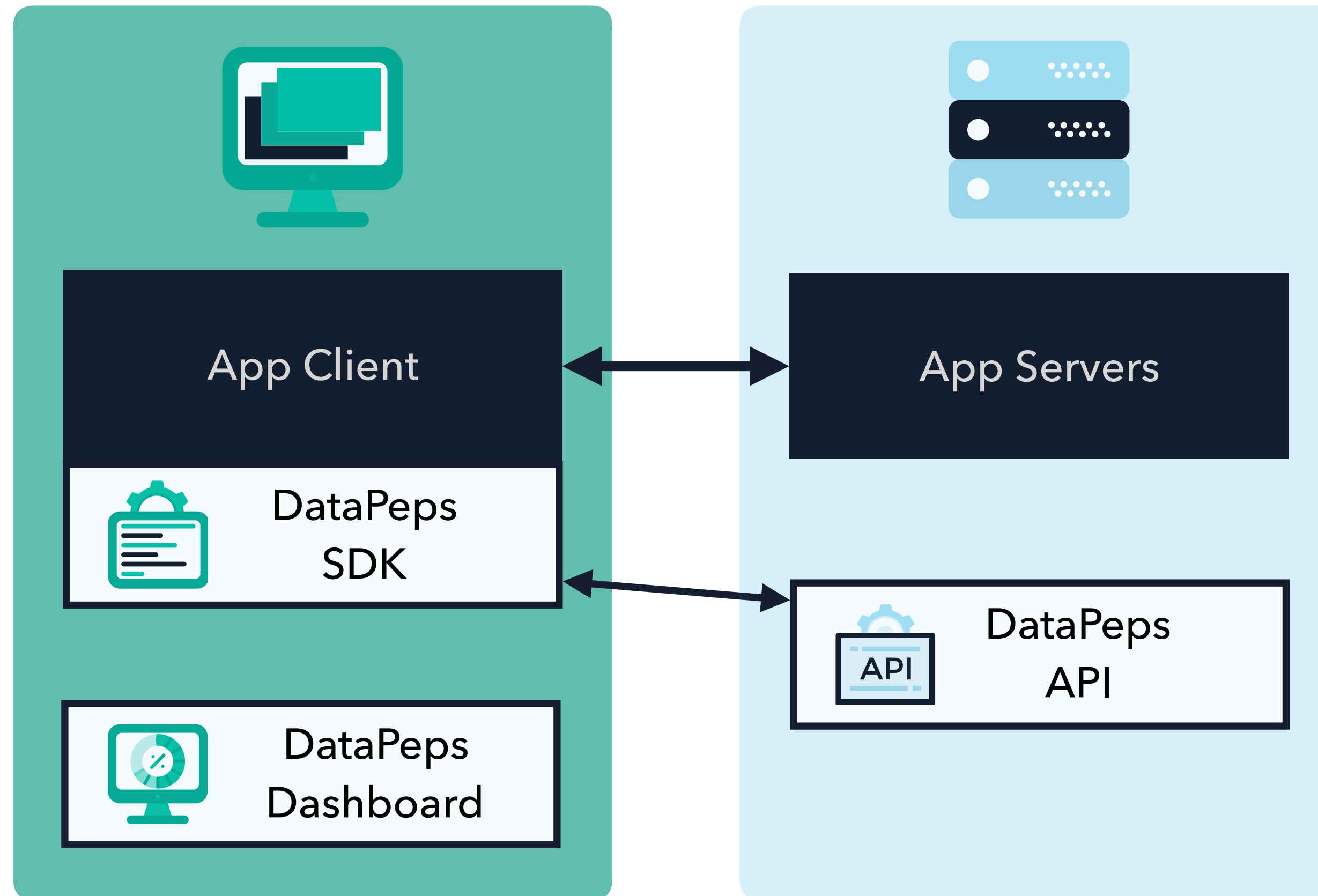
Dev(Ops) integrate
DataPeps SDK into a Client
App (web, desktop, ...)



The DataPeps API service
can run in the cloud or on-
premises

DataPeps

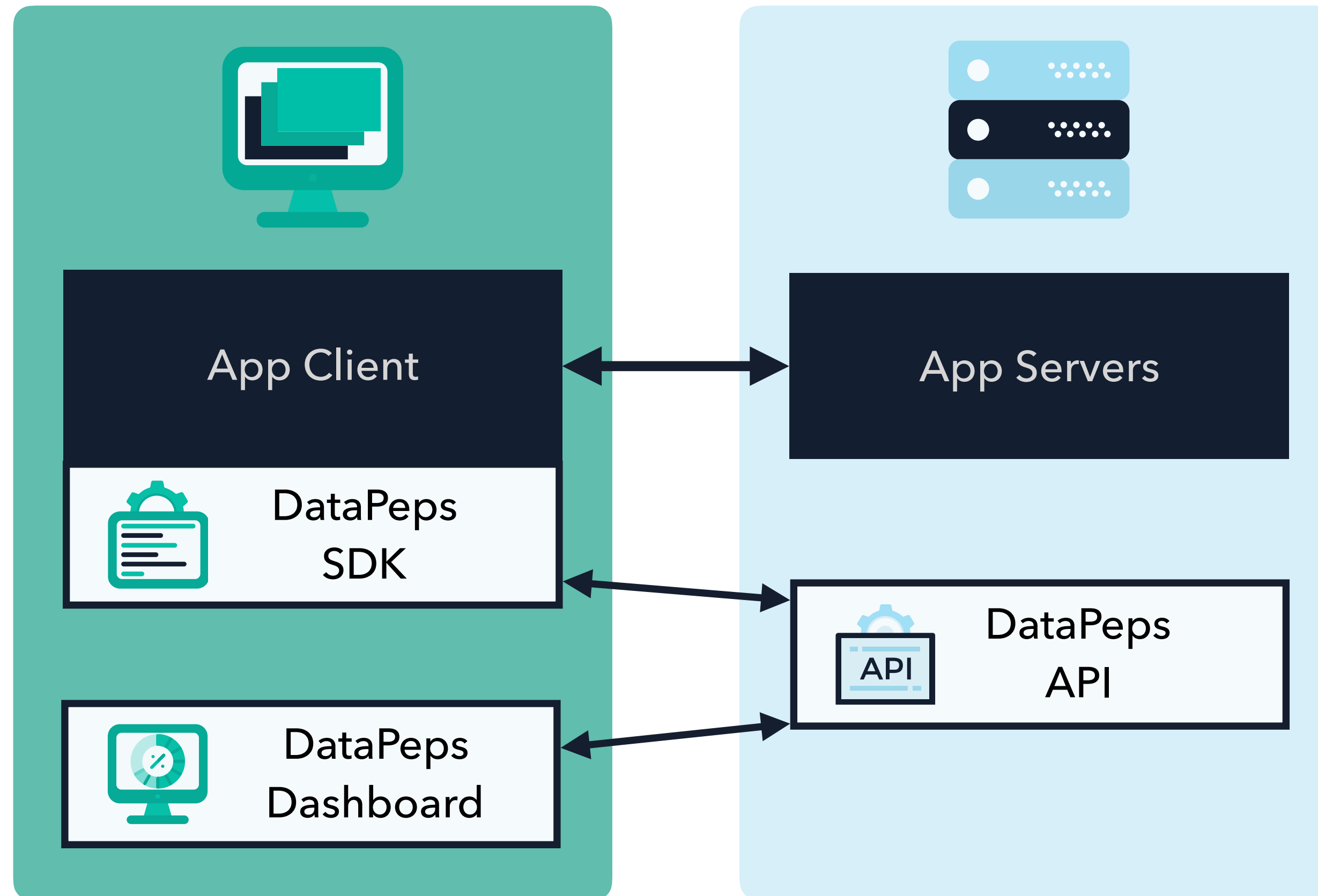
Dev(Ops) integrate
DataPeps SDK into a Client
App (web, desktop, ...)



The DataPeps API service
can run in the cloud or on-
premises

DataPeps

Dev(Ops) integrate
DataPeps SDK into a Client
App (web, desktop, ...)

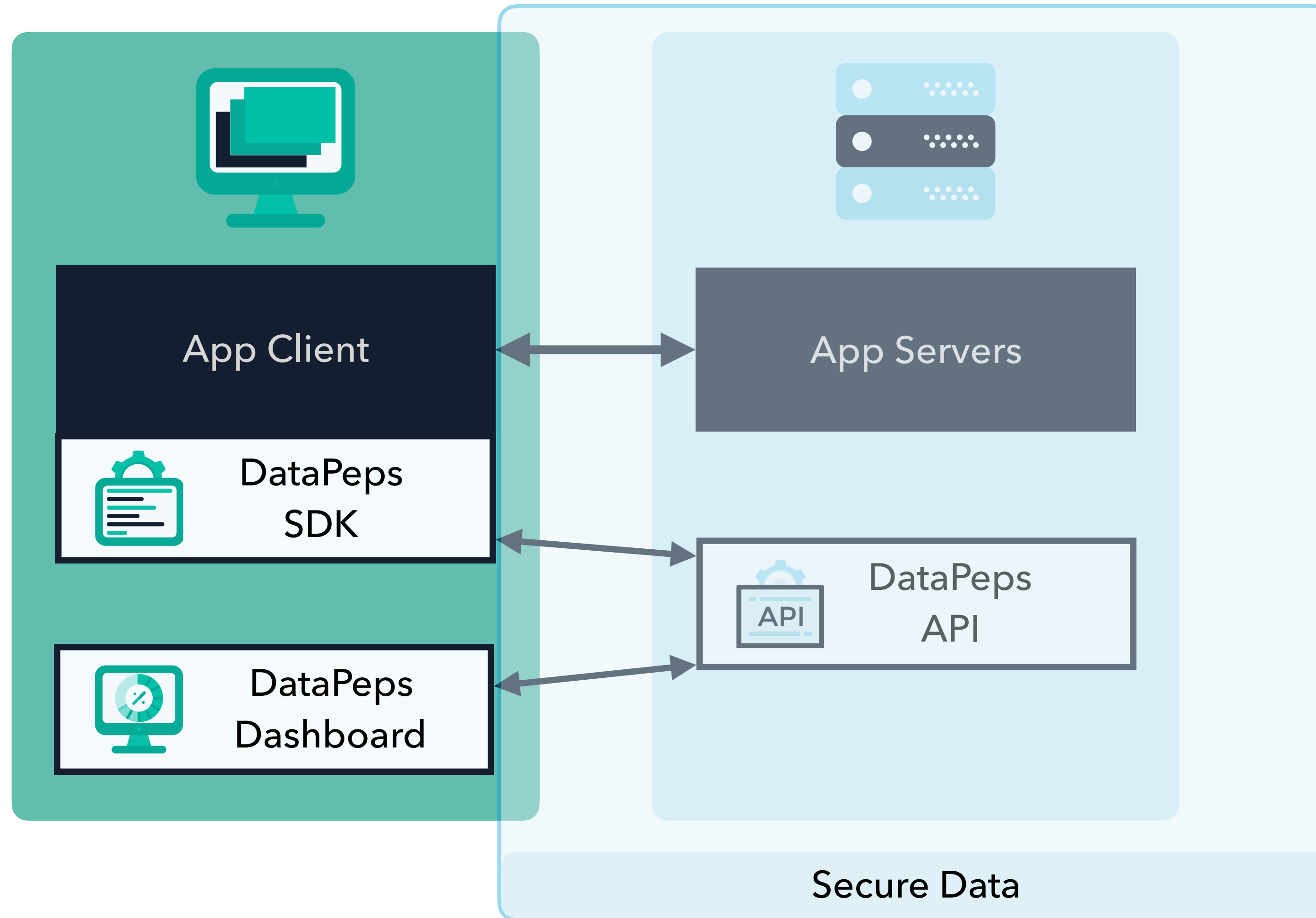


The DataPeps API service
can run in the cloud or on-
premises

DataPeps

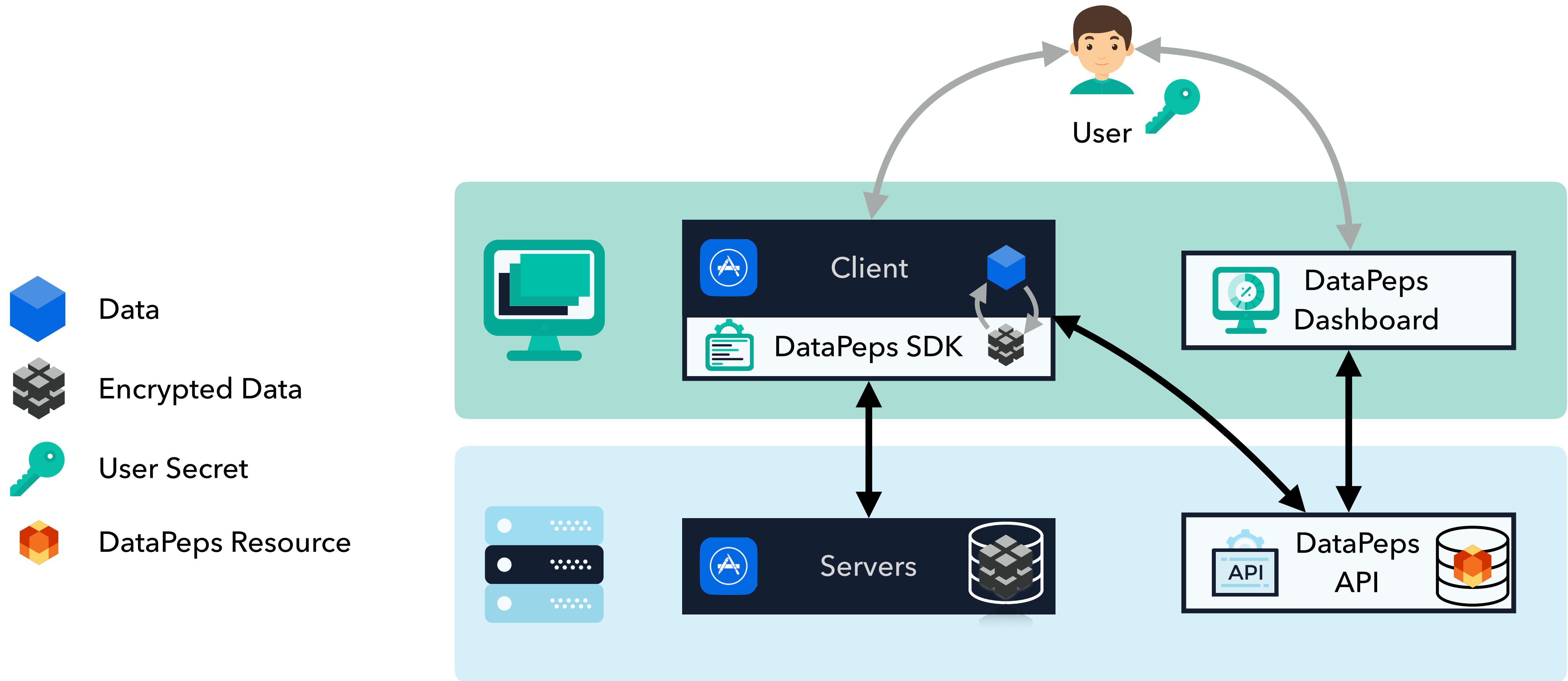
Everything beyond the client is protected by the use of cryptography

Dev(Ops) integrate DataPeps SDK into a Client App (web, desktop, ...)

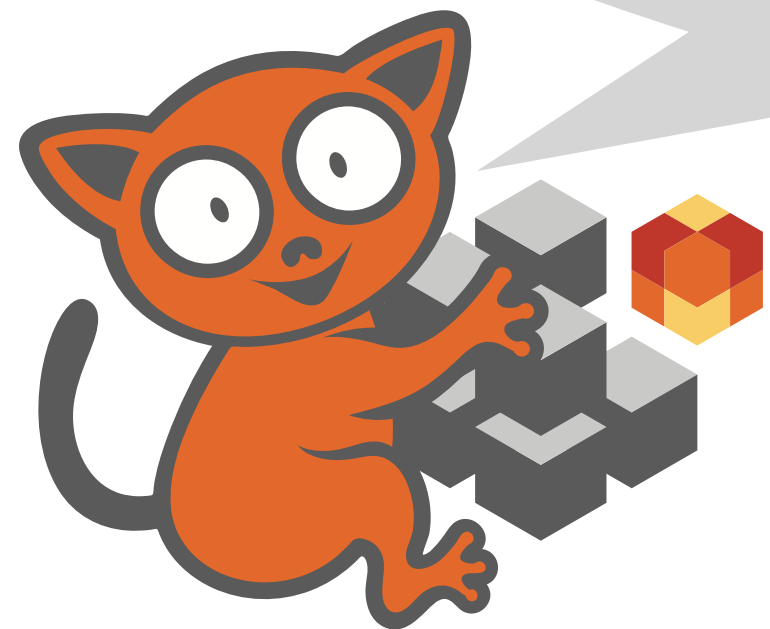


The DataPeps API service can run in the cloud or on-premises

Architecture

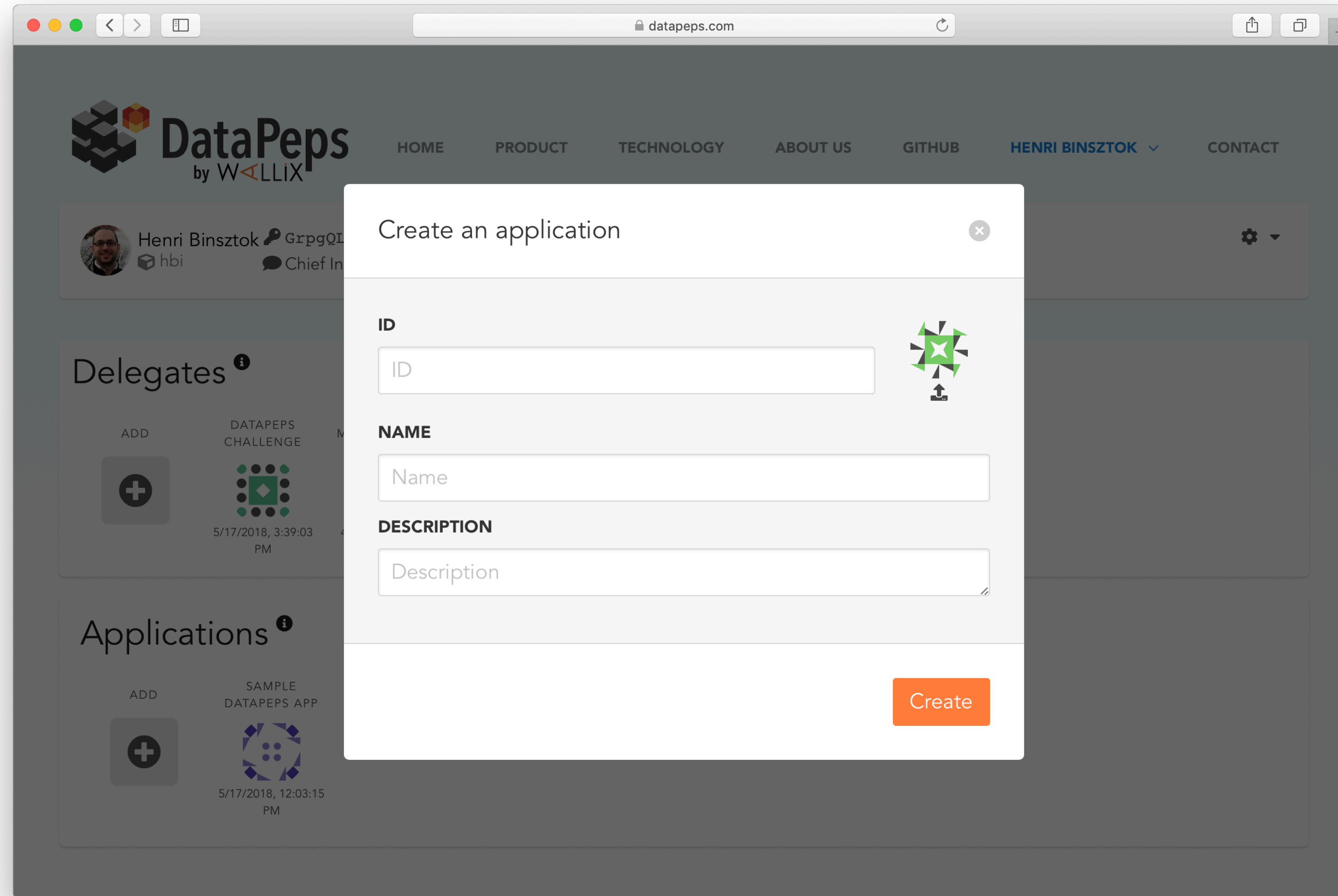


The Developer Mission



1

Create your App at DataPeps.com



2

Configure the JWT Public Key

The screenshot shows a web browser window at `datapeps.com` with a modal dialog titled "Configure application demo-note". The dialog contains the following configuration options:

- SIGN METHOD:** A dropdown menu set to "RS256".
- JWT KEY:** A text area containing a public key in PEM format:

```
-----BEGIN PUBLIC KEY-----  
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsbNOF38cqpFnH  
c3KTSye  
w5bYZI2bnqFiUbcdR/IlhbPI/Ne/HJjlGYESKHPXwQI2PV0ZsqyYyNM2pQvHtm  
Tt  
Z5g-BPL-QiAG...CY...T1E...HPOQ-080H...V2DDU...G...G...E...
```
- CLAIM TO USE FOR LOGIN:** A text input field containing the value "id".

A "Save" button is located at the bottom right of the dialog. The background interface shows the user profile for "Henri Binsztok" and a list of "Owners" including "FRANÇOIS RIEUCAU".

Update auth.js

actions/auth.js (partial)

```
function login(username, password) {  
  return async dispatch => {  
    try {  
      const user = await  
      authService.login(username, password);  
      dispatch(success(user));  
      history.push("/");  
    } catch (error) {  
      dispatch(failure(error));  
    }  
  };  
}
```

actions/auth.js (partial)

```
const connector = {  
  createSession: async (login, password) =>  
    await authService.login(login, password),  
  getToken: async user => user.token  
};  
const {  
  session: datapeps,  
  app: user,  
  new: firstTime  
} = await  
DataPeps.ApplicationAPI.createJWTSession(  
  config.dataPepsAppID,  
  username, password, connector  
);  
dispatch(success(user, datapeps));  
history.push("/");
```

Update newnote.js

components/newnote.js (partial)

```
async onAddNote() {  
  let title = this.state.title;  
  let content = this.state.content;  
  this.props.addNote(title, content);  
}
```

components/newnote.js (partial)

```
async onAddNote() {  
  ...  
  const { datapeps } = this.props;  
  const resource = await  
    datapeps.Resource.create(  
    "note",  
    {  
      description: title,  
      URI: `${config.apiUrl}/auth/notes`,  
      MIMEType: "text/plain"  
    },  
    [datapeps.login]  
  );  
  title = resource.encrypt(title);  
  title = clipID(resource.id, title);  
  content = resource.encrypt(content);  
  ...  
}
```

5

Update note.js

components/note.js (partial) **unchanged**

```
class Note extends React.Component {  
  
  render() {  
    const { Title, Content } = this.state;  
    return (  
      <Panel>  
        <Button  
          onClick={() =>  
            &times;  
          </Button>  
        <Panel.Heading>  
          <Panel.Title>  
            {Title}  
          </Panel.Title>  
        </Panel.Heading>  
        <Panel.Body>  
          {Content}  
        </Panel.Body>  
      </Panel>  
    );  
  }  
}
```

components/note.js (partial)

added

```
async decryptNote() {  
  try {  
    const { datapeps } = this.props;  
    const { id, data: encryptedTitle } = unclipID(this.state.Title);  
    const resource = await datapeps.Resource.get(id);  
    const Title = resource.decrypt(encryptedTitle);  
    const Content = resource.decrypt(this.state.Content);  
    this.setState({ ...this.state, Title, Content, style: "warning" });  
  } catch (_) {}  
}
```

Bonus: Add new features

New Note ×

 Protected

Notes

Notes is simple note-taking app
accompanying REST service is

Create an account

Login

Update Your Password ×

Notes now uses [end-to-end encryption](#) with [DataPeps](#) to protect your data!
Please update your password to automatically use encryption for your notes.

Change Password

Passwords must match and contain 7 or more characters

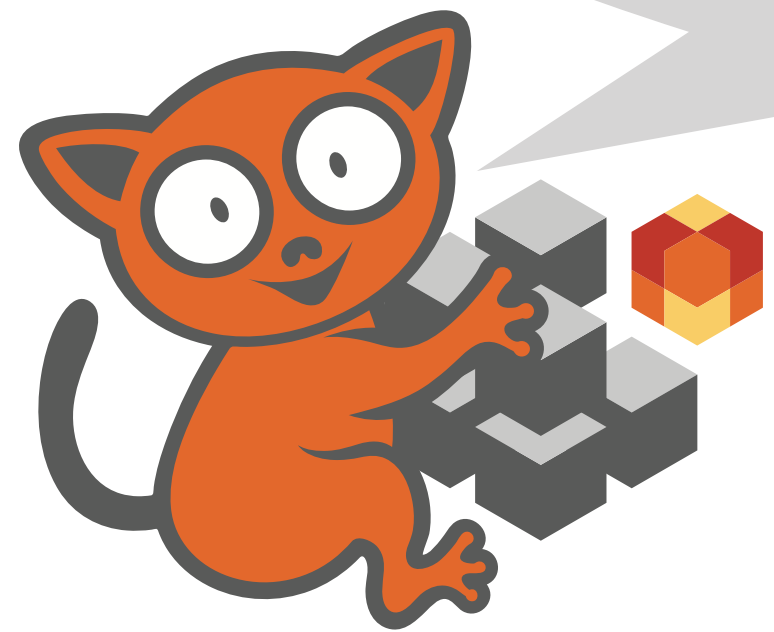
Cancel Update

Nothing else: No Server Modifications

And there's nothing left. **Zip. Zilch. Zero.**

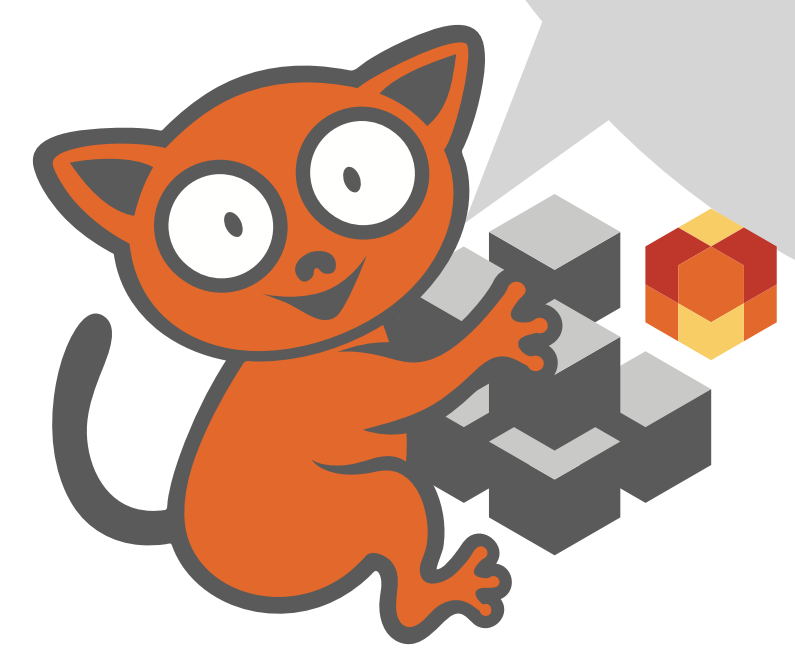
So the server can be a cloud service for which you have no source...

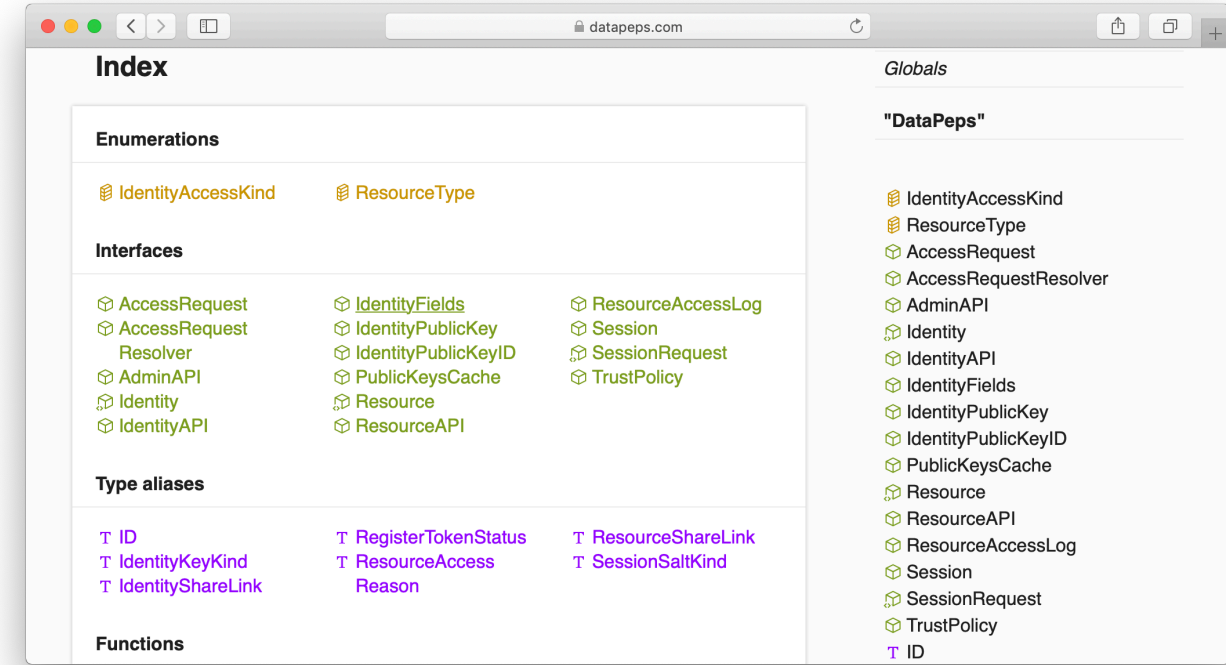
The Demo





datapeps.com



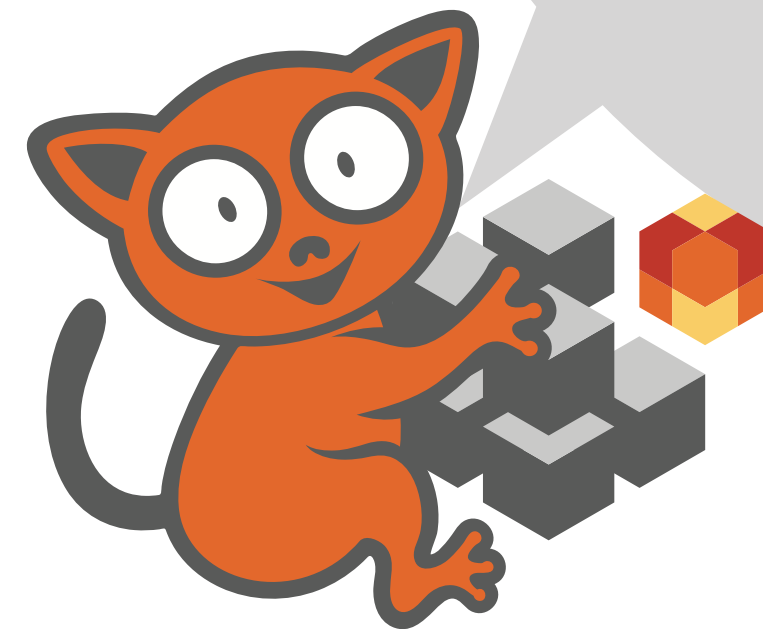


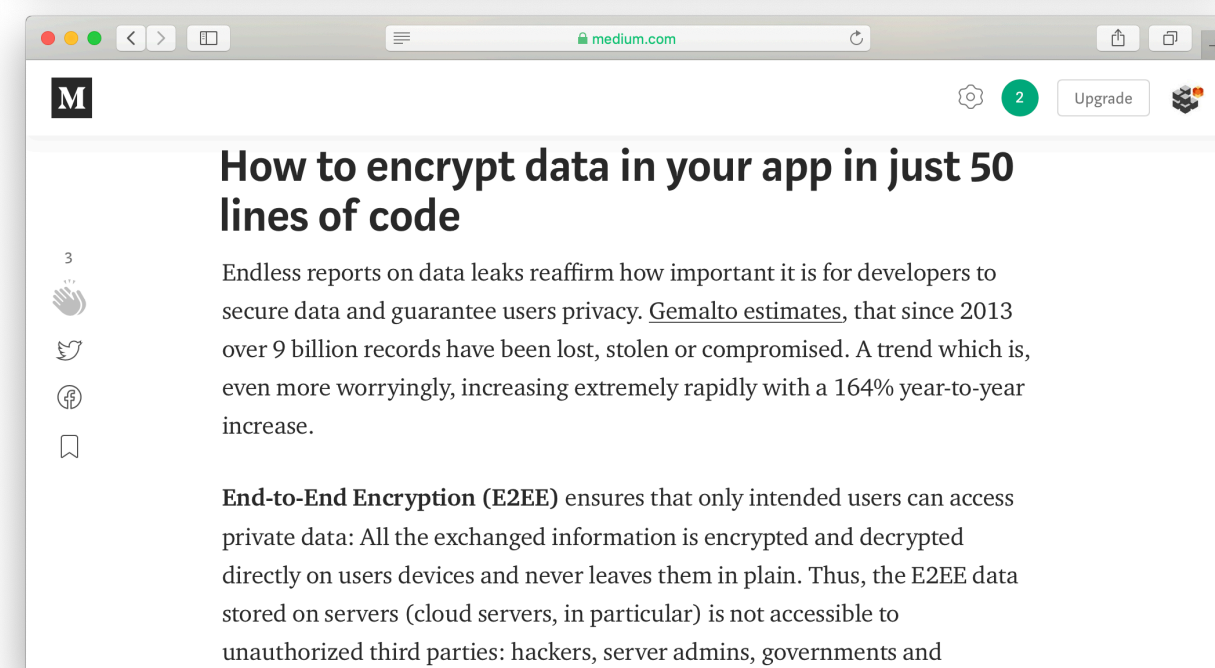
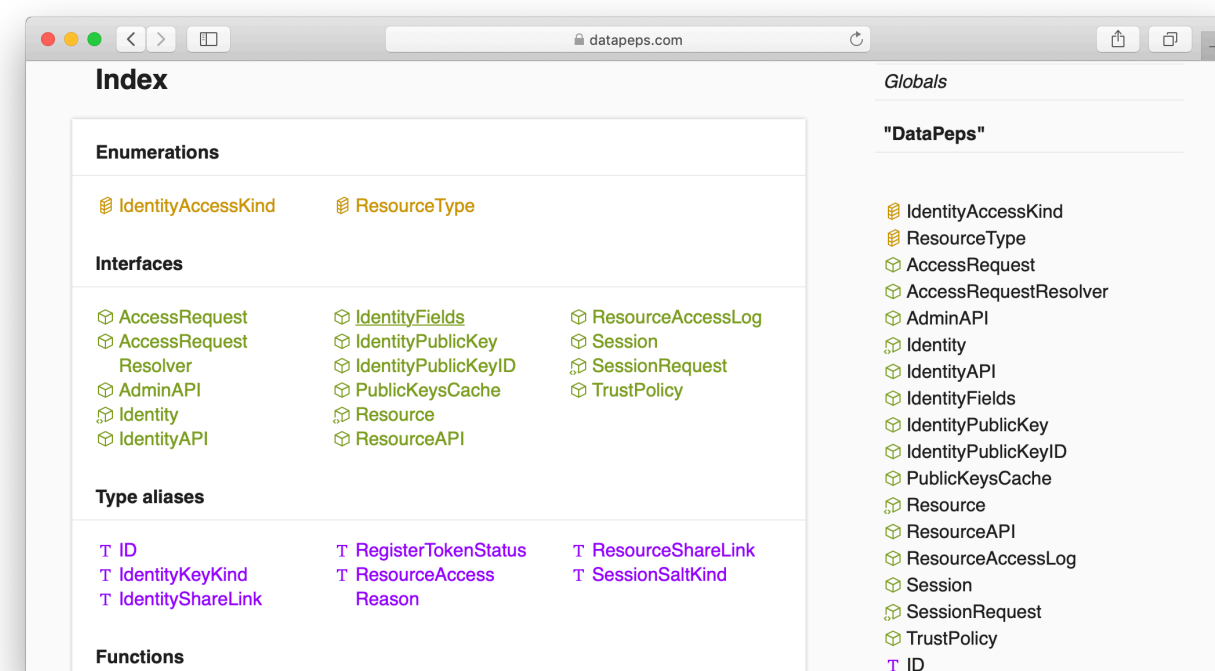
SDK Documentation

<https://datapeps.com/docs/sdk/js>



datapeps.com





SDK Documentation

<https://datapeps.com/docs/sdk/js>

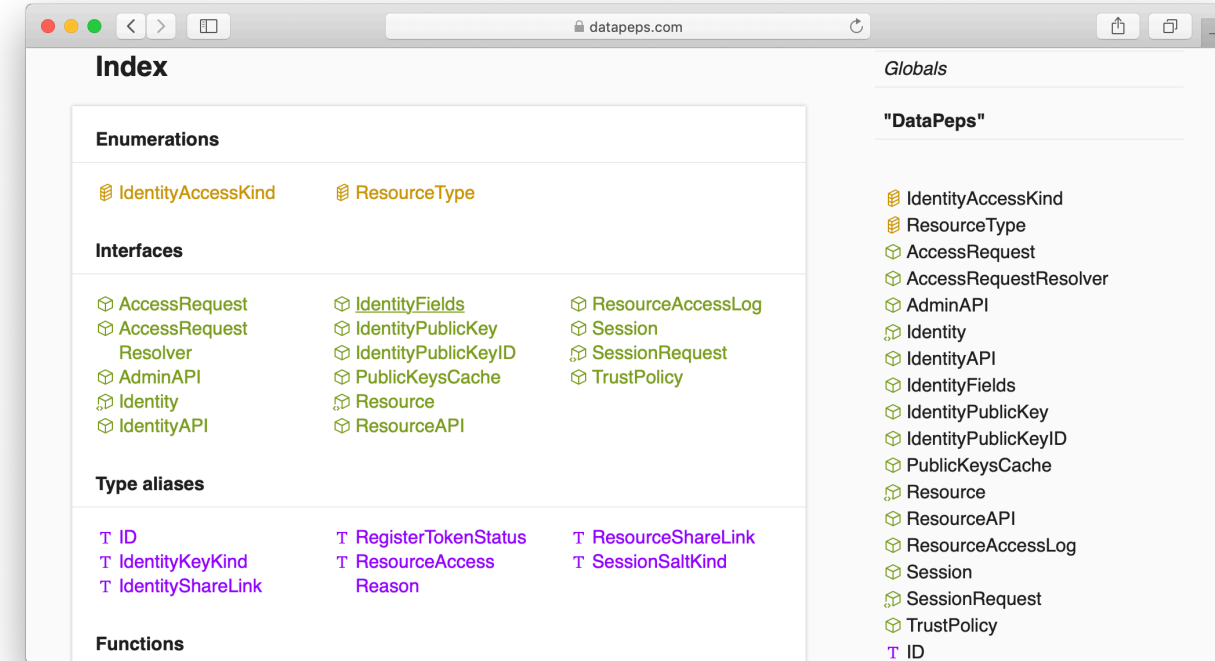
Tutorial

<https://medium.com/@datapeps>



datapeps.com





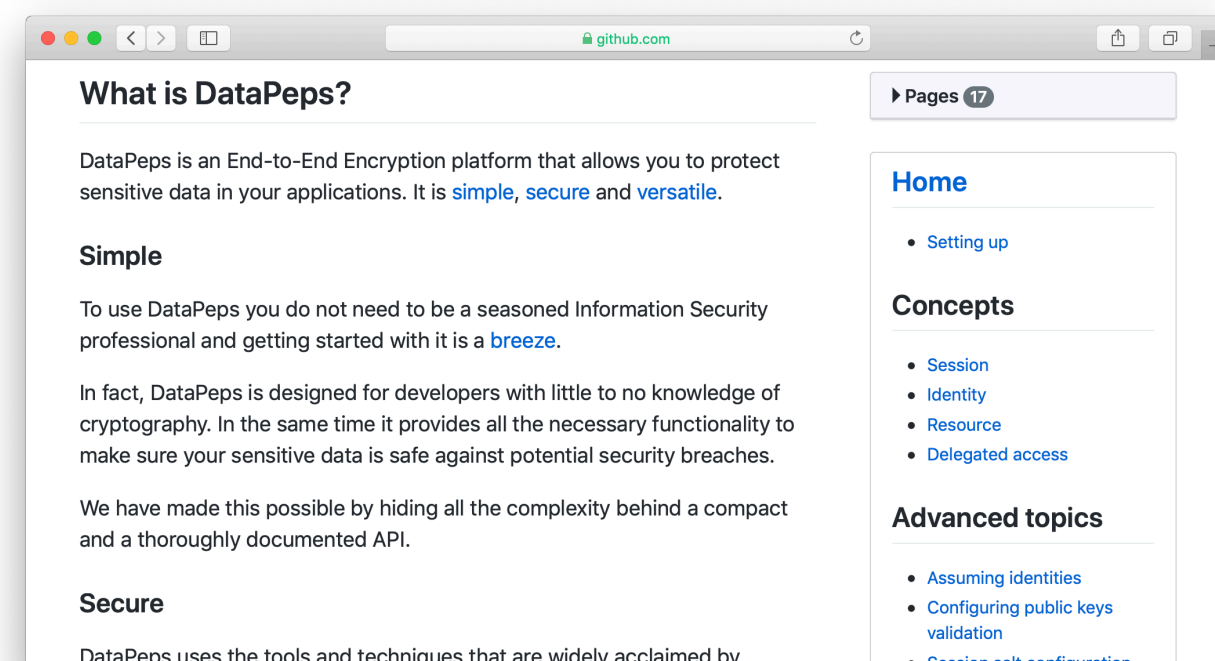
SDK Documentation

<https://datapeps.com/docs/sdk/js>



Tutorial

<https://medium.com/@datapeps>



Developer Manual

<https://github.com/wallix/datapeps-sdk-js/wiki>



datapeps.com