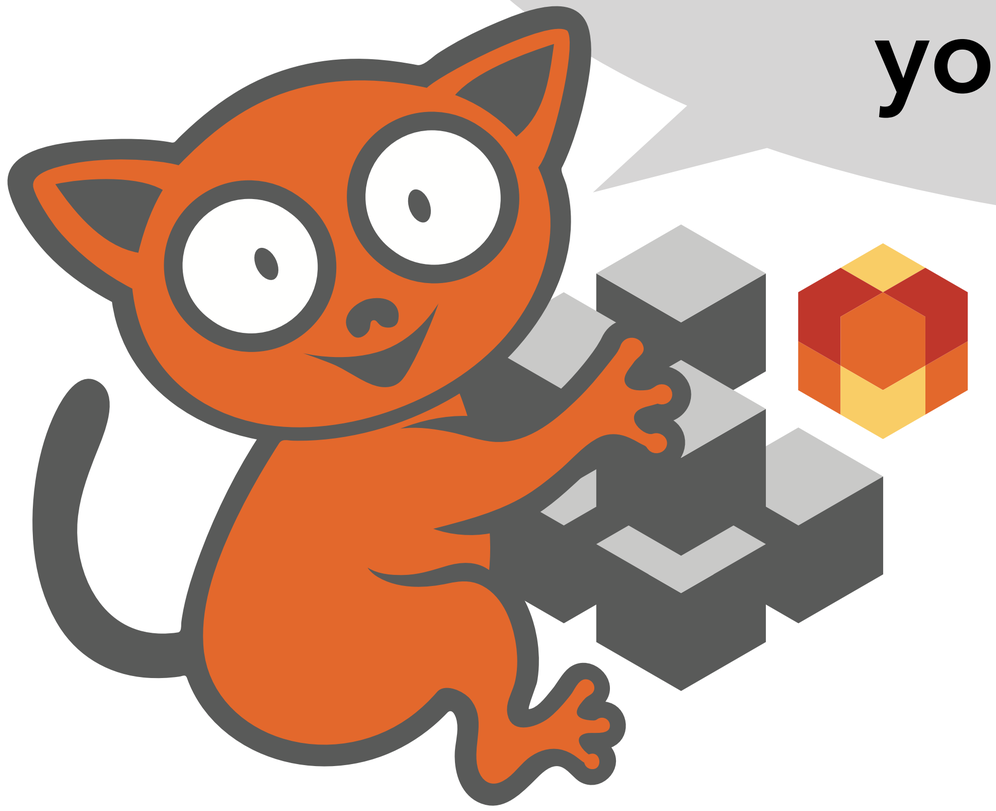
WALLIX

DATAPEPS

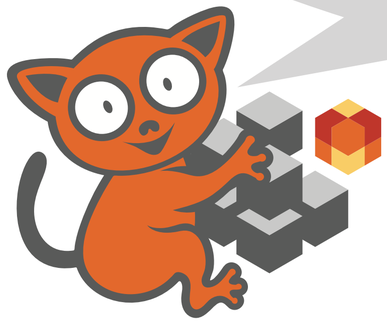
How to implement E2EE in your app in 50 minutes



Henri Binsztok

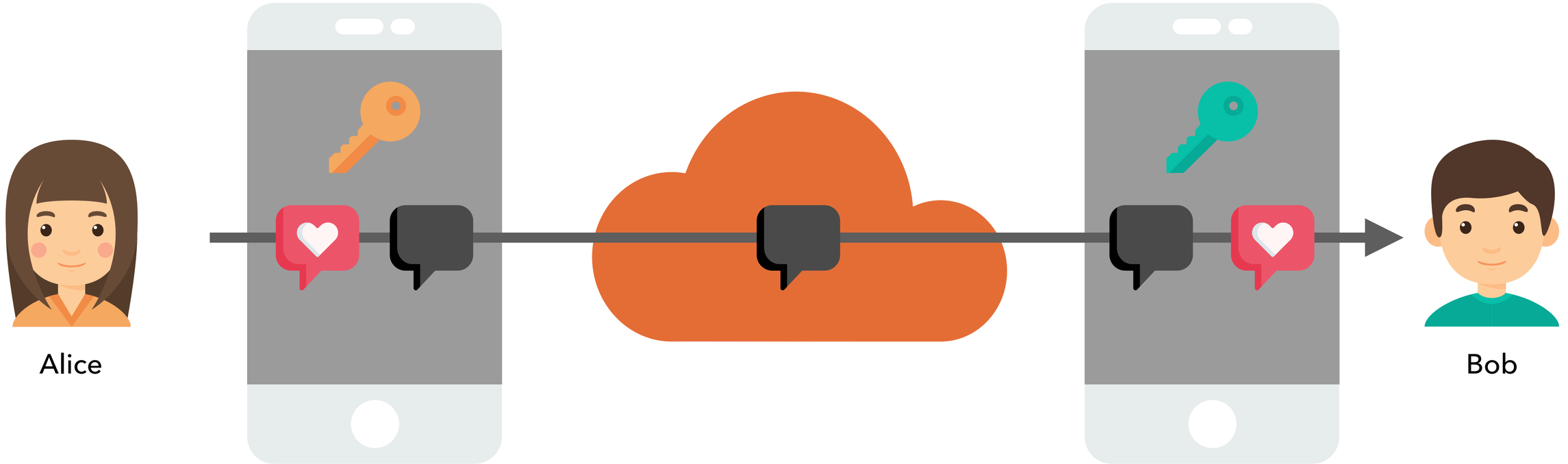
WALLIX

The Vision

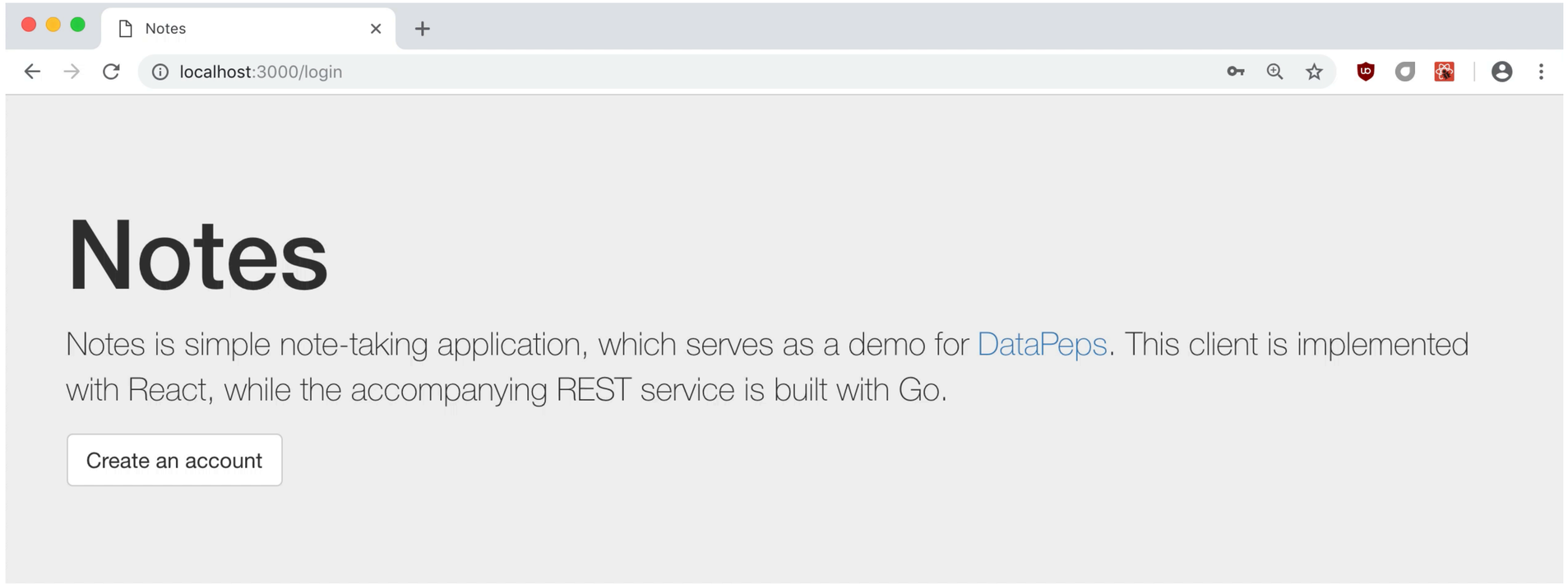


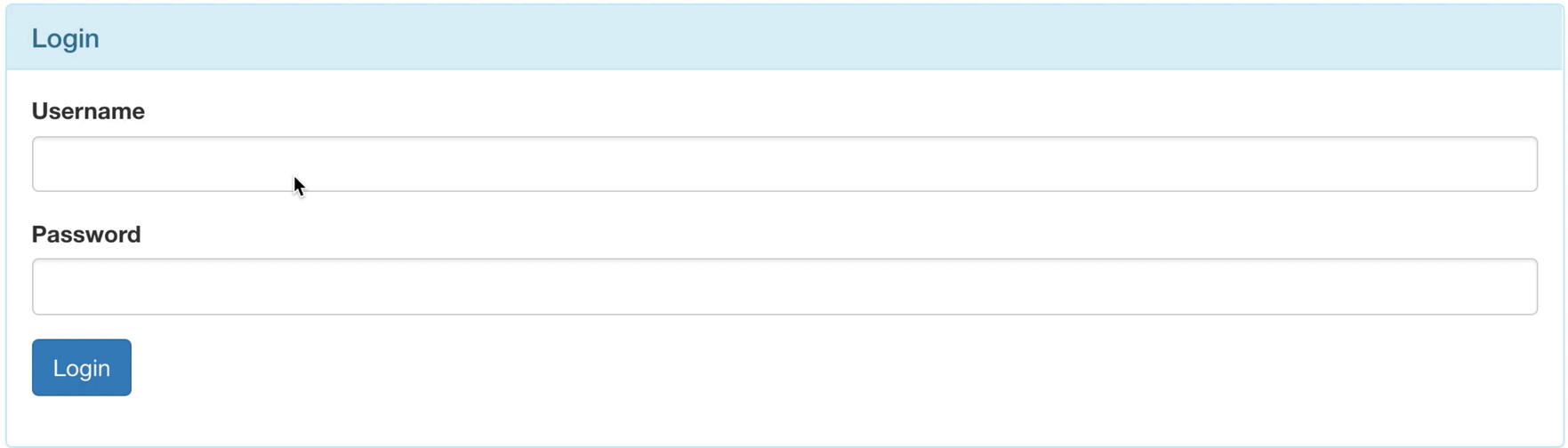
WALLIX
DATAPEPS

End-to-End Encryption



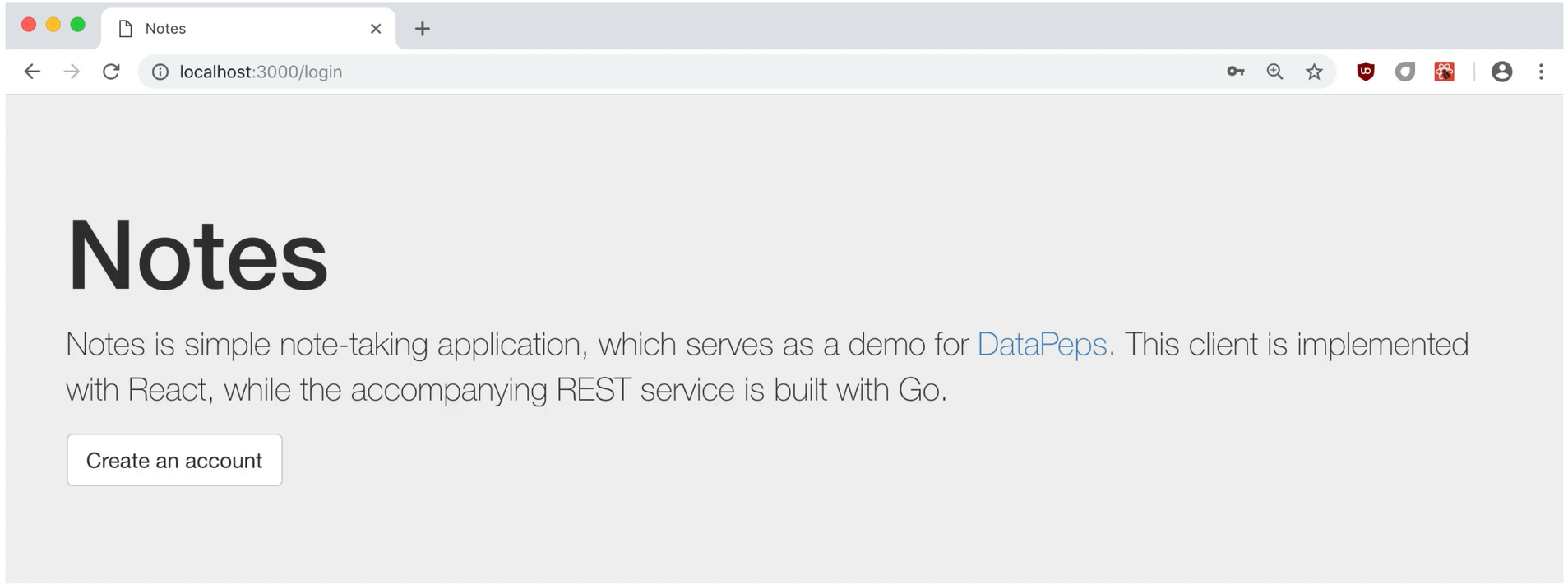
Our Demo App

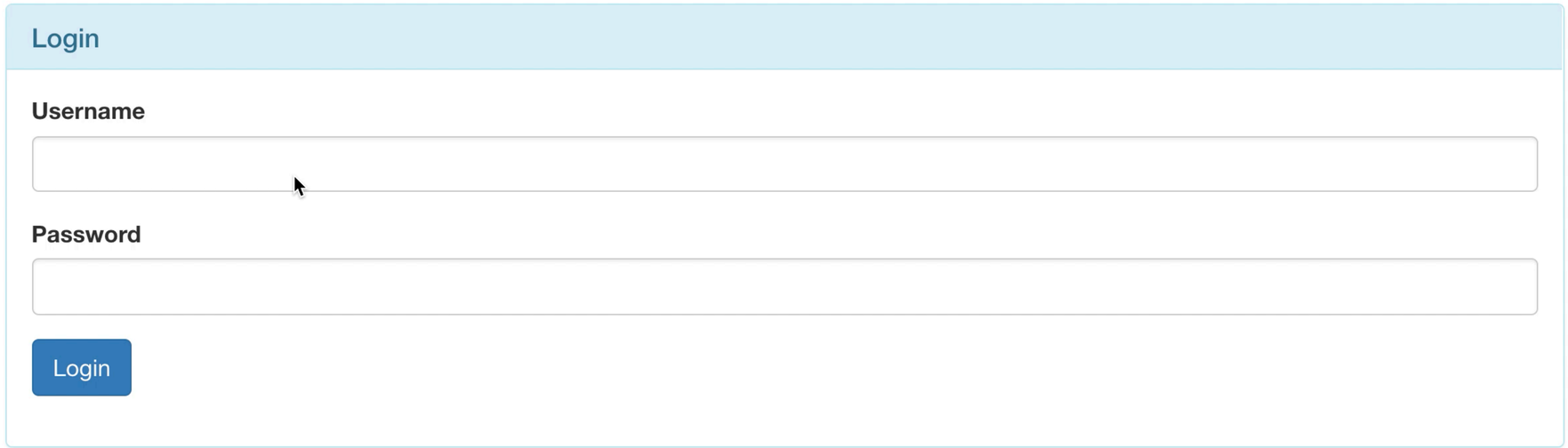




WALLIX DATAPEPS

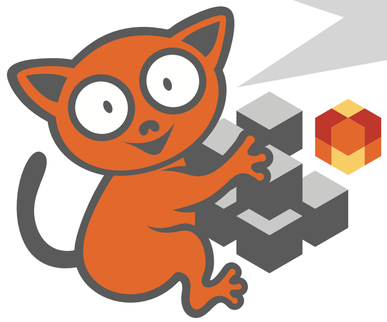
Our Demo App





WALLIX DATAPEPS

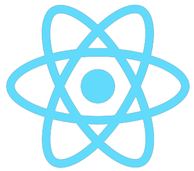
The Tour
Notes Frontend



WALLIX
DATAPEPS

Front-end Technologies

JavaScript



React

Redux

Bootstrap

Small project: 1200 LoC

package.json

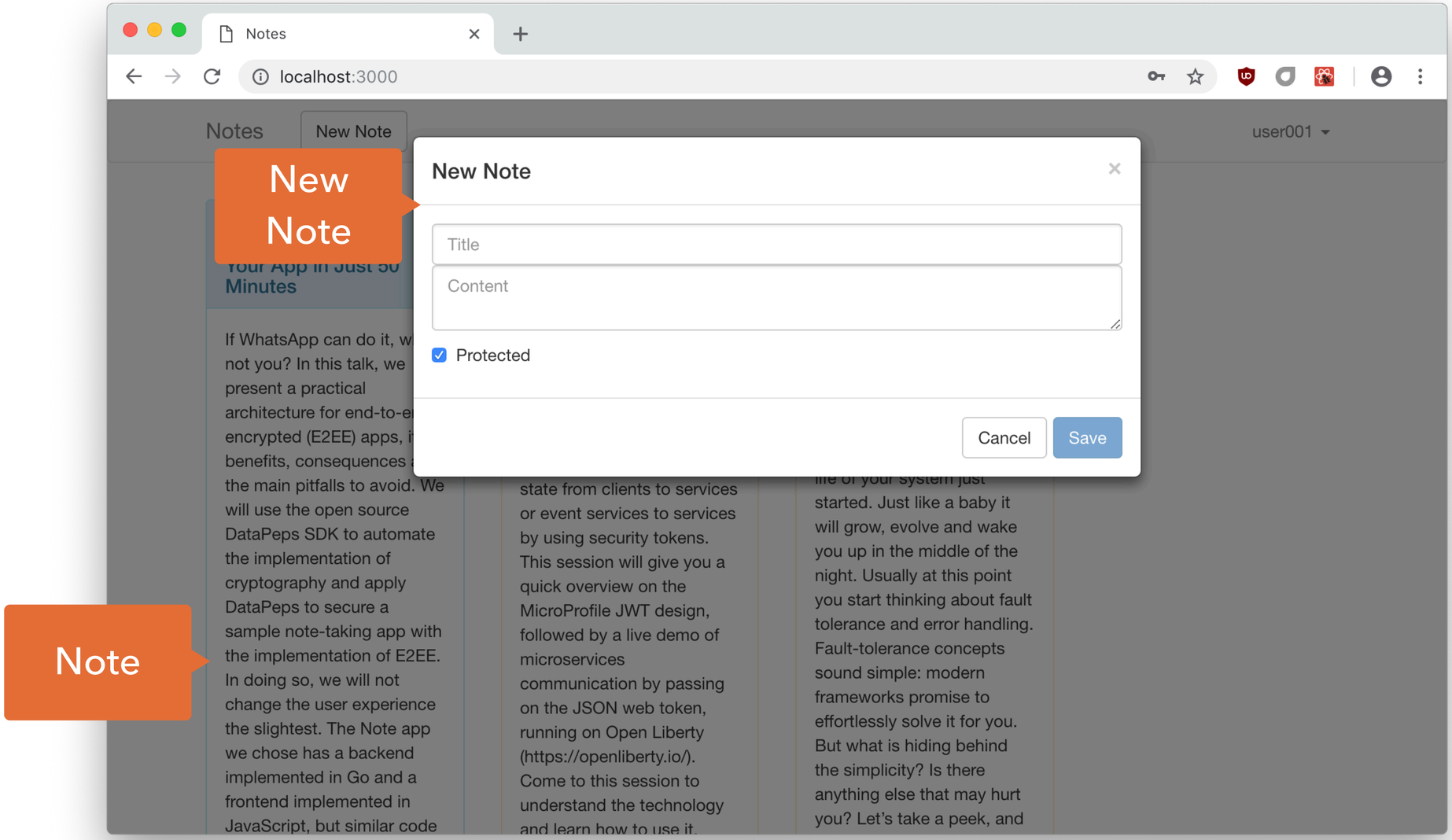
```
"devDependencies": {
    "react-scripts": "^2.0.5"
  },
  "dependencies": {
    "history": "^4.7.2",
    "long": "^4.0.0",
    "react": "^16.3.1",
    "react-bootstrap": "^0.32.4",
    "react-dom": "^16.3.1",
    "react-redux": "^5.0.7",
    "react-router-bootstrap": "^0.24.4",
    "react-router-dom":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

Authentication

actions/auth.js (partial)

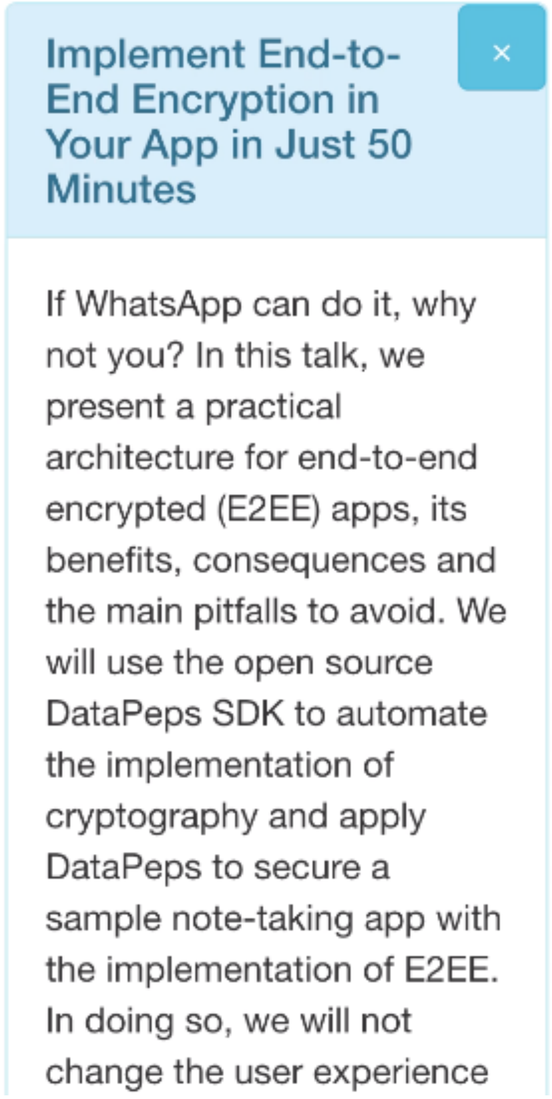
```
function login(username, password) {
  return async dispatch ⇒ {
    try {
      const user = await
authService.login(username, password);
      dispatch(success(user));
      history.push("/");
    } catch (error) {
      dispatch(failure(error));
    }
  };
}
```

Components Overview



WALLIX DATAPEPS

Note Component: Display a Note

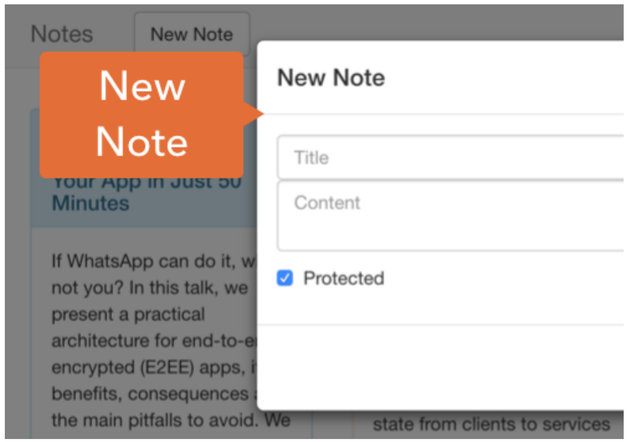


components/note.js (partial)

```
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>
    );
  } }
```

WALLIX DATAPEPS

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>
      …
}
```

WALLIX DATAPEPS

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);
}
```

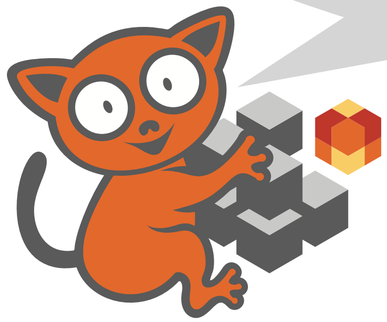
event

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



WALLIX
DATAPEPS

Technologies

Go

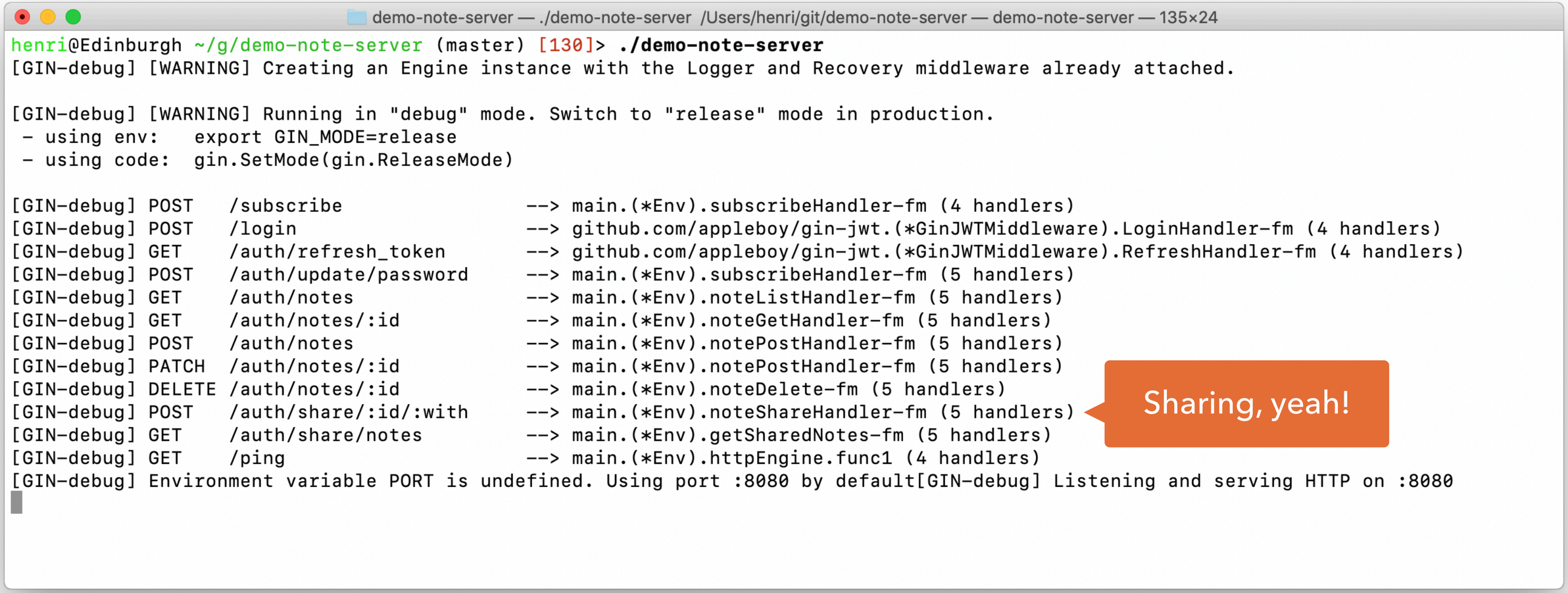
SQLite

JWT

Less than 600 LoC

including more than 200 test

Endpoints



WALLIX DATAPEPS

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
     }

     …
}
```

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4
gRG9lIiwiaXNTb2NpYWwiOnRydWV9.
4pcPyMD09olPSyXnrXCjTwXyr4BsezdI1AVTmud2fU4

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!

WALLIX DATAPEPS

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

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*	mplement 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

WALLIX DATAPEPS

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

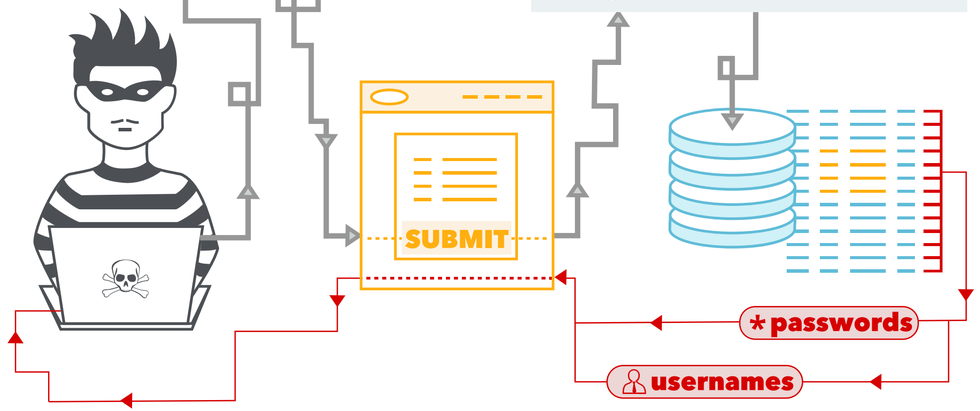
notes.go

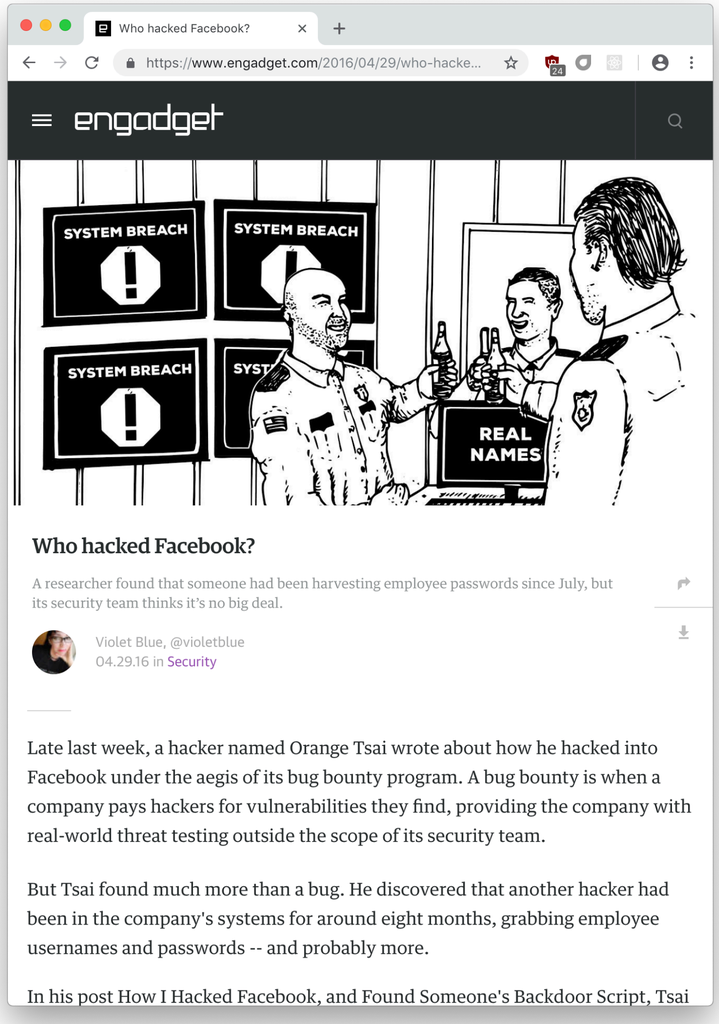
```
func (e *Env) noteListHandler(…) {
    …
    owner := getOwner(c)

    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})
}
```

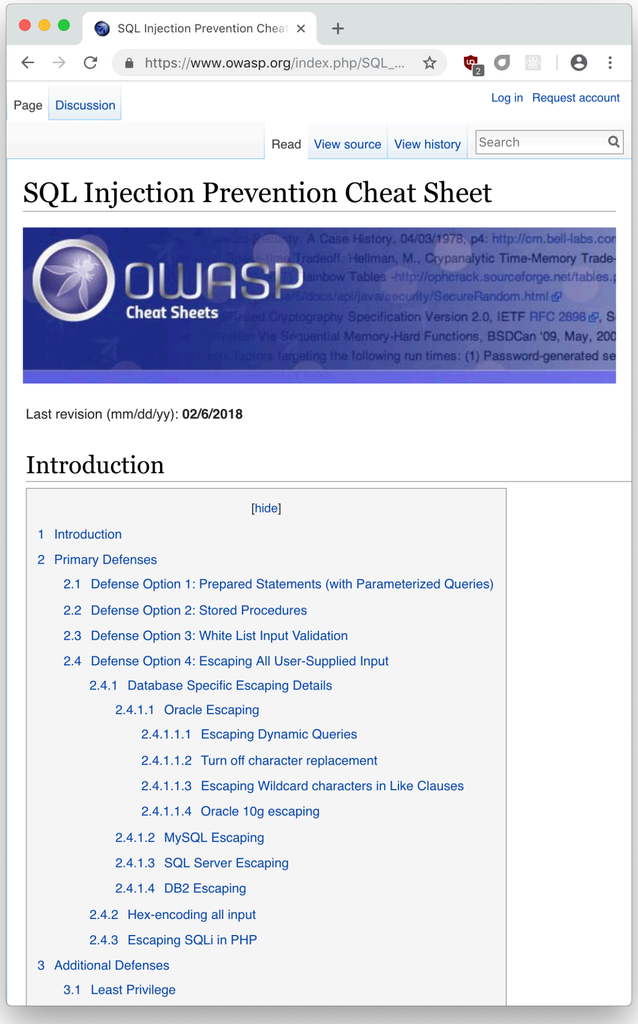
Problem: SQL Injections

```
' union select
username,
password from
users--
```

```
select name,
notes from notes
where title =
'cats' union
select username,
password from
users--'
```







WALLIX DATAPEPS

Problem: Cleartext Data (and passwords)

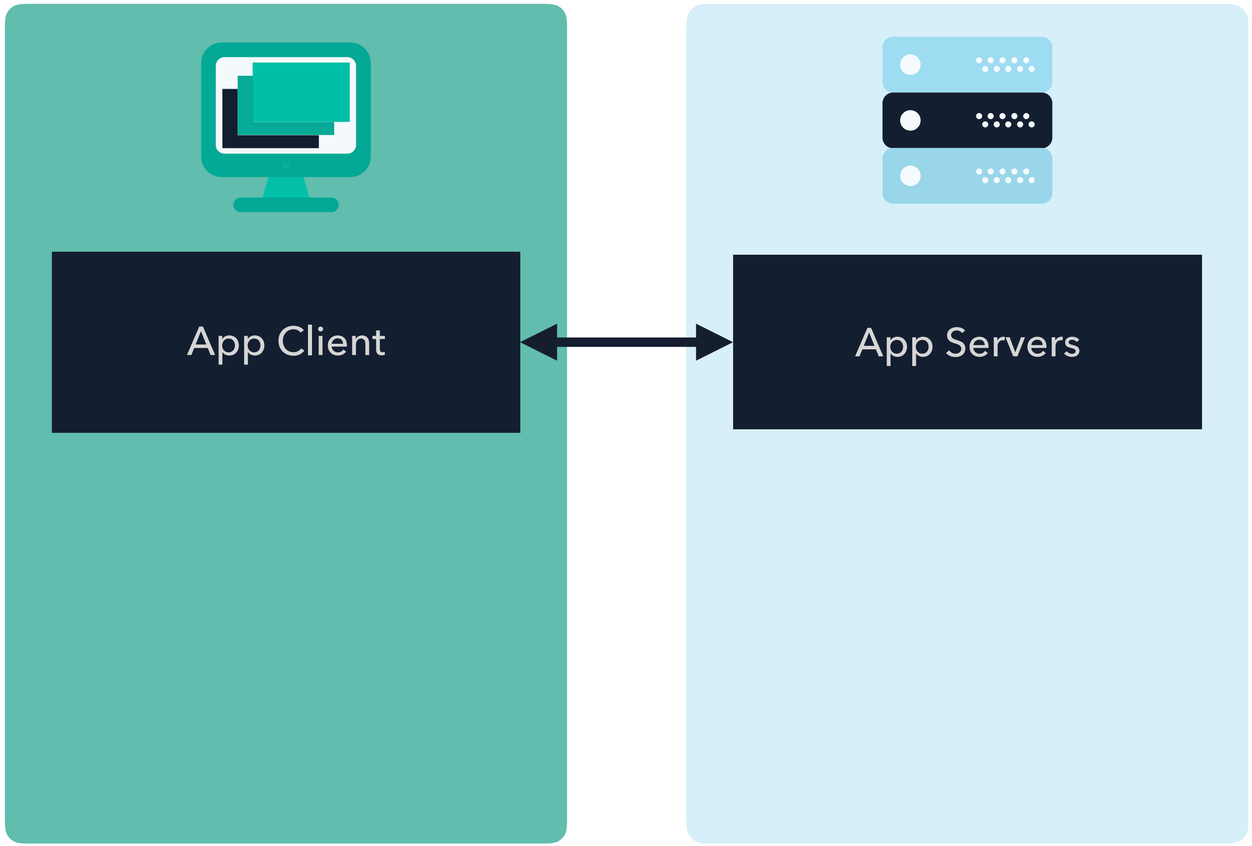
and let's not forget

#EVILADMIN

WALLIX
DATAPEPS

The Solution

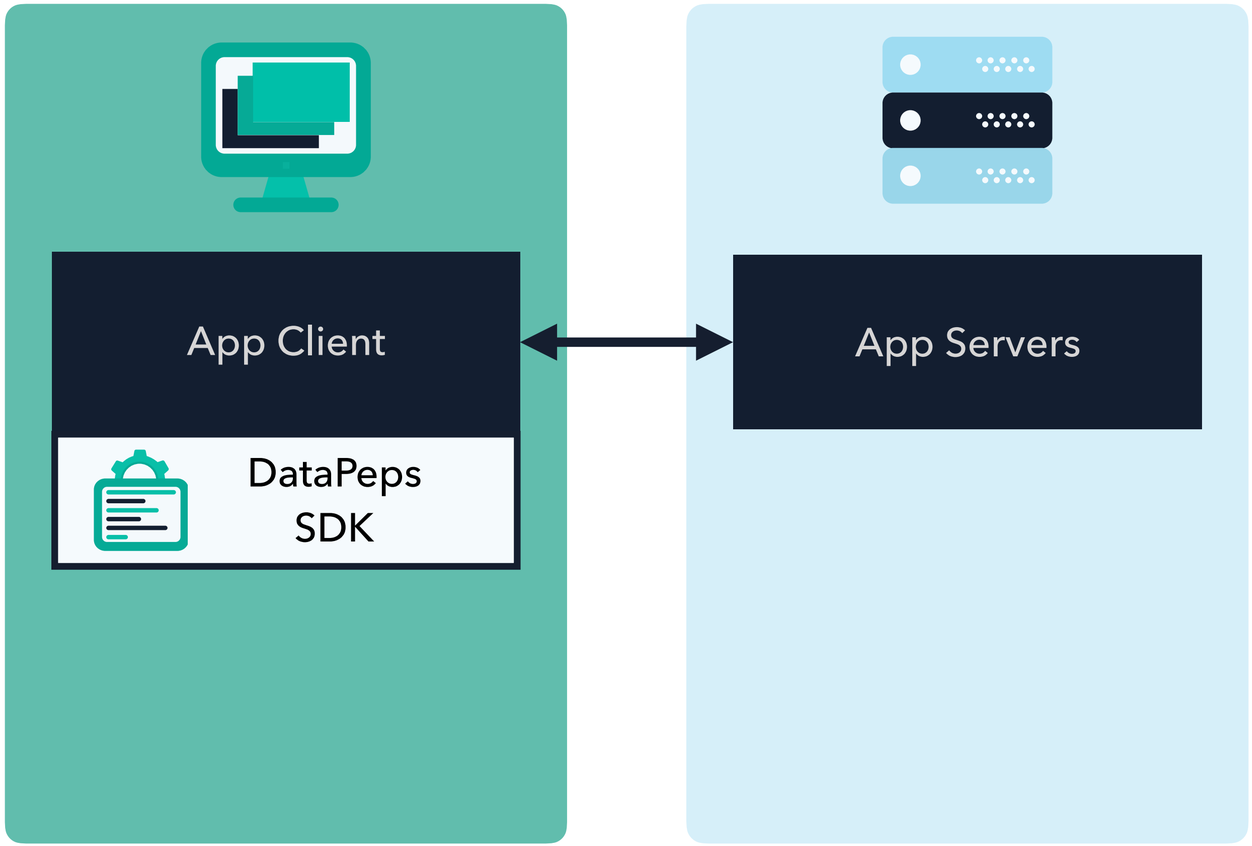
DataPeps



WALLIX
DATAPEPS

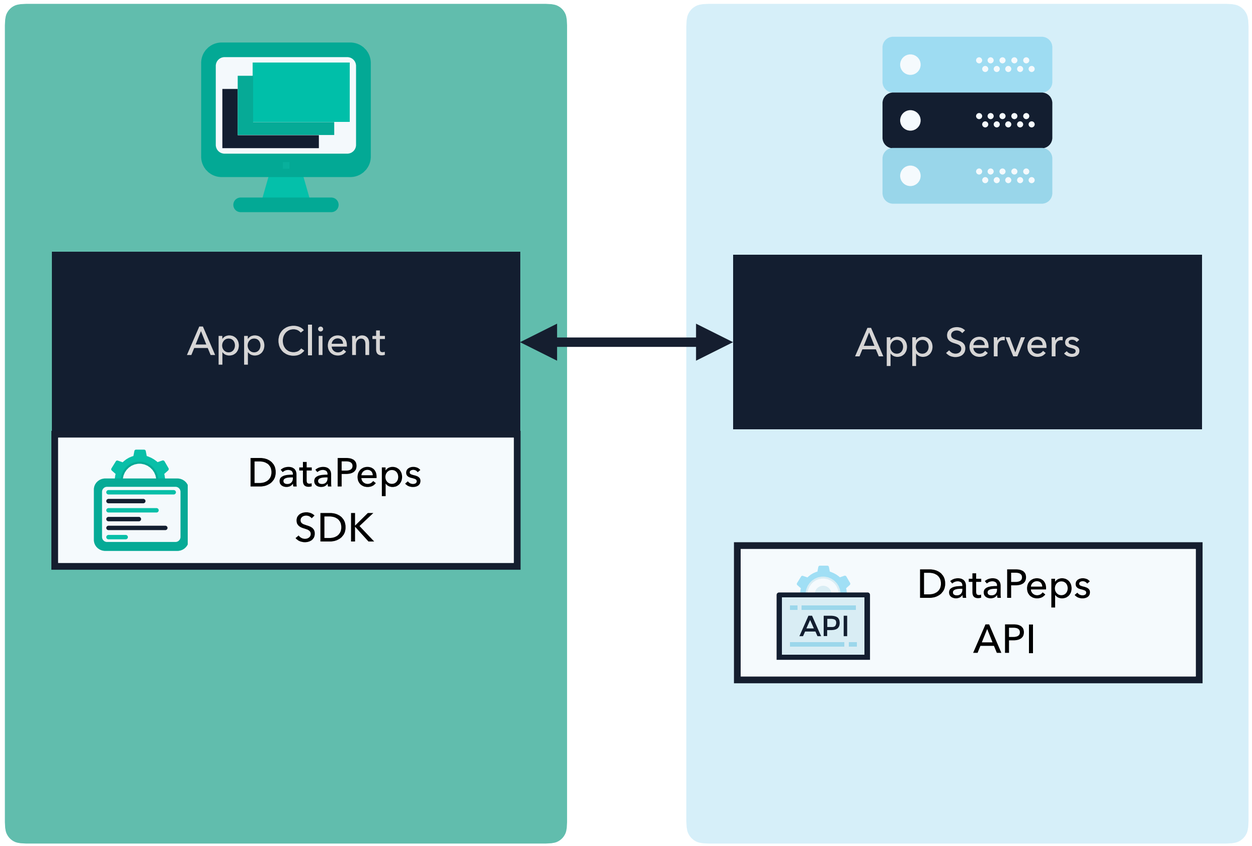
DataPeps

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



WALLIX DATAPEPS

DataPeps

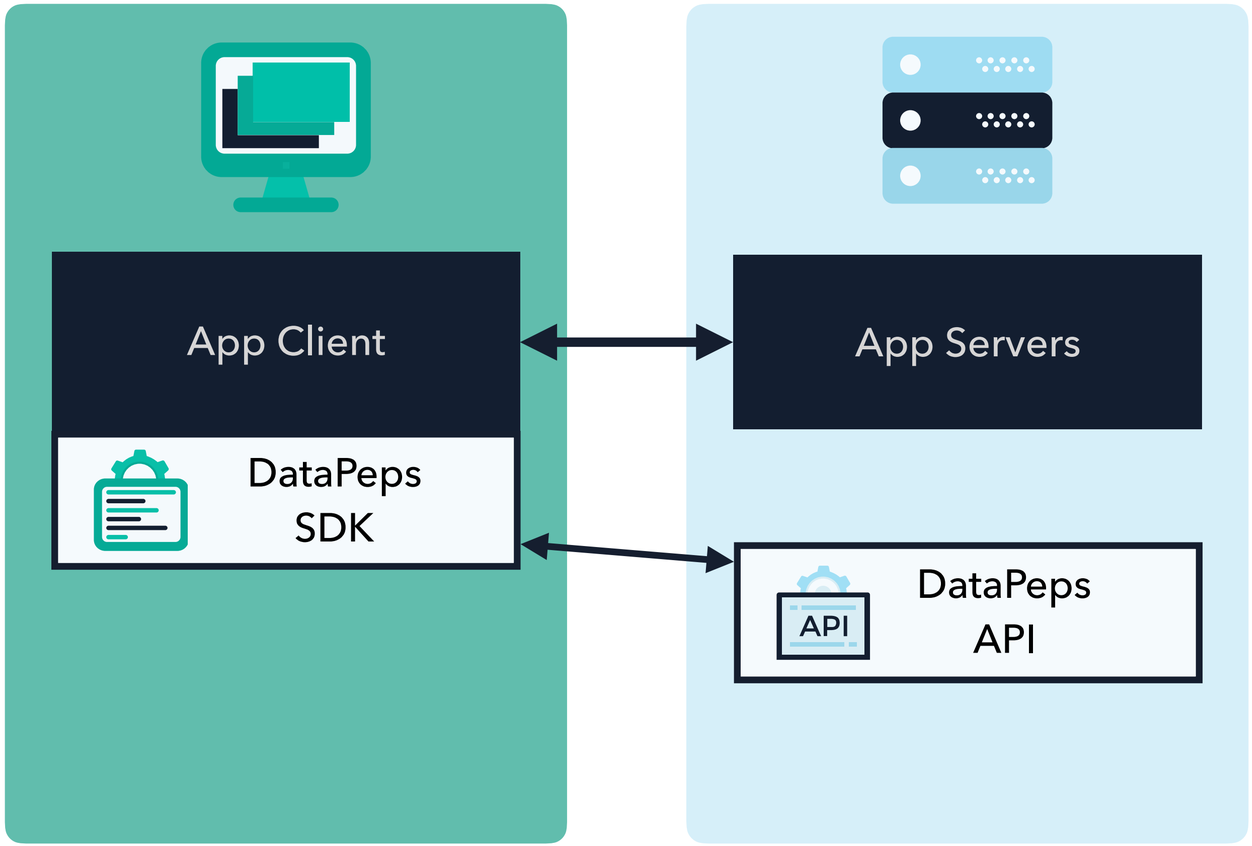


Dev(Ops) integrate DataPeps SDK into a Client App (web, desktop, …)

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

WALLIX DATAPEPS

DataPeps



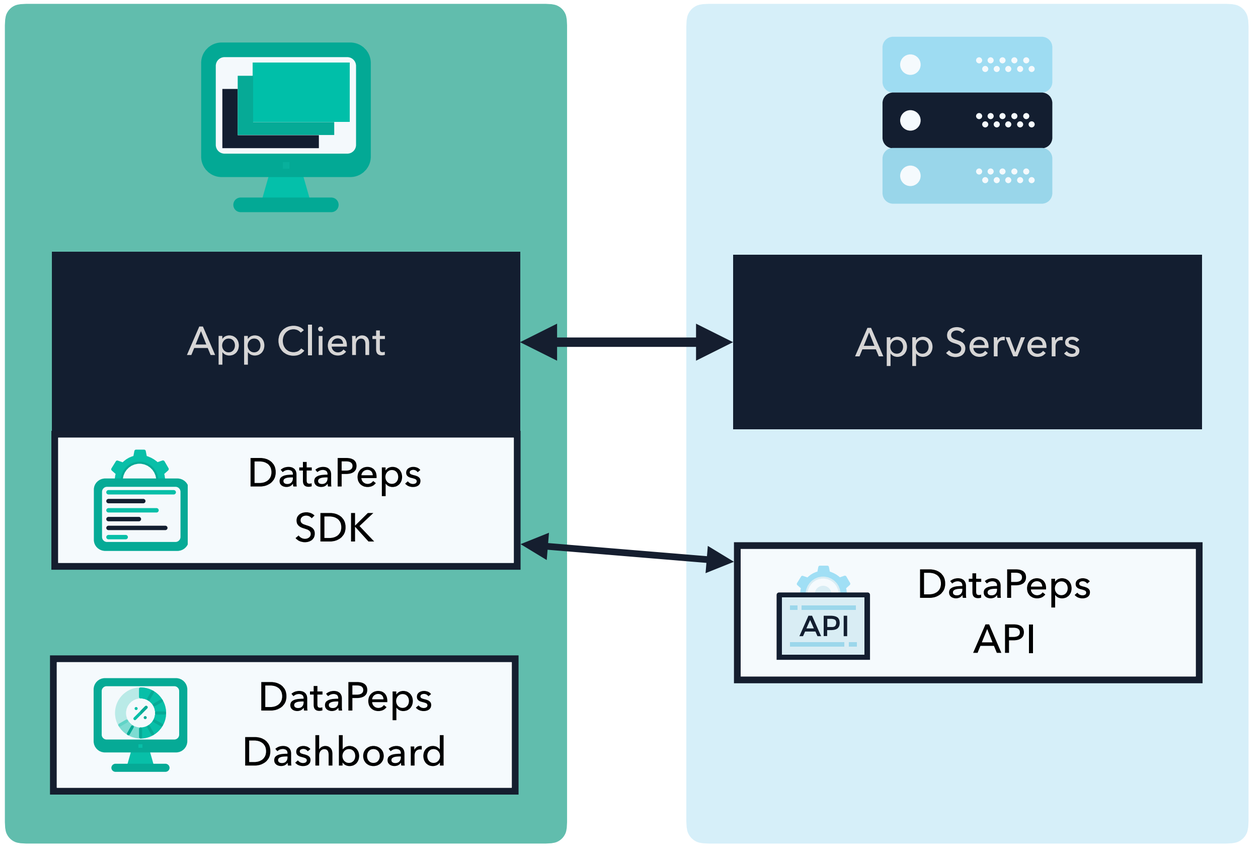
Dev(Ops) integrate DataPeps SDK into a Client App (web, desktop, …)

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

WALLIX DATAPEPS

DataPeps

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

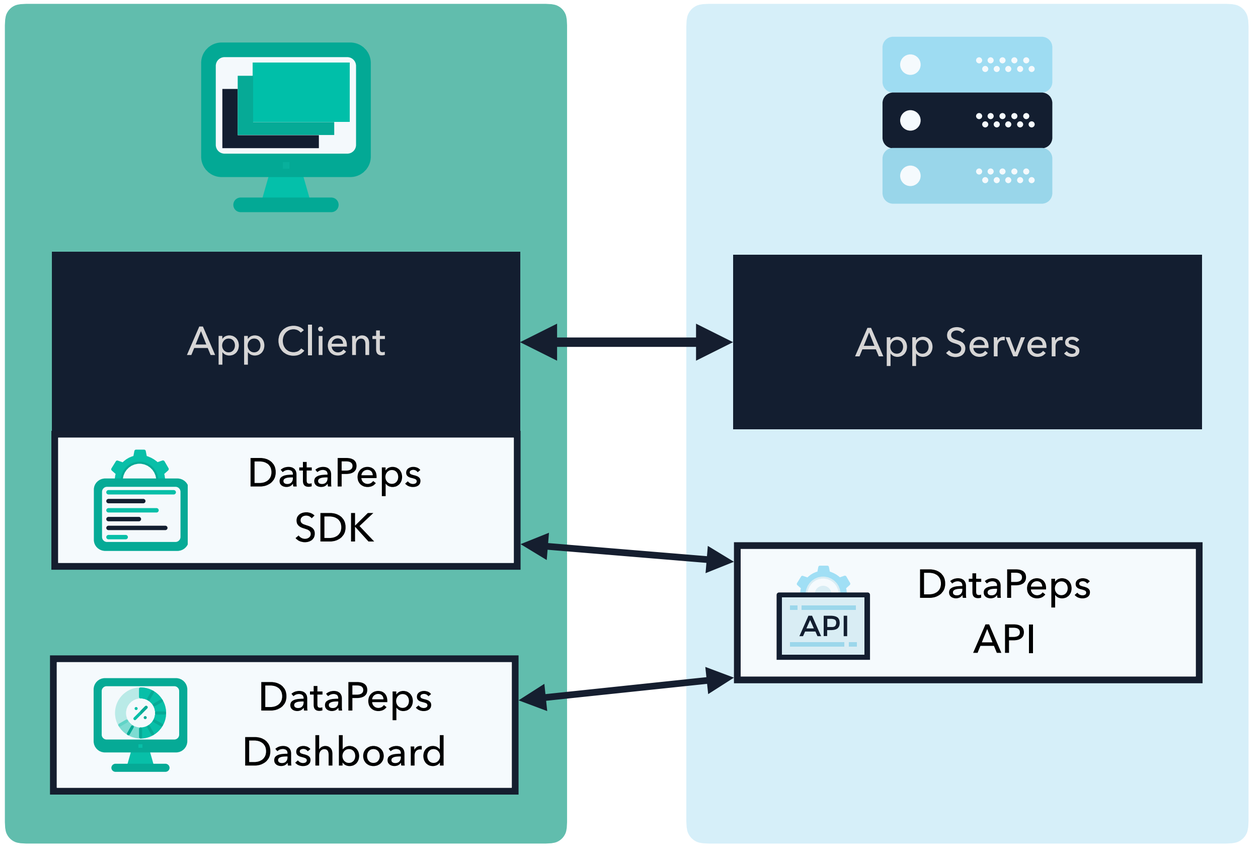


The DataPeps API service can run in the cloud or on-premises

WALLIX DATAPEPS

DataPeps

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

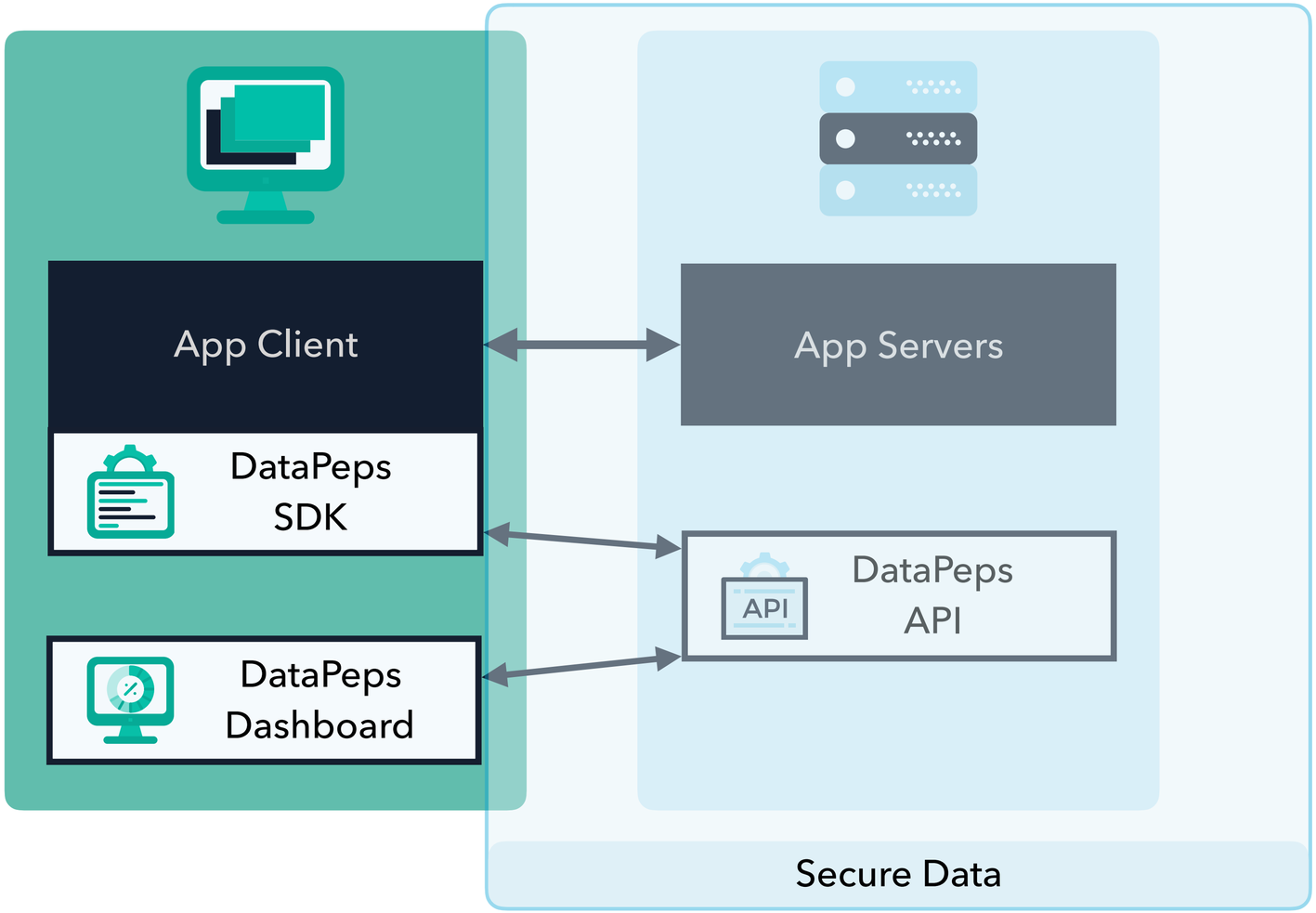


The DataPeps API service can run in the cloud or on-premises

WALLIX DATAPEPS

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

WALLIX DATAPEPS

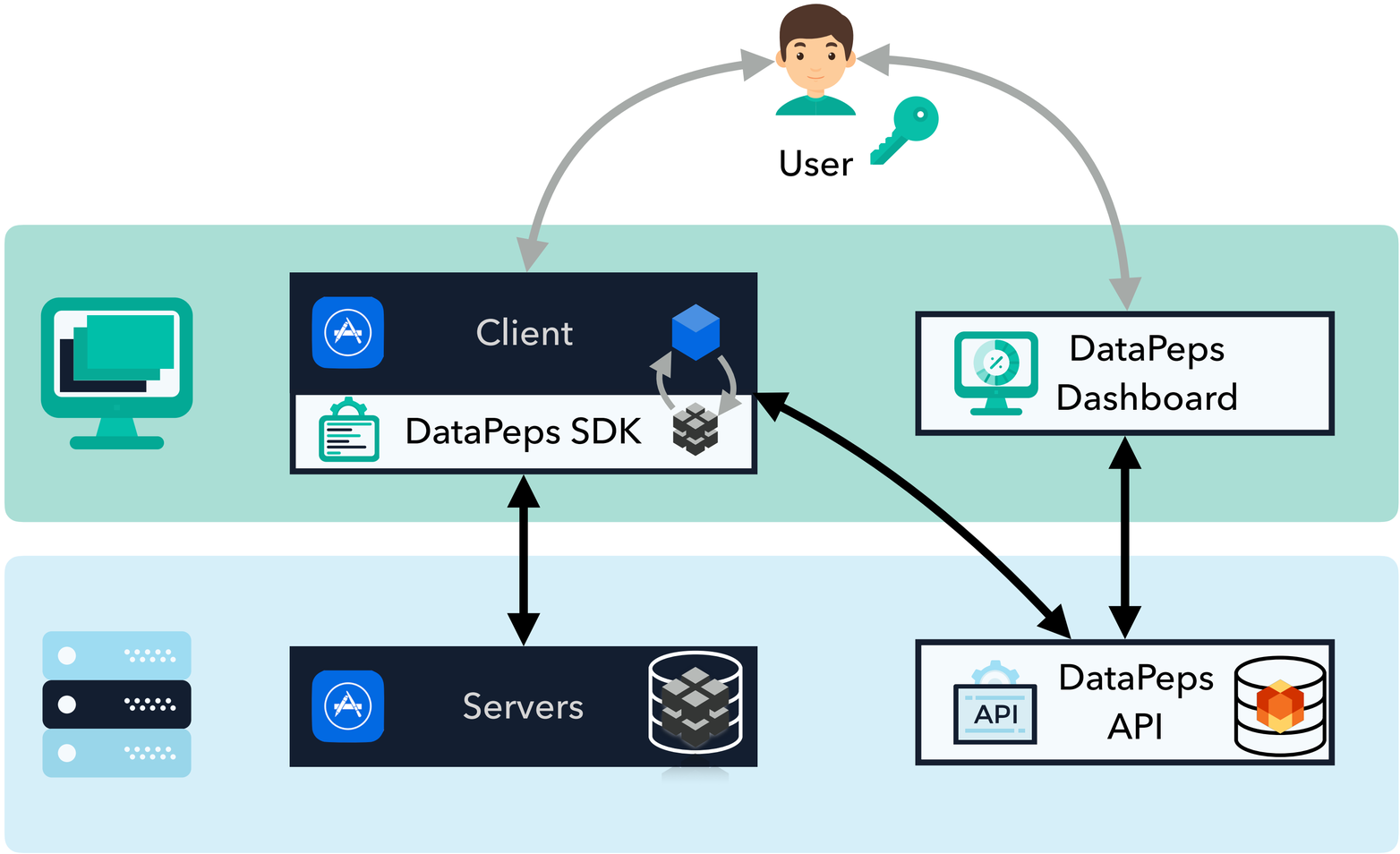
Architecture

Data

Encrypted Data

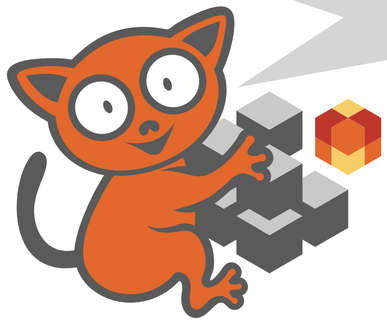
User Secret

DataPeps Resource



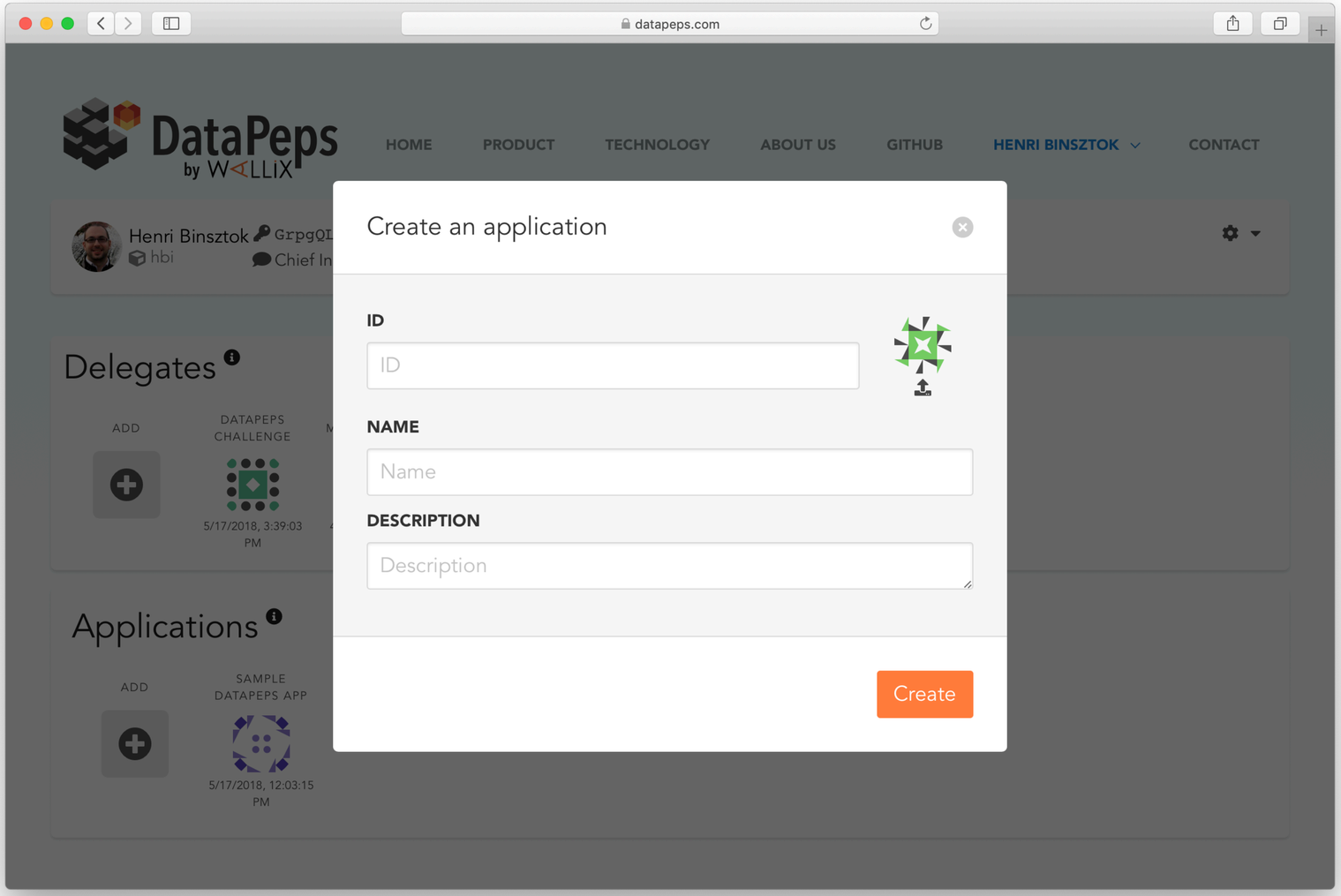
WALLIX DATAPEPS

The Developer Mission



WALLIX DATAPEPS

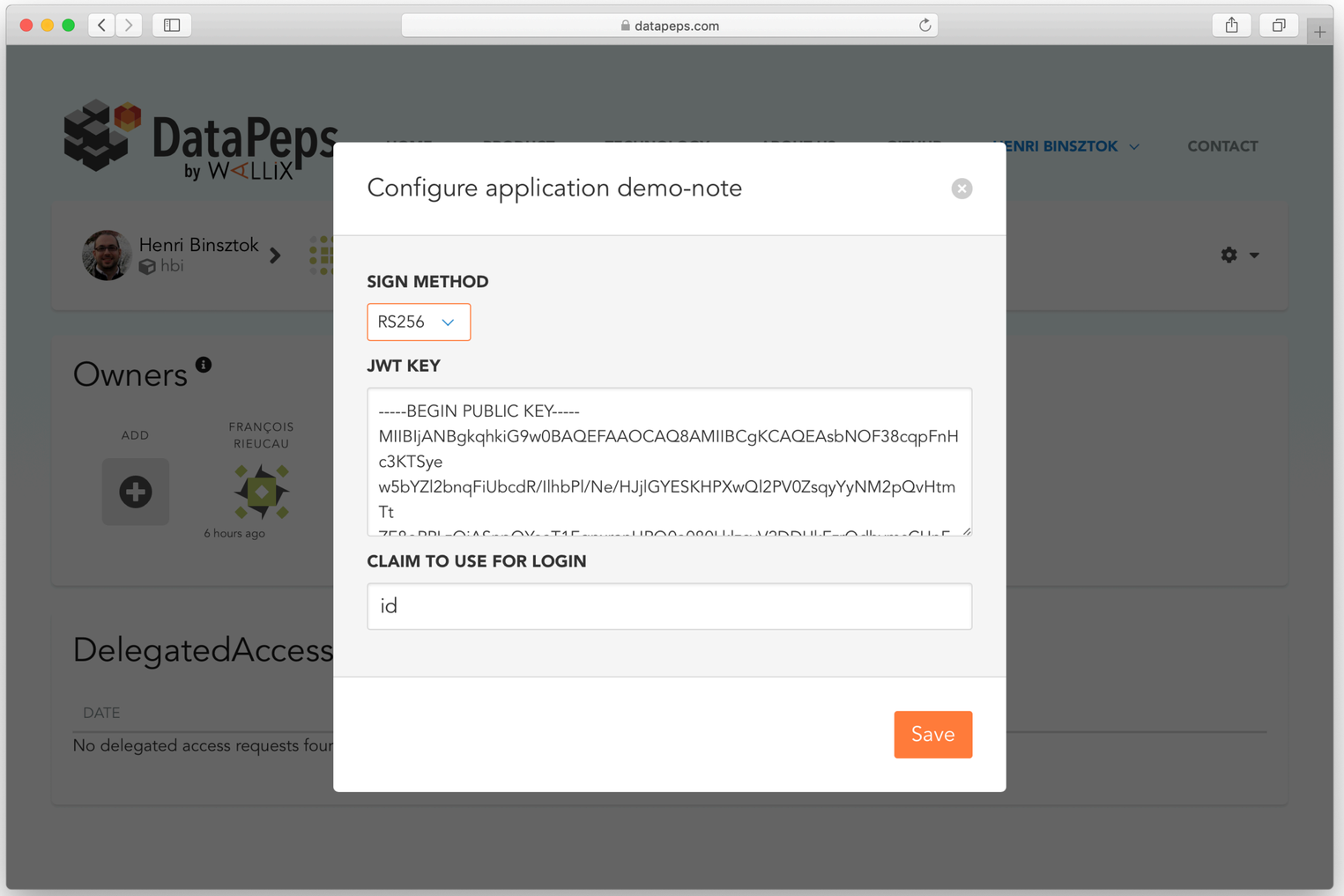
1 Create your App at DataPeps.com



WALLIX DATAPEPS

2

Configure the JWT Public Key



WALLIX DATAPEPS

3

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("/");
```

WALLIX DATAPEPS

4

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);
  …
```

WALLIX DATAPEPS

5

Update note.js

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

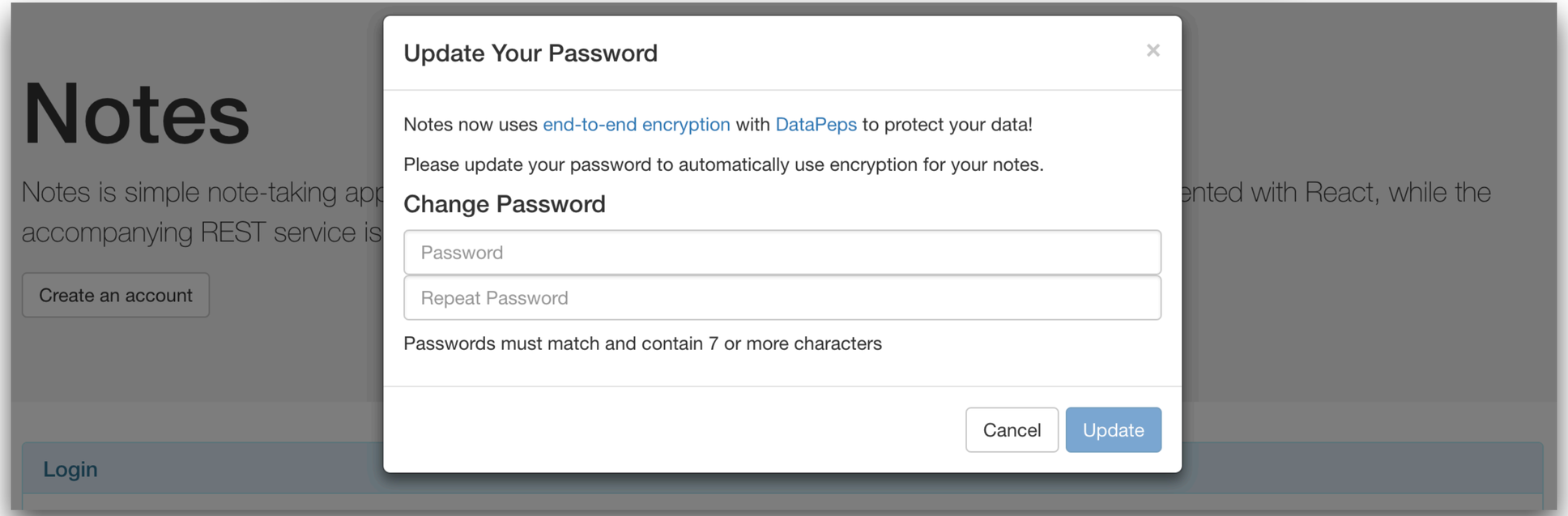
```
class Note extends React.Component {

  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 (_) {}
 }
```

WALLIX DATAPEPS

Bonus: Add new features



WALLIX DATAPEPS

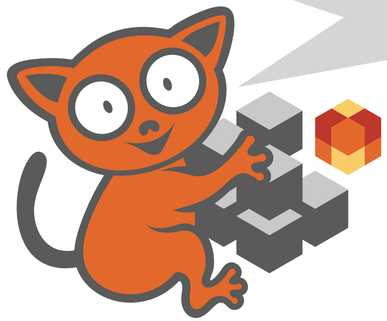
Nothing else: No Server Modifications

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

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

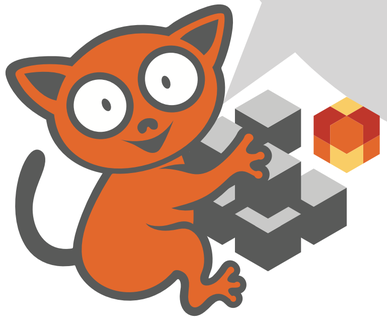
WALLIX DATAPEPS

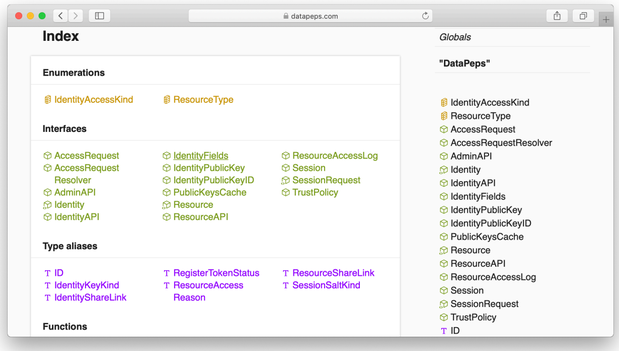
The Demo



WALLIX
DATAPEPS

datapeps.com

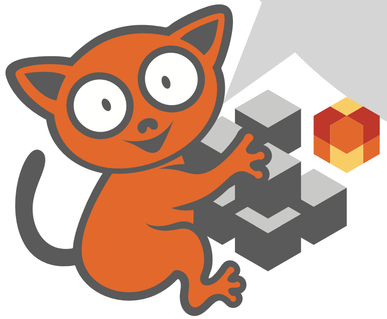


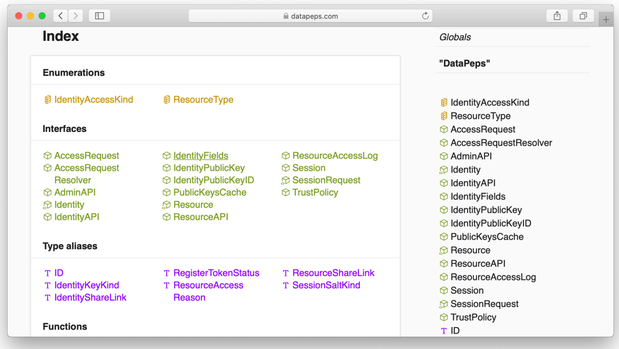


SDK Documentation

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

datapeps.com





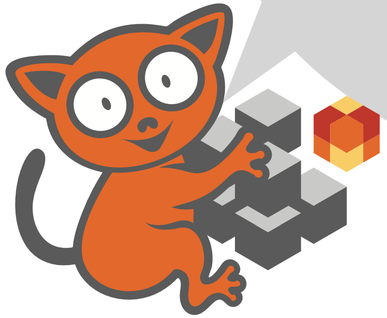
SDK Documentation

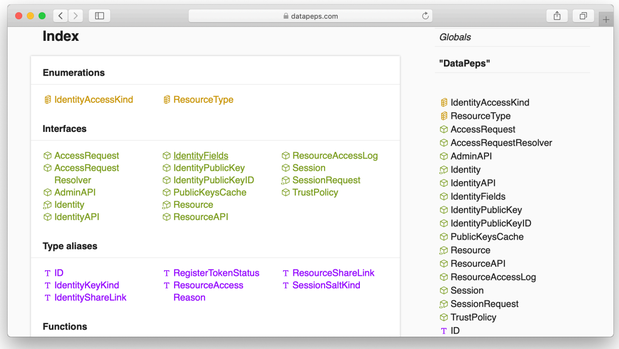
https://datapeps.com/docs/sdk/js

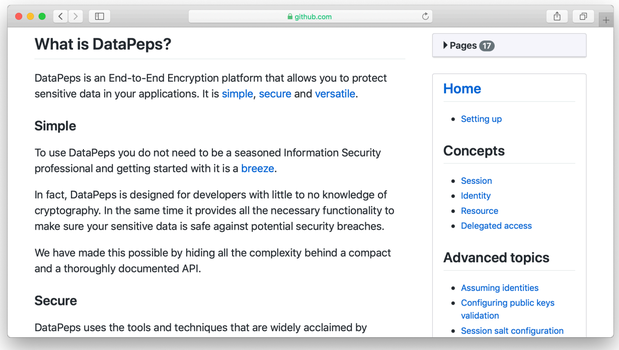
Tutorial

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