



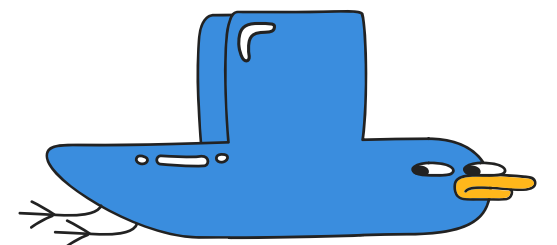
Présentation du titre

CONCEPTEUR DEVELOPPEUR D'APPLICATIONS

Audrey Ritzenthaler

BLOOMAYS PURCHASE SALE

On va vite pour le
sourcing, on prend
notre temps pour le
recrutement





Mehdi Lebel

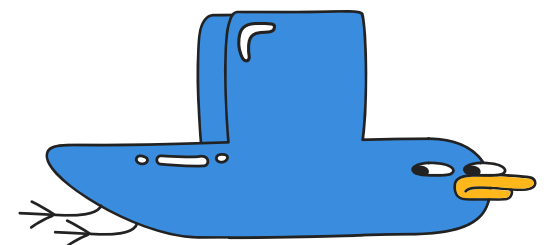
CEO Bloomays



Loïc Calvy

CTO Bloomays

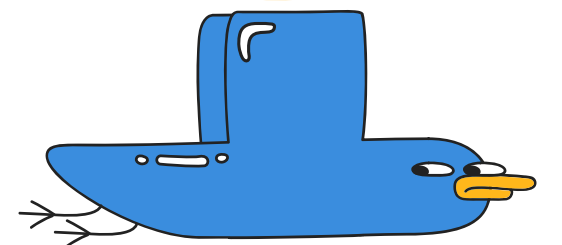
“Quoi de mieux que des techs pour recruter d'autres techs ?”



Le point technique

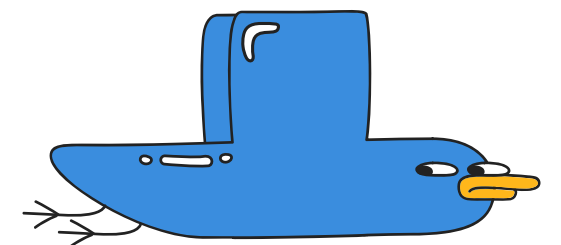
Ma user story

- En tant que bloomer
- Je veux importer un fichier
- Afin de pouvoir stocker mes CRAs/factures



Upload ApolloServer + AWS S3

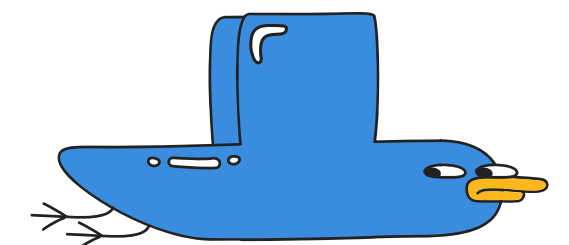
- Incompatibilité avec node 14 et +
- Type scalaire Upload
- Objet "file" reçu en argument
- CreateReadStream
- Amazon sdk



Incompatibilité avec node 14 et +

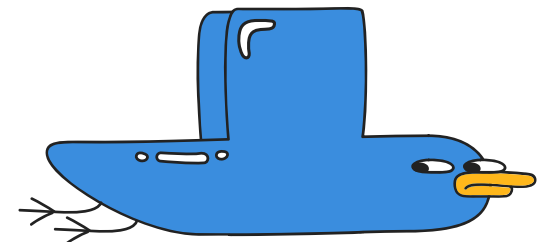
```
const { graphqlUploadExpress } = require('graphql-upload');  
expressServer.use('/', graphqlUploadExpress({ maxSize: 5000000 }));
```

```
const apolloServer = new ApolloServer({  
  schema: schemaWithResolversAndDirectives,  
  context,  
  uploads: false  
});
```



Type scalaire Upload

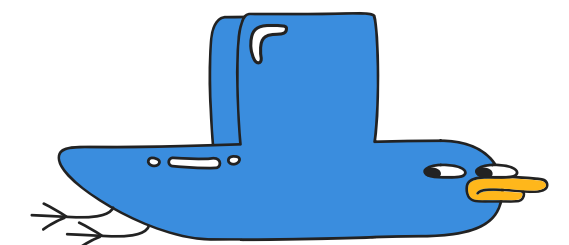
```
input upload {  
  activityRecordId: String!  
  file: Upload!  
  activityFileType: ActivityFileType!  
}  
  
type Mutation {  
  uploadActivityFile(input: upload!): Boolean @auth(requires: Bloomer)  
}
```



Objet "file" reçu en argument

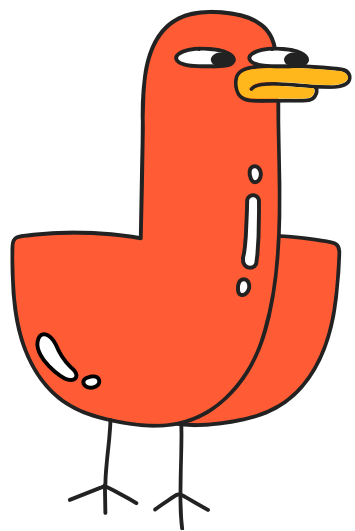
```
const { createReadStream, mimetype, filename } = await args.file;
const fileStream = createReadStream();
const path = getActivityS3Path(activity, args.activityFileType, filename);

let result;
try {
  result = await fileUpload(fileStream, mimetype, path);
} catch (errOnStorage) {
  throw new ApolloError(
    'Unable to store the uploaded file',
    'FILESTORAGE_ERROR'
  );
}
```

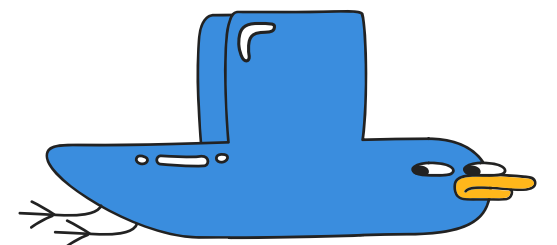


CreateReadStream

CreateReadStream() est le stream qui contient le contenu du fichier téléchargé



On rappelle qu'un stream est une méthode de transfert de grandes quantités de données sur des appareils mobiles ou des sites Web en décomposant le fichier ou les données en morceaux gérables



Amazon SDK

```
const aws = require('aws-sdk');

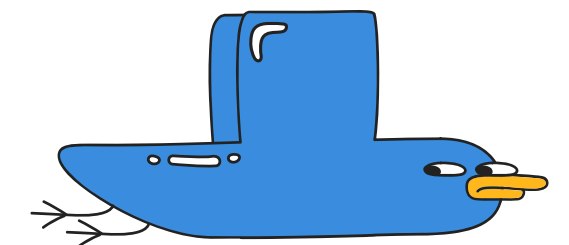
aws.config.update({
  secretAccessKey: process.env.AWS_SECRET_ACCESS_KEY,
  accessKeyId: process.env.AWS_ACCESS_KEY_ID,
  region: process.env.AWS_REGION
});

const s3 = new aws.S3();

module.exports = s3;
```

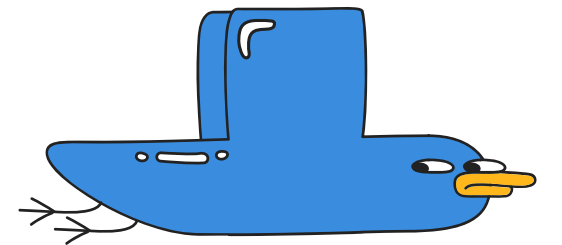
```
const fileUpload = async (stream, mimetype, pathname) => {
  const params = {
    Bucket: BUCKET,
    Key: `${pathname}`,
    Body: stream,
    ContentType: mimetype
  };

  return await s3.upload(params).promise();
};
```

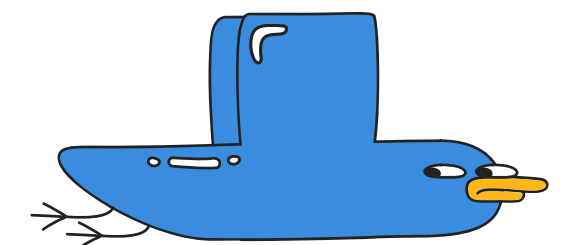
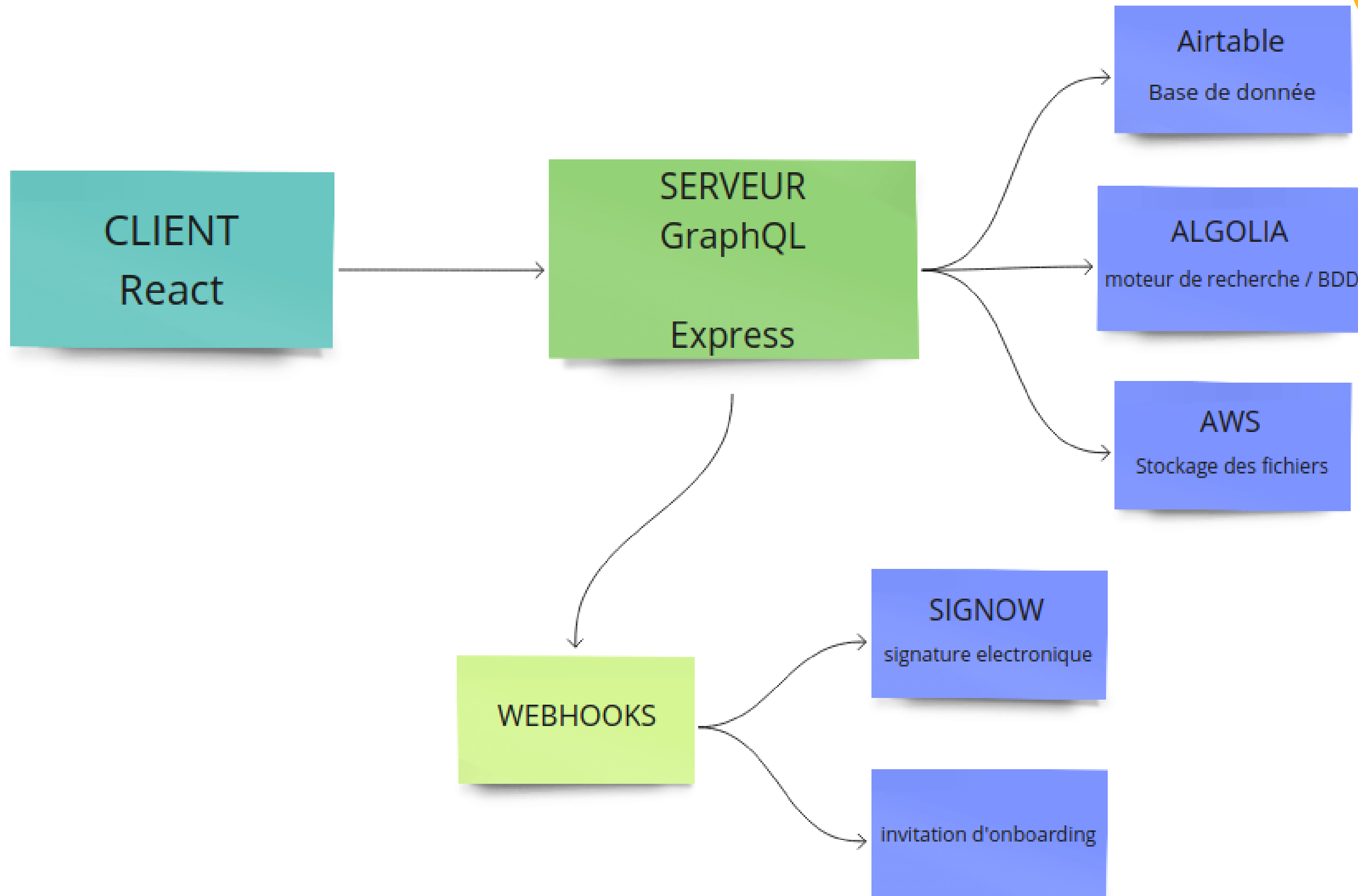


ARCHITECTURE DE L'APP

Présentation de
l'architecture de
l'application dans sa
globalité



Architecture



Plan de présentation

Présentation des projets

Bloomays

Tamplo

Organisation du projet

Scrum

Users stories

Maquettes

Front-end

React - React Router

React Native

Authentification

Tests

Back-end

Express - GraphQL

Tests

Base de donnée

Airtable - Algolia

MongoDB

Architecture

CI/CD

Docker

Micro-service

AWS

Sécurité

