

I-ON Teams

Grupo 4

Afonso Machado – 47217

Martim Francisco – 46912

Orientador: Pedro Félix

i-on Teams

- Iniciativa **i-on**
 - Sistemas para apoio a atividades académicas
 - Desenvolvimento e disponibilização *Open-source*
 - Múltiplos sistemas e aplicações
- **i-on Teams**
 - Serviço de gestão de entregas de trabalhos práticos através do GitHub
 - Usável por docentes e alunos

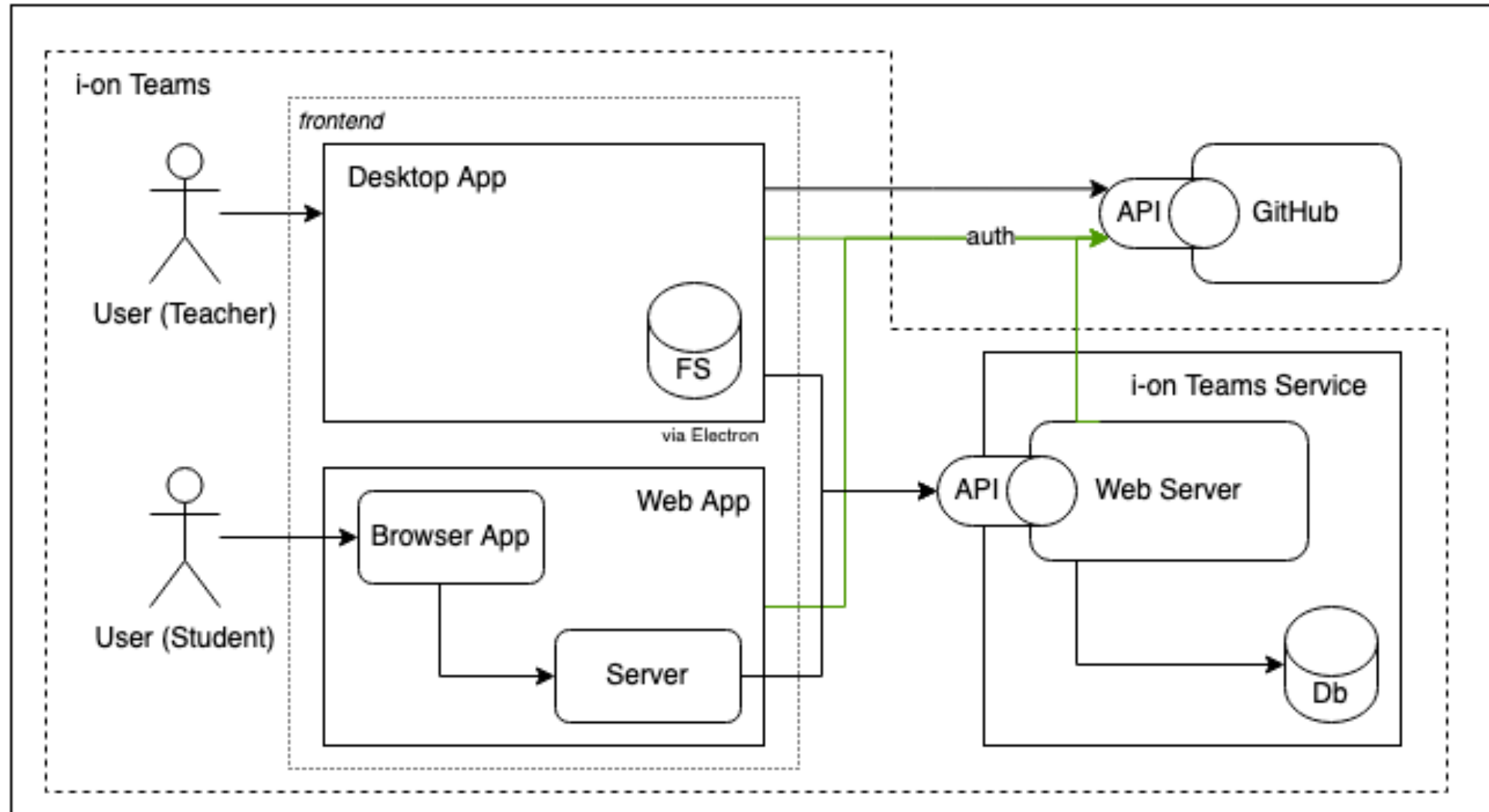
Motivação

- Lacunas do GitHub Classroom
 - Falta de associação entre o aluno e o seu nome de utilizador no GitHub
 - Dificuldade na verificação das entregas
 - Acesso ao código de vários grupos complexo para o docente
 - Falta de centralização da informação relevante
- Lacunas do i-on Codegarten (Projeto realizado em 2021 com um propósito semelhante)
 - Armazenamento de informação sensível no servidor
 - Impossibilidade do docente obter cópias dos trabalhos

Características e Funcionalidades

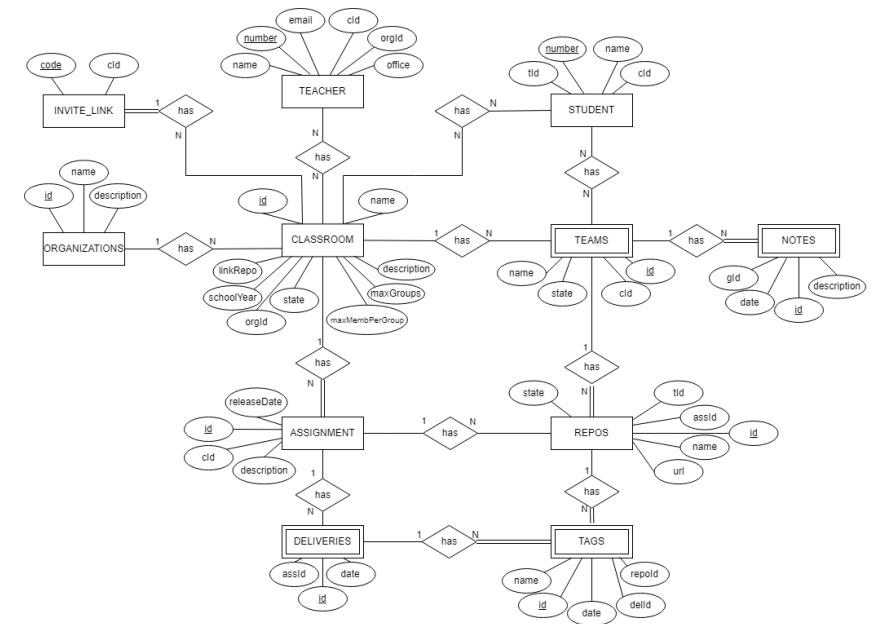
- Aplicações Desktop (em *ElectronJS*) e Web
 - Web para consulta por parte de alunos
 - Desktop para consulta e gestão por parte do docente
- i-on Teams Service
- Gestão de turmas, trabalho, grupos e entregas
 - Acesso controlado de alunos a turmas/grupos
 - Obtenção de todos os trabalhos de uma turma (múltiplos *git pull*)
- Omissão de informação sensível no serviço

Arquitetura do Sistema



Modelo de Dados

- Utilização do modelo relacional e do SGBD PostgreSQL
 - 11 entidades, 12 associações, 13 tabelas
- Recurso a vistas, funções e *triggers*
 - Ex. para implementação de soft deletes
 - Ex. geração automática de marcas temporais

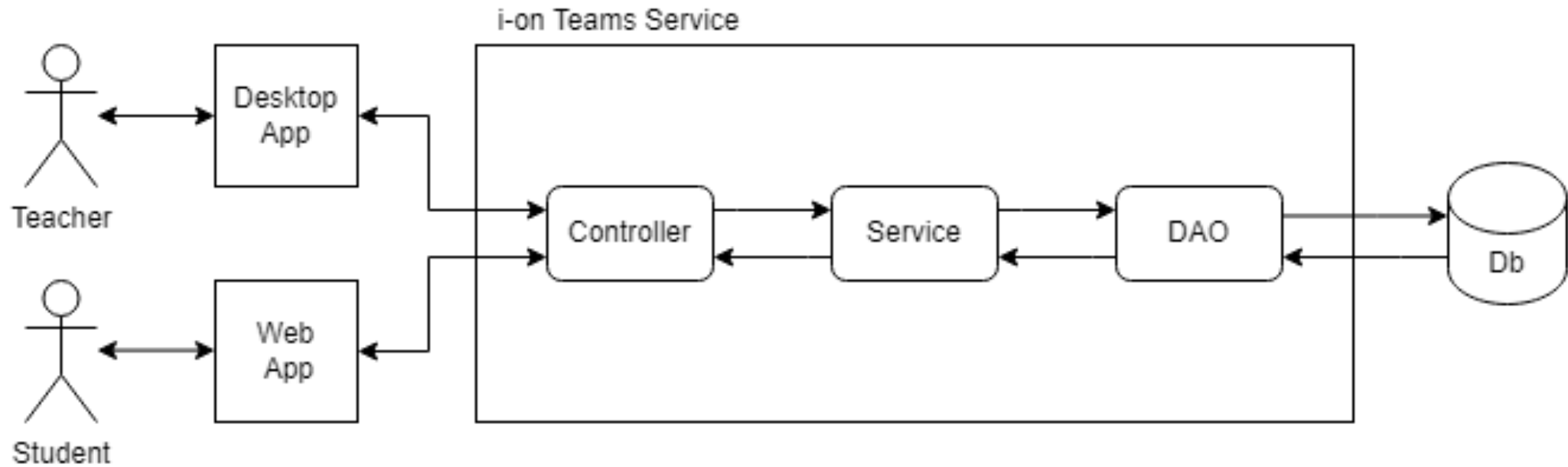


Modelo EA

i-on Teams Service

- Expõe uma Web API comum a todos os utilizadores
 - Implementada com recurso à framework *Spring Boot*
- Acesso a dados através da API declarativa da biblioteca JDBI
- Autentica utilizadores através da Web API do GitHub

i-on Teams Service – Arquitetura

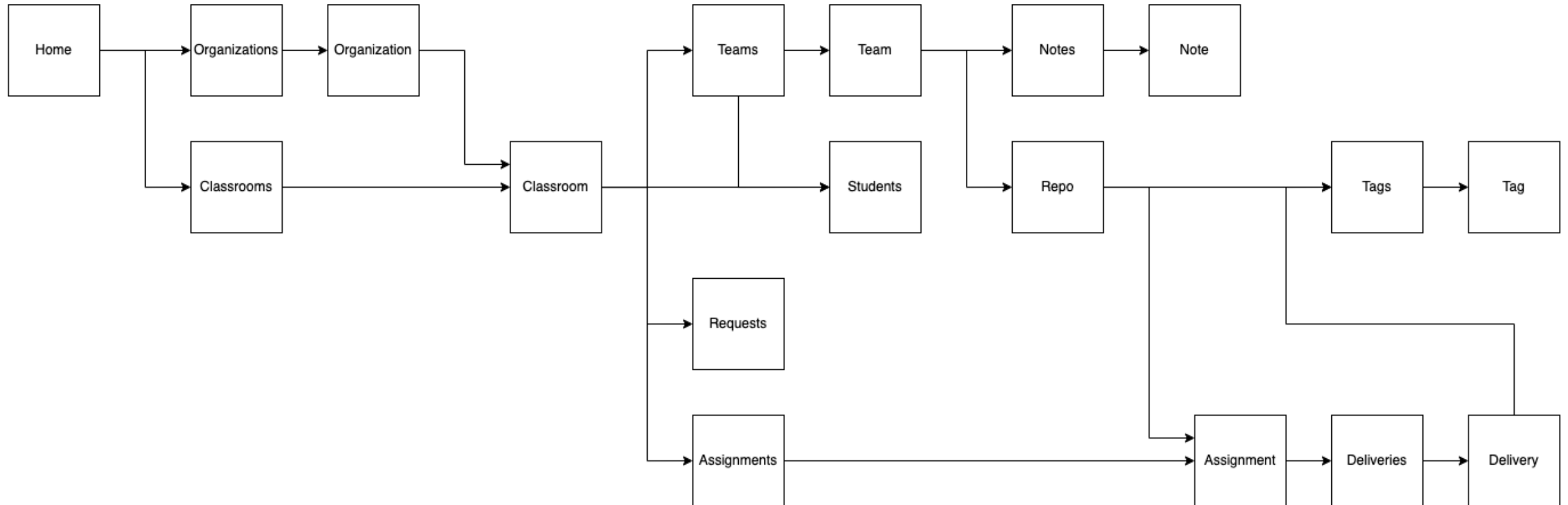


Desenho API

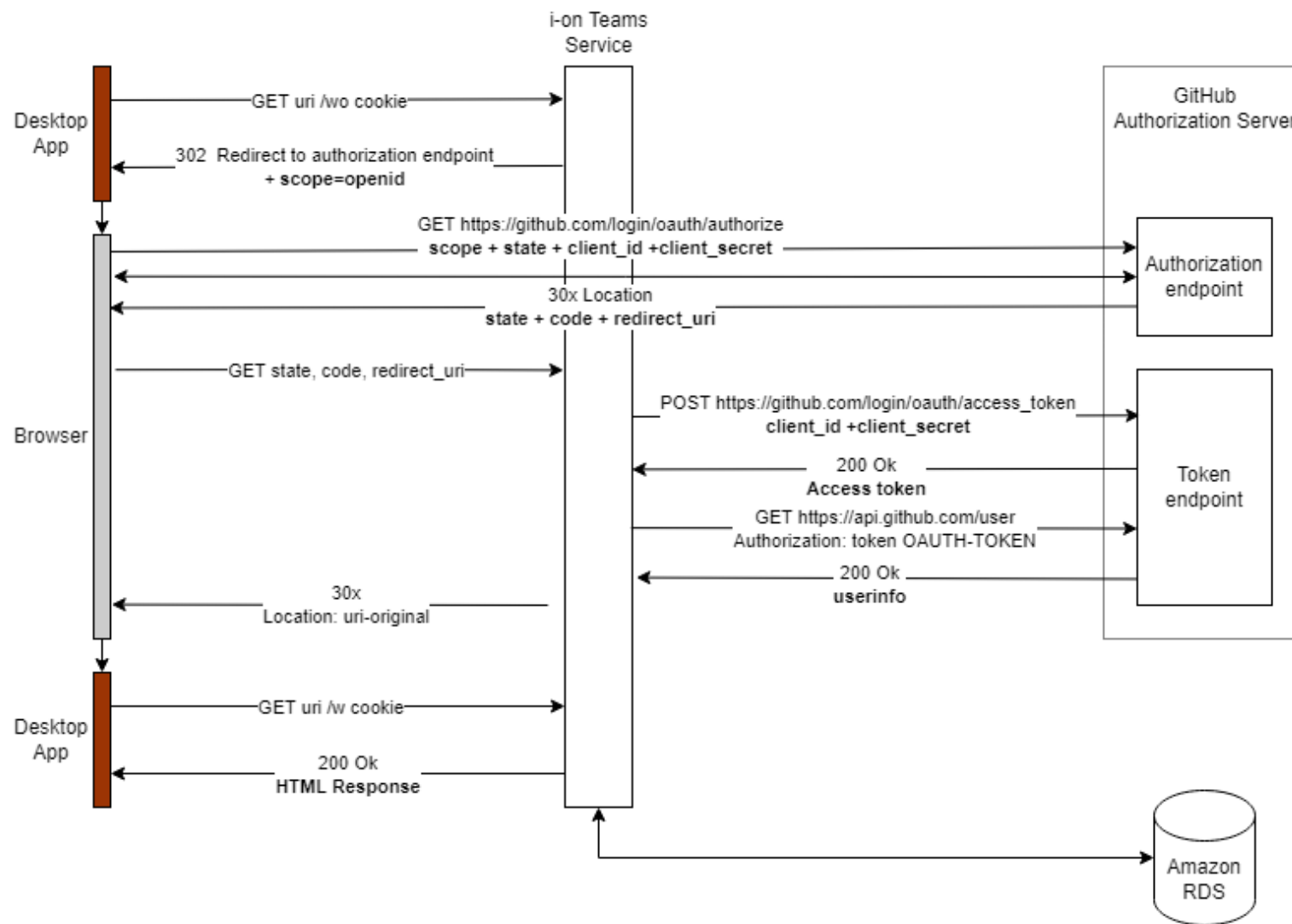
- API desenvolvida com recurso a *hypermedia*
- Em caso de sucesso o tipo de resposta será:
 - GET: application/siren+json
 - POST e PUT: application/json
 - DELETE: sem corpo de resposta
- Em caso de erro o tipo de resposta será application/problem+json

Guia da API – Professor

Teacher signed in



Autenticação



Documentação

- Documentação da [API por recurso](#)
 - Vocabulário existente em cada recurso (*domain specific* ou *standard*)
 - Especificação dos *endpoints* com recurso a exemplos de resposta
 - Descrição dos tipos de erro esperados
- Documentação dos [tipos de erros](#) implementados
- Documentação adicional referente ao [modelo de dados](#), [autenticação](#), etc. através de diagramas

Progresso – Presente

- i-on Teams Service
 - Documentação da API e do modelo de dados
 - Modelo de dados totalmente implementado em PostgreSQL
 - Acesso a dados através de JDBC
 - API em *Spring Boot*
 - Endpoints implementados
 - Respostas com recurso a hypermedia
 - Testes de integração da API

Progresso – Presente

- Arquitetura de autenticação
 - Definição e documentação da arquitetura de autenticação
- Desktop App
 - Familiarização com as tecnologias ElectronJS e ReactJS
 - Início da implementação de IPC em ElectronJS
 - Esboço gráfico das páginas da aplicação

Progresso - Futuro

- i-on Teams Service
 - Melhoramentos nos testes
 - Melhoramentos na documentação da API
 - Implementação da autenticação
- Desktop App
 - Implementação em ElectronJS + ReactJS
 - Ligação à API do GitHub
 - Implementação da autenticação
- Web App
 - Implementação em ReactJS
 - Implementação da autenticação

Questões?

