The Problem

Most children in UK schools don’t learn programming

> “ **Less than 50%** of UK secondary schools teach programming ”

Key Insights

Programming is an important skill

> “ Programming jobs are growing **12% faster** than the UK average ”

Key Insights

Children love coding!

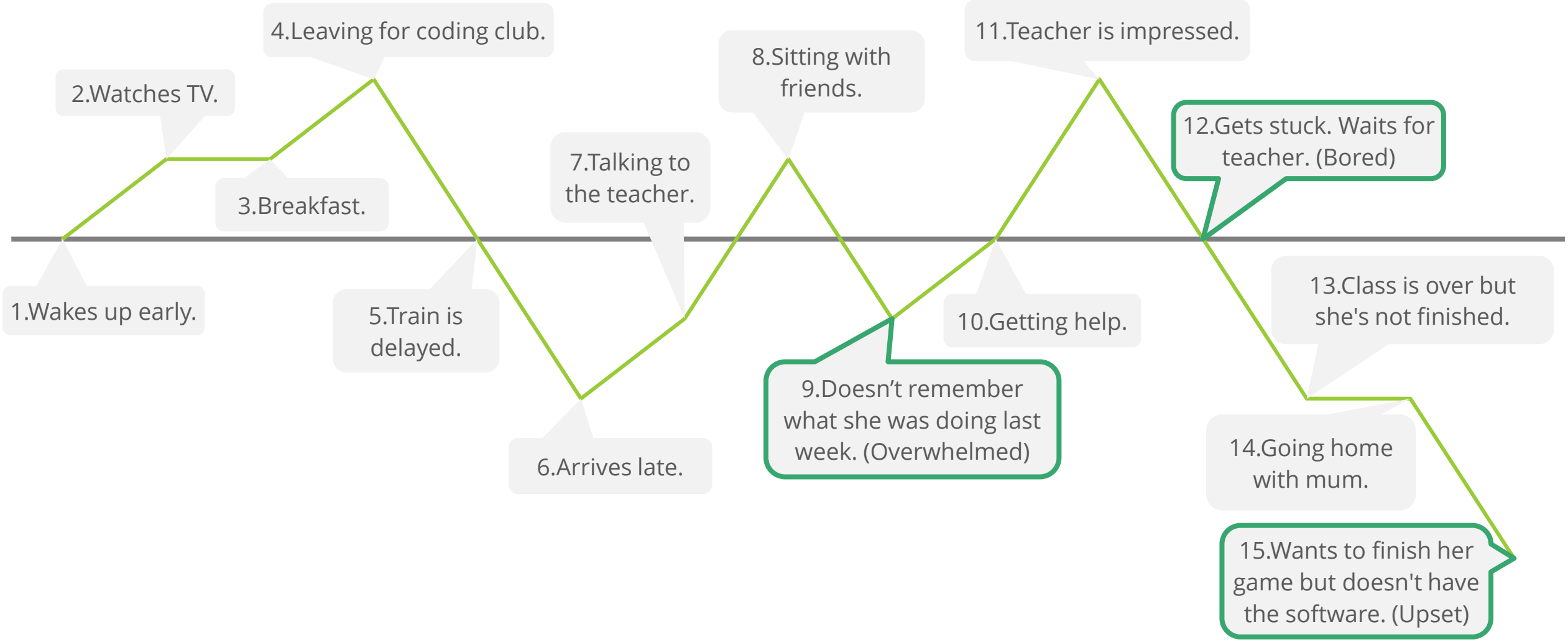
“ I like being able to use our own **creativity** to make new **interesting games**. ”

Our Target User

Jenny

- 11 years old
- Just starting secondary school
- No computing education

Jenny's Current User Journey

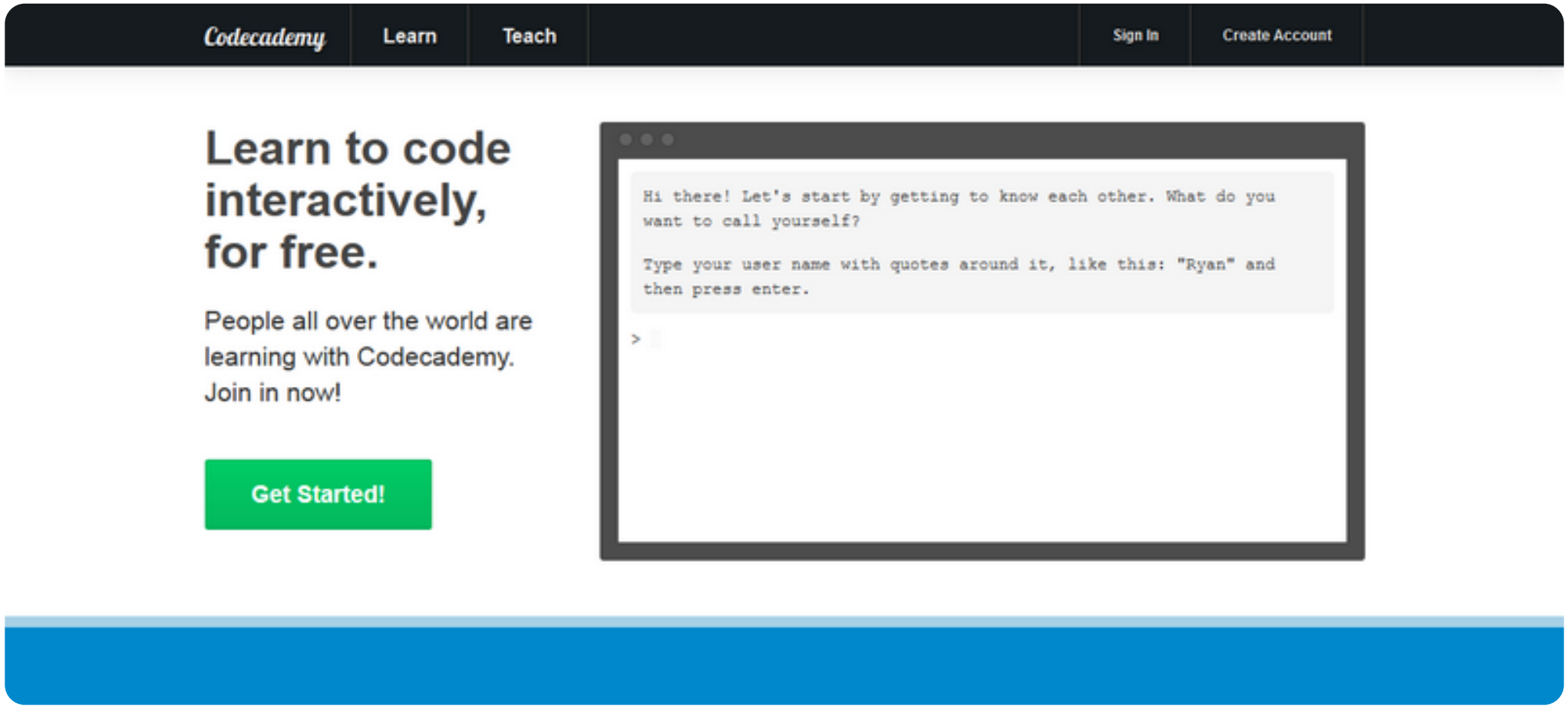


Another Key User

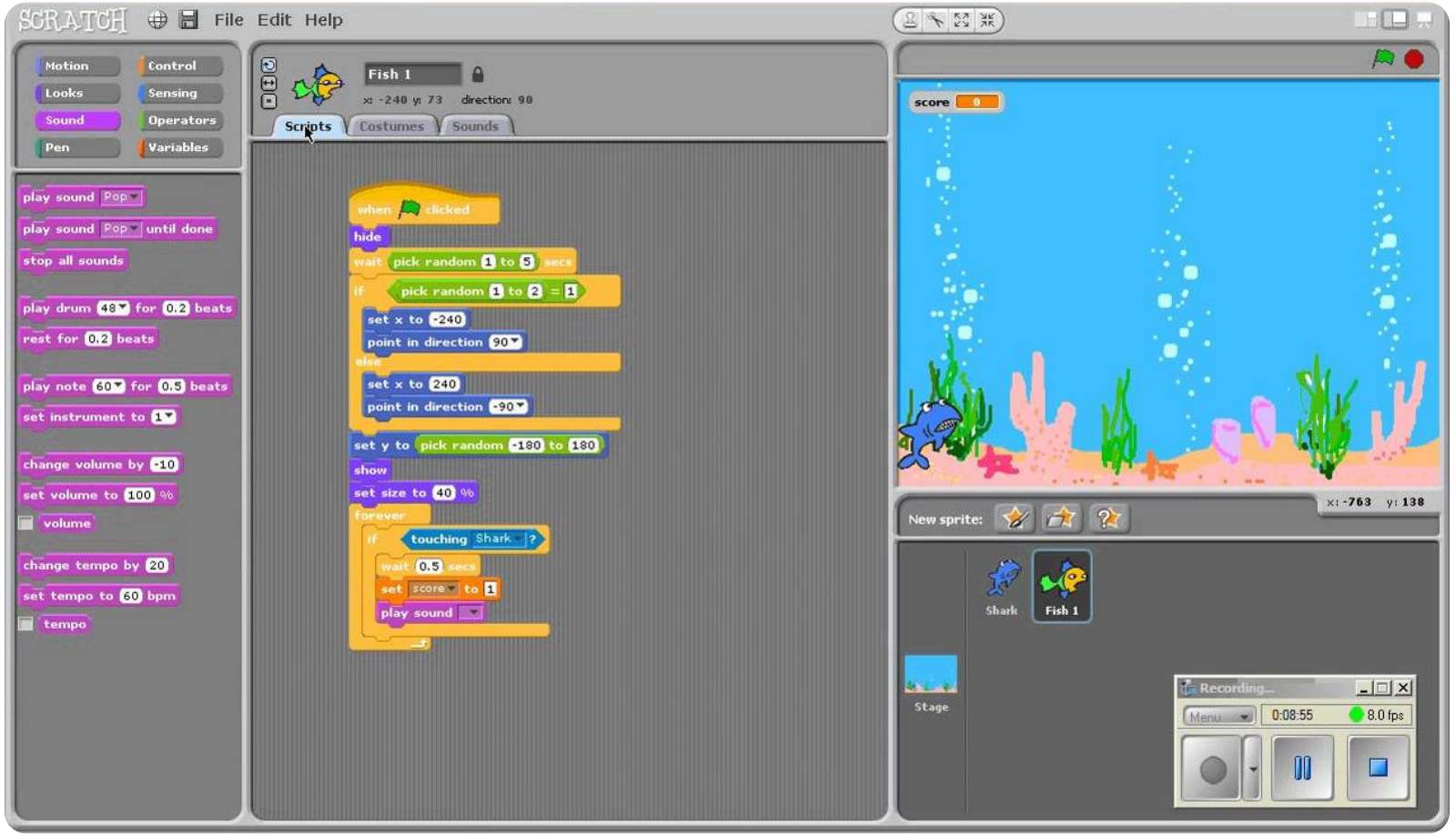
Keith

- 21 years old
- University student
- Volunteers at a local coding club

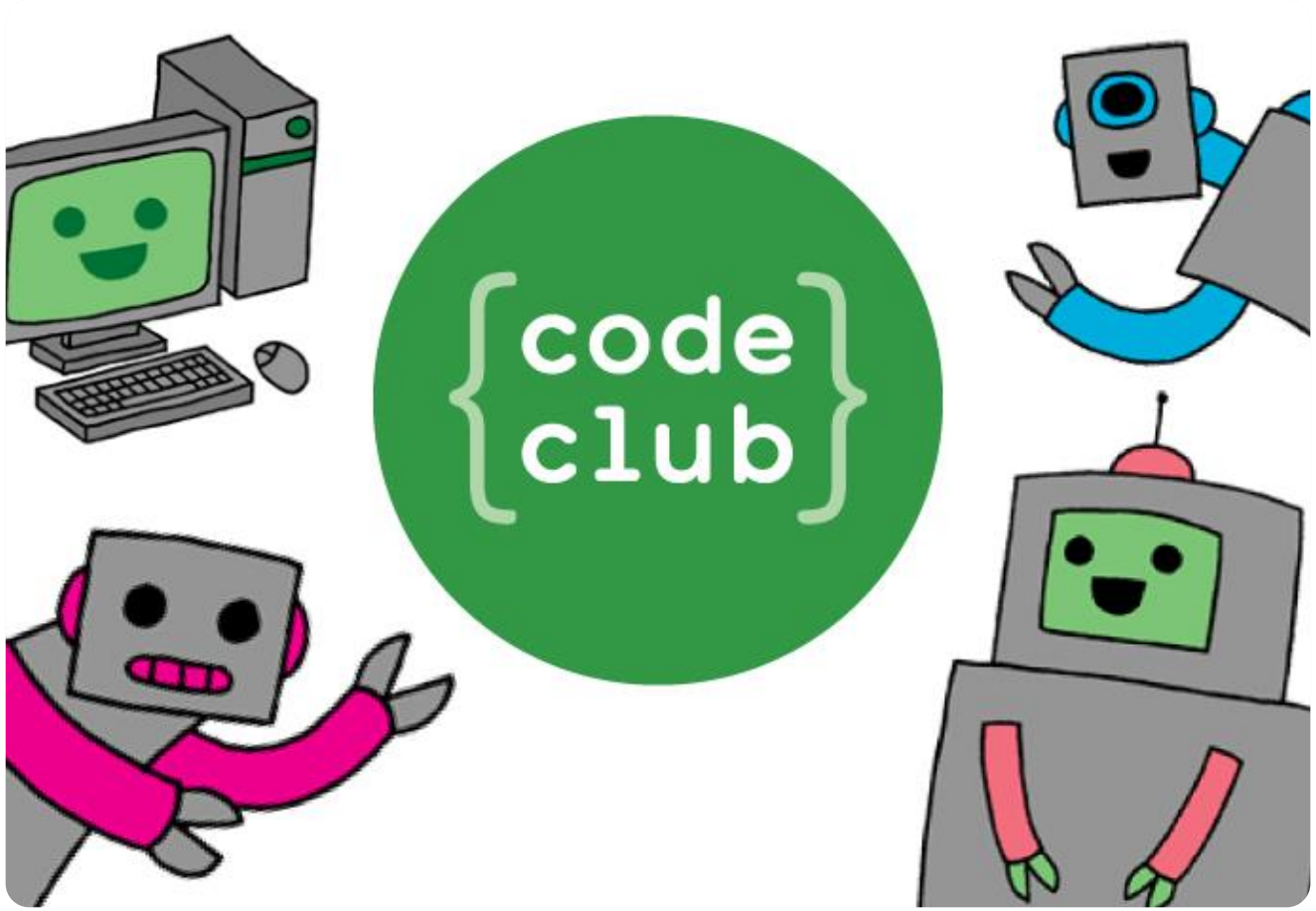
But What About...



But What About…



But What About...



The Problem - Restated

- **We want to** engage kids and teach them how to code.
- **Current solutions are not meeting this** as they:
  - are too **complicated**
  - are too **boring**
  - do not offer **enough guidance**
  - are too **simple**
  - require a lot of **organization**
  - are **hard to find**
  - are **expensive**
  - cannot be done in the kid's **own time and place**

The Problem - Restated

How might we improve

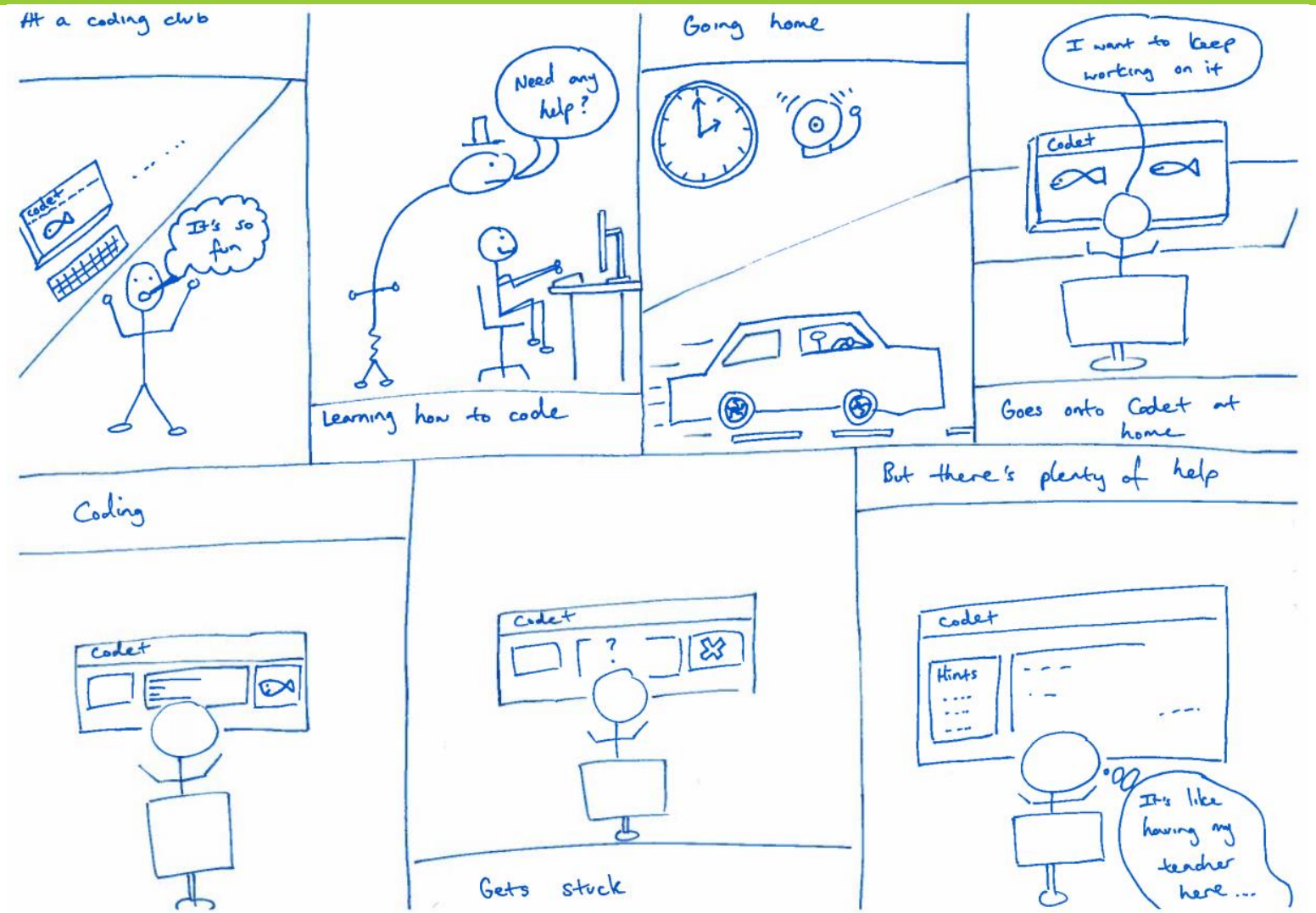
learning programming

to make it more accessible for kids?

Codet

Teaching the next generation of coding cadets

Our Proposed User Journey



Our Digital Touchpoint: Login

User Stories

- As a **student**, I want an account, so that my stats and code don't get overwritten.
- As an **instructor**, I want an admin account so that tutorials and challenges I write don't get edited by others.

User Feedback

- Users tended to click the wrong button.

Our Digital Touchpoint: Index

User Stories

- As a **student**, I want to know what to learn next and to see my progress.
- As an **instructor**, I want to tailor the learning content for my own students.

User Feedback

- Make the progression clearer.

Our Digital Touchpoint: Tutorial

User Stories

- As a **student**, I want to be guided through the process of learning a concept and be sure I understand it.
- As an **instructor**, I want to be sure my students understand the content.

User Feedback

- I want to know my progress through the tutorial.

Our Digital Touchpoint: Challenge

User Stories

- As a **student**, I want to use my knowledge and creativity to create cool games.
- As an **instructor**, I want my students to be as independent as possible, offering help when needed.

User Feedback

- Some users get stuck when doing a challenge.

Our Digital Touchpoint: Shop

User Stories

- As a **student**, I want to be rewarded for my learning.
- As an **instructor**, I want my students to be motivated to learn.

User Feedback

- "I want avatars to be like memes"

Our Digital Touchpoint: Friends

User Stories

- As a **student**, I want to share and see all my real friend's progress and avatars.

User Feedback

- I had to refresh the page to see new friend requests.

Our Digital Touchpoint: Admin

User Stories

- As an **instructor**, I want it to be easy to add new content for my students.

User Feedback

- "The admin page feels very hacky and finicky"

Our Digital Touchpoint: Classes

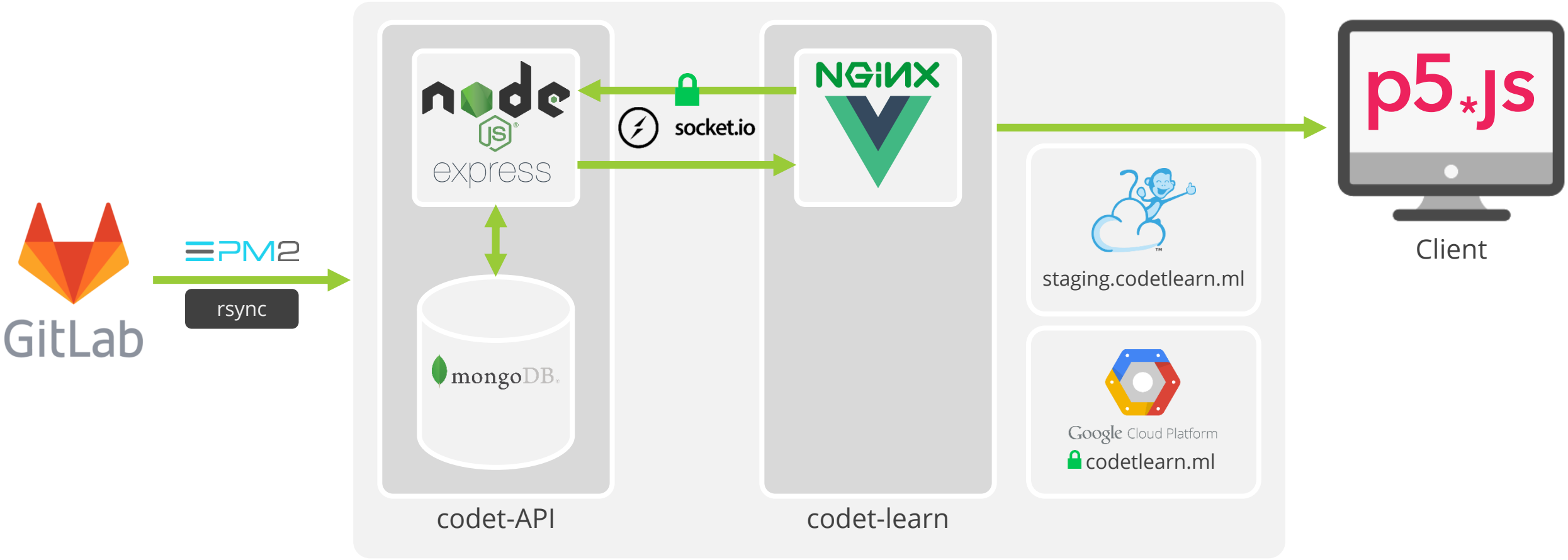
User Stories

- As a **student**, I want to be part of my coding class even when at home.
- As an **instructor**, I want to add content only for the classes I teach.

User Feedback

- It wouldn't tell me if I add the wrong usernames.

Our System Architecture



Why Vue.js?

Reactive and lightweight

- Based on reusable components
- Extremely lightweight
- More flexible than Angular
- Better separation of concerns than React

Why MEVN?

A popular, entirely JavaScript, tech stack

Node.js

- Async I/O
- Packages through npm
- Uniform data serialization

Express.js

- Widely used
- Extensible with middleware

Why MongoDB?

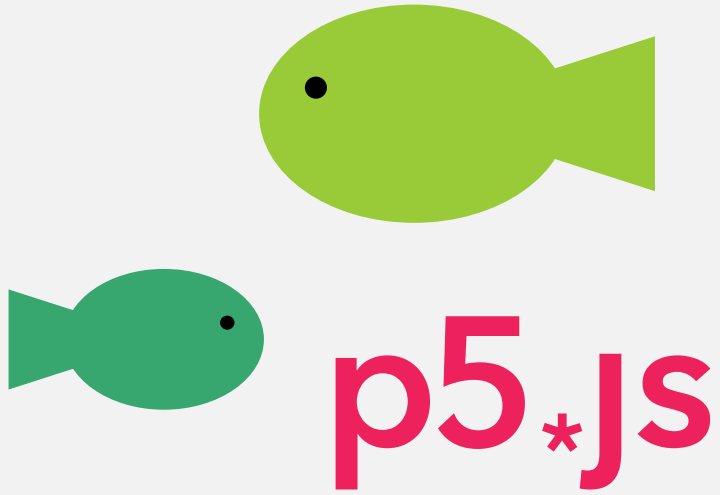
So that we can achieve really fast iterations...

- Natural front-end correspondence
- Handles complex data structures
- Flexible schema
- Fast and horizontally scalable

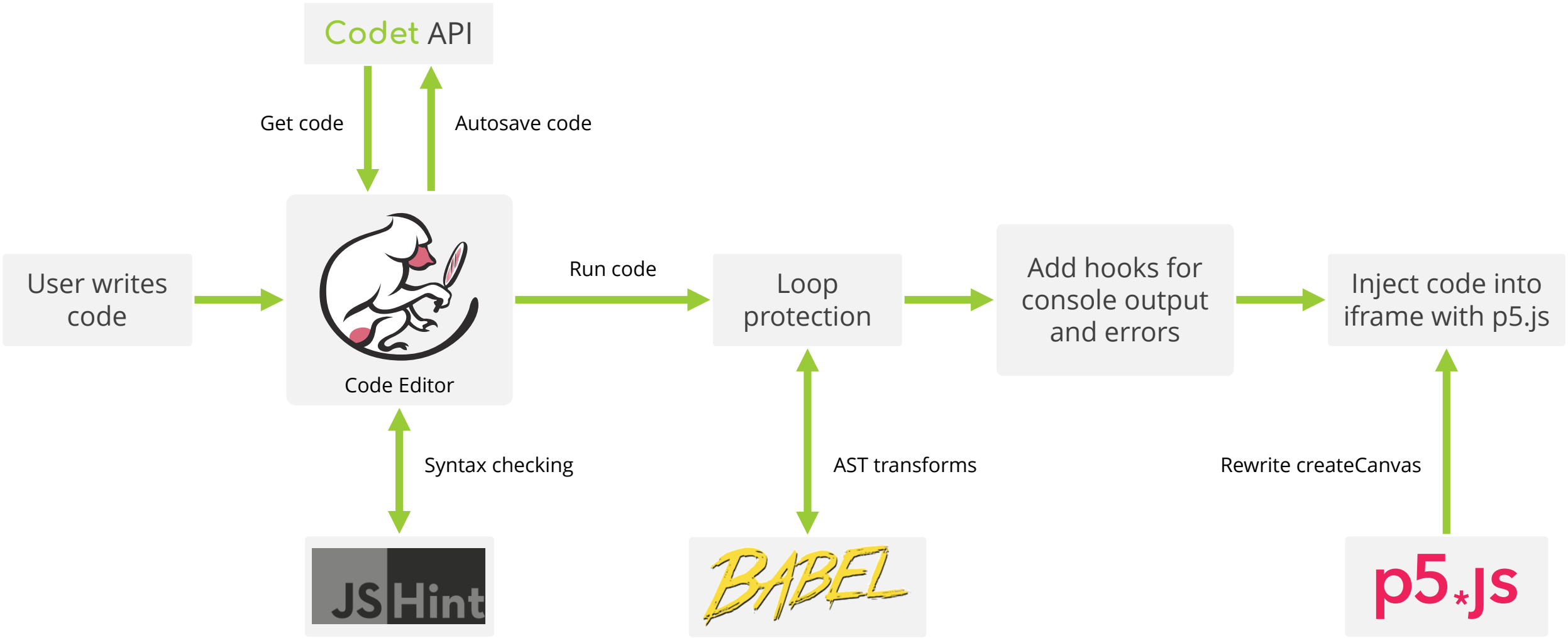
Why p5.js?

Perfect for teaching kids

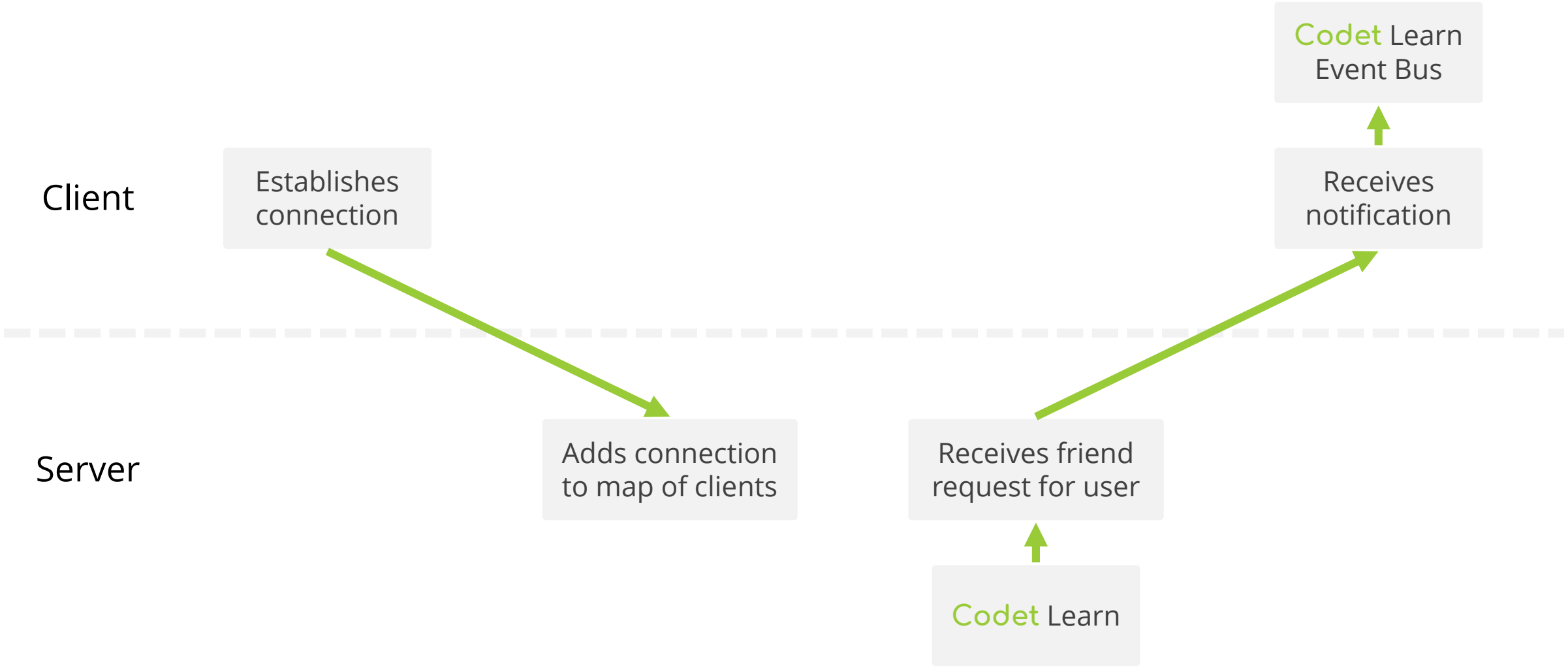
- Based on JavaScript
- Fewer difficult concepts
- Runs nicely client-side
- Visual feedback
- Easy to make animations / games



Challenges: Reporting Errors



Challenges: Notifications



Project Summary

Our project has been successful at solving many of the problems we identified

Tutorials solve:
- Too complicated
- Not offering enough guidance

Digital touchpoint solves:
- Requires a lot of organization
- Difficult to find
- Expensive
- Cannot be done in the kid's own time and place

Challenges solve:
- Too simple

Codettos / Incentives solve:
- Too boring

What we've learned...

Design and Organisation

- Speaking to a few users often is better than speaking to many users rarely
- How to prototype rapidly with users
- How to prioritise features

Technical

- Use a CSS framework
- Use a store instead of an event bus

What's next for Codet?

- Redesigning the index page
- Achievements
- Group projects
- Saving history
- Sharing tutorials

Anticipated Question:
User Interaction

Friends

- Sending, accepting friend requests, and unfriending
- Sharing projects between friends

Teaching

- Teachers can add content for students to attempt
- Students asking for help, marking as resolved
- Instructors viewing student code

Leaderboard

- Leaderboard: can see users' avatars and XP