

Why Onboarding to a Company's Legacy Codebase Sucks and How to Make it Work for Your Team

Shanea Leven, CodeSee.io





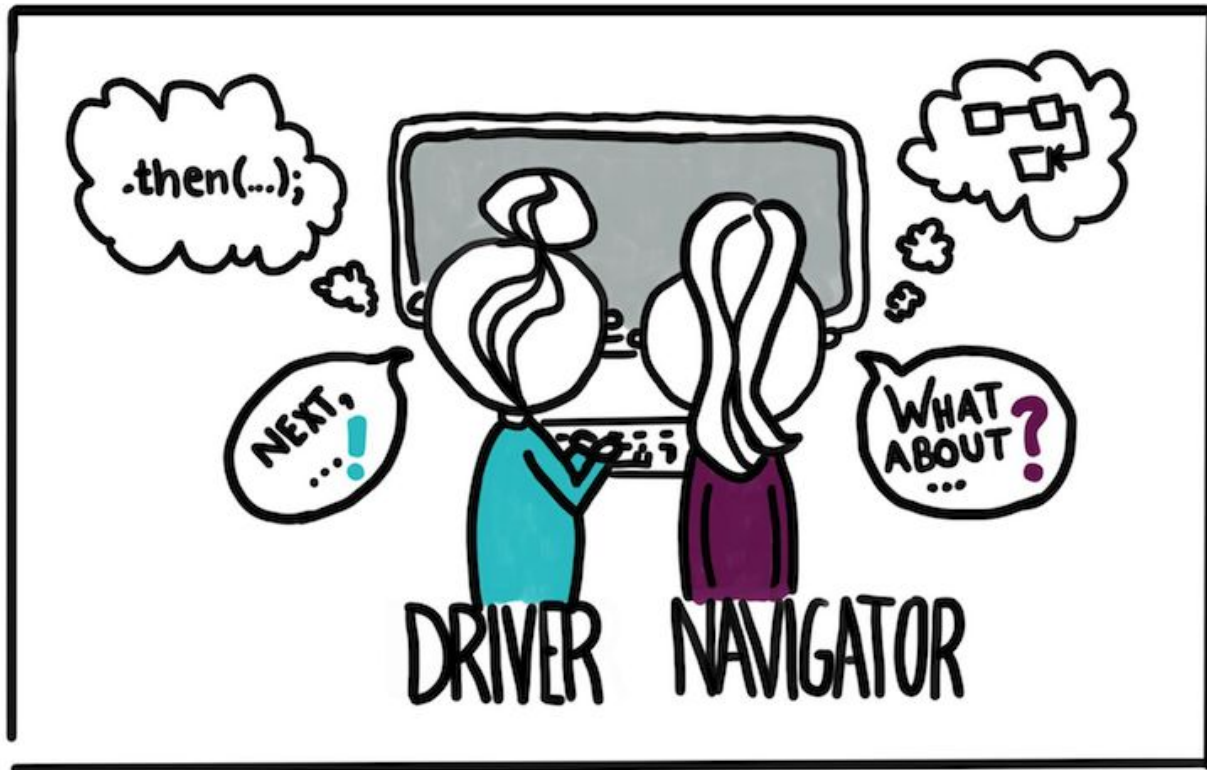
You must learn how the codebase works first
before you can modify it.



**Today i'm happy
Because**

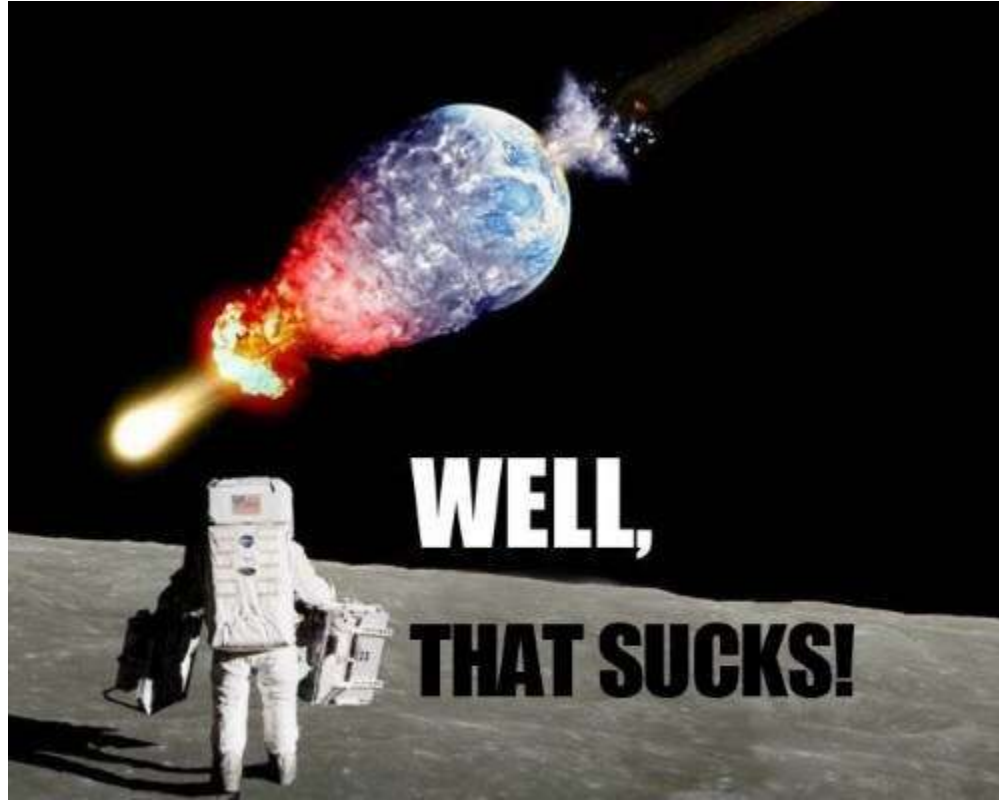
Shh! It's a

TOP SECRET



The secret is to pair... all of the time.

Shanea



The pasta theory of programming



Spaghetti code

Unstructured and hard-to-maintain code caused by lack of style rules or volatile requirements. This architecture resembles a tangled pile of spaghetti in a bowl.



Lasagna code

Source code with overlapping layers, like the stacked design of lasagna. This code structure makes it difficult to change one layer without affecting others.



Ravioli code

Isolated bits of code that resemble ravioli. These are easy to understand individually but—taken as a group—add to the app's call stack and complexity.



Pizza code

A codebase with interconnected classes or functions with unclear roles of responsibilities. These choices result in a flat architecture, like toppings on a pizza.

Everyone has their own plate of spaghetti code

3 major parts

Code/System

Working with
code/team

Being a human

Code/System

- How the actual code is connected?
- What are the execution paths?
- What functions call other functions?
- What is the high level architecture?
- What are all of the domain objects?



Working with the code/team

- Code review process
- Design patterns
- Testing patterns
- Gotchas in the code
- Deployment procedures



The reality of being human

- Ship code don't write
- Memory retention
- Hard to learn
- Big systems



The reality of being human

- Ship code don't write
- Memory retention
- Hard to learn
- Big systems





Mostly everything gets
worse before it gets better.

Wanda Sykes

“ quote fancy

Onboarding is continuous

Constant...never ending

New features

Systems go down

People leave

Restructuring

Refactoring

Onboarding is continuous

Constant...never ending

New features

Systems go down

People leave

Restructuring

Refactoring

Onboarding is continuous

Constant...never ending

New features

Systems go down

People leave

Restructuring

Refactoring

**WHAT
GIVES?**

3 types of advice

For the team

For the
onboardee

Continuous
onboarding

Advice for the team

- Start small in scope
- Expand strategically
- Choose tasks that build confidence

Advice for the team

- Bug fixes
- Small tasks the team knows already
- Isolated features
- Tech debt



Advice for the team

- Bug fixes
- Small tasks the team knows already
- Isolated features
- Tech debt



Advice for the onboardee

- Domain objects
- User examples
- Key flows
- Architecture
- Frameworks
- Coding Patterns
- Development process
- Tribal Knowledge

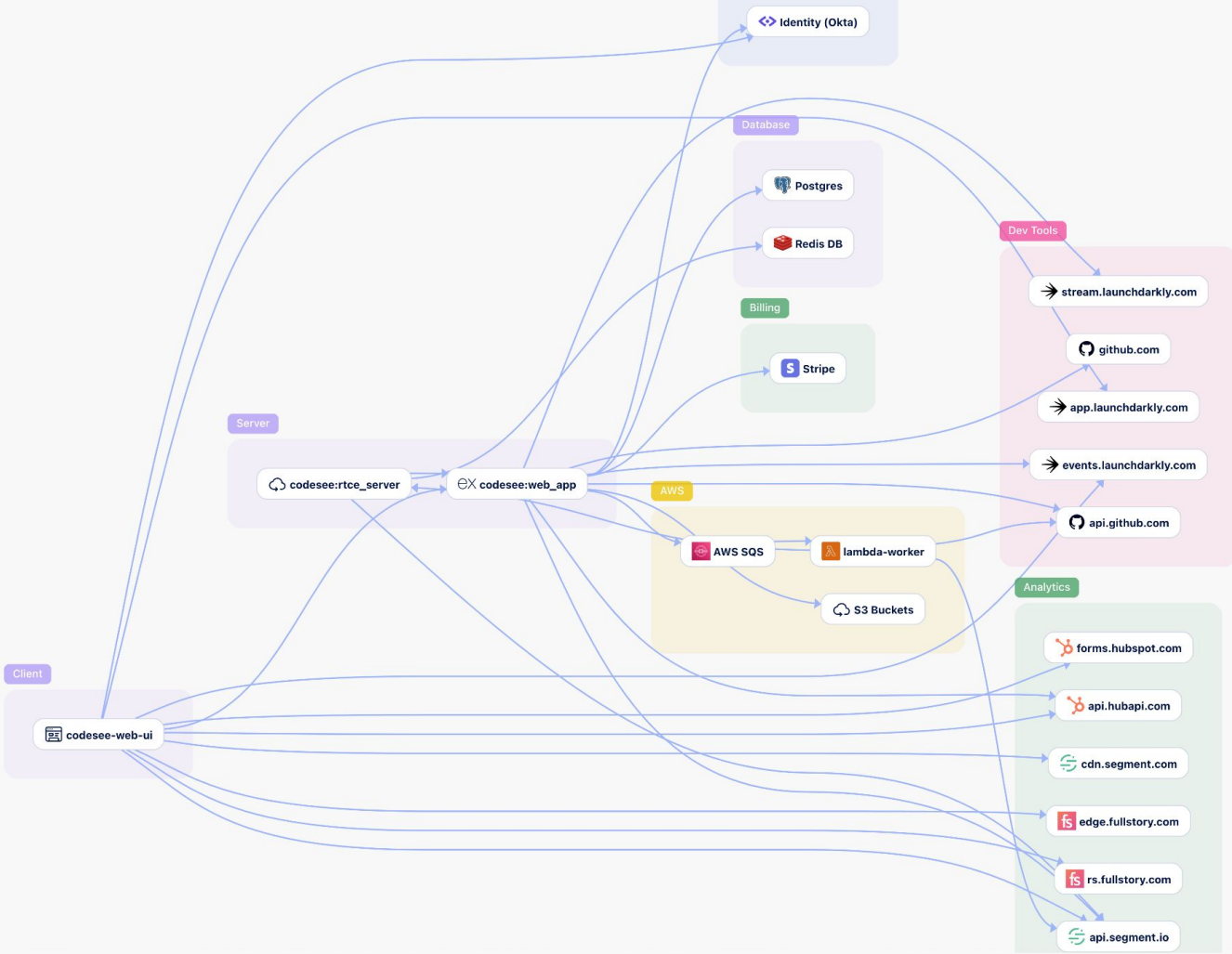
Advice for the onboardee

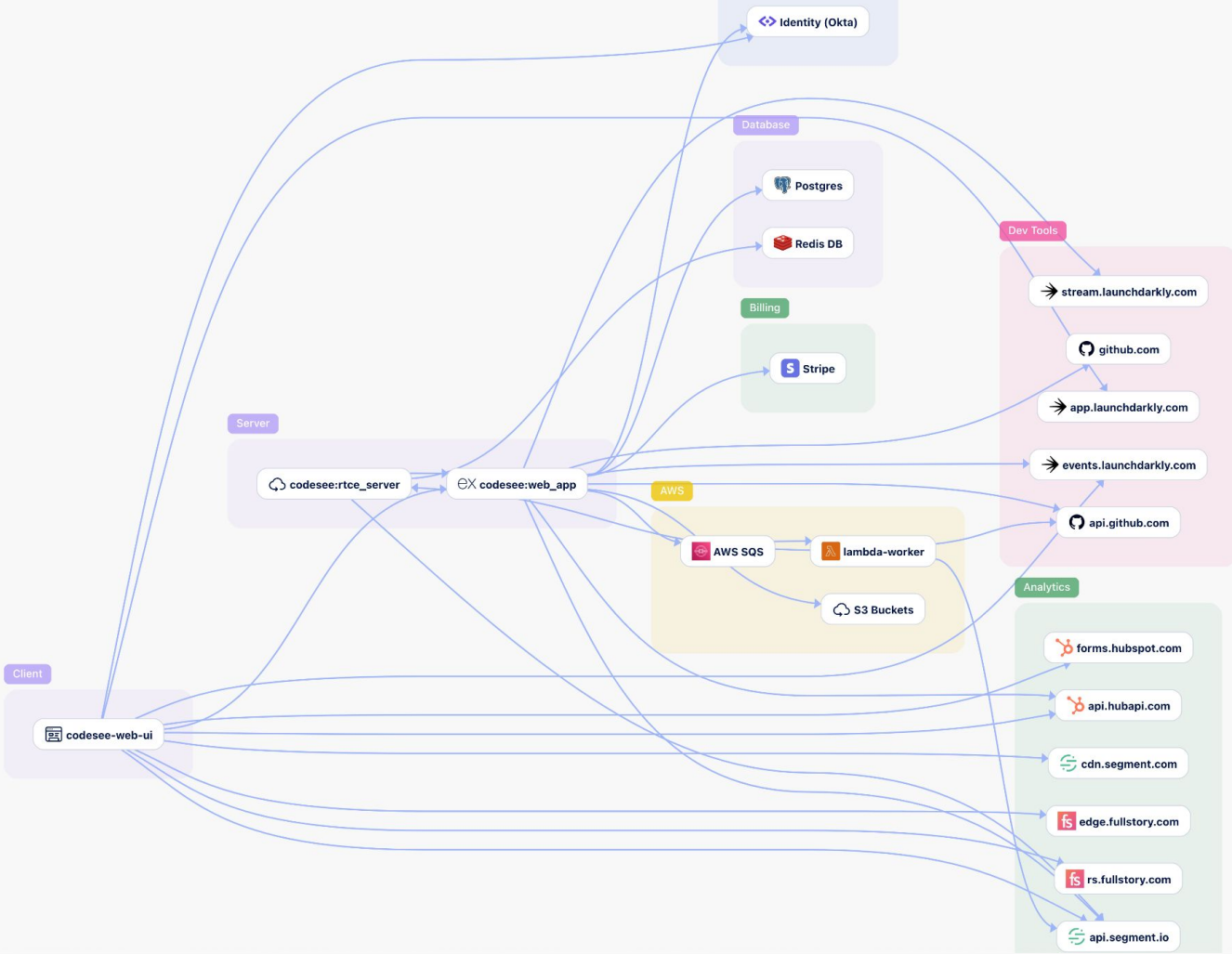
Domain objects, User examples, Key flows

Thing to know	Dev tools	Ed Tech	Ecomm
Domain	codebase, dependency, commit, pull request	assignment, activity, lesson, grade	product, shopping cart, shop, price, discount
User	developer, team lead	teacher, student	customer, seller
Key flows	developer creates a new commit, system analyzes the codebase	teacher creates an assignment, student answers a question	customer makes a purchase, seller lists a product

Advice for the onboardee

- Ask!
- Explore the marketing site
- Read the docs
- Write stuff down





Advice for the onboardee

- Frameworks
- Coding patterns
- Development process
- Tribal knowledge
- Write stuff down

Advice for the onboardee

- Frameworks
- Coding patterns
- Development process
- Tribal knowledge
- Write stuff down

Advice for the onboardee

- Frameworks
- Coding patterns
- Development process
- Tribal knowledge
- Write stuff down

Advice for the team/onboarders

- Domain objects
- User examples
- Key flows
- Architecture
- Frameworks
- Coding Patterns
- Development process
- Tribal Knowledge



You need better tools

- Separate tools
- Manual key flows
- Same patterns
- Code Reviews

Environment production

Show services active:

Past 15 days

POST /api/maps/dependency-map/new



Variant 1 of 701



Share

Flow details

Name

New Dependency Map

Entry point

POST /api/maps/dependency-map/new

Description

When the code analysis is complete, it is uploaded to this endpoint. The endpoint saves the analysis, then triggers lambdas to update existing maps and/or create a review map and post it on github.

Variants

Complete Review Map flow

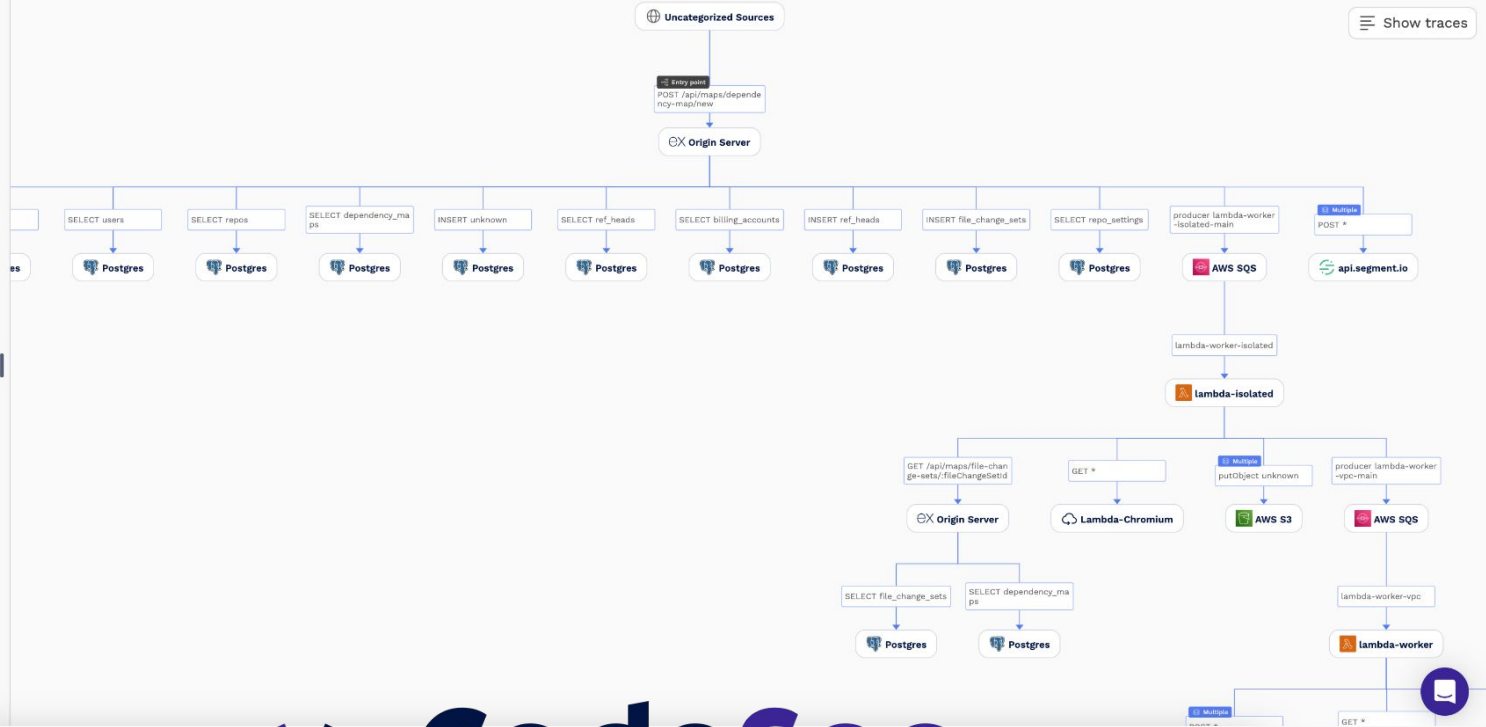
2% of flow traffic

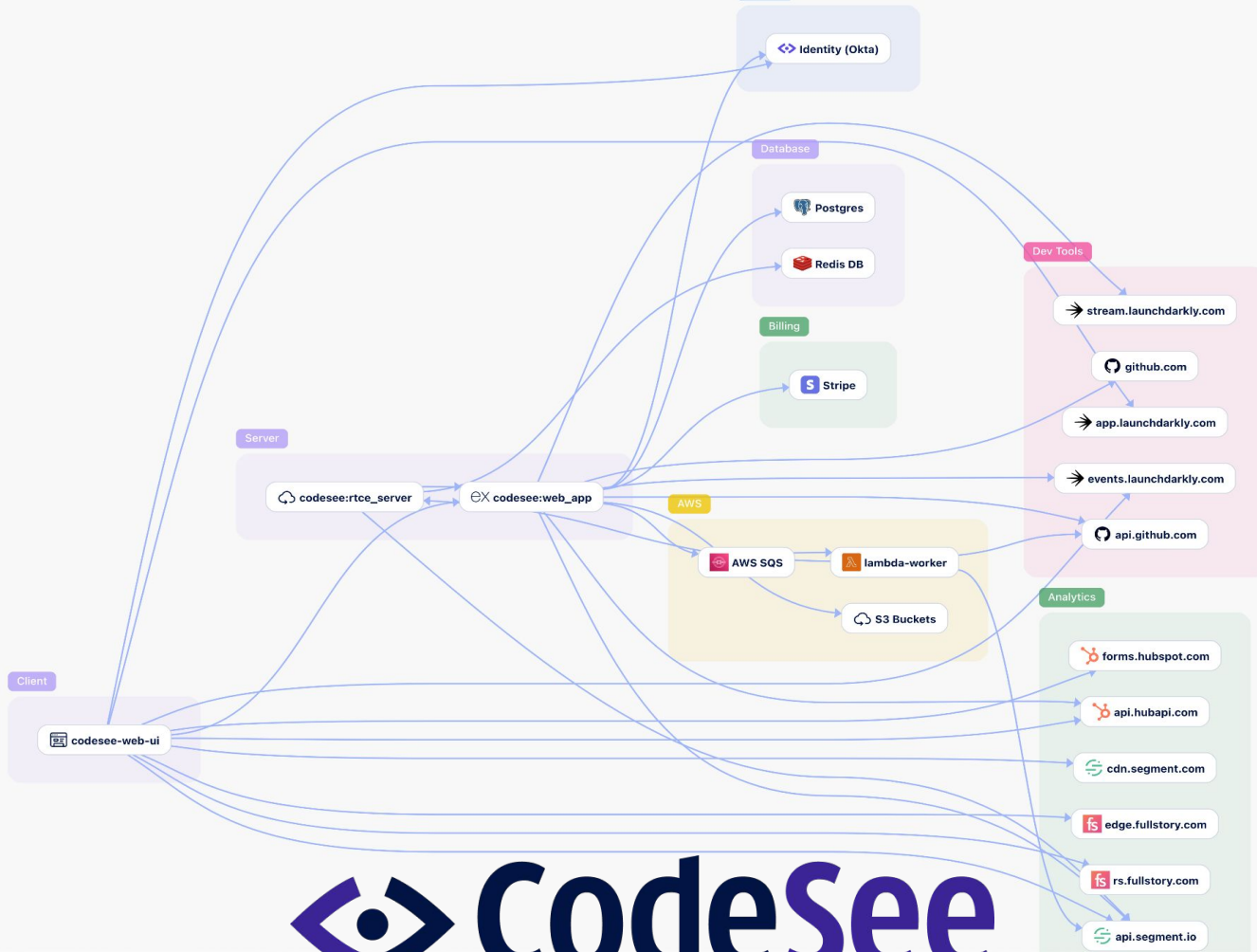
Description

Add a description

Variant 2

9% of flow traffic





The background of the slide is a light blue map with a grid of streets. A red line with circular markers at its vertices is drawn across the map, starting from the bottom left and moving towards the top right. A red location pin icon is placed on the map, slightly to the right of the center. The text is contained within a dark blue rounded rectangle on the left side of the slide.

Thank You!

CodeSee.io

@shanealeven or @codeseeyio