

LLD - Tic Tac Toe

① Design

② Improve Our design

③ Implement the system
- Models

④ Improve Our code

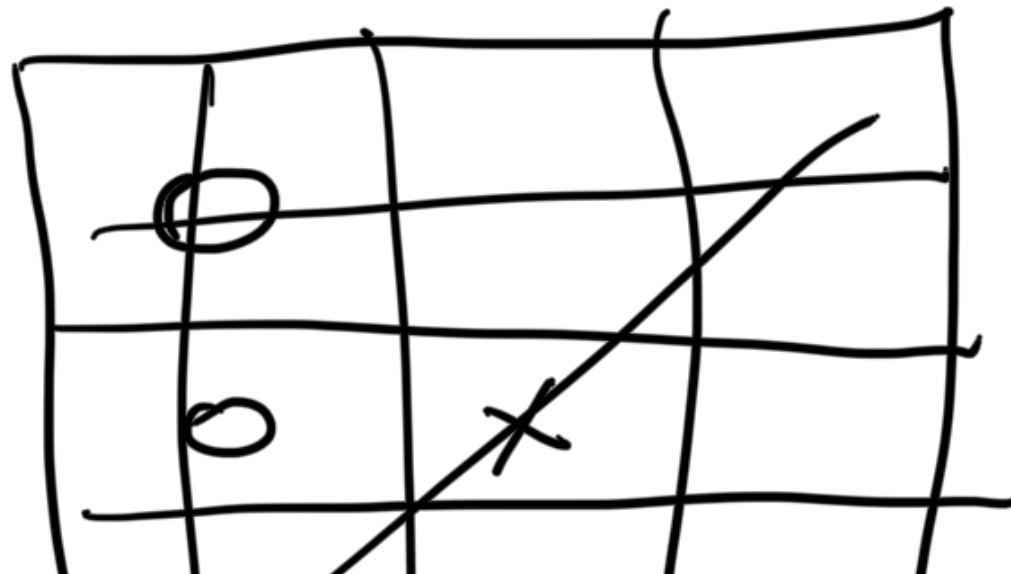
Assignment

- ① Class Diagram
 - ② Code assignment
-

Tic Tac Toe

→ Create a game called tictactoe

○



3x3



→ Requirements

→ Size of the board

→ Roles of the game

→ No of players

→ Type of players ✓

① Design TTT with a board of
size $n \times n$ (3×3)

② Played by 2 players

③ Turn by turn + 1st player
is randomly chosen

④

The game will go on till
one player wins (n conse. Symbols)
→ when all the cells are filled

	O	O	.
	O	O	.
	X	X	.

⑤

Each player will choose a
Symbol. O, X

⑥

2 types of players

— Human

— Bot

H v H }
H v B } ~~B v B~~

⑦

A bot will have a diff. level

— Hard

— medium

— Easy

⑧

Human → email
→ username

↗

→ profile photo

⑨ Who will use the system →

① CLI

② API



⑩ A player can play multiple games
git vs github

git status

A hand-drawn diagram where a bracket is drawn under the text 'git status'. An arrow points from the right end of this bracket to a rectangular box on the right side of the page.

API



Command

FE

BE

network

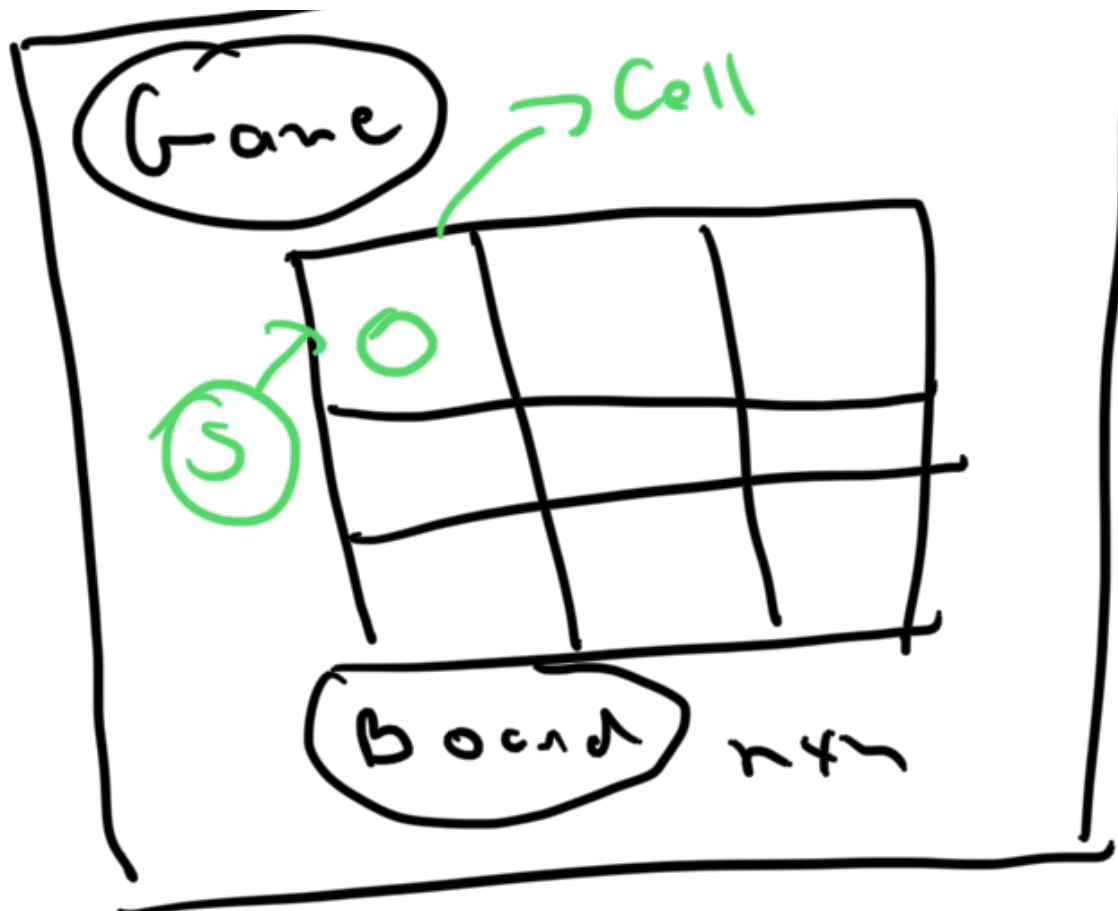


Entities

— nouns

— visualisation

Human



Keep it simple
(KISS)



register

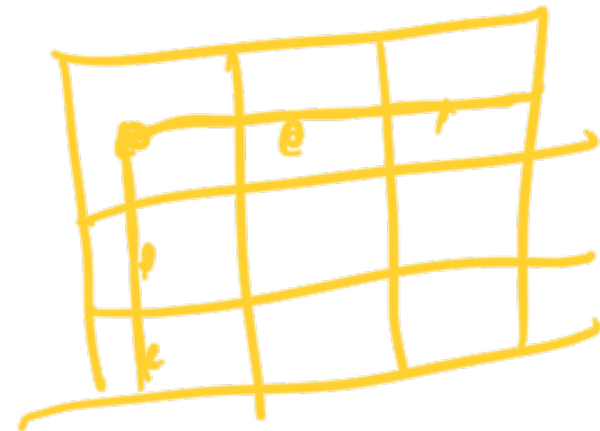
Create gone

no more

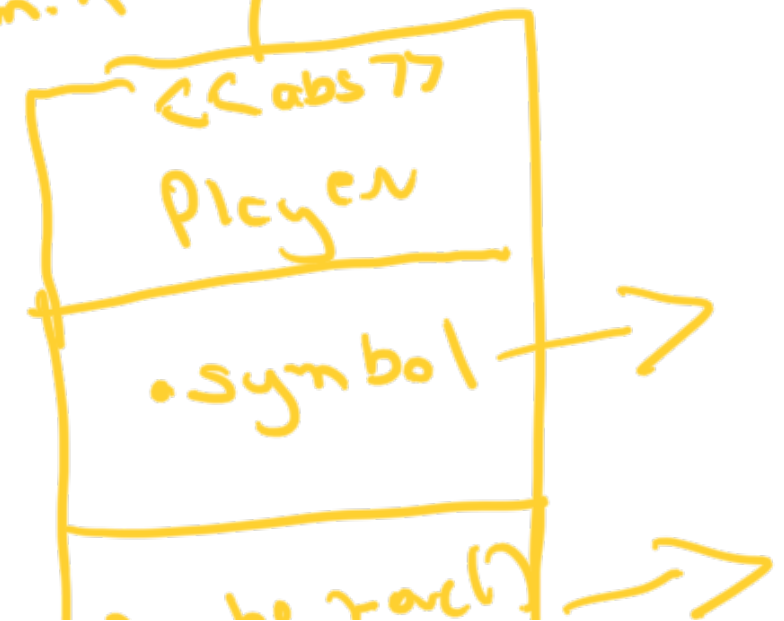


make it

→ Class Diagram

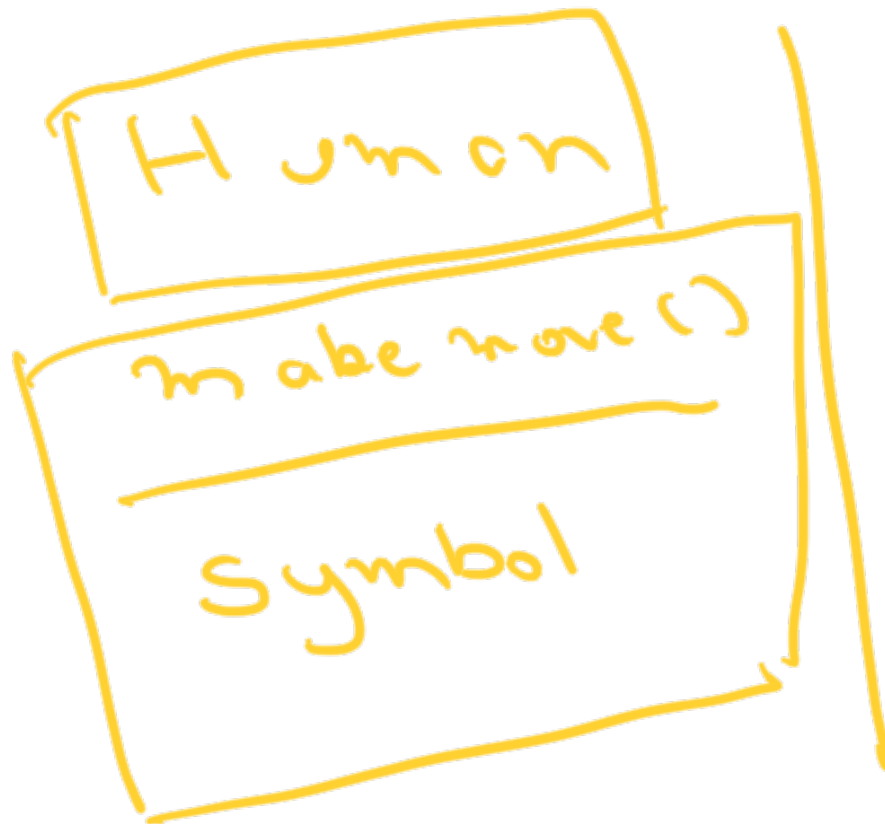


①





DDoS



userId name

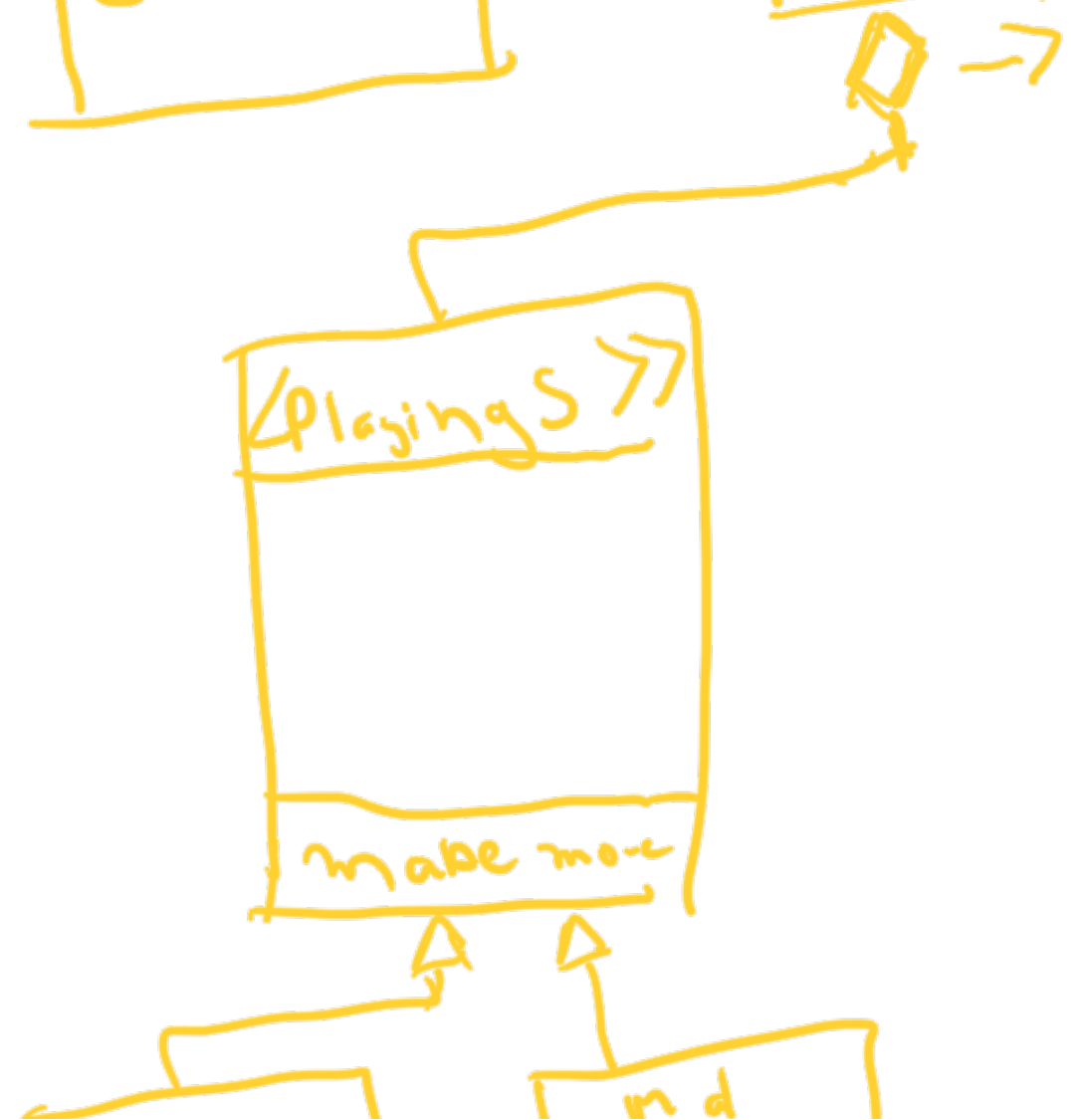


photo
email



①

common behaviour

no → separate classes

yes → interface or abs. class

②

common attributes

yes → abstract

no → interface

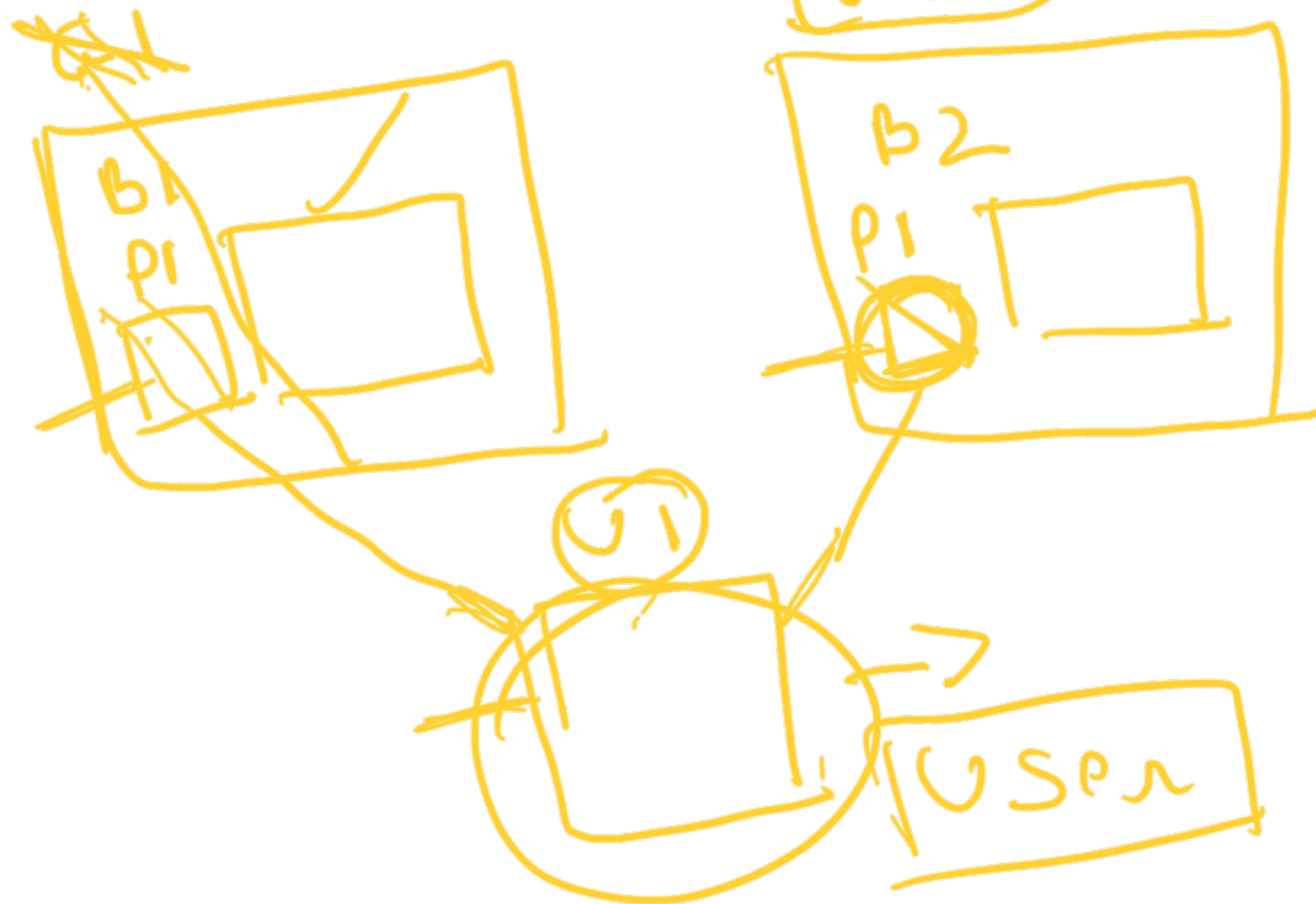
→ common behaviour

→ common attribute - symbol

int. vs abs

→ int. cannot have attributes

→ static
→ default





6:07 }
6:12 } 22:43

→ SRP

↓

A hand-drawn rectangular box at the bottom right of the page, with a downward arrow pointing towards it from the text '↓'.

Human Player

userId

listener of

[Code duplication]

Special

→ Easy + Medium

make more()

..... Easy ← }
..... Medium ← } []

}

[Strategy] = Easy, Medium

make more()

{ ↓ delegating
easy-play + medium-play

}

reduces instead of duplicating

New game

Create home

→ List < Player >

→ < board >

Home

System

~~extensible~~

~~maintable~~



3 Tiered architecture

Restu →

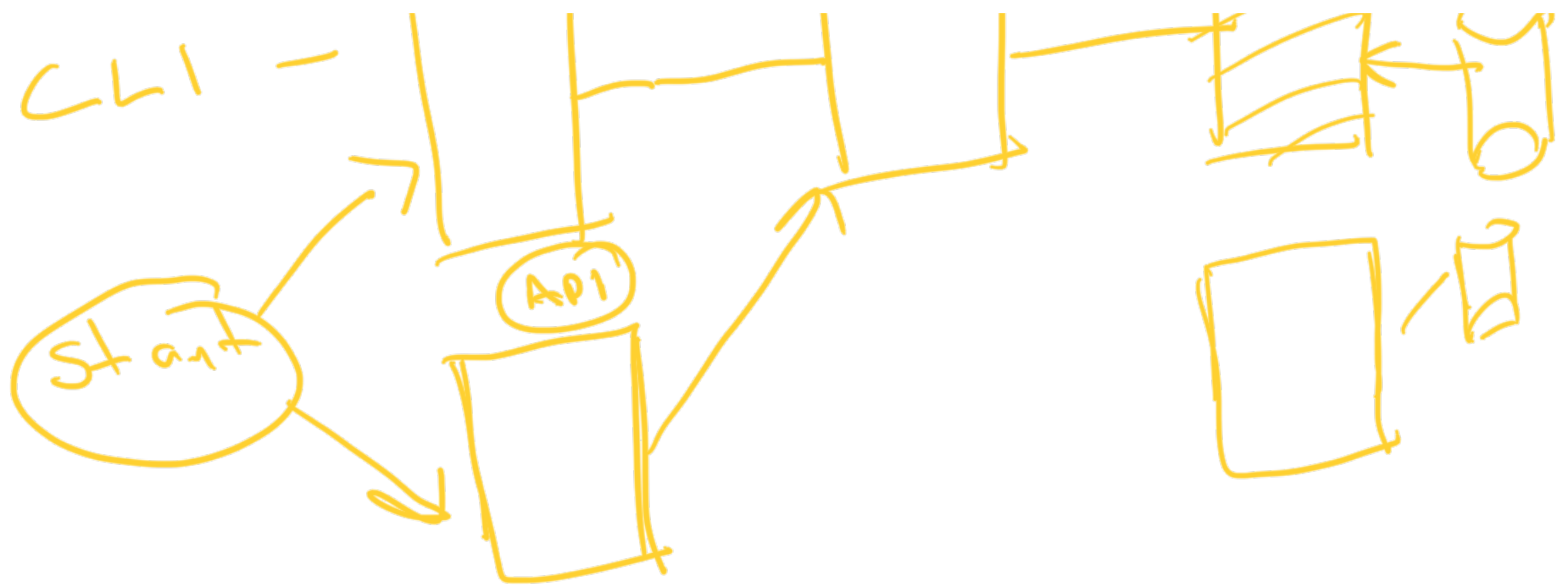


Controller



- Controller
- Service
- Repository
+ Models





→ Models