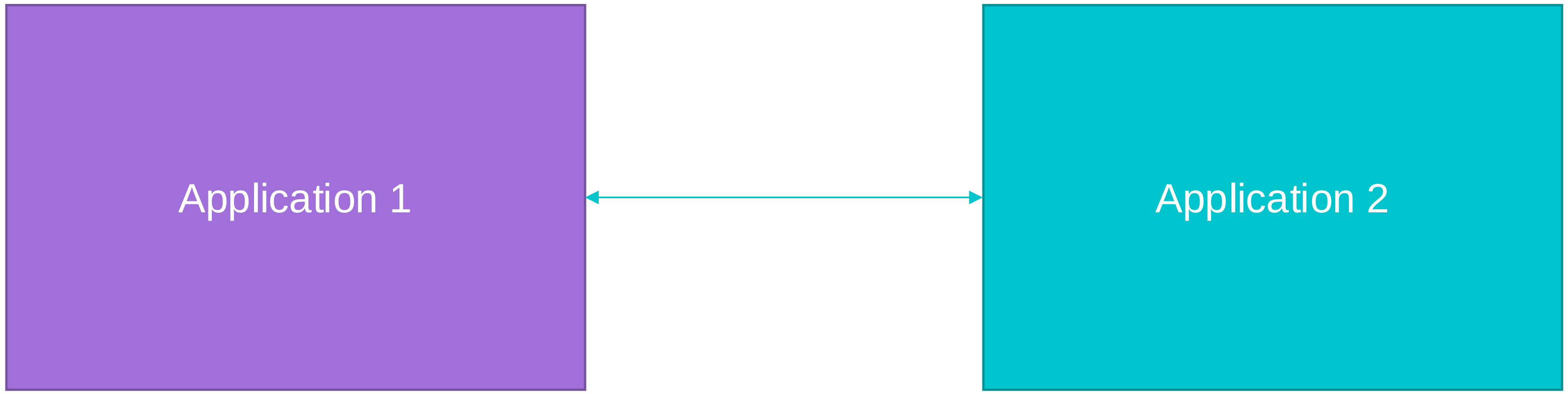
Getting Started with Designing APIs

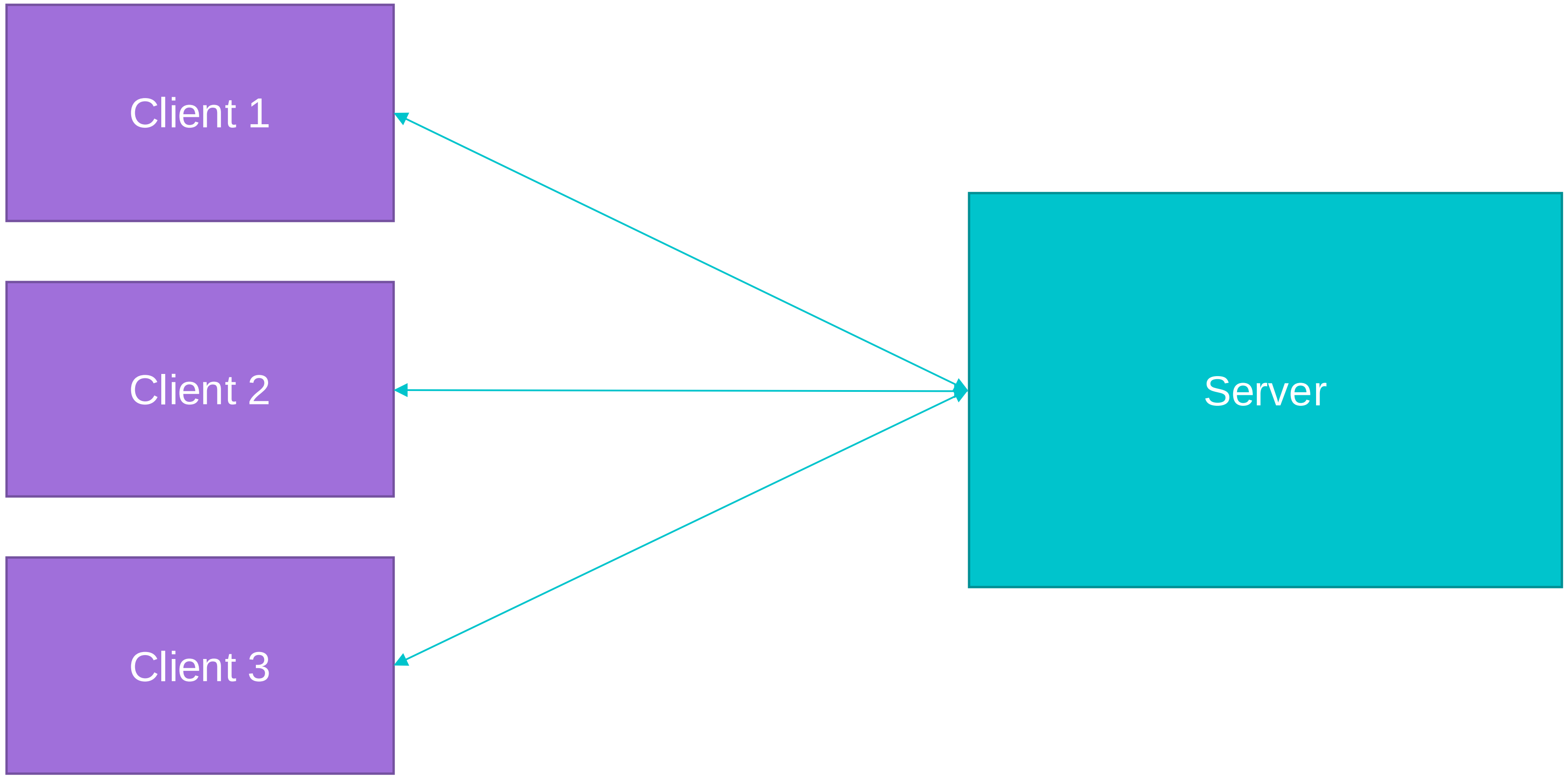
Introduction

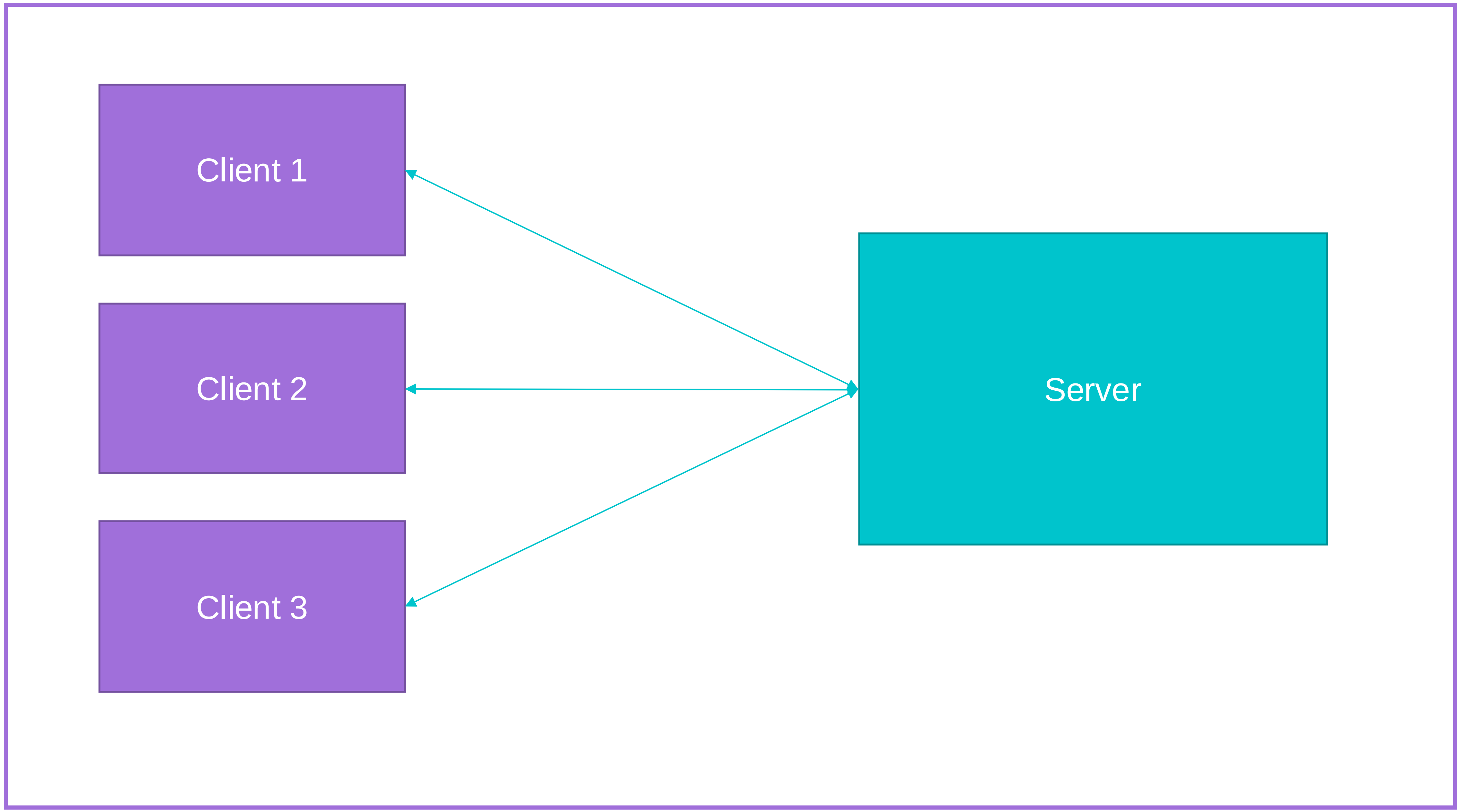
What is API?

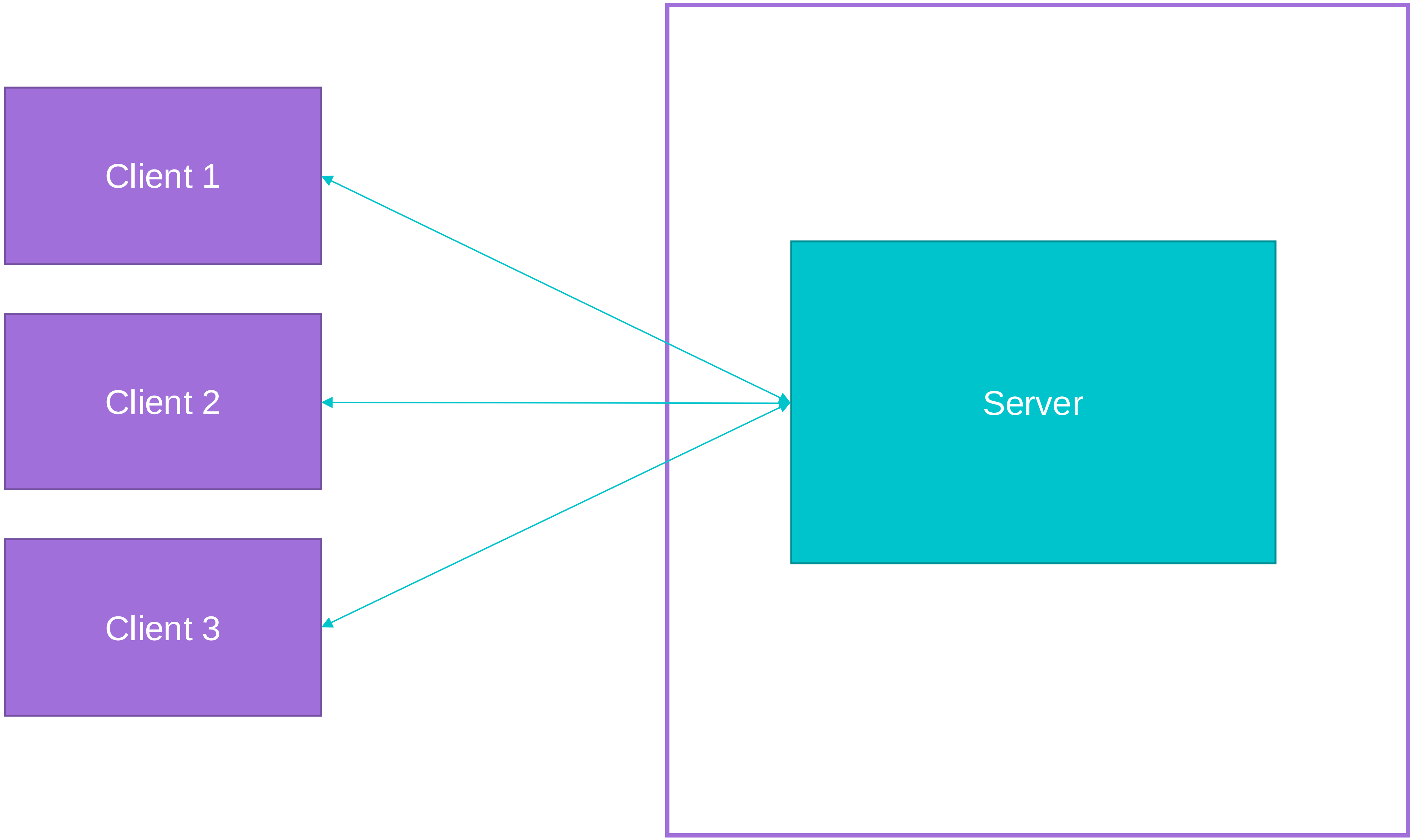
What is API?

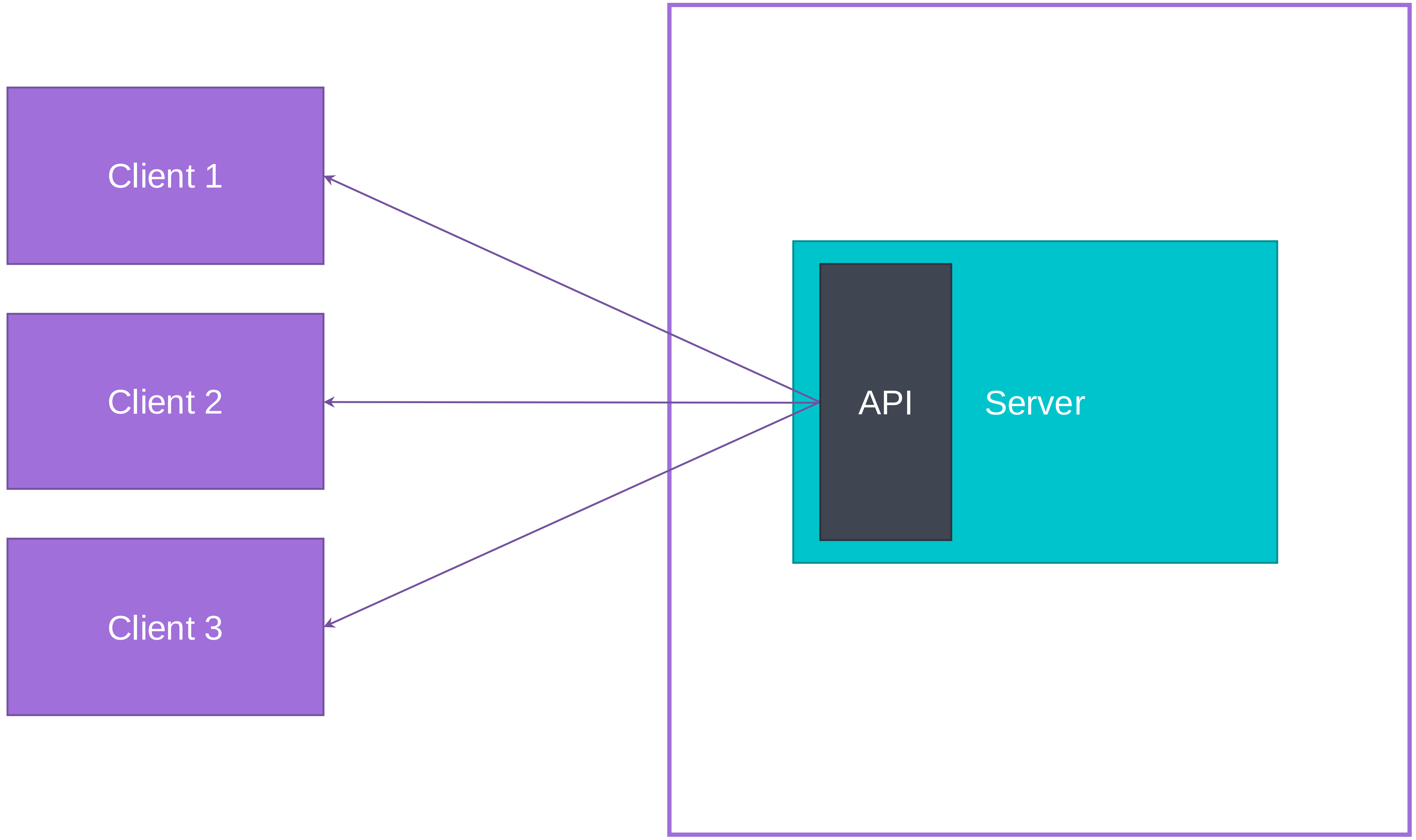
Application Programming Interface

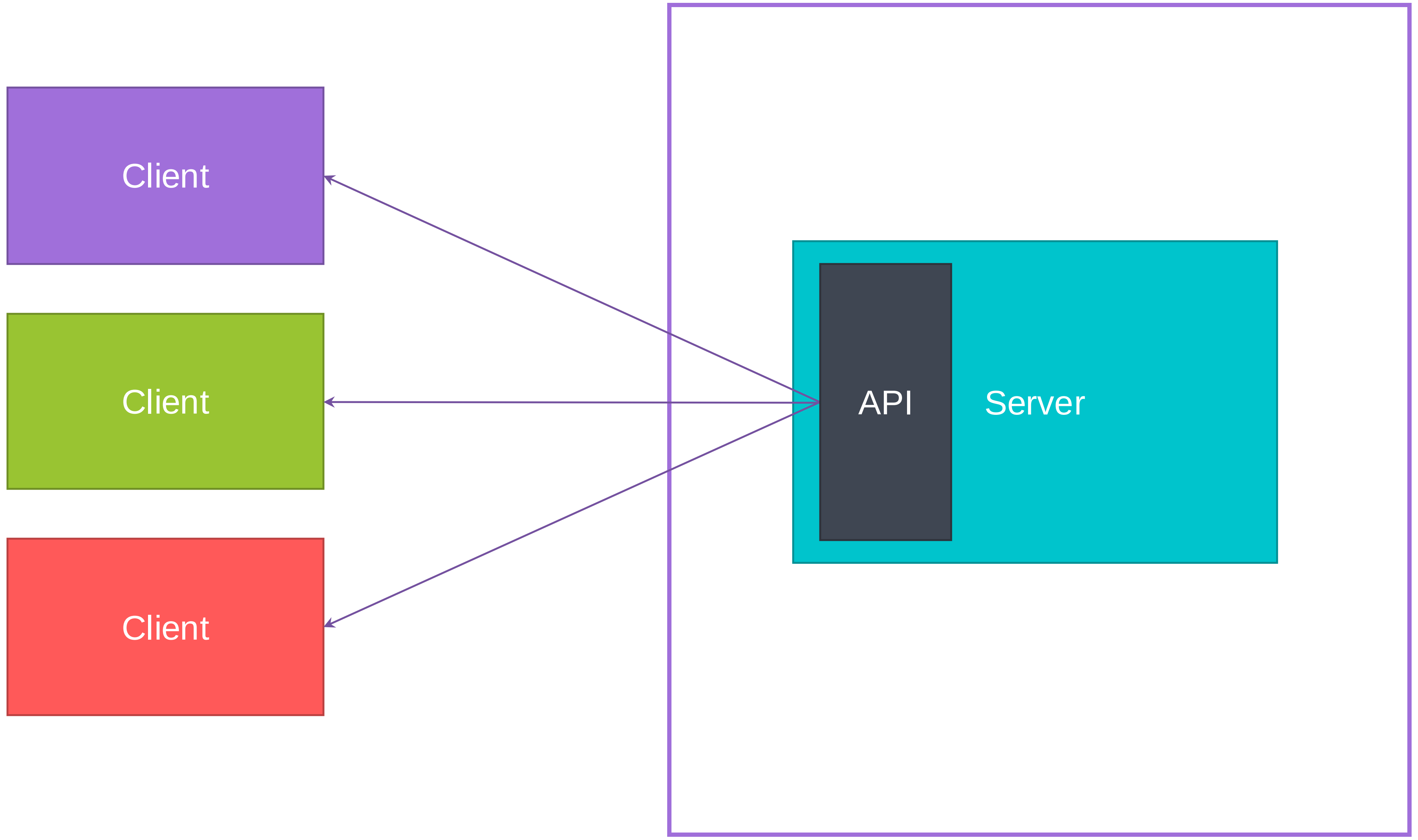

Application 1
Application 2

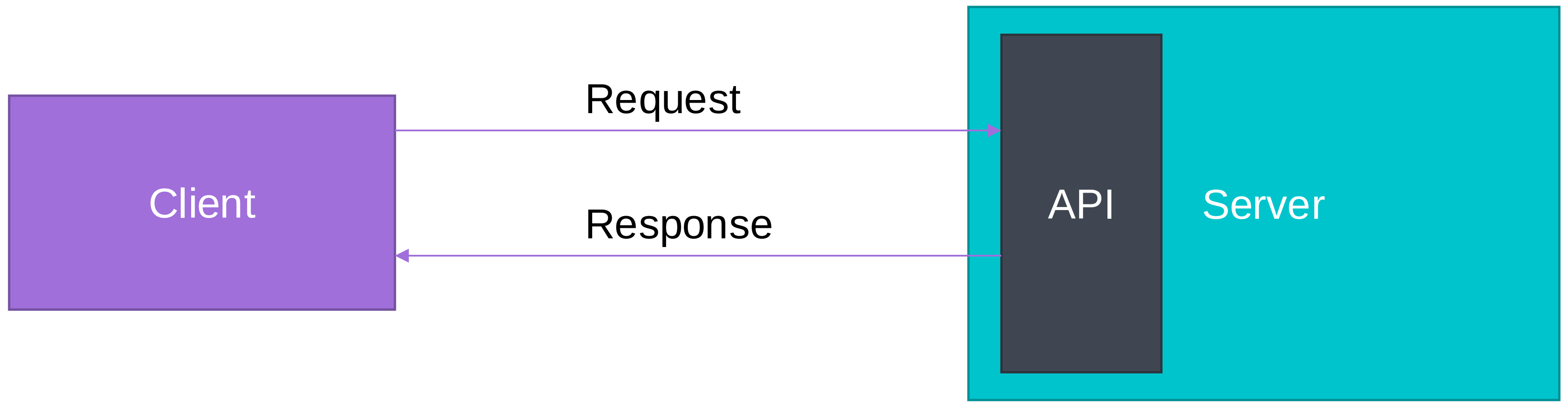

Client 1
Client 2
Client 3
Server


Client 1
Client 2
Client 3
Server


Client 1
Client 2
Client 3
Server


Client 1
Client 2
Client 3
API
Server


Client
Client
Client
API
Server


Request
Response
Client
API
Server

Why Should You Design API?

Cons of Not Designing APIs

- Efforts to change API are lot more.
- Adds to complexity of source code changes.
- Leads to more versioning.

Pros of Designing APIs

- Easy to manage.
- Team follows a certain standard.
- Increases the productivity of the team.
- Become professional.

Mandatory to design?

Mandatory to design?

Strongly recommended to design before starting implementation

STEP 1: Create a New API

API for College Management System (CMS)

API for College Management System

College

Course

Student

Course Subject

Types of APIs

Types of APIs

Public API
(or Open API)

Partner API

Private API
(or Internal API)

Public APIs

- Available in the internet for consumption by any company.
- Require some sort of authentication.
- Also referred to as Open APIs.
- Proper documentation needed.

Partner APIs

- Used for integration between two organizations.
- Require some sort of authentication.
- Proper documentation needed.

Private APIs

- Used within a product in an organization.
- Communicate across different components.
- Also called as Internal APIs.
- Documentation generally not done, although it is recommended to have basic details.

Types of APIs

Public (Exposed in the internet for public)

Partner (Integration with another vendor)

Private (Modules of the same product)

STEP 2: Identify the Type of API

Should You Change the Existing API Design?

Should You Change the Existing API Design?

- Best approach is to follow existing design.
- You can deviate from the ideal API design principles.

Example: API to Manage Courses

- Existing
  - /Course
- Ideal
  - /courses

Example: API to Manage Courses

- Existing
  - /Course
- Ideal
  - /courses
- During bug fix
  - /courses

Example: API to Manage Courses

- Existing
  - /Course
- Ideal
  - /courses
- During bug fix
  - /courses

Example: API to Manage Courses

- Existing
  - /Course
- Ideal
  - /courses
- During bug fix
  - /Course

Should You Change the Existing API Design?

- Best approach is to follow existing design.
- You can deviate from the ideal API design principles.
- Gradually apply new design during enhancements.

Getting Started with Designing APIs

Summary