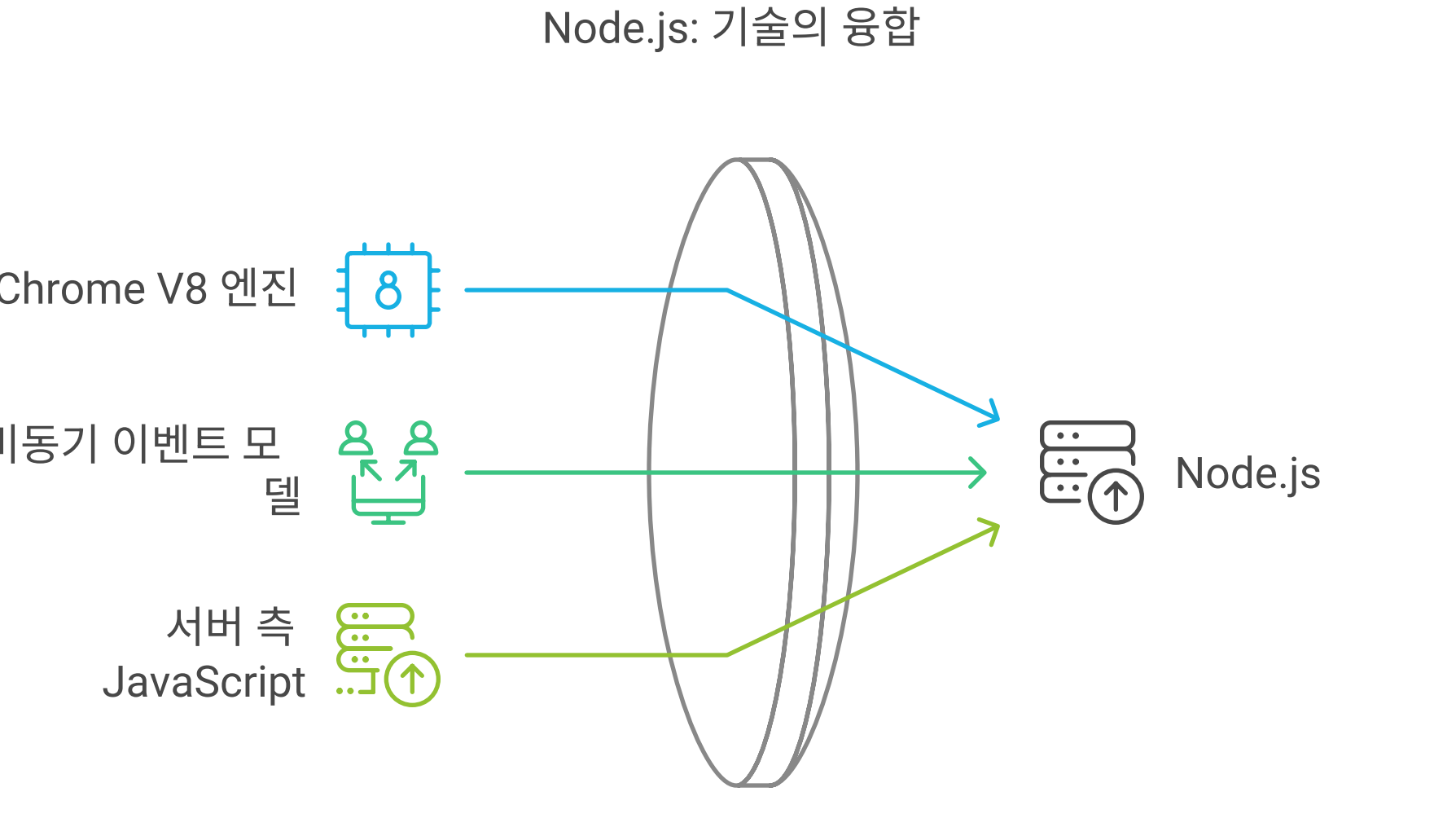


# Node.js 자료 및 구조

이 문서는 Node.js에 대한 기본적인 정보와 구조를 담고 있습니다. Node.js는 서버 사이드 애플리케이션을 구축하기 위해 널리 사용되는 JavaScript 런타임입니다. 이 문서에서는 Node.js의 특징, 구조, 주요 모듈 및 사용 사례에 대해 설명합니다.

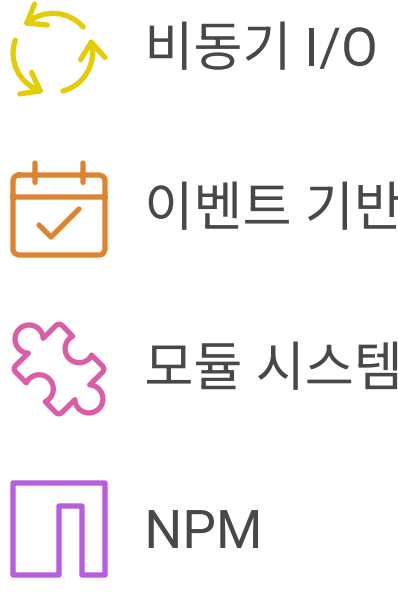
## 1. Node.js란?

Node.js는 Chrome V8 JavaScript 엔진을 기반으로 한 서버 사이드 플랫폼입니다. 비동기 이벤트 기반 모델을 사용하여 높은 성능과 확장성을 제공합니다. Node.js는 JavaScript를 사용하여 서버 측 코드를 작성할 수 있게 해주며, 이를 통해 개발자들은 클라이언트와 서버 모두에서 동일한 언어를 사용할 수 있습니다.



## 2. Node.js의 특징

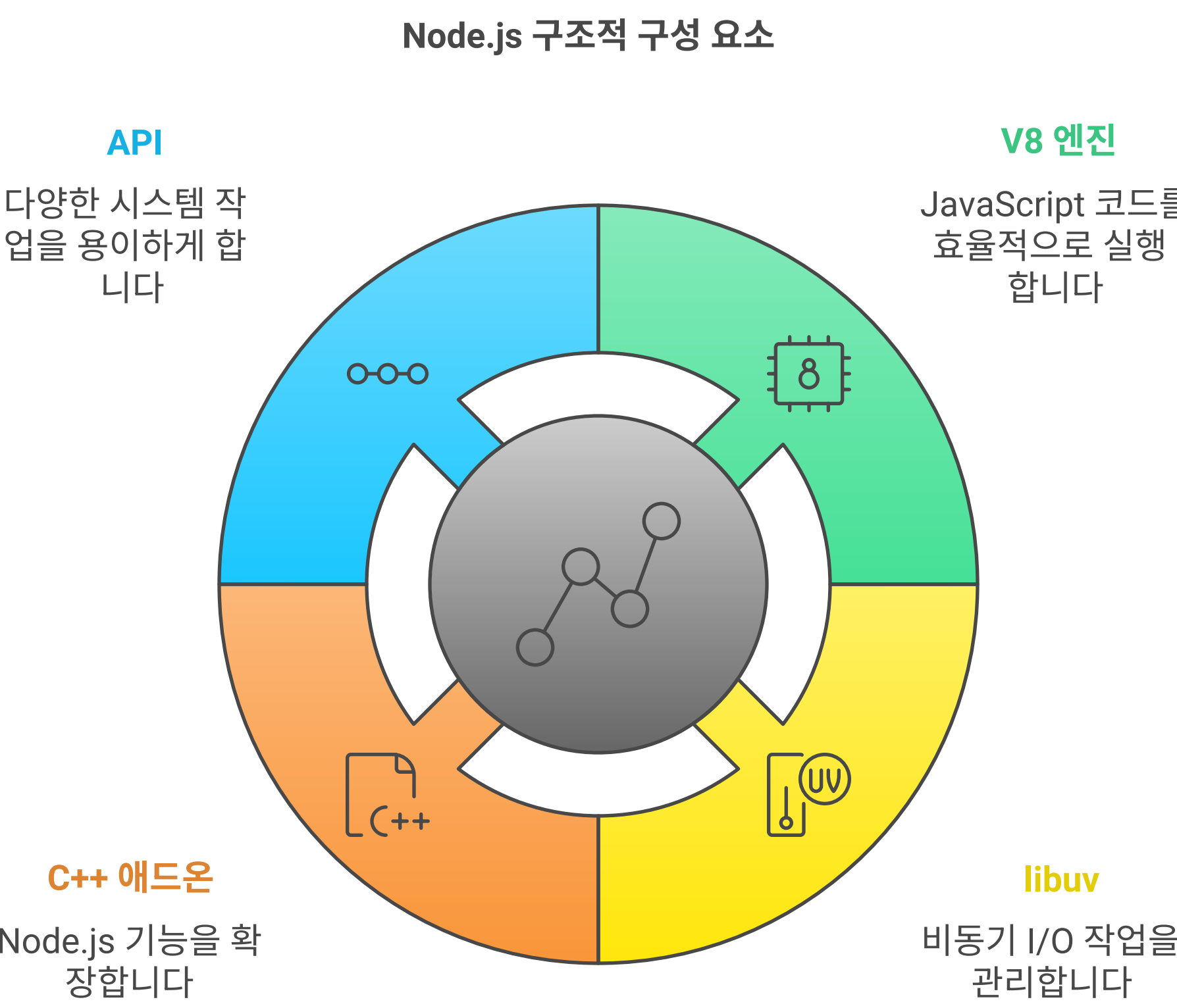
- 비동기 I/O: Node.js는 비동기 방식으로 I/O 작업을 처리하여 높은 처리량을 제공합니다.
- 이벤트 기반: 이벤트 루프를 통해 비동기 작업을 관리하며, 콜백 함수를 사용하여 작업 완료 시 실행할 코드를 정의합니다.
- 모듈 시스템: CommonJS 모듈 시스템을 사용하여 코드의 재사용성을 높이고, 다양한 패키지를 쉽게 관리할 수 있습니다.
- NPM**: Node Package Manager(NPM)를 통해 수많은 오픈 소스 라이브러리와 패키지를 쉽게 설치하고 관리할 수 있습니다.



## 3. Node.js 구조

Node.js의 구조는 다음과 같은 주요 구성 요소로 이루어져 있습니다:

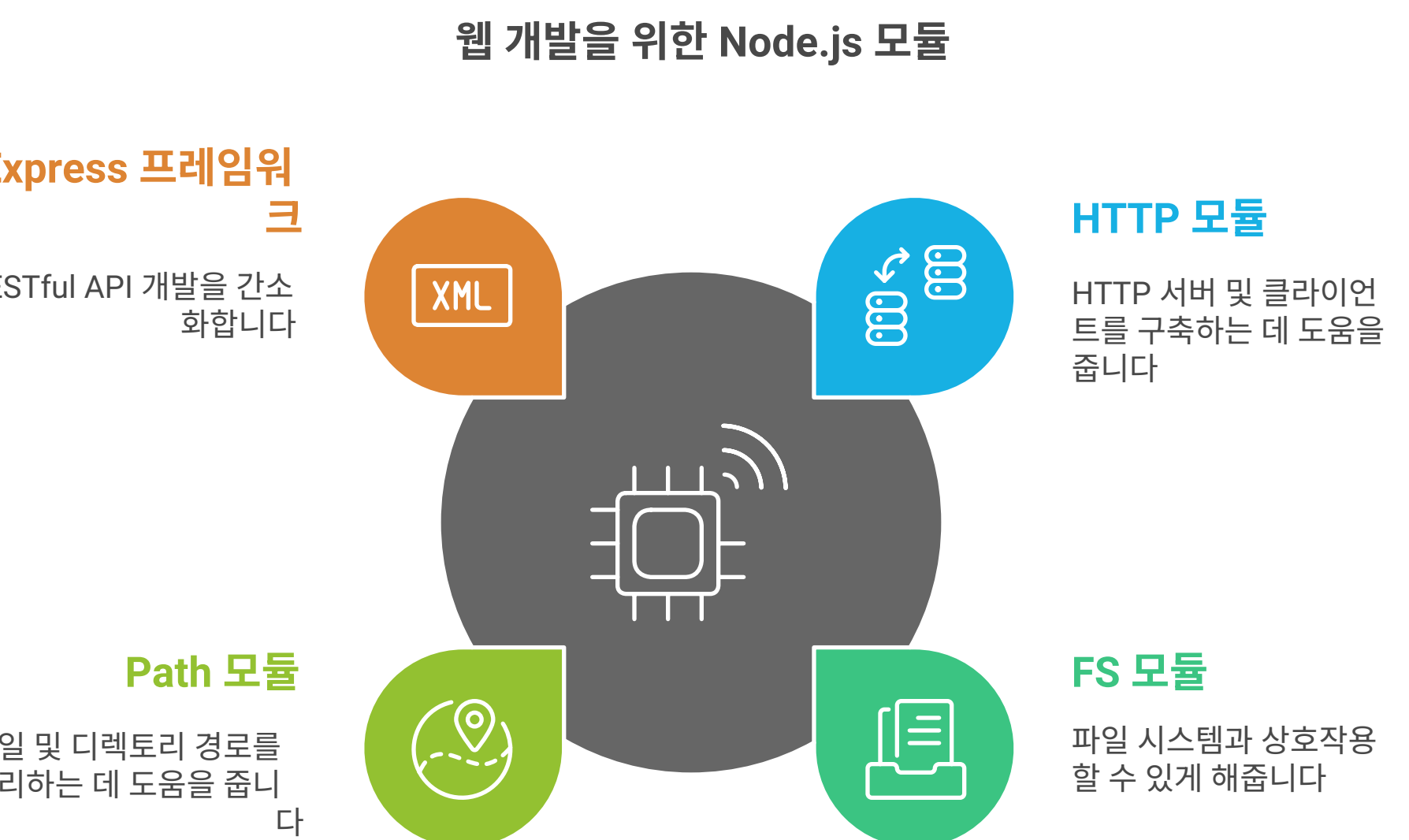
- V8 엔진**: JavaScript 코드를 실행하는 엔진입니다.
- libuv**: 비동기 I/O 작업을 처리하는 라이브러리로, 이벤트 루프와 스레드 풀을 관리합니다.
- C++ 애드온**: Node.js의 기능을 확장하기 위해 C++로 작성된 모듈입니다.
- API**: Node.js에서 제공하는 다양한 API를 통해 파일 시스템, 네트워크, HTTP 등 다양한 작업을 수행할 수 있습니다.



## 4. 주요 모듈

Node.js는 여러 내장 모듈을 제공합니다. 주요 모듈은 다음과 같습니다:

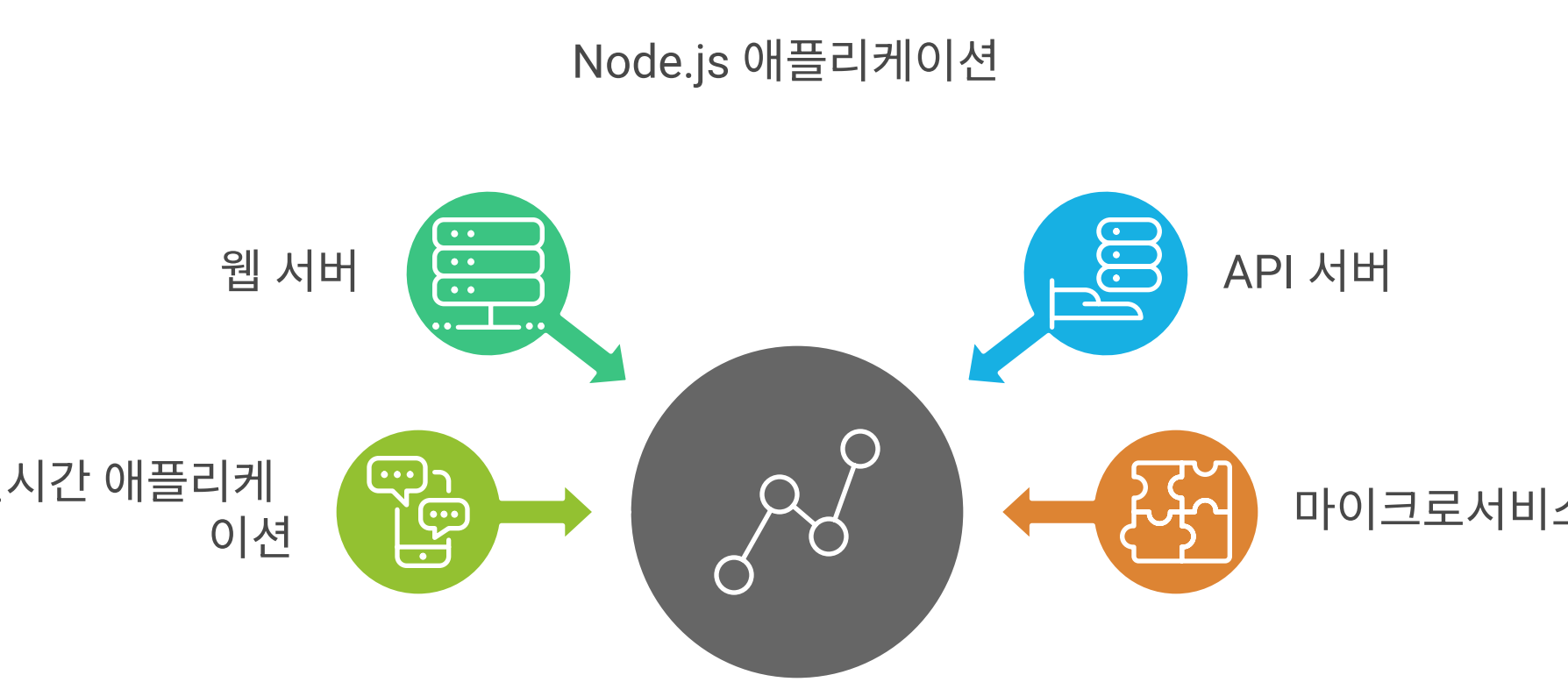
- http**: HTTP 서버 및 클라이언트를 구축하는 데 사용됩니다.
- fs**: 파일 시스템과 상호작용하기 위한 모듈입니다.
- path**: 파일 및 디렉토리 경로를 다루는 데 유용한 모듈입니다.
- express**: 웹 애플리케이션 프레임워크로, RESTful API를 쉽게 구축할 수 있습니다.



## 5. 사용 사례

Node.js는 다음과 같은 다양한 분야에서 사용됩니다:

- 웹 서버: 고성능 웹 서버를 구축할 수 있습니다.
- API 서버**: RESTful API를 쉽게 만들 수 있습니다.
- 실시간 애플리케이션: 채팅 애플리케이션, 게임 서버 등 실시간 데이터 전송이 필요한 애플리케이션에 적합합니다.
- 마이크로서비스: 마이크로서비스 아키텍처를 구현하는 데 유용합니다.



## 결론

Node.js는 비동기 이벤트 기반 모델을 통해 높은 성능과 확장성을 제공하는 JavaScript 런타임입니다. 다양한 내장 모듈과 NPM을 통해 개발자들은 효율적으로 서버 사이드 애플리케이션을 구축할 수 있습니다. Node.js의 특징과 구조를 이해하면, 다양한 웹 애플리케이션 및 서비스 개발에 큰 도움이 될 것입니다.