

IoT.js

iotjs.net

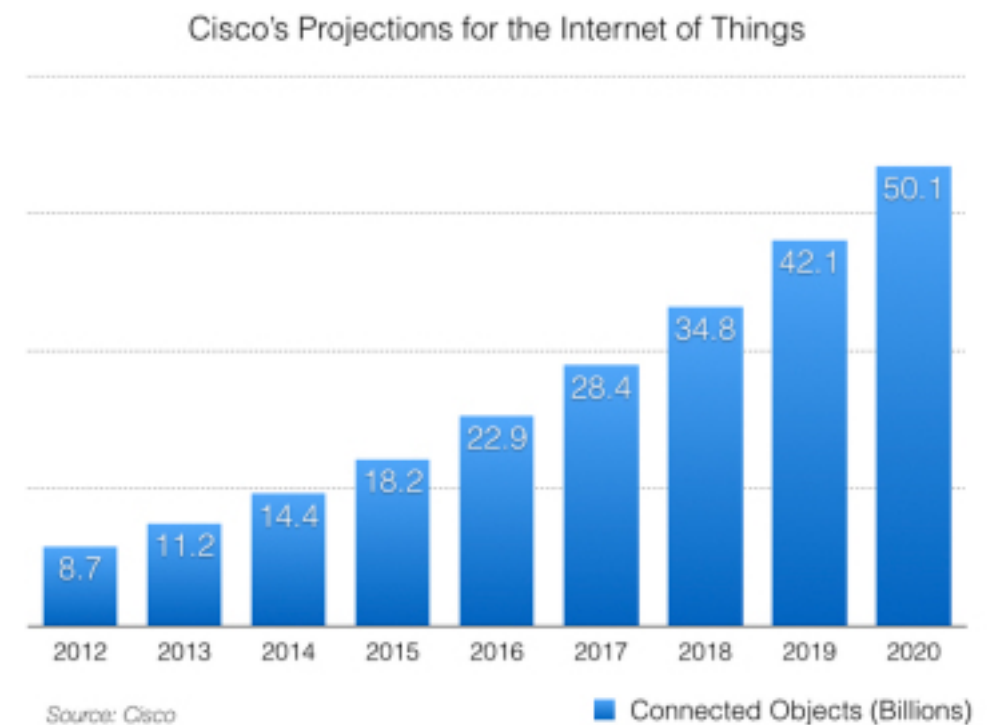
- 사물인터넷 *Internet of Things*
- IoT.js 소개
- IoT.js 를 이용한 프로그래밍

사물인터넷(IoT)

- 사물인터넷(Internet of Things)
 - 사물간의 네트워크
 - “The Internet of Things is the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment.” - Gatner

사물인터넷 기기 수

- 2020년 추정치
- 500억개 - 시스코
- 750억개 - 인텔



IoT.js?

- IoT 애플리케이션 개발을 위한 플랫폼
 - 자바스크립트 기반 프로그래밍 개발 환경
- 오픈 소스
 - iotjs.net
 - github.com/samsung/iotjs

node.js

- Javascript 기반 웹 서버 플랫폼
 - 단일 쓰레드, non-block I/O
 - npm
- 오픈소스
 - nodejs.org
 - github.com/joyent/node

node.js web server

```
var http = require('http');  
http.createServer(function (req, res) {  
    res.writeHead(200, { 'Content-Type': 'text/  
plain' });  
    res.end('Hello World\n');  
}).listen(1337, '127.0.0.1');  
console.log('Server running at http://  
127.0.0.1:1337/');
```


IoT.js vs node.js

- 하위 호환성 (backward compatibility)
 - IoT.js 에서 동작하는 프로그램은 node.js 에서 동작한다.
- 역은 성립하지 않음

왜 하필 IoT.js?

- 자바스크립트 개발자도 IoT 앱 개발하고 싶다.
 - node.js가 좋은 대안!!
- IoT 디바이스의 리소스 제약
 - node.js를 올리기 벅찬 환경ㅠㅠ
- 또 다른 대안
 - IoT.js !!

IoT.js vs node.js

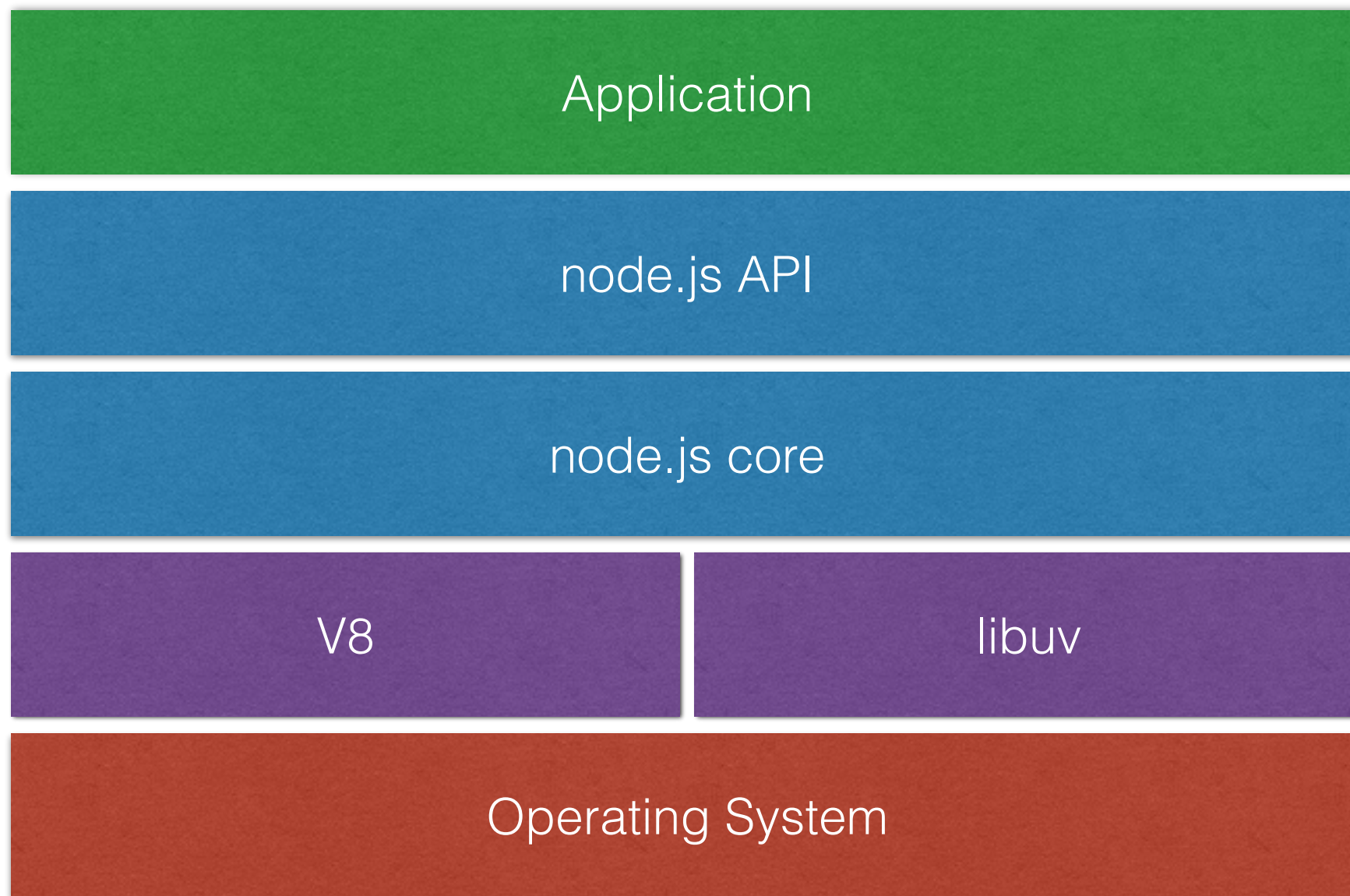
2nd round

- node.js
 - 크고 아름다운 고성능의 V8 자바스크립트 엔진
 - >1MB RAM
- IoT.js
 - 느리지만 작고 귀여운 JerryScript 자바스크립트 엔진
 - <64KB RAM

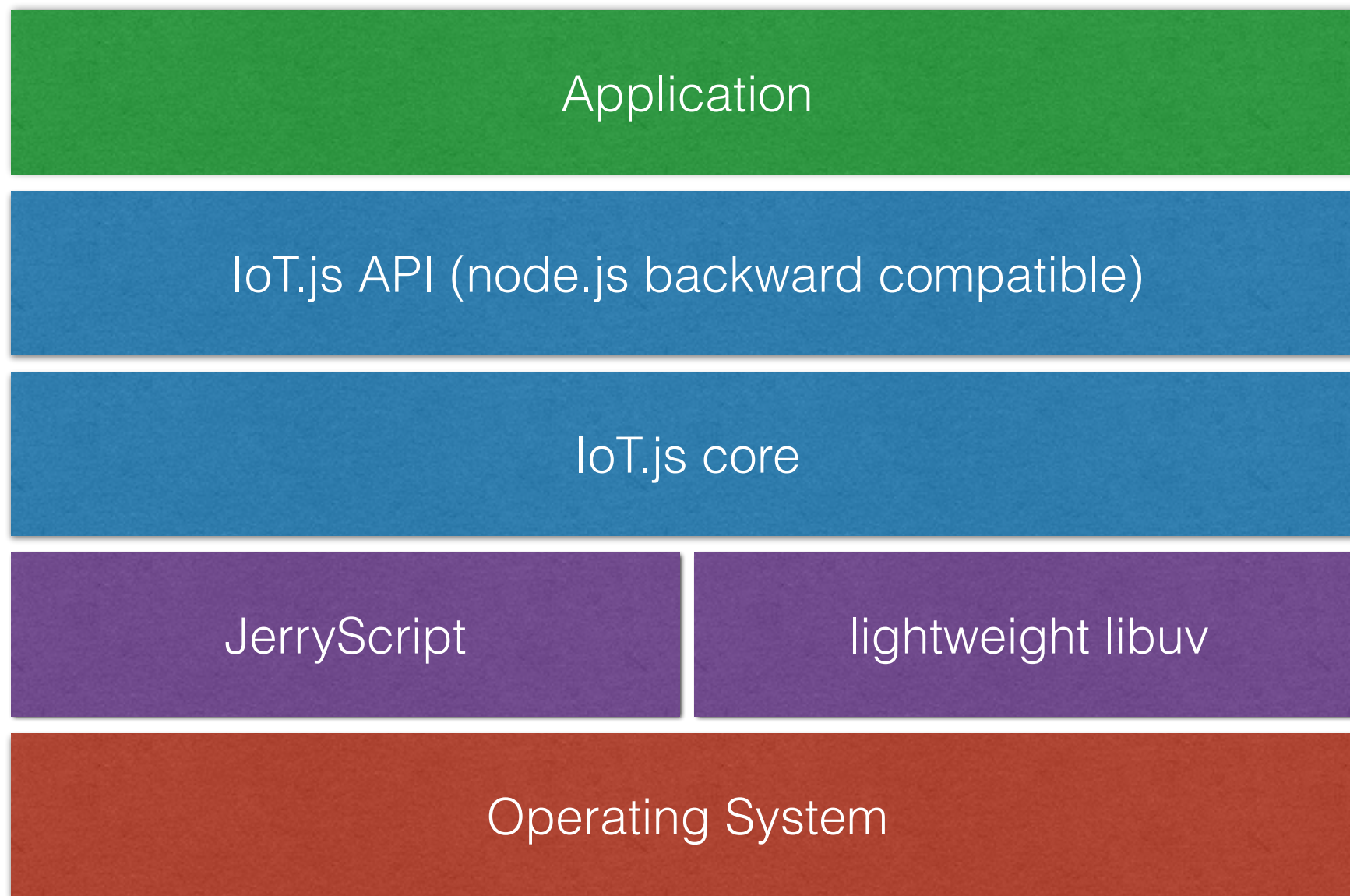
JerryScript

- 자바스크립트 엔진
- 성능 보다 가벼움을 추구
- 오픈소스
- jerryscript.net
- github.com/Samsung/jerryscript

node.js 구조



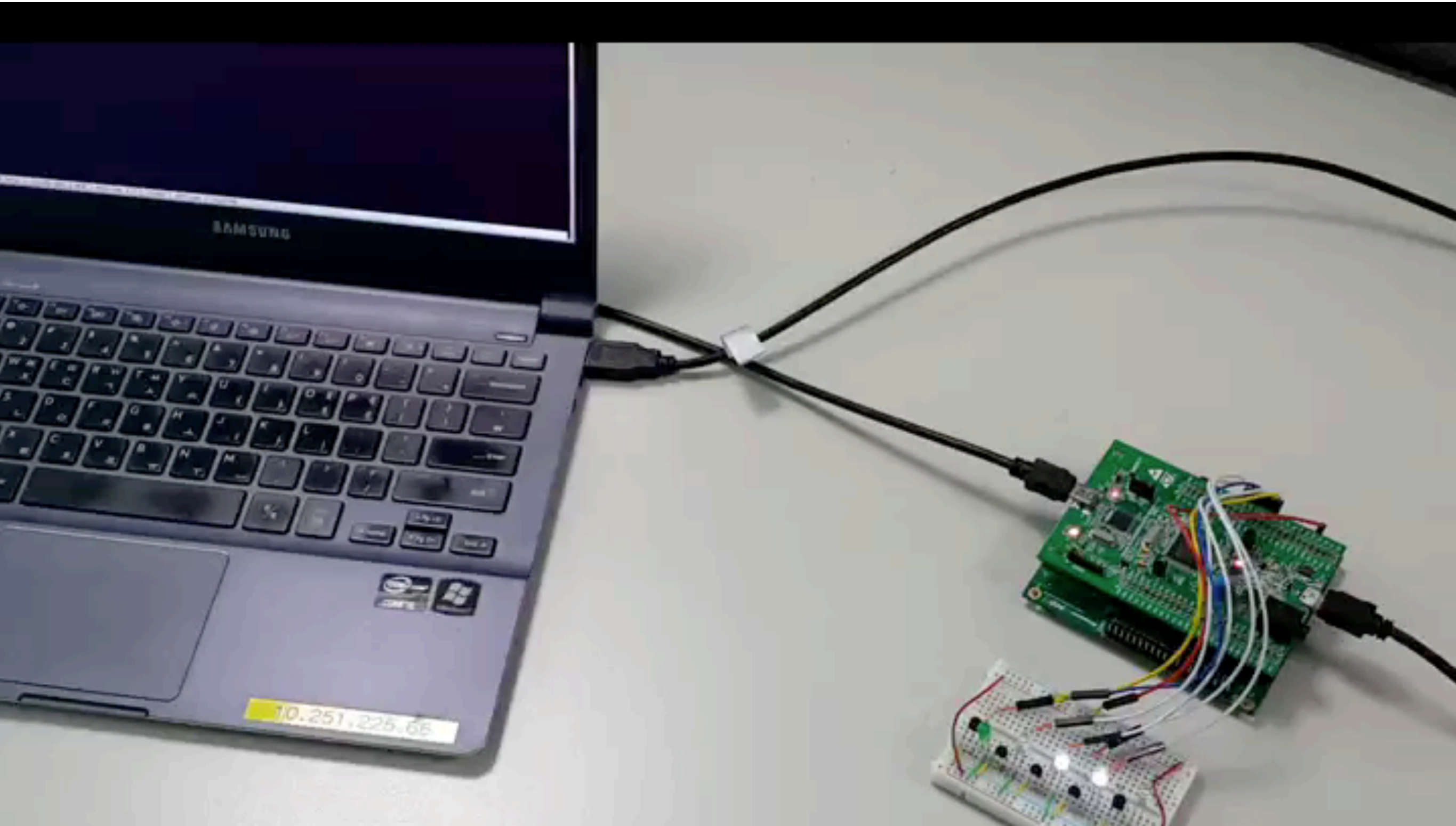
IoT.js 구조



IoT.js API

- node.js backward compatible
 - Process
 - Console
 - Filesystem
 - Network
 - HTTP
- Device Control
 - GPIO
- IoT Connectivity (예정)

IoT.js 데모



IoT.js 프로그래밍

LED

```
var gpio = require('gpio');
var LED1 = 1; // gpio pin number for LED1
var LED2 = 2; // gpio pin number for LED2
var toggle = 1;
// blink LEDs every 500ms.
setInterval(function() {
  gpio.writePin(LED1, toggle);
  gpio.writePin(LED2, 1 - toggle);
  toggle = 1 - toggle;
}, 500);
```


IoT.js 프로그래밍

온도계 서버 (1)

```
var gpio = require('gpio');
```

```
// GPIO port number for thermometer
```

```
var thermometerPort = 1;
```

```
exports.getTemperatur = function(callback) {  
  gpio.readPort(thermometerPort, callback);  
};
```

IoT.js 프로그래밍

온도계 서버 (2)

```
var http = require('http');
var thermometer = require('thermometer');

http.createServer(function(req, res) {
  thermometer.getTemperature(function(err, value) {
    if (err) {
      res.writeHead(500);
      res.end(JSON.stringify({error: err.toString()}));
    } else {
      res.writeHead(200);
      res.end(JSON.stringify({temperature: value}));
    }
  });
}).listen(8888);
```

Q&A

감사합니다